**NATIONAL INSTITUTE OF TECHNOLOGY KURUKSHETRA**



**COMPUTER NETWORKS LAB FILE**

**SUBMITTED BY:                                                        SUBMITTED TO:**

**NAME:VINAY KUMAR                                         MR.SAUMYA RANJAN SAHU**

**ROLL NO: 11610194 COMPUTER ENGG DEPT**

**SECTION: CO-4**

**Experiment 2**

1. **Create the following scenario with two nodes n0 and n1 and link in between.**

– Sender agent: Agent/UDP

– Receiver agent: Agent/Null

– Connect agents

– Data source: Application/Traffic/CBR

– Run from 0.5 to 4.5 sec, finish at 5.0 sec



**Code:**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

$ns color 1 red

# Open the trace file

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set send [$ns node]

set rec [$ns node]

$ns simplex-link $send $rec 2.0Mb 10ms DropTail

$ns queue-limit $send $rec 10

set udp [new Agent/UDP]

$ns color 1 red

set null [new Agent/Null]

set cbr [new Application/Traffic/CBR]

$ns attach-agent $send $udp

$ns attach-agent $rec $null

$ns connect $udp $null

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1mb

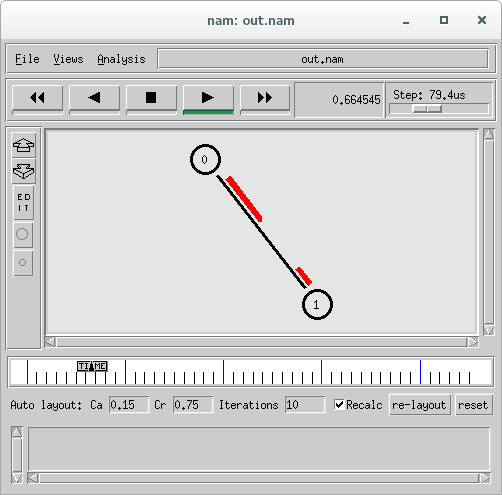
$ns at 0.1 "$cbr start"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run

**OUTPUT**

****

1. **Create the following scenario and connect the appropriate agents**

* Start the FTP application at t = 0.5s
* Start the CBR data source at t = 1s
* Terminate both at t = 4.5 s
* Visualize the bottle neck queue



**CODE:**

set ns [new Simulator]

set nf [open out2.nam w]

$ns namtrace-all $nf

$ns color 1 blue

$ns color 2 red

# Open the trace file

set nf [open tracefile2.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out2.nam &

exit 0

}

set tcpnode [$ns node]

set udpnode [$ns node]

set commonnode [$ns node]

set finalnode [$ns node]

$ns duplex-link $tcpnode $commonnode 1Mb 10ms DropTail

$ns simplex-link $udpnode $commonnode 2Mb 10ms DropTail

$ns duplex-link $commonnode $finalnode 2Mb 10ms DropTail

$ns queue-limit $commonnode $finalnode 80

#$ns queue-limit $udpnode $commonnode 80

$ns duplex-link-op $tcpnode $commonnode orient right-down

$ns simplex-link-op $udpnode $commonnode orient right-up

$ns duplex-link-op $commonnode $finalnode orient right

set tcp [new Agent/TCP]

set udp [new Agent/UDP]

set tcpsink [new Agent/TCPSink]

set null [new Agent/Null]

$tcp set fid\_ 1

$udp set fid\_ 2

$ns attach-agent $tcpnode $tcp

$ns attach-agent $udpnode $udp

$ns attach-agent $finalnode $null

$ns attach-agent $finalnode $tcpsink

$ns connect $tcp $tcpsink

$ns connect $udp $null

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1.5mb

$ns at 0.2 "$ftp start"

$ns at 0.5 "$cbr start"

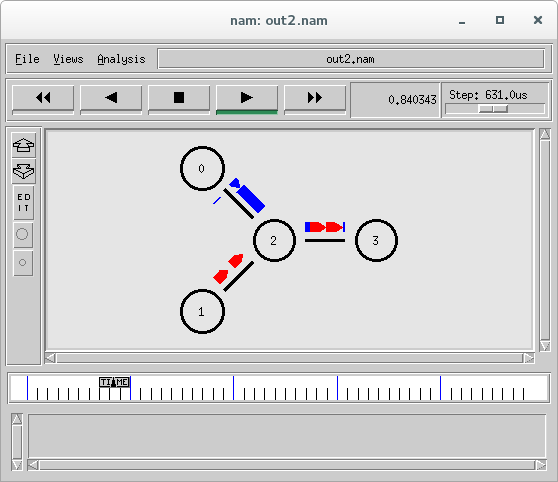
$ns at 4.5 "$ftp stop"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run

**OUTPUT:**



**3. Write a TCL script to simulate a file transfer with using ns 2:**

Consider a client and a server. The server is running a FTP application (over TCP). The client sends a request to download a file of size 10 MB from the server. Write a script to simulate this scenario. Let node #0 be the server and node #1 be the client. TCP packet size is 1500 B. Assume typical values for other parameters.

**CODE:**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

$ns color 1 green

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set send [$ns node]

set rec [$ns node]

$ns duplex-link $send $rec 2.0Mb 10ms DropTail

$ns queue-limit $send $rec 10

set tcp [new Agent/TCP]

$tcp set packet\_size\_ 1500

$tcp set fid\_ 1

set sink [new Agent/TCPSink]

set ftp [new Application/FTP]

$ns attach-agent $send $tcp

$ns attach-agent $rec $sink

$ns connect $tcp $sink

$ftp attach-agent $tcp

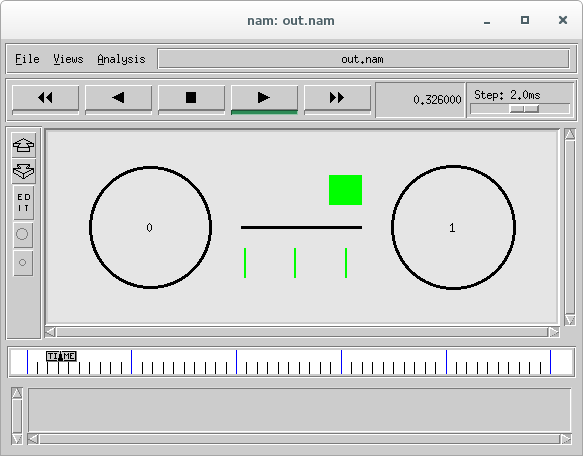
set size [expr 10\*1024\*1024]

$ns at 0.1 "$ftp send $size"

$ns at 5.0 "finish"

$ns run

**OUTPUT:**



**4. Write a TCL script to simulate the network described below using ns2.**

Consider a small network with five nodes n0, n1, n2, n3, n4, forming a star topology. The node n4 is at the center. Node n0 is a TCP source, which transmits packets to node n3 (a TCP sink) through the node n4. Node n1 is another traffic source, and sends UDP packets to node n2 through n4. The duration of the simulation time is 10 seconds.

**CODE:**

set ns [new Simulator]

set nf [open out3.nam w]

$ns namtrace-all $nf

$ns color 1 blue

$ns color 2 red

set nf [open tracefile3.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out3.nam &

exit 0

}

set n0 [ $ns node]

set n1 [ $ns node]

set n2 [ $ns node]

set n3 [ $ns node]

set n4 [ $ns node]

$ns duplex-link $n0 $n4 2Mb 10ms DropTail

$ns duplex-link $n1 $n4 2Mb 10ms DropTail

$ns duplex-link $n2 $n4 2Mb 10ms DropTail

$ns duplex-link $n3 $n4 2Mb 10ms DropTail

$ns duplex-link-op $n0 $n4 orient right-down

$ns duplex-link-op $n1 $n4 orient right-up

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

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

set n0\_tcp [new Agent/TCP]

set n1\_udp [new Agent/UDP]

set n3\_tcpsink [new Agent/TCPSink]

set n2\_null [new Agent/Null]

$n0\_tcp set fid\_ 1

$n1\_udp set fid\_ 2

$ns attach-agent $n3 $n3\_tcpsink

$ns attach-agent $n2 $n2\_null

$ns attach-agent $n0 $n0\_tcp

$ns attach-agent $n1 $n1\_udp

$ns connect $n0\_tcp $n3\_tcpsink

$ns connect $n1\_udp $n2\_null

set n0\_ftp [new Application/FTP]

set n1\_cbr [new Application/Traffic/CBR]

$n0\_ftp attach-agent $n0\_tcp

$n1\_cbr attach-agent $n1\_udp

$ns at 0.2 "$n0\_ftp start"

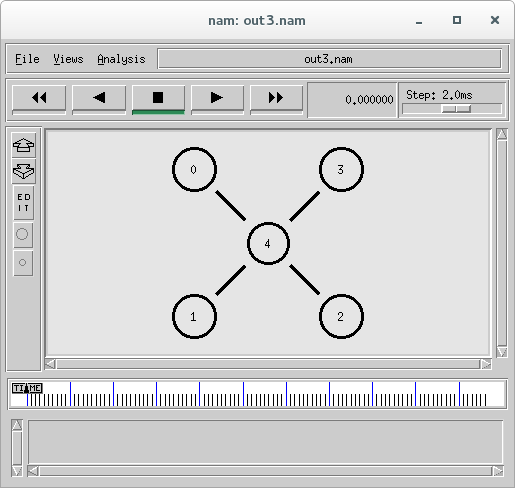
$ns at 0.6 "$n1\_cbr start"

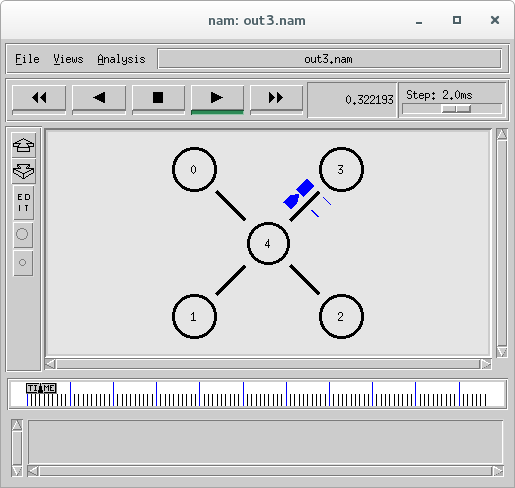
$ns at 10.2 "$n0\_ftp stop"

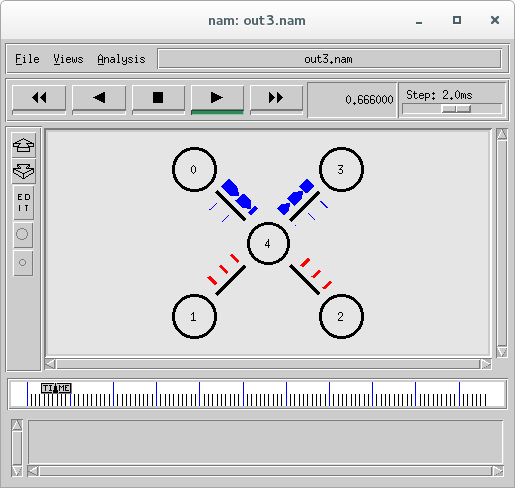
$ns at 10.6 "$n1\_cbr stop"

$ns at 10.7 "finish"

$ns run







**Experiment 3**

1. **Setting up a local area network with ns2. Consider the LAN with seven nodes to be an isolated one i.e. not connected to the Internet. Node # 0 in the LAN act as a UDP traffic source, and node # 6 is the destination node. Assume CBR traffic to be flowing between the nodes. The simulation lasts for 25 seconds. In Ethernet a packet is broadcasted in the shared medium, and only the destination node accepts the packet. Other nodes simply drop it. How many hops a packet should take to travel from node # 0 to node # 6? Verify this from the "Hop Count" plot.**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

set lan [$ns newLan "$n0 $n1 $n2 $n3 $n4 $n5 $n6" 1.0Mb 40ms LL Queue/DropTail MAC/Csma/Cd Channel]

set udp [new Agent/UDP]

$ns attach-agent $n0 $udp

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

set null [new Agent/Null]

$ns attach-agent $n6 $null

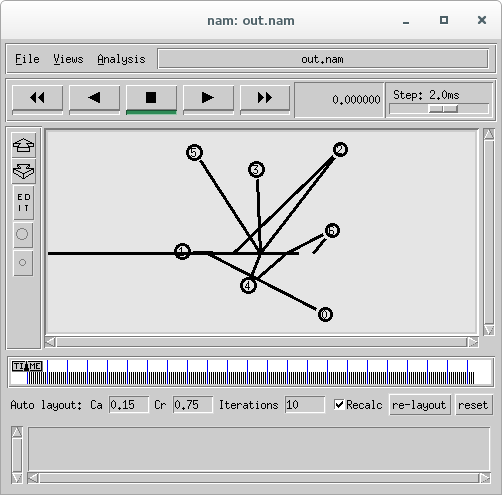
$ns connect $udp $null

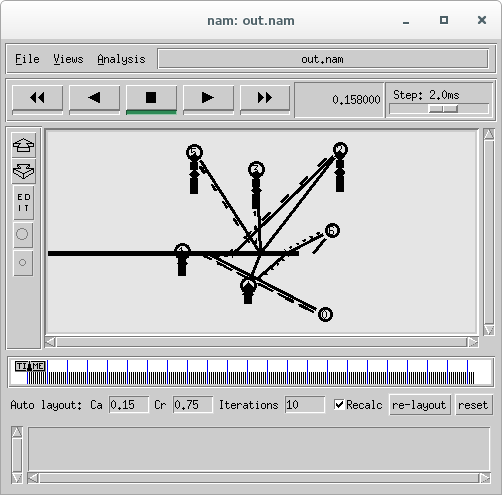
$ns at 0.1 "$cbr start"

$ns at 25.1 "$cbr stop"

$ns at 25.2 "finish"

$ns run





**Experiment 3.2**

**Create the following scenario with two nodes and link in between**.

**– Sender agent: Agent/UDP**

**– Receiver agent: Agent/Null**

**– Connect agents**

**– Data source: Application/Traffic/CBR**

**– Run from 0.5 to 4.5 sec, finish at 5.0 sec**



**CODE :**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

$ns color 1 green

# Open the trace file

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set send [$ns node]

set rec [$ns node]

$ns duplex-link $send $rec 2.0Mb 10ms DropTail

set udp [new Agent/UDP]

$udp set fid\_ 1

set null [new Agent/Null]

set cbr [new Application/Traffic/CBR]

$ns attach-agent $send $udp

$ns attach-agent $rec $null

$ns connect $udp $null

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

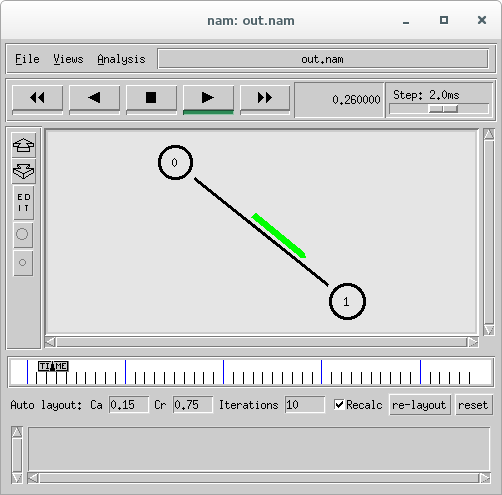
$cbr set rate\_ 1mb

$ns at 0.1 "$cbr start"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run



**EXPERIMENT 3.3**

**Create the following scenario and connect the appropriate agents**

* **Start the FTP application at t = 0.5s**
* **Start the CBR data source at t = 1s**
* **Terminate both at t = 4.5 s**
* **Visualize the bottle neck queue**



**CODE :**

set ns [new Simulator]

set nf [open out2.nam w]

$ns namtrace-all $nf

$ns color 1 blue

$ns color 2 red

# Open the trace file

set nf [open tracefile2.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out2.nam &

exit 0

}

set tcpnode [$ns node]

set udpnode [$ns node]

set commonnode [$ns node]

set finalnode [$ns node]

$ns duplex-link $tcpnode $commonnode 1Mb 10ms DropTail

$ns simplex-link $udpnode $commonnode 2Mb 10ms DropTail

$ns duplex-link $commonnode $finalnode 2Mb 10ms DropTail

$ns queue-limit $commonnode $finalnode 80

#$ns queue-limit $udpnode $commonnode 80

$ns duplex-link-op $tcpnode $commonnode orient right-down

$ns simplex-link-op $udpnode $commonnode orient right-up

$ns duplex-link-op $commonnode $finalnode orient right

set tcp [new Agent/TCP]

set udp [new Agent/UDP]

set tcpsink [new Agent/TCPSink]

set null [new Agent/Null]

$tcp set fid\_ 1

$udp set fid\_ 2

$ns attach-agent $tcpnode $tcp

$ns attach-agent $udpnode $udp

$ns attach-agent $finalnode $null

$ns attach-agent $finalnode $tcpsink

$ns connect $tcp $tcpsink

$ns connect $udp $null

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1.5mb

$ns at 0.2 "$ftp start"

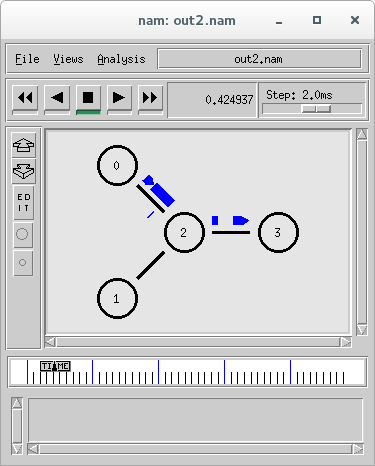
$ns at 0.5 "$cbr start"

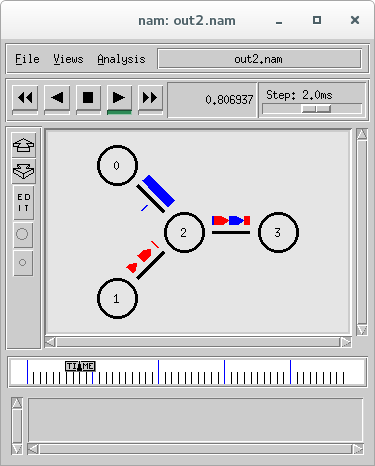
$ns at 4.5 "$ftp stop"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run

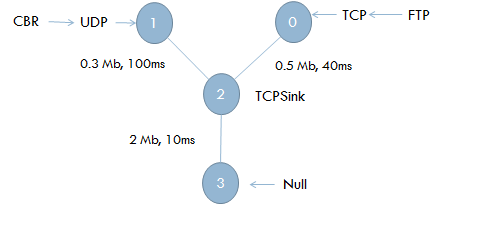


****

**Experiment 3.4**

**In the following simulation scenario set the following parameters:**

* + - **Duplex link between n1 and n2**
    - **Simplex link between node n0 and n2**
    - **Queue Size of link n2-n3 to 40**
    - **CBR traffic packet size: 1000B, inter-arrival time: 8ms, start at time 1.0; TCP window size 8000, packet size 512B**
    - **Generate nam trace and show animation**



**Code:**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

$ns color 1 Green

$ns color 2 red

# Open the trace file

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

set tcpnode [$ns node]

set udpnode [$ns node]

set commonnode [$ns node]

set nullnode [$ns node]

$ns duplex-link $tcpnode $commonnode 1Mb 10ms DropTail

$ns simplex-link $udpnode $commonnode 2Mb 10ms DropTail

$ns simplex-link $commonnode $nullnode 2Mb 10ms DropTail

$ns queue-limit $commonnode $nullnode 40

#$ns queue-limit $udpnode $commonnode 80

$ns duplex-link-op $tcpnode $commonnode orient right-down

$ns simplex-link-op $udpnode $commonnode orient right-up

$ns simplex-link-op $commonnode $nullnode orient right

set tcp [new Agent/TCP]

set udp [new Agent/UDP]

set tcpsink [new Agent/TCPSink]

set null [new Agent/Null]

$tcp set fid\_ 1

$udp set fid\_ 2

$ns attach-agent $tcpnode $tcp

$ns attach-agent $udpnode $udp

$ns attach-agent $nullnode $null

$ns attach-agent $commonnode $tcpsink

$ns connect $tcp $tcpsink

$ns connect $udp $null

set ftp [new Application/FTP]

$ftp attach-agent $tcp

$tcp set window\_ 8000

$tcp set packet\_size\_ 512

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set interval\_ 8ms

$ns at 0.2 "$ftp start"

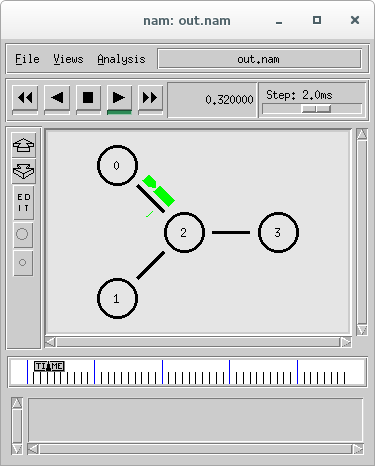
$ns at 1.0 "$cbr start"

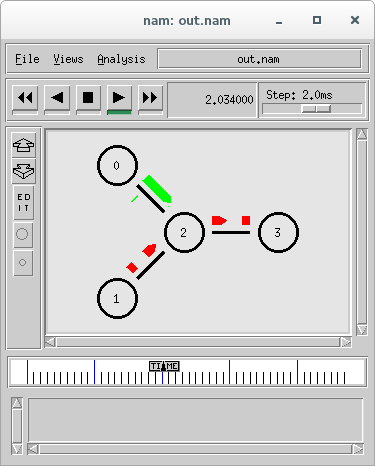
$ns at 4.5 "$ftp stop"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

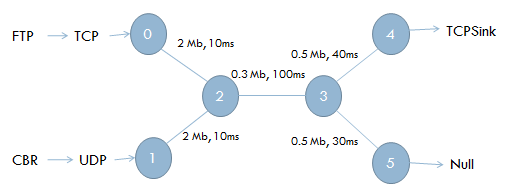
$ns run

****

****

**Experiment 3.5**

1. **Generate the following simulation scenario**



**CODE:**

set ns [new Simulator]

set nf [open out2.nam w]

$ns namtrace-all $nf

$ns color 1 blue

$ns color 2 red

# Open the trace file

set nf [open tracefile2.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out2.nam &

exit 0

}

set tcpnode [$ns node]

set udpnode [$ns node]

set commonnode1 [$ns node]

set commonnode2 [$ns node]

set sinknode [$ns node]

set nullnode [$ns node]

$ns duplex-link $tcpnode $commonnode1 2Mb 10ms DropTail

$ns simplex-link $udpnode $commonnode1 2Mb 10ms DropTail

$ns duplex-link $commonnode1 $commonnode2 0.3Mb 100ms DropTail

$ns duplex-link $commonnode2 $sinknode 0.5Mb 40ms DropTail

$ns simplex-link $commonnode2 $nullnode 0.5Mb 30ms DropTail

$ns duplex-link-op $tcpnode $commonnode1 orient right-down

$ns simplex-link-op $udpnode $commonnode1 orient right-up

$ns duplex-link-op $commonnode1 $commonnode2 orient right

$ns duplex-link-op $commonnode2 $sinknode orient right-up

$ns simplex-link-op $commonnode2 $nullnode orient right-down

set tcp [new Agent/TCP]

set udp [new Agent/UDP]

set tcpsink [new Agent/TCPSink]

set null [new Agent/Null]

$tcp set fid\_ 1

$udp set fid\_ 2

$ns attach-agent $tcpnode $tcp

$ns attach-agent $udpnode $udp

$ns attach-agent $nullnode $null

$ns attach-agent $sinknode $tcpsink

$ns connect $tcp $tcpsink

$ns connect $udp $null

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$cbr set packet\_size\_ 1000

$cbr set rate\_ 1.5mb

$ns at 0.1 "$ftp start"

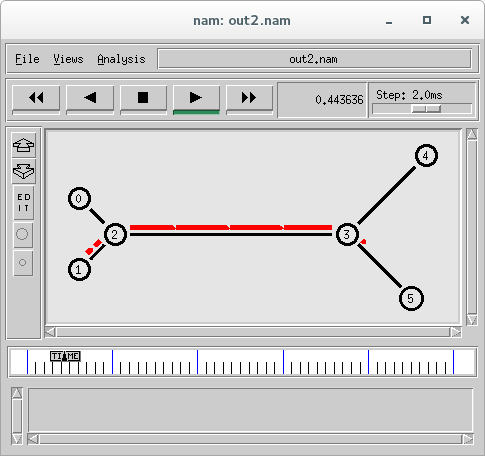
$ns at 0.3 "$cbr start"

$ns at 4.5 "$ftp stop"

$ns at 4.5 "$cbr stop"

$ns at 5.0 "finish"

$ns run



**Experiment 3.6**

**Design Ring Topology of 10 nodes (n0 to n9) with nodes connected in ascending order and n9 connected to n0. Create and connect the nodes using for loop.**

**Code:**

set ns [new Simulator]

set nf [open out.nam w]

$ns namtrace-all $nf

set nf [open tracefile.tr w]

$ns trace-all $nf

proc finish {} {

global ns nf

$ns flush-trace

close $nf

exec nam out.nam &

exit 0

}

for {set i 0} {$i < 10} {incr i} {

set n($i) [$ns node]

}

for {set i 0} {$i < 10} {incr i} {

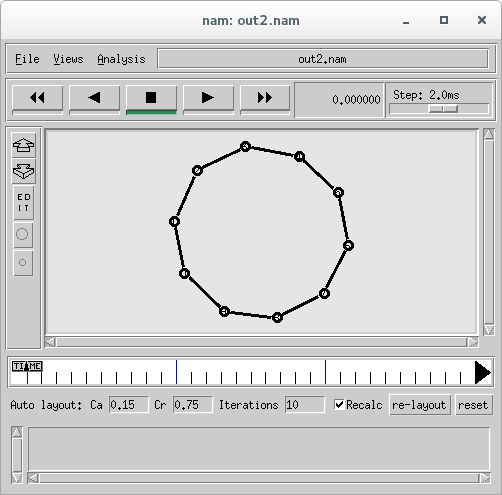
$ns duplex-link $n($i) $n([expr ($i+1)%10]) 0.3Mb 100ms DropTail

}

$ns at 1.0 "finish"

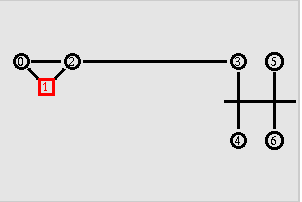
$ns run

**OUTPUT :**



**Experiment 4.1 Simulating Link Error using NS2**

Consider the following topology:



Here node # 2 act as a router. Any traffic to or from the LAN passes through it. Consider node # 1 running a FTP server, and node # 5 is downloading a file of size 4 MB. However, the link between node # 2 and # 3 is fault. It drops packets with a fixed probability of 0.2. Implement a link error model to reflect this.

It may be noted here that the file download time will be more than the we had in exercise # 2 of experiment # 1. Try different values of the simulation time to ensure that the file has been entirely transferred. Is the plot of bytes received a linear curve or non-linear? Why?

**Presence of link errors cause one or more packets to be retransmitted. Verify this from the "Packet Retransmissions" plot.**

**Code:**

set ns [new Simulator]

set namfile [open out.nam w]

$ns namtrace-all $namfile

set tracefile [open out.tr w]

$ns trace-all $tracefile

LanRouter set debug\_ 0

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

$n1 color red

$n1 shape square

$ns duplex-link $n0 $n1 1Mb 10ms DropTail

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

$ns duplex-link $n0 $n2 1Mb 10ms DropTail

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

$ns duplex-link $n1 $n2 1Mb 10ms DropTail

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

$ns duplex-link $n2 $n3 1Mb 10ms DropTail

$ns duplex-link-op $n2 $n3 orient right

set lan [$ns newLan "$n3 $n4 $n5 $n6" 10Mb 10ms LL Queue/DropTail Mac/802\_3 Channel]

set errmodel [new ErrorModel]

$errmodel set rate\_ 0.2

$errmodel ranvar [new RandomVariable/Uniform]

$errmodel drop-target [new Agent/Null]

$ns lossmodel $errmodel $n2 $n3

set tcp [new Agent/TCP]

$ns attach-agent $n1 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n5 $sink

$ns connect $tcp $sink

set ftp [new Application/FTP]

$ftp attach-agent $tcp

set filesize [expr 4\*1024\*1024]

$ns at 0.0 "$ftp send $filesize"

proc finish {} {

global ns namfile tracefile

$ns flush-trace

close $namfile

close $tracefile

set awkCode {

BEGIN{}

{

if($1 == "r" && $4 == 5 && $6 > 1000)

{

count\_bytes = count\_bytes + $6 - ($6 % 1000);

print $2,count\_bytes >> "exp4\_1.data";

}

else

if($1 == "d" && $5 == "tcp" && $6 > 1000)

{

count\_packets++; print $2,count\_packets >> "exp4\_1\_packets.data";

}

}

END{}

}

exec awk $awkCode out.tr

exec nam out.nam &

exec xgraph -bb -tk -x Time -y Bytes exp4\_1.data -bg white &

exec xgraph -bb -tk -x Time -y packets exp4\_1\_packets.data -bg white &

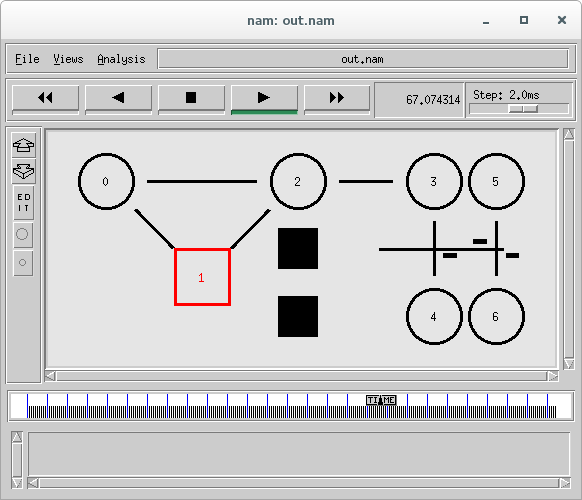
exit 0

}

$ns at 100.0 "finish"

$ns run

**OUTPUT :**



**EXP 5:**

**Assume node # 0 running a FTP application (over TCP) and sending data to node # 6. Node # 1 is sending CBR data node # 7. Assume all the links except 2-3 has a bandwidth of 1 Mb, propagation delay of 10ms and queue type as DropTail. (All are duplex links).**

Dumbbell 
topology with bottleneck

**CODE:**

set ns [new Simulator]

$ns color 1 Pink

$ns color 2 Yellow

set nf [open out36.nam w]

$ns namtrace-all $nf

proc finish {} {

  global ns nf

  $ns flush-trace

  close $nf

  exec nam out36.nam &

  exit 0

}

set n0 [$ns node]

set n1 [$ns node]

set n2 [$ns node]

set n3 [$ns node]

set n4 [$ns node]

set n5 [$ns node]

set n6 [$ns node]

set n7 [$ns node]

$n1 color Red

$n1 shape box

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

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

$ns duplex-link $n2 $n3 1Mb 10ms DropTail

$ns duplex-link $n3 $n4 2Mb 10ms DropTail

$ns duplex-link $n4 $n6 2Mb 10ms DropTail

$ns duplex-link $n3 $n5 2Mb 10ms DropTail

$ns duplex-link $n5 $n7 2Mb 10ms DropTail

$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 $n3 $n4 orient right-up

$ns duplex-link-op $n4 $n6 orient right

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

$ns duplex-link-op $n5 $n7 orient right

set tcp [new Agent/TCP]

$ns attach-agent $n0 $tcp

set sink [new Agent/TCPSink]

$ns attach-agent $n6 $sink

$ns connect $tcp $sink

set ftp [new Application/FTP]

$ftp set PacketSize\_ 1000

$ftp set Rate\_ 1mb

$ftp attach-agent $tcp

set udp [new Agent/UDP]

$ns attach-agent $n1 $udp

set null [new Agent/Null]

$ns attach-agent $n7 $null

$ns connect $udp $null

set cbr [new Application/Traffic/CBR]

$cbr attach-agent $udp

$cbr set type\_ CBR

$tcp set fid\_ 1

$udp set fid\_ 2

$ns at 0.5 "$cbr start"

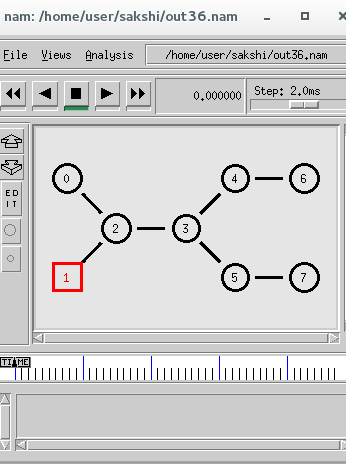
$ns at 0.9 "$ftp start"

$ns at 2.5 "$cbr stop"

$ns at 4.5 "$ftp stop"

$ns at 10.0 "finish"

$ns run



**EXP 6:**

***WIRESHARK EXPERIMENT:***

