

```
#——Event scheduler object creation——#  
set ns [new Simulator]
```

```
#——creating trace objects——#  
set nt [open PDR.tr w]  
$ns trace-all $nt
```

```
#——creating nam objects——#  
set nf [open PDR.nam w]  
$ns namtrace-all $nf
```

```
$ns use-newtrace
```

```
#——finish procedure——#  
proc finish {} {  
    global ns nf nt  
    $ns flush-trace  
    close $nf  
    close $nt  
    puts "running nam..."  
    exec nam PDR.nam &  
    exec awk -f PDR.awk PDR.tr &  
    exit 0  
}
```

```
#——Setting color ID——#  
$ns color 1 darkmagenta  
$ns color 2 yellow  
$ns color 3 blue  
$ns color 4 green  
$ns color 5 black
```

```
#—— Creating Network——#
```

```
set totalNodes 5  
for {set i 0} {$i < $totalNodes} {incr i} {  
    set node_($i) [$ns node]  
}
```

```
set server1 0
```

```
set router1 1
set client 2
set router2 3
set server2 4
```

```
#———— Creating Duplex Link————#
$ns duplex-link $node_($server1) $node_($router1) 2Mb 50ms DropTail
$ns duplex-link $node_($router1) $node_($client) 2Mb 50ms DropTail
$ns duplex-link $node_($server2) $node_($router2) 2Mb 50ms DropTail
$ns duplex-link $node_($router2) $node_($client) 2Mb 50ms DropTail
```

```
$ns duplex-link-op $node_($server1) $node_($router1) orient down
$ns duplex-link-op $node_($router1) $node_($client) orient right
$ns duplex-link-op $node_($server2) $node_($router2) orient down
$ns duplex-link-op $node_($router2) $node_($client) orient left
```

```
#———— Labelling————#
$ns at 0.0 "$node_($server1) label Server1"
$ns at 0.0 "$node_($router1) label Router1"
$ns at 0.0 "$node_($client) label Client"
$ns at 0.0 "$node_($server2) label Server2"
$ns at 0.0 "$node_($router2) label Router2"
```

```
$ns at 0.0 "$node_($server1) color blue"
$ns at 0.0 "$node_($server2) color blue"
$ns at 0.0 "$node_($client) color red"
$ns at 0.0 "$node_($router1) color green"
$ns at 0.0 "$node_($router2) color green"
```

```
$node_($server1) shape hexagon
$node_($server2) shape hexagon
$node_($router1) shape square
$node_($router2) shape square
```

```
#———— Data Transfer between Nodes————#
```

```
# Defining a transport agent for sending
set tcp1 [new Agent/TCP]
```

```
# Attaching transport agent to sender node
```

```
$ns attach-agent $node_($server1) $tcp1
```

```
# Defining a transport agent for receiving  
set sink [new Agent/TCPSink]
```

```
# Attaching transport agent to receiver node  
$ns attach-agent $node_($client) $sink
```

```
#Connecting sending and receiving transport agents  
$ns connect $tcp1 $sink
```

```
#Defining Application instance  
set ftp1 [new Application/FTP]
```

```
# Attaching transport agent to application agent  
$ftp1 attach-agent $tcp1
```

```
# Setting flow color  
$tcp1 set fid_ 4
```

```
# Defining another transport agent for sending  
set tcp2 [new Agent/TCP]  
$ns attach-agent $node_($server2) $tcp2  
set sink [new Agent/TCPSink]  
$ns attach-agent $node_($client) $sink  
$ns connect $tcp2 $sink
```

```
set ftp2 [new Application/FTP]  
$ftp2 attach-agent $tcp2
```

```
$tcp2 set fid_ 4
```

```
# data packet generation starting time  
$ns at 1.0 "$ftp1 start"
```

```
# data packet generation ending time  
$ns at 3.0 "$ftp1 stop"
```

```
# data packet generation starting time
```

\$ns at 2.0 "\$ftp2 start"

# data packet generation ending time

\$ns at 4.0 "\$ftp2 stop"

#Calling finish procedure

\$ns at 5.0 "finish"

\$ns run

## **AWK Script**

```
BEGIN{
    receive=0
    drop=0
    total=0
    ratio=0
}

{
    if($1=="r" && $8==4)
    {
        receive++
    }

    if($1=="d" && $8==4)
    {
        drop++
    }
}

END{
    total=receive+drop
    ratio=(receive/total)*100
    printf("\n Total Packet sent: %d", total)
    printf("\n Packet Delivery Ratio: %f", ratio)
}
```