**PERFORMANCE ANALYSIS USING XGRAPH**

**AIM**

To analyze performance of TCP and UDP using Xgraph

**ALGORITHM:**

1.START

2.Create a new simulator ns and open out.nam as nf.

3.Create five nodes and establish Link with appropriate Nodes.

4.Declare a Procedure attach-expoo-traffic and define it to simulate a UDP traffic.

5.Create sink agents and attach it with nodes.

6.Declare and Define Finish Procedure to be called at the End.

7.Decalre a Procedure record to Track the Bytes that are received by the traffic Links.

8. Calculate the bandwidth and write it to the file.

9. Reset the Bytes value on the Traffic sinks.

10.Reschedule the Procedures by calling each defined functions in the appropriate order.

11.Run the Simulator.

12.STOP

# SOURCE CODE

set ns [new Simulator]

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

#Get an instance of the simulator set ns [Simulator instance] #Create a UDP agent and attach it to the node set source [new Agent/UDP] $ns attach-agent $node $source

#Create an Expoo traffic agent and set its configuration parameters set traffic [new Application/Traffic/Exponential]

$traffic set packet-size $size

$traffic set burst-time $burst

$traffic set idle-time $idle

$traffic set rate $rate

# Attach traffic source to the traffic generator

$traffic attach-agent $source

#Connect the source and the sink $ns connect $source $sink return $traffic

}

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] set f0 [open out0.tr w] set f1 [open out1.tr w] set f2 [open out2.tr w]

proc finish {} { global f0 f1 f2 #Close the output files close $f0 close $f1 close $f2

#Call xgraph to display the results exec xgraph out0.tr out1.tr out2.tr -geometry 800x400 & exit 0 }

proc record {} {

global sink0 sink1 sink2 f0 f1 f2 #Get an instance of the simulator set ns [Simulator instance]

#Set the time after which the procedure should be called again set time 0.5

#How many bytes have been received by the traffic sinks?

set bw0 [$sink0 set bytes\_] set bw1 [$sink1 set bytes\_] set bw2 [$sink2 set bytes\_] #Get the current time set now [$ns now]

#Calculate the bandwidth (in MBit/s) and write it to the files puts $f0 "$now [expr $bw0/$time\*8/1000000]" puts $f1 "$now [expr $bw1/$time\*8/1000000]" puts $f2 "$now [expr $bw2/$time\*8/1000000]"

#Reset the bytes\_ values on the traffic sinks

$sink0 set bytes\_ 0

$sink1 set bytes\_ 0

$sink2 set bytes\_ 0

#Re-schedule the procedure

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

}

$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"

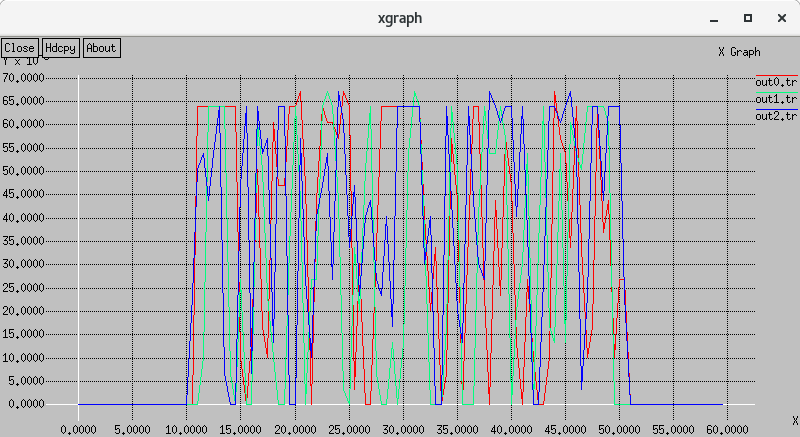
$ns at 50.0 "$source1 stop"

$ns at 50.0 "$source2 stop"

$ns at 60.0 "finish"

$ns run

# OUTPUT



# RESULT

Hence performance analysis is executed using xgraph in NS-2