

1.

```

# This script is created by NSG2 beta1
# <http://wushoupong.googlepages.com/nsg>

#=====
#   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 ;# 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 \
-routerTrace  ON \
-macTrace     OFF \
-movementTrace ON

```

```
#=====
```

```
# Nodes Definition
```

```
#=====
```

```
#Create 12 nodes
```

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

```
$n0 set X_ 0
```

```
$n0 set Y_ 0
```

```
$n0 set Z_ 0.0
```

```
$ns initial_node_pos $n0 50
```

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

```
$n1 set X_ 94.3
```

```
$n1 set Y_ 94.3
```

```
$n1 set Z_ 0.0
```

```
$ns initial_node_pos $n1 50
```

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

```
$n2 set X_ 188.6
```

```
$n2 set Y_ 188.6
```

```
$n2 set Z_ 0.0
```

```
$ns initial_node_pos $n2 50
```

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

```
$n3 set X_ 282.9
```

```
$n3 set Y_ 282.9
```

```
$n3 set Z_ 0.0
```

```
$ns initial_node_pos $n3 50
```

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

```
$n4 set X_ 377.2
```

```
$n4 set Y_ 377.2
```

```
$n4 set Z_ 0.0
```

```
$ns initial_node_pos $n4 50
```

```
set n5 [$ns node]
```

```
$n5 set X_ 471.5
```

```
$n5 set Y_ 471.5
```

```
$n5 set Z_ 0.0
```

```
$ns initial_node_pos $n5 50
```

```
set n6 [$ns node]
```

```
$n6 set X_ 565.8
```

```
$n6 set Y_ 565.8
```

```
$n6 set Z_ 0.0
```

```
$ns initial_node_pos $n6 50
```

```

set n7 [$ns node]
$n7 set X_ 660.1
$n7 set Y_ 660.1
$n7 set Z_ 0.0
$ns initial_node_pos $n7 50
set n8 [$ns node]
$n8 set X_ 754.4
$n8 set Y_ 754.4
$n8 set Z_ 0.0
$ns initial_node_pos $n8 50
set n9 [$ns node]
$n9 set X_ 848.7
$n9 set Y_ 848.7
$n9 set Z_ 0.0
$ns initial_node_pos $n9 50
set n10 [$ns node]
$n10 set X_ 943
$n10 set Y_ 943
$n10 set Z_ 0.0
$ns initial_node_pos $n10 50
set n11 [$ns node]
$n11 set X_ 1037.3
$n11 set Y_ 1037.3
$n11 set Z_ 0.0
$ns initial_node_pos $n11 50

```

```

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

```

```

set tcp1 [new Agent/TCP]
$ns attach-agent $n0 $tcp1
set sink1 [new Agent/TCPSink]
$ns attach-agent $n11 $sink1
$ns connect $tcp1 $sink1
$tcp1 set fid_ 1

```

```

set tcp2 [new Agent/TCP/Reno]
$ns attach-agent $n4 $tcp2
set sink2 [new Agent/TCPSink]
$ns attach-agent $n0 $sink2
$ns connect $tcp2 $sink2
$tcp2 set fid_ 2

```

```

set udp1 [new Agent/UDP]
$ns attach-agent $n4 $udp1
set null1 [new Agent/Null]
$ns attach-agent $n11 $null1
$ns connect $udp1 $null1
$udp1 set fid_ 2

```

```

#=====

```

```

#    Applications Definition
#=====
#Setup a FTP Application over TCP connection
set ftp1 [new Application/FTP]
$ftp1 attach-agent $tcp1
$ns at 0.1 "$ftp1 start"
$ns at 5.0 "$ftp1 stop"

#Setup a CBR Application over UDP connection
set cbr1 [new Application/Traffic/CBR]
$cbr1 attach-agent $udp1
$cbr1 set packetSize_ 1024
$cbr1 set rate_ 1.0Mb
$cbr1 set random_ null
$ns at 0.5 "$cbr1 start"
$ns at 4.5 "$cbr1 stop"

#Setup a FTP Application over TCP connection
set ftp2 [new Application/FTP]
$ftp2 attach-agent $tcp2
$ns at 0.75 "$ftp2 start"
$ns at 3.75 "$ftp2 stop"

#=====
#    Termination
#=====
#Define a 'finish' procedure
proc finish { } {
    global ns tracefile namfile
    $ns flush-trace
    close $tracefile
    close $namfile
    exec nam out.nam &
    exit 0
}
for {set i 0} {$i < $val(nn)} {incr i} {
    $ns at $val(stop) "\"$n$i reset"
}
$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

```

2.

```
BEGIN {
```

```
}
```

```
BEGIN {
```

```

sends=0.0;
recvs=0.0;
routing_packets=0.0;
droppedBytes=0;
droppedPackets=0;
highest_packet_id =0;
sum=0.0;
recvnum=0.0;

beacon_no=0;           #for control packet
frp_no=0;              #for control packet
rsup_no=0;             #for control packet
rrep_no=0;             #for control packet
rreq_no=0;             #for control packet
Control_Total_pkt=0;   #for control packet

cbr_no=0;              #for data packet
tcp_no=0;              #for data packet
ack_no=0;              #for data packet
Data_Total_pkt=0 ;     #for data packet

recvdSize = 0.0;       #for throughput
startTime = 1e6;       #for throughput
stopTime = 0.0;        #for throughput
}
{
# Trace line format: normal (old trace format)
if ($2 != "-t") {
    event = $1;         # event is the contents of field #1 in the trace file which is s,r,f or d
    time = $2;          # time = contents of 2nd field in the trace file // added by idris
    node_id = $3;
    pkt_id = $6;
    pkt_size = $8;
    flow_t = $5;
    level = $4;
    pkt_type = $7;
}
# Trace line format: new
if ($2 == "-t") {
    event = $1;         # event is the contents of field #1 in the trace file which is s,r,f or d
    time = $3;          # time = contents of 3rd field in the trace file // added by idris
    node_id = $5;
    flow_id = $39;
    packet_id = $41;     # packet_id = contents of field #41 in the trace file // added by idris
    pkt_size = $37;      # pkt_size is the contents of field #37 in the trace file
    flow_t = $45;
    level = $19;
    pkt_type= $35;       # pkt_type is the contents of field #35 in the trace file
}

#===== Start of Performance calculation

```

```

# CALCULATE PACKET DELIVERY FRACTION (Data Packets)
if (( event == "s" ) && ( pkt_type == "cbr" || pkt_type == "tcp" || pkt_type == "ack" ) &&
(level=="AGT")) { sends++; }

if (( event == "r" ) && ( pkt_type == "cbr" || pkt_type == "tcp" || pkt_type == "ack" ) &&
(level=="AGT")) { recvs++; }

# CALCULATE DELAY
if ( start_time[packet_id] == 0 ) start_time[packet_id] = time;
if (( event == "r" ) && ( pkt_type == "cbr" || pkt_type == "tcp" || pkt_type == "ack" ) &&
(level=="AGT")) { end_time[packet_id] = time; }
else { end_time[packet_id] = -1; }

# CALCULATE TOTAL OVERHEAD (Control Packets)
if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "DSDV" || pkt_type
=="REQUEST" || pkt_type == "REPLY" || pkt_type == "AODV" ||
    pkt_type == "DSR" || pkt_type == "message" || pkt_type == "BEACON" || pkt_type == "FRIP" ||
pkt_type == "RSUP" || pkt_type == "ROUTE_REQ" || pkt_type == "ROUTE_REP"))
    routing_packets++;

#=====
=====

if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "BEACON"))
    beacon_no++;

if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "FRIP"))
    frip_no++;

if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "RSUP"))
    rsup_no++;

if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "ROUTE_REQ"))
    rreq_no++;

if ((event == "s" || event == "f") && (level=="RTR") && (pkt_type == "ROUTE_REP"))
    rrep_no++;

if ((event == "s" || event == "f") && (level=="RTR" ) && (pkt_type != "cbr" && pkt_type !=
"ack" && pkt_type != "tcp"))
{
    #printf("event %c level %s pkt_type %s \n",event,level,pkt_type);
    Control_Total_pkt++;
}

```

```

if ((event == "s") && (level=="AGT") && (pkt_type == "cbr"))
    cbr_no++;
if ((event == "s") && (level=="AGT") && (pkt_type == "tcp"))
    tcp_no++;
if ((event == "s") && (level=="AGT") && (pkt_type == "ack"))
    ack_no++;
if ((event == "s") && (level=="AGT") && (pkt_type == "cbr" || pkt_type == "ack" || pkt_type ==
"tcp"))
    Data_Total_pkt++;

#=====

=====

# DROPPED DSR PACKETS
if (( event == "d" ) && ( pkt_type == "cbr" || pkt_type == "tcp" ) && ( time > 0 ))
{
    droppedBytes=droppedBytes+ pkt_size;
    droppedPackets=droppedPackets+1;
}

#find the number of packets in the simulation
if (packet_id > highest_packet_id)
    highest_packet_id = packet_id;

#===== End of Performance calculation

#===== Start of Throughput calculation
# Store start time
if ((level == "AGT") && (event == "+" || event == "s") && pkt_size >= 512) {
    if (time < startTime) {
        startTime = time;
    }
}

# Update total received packets' size and store packets arrival time
if ((level == "AGT") && event == "r" && pkt_size >= 512) {
    if (time > stopTime) {
        stopTime = time;
    }
    # Rip off the header
    hdr_size = pkt_size % 512;
    pkt_size -= hdr_size;
    # Store received packet's size;
    recvdSize += pkt_size;
}
#===== End of Throughput calculation

}

END {

#===== This part for Performance calculation

```

```

for ( i in end_time )
{
start = start_time[i];
end = end_time[i];
packet_duration = end - start;
if ( packet_duration > 0 )
{
sum += packet_duration;
recvnum++;
}
}

if(recvnum==0)
recvnum++; # set to 1
if(sends==0)
{ printf("NOTE: No Data Sent.....");
sends++;} # set to 1
if(recvs==0)
{ printf("No Data Received.....");
recvs++;} # set to 1
print"\n=====\\n"
delay=sum/recvnum;
NRL = routing_packets/recvs; #normalized routing load
PDF = (recvs/sends)*100; #packet delivery ratio[fraction]

printf("No. of data send = %.2f\\n",sends);
#printf("No. of data recv = %.2f\\n",recvs);
printf("No. of dropped data (packets) = %d\\n",droppedPackets);
#printf("No. of dropped data (bytes) = %d\\n",droppedBytes);
#printf("lost = %.2f\\n",sends-recvs-droppedPackets);
#printf("No. of routing packets = %.2f\\n",routing_packets);
#printf("No. of Controll packets = %.2f\\n",Control_Total_pkt);
#printf("TCP\\tCBR\\tACK\\tTOTAL\\n");
#printf("%5d\\t%5d\\t%5d\\t%5d\\n",tcp_no,cbr_no,ack_no,Data_Total_pkt);

# printf("\\t\\t BEACON FRIP RSUP RREQ RREP TOTAL\\n");
# printf(" \\t\\t\\t\\t\\t\\t\\t\\t %d \\t %d \\t %d \\t %d \\t\\t %d \\t\\t %d\\n",beacon_no,frp_no,rsup_no,rreq_no,rrep_no,Control_Total_pkt);

#printf("Packet Delivery Ratio = %.2f\\n",PDF);
#printf("Normalized Routing Load = %.2f\\n",NRL);
#printf("E-to-E Delay = %.2f\\n",delay*1000);
#printf("Throughput = %.2f\\n",(recvdSize/(stopTime-startTime))*(8/1000));
#printf("Start time = %.2f, End time = %.2f \\n",startTime,stopTime);
}

function abs(value) {
if (value < 0) value = 0-value
return value
}

```



```
}
```

```
nit@nit-HP-Compaq-Elite-8300-SFF:~$ awk -f 1.awk out.tr
```

```
=====
```

```
No. of data send = 26874.00
```

```
No. of dropped data (packets) = 0
```