

Appendix A Configuration and Extension Files of Simulator

A.1 Configuration File

```
#omnetpp.ini
[General]
network = btes
ini-warnings = no
snapshot-file = snapShot.sna
output-vector-file = xputVect.vec
output-scalar-file = yputVect.sca
sim-time-limit = 5m
total-stack-kb = 32768
num-rngs=5

[Parameters]
#btes module
btes.height = 100
btes.width = 100
btes.dim = 50

#mobile host module
btes.mobileHost[*].x = intuniform(5,195)
btes.mobileHost[*].y = intuniform(5,195)
btes.mobileHost[*].routeAlgorithm = "AODV"
btes.mobileHost[*].macAlgorithm = "SimpleMac"

#pyisic module
btes.mobileHost[*].physic.txPower = uniform(9000,9900)
btes.mobileHost[*].physic.rxThreshold = 1
btes.mobileHost[*].physic.channelDelay = 0.0001
btes.mobileHost[*].physic.channelDatarate = 11.04858e+6
btes.mobileHost[*].physic.channelError = 0.000000
```

```

#mac module
btes.mobileHost[*].mac.promisueMode = true;
btes.mobileHost[*].mac.inBufferSize =0.1e6;8.38864e6

#application module
;pakets per secod
btes.mobileHost[*].app.rate = truncnormal(0.01,0.03)
btes.mobileHost[*].app.pktSize = 50
;time elapsed between two data burst
btes.mobileHost[*].app.burstInterval = truncnormal(0.01,0.03)
;indicate the active hosts, they will generate traffic in all the
;simulator run.
;Due to the different random position of each node
;fixing the host name that will work will not cause any
;statistical problem
; btes.mobileHost[1].app.active = 1
; btes.mobileHost[3].app.active = 1
; btes.mobileHost[7].app.active = 1
btes.mobileHost[*].app.active = 1
;it is necessary to add this due to a probable bug

```

A.2 Extension Files

```

module BTES
  parameters:
    dim: numeric,
    width: numeric,
    height: numeric;
  submodules:
    mobileHost: MobileHost[dim];
  parameters:
    numHost = dim,
    Xbound = width,

```

```

        Ybound = height,
        //x = width /2,
        //y = height /2;
        x = intuniform(20, width -20),
        y = intuniform(20, height -20);
        //x = 60 + (index % 5 ) * 120,
        //y = 30 + (index - index % 5 ) * 30 ;
        display: "p=53,40;i=misc/node";
        display: "p=10,10;b=$width,$height";
endmodule

network btes: BTES
    parameters:
        dim = input,
        width = input,
        height = input;
endnetwork

import
    "simple";

module MobileHost
    parameters:
        numHost : numeric const,
        x : numeric, // actual position
        y : numeric, // "
        Xbound : numeric,
        Ybound : numeric,
        macAlgorithm : string,
        routeAlgorithm : string;

```

submodules:

```

    physic: Physic;
    parameters:
        txPower = input,
        rxThreshold = input,

```

```
channelDelay = input,  
channelDatarate = input,  
channelError = input;  
display: "p=148,62;b=104,10";  
mac: macAlgorithm like Mac;
```

parameters:

```
inBufferSize = input,  
promisqueMode = input;  
display: "p=148,116;b=104,10";
```

route: routeAlgorithm like Routing;

parameters:

```
display: "p=148,171;b=104,10";
```

app: Application;

parameters:

```
rate = input,  
pktSize = input,  
burstInterval = input,  
hostNum = numHost,  
active = input;  
display: "p=148,216;b=104,10";
```

connections:

```
app.out --> route.fromApp display "m=m,24,16,24,16";  
route.toMac --> mac.fromRoute display "m=m,24,0,24,4";  
mac.toRoute --> route.fromMac display "m=m,80,0,80,0";  
mac.toPh --> physic.fromMac display "m=m,24,12,24,12";  
physic.toMac --> mac.fromPh display "m=m,80,0,80,8";  
display: "p=10,10;b=287,250";  
endmodule
```

simple Physic

parameters:

```

        txPower,
        rxThreshold,
        channelDelay,
        channelDatarate,
        channelError;
    gates:
        in: fromMac;
        out: toMac;
endsimple

```

simple Mac

```

    parameters:
        inBufferSize,
        promisqueMode : bool;
    gates:
        in: fromPh;
        in: fromRoute;
        out: toRoute;
        out: toPh;
endsimple

```

simple Application

```

    parameters:
        rate, //paket per second
        pktSize,
        hostNum,
        active,
        burstInterval; // time(s) between two data bursts
    gates:
        out: out;
endsimple

```

simple Routing

```

    gates:

```

```
        in: fromMac;
        in: fromApp;
        out: toMac;
endsimple

//IEEE-802.11a channel
//channel etere
//    delay 0.0001;
//    datarate 11.04858e+6 ;
//    error 0.000001;
//endchannel
```