1. Set up a 2-node wireless network. Analyze TCP performance for this scenario with DSDV as routing protocol.
2. 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 routing protocol.
3. Set up a 6-node wireless network; analyze TCP performance when nodes are static and mobile.
4. 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.
5. Set up a wireless network with mobile nodes, induce 1 to 10% error to the network using uniform error model. Plot the congestion window for TCP connections. Write your observation on TCP performance as error increases in the network.

**PROCEDURE TO SET UP WIRELESS NETWORK IN NETWORK SIMULATIOR-2**

Step1: Define the type for the network components

Step2: Create an instance of the simulator.

Step3: Setup trace and call the method trace-all {}

Step4: Setup nam and call the method namtrace-all-wireless

Step5: Create a topology object with x and y co-ordinates.

Step6: Create the object God

Step7: Configure nodes before we can create them

Step8: Create the nodes

Step9: Give them a position to start

Step10: Define node movements

Step11: Setup traffic flow

Step12: Define stop time when the simulation should end.

Step13: Start the simulation

1. **Set up a 2-node wireless network. Analyze TCP performance for this scenario with DSDV as routing protocols.**

***Aim:***To understand and design the two node wireless network. Also here we understand the working of DSDV routing protocol and analyze its performance.

***Theory:***

***Theory:***

* Define the options for the wireless network.(Choose routing protocol as DSDV, No. of nodes 2)
* Create a simulator object.
* Setup trace and nam and call methods traceall,namtrace-all-wireless
* Create topology, God
* Configure and then create nodes.
* Define node movement.
* Generate TCP traffic and Termination of simulated events
* Start the simulation

Save the following program as 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

**Expected Output:** Demonstration of wireless network, Analysis of DSDV routing protocol.

1. **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.**

***Aim:***To understand and design the three node wireless network. Here we implement the working of DSDV, AODV, DSR routing protocol and analyze their performance.

***Theory:***

* Define the options for the wireless network.(Choose routing protocol as DSDV,AODV,DSR(one at a time) ,No. of nodes 3)
* Create a simulator object.
* Setup trace and nam and call methods traceall,namtrace-all-wireless
* Create topology, God
* Configure and then create nodes.
* Define node movement i.e n0 and n2 moves towards each other and after some time move away from each other.
* Generate TCP traffic and Termination of simulated events
* Start the simulation

Save the following program as 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

Repeat the simulation for AODV and DSR Routing protocols.

**Output**: Demonstration of wireless network, Analysis of routing protocols.

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

***Aim:***To simulate 6 node wireless network. Here we analyze performance of nodes when they are static and mobile.

***Theory:***

***Theory:***

* Define the options for the wireless network.(Choose No. of nodes 6)
* Create a simulator object.
* Setup trace and nam and call methods traceall,namtrace-all-wireless
* Create topology, God
* Configure and then create nodes.
* Define node movement.
* Generate TCP traffic and Termination of simulated events
* Start the simulation

**To make nodes static and mobile follow the setup below**

**Desktop>ns-2.35>ns-2.35all\_in\_one>indeputils>lib>cmu-scen-gen>cbrgen.tcl**

**nodes: 6, max conn: 8, send rate: 0.0, seed: 1.0**

**Desktop>ns-2.35>ns-2.35all\_in\_one>indeputils>lib>cmu-scen-gen>setdest>setdest.h**

**nodes: 6, pause: 2.00, max speed: 40.00, max x: 200.00, max y: 300.00**

Save the following program as 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

**Expected Output:** Setup of wireless network and Performance analysis of static and mobile nodes.

1. **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.**



***Aim:***To simulate wireless network with six nodes. Here we implement the working of DSR routing protocol, IEEE 802.11 and analyze performance when a specific path breaks.

***Theory:***

* Define the options for the wireless network.(Choose routing protocol as DSR,No. of nodes 6)
* Create a simulator object.
* Setup trace and nam and call methods traceall,namtrace-all-wireless
* Create topology, God
* Configure and then create nodes.
* Define node movement.
* Generate TCP traffic and Termination of simulated events
* Start the simulation

Save the following program as 004tcl

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:** Demonstration of wireless network, Analysis of DSR routing protocol.

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

*Solution:*

***Aim:***To simulate wireless network with mobile nodes and induce error to analyze performance..

***Theory:***

* Define the options for the wireless network.
* Create a simulator object.
* Setup trace and nam and call methods traceall,namtrace-all-wireless
* Create topology, God
* Create uniform error procedure and induce the interval at which the error is to be generated.
* Configure and then create nodes.
* Set up the LAN
* Generate TCP traffic and Termination of simulated events

Start the simulation

Save the following program as 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

***Expected output:*** Animated nodes structure is displayed. We need to see the performance of the network with a varying error rate.