Tcl Script:

```
set ns [new Simulator]
set tr [open ping_test_6_node.tr w]
set nf [open ping_test_6_node.nam w]

$ns trace-all $tr
$ns namtrace-all $nf

$ns color 1 "Blue"
$ns color 2 "Green"
$ns color 3 "Red"
$ns color 4 "Yellow"

proc finish {} {

      global ns nf tr
      $ns flush-trace
      close $tr
      close $nf
      exec nam ping_test_6_node.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]

# puts "nodes declared"

$ns duplex-link $n0 $n2 0.11Mb 10ms DropTail
$ns duplex-link $n1 $n2 0.11Mb 10ms DropTail
$ns duplex-link $n3 $n4 0.11Mb 10ms DropTail
$ns duplex-link $n3 $n5 0.11Mb 10ms DropTail
$ns duplex-link $n3 $n2 0.11Mb 10ms DropTail

$ns queue-limit $n2 $n3 10

$ns duplex-link-op $n0 $n2 orient right-down
$ns duplex-link-op $n1 $n2 orient right-up
$ns duplex-link-op $n3 $n4 orient right-up
$ns duplex-link-op $n3 $n5 orient right-down
$ns duplex-link-op $n3 $n2 orient left

# puts "nodes initialized"

set ping0 [new Agent/Ping]
set ping1 [new Agent/Ping]
set ping4 [new Agent/Ping]
set ping5 [new Agent/Ping]
```

```
# puts "pings declared"

$ping0 set class_ 1
$ping1 set class_ 2
$ping4 set class_ 3
$ping5 set class_ 4

$ns attach-agent $n0 $ping0
$ns attach-agent $n1 $ping1
$ns attach-agent $n4 $ping4
$ns attach-agent $n5 $ping5

# puts "nodes attached to the pings"

$ns connect $ping0 $ping5
$ns connect $ping0 $ping4
$ns connect $ping1 $ping5
$ns connect $ping1 $ping4

proc sendPingPacket {} {
      global ns ping0 ping1 ping4 ping5
      set now [$ns now]
      set pinginterval 0.01
      $ns at [expr $now + $pinginterval] "$ping0 send"
      $ns at [expr $now + $pinginterval] "$ping1 send"
      $ns at [expr $now + $pinginterval] "$ping4 send"
      $ns at [expr $now + $pinginterval] "$ping5 send"
      $ns at [expr $now + $pinginterval] "sendPingPacket"
}

Agent/Ping instproc recv { from rtt } {
      $self instvar node_
      puts "ping from $from to [$node_ id] with rtt=$rtt"
}

$ns at 0.01 "sendPingPacket"
$ns rtmodel-at 3.0 down $n2 $n3
$ns rtmodel-at 5.0 up $n2 $n3
$ns at 10.0 "finish"

$ns run
```

AWK File:

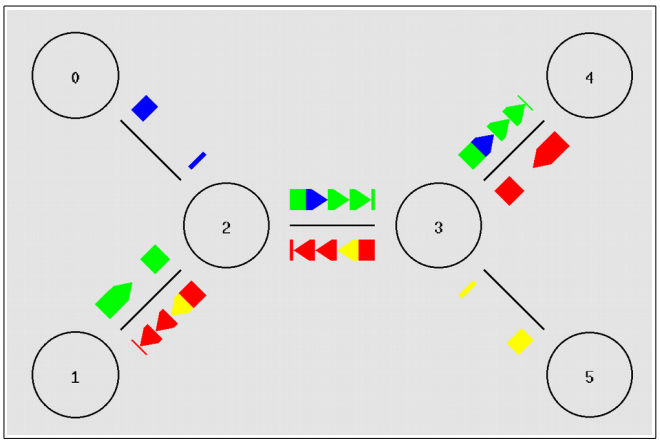
```
BEGIN {
      droppedPackets = 0;
}
{
      if($1 == "d"){ droppedPackets++; }
}
END {
      printf("No. of dropped packets = %d\n", droppedPackets);
}
```

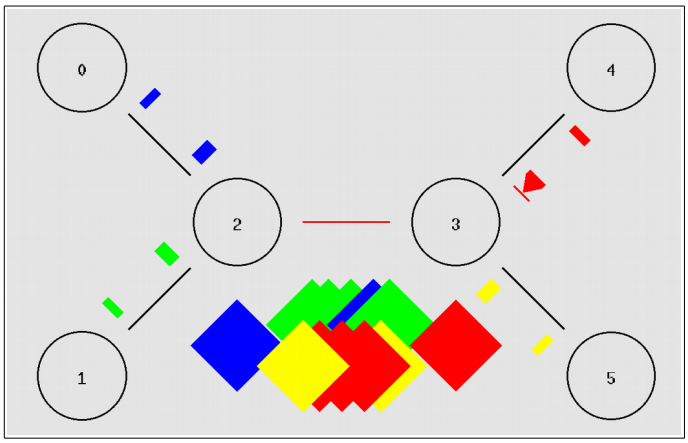
Xgraph File:

```
TitleText: Packet dropping in ping testing of 6 nodes
XUnitText: Bandwidth in Mbps
YUnitText: Number of packets dropped

"Ping Test"
0.11 1558
0.12 1401
0.13 1395
0.14 1007
0.15 864
0.16 710
0.17 891
0.18 538
0.19 240
0.20 66
```


Close
Hdcpy
About
Number of packets dropped x 10^2
Packet dropping in ping testing of 6 nodes
Ping Test
1.6000
1.5500
1.5000
1.4500
1.4000
1.3500
1.3000
1.2500
1.2000
1.1500
1.1000
1.0500
1.0000
0.9500
0.9000
0.8500
0.8000
0.7500
0.7000
0.6500
0.6000
0.5500
0.5000
0.4500
0.4000
0.3500
0.3000
0.2500
0.2000
0.1500
0.1000
0.0500
0.0000
110.0000
115.0000
120.0000
125.0000
130.0000
135.0000
140.0000
145.0000
150.0000
155.0000
160.0000
165.0000
170.0000
175.0000
180.0000
185.0000
190.0000
195.0000
200.0000
Bandwidth in Mbps x 10^-3


0
1
2
3
4
5


0
1
2
3
4
5