Metrics achieved-

Idle percentage

CPU Power

LPM Power

TX Power

RX Power

Total Power

All CPU Power

All LPM Power

All TX Power

All RX Power

Method-

1. Download the compiler tar zip from - <https://github.com/pksec/msp430-gcc-4.7.3>
2. Follow the instructions given in the README.md
3. The only change to be made is, export PATH=/opt/compilers/mspgcc-4.7.3/bin:$PATH
4. Next while compiling the nodes for the network add the following to the compile command, WITH\_COMPOWER=1 , at the end of the command.
5. If there are any errors related to memory, use motes with higher memory capacity(tested for Z1).
6. This method prints the power parameter in the log display.
7. Save the logs at the end of the simulation.
8. Run the following commands on the log file,

cat /path/to/log/file | grep P | cut -f 2- | grep ' P ' | tr -t '\t' ' ' | cut -d ' ' -f 1,3-18 | sed 's/ID://g' | awk '{print $2 " " $3 " " $1 ".0 " $5 " " $6 " " $7 " " $8 " " $9 " " $10 " " $11 " " $12 " " $13 " " $14 " " $15 " " $16 " " $17 " " $18}' > /path/to/result/file

1. Next save and run this perl script in the terminal on the result file.

#!/usr/bin/perl

#/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

print " Start of Power Analysis Program\n"; # print a message

$file\_path = '/path/to/result/file’;

open(INFO, $file\_path);

#@lines = <INFO>;

#close(INFO);

#print @lines;

$\_ = $file\_path;

$max\_seq = 0;

$voltage = 3;

$power\_cpu = 1.800 \* $voltage;

$power\_lpm = 0.0545 \* $voltage;

$power\_tx = 17.7 \* $voltage;

$power\_rx = 20.0 \* $voltage;

while(<INFO>)

{

if(/P (\d+)\.\d+ (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+) (\d+)/)

{

$node = $1;

$seq = $2;

$all\_cpu = $3;

$all\_lpm = $4;

$all\_tx = $5;

$all\_rx = $6;

$all\_idle\_tx = $7;

$all\_idle\_rx = $8;

$cpu = $9;

$lpm = $10;

$tx = $11;

$rx = $12;

$idle\_tx = $13;

$idle\_rx = $14;

$total\_time = $all\_cpu + $all\_lpm;

$nodes{$node} = 1;

$radio\_now = $tx + $rx;

$idle\_now = $idle\_tx + $idle\_rx;

$cpu\_now = $lpm + $cpu;

$dutycycle = $radio\_now / $cpu\_now;

$dutycycle\_for\_node[$node][$seq] = $dutycycle;

$idle\_for\_node[$node][$seq] = $idle\_now / $cpu\_now;

if($seq > $max\_seq) {

$max\_seq = $seq;

}

}

}

foreach $j (keys %nodes) {

$avg = 0;

for($i = 0; $i < $max\_seq; $i++) {

$avg += $dutycycle\_for\_node[$j][$i];

}

$idle\_avg = 0;

for($i = 0; $i < $max\_seq; $i++) {

$idle\_avg += $idle\_for\_node[$j][$i];

}

print $avg / $max\_seq . " " . $idle\_avg / $max\_seq . " $j\n";

$total\_avg += $avg;

$total\_idle += $idle\_avg;

}

print "\n";

print STDERR "Idle percentage " . $total\_idle / $total\_avg . "\n";

print STDERR "CPU Power" .$all\_cpu\*$power\_cpu/$total\_time. "\n";

print STDERR "LPM Power" .$all\_lpm\*$power\_lpm/$total\_time. "\n";

print STDERR "TX Power" .$all\_tx\*$power\_tx/$total\_time. "\n";

print STDERR "RX Power" .$all\_rx\*$power\_rx/$total\_time. "\n";

print STDERR "Total Power" .($all\_cpu\*$power\_cpu + $all\_lpm\*$power\_lpm + $all\_tx\*$power\_tx + $all\_rx\*$power\_rx)/$total\_time. "\n";

print STDERR "All CPU Power" .$all\_cpu. "\n";

print STDERR "All LPM Power" .$all\_lpm. "\n";

print STDERR "All TX Power" .$all\_tx. "\n";

print STDERR "All RX Power" .$all\_rx. "\n";

#/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

1. The output should be displayed in the terminal.