

## Q1

## Code:

```

set ns [new Simulator -multicast on]
$ns color 1 Blue
$ns color 2 Red
set tracefile1 [open out.tr w]
$ns trace-all $tracefile1
set namfile [open out.nam w]
$ns namtrace-all $namfile

proc finish {} {
    global ns tracefile1 namfile
    $ns flush-trace
    close $tracefile1
    close $namfile
    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 [$ns node]
set n7 [$ns node]
set n8 [$ns node]
# Creating the core event scheduler
set ns [new Simulator]# Turn on multicast support
$ns multicast
$ns color 1 red
$ns color 2 blue# Creating nodes...FILL IN...#
Outputs nam traces
set nf [open out.nam w]
$ns namtrace-all $nf# Creating links#
All with bandwidth Xmb, delay Yms, and queue type# Set queue limit (buffer size)
to Q packets in the bottleneck link
$ns queue-limit node node ...FILL IN...# Set
link orientation for nam ...FILL IN...# Set queue position for nam
$ns duplex-link-op node node queuePos -0.5# Set multicast routing to dense mode...FILL
IN...# Use nam trace format for TCP variable trace, this is required by # nam
visualization of TCP.Agent/TCP
set nam_tracevar true# Create a TCP connection
from sender to receiver...FILL IN...$tcp set fid_ 1
$tcp set window_ 20# Create an FTP source and attach it to the TCP connection...FILL IN...# Create a UDP
connection from node to a multicast group...FILL IN...$udp set fid_ 2# Create a
CBR source and attach it to the UDP connection...FILL IN...# Create dummy
receivers and attach them to thereceiving nodes
set rcvr0 [new Agent/Null]
$ns attach-agent node $rcvr0# Schedule events to happen...FILL IN...# A finish proc
to flush traces and out call nam...FILL IN...# Start the event scheduler
$ns run

$ns duplex-link $n0 $n1 5Mb 2ms DropTail
$ns duplex-link $n1 $n2 5Mb 1ms DropTail
$ns duplex-link $n2 $n3 5Mb 1ms DropTail
$ns duplex-link $n0 $n4 5Mb 2ms DropTail
$ns duplex-link $n4 $n3 5Mb 2ms DropTail
$ns duplex-link $n3 $n5 5Mb 2ms DropTail
$ns duplex-link $n5 $n6 5Mb 2ms DropTail
$ns duplex-link $n5 $n7 5Mb 2ms DropTail
$ns duplex-link $n6 $n8 5Mb 2ms DropTail
$ns duplex-link $n7 $n8 5Mb 2ms DropTail

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

```

```
$ns duplex-link-op $n5 $n6 orient right-up
$ns duplex-link-op $n5 $n7 orient right-down
$ns duplex-link-op $n6 $n8 orient right-down
$ns duplex-link-op $n7 $n8 orient right-up
```

```
set mproto DM
set mrthandle [$ns mrtproto $mproto {}]
```

```
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
set sink [new Agent/TCPSink]
$ns attach-agent $n8 $sink
$ns connect $tcp $sink
$tcp set fid_ 1
```

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

```
set grp0 [Node allocaddr]
```

```
set udp [new Agent/UDP]
$ns attach-agent $n0 $udp
$udp set fid_ 2
$udp set dst_addr_ $grp0
$udp set dst_port_ 0
```

```
set cbr [new Application/Traffic/CBR]
$cbr attach-agent $udp
```

```
set rcvr1 [new Agent/LossMonitor]
set rcvr2 [new Agent/LossMonitor]
set rcvr3 [new Agent/LossMonitor]
```

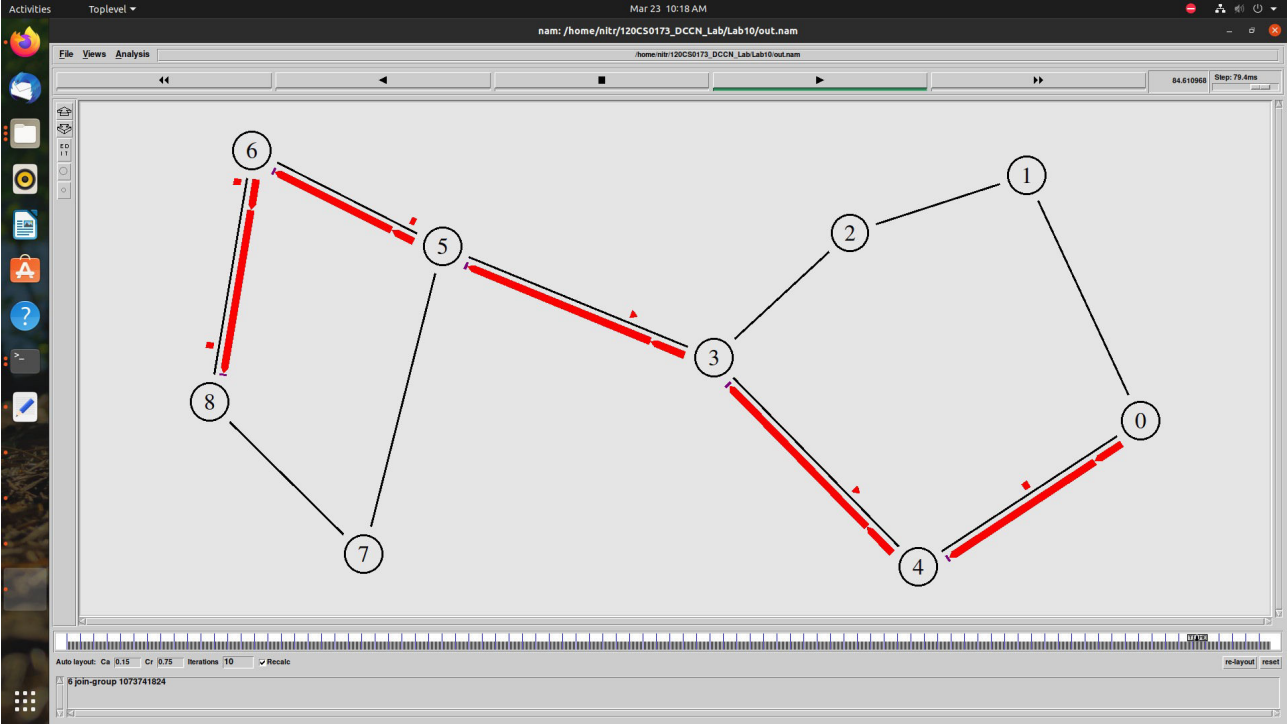
```
$ns attach-agent $n6 $rcvr1
$ns attach-agent $n7 $rcvr2
$ns attach-agent $n8 $rcvr3
```

```
$ns at 0.3 "$n6 join-group $rcvr1 $grp0"
$ns at 0.3 "$n7 join-group $rcvr2 $grp0"
$ns at 0.3 "$n8 join-group $rcvr3 $grp0"
```

```
$ns at 0.1 "$ftp start"
$ns at 0.1 "$cbr start"
$ns at 89.5 "$ftp stop"
$ns at 89.5 "$cbr stop"
```

```
$ns at 90.0 "finish"
```

Output:



**Q2****Code:**

```

=====
#      Simulation parameters setup
=====
set val(chan)    Channel/WirelessChannel    ;# channel type
set val(prop)    Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)   Phy/WirelessPhy           ;# network interface type
set val(mac)     Mac/802_11                ;# MAC type
set val(ifq)     Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)      LL                        ;# link layer type
set val(ant)     Antenna/OmniAntenna        ;# antenna model
set val(ifqlen)  15                        ;# max packet in ifq
set val(nn)      12                        ;# number of mobilenodes
set val(rp)      AODV                      ;# routing protocol
set val(x)       800                       ;# X dimension of topography
set val(y)       800                       ;# Y dimension of topography
set val(stop)    100.0                    ;# time of simulation end

=====
#      Initialization
=====
#Create a ns simulator
set ns [new Simulator]

#Setup topography object
set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)

#Open the NS trace file
set tracefile [open out.tr w]
$ns trace-all $tracefile

#Open the NAM trace file
set namfile [open out.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel

=====
#      Mobile node parameter setup
=====
$ns node-config -adhocRouting $val(rp) \
                -llType      $val(ll) \
                -macType     $val(mac) \
                -ifqType     $val(ifq) \
                -ifqLen      $val(ifqlen) \
                -antType     $val(ant) \
                -propType    $val(prop) \
                -phyType     $val(netif) \
                -channel      $chan \
                -topoInstance $topo \
                -agentTrace  ON \

1,0-1      Top

=====
#      Nodes Definition
=====

```

```

#Create 12 nodes
set n1 [$ns node]
$n1 set X_ 50
$n1 set Y_ 50
$n1 set Z_ 0.0
$ns initial_node_pos $n1 20

set n4 [$ns node]
$n4 set X_ 100
$n4 set Y_ 100
$n4 set Z_ 0.0
$ns initial_node_pos $n4 20

set n2 [$ns node]
$n2 set X_ 150
$n2 set Y_ 150
$n2 set Z_ 0.0
$ns initial_node_pos $n2 20

set n3 [$ns node]
$n3 set X_ 200
$n3 set Y_ 200
$n3 set Z_ 0.0
$ns initial_node_pos $n3 20

set n5 [$ns node]
$n5 set X_ 250
$n5 set Y_ 250
$n5 set Z_ 0.0
$ns initial_node_pos $n5 20

set n6 [$ns node]
$n6 set X_ 300
$n6 set Y_ 300
$n6 set Z_ 0.0
$ns initial_node_pos $n6 20

set n7 [$ns node]
$n7 set X_ 350
$n7 set Y_ 350
$n7 set Z_ 0.0
$ns initial_node_pos $n7 20

set n8 [$ns node]
$n8 set X_ 400
$n8 set Y_ 400
$n8 set Z_ 0.0

et n12 [$ns node]
$n12 set X_ 600
$n12 set Y_ 600
$n12 set Z_ 0.0
$ns initial_node_pos $n12 20

#=====
#           Agents Definition
#=====

set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set sink1 [new Agent/TCPSink]

```

```

$ns attach-agent $n12 $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 0.1 "$ftp1 start"

set tcp2 [new Agent/TCP]
$ns attach-agent $n5 $tcp2
set sink2 [new Agent/TCPSink]
$ns attach-agent $n12 $sink2
$ns connect $tcp2 $sink2
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns at 0.75 "$ftp2 start"

set udp1 [new Agent/UDP]
$ns attach-agent $n5 $udp1
set null1 [new Agent/Null]
$ns attach-agent $n1 $null1
$ns connect $udp1 $null1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns at 0.5 "$cbr1 start"

#=====
#           Applications Definition
#=====

#=====
#           Termination
#=====
#Define a 'finish' procedure
proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}

```

Output:

