

dsr.tcl
#Dynamic Source Routing

Define options

```
set chan      Channel/WirelessChannel ; # channel type
set prop      Propagation/TwoRayGround ; # radio-propagation model
set netif     Phy/WirelessPhy ; # network interface type
set mac       Mac/802_11 ; # MAC type
set ifq       Queue/DropTail/PriQueue ; # interface queue type
set ll        LL ; # link layer type
set antenna   Antenna/OmniAntenna ; # antenna model
set ifqlength 50 ; # max packet in ifq ( use to assign
                                     #the buffering capacity of wireless interface)
set nodes     3 ; # number of mobilenodes
set rprotocol DSR ; # routing protocol
set xaxis     500 ; # X dimension of topography
set yaxis     400 ; # Y dimension of topography
set simstop   150 ; # time of simulation end

set ns        [new Simulator]
```

#newtrace (new format of trace file for wireless)

for using this write as below

```
$ns use-newtrace
```

```
set tracefd   [open simple.tr w]
set windowVsTime2 [open win.tr w]
set namtrace   [open simple.nam w]
```

trace-all \$filename causes trace objects to be pushed on all links. If you only want to trace one link, there's no need for this overhead. Saving is about 14 KB/link.

```
$ns trace-all $tracefd
```

```
$ns namtrace-all-wireless $namtrace $xaxis $yaxis
```

Topography is the study of Earth's surface shape and features or those of planets, moons, and asteroids

set up topography object

Create and configure topography object (Used for mobile scenario)

```
set topo [new Topography]
```

The load_flatgrid object is used to specify a 2-D terrain. Support is available for simulation of 3D terrains for more realistic depiction of scenarios.

```
$topo load_flatgrid $xaxis $yaxis
```

GOD or General Operations Director is a ns-2 simulator object, which is used to store global information about the state of the environment, network, or nodes that an omniscient observer would have, but that should not be made known to any participant in the simulation

```
create-god $nodes
```

configure the nodes

```
$ns node-config -adhocRouting $rprotocol \  
    -llType $ll \  
    -macType $mac \  
    -ifqType $ifq \  
    -ifqLen $ifqlength \  
    -antType $antenna \  
    -propType $prop \  
    -phyType $netif \  
    -channelType $chan \  
    -topoInstance $topo \  
    -agentTrace ON \  
    -
```

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

```
# Create the specified number of nodes [$nodes] and  
"attach" them  
# to the channel.
```

```
for {set i 0} {$i < $nodes} {incr i} {  
    set n($i) [$ns node]  
}
```

```
#By default, a node is specified as a unicast node. If a multicast protocol is  
desired, a #separate clause has to be specified during simulator initialization-
```

```
# set ns [new Simulator -multicast on]
```

```
# Provide initial location of mobilenodes
```

```
$n(0) set X5.0
```

```
$n(0) set Y5.0
```

```
$n(0) set Z0.0
```

```
$n(1) set X490.0
```

```
$n(1) set Y285.0
```

```
$n(1) set Z0.0
```

```
$n(2) set X150.0
```

```
$n(2) set Y240.0
```

```
$n(2) set Z0.0
```

```
# Generation of movements
```

```
$ns at 10.0 "$n(0) setdest 250.0 250.0 3.0"
```

```
$ns at 15.0 "$n(1) setdest 45.0 285.0 5.0"
```

```
$ns at 110.0 "$n(2) setdest 480.0 300.0 5.0"
```

Set a TCP connection between n(0) and n(1)

```
set tcp [new Agent/TCP/Newreno]
$tcp set class_ 2
set sink [new Agent/TCPSink]
$ns attach-agent $n(0) $tcp
$ns attach-agent $n(1) $sink
$ns connect $tcp $sink
set ftp [new Application/FTP]
$ftp attach-agent $tcp
$ns at 10.0 "$ftp start"
```

Printing the window size

```
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 10.1 "plotWindow $tcp $windowVsTime2"
```

Define node initial position in nam

```
for {set i 0} {$i < $nodes} {incr i} {
# 30 defines the node size for nam
  $ns initial_npos $n($i) 30
}
```

Telling nodes when the simulation ends

```
for {set i 0} {$i < $nodes} {incr i} {
  $ns at $simstop "$n($i) reset";
}
```

ending nam and the simulation

```
$ns at $simstop "$ns nam-end-wireless $simstop"
```

```
$ns at $simstop "stop"
```

```
$ns at 150.01 "puts \"end simulation\" ; $ns halt"
```

```
proc stop {} {
```

```
    global ns tracefd namtrace
```

```
    $ns flush-trace
```

```
    close $tracefd
```

```
    close $namtrace
```

```
}
```

```
$ns run
```

