

TCL FILE

```
# using parameters from reference guide
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) 50 ;# max packet in ifq
set val(nn) 10 ;# number of mobilenodes
set val(rp) DSDV ;# routing protocol
set val(x) 750 ;# X dimension of topography
set val(y) 750 ;# Y dimension of topography
set val(stop) 100.0 ;# time of simulation end

# init conditions

# 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 \
    -routerTrace  ON \
    -macTrace     ON \
    -movementTrace ON
```

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

```
#Create 10 nodes
```

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

```
$n0 set X_ 550
```

```
$n0 set Y_ 671
```

```
$n0 set Z_ 0.0
```

```
$ns initial_node_pos $n0 20
```

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

```
$n1 set X_ 175
```

```
$n1 set Y_ 108
```

```
$n1 set Z_ 0.0
```

```
$ns initial_node_pos $n1 20
```

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

```
$n2 set X_ 532
```

```
$n2 set Y_ 157
```

```
$n2 set Z_ 0.0
```

```
$ns initial_node_pos $n2 20
```

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

```
$n3 set X_ 165
```

```
$n3 set Y_ 60  
$n3 set Z_ 0.0  
$ns initial_node_pos $n3 20
```

```
set n4 [$ns node]  
$n4 set X_ 547  
$n4 set Y_ 368  
$n4 set Z_ 0.0  
$ns initial_node_pos $n4 20
```

```
set n5 [$ns node]  
$n5 set X_ 726  
$n5 set Y_ 560  
$n5 set Z_ 0.0  
$ns initial_node_pos $n5 20
```

```
set n6 [$ns node]  
$n6 set X_ 197  
$n6 set Y_ 85  
$n6 set Z_ 0.0  
$ns initial_node_pos $n6 20
```

```
set n7 [$ns node]  
$n7 set X_ 114  
$n7 set Y_ 107  
$n7 set Z_ 0.0  
$ns initial_node_pos $n7 20
```

```
set n8 [$ns node]  
$n8 set X_ 354  
$n8 set Y_ 680  
$n8 set Z_ 0.0  
$ns initial_node_pos $n8 20
```

```
set n9 [$ns node]  
$n9 set X_ 38  
$n9 set Y_ 89  
$n9 set Z_ 0.0  
$ns initial_node_pos $n9 20
```

```
# create connections  
#Setup a TCP connection
```

```
set tcp1 [new Agent/TCP]
$ns attach-agent $n1 $tcp1
set sink [new Agent/TCPSink]
$ns attach-agent $n0 $sink
$ns connect $tcp1 $sink
```

```
#Setup a FTP over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ftp1 set type_ FTP
```

```
#Setup a TCP connection
set tcp2 [new Agent/TCP]
$ns attach-agent $n2 $tcp2
$ns connect $tcp2 $sink
```

```
#Setup a FTP over TCP connection
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ftp2 set type_ FTP
```

```
#Setup a TCP connection
set tcp3 [new Agent/TCP]
$ns attach-agent $n3 $tcp3
$ns connect $tcp3 $sink
```

```
#Setup a FTP over TCP connection
set ftp3 [new Application/FTP]
$ftp3 attach-agent $tcp3
$ftp3 set type_ FTP
```

```
#Setup a TCP connection
set tcp4 [new Agent/TCP]
$ns attach-agent $n4 $tcp4
$ns connect $tcp4 $sink
```

```
#Setup a FTP over TCP connection
set ftp4 [new Application/FTP]
$ftp4 attach-agent $tcp4
$ftp4 set type_ FTP
```

```
#Setup a TCP connection
```

```
set tcp5 [new Agent/TCP]
$ns attach-agent $n5 $tcp5
$ns connect $tcp5 $sink
```

```
#Setup a FTP over TCP connection
set ftp5 [new Application/FTP]
$ftp5 attach-agent $tcp5
$ftp5 set type_ FTP
```

```
#Setup a TCP connection
set tcp6 [new Agent/TCP]
$ns attach-agent $n6 $tcp6
$ns connect $tcp6 $sink
```

```
#Setup a FTP over TCP connection
set ftp6 [new Application/FTP]
$ftp6 attach-agent $tcp6
$ftp6 set type_ FTP
```

```
#Setup a TCP connection
set tcp7 [new Agent/TCP]
$ns attach-agent $n7 $tcp7
$ns connect $tcp7 $sink
```

```
#Setup a FTP over TCP connection
set ftp7 [new Application/FTP]
$ftp7 attach-agent $tcp7
$ftp7 set type_ FTP
```

```
#Setup a TCP connection
set tcp8 [new Agent/TCP]
$ns attach-agent $n8 $tcp8
$ns connect $tcp8 $sink
```

```
#Setup a FTP over TCP connection
set ftp8 [new Application/FTP]
$ftp8 attach-agent $tcp8
$ftp8 set type_ FTP
```

```
#Setup a TCP connection
set tcp9 [new Agent/TCP]
$ns attach-agent $n9 $tcp9
$ns connect $tcp9 $sink
```

```

#Setup a FTP over TCP connection
set ftp9 [new Application/FTP]
$ftp9 attach-agent $tcp9
$ftp9 set type_ FTP

$ns at 0.0 "destination"

# define required procedures
proc destination {} {
    global ns n0
    set time 20.0
    set now [$ns now]
    set xx [expr rand()*500]
    set yy [expr rand()*400]
    $ns at $now "$n0 setdest $xx $yy 10.0"
    $ns at [expr $now+$time] "waitfor"
}

proc waitfor {} {
    global ns n0
    set time 2.0
    set now [$ns now]
    set xx [$n0 set X_]
    set yy [$n0 set Y_]
    $ns at $now "$n0 setdest $xx $yy 100.0"
    $ns at [expr $now+$time] "destination"
}

proc finish {} {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    # print in console
    exec python analysis.py
    # plot graph
    exec xgraph graph.txt &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {

```

```

    $ns at $val(stop) "\$n$i reset"
}
$ns at 10 "$ftp1 start"
$ns at 10 "$ftp2 start"
$ns at 10 "$ftp3 start"
$ns at 10 "$ftp4 start"
$ns at 10 "$ftp5 start"
$ns at 10 "$ftp6 start"
$ns at 10 "$ftp7 start"
$ns at 10 "$ftp8 start"
$ns at 10 "$ftp9 start"

$ns at 95 "$ftp1 stop"
$ns at 95 "$ftp2 stop"
$ns at 95 "$ftp3 stop"
$ns at 95 "$ftp4 stop"
$ns at 95 "$ftp5 stop"
$ns at 95 "$ftp6 stop"
$ns at 95 "$ftp7 stop"
$ns at 95 "$ftp8 stop"
$ns at 95 "$ftp9 stop"

$ns at 100 "finish"
$ns at $val(stop) "$ns nam-end-wireless $val(stop)"
$ns at $val(stop) "finish"
$ns at $val(stop) "puts \"done\" ; $ns halt"
$ns run

```

PYTHON SCRIPT FOR GRAPH AND ANALYSIS

```
=====
```

```

f = open('out.tr', 'r')

plot_file = open('graph', 'w')

drop = 0
packets = 0

```

```

for line in f:
    if(line[0] == 'd'):
        drop += 1

    if 'ack' not in line:
        packets += 1

    line = line.split(' ')
    plot_file.write(line[1]+" "+str(packets*1.0/float(line[1]))+"\n")

print "no of packets dropped are: ", drop
print "throughput is", packets*1.0/5, "kpbs"

f.close()
plot_file.close()

```

