Part -B
NS2 Programming

1. Simulate a three nodes point – to – point network with duplex links between them. Set the queue size and vary the bandwidth and find the number of packets dropped.

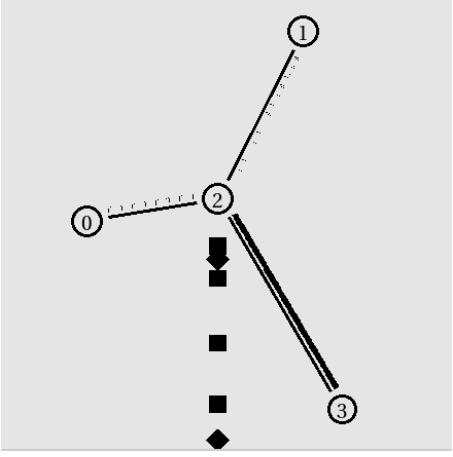
Program :

```
#Initialise new simulator
set ns [new Simulator]
#nf -> lab.tr in write mode
set nf [open lab.tr w]
#tf -> lab.nam in write mode
set tf [open lab.nam w]

$ns trace-all $nf
$ns namtrace-all $tf

#      (0)   (1)
#         \   /
#          (2)
#           |
#          (3)
#Connection : (0) ----> (3)
#                     (1) ----> (3)


#Create nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]

#Initialise Application Layer Traffic (Constant Bit Rate) for senders
set cbr0 [new Application/Traffic/CBR]
set cbr1 [new Application/Traffic/CBR]

#Initialise Transport Layer Protocols (UDP) for senders
set udp0 [new Agent/UDP]
set udp1 [new Agent/UDP]

#Initialise a Null agent for receiver
set null3 [new Agent/Null]

#Establish links between the nodes.
$ns duplex-link $n0 $n2 10Mb 300ms DropTail
$ns duplex-link $n1 $n2 10Mb 300ms DropTail
$ns duplex-link $n2 $n3 1Mb 300ms DropTail
#Attach Transport Layer Protocol to Network Layer
```

```
$ns attach-agent $n0 $udp0
$ns attach-agent $n1 $udp1
$ns attach-agent $n3 $null3

#Attach Application Layer Traffic to Transport Layer Protocol
$cbr0 attach-agent $udp0
$cbr1 attach-agent $udp1

#Establish connections between senders and receivers
$ns connect $udp0 $null3
$ns connect $udp1 $null3

#Starting process
$cbr0 set packetSize_ 500Mb
$cbr0 set interval_ 0.005

proc finish { } {
        exec nam lab.nam &
        set ctr 0
        set fid [open lab.tr r]
        while { [gets $fid line] != -1 } {
                if { [string match "*d*" $line] } {
                        set ctr [expr $ctr + 1]
                }
        }
        puts "No of packets dropped: $ctr"
        exit 0
}

$ns at 0.01 "$cbr0 start"
$ns at 0.01 "$cbr1 start"
$ns at 5.0 "finish"
$ns run
```

Output:

No of packets dropped: 345



2. Simulate the different types of Internet traffic such as FTP and TELNET over a network and analyze the throughput.

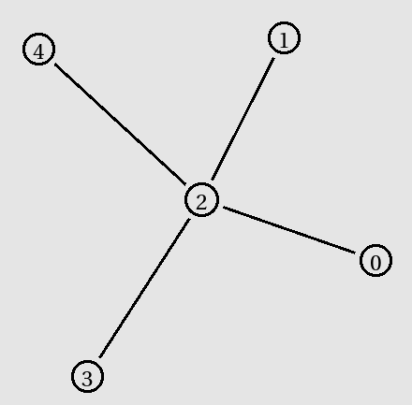
Program:

```
#New Simulator
set ns [new Simulator]

set nf [open lab.nam w]
set tf [open lab.tr w]

$ns trace-all $tf
$ns namtrace-all $nf

#   (0)   (1)
#     \   /
#      (2)
#     /   \
#   (3)   (4)
# Connection : 0 ----> 3
#                    1 ----> 4

#Create nodes
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#Initialise transport layer protocols
set tcp0 [new Agent/TCP]
set tcp1 [new Agent/TCP]
set sink3 [new Agent/TCPSink]
set sink4 [new Agent/TCPSink]

#Initialise application layer protocols
set ftp0 [new Application/FTP]
set telnet1 [new Application/Telnet]

#Establish links between the nodes
$ns duplex-link $n0 $n2 100Mb 300ms DropTail
$ns duplex-link $n1 $n2 100Mb 300ms DropTail
$ns duplex-link $n3 $n2 100Mb 300ms DropTail
$ns duplex-link $n4 $n2 100Mb 300ms DropTail

#Attach transport layer protocols to network layer
$ns attach-agent $n0 $tcp0
$ns attach-agent $n1 $tcp1
$ns attach-agent $n3 $sink3
$ns attach-agent $n4 $sink4
```

```
#Attach application layer protocols to transport layer
$ftp0 attach-agent $tcp0
$telnet1 attach-agent $tcp1

#Connect the nodes
$ns connect $tcp0 $sink3
$ns connect $tcp1 $sink4

$telnet1 set packetSize_ 1000Mb
$telnet1 set interval_ 0.0001

#Process
proc finish { } {
        exec nam lab.nam &
        set ctr0 0
        set ctr1 0
        set tf [open lab.tr r]
        while {[gets $tf line] != -1} {
                if { [string match "*tcp*0.0*3.0*" $line] } {
                        set ctr0 [expr $ctr0 + 1]
                }
                if { [string match "*tcp*1.0*4.0*" $line] } {
                        set ctr1 [expr $ctr1 + 1]
                }
        }

        set thr0 [expr $ctr0/5]
        set thr1 [expr $ctr1/5]
        puts "No of packets FTP: $ctr0"
        puts "Throughput FTP: $thr0"
        puts "No of packets TELNET: $ctr1"
        puts "Throughput TELNET: $thr1"
        exit 0
}

$ns at 0.01 "$ftp0 start"
$ns at 0.01 "$telnet1 start"
$ns at 5.0 "finish"
$ns run
```

Output:

```
No of packets FTP: 96
Throughput FTP: 19
No of packets TELNET: 93
Throughput TELNET: 18
```



3. Simulate an Ethernet LAN using n nodes (6-10), change error rate and data rate and compare the throughput.

Program:

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

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]


#    _____________________
#   |                     |
# [0123](LAN-7)          [456](LAN-8)
#Connection : [2(source)----->5(destination)]

$ns make-lan "$n0 $n1 $n2 $n3" 10Mb 10ms LL Queue/DropTail Mac/802_3
$ns make-lan "$n4 $n5 $n6" 10Mb 10ms LL Queue/DropTail Mac/802_3

$ns duplex-link $n3 $n6 100Mb 10ms DropTail

set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1

set null5 [new Agent/Null]
$ns attach-agent $n5 $null5
```

```
$ns connect $udp1 $null5

$cbr1 set packetSize_ 500Mb
$cbr1 set interval_ 0.005

set err [new ErrorModel]
$ns lossmodel $err $n3 $n6
$err set rate_ 0.2

proc finish { } {
        exec nam lab.nam &
        set ctr0 0
        set thr0 0
        set fid [open lab.tr r]
        while {[gets $fid line] != -1} {
                # 5 - Node 5
                # 8 - LAN-8
                if {[string match "* 8 5 *" $line]} {
                        set ctr0 [expr $ctr0+1]
                }
        }

        set thr [expr $ctr0/5]
        puts "No of packets: $ctr0"
        puts "Throughput : $thr"
        exit 0
}
$ns at 0.01 "$cbr1 start"
$ns at 5.0 "finish"
$ns run
```

Output:

```
No of packets: 791
Throughput : 158
```

4. Simulate an Ethernet LAN using n nodes and set multiple traffic nodes and determine the collision across different nodes.

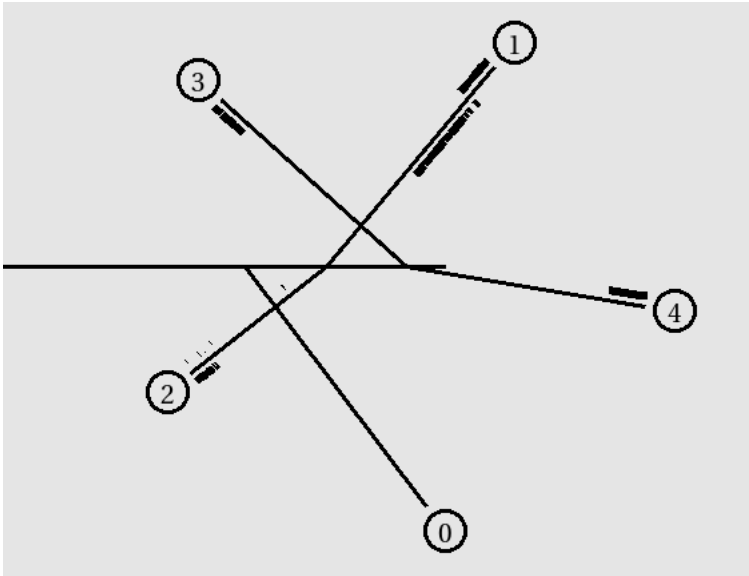
Program:

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

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

```
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]

#       [01234](LAN)

$ns make-lan -trace on "$n0 $n1 $n2 $n3 $n4" 100Mb 10ms LL Queue/DropTail Mac/802_3

set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0

set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1

set udp2 [new Agent/UDP]
$ns attach-agent $n2 $udp2
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp2

set null1 [new Agent/Null]
$ns attach-agent $n1 $null1

set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2

set sink3 [new Agent/TCPSink]
$ns attach-agent $n3 $sink3

$ns connect $tcp0 $sink2
$ns connect $udp2 $null1
$ns connect $tcp1 $sink3

$ftp0 set interval_ 0.001
$ftp1 set interval_ 0.001
$cbr2 set interval_ 0.01

proc finish {} {
exec nam lab.nam &
set ctr0 0
set fid [open lab.tr r]
while {[gets $fid line] !=-1 } {
        if { [string match "c*" $line] } {
                set ctr0 [expr $ctr0 + 1]
        }
}
```

```
puts "No of packets collided: $ctr0"
exit 0
}
$ns at 0.1 "$cbr2 start"
$ns at 1.2 "$ftp0 start"
$ns at 1.3 "$ftp1 start"
$ns at 5.0 "finish"
$ns run
```

Output:
No of packets collided: 134



5. Simulate the transmission of ping messages over a network topology consisting of 6 nodes and find the number of packets dropped due to congestion.

Program:

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

#   [s0][ping]     [ping]     [ping]
#      [n0]         [n1]       [n3]
#         '.         |         .'
#           '.       |       .'
#             '.     |     .'
#               '.   |   .'
#                 '. | .'
#                   [n2]
#                 .' | '.
#               .'   |   '.
#             .'     |     '.
#           .'       |       '.
#         .'         |         '.
#       .'           |           '.
#     [n4]          [n5]         [n6]
#   [ping][d0]   [s1][ping]   [ping][d1]

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]

$ns duplex-link $n0 $n2 100Mb 300ms DropTail
$ns duplex-link $n5 $n2 100Mb 300ms DropTail
$ns duplex-link $n1 $n2 1Mb 300ms DropTail
$ns duplex-link $n3 $n2 1Mb 300ms DropTail
$ns duplex-link $n2 $n4 1Mb 300ms DropTail
$ns duplex-link $n2 $n6 1Mb 300ms DropTail

$ns queue-limit $n0 $n2 5
$ns queue-limit $n5 $n2 5
$ns queue-limit $n2 $n4 3
$ns queue-limit $n2 $n6 2

set ping0 [new Agent/Ping]
```

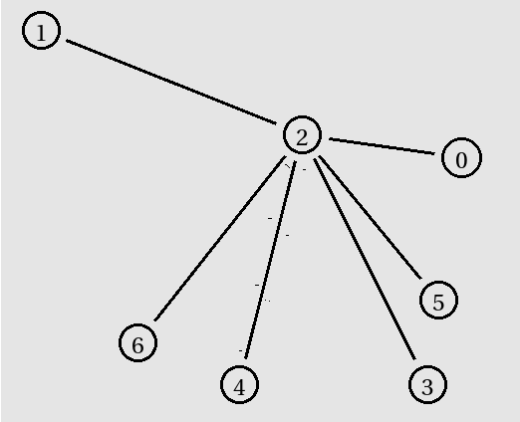
```
set ping4 [new Agent/Ping]
set ping5 [new Agent/Ping]
set ping6 [new Agent/Ping]

$ns attach-agent $n0 $ping0
$ns attach-agent $n4 $ping4
$ns attach-agent $n5 $ping5
$ns attach-agent $n6 $ping6

$ns connect $ping0 $ping4
$ns connect $ping5 $ping6

#Procedure for pinging
Agent/Ping instproc recv {from rtt} {
        $self instvar node_
        puts "The node [$node_ id] received a reply from $from with an RTT of $rtt"
}

#Procedure for packets dropped
proc finish {} {
        exec nam lab.nam &
        set ctr0 0
        set fid [open lab.tr r]
        while {[gets $fid line] !=-1 } {
                if { [string match "*d*" $line] } {
                        set ctr0 [expr $ctr0 + 1]
                }
        }
        puts "No. of packets dropped: $ctr0"
        exit 0
}

$ns rtmodel-at 0.9 down $n2 $n6
$ns rtmodel-at 1.5 up $n2 $n6
$ns at 0.1 "$ping0 send"
$ns at 0.2 "$ping0 send"
$ns at 0.3 "$ping0 send"
$ns at 0.4 "$ping0 send"
$ns at 0.5 "$ping0 send"
$ns at 0.6 "$ping0 send"
$ns at 0.7 "$ping0 send"
$ns at 0.8 "$ping0 send"
$ns at 0.9 "$ping0 send"
$ns at 1.0 "$ping0 send"
$ns at 1.1 "$ping0 send"
$ns at 1.2 "$ping0 send"
$ns at 1.3 "$ping0 send"
$ns at 1.4 "$ping0 send"
$ns at 1.5 "$ping0 send"
```

```
$ns at 1.6 "$ping0 send"
$ns at 1.7 "$ping0 send"
$ns at 1.8 "$ping0 send"
$ns at 1.9 "$ping0 send"
$ns at 0.1 "$ping5 send"
$ns at 0.2 "$ping5 send"
$ns at 0.3 "$ping5 send"
$ns at 0.4 "$ping5 send"
$ns at 0.5 "$ping5 send"
$ns at 0.6 "$ping5 send"
$ns at 0.7 "$ping5 send"
$ns at 0.8 "$ping5 send"
$ns at 0.9 "$ping5 send"
$ns at 1.0 "$ping5 send"
$ns at 1.1 "$ping5 send"
$ns at 1.2 "$ping5 send"
$ns at 1.3 "$ping5 send"
$ns at 1.4 "$ping5 send"
$ns at 1.5 "$ping5 send"
$ns at 1.6 "$ping5 send"
$ns at 1.7 "$ping5 send"
$ns at 1.8 "$ping5 send"
$ns at 1.9 "$ping5 send"
$ns at 5.0 "finish"
$ns run
```

Output:

```
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
```

The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
The node 0 received a reply from 4 with an RTT of 1201.0
The node 5 received a reply from 6 with an RTT of 1201.0
No. of packets dropped: 9



6. Simulate simple ESS with transmitting nodes in wire-less LAN and determine the performance with respect to transmission of packets.

(and)

7. Simulate simple ad-hoc network with transmitting nodes and determine the performance with respect to transmission of packets.

Program:

```
set ns [new Simulator]
set tf [open lab.tr w]
$ns trace-all $tf
set topo [new Topography]
$topo load_flatgrid 1000 1000
set nf [open lab.nam w]
$ns namtrace-all-wireless $nf 1000 1000

$ns node-config -adhocRouting DSDV \
-llType LL \
-macType Mac/802_11 \
-ifqType Queue/DropTail \
-ifqLen 50 \
-phyType Phy/WirelessPhy \
-channelType Channel/WirelessChannel \
-propType Propagation/TwoRayGround \
-antType Antenna/OmniAntenna \
-topoInstance $topo \
-agentTrace ON \
-routerTrace ON
```

```
#GOD or General Operations Director is a ns-2 simulator object, which is used to store global information about the state of the environment,
# network, or nodes

create-god 3
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
$n0 label "tcp0"
$n1 label "sink1/tcp1"
$n2 label "sink2"
#The below code is used to give the initial node positions.
$n0 set X_ 50
$n0 set Y_ 50
$n0 set Z_ 0
$n1 set X_ 100
$n1 set Y_ 100
$n1 set Z_ 0
$n2 set X_ 600
$n2 set Y_ 600
$n2 set Z_ 0


# the simulation  time = 0.1 seconds
# coordinate =  50,50 at a speed of 15 meters per second
# the node n0 will move in this speed towards the coordinate specified


$ns at 0.1 "$n0 setdest 50 50 15"
$ns at 0.1 "$n1 setdest 100 100 25"
$ns at 0.1 "$n2 setdest 600 600 25"
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
set sink1 [new Agent/TCPSink]
$ns attach-agent $n1 $sink1
$ns connect $tcp0 $sink1
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
set sink2 [new Agent/TCPSink]
$ns attach-agent $n2 $sink2
$ns connect $tcp1 $sink2
$ns at 5 "$ftp0 start"
$ns at 5 "$ftp1 start"
#The below code is used to provide the node movements.
```

```
#node n1 will start moving to 550,550 at speed = 15m/s at simulation time = 100

$ns at 20 "$n1 setdest 550 550 150"
$ns at 30 "$n1 setdest 70 70 150"
proc finish {} {
        exec nam lab.nam &
        set ctr1 0
        set ctr2 0
        set thr1 0
        set thr2 0
        set fid [open lab.tr r]

        # 0   1      2 3  4
        # r 0.101003 0 2 tcp 40 ------- 1 0.0 3.0 0 0
        # c 1.592044 2 5 tcp 1054 ------- 0 0.0 2.0 87 1642
        # r 5.026987007 _1_ AGT  --- 7 tcp 1060 [13a 1 0 800] ------- [0:0 1:0 32 1] [2 0] 1 0
        # s 5.000000000 _0_ AGT  --- 3 tcp 40 [0 0 0 0] ------- [0:0 1:0 32 0] [0 0] 0 0

        while {[gets $fid line] != -1} {
            if { [string match "*s*" $line] && [string match "*AGT*" $line]  } {
                  set ctr1 [expr $ctr1 + 1 ]
            }
            set thr1 [expr $ctr1 / 5 ]

        puts "No of  packets transmitted  =   $ctr1"
        puts "Throuput  =   $thr1 packets/sec "
        exit 0
}

$ns at 0.1 "$ns set-animation-rate 1000.0ms"
$ns at 50 "finish"
$ns run
```

Output:

No of packets transmitted = 5234

Throuput = 1046 packets/sec