CSE 589 - MODERN NETWORKING CONCEPTS

Project Assignment – 3

<u>DETAILED DESIGN</u>

MAC Random Transmission Protocol Using NS-2

<u>Team</u>

Anirudh Yellapragada

Chaitanya Vedurupaka

Wireless Sensor Networks :

In this project, the design of the network has been made using the NS2 tool using below mentioned files :

1. TCL Script
2. Random MAC protocol
3. Python script

TCL Script :

randomMAC.tcl:

For the network simulator to work, we need to write a simulation script in Tool Command Language(TCL). In the script different components are as follows :

1. Specifying the parameters for a network as per project requirement.

```
#Project Parameters
set duration                    5;
set val(intervalVal)            0.02;
set val(packetSizeVal)          16; #128 bits
set val(repeatTxCopies)       1
set val(node_num)               101
set val(dimx)                   50
set val(dimy)                   50
```

2. Specifying the node parameters.

For example the channel type(Wired/Wireless), the propagation channel(free-space/two-ray ground reflection/shadowing), the network interface type(wireless shared media implemented as class Phy/WirelessPhy), mac type which specifies which MAC layer protocol to be used(802.11, our own implemented protocol), interface queue type (Queue, DropTail, PriQueue) and various other node related parameters which indicate the kinds of protocols and channels to be used in the network.

```
#Node Parameters
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/_randomMAC                  ;# 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)             $val(node_num)                           ;# number of mobilenodes
# routing protocol as specified in piazza
set val(rp)              DumbAgent;
```

3. Creating a new simulator object.

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

4. Creating a topology consisting of 100 source nodes and one sink node.

```
set topo                      [new Topography]
$topo load_flatgrid           $val(dimx) $val(dimy)
```

5. Creating sink node.

```
#Creating the sink agent and connecting it to sink node
set sinkNode [$ns node]
$sinkNode random-motion 0
$sinkNode set X_ [expr $val(dimx)/2]
$sinkNode set Y_ [expr $val(dimy)/2]
$sinkNode set Z_ 0
$ns initial_node_pos $sinkNode 5

set null0 [new Agent/Null]
$ns attach-agent $sinkNode $null0
```

6. Creating the source nodes

```
for {set i 0} {$i <  $val(nn)-1} {incr i} {
    set node($i) [$ns node]
    set random1 [$randomX value]
    set random2 [$randomY value]
    $node($i) set X_ $random1
    $node($i) set Y_ $random2
    $node($i) set Z_ 0
    $ns initial_node_pos $node($i) 1

    set udp($i) [new Agent/UDP]
  $udp($i) set class_ $i
    $ns attach-agent $node($i) $udp($i)

    #Connecting te agents
    $ns connect $udp($i) $null0

    set cbr($i) [new Application/Traffic/CBR]
    $cbr($i) set packetSize_ $val(packetSizeVal)
    $cbr($i) set interval_ $val(intervalVal)
    $cbr($i) attach-agent $udp($i)

    set time1 $randomTime
    $ns at $time1 "$cbr($i) start"
    $ns at $duration "$cbr($i) stop"

}
```

<u>Key Note</u> : The number of retransmissions and the packet generation time interval are specified as part of TCL which will be used in the new MAC rotocol files as we bind these values to the local variables using the TCL object.

Random Transmission Protocol:

randomMAC.cc:

The reference content for this new protocol we have implemented is the content from mac-simple.cc. In this MAC layer protocol, we need to handle two functions in general. Below are the two different scenarios that can happen in any protocol layer of a network model.

1. If a node A wants to send a packet to node B, then the packet from the above layers in node A has to send a packet to the link layer MAC protocol.

So, the first scenario is that this new protocol can receive a packet from its upper layers.

Action to be taken: Here this new protocol just needs to forward the new packet to the physical layer.

2. If a node A receives a packet from node B, then the physical layer of the node A sends the packet to its upper link layer.

So, the second scenario is that this new protocol can receive a packet from its lower layers.

Action to be taken: Here this new protocol just needs to forward the new packet to the upper network layer.

For this project we need to handle retransmissions from the source nodes. For this purpose, the send function of the new protocol is enhanced to retransmit X number of times, where X can be specified from the TCL script.

```
for(int i=0; i<repeat_; i++){
pkt_time[i] = (rand()%100)/100.0 * interval_;
if(final_pkt_time < pkt_time[i])
    final_pkt_time = pkt_time[i];
}

//Schedule for retransmission.
for(int i=0;i<repeat_;i++){

    if(pkt_time[i]!=final_pkt_time) {
        _schedule.schedule(this, (Event*)p->copy(),pkt_time[i]);
    }
}
waitTimer->restart(final_pkt_time);
```

Python script:

We have written a python script in order to process the resultant trace file for obtaining the average delivery probability.
The trace file contains number of lines, each line specifying different characteristics of a packet. Below is a sample line from a generated trace file.

```
s 2.925025000 _1_ MAC  --- 280 cbr 16 [0 1000000 8 0] ------- [1:0 0:0 32 0] [140] 0 0
D 2.925089079 _0_ MAC  COL 270 cbr 16 [0 1000000 8 0] ------- [1:0 0:0 32 0] [135] 0 0
```

1. The first field specifies the event type of the packet.
    s – indicates that it is a sending packet
    r – indicates that it is a receiving packet
    D – indicates a packet which is dropped
2. The second field specifies the Time when it was generated.
3. The third field specifies the node number which is sending the packet.
4. The fourth field specifies the layer through which the packet Is being sent.
5. The sixth field specifies the unique Identifier for the packet.
6. The seventh field specifies the traffic type of the packet.
7. The eighth field specifies the size of the packet.

Out of the above fields, we require the traces with the event type as s, r or D, the layer as MAC and the traffic type as cbr. In order to calculate the average delivery probability we need to calculate unique packets sent and unique packets received. The number of unique packets sent would be obtained by using a set function of python3. Now the average probability is calculate using the below formula :
Average delivery probability = Number of successfully received packets/Number of packets sent

Once the above probability s obtained, we plot a graph of probability vs X using xgraph of the NS2 tool.