

Лабораторная Работа №1

Простые модели компьютерной сети

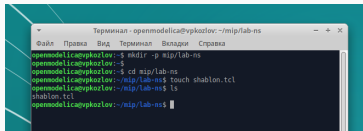
Козлов В.П.

Российский университет дружбы народов им. Патриса Лумумбы, Москва, Россия

- Козлов Всеволод Павлович
- НФИбд-02-22
- Российский университет дружбы народов
- [1132226428@pfur.ru]

Выполнение лабораторной работы

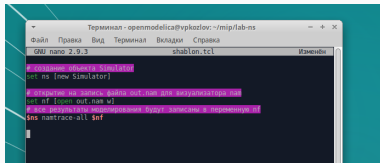
Создал директорию и файл шаблона



```
Терминал · orenmodelica@virkozlov: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
orenmodelica@virkozlov:~$ mkdir -p mip/lab-ns
orenmodelica@virkozlov:~$ cd mip/lab-ns
orenmodelica@virkozlov:~/mip/lab-ns$ touch shablon.tcl
orenmodelica@virkozlov:~/mip/lab-ns$ ls
shablon.tcl
orenmodelica@virkozlov:~/mip/lab-ns$
```

Рис. 1: Файл шаблона

Создал переменную `nf` и указал, что требуется открыть на запись пат-файл для регистрации выходных результатов моделирования



```
Терминал - openmodelica@vypkozlov: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 2.9.3  shablon.tcl

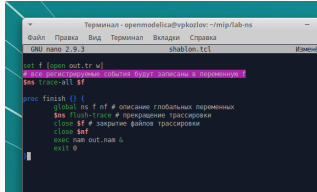
# Создание объекта Simulator
set ns [new Simulator]

# Открытие на запись файла out.nam для сохранения данных
set nf [open out.nam w]

# Все результаты моделирования будут записаны в переменную nf
$ns namtrace all $nf
```

Рис. 2: Переменная `nf`

Создал переменную `f` и открыл на запись файл трассировки для регистрации всех событий модели. Написал процедуру `finish`



```
Терминал - orenmodelica@yurkov: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 2.9.3  shablon.tcl

set f {open out.tr w}
$ns flush-trace # прекращение трассировки
close $f # закрытие файла трассировки
close $nf
echo nam out.nam &
exit 0

proc finish {} {
    global ns f nf # описание глобальных переменных
    $ns flush-trace # прекращение трассировки
    close $f # закрытие файла трассировки
    close $nf
    echo nam out.nam &
    exit 0
}
```

Рис. 3: Переменная `f`. Процедура `finish`

С помощью команды `at` указал планировщику событий, что процедуру `finish` следует запустить через 5 с после начала моделирования, после чего запустить симулятор `ns`



```
ns at 5.0 "finish"  
# ns-3.26.0  
ns run  
█
```

Рис. 4: Запуск процедуры `finish`

Сохранил изменения и запустил программу

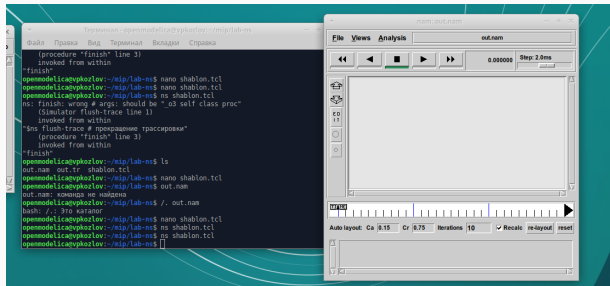
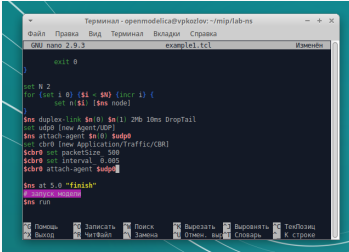


Рис. 5: Запуск программы

Создал example1.tcl. Написал программу



```
Терминал - openmodelica@vpkozlov: ~/mip/lab-ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
GNU nano 2.9.3  example1.tcl
)
    exit 0
}

set N 2
for {set i 0} {$i < $N} {incr i} {
    set n$i [new node]
}

$nns duplex-link $n0 $n1 2Mb 10ms DropTail
set udp0 [new Agent/UDP]
$nns attach-agent $n0 $udp0
set cbr0 [new Application/Traffic/CBR]
$cbro set packetSize 500
$cbro set interval 0.005
$cbro attach-agent $udp0

$nns at 5.0 "finish"
# $nns stop
$nns run
```

Рис. 6: Редактирование example1.tcl

Написал программу (часть 2)

```
set null0 [new Agent/hull]
$ns attach-agent $n1 $null0
$ns connect $udp0 $null0
$ns at 0.5 "$scr0 start"
$ns at 4.5 "$scr0 stop"
}
$ns at 5.0 "finish"
#-janyx 000000
$ns run
```

Рис. 7: Редактирование example1.tcl

Запустил код программы example1.tcl. Просмотрел движение пакетов данных

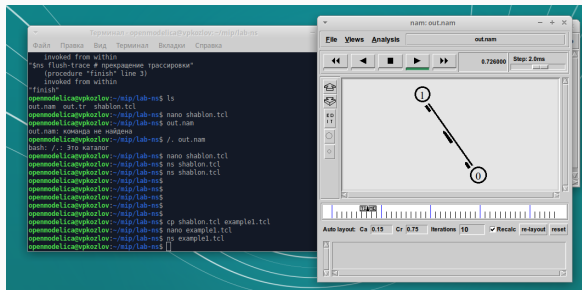
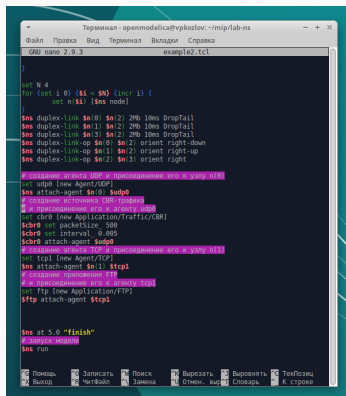


Рис. 8: Запуск программы

Создал новый файл example2.tcl. В нем создал 4 узла и 3 дуплексных соединения с указанием направления



```
Терминал - orelmodelica@vprkzlov: ~/mlp/lab-ns
Файл Правка Вид Терминал Выход Справка
GNU nano 2.9.3 example2.tcl

)
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 n0
set udp0 [new Agent/UDP]
$ns attach-agent $n(0) $udp0
# создание источника CBR-трафика
# и присоединение его к агенту UDP
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 n1
set tcp1 [new Agent/TCP]
$ns attach-agent $n(1) $tcp1
# создание приложения FTP
# и присоединение его к агенту TCP
set ftp [new Application/FTP]
$ftp attach-agent $tcp1

$ns at 5.0 "finish"
# завершение работы
$ns run
```

Рис. 9: Редактирование example2.tcl

Создал агенты-получатели. Соединил агенты udr0 и tcp1 и их получателей

```
Терминал - openmodelica@yurkovlev: ~/mip/lab/ns
Файл  Правка  Вид  Терминал  Вкладки  Справка
OM2 nano 2.9.3 example2.tcl  Изменен

set cbr0 [new Application/Traffic/CBR]
$cbr0 set packetSize 500
$cbr0 set interval 0.005
$cbr0 attach-agent $udr0
# создание агента tcp1 и его получателя $tcp1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
# создание приложения ftp и его получателя $sftp
set ftp [new Application/FTP]
$ftp attach-agent $tcp1
# создание агента null0 и его получателя
set null0 [new Agent/Null]
$ns attach-agent $n3 $null0
# создание агента sink1 и его получателя
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $udr0 $null0
$ns connect $tcp1 $sink1
$ns color 1 Blue
$ns color 2 Red
$udr0 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"
$ns at 1.0 "$sftp start"
$ns at 4.0 "$sftp stop"
$ns at 4.5 "$cbr0 stop"

$ns at 5.0 "finish"
# запускаем модель
$ns run
```

Рис. 10: Редактирование example2.tcl

Запустил код программы example2.tcl. Просмотрел движение пакетов данных

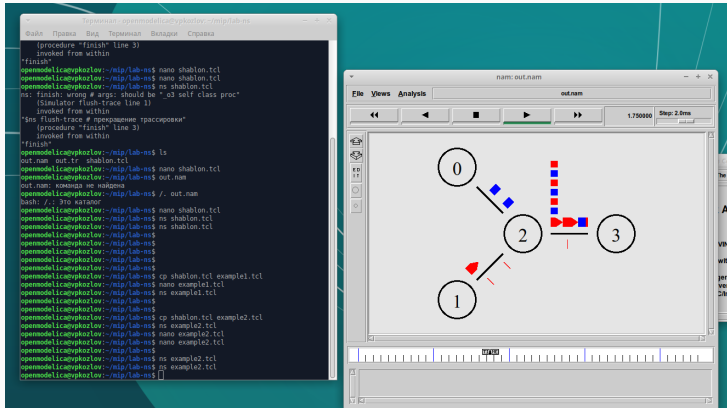
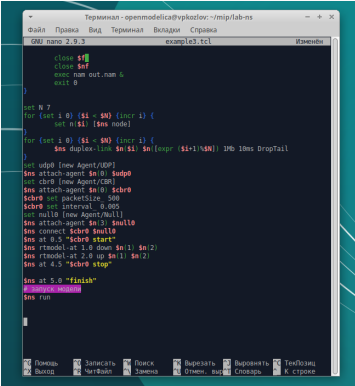


Рис. 11: Запуск программы

Создал новый файл example2.tcl. Написал первую часть программы



```
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} {
        $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)
    $ns rtmodel-at 2.0 up $n(1) $n(2)
    $ns at 4.5 "$cbr0 stop"

    $ns at 5.0 "finish"
    # $nsack monitor
    $ns run
}
```

Рис. 12: Редактирование example3.tcl

Запустил код программы `example2.tcl`. Просмотрел движение пакетов данных в случае разрыва соединения

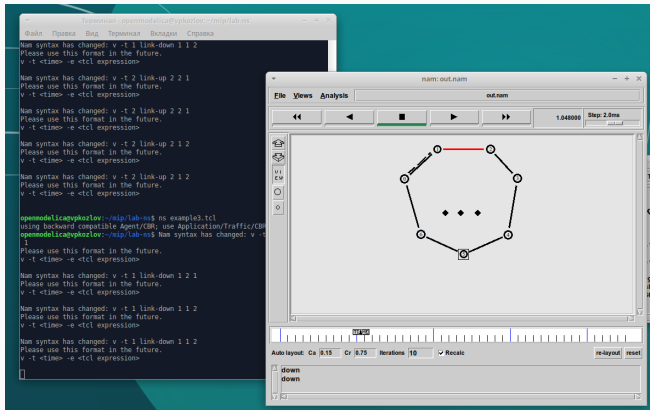


Рис. 13: Запуск программы

Просмотрел движение пакетов данных с использованием команды \$ns rtproto DV

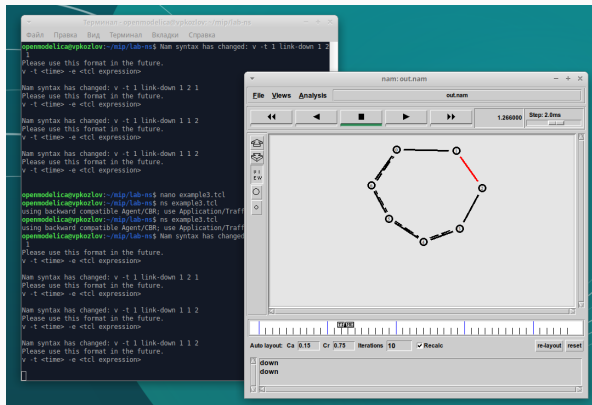
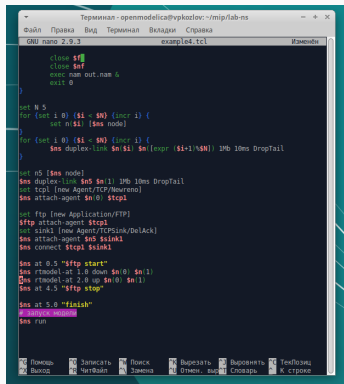


Рис. 14: Редактирование example3.tcl

Написал код для программы из Упражнения



```
Терминал - openmodelica@vprkzlov:~/mip/lab-ns
Файл Правка Вид Терминал Вкладки Справка
GNU nano 2.9.3 example4.tcl

    close $f
    close $nf
    exec nam out.nam &
    exit 0
}

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}] 1Mb 10ms DropTail
}

set n5 [$ns node]
$ns duplex-link $n5 $n(1) 1Mb 10ms DropTail
set tcp1 [new Agent/TCP/Neurono]
$ns attach-agent $n(0) $tcp1

set ftp [new Application/FTP]
$ftp 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"
# $nsack module
$ns run
```

Рис. 15: Код программы из Упражнения

Запустил программу. Вначале пакеты идут по кратчайшему пути

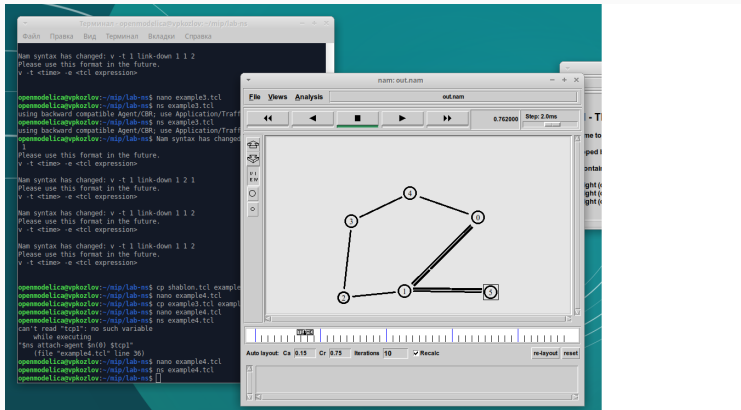


Рис. 16: Движение пакетов по кратчайшему пути

Движение пакетов в случае разрыва соединения

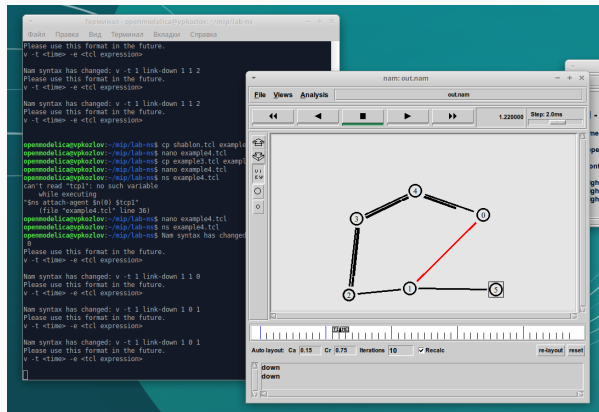


Рис. 17: Движение пакетов в случае разрыва соединения

Пакеты снова идут по кратчайшему пути

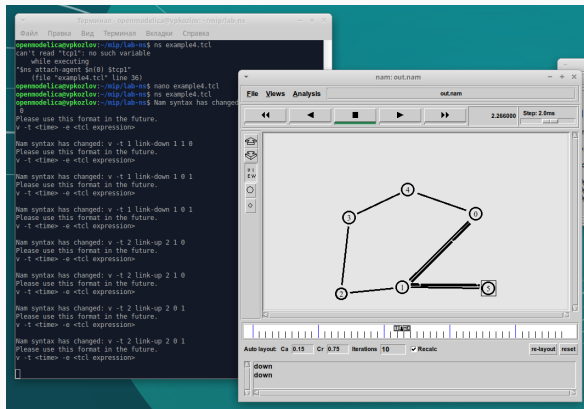


Рис. 18: Движение пакетов по кратчайшему пути

Приобрел навыки моделирования сетей передачи данных с помощью средств имитационного моделирования NS-2, а также анализа полученных результатов моделирования.