Table of contents

Part-A

TCP and UDP

Question: Simulate a point-to-point network with duplex link as follows: n0-n2, n1-n2 and n2-n3. Apply TCPagent between n0-n3 and UDP agent between n1-n3. Apply relevant applications over TCP and UDPagents. Set the queue size to 5 and vary the bandwidth to find the number of packets dropped and received by TCP and UDP agents using awk script and grep command.

Code:

1.tcl:

```
set ns [new Simulator]
```

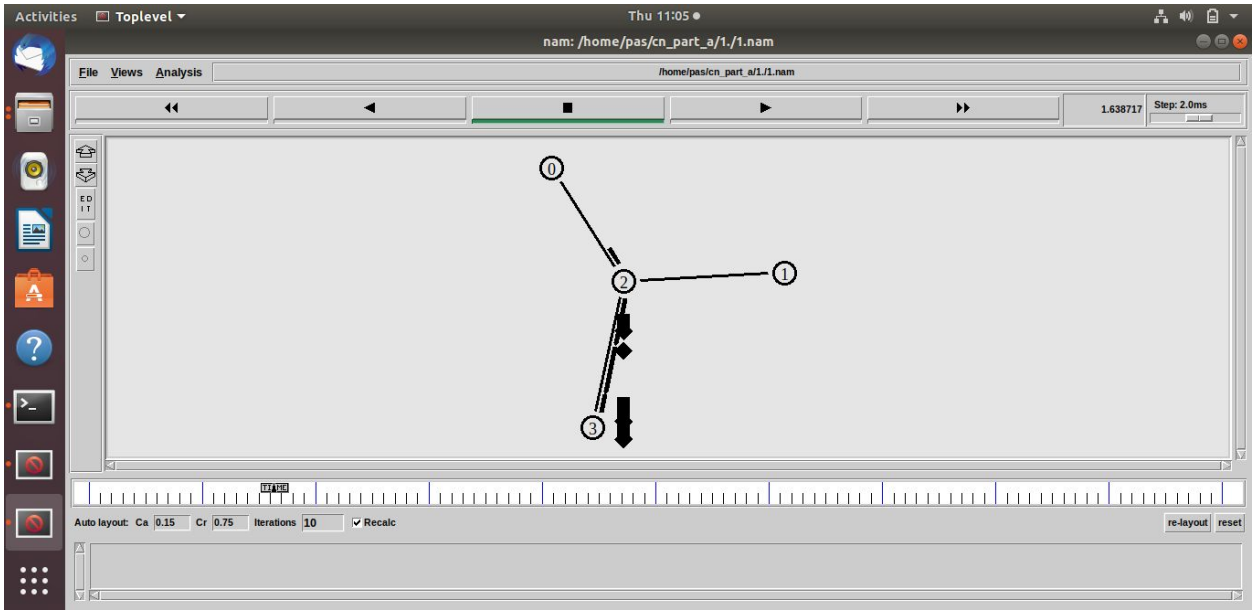
```
set tf [open 1.tr w]
$ns trace-all $tf
set nf [open 1.nam w]
$ns namtrace-all $nf
$ns color 1 Blue
$ns color 2 Red
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
$ns duplex-link $n0 $n2 2Mb 2ms DropTail
$ns duplex-link $n1 $n2 2Mb 2ms DropTail
$ns duplex-link $n2 $n3 0.4Mb 10ms DropTail
$ns queue-limit $n2 $n3 5
set udp1 [new Agent/UDP]
$ns attach-agent $n0 $udp1
set null1 [new Agent/Null]
$ns attach-agent $n3 $null1
$ns connect $udp1 $null1
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$ns at 1.1 "$cbr1 start"
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n3 $sink1
$ns connect $tcp1 $sink1
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 0.1 "$ftp1 start"
$ns at 10.0 "finish"
proc finish {} {
    global ns tf nf
    $ns flush-trace
    close $tf
    close $nf
    puts "running nam..."
    exec nam 1.nam &
    exit 0
}
```

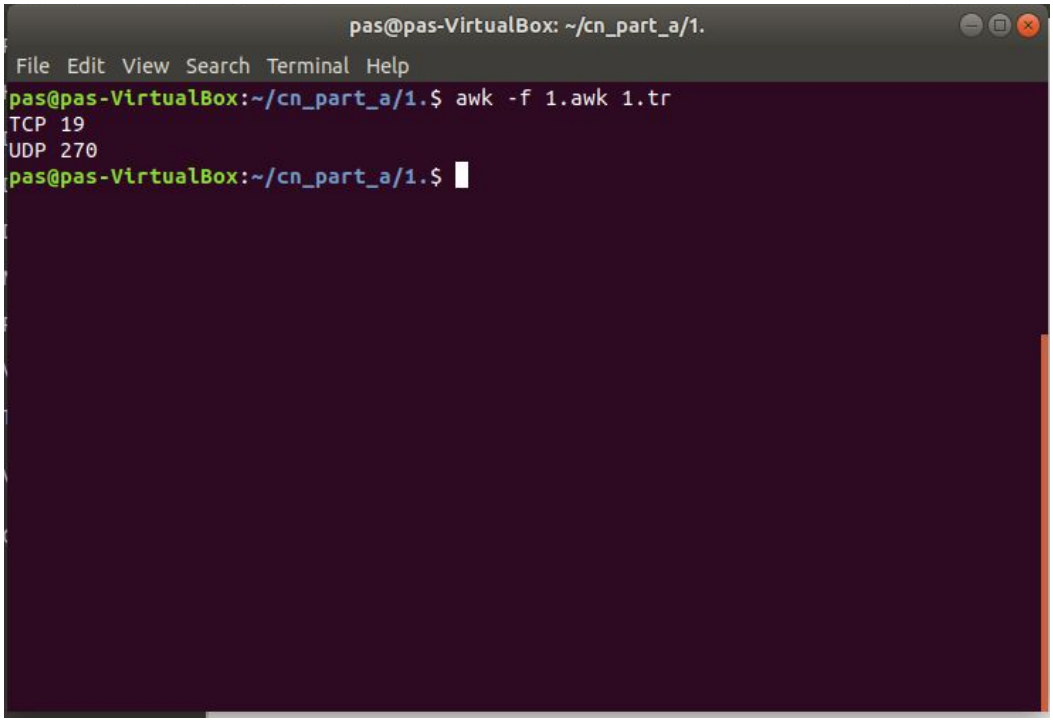
$ns run

1.awk:

```
BEGIN {
tcp_count=0;
udo_count=0;
}{
if($1 == "d" && $5 == "tcp")
tcp_count++;
if($1 == "d" && $5 == "cbr")
udp_count++;
} END {
printf("TCP %d\n",tcp_count);
printf("UDP %d\n",udp_count);
}
```

Output:





FTP and TELNET

Question: Set up the network topology as shown in fig 1. Simulate different types of internet traffic Such as traffic using FTP between the nodes n1 – n6 and Telnet between the nodes n2-n5.

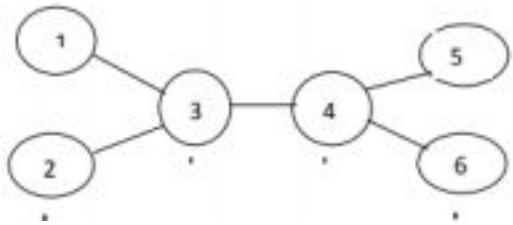


Fig. 1: Network Topology

Plot congestion window for FTP and Telnet, and analyze the throughput.

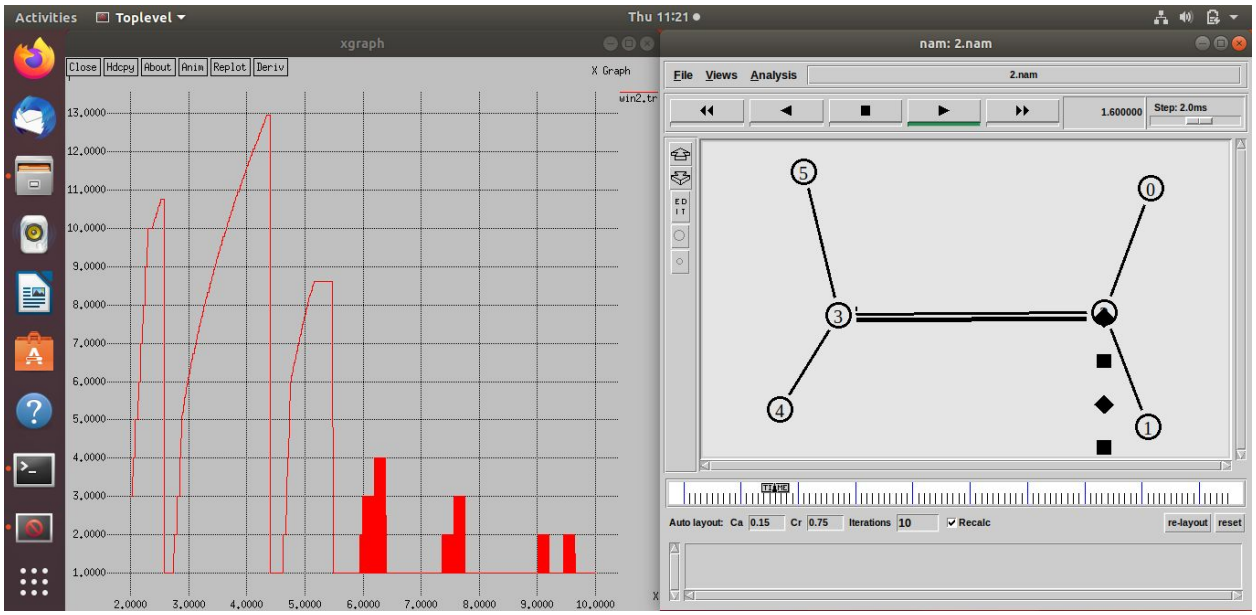
Code:

```
set ns [new Simulator]
set nf [open 2.nam w]
set tf [open 2.tr w]
set cwind [open win2.tr w]
$ns color 1 Blue
```

```
$ns color 2 Red
$ns namtrace-all $nf
$ns trace-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]
$ns duplex-link $n0 $n2 2Mb 2ms DropTail
$ns duplex-link $n2 $n3 0.4Mb 5ms DropTail
$ns duplex-link $n1 $n2 2Mb 2ms DropTail
$ns duplex-link $n3 $n4 2Mb 2ms DropTail
$ns duplex-link $n3 $n5 2Mb 2ms DropTail
$ns queue-limit $n2 $n3 10
set tcp1 [new Agent/TCP]
set sink1 [new Agent/TCPSink]
set ftp1 [new Application/FTP]
$ns attach-agent $n0 $tcp1
$ns attach-agent $n5 $sink1
$ns connect $tcp1 $sink1
$ftp1 attach-agent $tcp1
$ns at 1.2 "$ftp1 start"
set tcp2 [new Agent/TCP]
set sink2 [new Agent/TCPSink]
set telnet1 [new Application/Telnet]
$ns attach-agent $n1 $tcp2
$ns attach-agent $n4 $sink2
$ns connect $tcp2 $sink2
$telnet1 attach-agent $tcp2
$ns at 5.1 "$telnet1 start"
$ns at 5.0 "$ftp1 stop"
$ns at 10.0 "finish"
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file"
}
$ns at 2.0 "plotWindow $tcp1 $cwind"
$ns at 5.5 "plotWindow $tcp2 $cwind"
```

```
proc finish {} {
    global ns tf nf cwind
    $ns flush-trace
    close $tf
    close $nf
    #close $cwind
    puts "running nam..."
    exec nam 2.nam &
    exec xgraph win2.tr &
    exit 0
}
$ns run
```

Output:



TCP agent

Question: Design networks that demonstrate the working of Distance vector routing protocol. The link between node 1 and 4 breaks at 1.0 ms and comes up at 3.0ms. Assume that the source node 0 transmits packets to node 4. Plot the congestion window when TCP

sends packets via other nodes. Assume your own parameters for bandwidth and delay.

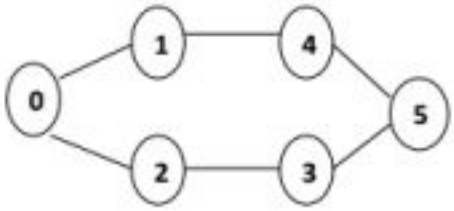


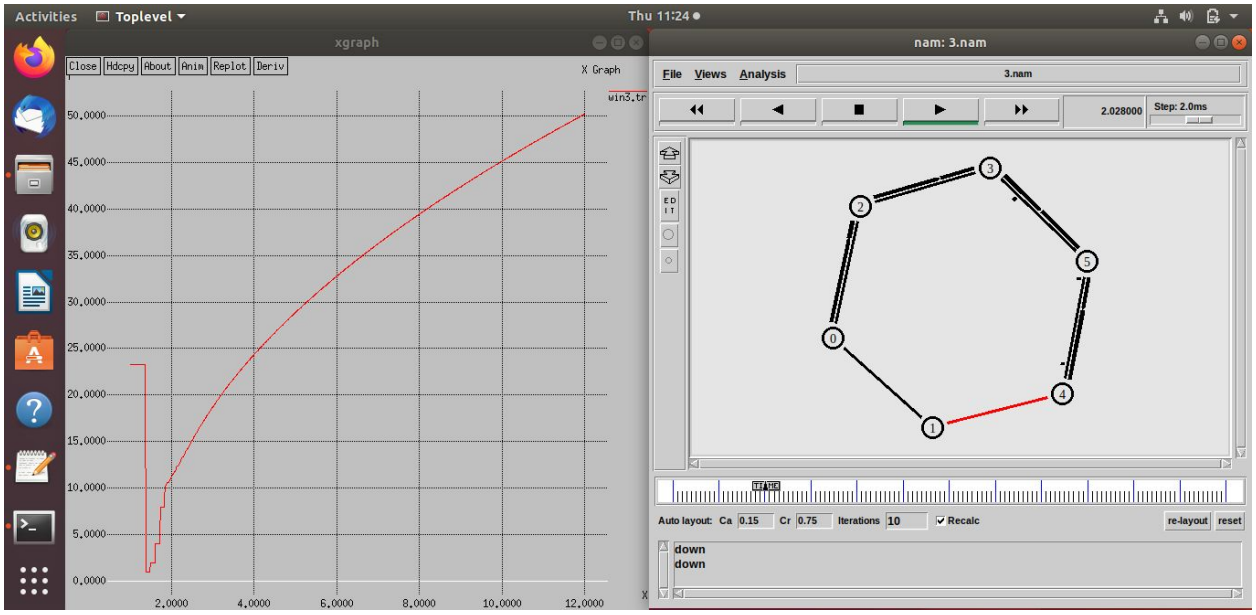
Fig 2: Network Topology

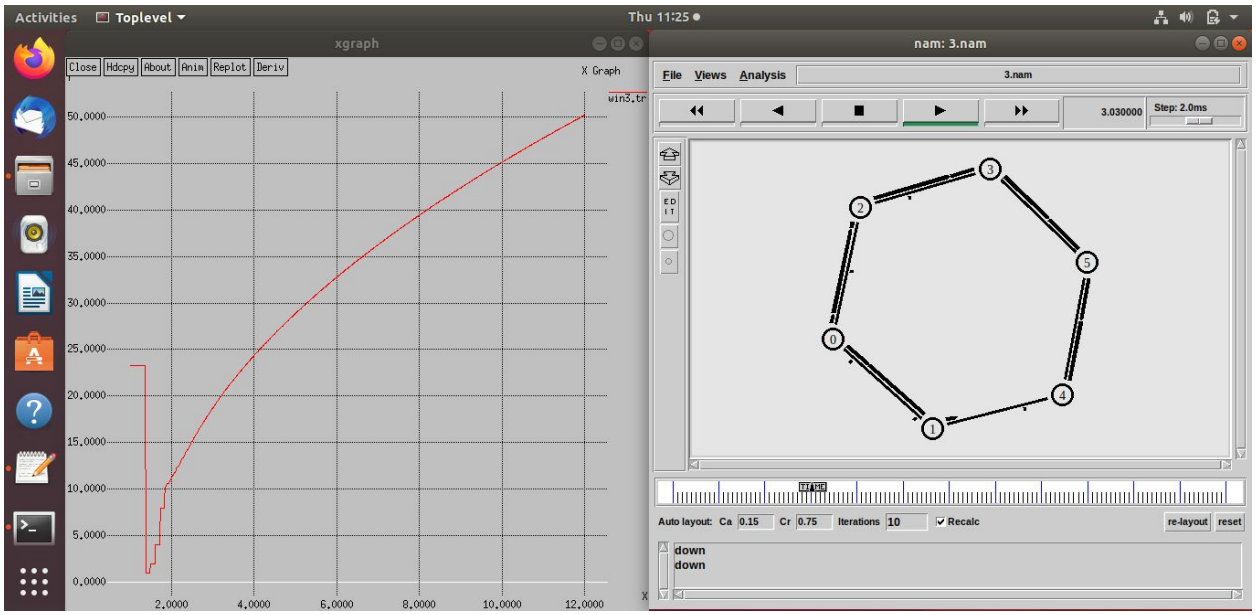
Code:

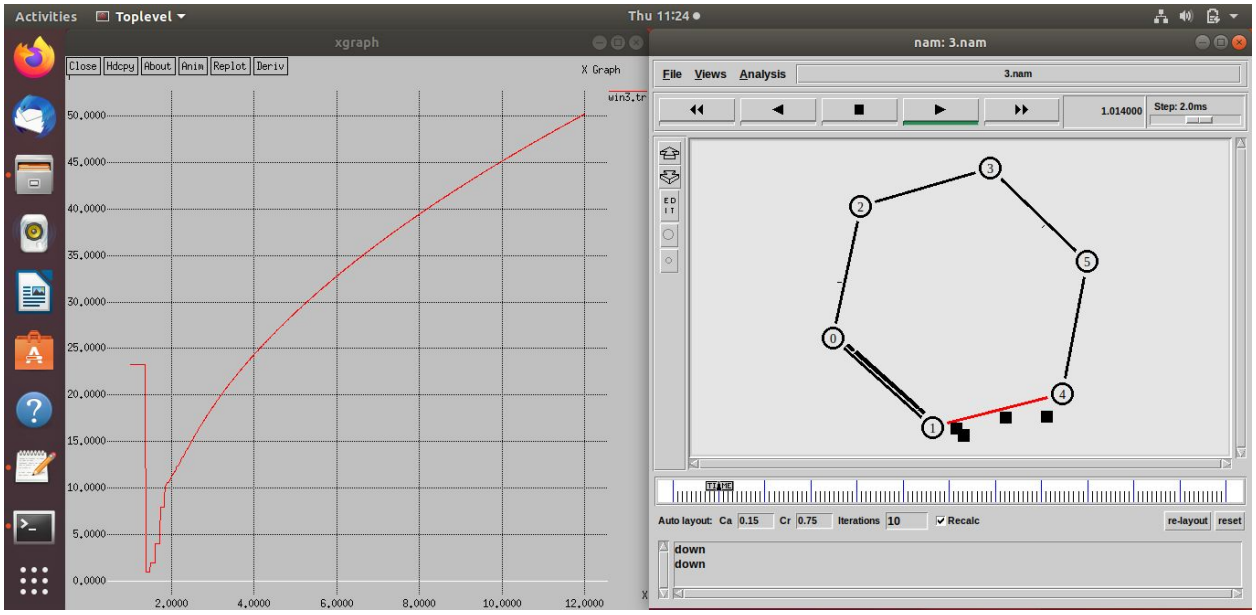
```
set ns [new Simulator]
set tf [open 3.tr w]
$ns trace-all $tf
set nf [open 3.nam w]
$ns namtrace-all $nf
set cwind [open win3.tr w]
$ns rtproto DV
set n0 [$ns node]
set n1 [$ns node]
set n2 [$ns node]
set n3 [$ns node]
set n4 [$ns node]
set n5 [$ns node]
$ns duplex-link $n0 $n1 1Mb 10ms DropTail
$ns duplex-link $n1 $n4 1Mb 10ms DropTail
$ns duplex-link $n4 $n5 1Mb 10ms DropTail
$ns duplex-link $n5 $n3 1Mb 10ms DropTail
$ns duplex-link $n3 $n2 1Mb 10ms DropTail
$ns duplex-link $n2 $n0 1Mb 10ms DropTail
$ns queue-limit $n1 $n4 10
$ns queue-limit $n2 $n3 10
set tcp0 [new Agent/TCP]
$ns attach-agent $n0 $tcp0
set sink0 [new Agent/TCPSink]
$ns attach-agent $n4 $sink0
$ns connect $tcp0 $sink0
set ftp0 [new Application/FTP]
$ftp0 attach-agent $tcp0
$tcp0 set fid_ 1
$ns rtmodel-at 1.0 down $n1 $n4
$ns rtmodel-at 3.0 up $n1 $n4
```

```
$ns at 0.1 "$ftp0 start"
proc plotWindow {tcpSource file} {
global ns
set time 0.01
set now [$ns now]
set cwnd [$tcpSource set cwnd_]
puts $file "$now $cwnd"
$ns at [expr $now+$time] "plotWindow $tcpSource $file" }
$ns at 1.0 "plotWindow $tcp0 $cwind"
proc finish {} {
global ns tf nf cwind
$ns flush-trace
close $tf
close $nf
puts "running nam..."
exec xgraph win3.tr &
exec nam 3.nam &
exit 0
}
$ns at 12.0 "finish"
$ns run
```

Output:


Activities
Toplevel
Thu 11:24
xgraph
Close
Hdcpy
About
Anim
Replot
Deriv
X Graph
win3.tr
50.0000
45.0000
40.0000
35.0000
30.0000
25.0000
20.0000
15.0000
10.0000
5.0000
0.0000
2.0000
4.0000
6.0000
8.0000
10.0000
12.0000
nam: 3.nam
File
Views
Analysis
3.nam
2.028000
Step: 2.0ms
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout
reset
down
down


Activities
Toplevel
Thu 11:25
xgraph
Close
Hdcpy
About
Anim
Replot
Deriv
X Graph
win3.tr
50.0000
45.0000
40.0000
35.0000
30.0000
25.0000
20.0000
15.0000
10.0000
5.0000
0.0000
2.0000
4.0000
6.0000
8.0000
10.0000
12.0000
nam: 3.nam
File
Views
Analysis
3.nam
3.030000
Step: 2.0ms
Auto layout: Ca 0.15 Cr 0.75 Iterations 10 Recalc
re-layout
reset
down
down



Client and Server

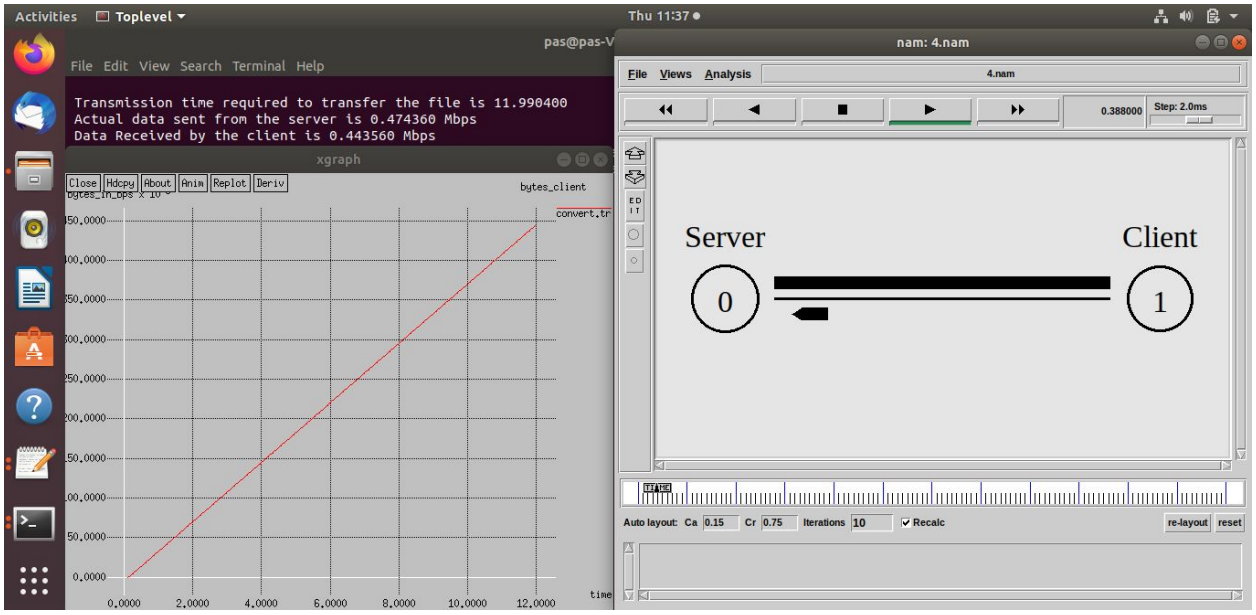
Question: Consider a client and a server. The server is running an FTP application over TCP. The client sends a request to download a file of size 10 MB from the server. Write a TCL script to simulate this scenario. Let node n0 be the server and node n1 be the client. TCP packet size is 1500 Bytes.

Code:

```
set ns [new Simulator]
set tf [open 4.tr w]
$ns trace-all $tf
set nf [open 4.nam w]
$ns namtrace-all $nf
$ns color 1 Blue
set n0 [$ns node]
set n1 [$ns node]
$n0 label "Server"
$n1 label "Client"
$ns duplex-link $n0 $n1 0.3Mb 10ms DropTail
set tcp [new Agent/TCP]
$ns attach-agent $n0 $tcp
$tcp set packetSize_ 1500
set sink [new Agent/TCPSink]
$ns attach-agent $n1 $sink
$ns connect $tcp $sink
```

```
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 0.1 "$ftp start"
$ns at 12.0 "finish"
proc finish {} {
    global ns tf nf cwind
    $ns flush-trace
    close $tf
    close $nf
    exec nam 4.nam &
    exec awk -f 4transfer.awk 4.tr &
    exec awk -f 4convert.awk 4.tr > convert.tr &
    exec xgraph convert.tr -geometry 800*400 -t "bytes_client" -x "time" -y "bytes_in_bps" &
    exit 0
}
$ns run
```

Output:



Multicasting

Question: Demonstrate the working of multicast routing protocol. Assume your own parameters for bandwidth and delay.

Code:

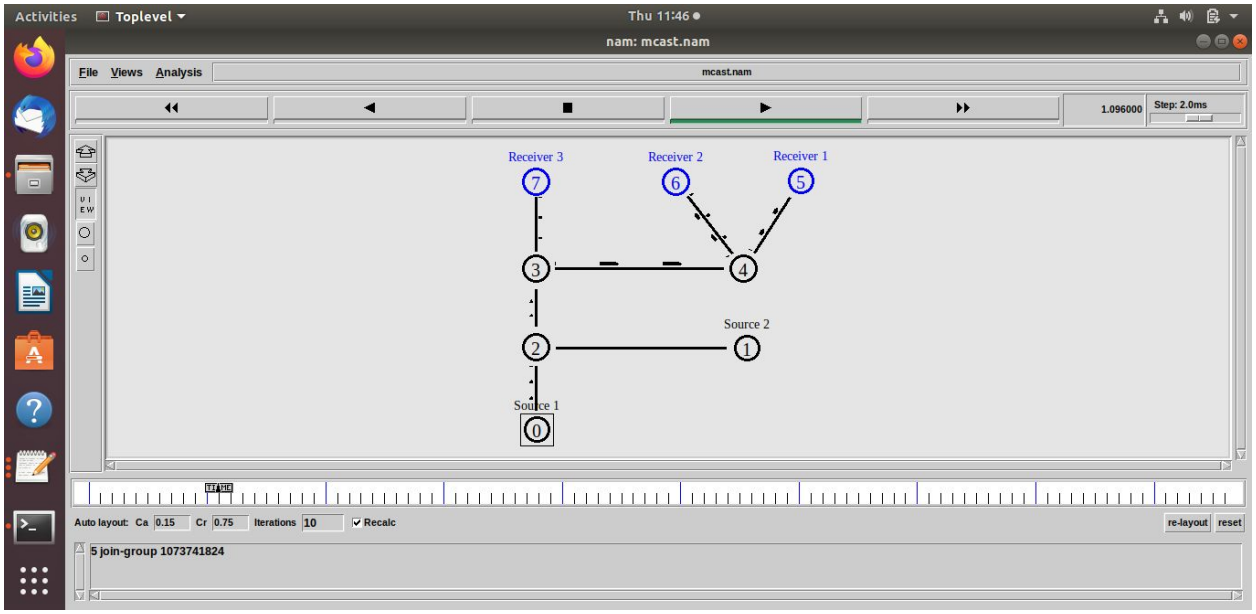
```
#Create an event scheduler wit multicast turned on
set ns [new Simulator -multicast on]
#$ns multicast

#Turn on Tracing
set tf [open mcast.tr w]
$ns trace-all $tf

# Turn on nam Tracing
set fd [open mcast.nam w]
$ns namtrace-all $fd

# Create nodes
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]

# Create links
$ns duplex-link $n0 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n1 $n2 1.5Mb 10ms DropTail
$ns duplex-link $n2 $n3 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n4 1.5Mb 10ms DropTail
$ns duplex-link $n3 $n7 1.5Mb 10ms DropTail
$ns duplex-link $n4 $n5 1.5Mb 10ms DropTail
$ns duplex-link $n4 $n6 1.5Mb 10ms DropTail

# Routing protocol: say distance vector
#Protocols: CtrMcast, DM, ST, BST
set mproto DM
set mrthandle [$ns mrtproto $mproto {}]

# Allocate group addresses
set group1 [Node allocaddr]
set group2 [Node allocaddr]

# UDP Transport agent for the traffic source
set udp0 [new Agent/UDP]
$ns attach-agent $n0 $udp0
```

```
$udp0 set dst_addr_ $group1
$udp0 set dst_port_ 0
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp0

# Transport agent for the traffic source
set udp1 [new Agent/UDP]
$ns attach-agent $n1 $udp1
$udp1 set dst_addr_ $group2
$udp1 set dst_port_ 0
set cbr2 [new Application/Traffic/CBR]
$cbr2 attach-agent $udp1

# Create receiver
set rcvr1 [new Agent/Null]
$ns attach-agent $n5 $rcvr1
$ns at 1.0 "$n5 join-group $rcvr1 $group1"

set rcvr2 [new Agent/Null]
$ns attach-agent $n6 $rcvr2
$ns at 1.5 "$n6 join-group $rcvr2 $group1"

set rcvr3 [new Agent/Null]
$ns attach-agent $n7 $rcvr3
$ns at 2.0 "$n7 join-group $rcvr3 $group1"

set rcvr4 [new Agent/Null]
$ns attach-agent $n5 $rcvr1
$ns at 2.5 "$n5 join-group $rcvr4 $group2"
set rcvr5 [new Agent/Null]
$ns attach-agent $n6 $rcvr2
$ns at 3.0 "$n6 join-group $rcvr5 $group2"
set rcvr6 [new Agent/Null]
$ns attach-agent $n7 $rcvr3
$ns at 3.5 "$n7 join-group $rcvr6 $group2"

$ns at 4.0 "$n5 leave-group $rcvr1 $group1"
$ns at 4.5 "$n6 leave-group $rcvr2 $group1"
$ns at 5.0 "$n7 leave-group $rcvr3 $group1"

$ns at 5.5 "$n5 leave-group $rcvr4 $group2"
$ns at 6.0 "$n6 leave-group $rcvr5 $group2"
$ns at 6.5 "$n7 leave-group $rcvr6 $group2"
```

```
# Schedule events
$ns at 0.5 "$cbr1 start"
$ns at 9.5 "$cbr1 stop"

$ns at 0.5 "$cbr2 start"
$ns at 9.5 "$cbr2 stop"

$ns at 10.0 "finish"

proc finish {} {
        global ns tf fd
        $ns flush-trace
        close $tf
        close $fd
        exec nam mcast.nam &
        exit 0
}


# For nam

# Group 0 source
#$udp0 set fid_ 1
#$n0 color red
$n0 label "Source 1"

# Group 1 source
#$udp1 set fid_ 2
#$n1 color green
$n1 label "Source 2"

#Colors for packets from two mcast groups
$ns color 1 red
$ns color 2 green


$n5 label "Receiver 1"
$n5 color blue
$n6 label "Receiver 2"
$n6 color blue
$n7 label "Receiver 3"
$n7 color blue
```

$ns run

Output:



2-Node wireless network

Question: Set up a 2-node wireless network. Analyze TCP performance for this scenario with DSDV as routing protocol.

Code:

001.tcl

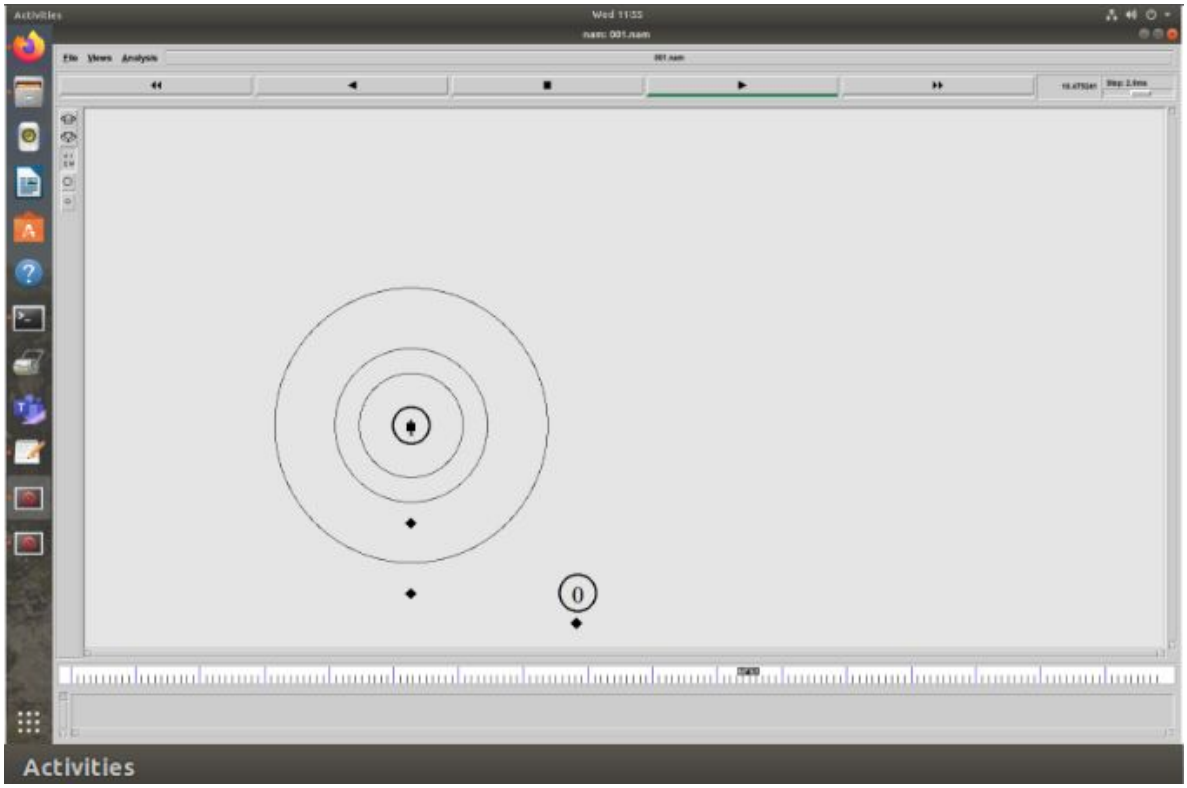
#Setting the Default Parameters

Setting the Default Parameters # ## set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 500

```
set val(y) 500
set val(ifqlen) 50
set val(nn) 2
set val(stop) 20.0
set val(rp) DSDV


set ns_ [new Simulator]

set tracefd [open 001.tr w]
$ns_ trace-all $tracefd

set namtrace [open 001.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop [new $val(prop)]

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

#Node Configuration

 $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) \
                                -channelType $val(chan) \
                                -topoInstance $topo \
                                -agentTrace ON \
                                -routerTrace ON \
                                -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
 set node_($i) [$ns_ node]
 $node_($i) random-motion 0
 }

#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
```

```
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design

$ns_ at 1.1 "$node_(0) setdest 310.0 10.0 20.0"
$ns_ at 1.1 "$node_(1) setdest 10.0 310.0 20.0"
#Generating Traffic

 set tcp0 [new Agent/TCP]
 set sink0 [new Agent/TCPSink]
 $ns_ attach-agent $node_(0) $tcp0
 $ns_ attach-agent $node_(1) $sink0
 $ns_ connect $tcp0 $sink0
 set ftp0 [new Application/FTP]
 $ftp0 attach-agent $tcp0
 $ns_ at 1.0 "$ftp0 start"
 $ns_ at 18.0 "$ftp0 stop"

#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset";
}
$ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

puts "Starting Simulation..."

$ns_ run
```

Output:



3-Node wireless network

Question: Set up 3-node wireless network with node N1 between N0 and N2. As the nodes N0 and N2 moves towards each other they exchange packets. As they move out of each other's range they drop some packets. Analyze TCP performance for this scenario with AODV, DSDV and DSR as routing protocols.

Code:

002.tcl

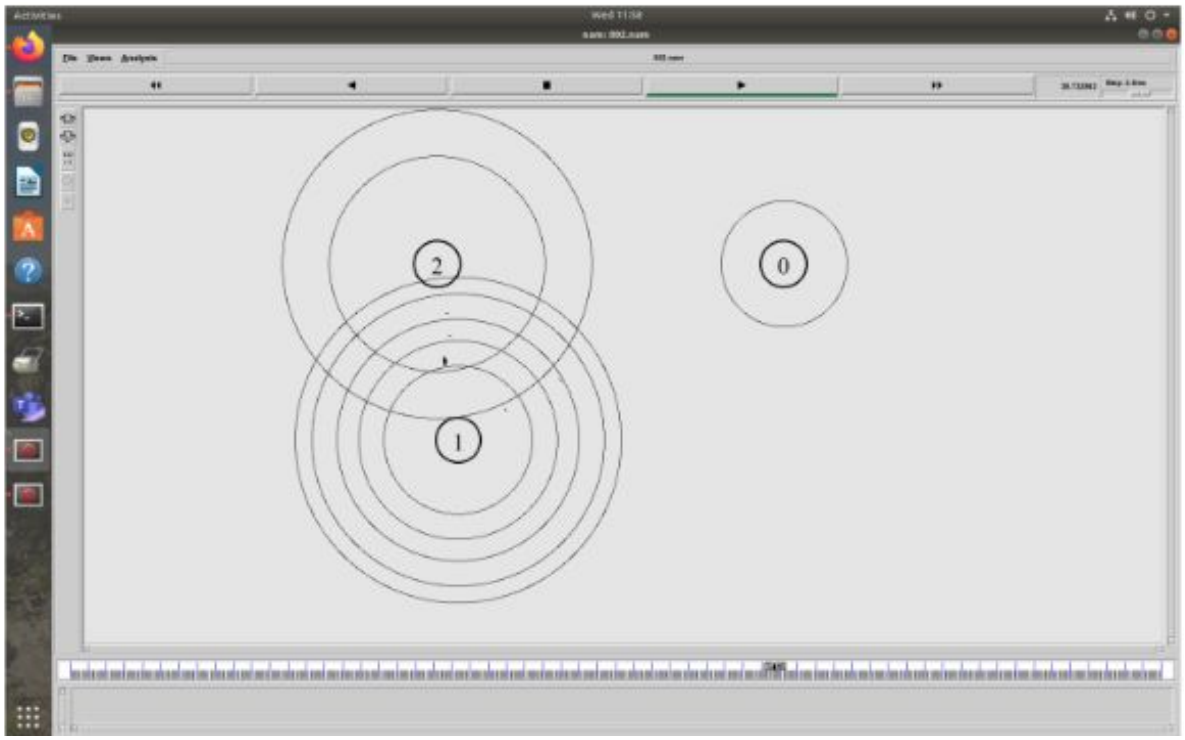
```
#Setting the Default Parameters
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
 #set val(ifq) CMUPriQueue
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 500
```

```
set val(y) 400
set val(ifqlen) 50
set val(nn) 3
set val(stop) 60.0
 set val(rp) AODV


set ns_ [new Simulator]

set tracefd [open 002.tr w]
$ns_ trace-all $tracefd

set namtrace [open 002.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop [new $val(prop)]

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

#Node Configuration

 $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) \
 -channelType $val(chan) \
                              -topoInstance $topo \
                              -agentTrace ON \
                              -routerTrace ON \
                              -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
 set node_($i) [$ns_ node]
 $node_($i) random-motion 0
}

#Initial Positions of Nodes
$node_(0) set x_ 5.0
$node_(0) set y_ 5.0
```

```
$node_(0) set z_ 0.0

$node_(1) set x_ 490.0
$node_(1) set y_ 285.0
$node_(1) set z_ 0.0

$node_(2) set x_ 150.0
$node_(2) set y_ 240.0
$node_(2) set z_ 0.0


for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design

$ns_ at 0.0 "$node_(0) setdest 450.0 285.0 30.0"
$ns_ at 0.0 "$node_(1) setdest 200.0 285.0 30.0"
$ns_ at 0.0 "$node_(2) setdest 1.0 285.0 30.0"


$ns_ at 25.0 "$node_(0) setdest 300.0 285.0 10.0"
$ns_ at 25.0 "$node_(2) setdest 100.0 285.0 10.0"


$ns_ at 40.0 "$node_(0) setdest 490.0 285.0 5.0"
$ns_ at 40.0 "$node_(2) setdest 1.0 285.0 5.0"


#Generating Traffic
 set tcp0 [new Agent/TCP]
 set sink0 [new Agent/TCPSink]
 $ns_ attach-agent $node_(0) $tcp0
 $ns_ attach-agent $node_(2) $sink0
 $ns_ connect $tcp0 $sink0
 set ftp0 [new Application/FTP]
 $ftp0 attach-agent $tcp0
 $ns_ at 10.0 "$ftp0 start"


#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset";
}
 $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"
```

```
puts "Starting Simulation..."
 $ns_ run
```

Output:



6-Node wireless network

Question: Set up a 6-node wireless network; analyze TCP performance when nodes are static and mobile.

Code:

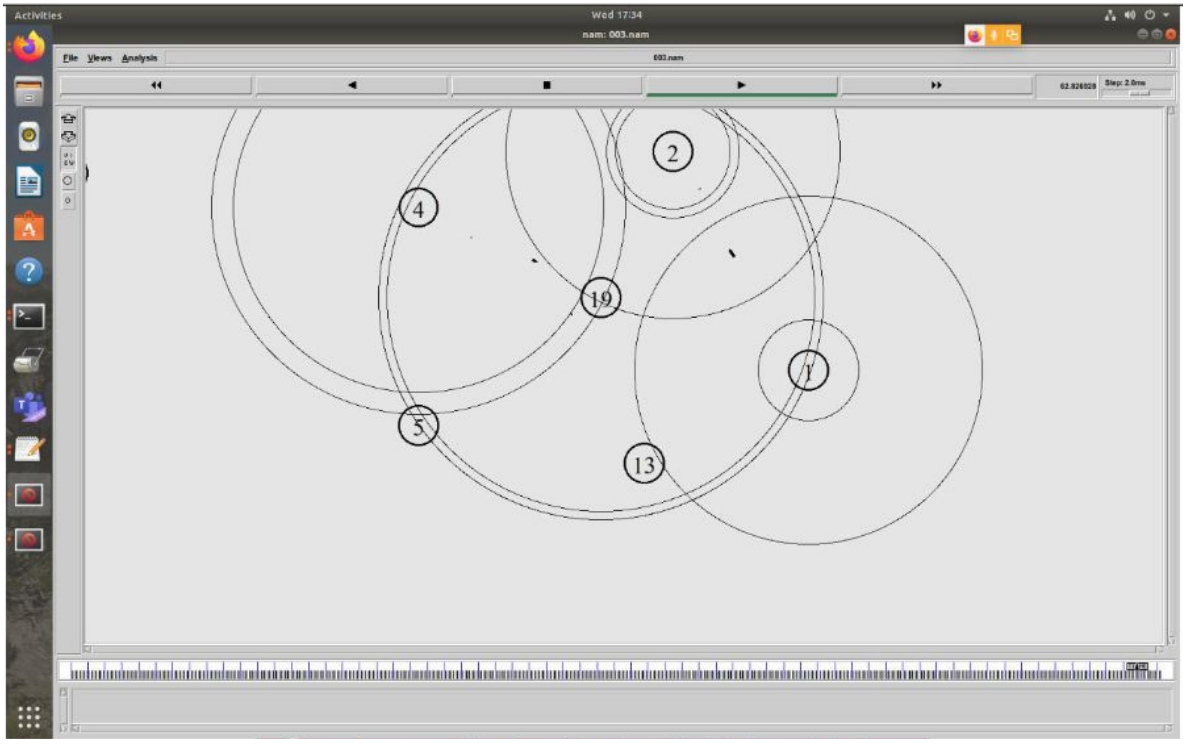
003.tcl

```
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 500
set val(y) 500
set val(ifqlen) 50
set val(nn) 25
set val(stop) 100.0
```

```
 set val(rp) AODV
 #set val(sc) "mob-25-50"
set val(cp) "tcp-25-8"

set ns_ [new Simulator]

set tracefd [open 003.tr w]
$ns_ trace-all $tracefd

set namtrace [open 003.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
set prop [new $val(prop)]

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

set god_ [create-god $val(nn)]

 #Node Configuration

 $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) \
 -channelType $val(chan) \
                              -topoInstance $topo \
                              -agentTrace ON \
                              -routerTrace ON \
                              -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
 set node_($i) [$ns_ node]
 $node_($i) random-motion 0
 }


 for {set i 0} {$i < $val(nn) } { incr i } {
 set xx [expr rand()*500]
 set yy [expr rand()*400]
 $node_($i) set X_ $xx
 $node_($i) set Y_ $yy
```

```
}


#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}


#puts "Loading scenario file..."
#source $val(sc)
puts "Loading connection file..."
source $val(cp)
#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset";
 }
 $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

puts "Starting Simulation..."
 $ns_ run
```

Output:



Scenario of 6-Node wireless network

Question: Write a TCL script to simulate the following scenario. Consider six nodes, (as shown in the figure below) moving within a flat topology of 700m x 700m. The initial positions of nodes are: n0 (150, 300), n1 (300, 500), n2(500, 500), n3 (300, 100), n4(500, 100) and n5(650, 300) respectively. A TCP connection is initiated between n0 (source) and n5 (destination) through n3 and n4 i.e., the route is 0- 3-4-5. At time t = 3 seconds, the FTP application runs over it. After time t = 4 seconds, n3 (300,100) moves towards n1 (300, 500) with a speed of 5.0m/sec and after some time the path breaks. The data is then transmitted with a new path via n1 and n2 i.e., the new route is 0-1-2-5. The simulation lasts for 60 secs. In the above said case both the routes have equal cost. Use DSR as the routing protocol and the IEEE 802.11 MAC protocol.

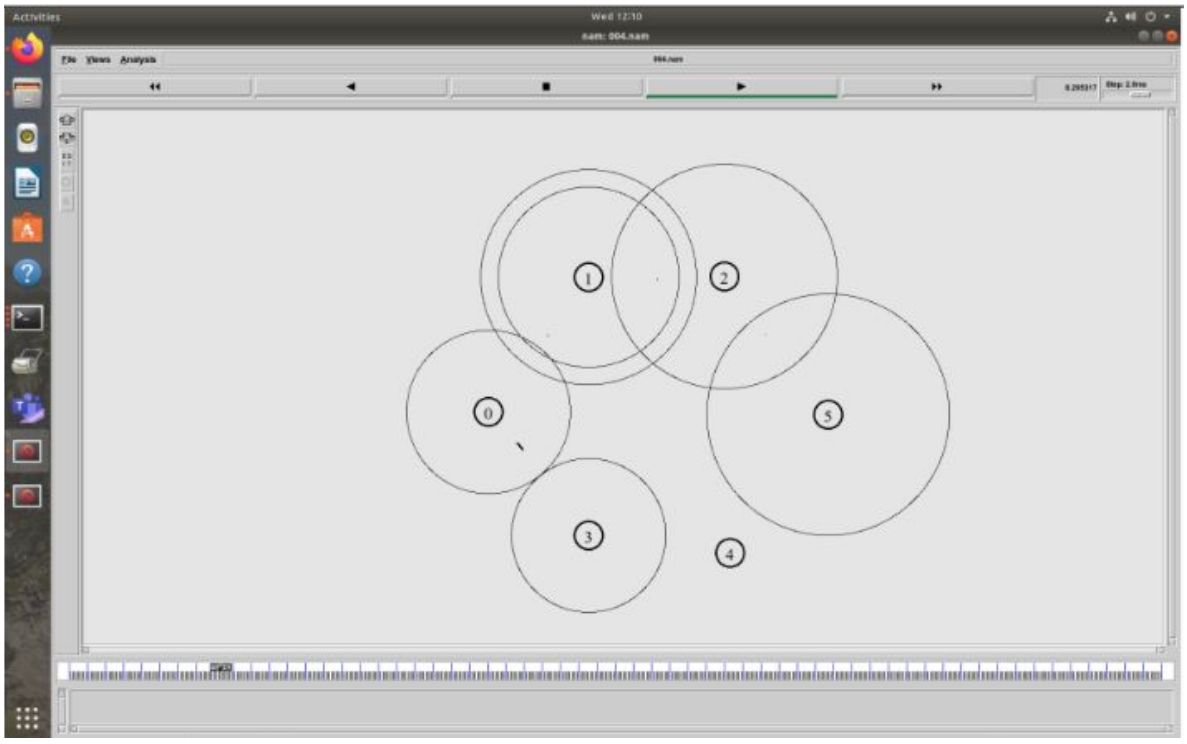
Code:

```
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
#set val(ifq) Queue/DropTail/PriQueue
set val(ifq) CMUPriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 700
set val(y) 700
set val(ifqlen) 50
set val(nn) 6
set val(stop) 60.0
 set val(rp) DSR


set ns_ [new Simulator]

set tracefd [open 004.tr w]
$ns_ trace-all $tracefd

set namtrace [open 004.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)

set prop [new $val(prop)]

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)
```

```
set god_ [create-god $val(nn)]

 #Node Configuration

 $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) \
 -channelType $val(chan) \
                          -topoInstance $topo \
                          -agentTrace ON \
                          -routerTrace ON \
                          -macTrace ON

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
 set node_($i) [$ns_ node]
 $node_($i) random-motion 0
 }
#Initial Positions of Nodes



$node_(0) set X_ 150.0
$node_(0) set Y_ 300.0
$node_(0) set Z_ 0.0
$node_(1) set X_ 300.0
$node_(1) set Y_ 500.0
$node_(1) set Z_ 0.0

$node_(2) set X_ 500.0
$node_(2) set Y_ 500.0
$node_(2) set Z_ 0.0

$node_(3) set X_ 300.0
$node_(3) set Y_ 100.0
$node_(3) set Z_ 0.0

$node_(4) set X_ 500.0
$node_(4) set Y_ 100.0
$node_(4) set Z_ 0.0

$node_(5) set X_ 650.0
```

```
$node_(5) set Y_ 300.0
$node_(5) set Z_ 0.0
for {set i 0} {$i < $val(nn)} {incr i} {
        $ns_ initial_node_pos $node_($i) 40
}

#Topology Design
$ns_ at 1.0 "$node_(0) setdest 160.0 300.0 2.0"
$ns_ at 1.0 "$node_(1) setdest 310.0 150.0 2.0"
$ns_ at 1.0 "$node_(2) setdest 490.0 490.0 2.0"
$ns_ at 1.0 "$node_(3) setdest 300.0 120.0 2.0"
$ns_ at 1.0 "$node_(4) setdest 510.0 90.0 2.0"
$ns_ at 1.0 "$node_(5) setdest 640.0 290.0 2.0"

$ns_ at 4.0 "$node_(3) setdest 300.0 500.0 5.0"

#Generating Traffic
 set tcp0 [new Agent/TCP]
 set sink0 [new Agent/TCPSink]
 $ns_ attach-agent $node_(0) $tcp0
 $ns_ attach-agent $node_(5) $sink0
 $ns_ connect $tcp0 $sink0
 set ftp0 [new Application/FTP]
 $ftp0 attach-agent $tcp0
 $ns_ at 5.0 "$ftp0 start"
 $ns_ at 60.0 "$ftp0 stop"


#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset";
 }
$ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"
puts "Starting Simulation..."
 $ns_ run
```

Output:



Wireless network with mobile Nodes

Question: Set up a wireless network with mobile nodes, induce 1 to 10% error to the network using a uniform error model. Plot the congestion window for TCP connections. Write your observation on TCP performance as error increases in the network.

Code:

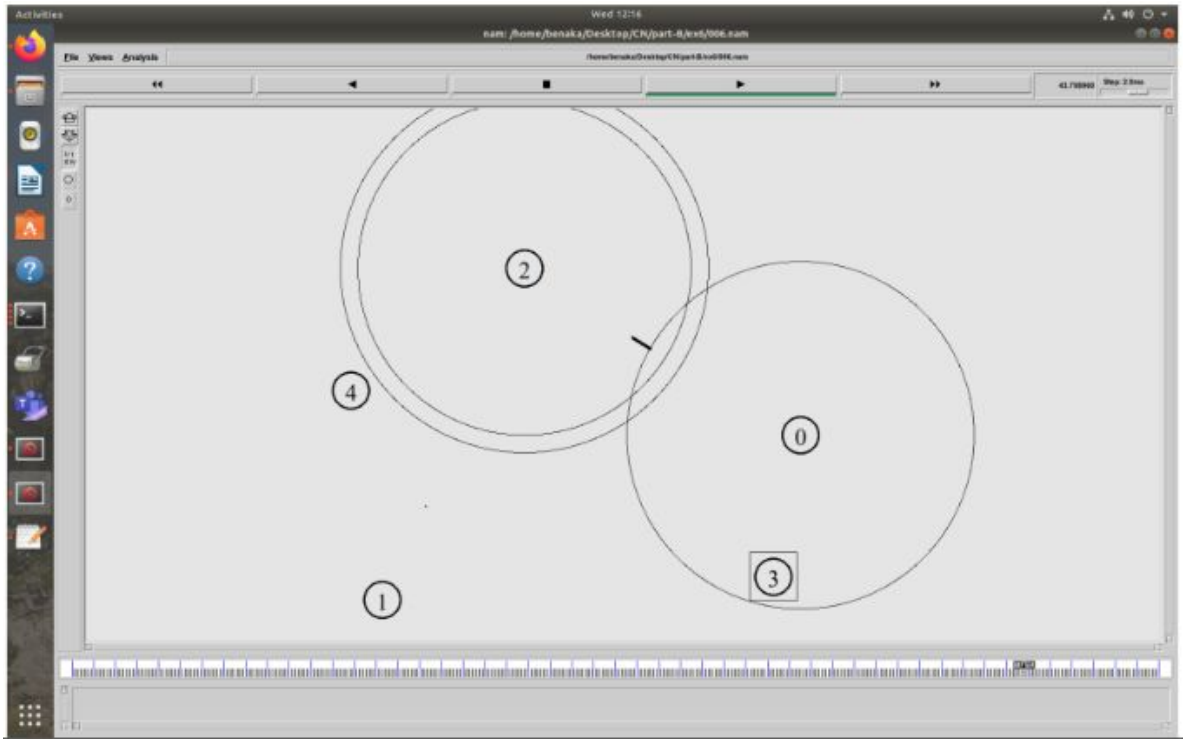
006.tcl.

```
set val(chan) Channel/WirelessChannel
set val(prop) Propagation/TwoRayGround
set val(netif) Phy/WirelessPhy
set val(mac) Mac/802_11
set val(ifq) Queue/DropTail/PriQueue
set val(ll) LL
set val(ant) Antenna/OmniAntenna
set val(x) 500
set val(y) 500
set val(ifqlen) 50
set val(nn) 5
set val(stop) 50.0
 set val(rp) AODV
```

```
set ns_ [new Simulator]

set tracefd [open 006.tr w]
$ns_ trace-all $tracefd

set namtrace [open 006.nam w]
$ns_ namtrace-all-wireless $namtrace $val(x) $val(y)
set prop [new $val(prop)]

set topo [new Topography]
$topo load_flatgrid $val(x) $val(y)

create-god $val(nn)

 #Node Configuration

 $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) \
 -channelType $val(chan) \
                             -topoInstance $topo \
                             -agentTrace ON \
                             -routerTrace ON \
                             -macTrace ON \
 -IncomingErrProc "uniformErr" \
                             -OutgoingErrProc "uniformErr"

proc uniformErr {} {
set err [new ErrorModel]
$err unit pkt
$err set rate_ 0.01
return $err
}

#Creating Nodes
for {set i 0} {$i < $val(nn) } {incr i} {
 set node_($i) [$ns_ node]
 $node_($i) random-motion 0
 }
#Initial Positions of Nodes

for {set i 0} {$i < $val(nn)} {incr i} {
```

```
        $ns_ initial_node_pos $node_($i) 40
 }
#Topology Design

$ns_ at 1.0 "$node_(0) setdest 10.0 10.0 50.0"
$ns_ at 1.0 "$node_(1) setdest 10.0 100.0 50.0"
$ns_ at 1.0 "$node_(4) setdest 50.0 50.0 50.0"
$ns_ at 1.0 "$node_(2) setdest 100.0 100.0 50.0"
$ns_ at 1.0 "$node_(3) setdest 100.0 10.0 50.0"

#Generating Traffic
 set tcp0 [new Agent/TCP]
 set sink0 [new Agent/TCPSink]
 $ns_ attach-agent $node_(0) $tcp0
 $ns_ attach-agent $node_(2) $sink0
 $ns_ connect $tcp0 $sink0
 set ftp0 [new Application/FTP]
 $ftp0 attach-agent $tcp0
 $ns_ at 1.0 "$ftp0 start"
 $ns_ at 50.0 "$ftp0 stop"

 set tcp1 [new Agent/TCP]
 set sink1 [new Agent/TCPSink]
 $ns_ attach-agent $node_(1) $tcp1
 $ns_ attach-agent $node_(2) $sink1
 $ns_ connect $tcp1 $sink1
 set ftp1 [new Application/FTP]
 $ftp1 attach-agent $tcp1
 $ns_ at 1.0 "$ftp1 start"
 $ns_ at 50.0 "$ftp1 stop"

#Simulation Termination

for {set i 0} {$i < $val(nn) } {incr i} {
 $ns_ at $val(stop) "$node_($i) reset";
  }
 $ns_ at $val(stop) "puts \"NS EXITING...\" ; $ns_ halt"

puts "Starting Simulation..."
 $ns_ run
```

Output:



PART-B

Question: Study of network IP Experiments

i. Classification of IP address ii. Subnetting iii. Super netting

Answer:

a. Demonstration of Class C IP address.

Wired network of 3 systems, all 3 systems have the same network ids. The Network id for this network is 192.168.0 and the host id is 1, 2 and 3. These addresses are class C IP addresses, in which the first 3 octets represent network Id (netid) and the 4th octet represents host Id. Out of the last octet (8-bit value) 256 IP addresses can be generated out of which the first host address 0 is not assigned to any host and IP address with 0 in 4th octet place is used to refer to network Id and last addresses are 255 is used as broadcast Id. So effectively 254 Host addresses are used in Class C addresses.

PC0 command Prompt

C:\>ping 192.168.0.2
Pinging 192.168.0.2 with 32 bytes of data:
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time=1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Reply from 192.168.0.2: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.0.2:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 1ms, Average = 0ms

C:\>ping 192.168.0.3
Pinging 192.168.0.3 with 32 bytes of data:
Reply from 192.168.0.3: bytes=32 time=1ms TTL=128
Reply from 192.168.0.3: bytes=32 time=3ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Reply from 192.168.0.3: bytes=32 time<1ms TTL=128
Ping statistics for 192.168.0.3:
Packets: Sent = 4, Received = 4, Lost = 0 (0% loss),
Approximate round trip times in milli-seconds:
Minimum = 0ms, Maximum = 3ms, Average = 1ms
Similarly

1. Ping from PC1 to PC0 and PC2
2. Ping from PC2 to PC0 and PC1

b. Subnetting

Subnetting allows you to create multiple logical networks that exist within a single Class A, B, or C network. If you do not subnet, you are only able to use one network from your Class A, B, or C network, which is unrealistic. Each data link on a network must have a unique network ID, with every node on that link being a
member of the same network. If you break a major network (Class A, B, or C) into smaller subnetworks, it allows you to create a network of interconnecting subnetworks. Each data link on this network would then have a unique network/subnetwork ID. Any device, or gateway, that connects

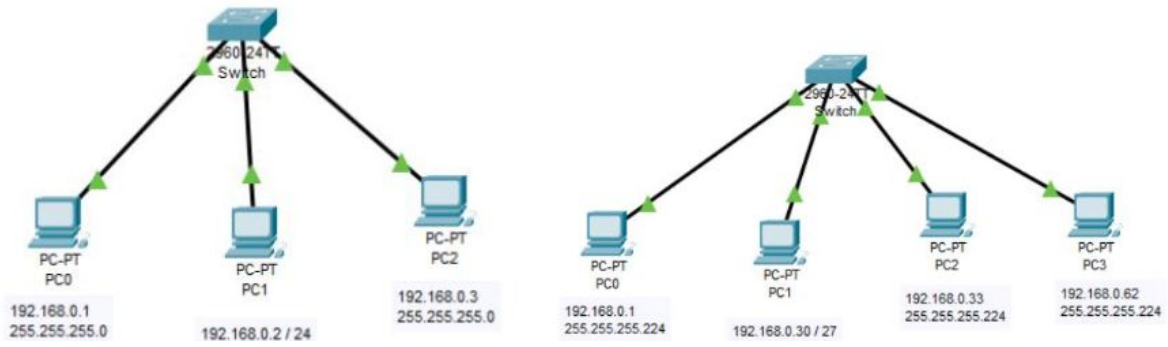
networks/subnetworks has n distinct IP addresses, one for each network / subnetwork that it interconnects. In order to subnet a network, extend the natural mask with some of the bits from the host ID portion of the address in order to create a subnetwork ID. For example, given a Class C network of 204.17.5.0
which has a natural mask of 255.255.255.0, you can create subnets in this manner:

204.17.5.0 - 11001100.00010001.00000101.00000000
255.255.255.224 - 11111111.11111111.11111111.11100000
-------------------------|sub|----

By extending the mask to be 255.255.255.224, you have taken three bits (indicated by "sub") from the original host portion of the address and used them to make subnets. With these three bits, it is possible to create eight subnets. With the remaining five host ID bits, each subnet can have up to 32 host addresses (25), 30 of which can actually be assigned to a device since host ids of all zeros or all ones are not allowed (it is very important to remember this). So, with this in mind, these subnets have been created.
204.17.5.0 255.255.255.224 host address range 1 to 30
204.17.5.32 255.255.255.224 host address range 33 to 62
204.17.5.64 255.255.255.224 host address range 65 to 94
204.17.5.96 255.255.255.224 host address range 97 to 126
204.17.5.128 255.255.255.224 host address range 129 to 158
204.17.5.160 255.255.255.224 host address range 161 to 190
204.17.5.192 255.255.255.224 host address range 193 to 222
204.17.5.224 255.255.255.224 host address range 225 to 254

PC0 and PC1 are in one subnet and PC2 and PC3 are another subnet. PC0 can communicate with PC1 and vice versa but not with PC2 and PC3 Similarly PC2 and PC3 can communicate with each other but cannot communicate PC0 and PC1.

Screenshot:



Question: Configure static and Dynamic Routing Information in the router and test the connectivity between networks.

Answer:

Static Routing

Static Routing is also known as non-adaptive routing which doesn't change the routing table unless the network administrator changes or modify them manually. Static routing does not use complex routing algorithms and It provides high or more security than dynamic routing.Dynamic routing is also known as adaptive routing which changes the routing table according to the change in topology. Dynamic routing uses complex routing algorithms and it does not provide high security like static routing. When the network change(topology) occurs, it sends the message to the router to ensure that changes then the routes are recalculated for sending updated routing information.

On router-3

Router>enable

```
Router#config ter
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int
Router(config)#int fa0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed  state to up
Router(config-if)#inter se2/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to
down Router(config-if)#clock rate 64000
Router(config-if)#exit
Router(config)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to  up
Router(config)#ip route 192.168.3.0 255.255.255.0 192.168.2.2
Router(config)#exit
Router#
%SYS-5-CONFIG_I: Configured from console by console
Router#copy run star
Router#copy run startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

On router-4

```
Router>en
Router#conf t
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#int fa0/0
Router(config-if)#ip address 192.168.2.2 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed  state to up
Router(config-if)#
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#inte se2/0
Router(config-if)#ip address 192.168.2.2 255.255.255.0
```

```
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to  up
Router(config-if)#exit
```

```
Router(config)#ip route 192.168.1.0 255.255.255.0 192.168.2.1
Router#wr
Building configuration...
[OK]
```

Dynamic Routing Demonstration

In Router-0

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed  state to up
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to  up
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 11.0.0.0
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#exit
Router(config)#exit
Router#copy run startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

In Router-1

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
```

%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed state to up
Router(config-if)#exit
Router(config)#interface Serial2/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0

```
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial2/0, changed state to up
Router(config-if)#exit
Router(config)#interface Serial3/0
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial2/0, changed state
to  up
ip address 11.0.0.1 255.0.0.0
Router(config-if)#ip address 11.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to  up
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 11.0.0.0
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config)#exit
Router#copy run start
Router#copy run startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
Router#
```

In Router-2

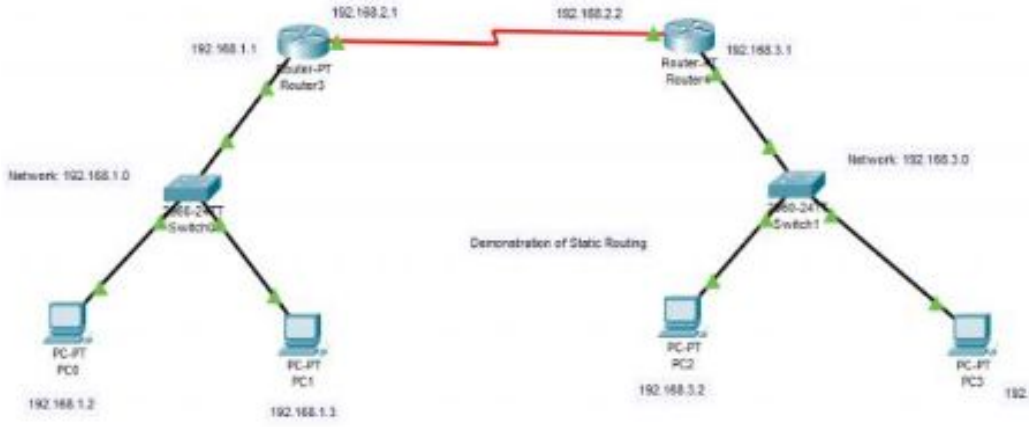
```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.3.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 11.0.0.2 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface Serial3/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface Serial3/0, changed state
to  up%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed  state to up
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
```

```
Router(config-router)#network 11.0.0.0
Router(config-router)#network 192.168.1.0
Router(config-router)#network 192.168.2.0
Router(config-router)#network 192.168.3.0
Router(config-router)#exit
Router(config)#exit
```

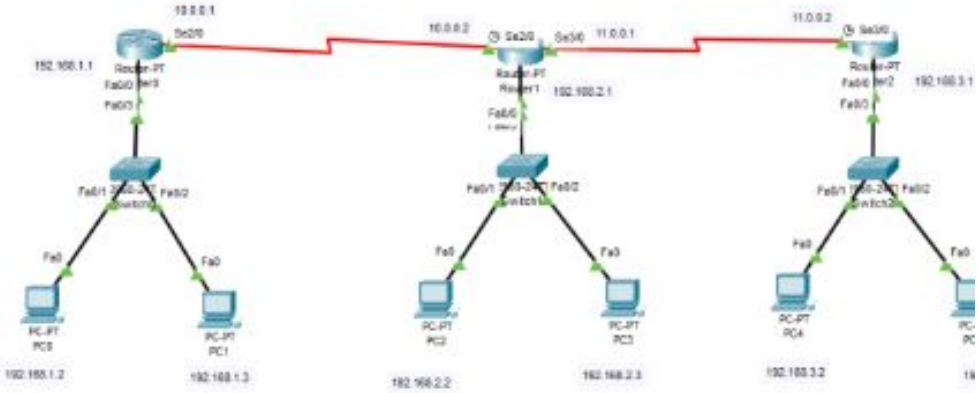
```
Router#copy run start
Destination filename [startup-config]?
Building configuration...
[OK]
```

Screenshot:

a. Static Routing Demonstration



b. Dynamic Routing Demonstration



Question: Configure Network Address Translation (NAT) and test static NAT, dynamic NAT and PAT.

Answer:

a. Static NAT

Static NAT (Network Address Translation) - Static NAT (Network Address Translation) is one-to-one mapping of a private IP address to a public IP address. Static NAT (Network Address Translation) is useful when a network device inside a private network needs to be accessible from internet.

Router-0

```
Router>enable
Router#configure terminal
Router(config)#interface Serial2/0
Router(config-if)#ip address 10.0.0.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface FastEthernet1/0
Router(config-if)#ip address 192.168.1.1 255.255.255.0
Router(config-if)#no shutdown
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 192.168.2.1 255.255.255.0
Router(config-if)#no shutdown
Router(config-if)#exit
```

```
Router(config)#router rip
Router(config-router)#network 192.168.2.0
Router(config-router)#network 10.0.0.0
Router(config-router)#exit
Router(config)#inter se2/0
Router(config-if)#ip nat out
Router(config-if)#inter fa1/0
Router(config-if)#ip nat in
Router(config)#ip nat inside source static 192.168.1.2 10.0.0.3
Router(config)#exit
Router#wr
Building configuration...
[OK]
```

Router-1

```
Router>enable
Router#configure terminal
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 8.8.8.1 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#interface Serial3/0
Router(config-if)#ip address 10.0.0.2 255.0.0.0
Router(config-if)#no shutdown
Router(config-if)#exit
Router(config)#router rip
Router(config-router)#network 10.0.0.0
Router(config-router)#network 8.0.0.0
```

b. Dynamic NAT

Dynamic NAT can be defined as mapping of a private IP address to a public IP address from a group of public IP addresses called a NAT pool. Dynamic NAT establishes a one-to-one mapping between a private IP address to a public IP address. Here the public IP address is taken from the pool of IP addresses configured on the end NAT router. The public to private mapping may vary based on the available public IP address in NAT pool.

Router-0

```
Router>enable
Router#conf t
Router(config)#int se2/0
Router(config-if)#ip nat outside
Router(config)#int fa0/0
Router(config-if)#ip nat inside
Router(config-if)#exit
Router(config)#acc
Router(config)#access-list 1 permit 192.168.2.0 0.0.0.255
```

Router(config)#ip nat pool NAT 10.0.0.5 10.0.0.10 netmask
255.255.255.0 Router(config)#ip nat inside source list 1 pool NAT
Router(config)#exit
Router#show ip nat translations

```
Pro  Inside global     Inside local       Outside local      Outside global
icmp 10.0.0.5:10       192.168.2.3:10     8.8.8.8:10         8.8.8.8:10
icmp 10.0.0.5:11       192.168.2.3:11     8.8.8.8:11         8.8.8.8:11
icmp 10.0.0.5:12       192.168.2.3:12     8.8.8.8:12         8.8.8.8:12
icmp 10.0.0.5:13       192.168.2.3:13     8.8.8.8:13         8.8.8.8:13
---  10.0.0.3          192.168.1.2        ---                ---
tcp 10.0.0.6:1025      192.168.2.2:1025   8.8.8.8:80         8.8.8.8:80
```

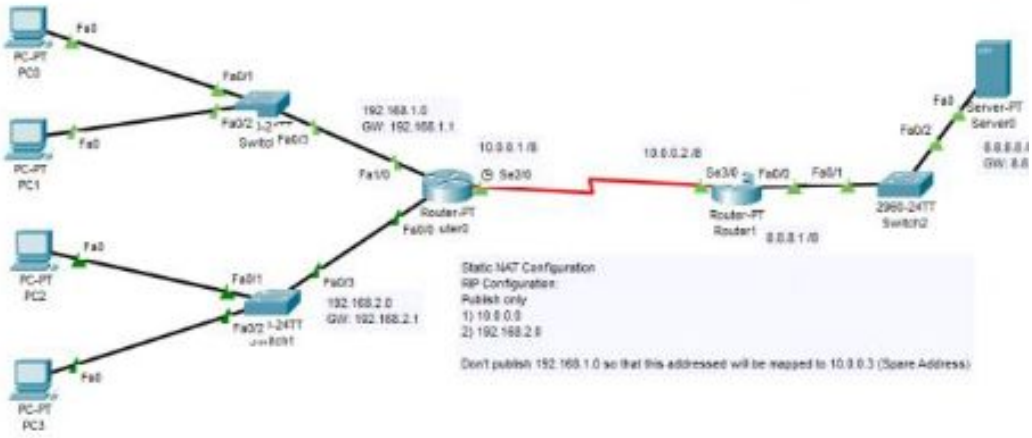
PAT

PAT (Port Address Translation) - Port Address Translation (PAT) is another type of dynamic NAT which can map multiple private IP addresses to a single public IP address by using a technology known as Port Address Translation. Here when a client from inside the network communicates to a host on the internet, the router changes the source port (TCP or UDP) number with another port number. These port mappings are kept in a table. When the router receives from the internet, it will refer to the table which keeps the port mappings and forward the data packet to the original sender.

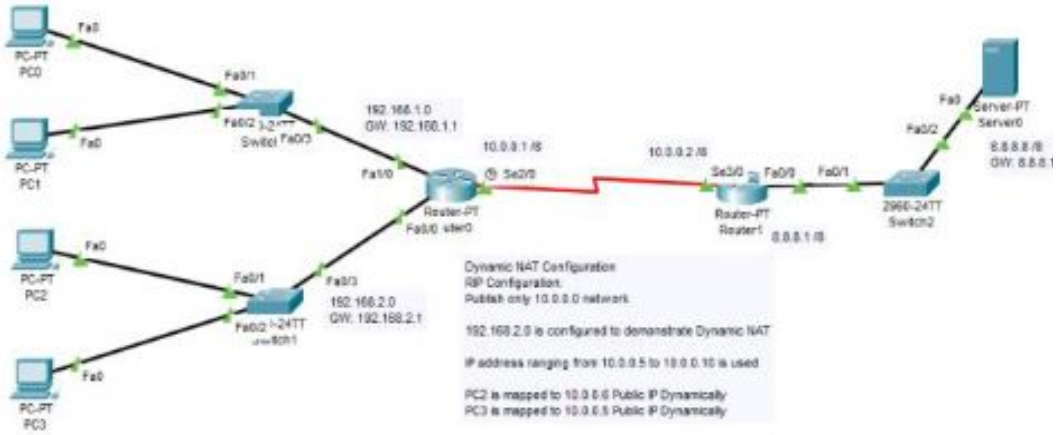
Router - 0

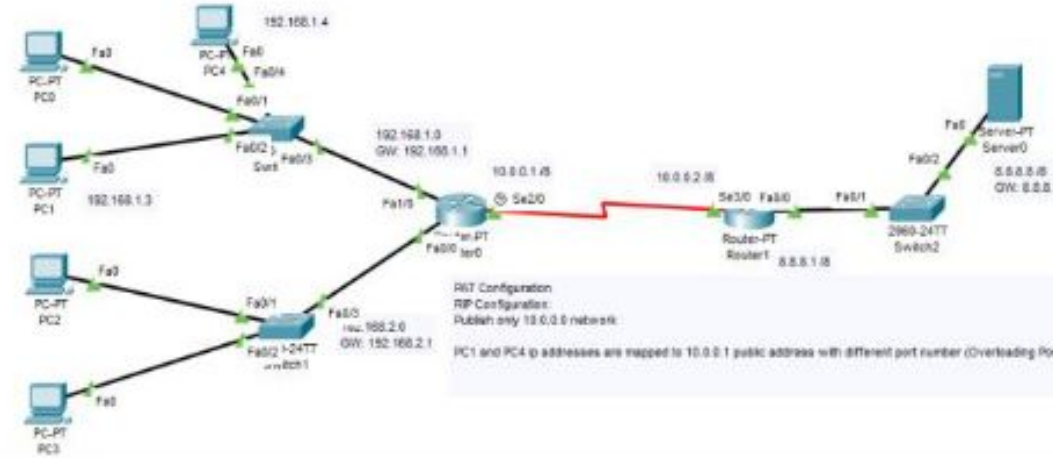
Router#conf t
Router(config)#inte se2/0
Router(config-if)#ip nat outside
Router(config-if)#int fa0/0
Router(config-if)#ip nat inside
Router(config)#access-list 1 permit 192.168.1.0 0.0.0.255
Router(config-if)#ip nat inside source list 1 interface se2/0
overload Router(config)#exit

```
Router#show ip nat translations
Pro  Inside global     Inside local       Outside local      Outside glo
---  10.0.0.3          192.168.1.2        ---                ---
tcp 10.0.0.1:1024      192.168.1.4:1025   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.1:1025      192.168.1.3:1025   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1025      192.168.1.2:1025   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1026      192.168.1.2:1026   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1027      192.168.1.2:1027   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1028      192.168.1.2:1028   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1029      192.168.1.2:1029   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.3:1030      192.168.1.2:1030   8.8.8.8:80         8.8.8.8:80
tcp 10.0.0.6:1025      192.168.2.2:1025   8.8.8.8:80         8.8.8.8:80
```



Ip address 192.168.1.2 is statically mapped to 10.0.0.3 Public IP



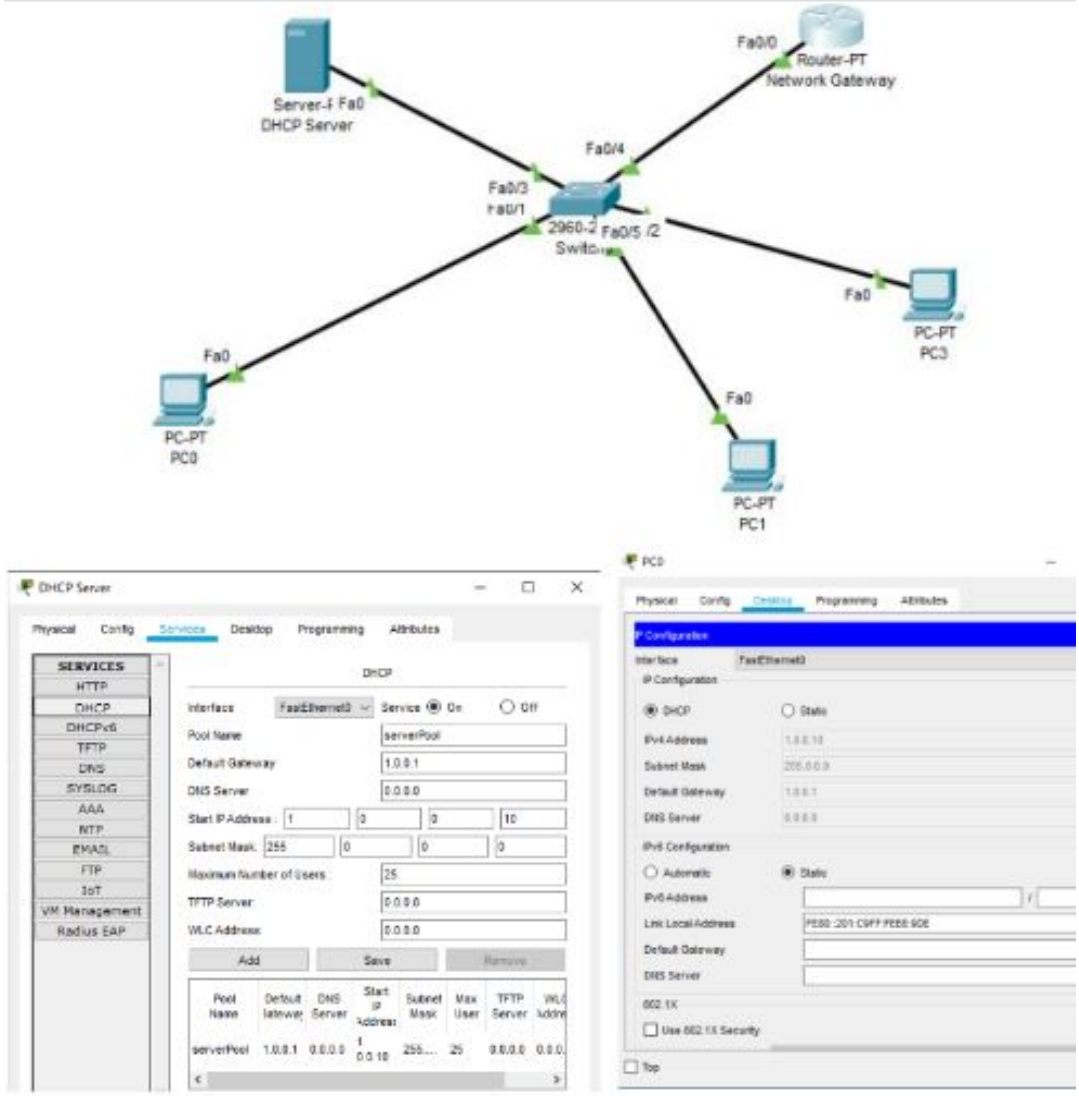


Question: Configure a DHCP server to dynamically assign IP address, subnet mask and default gateway to the hosts in the network.

Answer:
A DHCP Server is a network server that automatically provides and assigns IP addresses, default gateways and other network parameters to client devices. It relies on the standard protocol known as Dynamic Host Configuration Protocol or DHCP to respond to broadcast queries by clients. A DHCP server automatically sends the required network parameters for clients to properly communicate on the network. Without it, the network administrator has to manually set up every client that joins the network, which can be cumbersome, especially in large networks. DHCP servers usually assign each client with a unique dynamic IP address, which changes when the client's lease for that IP address has expired.

```
Router>enable
Router#configure terminal
Enter configuration commands, one per line. End with CNTL/Z.
Router(config)#interface FastEthernet0/0
Router(config-if)#ip address 1.0.0.1 255.0.0.0
Router(config-if)#no shutdown
%LINK-5-CHANGED: Interface FastEthernet0/0, changed state to up
%LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0,
changed  state to up
Router#copy run start
Router#copy run startup-config
Destination filename [startup-config]?
Building configuration...
[OK]
```

Screenshot:



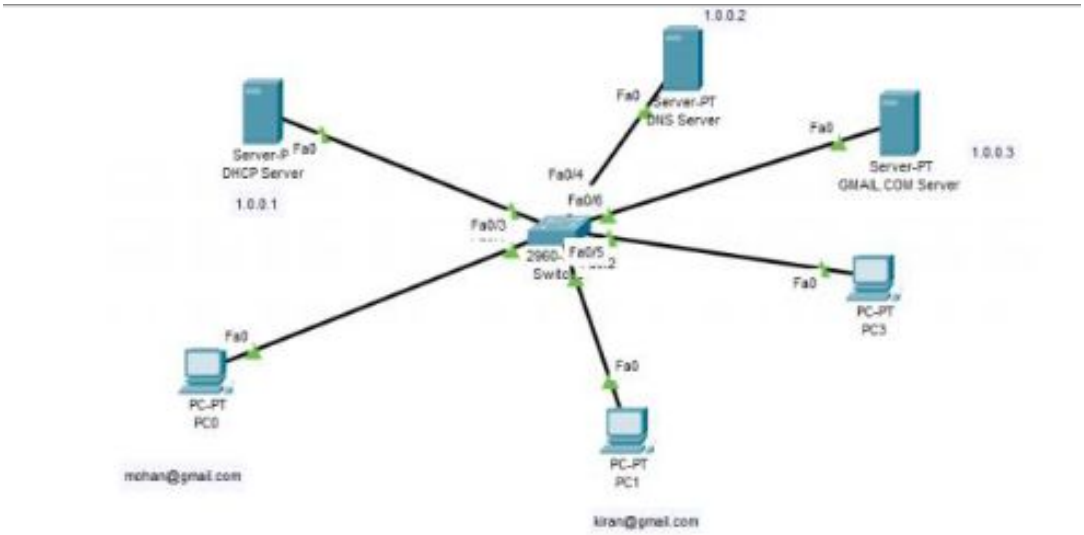
Question: Configure and test DNS and Email server in a network.

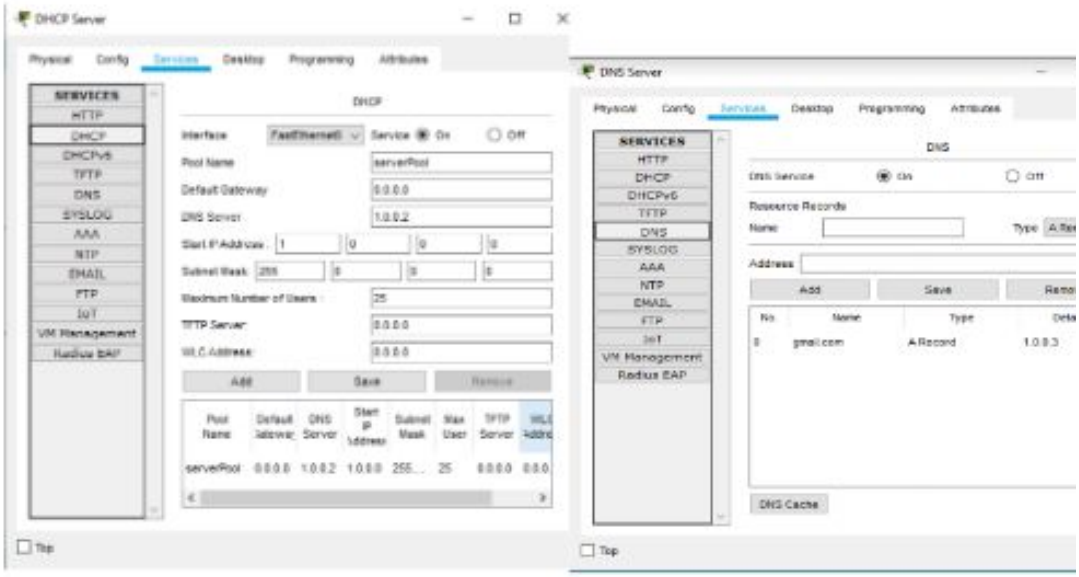
Answer:

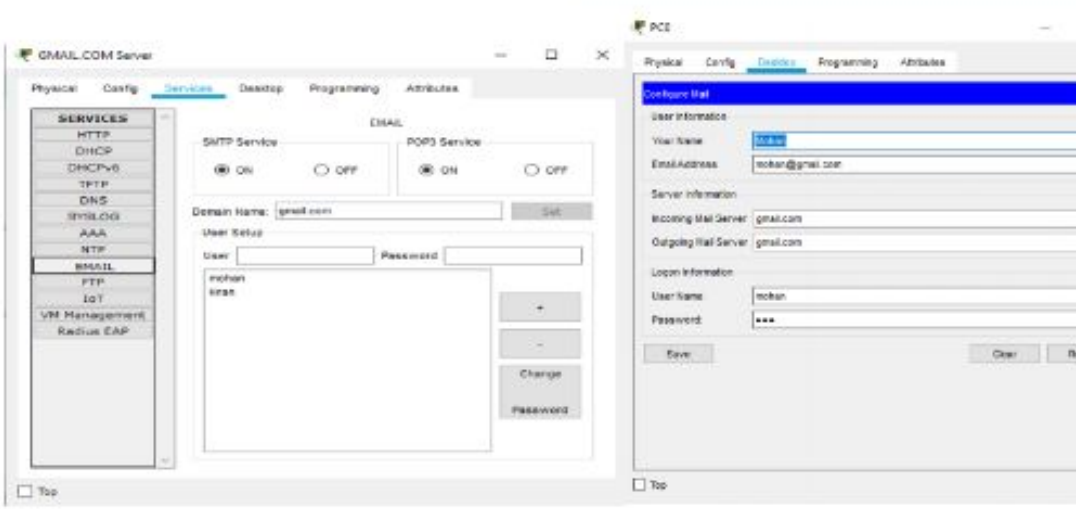
The Domain Name System (DNS) is the phonebook of the Internet. Humans access information online through domain names, like nytimes.com or espn.com. Web browsers interact through Internet Protocol (IP) addresses. DNS translates domain names to IP addresses so browsers can load Internet resources. Each device connected to the Internet has a unique IP address which other machines use to find the device. DNS servers eliminate the need for humans to memorize IP addresses such as 192.168.1.1 (in IPv4), or more complex newer alphanumeric IP

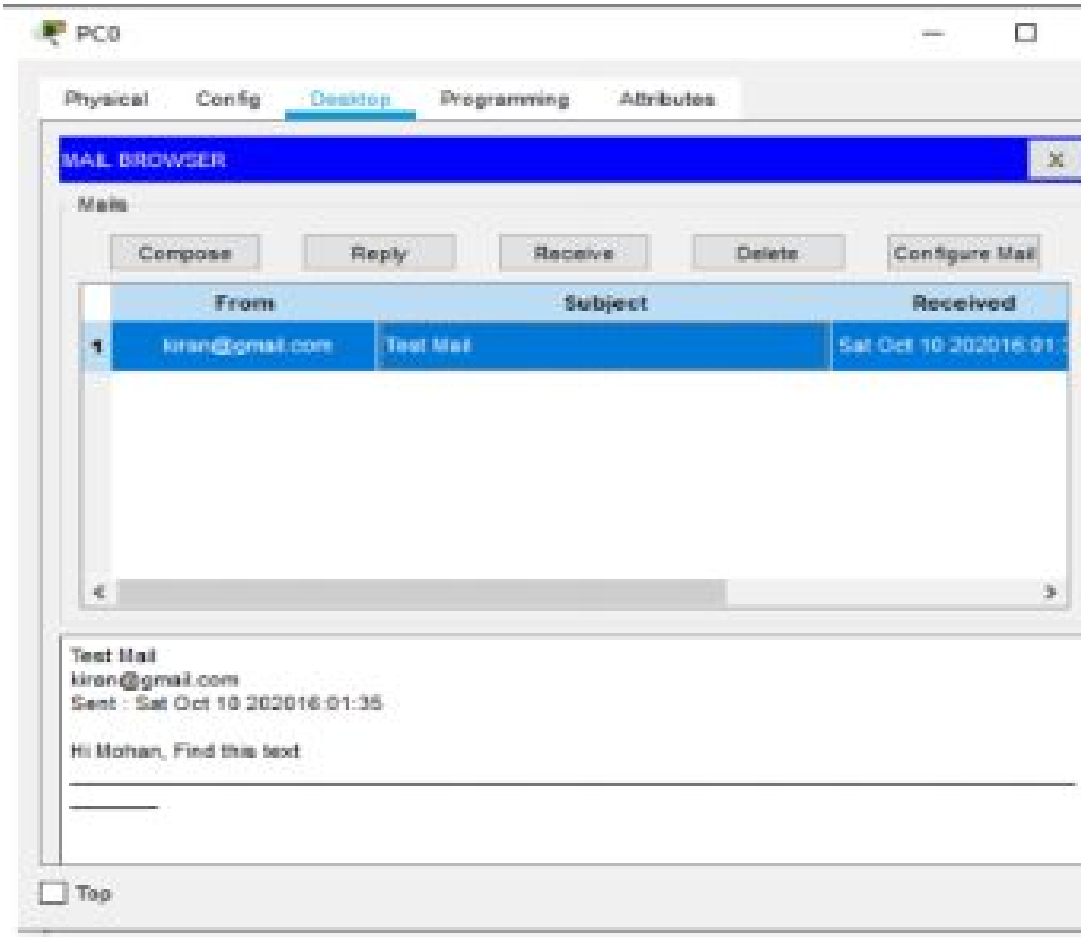
addresses such as 2400:cb00:2048:1::c629:d7a2 (in IPv6).

Screenshot:


1.0.0.2
Server-PT
DNS Server
Fa0
Server-P
DHCP Server
Fa0
1.0.0.1
Fa0/4
Fa0/6
Fa0/3
Fa0/5
Fa0
Server-PT
GMAIL.COM Server
1.0.0.3
Fa0
PC-PT
PC3
Fa0
PC-PT
PC0
mohan@gmail.com
Fa0
PC-PT
PC1
kiran@gmail.com

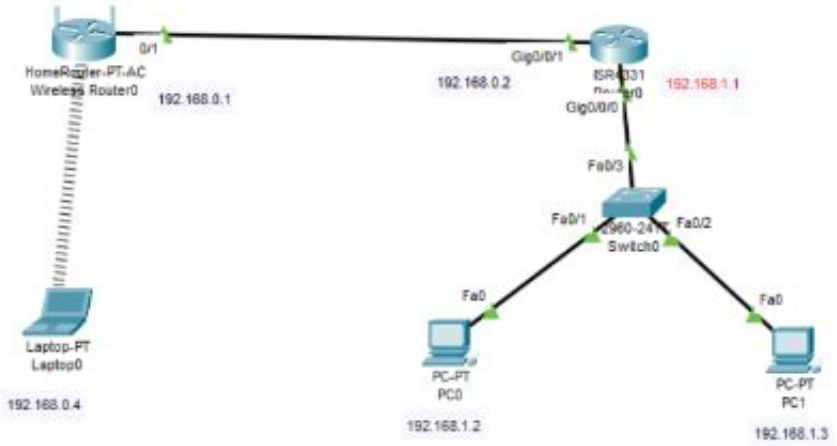

DHCP Server
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DHCP
Interface
FastEthernet0
Service
On
Off
Pool Name
serverPool
Default Gateway
0.0.0.0
DNS Server
1.0.0.2
Start IP Address
Subnet Mask
Maximum Number of Users
25
TFTP Server
0.0.0.0
WLC Address
0.0.0.0
Add
Save
Remove
Pool Name
Default Gateway
DNS Server
Start IP Address
Subnet Mask
Max User
TFTP Server
WLC Address
serverPool
0.0.0.0
1.0.0.2
1.0.0.0
255...
25
0.0.0.0
0.0.0.0
Top
DNS Server
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
DNS
DNS Service
On
Off
Resource Records
Name
Type
A Rec
Address
Add
Save
Remo
No.
Name
Type
Deta
0
gmail.com
A Record
1.0.0.3
DNS Cache
Top

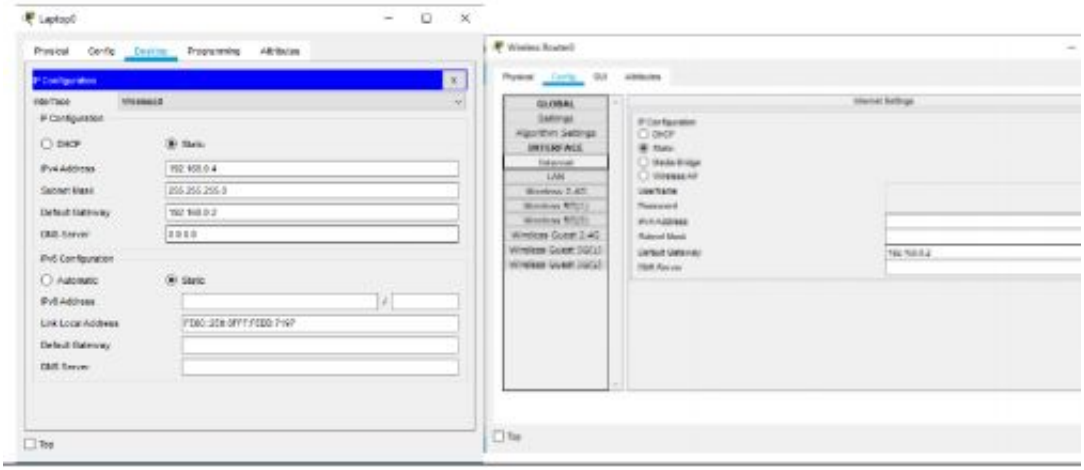

GMAIL.COM Server
Physical
Config
Services
Desktop
Programming
Attributes
SERVICES
HTTP
DHCP
DHCPv6
TFTP
DNS
SYSLOG
AAA
NTP
EMAIL
FTP
IoT
VM Management
Radius EAP
EMAIL
SMTP Service
POP3 Service
ON
OFF
Domain Name:
gmail.com
Set
User Setup
User
Password
mohan
kiran
Change
Password
Top
PC0
Physical
Config
Desktop
Programming
Attributes
Configure Mail
User Information
Your Name
Email Address
mohan@gmail.com
Server Information
Incoming Mail Server
gmail.com
Outgoing Mail Server
gmail.com
Logon Information
User Name
mohan
Password
Save
Clear
Top

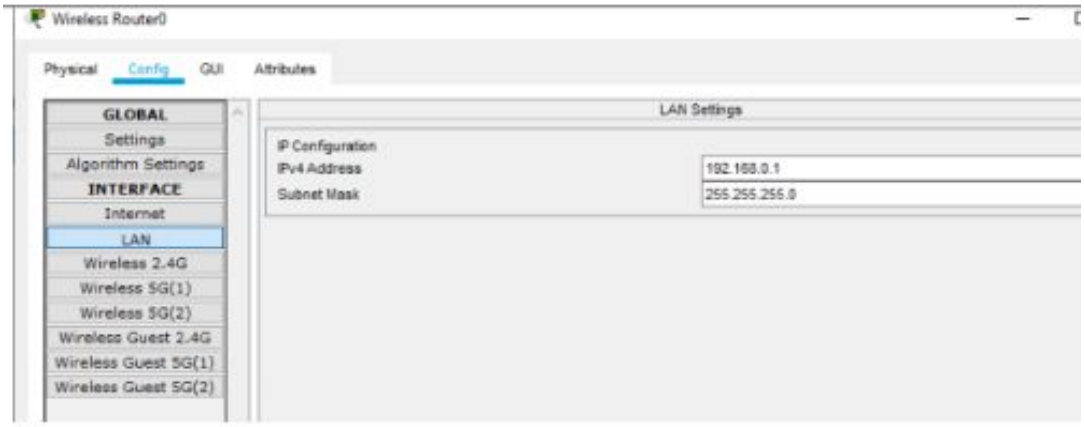

PC0
Physical
Config
Desktop
Programming
Attributes
MAIL BROWSER
Mails
Compose
Reply
Receive
Delete
Configure Mail
From
Subject
Received
kiran@gmail.com
Test Mail
Sat Oct 10 202016 01:
Test Mail
kiran@gmail.com
Sent : Sat Oct 10 202016 01:35
Hi Mohan, Find this text
Top

Question: Configure Wireless router to support mobile devices to connect to the internet.

Screenshot:







Thank you Madam for your support.

Done by A.S.Prithvi Raj
Usn: 1NT18CS001