**Networking Lab**

**Lab Assignment No 5**

**Name:** Iyer Sreekesh Subramanian **Roll No**. 24

**Aim**:

Graphical simulation of a network with Routing Protocol (Distance Vector/ Link State Routing) using NAM and plot the graph of traffic consideration (TCP and UDP)

**Code and Output:**

Code in graphTraffic.tr:

#Create a new Simulator instance

set ns [new Simulator]

#Open the NAM tracefile

set nf [open out.nam w]

#Open the tracefile

$ns namtrace-all $nf

set tf [open tracefile.tr w]

$ns trace-all $tf

# We have two different flows i.e TCP & UDP so to record we store data in two different files.

#These are simple trace files hence the extensions .tr

set f0 [open out0.tr w]

set f1 [open out1.tr w]

#Define the Finish Procedure

proc finish {} {

global ns nf tf f0 f1

$ns flush-trace

#Close the NAM trace file

close $nf

#Close the trace file

close $tf

#close the Xgraph file

close $f0

close $f1

#Execute nam files

exec nam out.nam &

#Execute Xgraph files

exec xgraph out0.tr out1.tr -geometry 800x400 &

exit 0

}

#Create 6 nodes

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

#Define Links

$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

$ns duplex-link $n2 $n4 2.7Mb 20ms DropTail

$ns duplex-link $n4 $n3 1.7Mb 20ms DropTail

$ns duplex-link $n3 $n5 1.7Mb 20ms DropTail

# Define Queue Size

$ns queue-limit $n2 $n3 10

#Define Queue Positioning

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

#Define Link Orientation

$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

$ns duplex-link-op $n2 $n4 orient right-up

$ns duplex-link-op $n4 $n3 orient right-down

$ns duplex-link-op $n3 $n5 orient right-down

#Define Link Labels

$ns duplex-link-op $n0 $n2 label "TCP"

$ns duplex-link-op $n1 $n2 label "UDP"

#Define Link Colours

$ns color 1 Blue

$ns color 2 Red

#Setup TCP Connection

set tcp [new Agent/TCP]

set sink [new Agent/TCPSink]

$ns attach-agent $n0 $tcp

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

# setting TCP flow id

$tcp set fid\_ 1

#Setup FTP application over TCP Connection

set ftp [new Application/FTP]

$ftp attach-agent $tcp

#Setup UDP Connection

set udp [new Agent/UDP]

$ns attach-agent $n1 $udp

#We normally call Destination agent as Null. But Null Doesn’t

set null [new Agent/LossMonitor]

#record traffic it discards the packets as they come.

$ns attach-agent $n5 $null

$ns connect $udp $null

# setting UDP flow id

$udp set fid\_ 2

#Setup CBR application over UDP Connection

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

# setting packet size

$cbr set packet\_size\_ 1000

#setting bit rate

$cbr set rate\_ 1mb

#setting random false means no noise

$cbr set random\_ false

#Record procedure which records traffic between endpoints

proc record {} {

global sink null f0 f1

set ns [Simulator instance]

#set the time after which the procedure should be called again

set time 0.5

#get current time

set now [$ns now]

#Find how many bytes have been received

set bw0 [$sink set bytes\_]

set bw1 [$null set bytes\_]

#Calculate the bandwidth (Mbit/s) and write to file

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

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

$sink set bytes\_ 0

$null set bytes\_ 0

# call procedure after fixed interval

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

}

$ns rtproto LS

$ns rtmodel-at 20.0 down $n2 $n4

$ns rtmodel-at 25.0 down $n3 $n5

$ns at 1.0 "$cbr start"

$ns at 2.0 "$ftp start"

$ns at 0.0 "record"

$ns at 30.0 "finish"

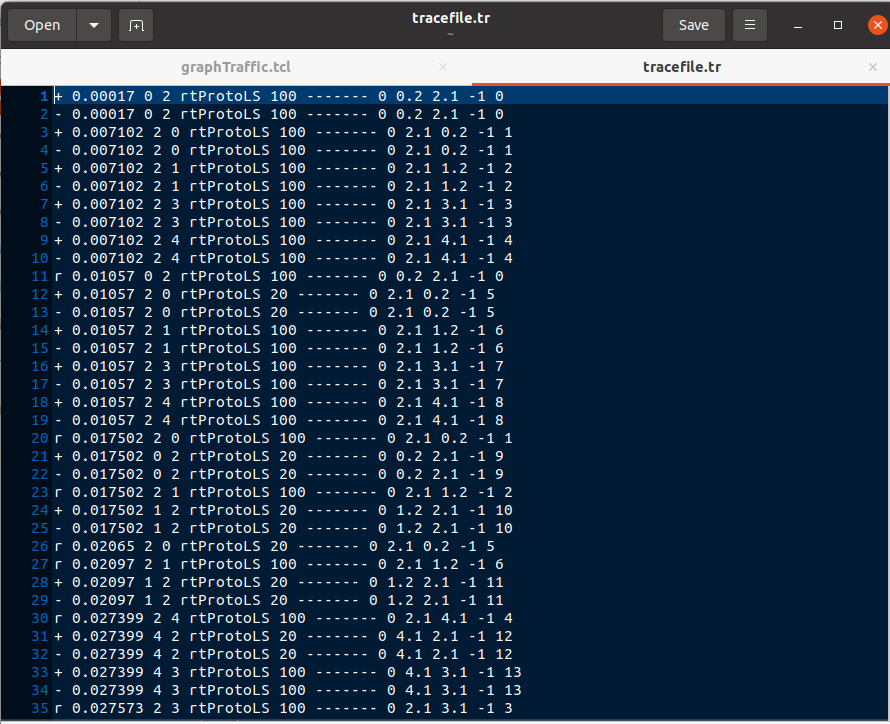
puts "CBR packet size= [$cbr set packet\_size\_]"

puts "CBR interval= [$cbr set interval\_]"

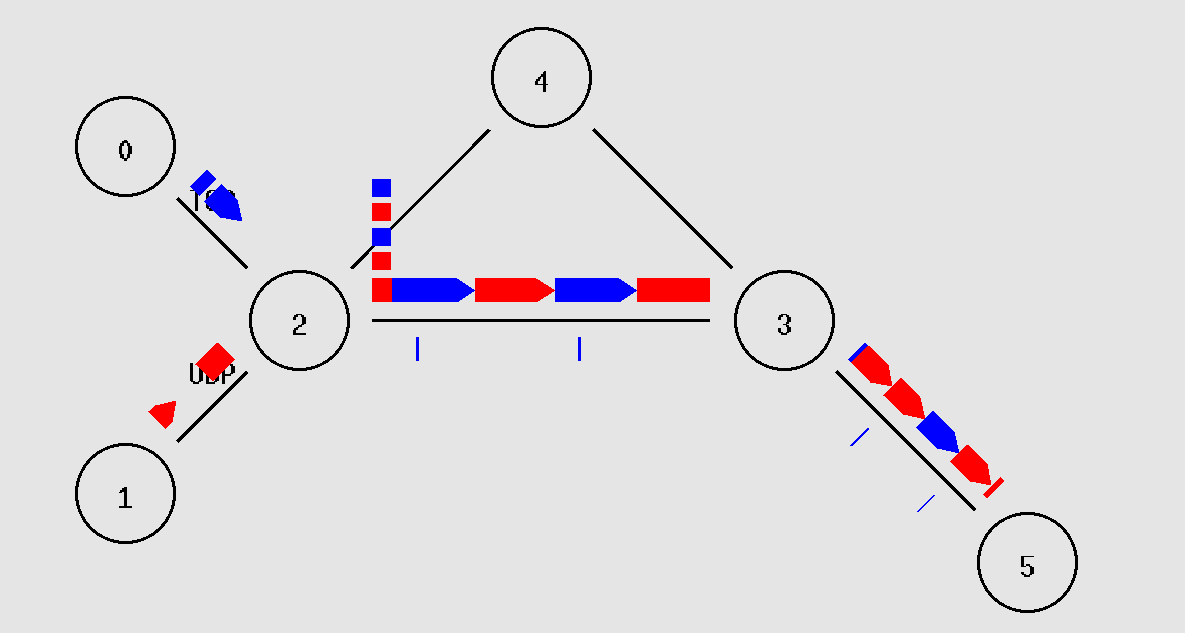
#running the simulation

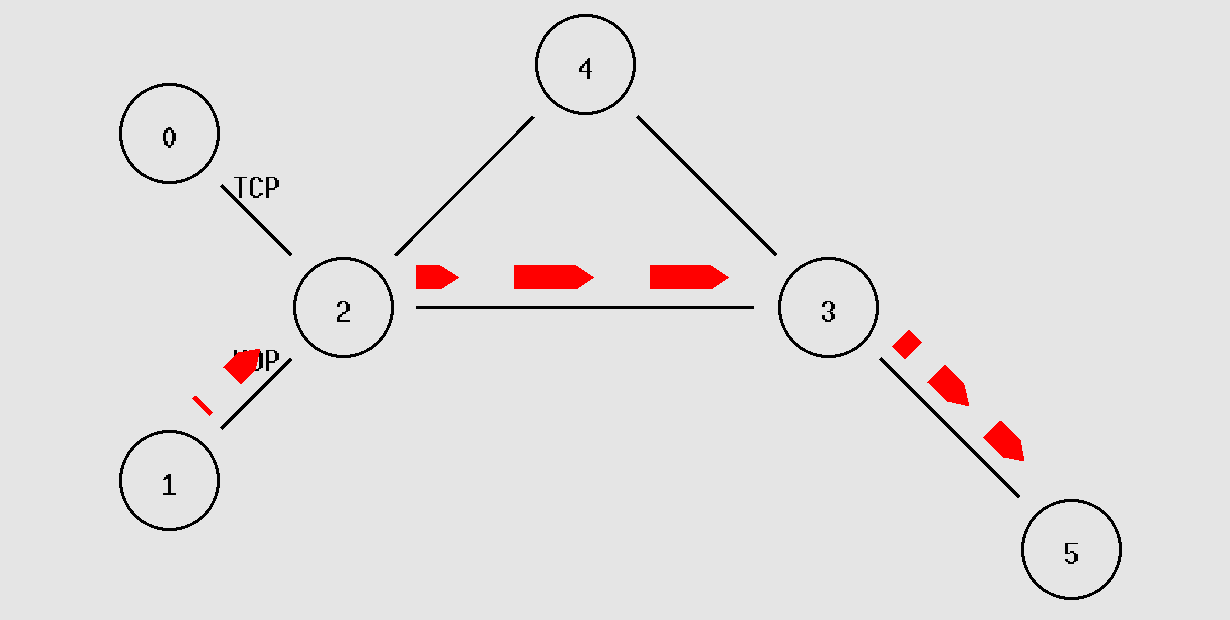
$ns run

**Screenshot of the Trace file**

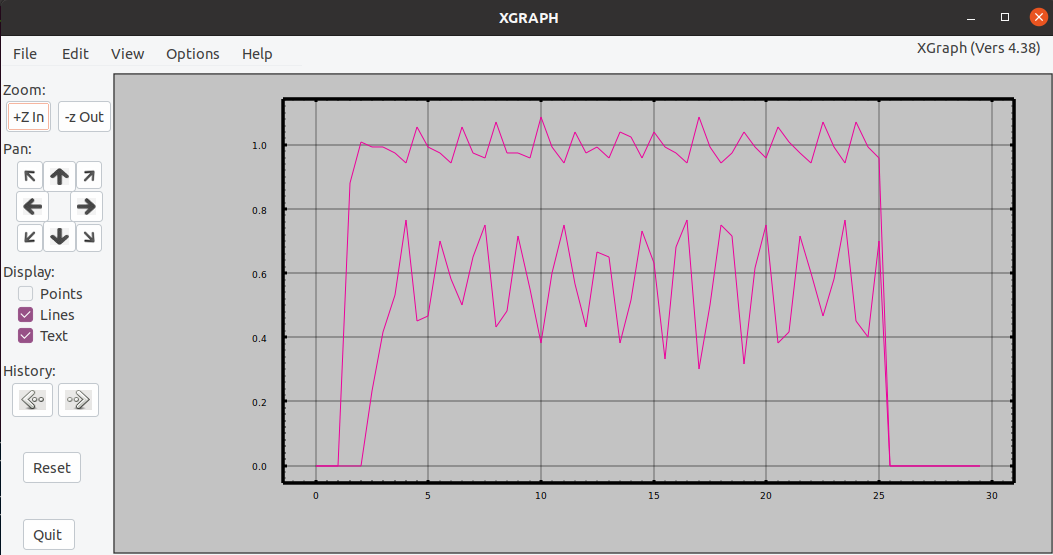


**Screenshot of the NAM simulation**





**Screenshot of XGraph plot:**

****