

Block Diagram

2) Implement transmission of ping messages / trace route over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

```
set ns [new Simulator]
set f [open lab2.tr w]
set nf [open lab2.nam w]
```

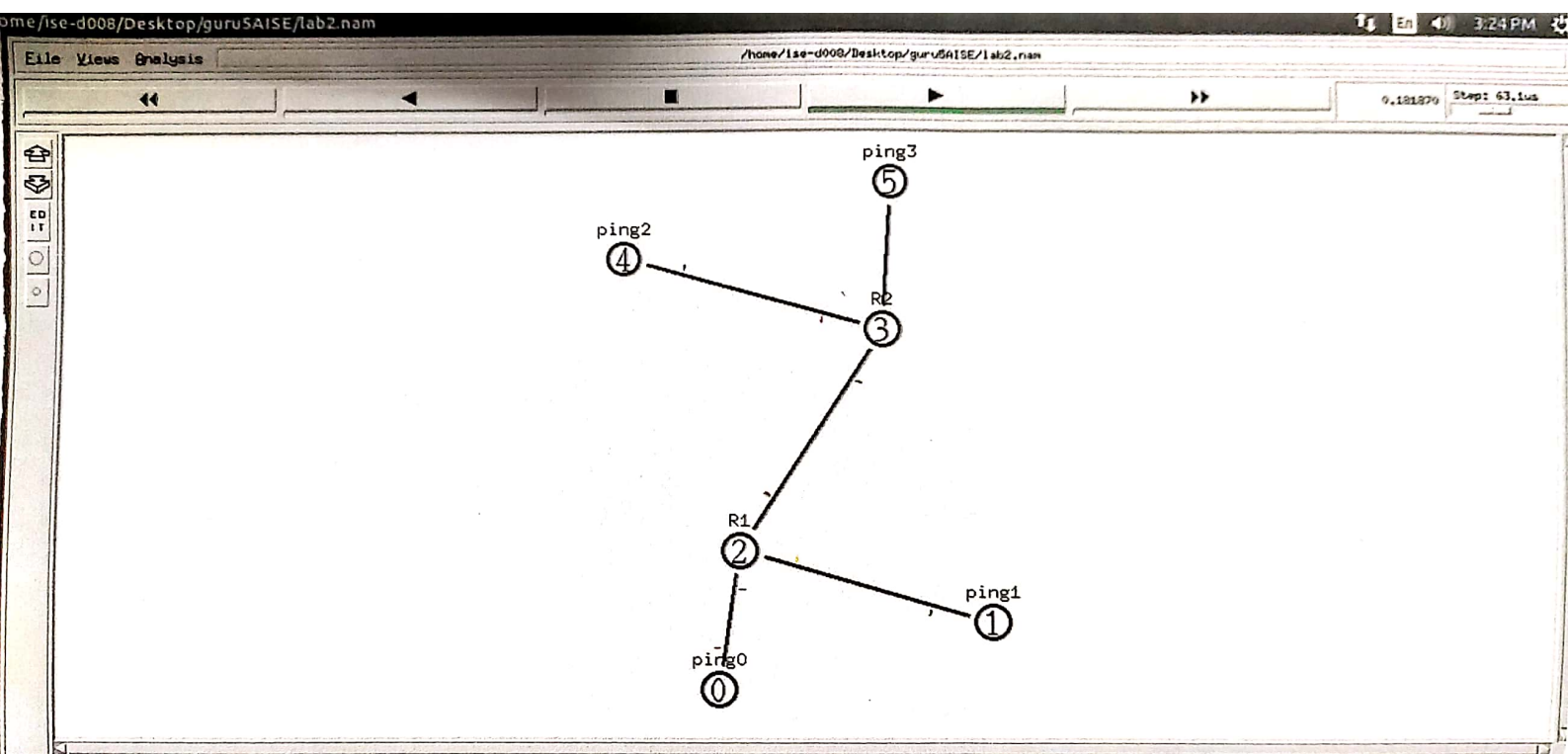
```
$ns trace-all $f
$ns namtrace-all $nf
```

```
proc finish {} {
```

```
    global ns f nf
    $ns flush-trace
    close $f
    close $nf
    exec nam lab2.nam &
    exit 0
}
```

# to create node

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





```

set n3 [$np node]
set n4 [$np node]
set n5 [$ns node]

```

# to name the nodes

```

$no label "ping0"
$n1 label "ping1"
$n2 label "p1"
$n3 label "p2"
$n4 label "ping2"
$n5 label "ping3"

```

# to colour

```

$np color 1 red
$np color 2 green
$np color 3 blue
$ns color 4 yellow

```

# assign link

```

$np duplex-link $no $n2 1Mb 10Mbps PropTail
$np duplex-link $n1 $n2 1Mb 10Mbps DropTail
$np duplex-link $n2 $n3 1Mb 10Mbps DropTail
$ns duplex-link $n3 $n4 1Mb 10Mbps DropTail
$np duplex-link $n3 $n5 1Mb 10Mbps PropTail

```

set ping0 [new Agent / Ping]

\$ns attach-agent \$no \$ping0

set ping1 [new Agent / Ping]

\$ns attach-agent \$n1 \$ping1

set ping4 [new Agent / Ping]

\$ns attach-agent \$n4 \$ping4

set ping5 [new Agent / Ping]

\$ns attach-agent \$n5 \$ping5

# connect nodes

\$ns connect \$ping0 \$ping4

\$ns connect \$ping1 \$ping5

proc sendPingPacket \$3 {

global ns ping0 ping1  
set intervalTime 0.01

set now [\$ns now]

\$ns at [\$now \$now + \$intervalTime]

"\$ping0 send"

\$ns at [\$now \$now + \$intervalTime]

"\$ping1 send"

\$ns at [\$now \$now + \$intervalTime]

"sendPingPacket"

# below will be called automatically as the node receives from the sender

Agent/Ring instrproc recn ? from rtt}

```
{
  $self->instrproc node-
  puts "The node. [$node_id] received an
  Ack from the node $from
  with RTT $rtt ms"
}
```

```
$ping0 set class - 1
$ping1 set class - 2
$ping4 set class - 4
$ping5 set class - 5
```

```
$ns at 0.01 "sendRingPacket"
$ns at 4.0 "finish"
$ns run
```

ACK file

BEGIN {

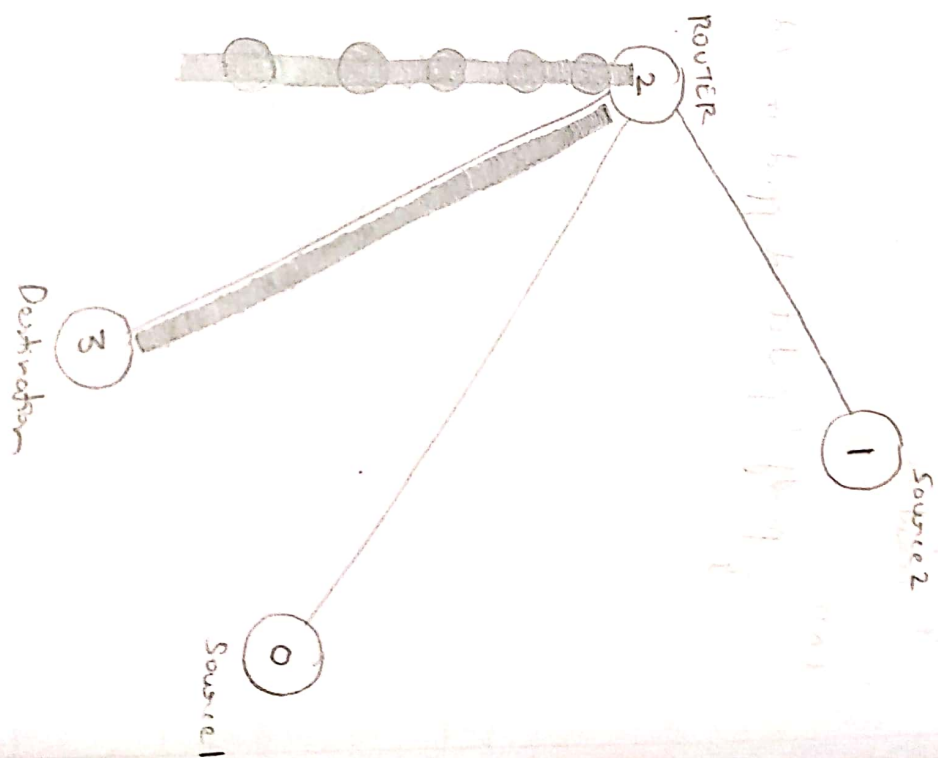
# include <stdio.h>

count = 0

}



```
{  
    if ($1 == "d")  
        count++  
}  
END {  
    print ("packet dropped is " $1 ", count")  
}
```





3] Aim: Implement an ethernet LAN using  $n$  nodes and set multiple traffic nodes and plot congestion window for different source / destination.

```
set ns [new Simulator]
set tf [open lab3.tr w]
set nf [open lab3.nam w]
$ns trace-all $tf
$ns namtrace-all $nf
```

# create nodes

```
set no [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
```

# color and label the nodes

```
$no color "magenta"
$no label "src1"
$ns color "magenta"
$ns label "src2"
$ns color "blue"
$ns label "dst2"
```

```
$n3 color "blue"
$n3 label "dest1"
```

```
$ns make-lan "$no $n1 $n2 $n3 $n4" 10mb 10ms
  lt Queue / DropTail Mac/802.3
```

```
$ns duplex-link $n4 $n5 1mb 1ms DropTail
```

```
set tcpo [new Agent/Tcp]
$ns attach-agent $no $tcpo
```

```
set tcp1 [new Agent/ # ignore
```

```
set tcpo [new Application/FTP]
$tcpo attach-agent $tcpo
```

```
$tcpo set packet-size 5000
$tcpo set interval 0.0001
```

```
set sinks [new Agent/TcpSink]
$ns attach-agent $ns $sinks
```

```
$ns connect $tcpo $sinks
```

```
# connect n2 to n3 using Tcp connection
```

```
set tcp2 [new Agent/Tcp]
$ns attach-agent $n2 $tcp2
```

```
set $hp2 [new Application /FTP]
$hp2 attach-agent $tcp2
```

```
$hp2 set packetSize - 600
$hp2 set interval - 0.001
```

```
set sink3 [new Agent /TCPSink]
$ns attach-agent $ns3 $sink3
```

```
$ns connect $tcp2 $sink3
```

```
set file1 [open file1.txt w]
set file2 [open file2.txt w]
$tcp0 attach $file1
$tcp2 attach $file2
```

```
$tcp0 trace cwnd-
$tcp2 trace cwnd-
```

```
proc finish {} {
```

```
    global ns nf tf
    $ns flush-trace
    close $tf
    close $nf
    exec nam lab3.nam &
    exit 0
}
```



Ank file

$\{$   
 $\Gamma_b (\$b = "cond")$   
 $\quad \text{part}_b ("f \vee t \vee f \vee t \vee n", \$1, \$7);$   
 $\}$

END