Санкт-Петербургский Политехнический Университет Петра Великого

Институт металлургии, машиностроения и транспорта

Кафедра «Мехатроника и роботостроение»

Отчет по лабораторной работе

Дисциплина «Сети ЭВМ»

Тема: «Графическое отображение результатов моделирования. Утилита Xgraph»

Выполнил:

Студент гр. 33328/1

Е. Э. Хомутов

Санкт-Петербург

2018

Цель работы

* ознакомление с работой утилиты графического отображения данных Xgraph;
* ознакомление с методами мониторинга трафика в симуляторе ns2.

Программа работы

1. Создайте файл work4\_0.tcl с приведенным в пункте 3.1 текстом скрипта и запустите его на моделирование. Ознакомьтесь с особенностями работы пакета xgraph.

**Текст скрипта work4\_0.tcl**

set ns [new Simulator]

set f0 [open out0.tr w]

set f1 [open out1.tr w]

set f2 [open out2.tr w]

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

$ns duplex-link $n0 $n3 1Mb 100ms DropTail

$ns duplex-link $n1 $n3 1Mb 100ms DropTail

$ns duplex-link $n2 $n3 1Mb 100ms DropTail

$ns duplex-link $n3 $n4 1Mb 100ms DropTail

proc finish {} {

global f0 f1 f2

close $f0

close $f1

close $f2

exec xgraph out0.tr out1.tr out2.tr -geometry 1920x1280 \

-0 source0 -1 source1 -2 source2 &

exit 0

}

proc attach-expoo-traffic { node sink size burst idle rate } {

set ns [Simulator instance]

set source [new Agent/CBR/UDP]

$ns attach-agent $node $source

set traffic [new Traffic/Expoo]

$traffic set packet-size $size

$traffic set burst-time $burst

$traffic set idle-time $idle

$traffic set rate $rate

$source attach-traffic $traffic

$ns connect $source $sink

return $source

}

proc record {} {

global sink0 sink1 sink2 f0 f1 f2

set ns [Simulator instance]

set time 0.5

set bw0 [$sink0 set bytes\_]

set bw1 [$sink1 set bytes\_]

set bw2 [$sink2 set bytes\_]

set now [$ns now]

puts $f0 "$now [expr $bw0/$time\*8/1000000]"

puts $f1 "$now [expr $bw1/$time\*8/1000000]"

puts $f2 "$now [expr $bw2/$time\*8/1000000]"

$sink0 set bytes\_ 0

$sink1 set bytes\_ 0

$sink2 set bytes\_ 0

$ns at [expr $now+$time] "record"

}

set sink0 [new Agent/LossMonitor]

set sink1 [new Agent/LossMonitor]

set sink2 [new Agent/LossMonitor]

$ns attach-agent $n4 $sink0

$ns attach-agent $n4 $sink1

$ns attach-agent $n4 $sink2

set source0 [attach-expoo-traffic $n0 $sink0 200 2s 1s 100k]

set source1 [attach-expoo-traffic $n1 $sink1 200 2s 1s 200k]

set source2 [attach-expoo-traffic $n2 $sink2 200 2s 1s 300k]

$ns at 0.0 "record"

$ns at 10.0 "$source0 start"

$ns at 10.0 "$source1 start"

$ns at 10.0 "$source2 start"

$ns at 50.0 "$source0 stop"

$ns at 50.0 "$source1 stop"

$ns at 50.0 "$source2 stop"

$ns at 60.0 "finish"

$ns run

1. В соответствии с заданием пункта 3.2 произведите моделирование работы сети с заданной топологией. Скрипт сохраните в файле work4\_1.tcl.

**Текст скрипта work4\_1.tcl**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

set f [open out0.tr w]

$ns color 1 Green

$ns color 2 Red

for {set index 1} {$index <= 3} {incr index} {

set s($index) [$ns node]

}

for {set index 1} {$index <= 5} {incr index} {

set r($index) [$ns node]

}

for {set index 1} {$index <= 3} {incr index} {

set k($index) [$ns node]

}

set udp(1) [new Agent/CBR/UDP]

$ns attach-agent $s(1) $udp(1)

$udp(1) set fid\_ 1

set traffic [new Traffic/Expoo]

$traffic set packet-size 300

$traffic set burst-time 0.1s

$traffic set idle-time 0.1s

$traffic set rate 150k

$udp(1) attach-traffic $traffic

set null(1) [new Agent/Null]

$ns attach-agent $k(1) $null(1)

$ns connect $udp(1) $null(1)

set udp(2) [new Agent/CBR/UDP]

$ns attach-agent $s(2) $udp(2)

$udp(2) set fid\_ 2

set traffic [new Traffic/Expoo]

$traffic set packet-size 300

$traffic set burst-time 0.1s

$traffic set idle-time 0.1s

$traffic set rate 250k

$udp(2) attach-traffic $traffic

set null(2) [new Agent/Null]

$ns attach-agent $k(2) $null(2)

$ns connect $udp(2) $null(2)

$ns duplex-link $s(1) $r(1) 128Kb 20ms DropTail

$ns duplex-link $s(2) $r(1) 128Kb 20ms DropTail

$ns duplex-link $s(3) $r(1) 1Mb 100ms DropTail

$ns duplex-link $r(1) $r(2) 128Kb 20ms DropTail

$ns duplex-link $r(1) $r(4) 1Mb 100ms DropTail

$ns duplex-link $r(2) $r(3) 1Mb 100ms DropTail

$ns duplex-link $r(3) $k(1) 1Mb 100ms DropTail

$ns duplex-link $r(4) $r(5) 1Mb 100ms DropTail

$ns duplex-link $r(3) $r(5) 1Mb 100ms DropTail

$ns duplex-link $r(5) $k(3) 1Mb 100ms DropTail

$ns duplex-link $r(5) $k(2) 1Mb 100ms DropTail

$ns duplex-link-op $r(1) $s(1) orient left-up

$ns duplex-link-op $r(1) $s(2) orient left

$ns duplex-link-op $r(1) $r(2) orient right-up

$ns duplex-link-op $r(1) $s(3) orient left-down

$ns duplex-link-op $r(1) $r(4) orient right-down

$ns duplex-link-op $r(2) $r(3) orient right

$ns duplex-link-op $r(3) $k(1) orient right

$ns duplex-link-op $r(4) $r(5) orient right

$ns duplex-link-op $r(3) $r(5) orient down

$ns duplex-link-op $r(5) $k(2) orient right

$ns duplex-link-op $r(5) $k(3) orient down

$ns duplex-link-op $r(1) $r(2) queuePos 0.5

$ns queue-limit $r(1) $r(2) 30

set qm [$ns monitor-queue $r(1) $r(2) [$ns get-ns-traceall]]

proc trqueue {} {

global qm f

set ns [Simulator instance]

set time 0.1

set q(0) [$qm set pkts\_]

set now [$ns now]

puts $f "$now $q(0)"

$qm reset

$ns at [expr $now+$time] "trqueue"

}

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec xgraph out0.tr -geometry 800x600 &

exit 0

}

$ns at 0.0 "trqueue"

$ns at 0.1 "$udp(1) start"

$ns at 0.1 "$udp(2) start"

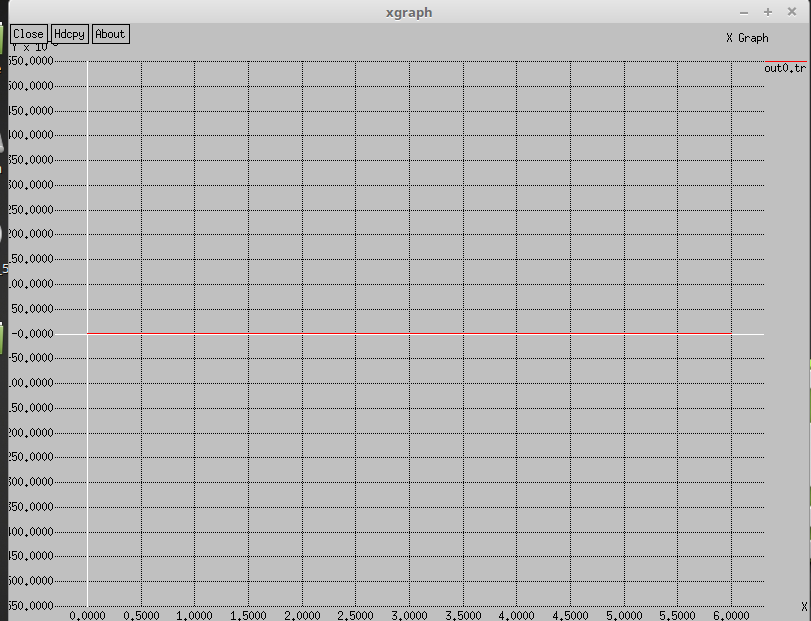
$ns at 5.0 "$udp(1) stop"

$ns at 5.0 "$udp(2) stop"

$ns at 6.0 "finish"

$ns run

Очередь не образуется:



1. Произведите модификацию топологии в соответствии с заданием пункта 3.2, результат сохраните в файле work4\_2.tcl.

…

proc trqueue {} {

global qm f

set ns [Simulator instance]

set time 0.1

set q(0) [$qm set pkts\_]

set q(1) [$qm set parrivals\_]

set q(2) [$qm set pdrops\_]

set now [$ns now]

puts $f(0) "$now $q(0)"

puts $f(1) "$now $q(1)"

puts $f(2) "$now $q(2)"

$qm reset

$ns at [expr $now+$time] "trqueue"

}

proc finish {} {

global ns nf

$ns flush-trace

close $nf

set tt "Work4-2"

set tx "time(sec)"

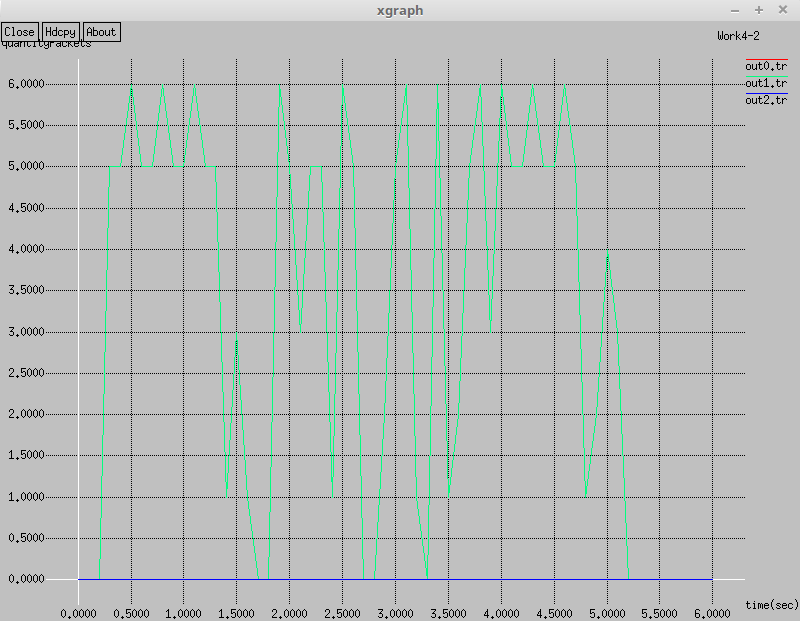
set ty "quantityPackets"

exec xgraph -geometry 800x600 -t $tt -x $tx -y $ty out0.tr out1.tr out2.tr &

exit 0

}

…



1. Создайте скрипт содержащий команды обработки выходного файла симулятора пункта 4.1 (файл work4\_3.tcl), ознакомьтесь с приведенными в примере командами.

proc finish {label mod} {

exec rm -f temp.rands

set f [open temp.rands w]

puts $f "TitleText: $label"

puts $f "Device: Postscript"

exec rm -f temp.p

exec touch temp.p

exec awk {

{

if (($1 == "+" || $1 == "-") && ($5 == "exp")) \

print $2, $8 \* (mod + 10) + ($11 % mod)

}

} mod=$mod out0.tr > temp.p

exec rm -f temp.d

exec touch temp.d

exec awk {

{

if ($1 == "d") \

print $2, $8 \* (mod + 10) + ($11 % mod)

}

} mod=$mod out0.tr > temp.d

puts $f \"enque/deque

#flush $f

exec cat temp.p >@ $f

#flush $f

puts $f \n\"drops

#flush $f

#exec head -1 temp.d >@ $f

exec cat temp.d >@ $f

close $f

set tx "time (sec)"

set ty "packet number (mod $mod)"

exec xgraph -bb -tk -nl -m -zg 0 -x $tx -y $ty temp.rands &

exit 0

}

proc attach-expoo-traffic { node sink size burst idle rate } {

set ns [Simulator instance]

set source [new Agent/CBR/UDP]

$ns attach-agent $node $source

set traffic [new Traffic/Expoo]

$traffic set packet-size $size

$traffic set burst-time $burst

$traffic set idle-time $idle

$traffic set rate $rate

$source attach-traffic $traffic

$ns connect $source $sink

return $source

}

set ns [new Simulator]

set label "Expoo\_Traffic"

set mod 50

$ns color 0 Blue

$ns color 1 Red

for {set index 0} {$index <= 3} {incr index} {

set n($index) [$ns node]

}

$ns duplex-link $n(0) $n(2) 1Mb 100ms DropTail

$ns duplex-link $n(1) $n(2) 1Mb 100ms DropTail

$ns duplex-link $n(2) $n(3) 128kb 100ms DropTail

#$ns queue-limit $n2 $n3 10

exec rm -f out0.tr

set fout [open out0.tr w]

set sink(0) [new Agent/Null]

set sink(1) [new Agent/Null]

$ns attach-agent $n(3) $sink(0)

$ns attach-agent $n(3) $sink(1)

$ns queue-limit $n(2) $n(3) 15

$ns trace-queue $n(2) $n(3) $fout

set source(0) [attach-expoo-traffic $n(0) $sink(0) 500 0.1s 0.1s 150k]

set source(1) [attach-expoo-traffic $n(1) $sink(1) 500 0.1s 0.1s 250k]

$source(0) set fid\_ 0

$source(1) set fid\_ 1

$ns at 0.1 "$source(0) start"

$ns at 0.1 "$source(1) start"

$ns at 2.5 "$source(0) stop"

$ns at 2.5 "$source(1) stop"

$ns at 3.0 "ns flush-trace; close $fout;\

finish $label $mod"

$ns run

1. Создайте скрипт в соответствии с заданием пункта 4.2., сохраните его в виде файла work4\_4.tcl, и запустите на моделирование. Проконтролируйте корректное отображение данных всех трех источников трафика.

exec rm -f out0.tr

set fout [open out0.tr w]

$ns color 1 Blue

$ns color 2 Red

$ns color 3 Black

for {set index 1} {$index <= 3} {incr index} {

set s($index) [$ns node]

}

for {set index 1} {$index <= 5} {incr index} {

set r($index) [$ns node]

}

for {set index 1} {$index <= 3} {incr index} {

set k($index) [$ns node]

}

set sink(1) [new Agent/Null]

set sink(2) [new Agent/Null]

set sink(3) [new Agent/Null]

$ns attach-agent $k(1) $sink(1)

$ns attach-agent $k(2) $sink(2)

$ns attach-agent $k(3) $sink(3)

set source(1) [attach-expoo-traffic $s(1) $sink(1) 500 0.1s 0.1s 150k]

set source(2) [attach-expoo-traffic $s(2) $sink(2) 500 0.1s 0.1s 250k]

set source(3) [attach-expoo-traffic $s(3) $sink(3) 1000 0.1s 0.1s 100k]

...

$ns queue-limit $r(1) $r(2) 15

$ns trace-queue $r(1) $r(2) $fout

$source(1) set fid\_ 0

$source(2) set fid\_ 1

$source(3) set fid\_ 2

$ns at 0.1 "$source(1) start"

$ns at 0.1 "$source(2) start"

$ns at 0.1 "$source(3) start"

$ns at 2.5 "$source(1) stop"

$ns at 2.5 "$source(2) stop"

$ns at 2.5 "$source(3) stop"

$ns at 3.0 "ns flush-trace; close $fout; finish $label $mod"

$ns run