Simulation of congestion control algorithms in NS2

AIM

To simulate and analyze the performance of different TCP congestion control algorithms, TCP Tahoe and TCP Reno, using NS2, and observe their impact on network performance through Network Animator (NAM) and trace files.

Questions

1. Write tcl script to simulate:
    i. Create 3 nodes and the links between the nodes as
        a. 0 → 1 10Mb 10 ms duplex link
        b. 1 → 2 2Mb 10 ms duplex link
    ii. Align the nodes properly
    iii. Setup a TCP/Tahoe connection over 0 and 2 and its flow id, window size,packet
    iv. Show the simulation in network animator and in trace file.
2. Write tcl script to simulate:
    i. Create 3 nodes and the links between the nodes as
        a. 0 → 1 10Mb 10 ms duplex link
        b. 1 → 2 2Mb 10 ms duplex link
    ii. Align the nodes properly
    iii. Setup a TCP/Reno connection over 0 and 2 and mention the same flow id,window size, packet used for TCP/Tahoe
    iv. Show the simulation in network animator and in trace file.

Algorithm:

1. Create the simulator object and open a trace file for Network Animator (NAM) to record the simulation events.
2. Define different colour nodes
3. Create three nodes (n0, n1, and n2) to represent the network topology.
4. Establish links by connecting n0 and n1 with a 10 Mbps duplex link with 10 ms delay and connecting n1 and n2 with a 2 Mbps duplex link with 10 ms delay.
5. Align the nodes for better visualisation
6. Create a TCP agent and set its window size and flow id
7. For TCP Tahoe simulation: Set the type to Tahoe

For TCP Reno simulation: Set the type to Reno

8. Attach the TCP agent to n0
9. Create a TCP Sink agent and attach it to n2.
10. Connect the TCP agent at n0 to the TCP Sink agent at n2 to establish the connection.
11. Connect the TCP agent at n0 to the TCP Sink agent at n2 to establish the connection.
12. Set packet size to 1000 and interval to 0.1 seconds.
13. Attach the CBR traffic source to the TCP agent.
14. Start the CBR traffic and stop it
15. Start the finish procedure after the simulation ends and launch NAM

TAHOE:

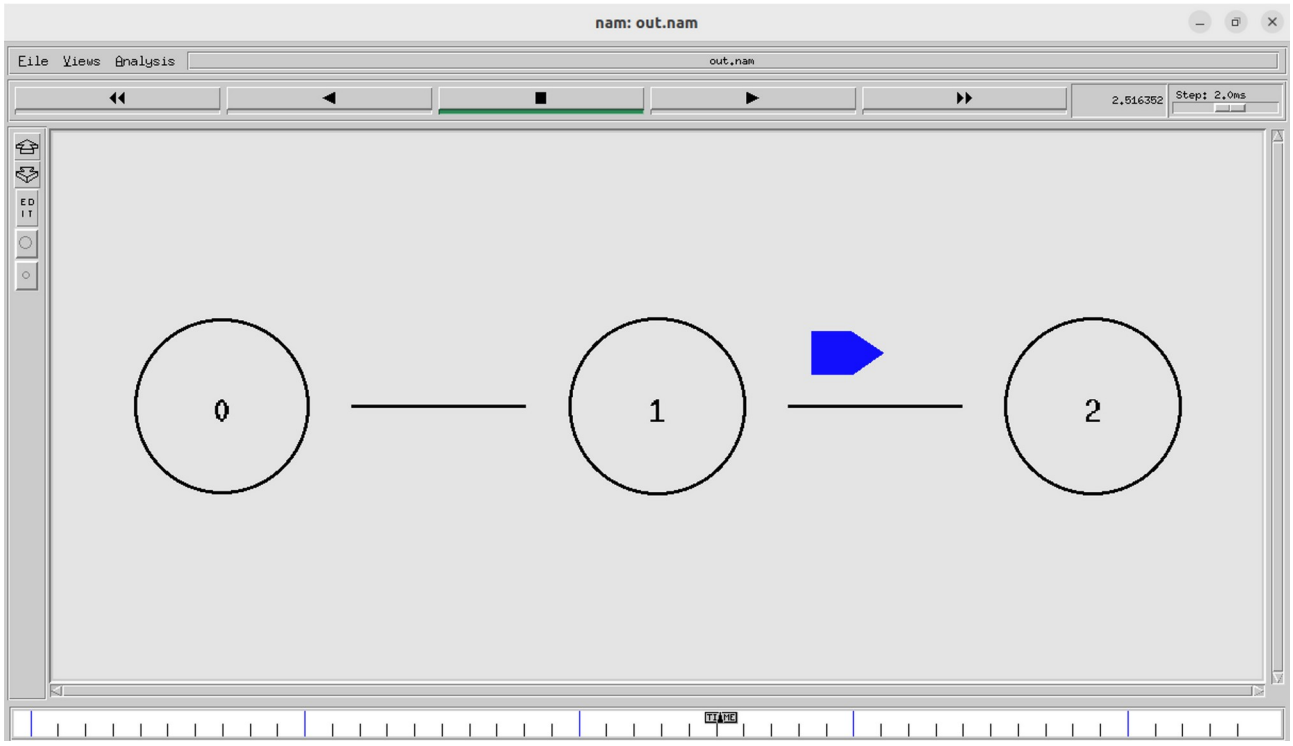
TCL Code

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
  global ns nf
  $ns flush-trace
  close $nf
  exec nam out.nam &
  exec xgraph congestion.xg -geometry 300x300 &
  exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail

$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right

set tcp [new Agent/TCP]
$tcp set class_ 1
$tcp set window_ 32
$tcp set packetSize_ 1000
$tcp set fid_ 2
```

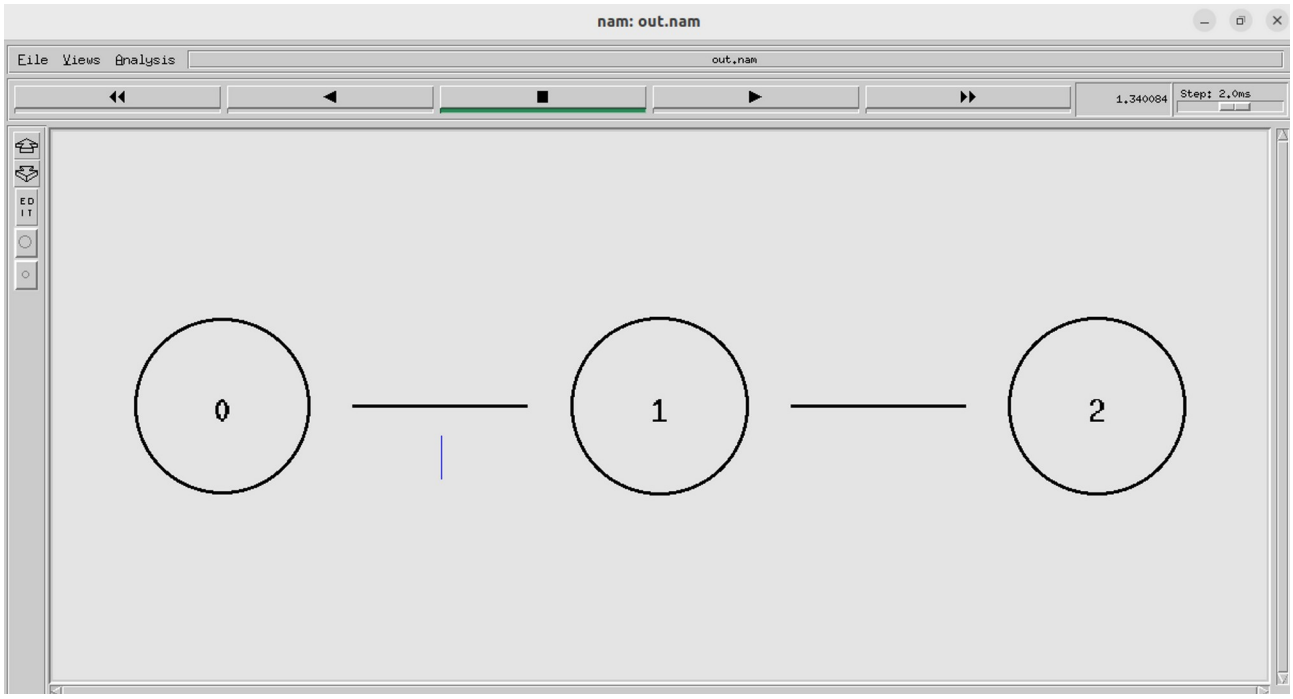
```
$tcp set type_ Tahoe
$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink

set cbr [new Application/Traffic/CBR]
$cbr set packetSize_ 1000
$cbr set interval_ 0.1000
$cbr attach-agent $tcp

$ns at 0.2 "$cbr start"
$ns at 4.5 "$cbr stop"
$ns at 5.0 finish

$ns run
```

EXECUTION SNAPSHOTS:

i. Message



ii. Acknowledgement



RENO:

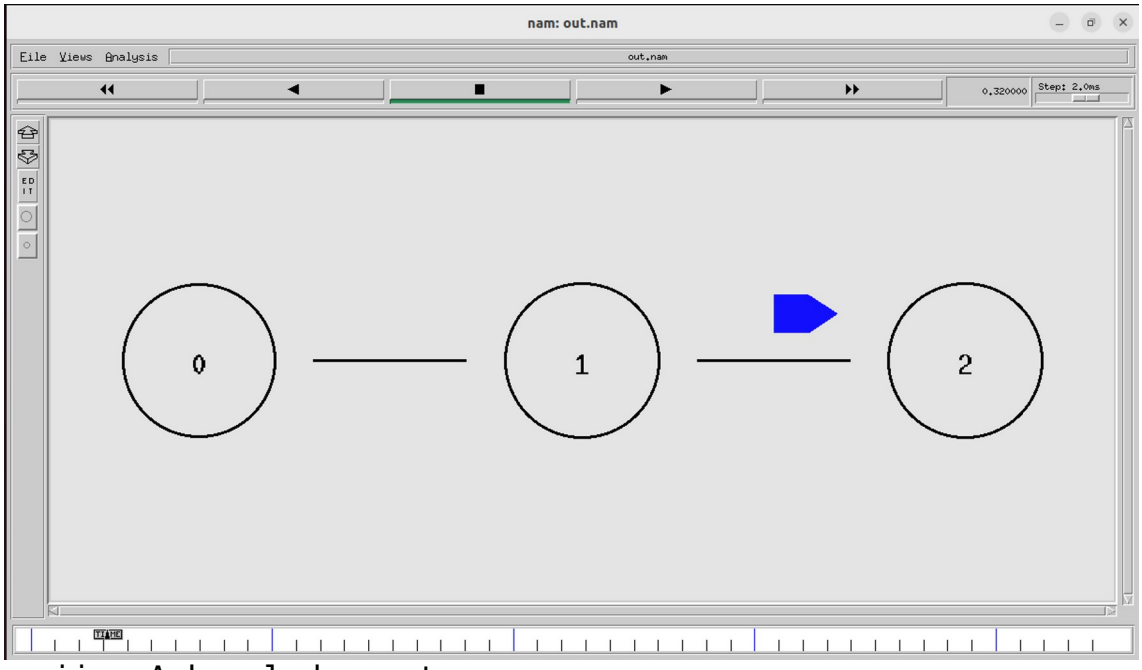
TCL Code:

```
set ns [new Simulator]

$ns color 1 Red
$ns color 2 Blue

set nf [open out.nam w]
$ns namtrace-all $nf

proc finish {} {
  global ns nf
  $ns flush-trace
  close $nf
  exec nam out.nam &
  exec xgraph congestion.xg -geometry 300x300 &
  exit 0
}

set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]

$ns duplex-link $n0 $n1 10Mb 10ms DropTail
$ns duplex-link $n1 $n2 2Mb 10ms DropTail
```

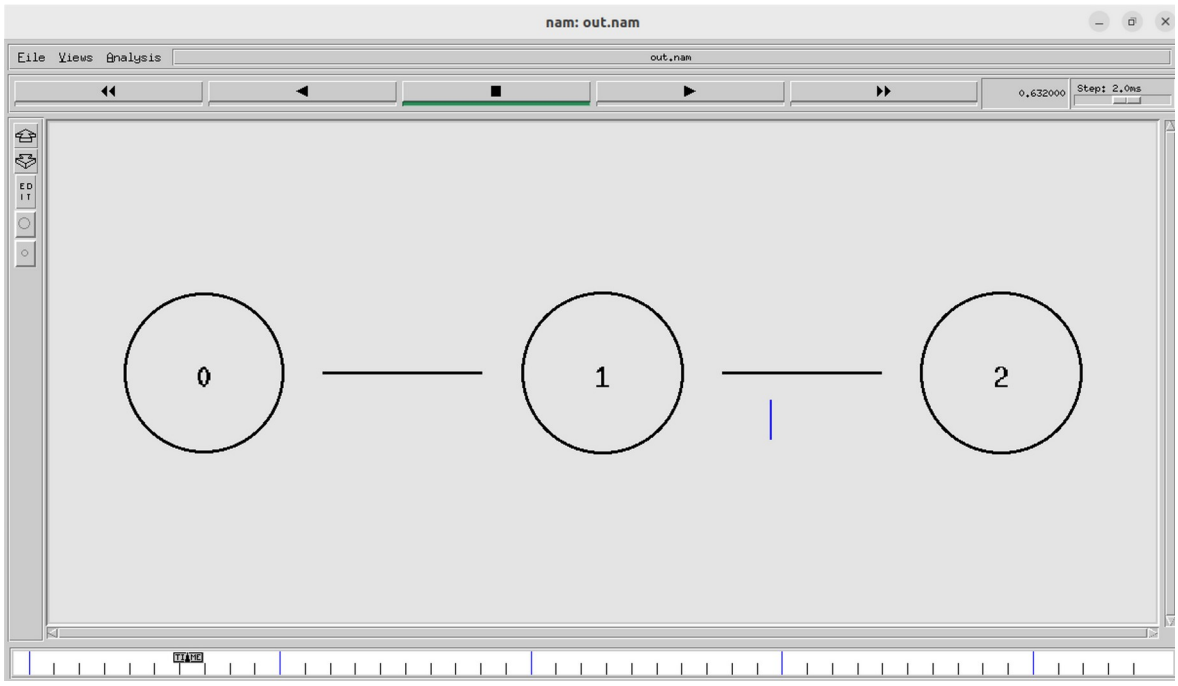
```
$ns duplex-link-op $n0 $n1 orient right
$ns duplex-link-op $n1 $n2 orient right

set tcp [new Agent/TCP]
$tcp set class_ 2
$tcp set window_ 32
$tcp set packetSize_ 1000
$tcp set fid_ 2
$tcp set type_ Reno
$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]
$ns attach-agent $n2 $sink
$ns connect $tcp $sink

set cbr [new Application/Traffic/CBR]
$cbr set packetSize_ 1000
$cbr set interval_ 0.1000
$cbr attach-agent $tcp

$ns at 0.2 "$cbr start"
$ns at 4.5 "$cbr stop"
$ns at 5.0 finish

$ns run
```

EXECUTION SNAPSHOTS:

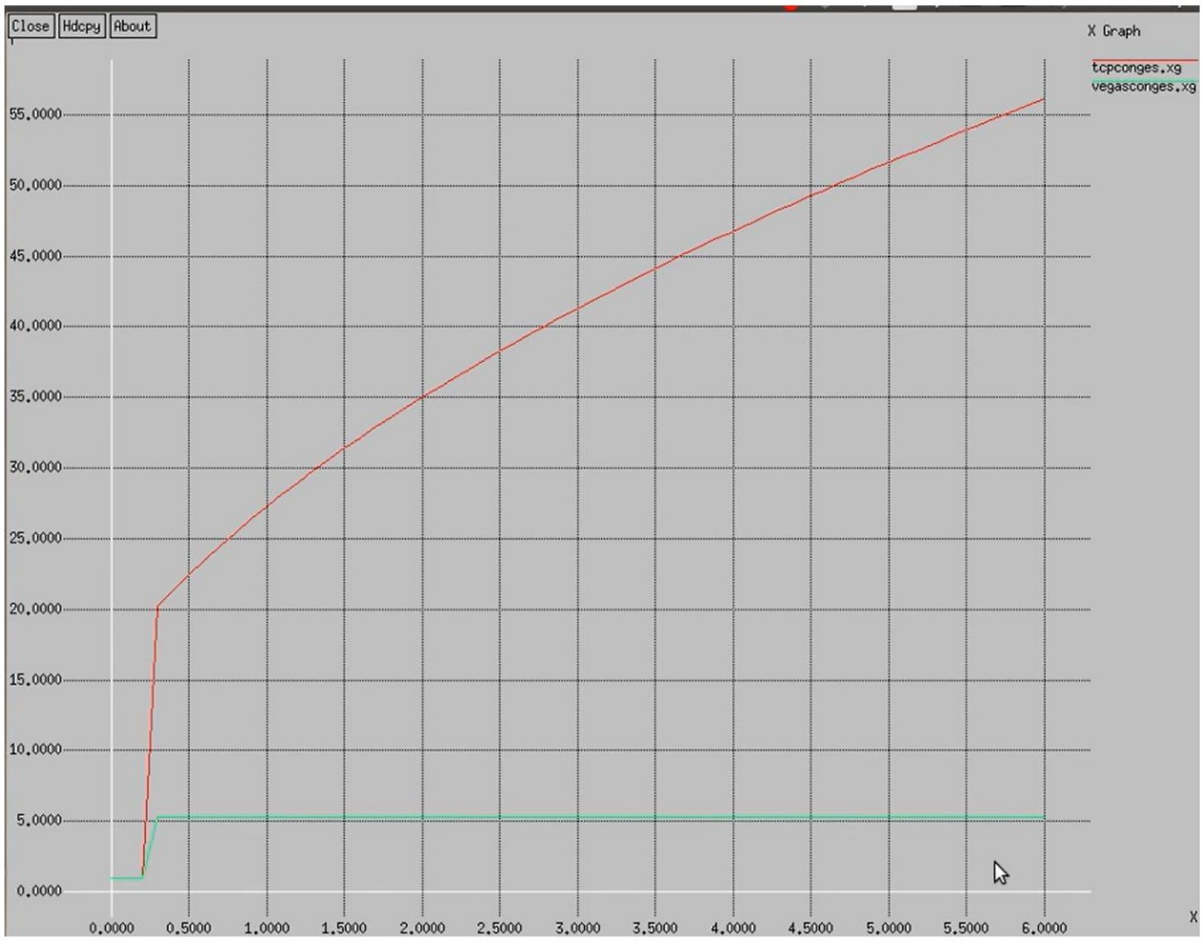
i. Message

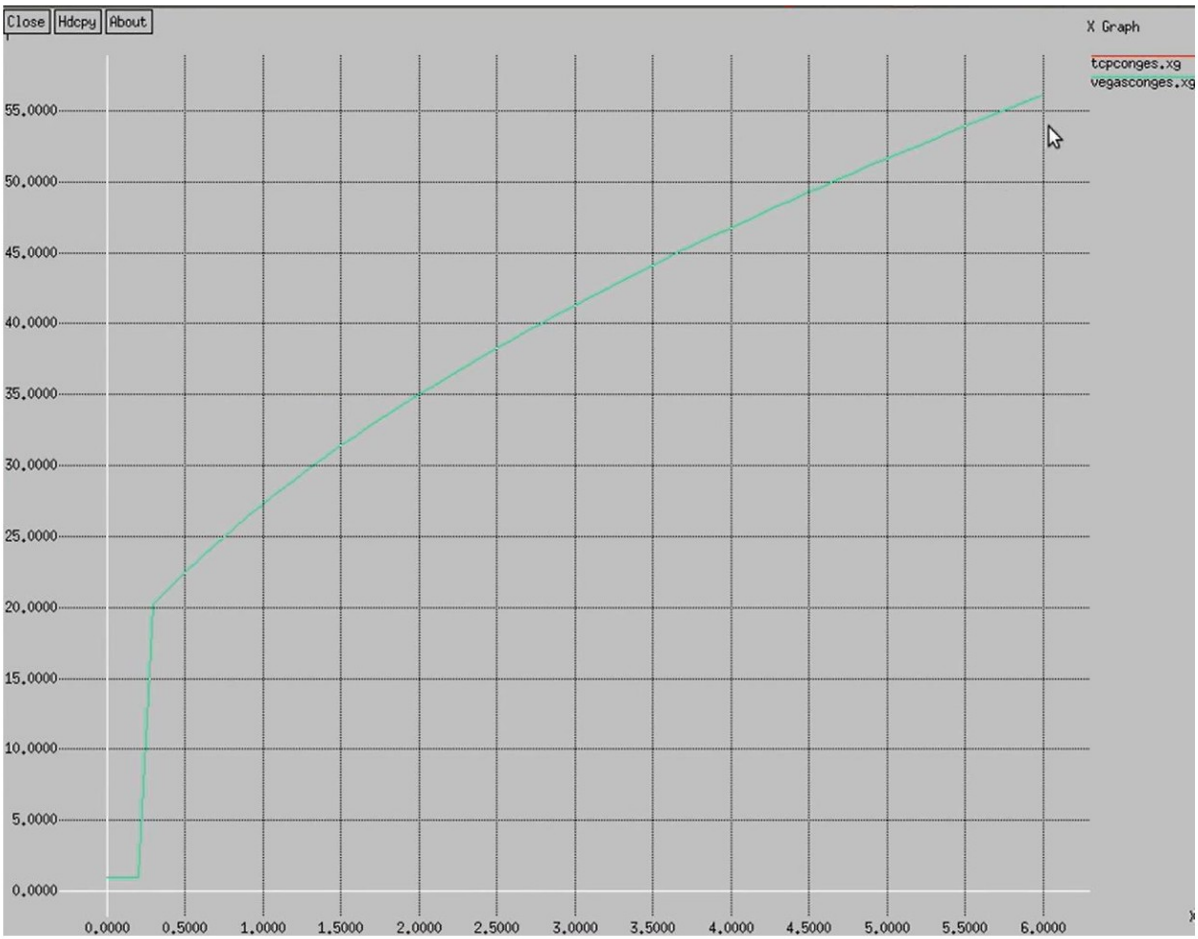


ii. Acknowledgement



Xgraph Output:





Learning Outcomes:

1. Learn the basics of TCP congestion control and differences between Tahoe and Reno algorithms.
2. Learnt to configure and manage traffic flows using CBR traffic over TCP connections.
3. Learnt to set network topologies and simulate link configurations and delays

Best Practices:

1. Set flow IDs and assign colors to distinguish traffic flows, making it easier to interpret simulation output in NAM.
2. Flush traces and close files properly in finish procedure