EX8 – PERFORMANCE EVALUATION OF TCP AND UDP

- S. Vishakan CSE – C 18 5001 196

NS2 Script File:

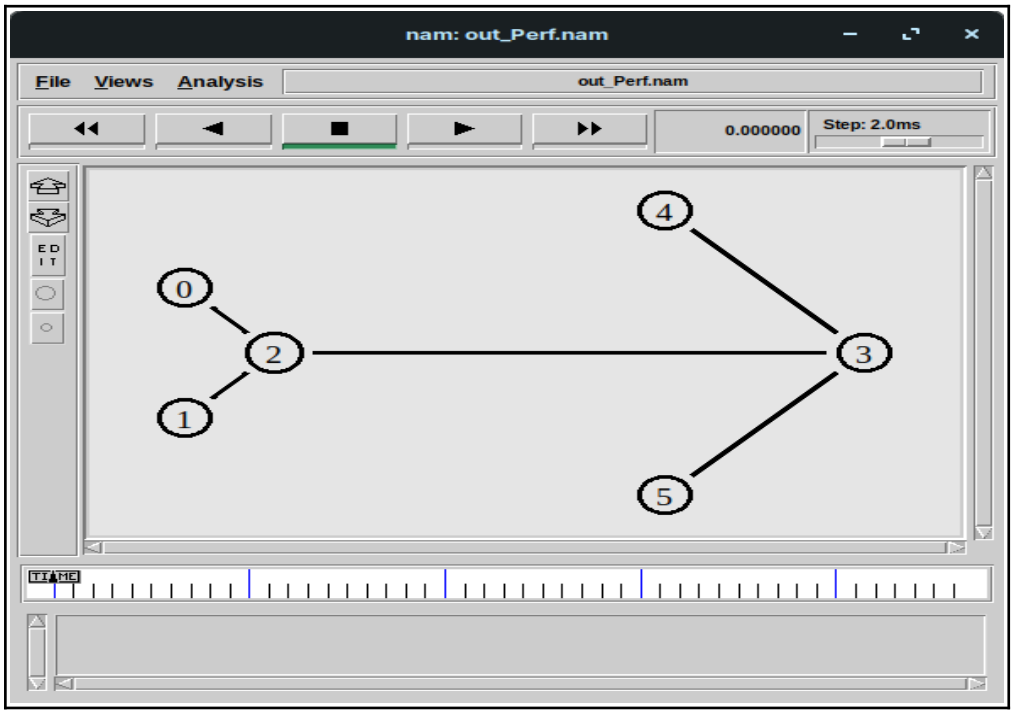
```
#Performance Evaluation of TCP and UDP sharing a bottleneck

#Create a new simulator object
set ns [new Simulator]

#Define different colors for data flows (for NAM)
$ns color 1 Blue
$ns color 2 Red

#Open the NAM trace file
set nf [open out_Perf.nam w]
$ns namtrace-all $nf

#Define a 'finish' procedure
proc finish {} {
global ns nf
$ns flush-trace

#Close the NAM trace file
close $nf
#Execute NAM on the trace file
exec nam out_Perf.nam &
exit 0
}

#Create the six 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]

#Create links between the nodes
$ns duplex-link $n0 $n2 2Mbps 10ms DropTail
$ns duplex-link $n1 $n2 2Mbps 10ms DropTail
$ns simplex-link $n2 $n3 0.3Mbps 100ms DropTail
$ns simplex-link $n3 $n2 0.3Mbps 100ms DropTail
$ns duplex-link $n3 $n4 0.5Mbps 40ms DropTail
$ns duplex-link $n3 $n5 0.5Mbps 40ms DropTail
```

```
#Give node position (for NAM)
$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns simplex-link-op $n2 $n3 orient right
$ns simplex-link-op $n3 $n2 orient right
$ns duplex-link-op $n3 $n4 orient left-up
$ns duplex-link-op $n3 $n5 orient left-down

#Set Queue size of link n2-n3 to 10
$ns queue-limit $n2 $n3 10

#Monitor the queue for the link n2-n3
$ns simplex-link-op $n2 $n3 queuePos 0.5
$ns simplex-link-op $n3 $n2 queuePos 1.5

#Set up a TCP connection over n0 & n4 and its flow id, window size and packet size
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp

set tcp_sink [new Agent/TCPSink/DelAck]
$ns attach-agent $n4 $tcp_sink

$ns connect $tcp $tcp_sink

$tcp set fid_ 1
$tcp set window_ 8000
$tcp set packetSize_ 512

#Set up a FTP over TCP connection
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$tcp set type_ TCP

#Set up a UDP connection over n1 & n5 and its flow id
set udp [new Agent/UDP]
$ns attach-agent $n1 $udp

set null [new Agent/Null]
$ns attach-agent $n5 $null

$ns connect $udp $null

$udp set fid_ 2
```

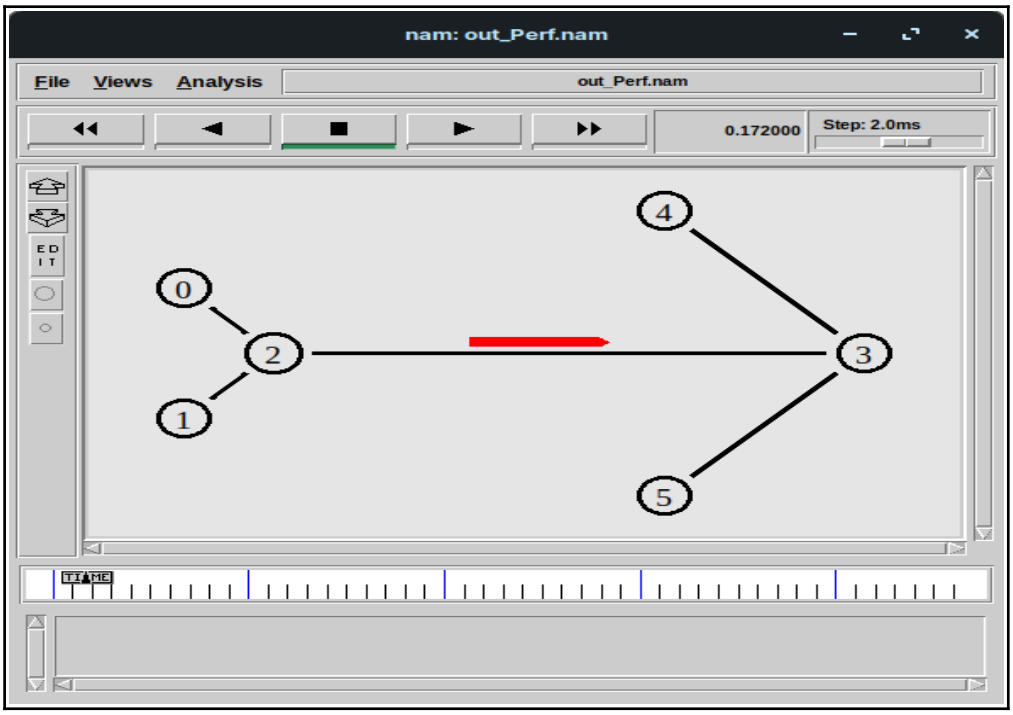
```
#Set up a CDR over UDP connection with type, packet size, rate, random fields
set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp
$cbr set type_ CBR
$cbr set packet_size_ 1024
$cbr set rate_ 0.01Mb
$cbr set random_ false

#Schedule events for the CBR and FTP agents
$ns at 0.1 "$cbr start"
$ns at 1.0 "$ftp start"
$ns at 4.5 "$ftp stop"
$ns at 5.0 "$cbr stop"

#Detach TCP and Sink agents (not necessary)
$ns at 4.5 "$ns detach-agent $n0 $tcp; $ns detach-agent $n4 $tcp_sink"

#Call the finish procedure after 5 seconds of simulation time
$ns at 5.0 "finish"

#Print CBR packet size and interval
puts "CBR Packet Size = [$cbr set packet_size_]"
puts "CBR Interval = [$cbr set interval_]"

#Run the simulation
puts "Running the simulation..."
$ns run
```

Output:

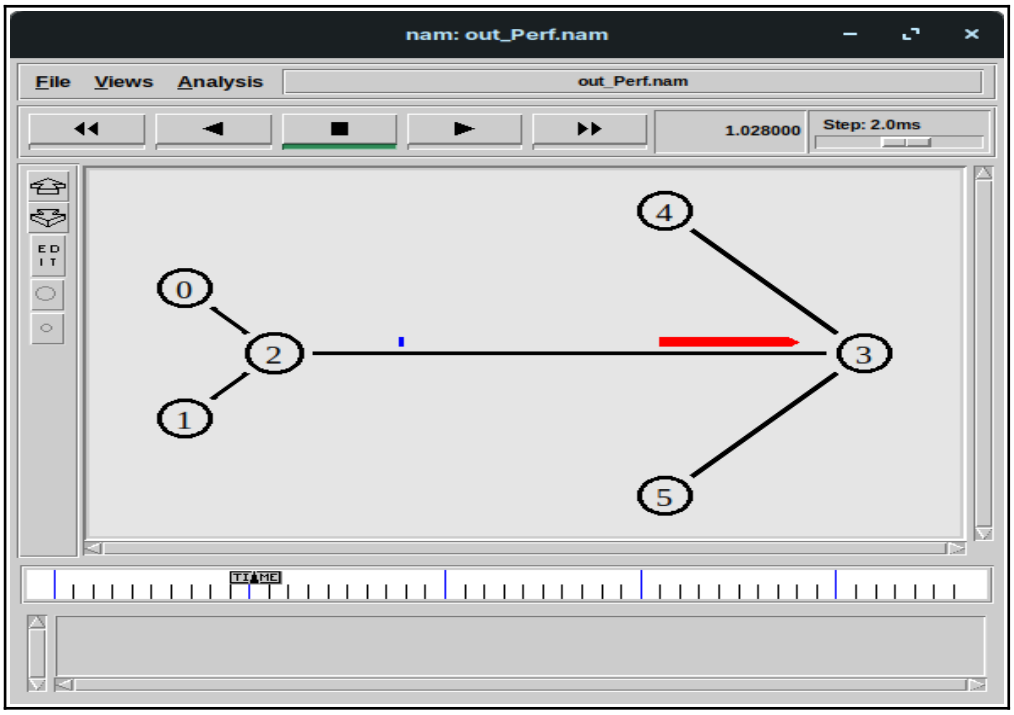
Node Layout:



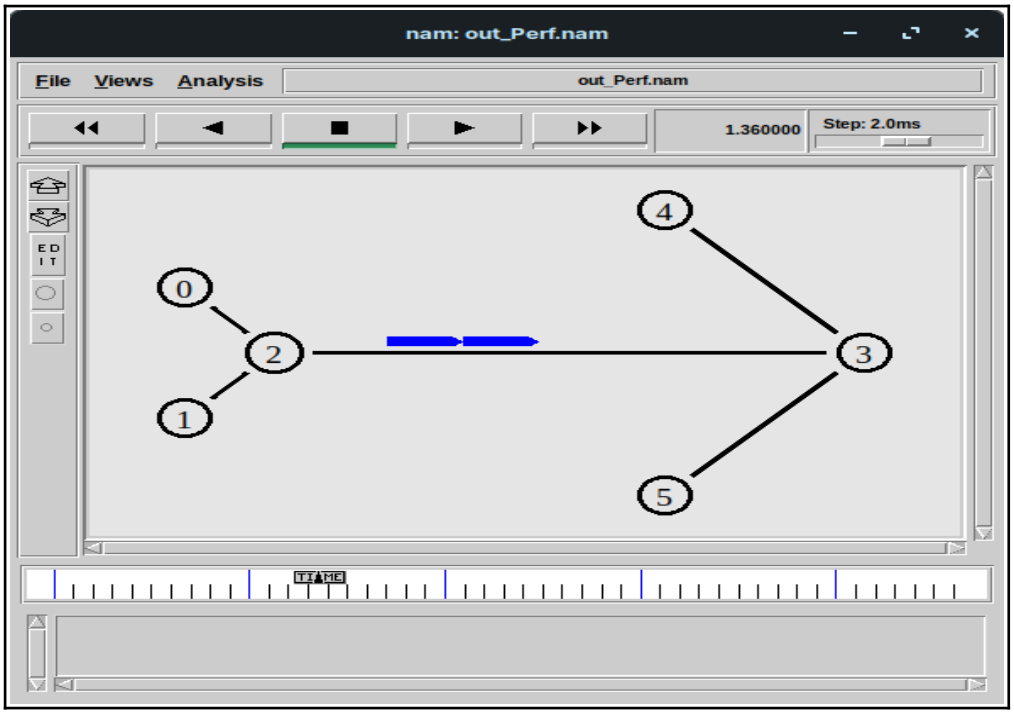
N1 to N5 CBR Traffic via UDP:



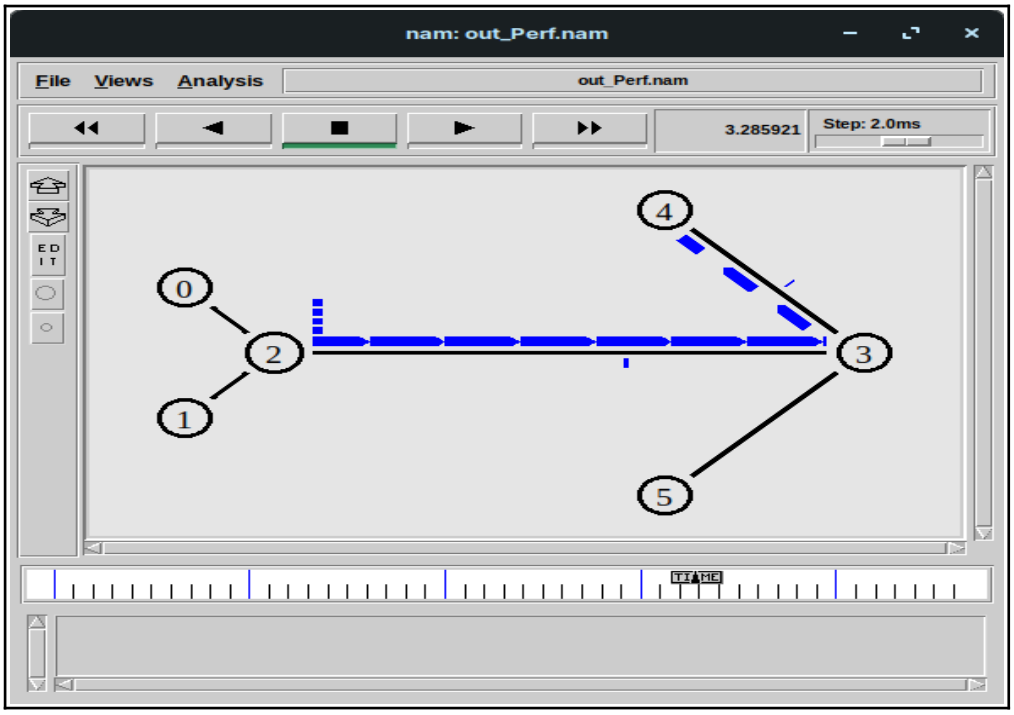
N0 to N4 FTP via TCP Acknowledgement:



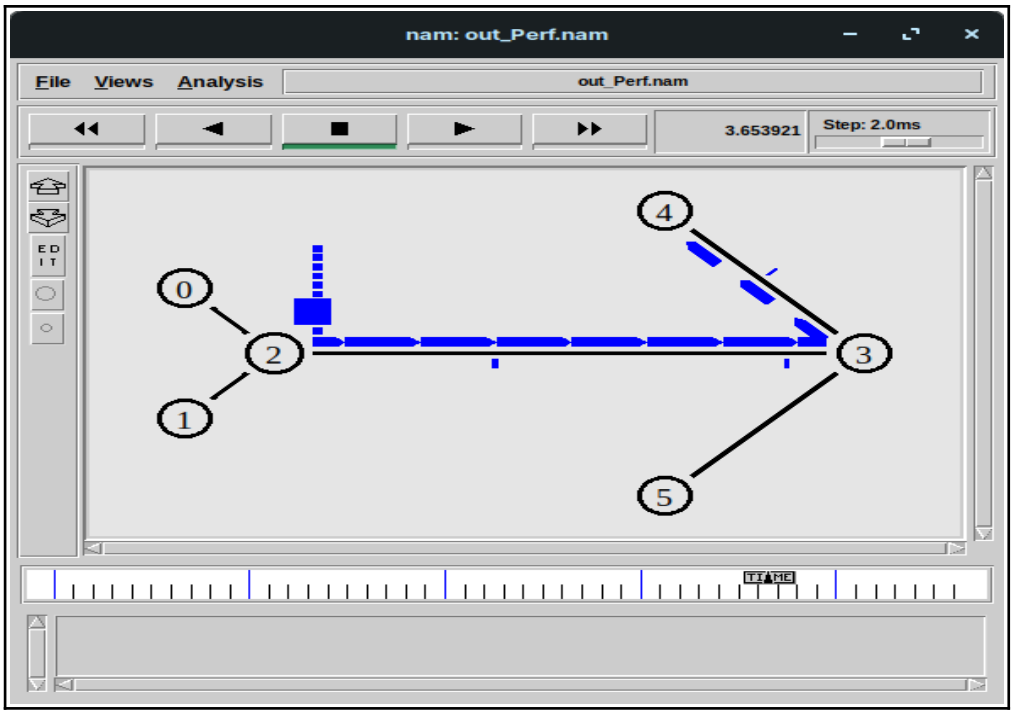
N0 to N4 Transmission of Packets:



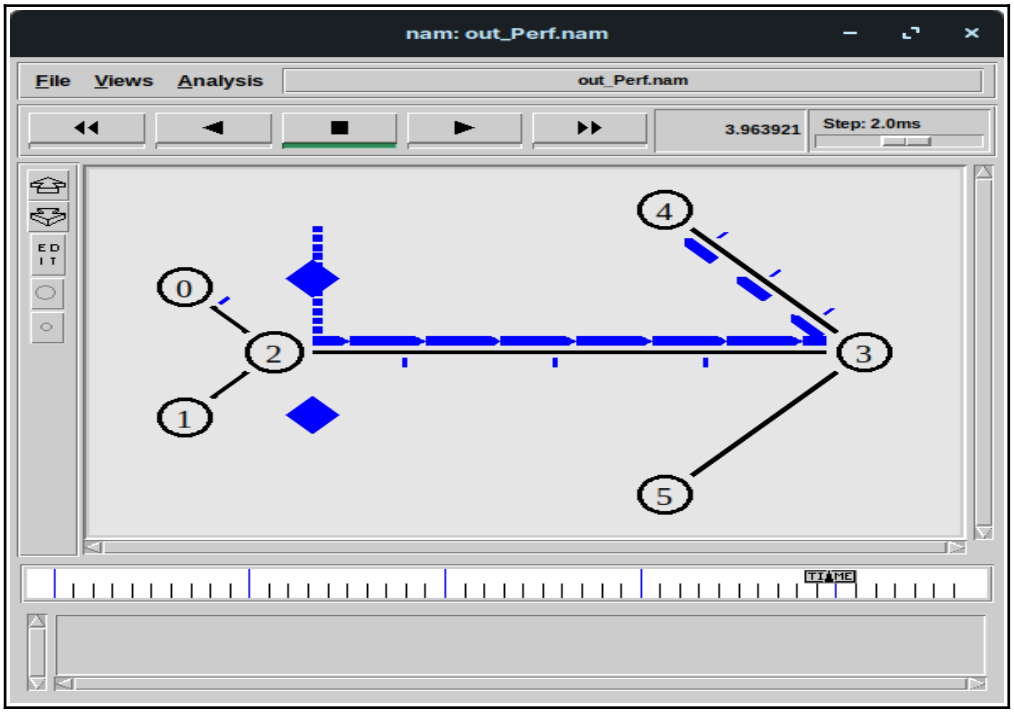
Node N2-N3’s Queue in Action:



First Packet Drop due to Bottlenecked Link (N2 to N3):



More Packet Drops:



Inference:

A basic TCL script was written to evaluate the performance measure of TCP and UDP Protocol that shared a common bottlenecked link.

An FTP Application used the TCP link to transfer information from N0 to N4 whereas a CBR Traffic used the UDP link to transfer information from node N1 to N5.

Link between N2 and N3 is a common link which was bottlenecked at 0.3Mbps bandwidth and 100 ms delay.

The initial transfer of packets were lossless.

At around 3.5 seconds of simulation, the link between N2 and N3 were getting overworked due to multiple packets from N0 (TCP) and N2's queue started to reach full capacity.

At 3.6 seconds, the node N2 dropped the first packet from node N0 using the DropTail mechanism due to its queue reaching full capacity.

More packets were dropped during the interval from 3.8 seconds to 4.2 seconds due to the bottleneck between the link N2 and N3 forcing node N2 to drop received packets.

Once node N0 stopped transmitting TCP packets at 4.5 seconds, the load eased between node N2 and N3 and no further packets were dropped.

The simulation thus ended at 5.0 seconds with node N1 stopping transmission of UDP packets.

It was clearly seen that a common bottlenecked node affects the overall efficiency of the entire network, forcing packets to be dropped when the link gets overworked.

To increase efficiency, the bottleneck must be resolved, i.e the transmission delay between node N2 and N3 should be reduced and its bandwidth should be increased.

One other possibility is that the other source nodes recognize the bottleneck at N2-N3 link and adaptively reduce its transmission rate, sending packets at higher time intervals, so that packets do not get lost during transmission.

Alternatively, the queue size of the bottlenecked node could also be increased.

This fact was observed from the CBR traffic transmitted by the node N1 via UDP packets as the CBR interval was much lower (0.01 Mbps) than the bandwidth at the bottlenecked link (0.3 Mbps)

Thus, UDP packets were not dropped in transmission. Only FTP packets were dropped as it kept sending packets more and more frequently as time passed.

In conclusion, the effect of a common bottleneck was observed in the course of the experimental simulation and some theoretical remedies were proposed to resolve the issue caused by the bottleneck.