UCS1511 - COMPUTER NETWORKS

Comparison of Congestion Control Algorithms

REG NO : 205001085 **EX.NO : 14**
NAME : SABARIVASAN
DATE : 23.11.22

OBJECTIVE :

To compare the congestion control algorithms.

CODE :

```
#Create a simulator object
set ns [new Simulator]
$ns rtproto LS

# Opening NAM and trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
set tracefile [open out.tr w]
$ns trace-all $tracefile

# Finish procedure
proc finish {} {
        global ns namfile tracefile
        $ns flush-trace
        #Close the NAM trace file
```

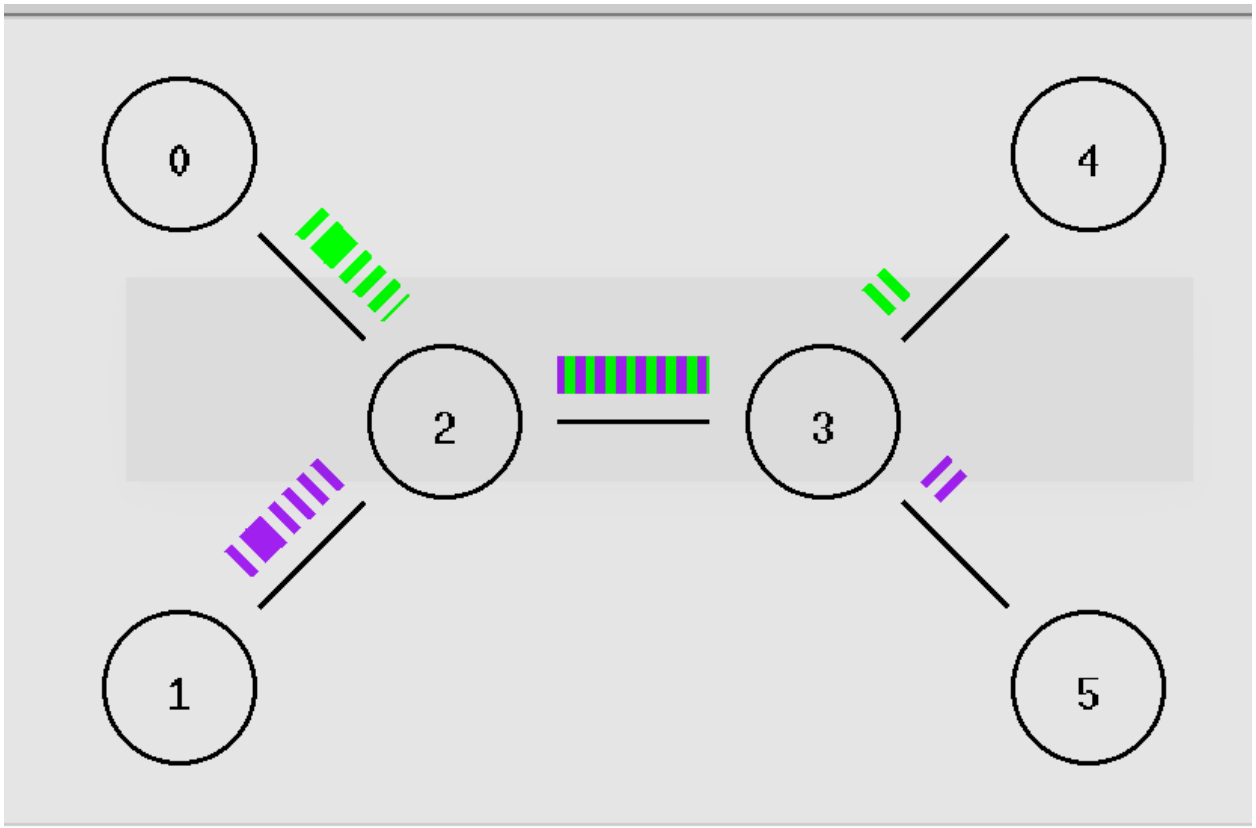
```
        close $namfile
        close $tracefile
        #Execute NAM on the trace file
        exec nam out.nam &
        exit 0
}

#Define different colors for data flows (for NAM)
$ns color 0 Red
$ns color 1 Green

# Create nodes
set node0 [$ns node]
set node1 [$ns node]
set node2 [$ns node]
set node3 [$ns node]
set node4 [$ns node]
set node5 [$ns node]

# Create links between nodes
$ns duplex-link $node0 $node2 10Mb 10ms DropTail
$ns duplex-link $node1 $node2 10Mb 10ms DropTail
$ns duplex-link $node2 $node3 10Mb 10ms DropTail
$ns duplex-link $node3 $node4 10Mb 10ms DropTail
$ns duplex-link $node3 $node5 10Mb 10ms DropTail

# The queue size at $R is to be 7, including the packet being sent
$ns queue-limit $node2 $node3 7

# Orient Links
$ns duplex-link-op $node0 $node2 orient right-down
$ns duplex-link-op $node1 $node2 orient right-up
```

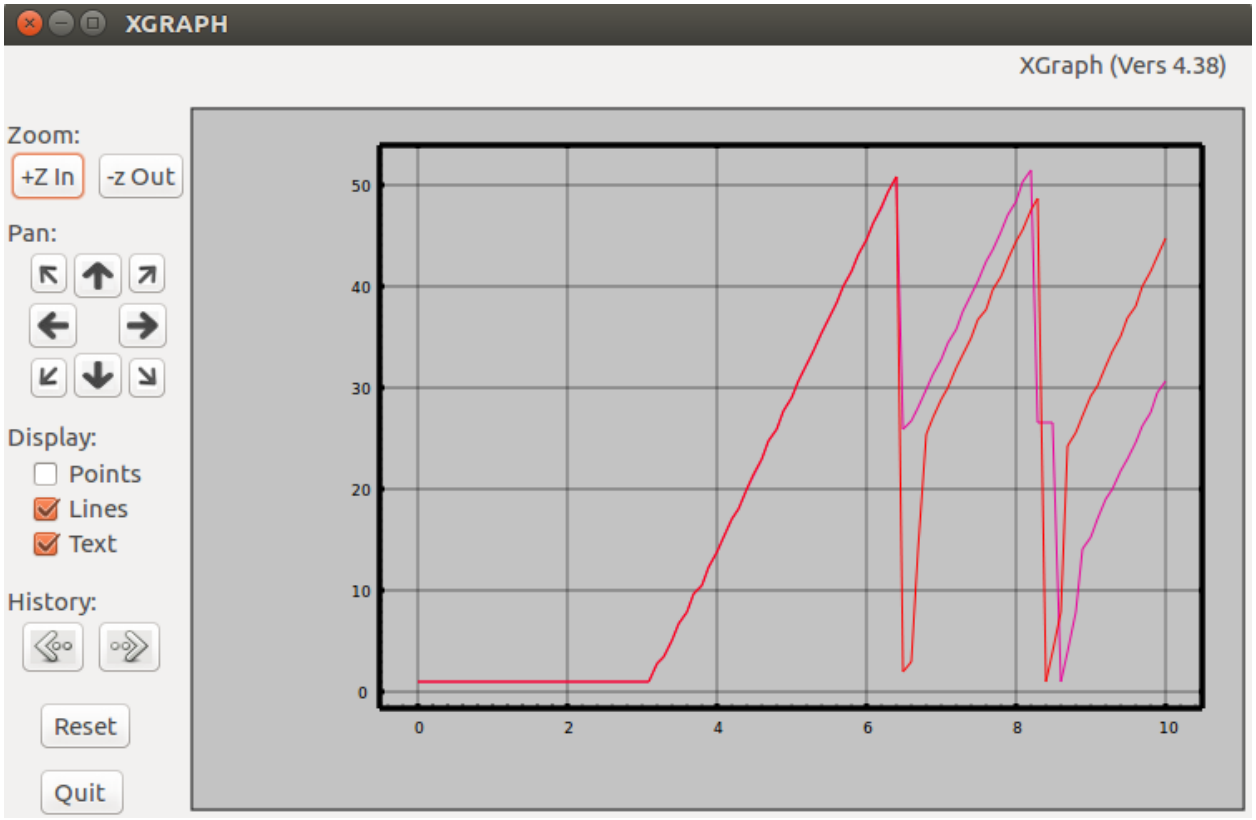
```
$ns duplex-link-op $node2 $node3 orient right
$ns duplex-link-op $node3 $node4 orient right-up
$ns duplex-link-op $node3 $node5 orient right-down
$ns duplex-link-op $node2 $node3 queuePos 0.5

# Creating a TCP sender Tahoe and attach to node 0 (default)
set tcptahoe [new Agent/TCP]
# Creating a TCP sender Reno and attach to node 1
set tcpreno [new Agent/TCP/Reno]
# Setting flow
$tcptahoe set class_ 0
$tcptahoe set window_ 100
$tcptahoe set packetSize_ 800
$tcpreno set class_ 1
$tcpreno set window_ 100
$tcpreno set packetSize_ 800
$ns attach-agent $node0 $tcptahoe
$ns attach-agent $node1 $tcpreno

# Trace variables
$tcptahoe attach $tracefile
$tcptahoe tracevar cwnd_
$tcptahoe tracevar ssthresh_
$tcptahoe tracevar ack_
$tcptahoe tracevar maxseq_

#Create a TCP receive agent (a traffic sink) and attach it to B
set endtahoe [new Agent/TCPSink]
$ns attach-agent $node4 $endtahoe

set endreno [new Agent/TCPSink]
$ns attach-agent $node5 $endreno
```

```
#Connect the traffic source with the traffic sink
$ns connect $tcptahoe $endtahoe
$ns connect $tcpreno $endreno

#Schedule the connection data flow; start sending data at T=0, stop at
T=10.0
set ftptahoe [new Application/FTP]
$ftptahoe attach-agent $tcptahoe
$ns at 0.0 "$ftptahoe start"
$ns at 10.0 "finish"

set ftpreno [new Application/FTP]
$ftpreno attach-agent $tcpreno
$ns at 0.0 "$ftpreno start"
$ns at 10.0 "finish1"

#Plot Congestion Window Graph
proc plotWindow {tcpSource outfile} {
   global ns
   set now [$ns now]
   set cwnd [$tcpSource set cwnd_]

# the data is recorded in a file called congestion.xg (this can be plotted
# using xgraph or gnuplot. this example uses xgraph to plot the cwnd_
   puts  $outfile  "$now $cwnd"
   $ns at [expr $now+0.1] "plotWindow $tcpSource  $outfile"
}

set outfile [open  "congestiontahoe.xg"  w]
$ns color 0 Green
$ns  at  0.0  "plotWindow $tcptahoe  $outfile"
```

```
proc finish_cong {} {
 exec xgraph congestion.xg -geometry 300x300 &
   exit 0
}


set outfile [open  "congestionreno.xg"  w]
$ns color 1 Purple
$ns  at  0.0  "plotWindow $tcpreno  $outfile"
proc finish_cong {} {
 exec xgraph congestion.xg -geometry 300x300 &
   exit 0
}

#Run the simulation
$ns run
```

OUTPUT :






```
root@OSNPL-C6: ~/Desktop/NetworksLAB/14/Final Correct
root@OSNPL-C6:~/Desktop/NetworksLAB/14/Final Correct# python3 throughput.py
Throughput =  7.4440704 mbps
root@OSNPL-C6:~/Desktop/NetworksLAB/14/Final Correct#
```