```
%% Setup
clc; clear;

max_datarate   = 250*1000*8;%250k-byte Telos_B datasheet
packagesize    = 128*8;     %128k-byte WSN Problem Description
TransmitPeriod = packagesize/max_datarate;                %Seconds
ReceivePeriod  = packagesize/max_datarate;                %Seconds
latency        = 6*10^(-3)*2;                             %Seconds
timesPerSecond = 4;                                       %Seconds
packagePeriod  = TransmitPeriod + ReceivePeriod + latency;%Seconds
os_time        = 0.05; %-os = overshoot                    %Seconds

sleep_Period    = 1-packagePeriod*timesPerSecond; %Sleep period for one package every
second
sleep_Period_OS = sleep_Period-os_time*timesPerSecond;
R = sqrt(3);                                        %Measurement Shunt-Resistance
%R has been redefined so that it will give sense

I_tx_max  = 17.5*10.^(-3);                  %Transmit current
I_tx_min  = 8*10.^(-3);                     %I_tx_max - 24dBm
I_rx      = 23.0*10.^(-3);                  %Receive Current
I_sleep   = 1*10.^(-6);                     %Sleep current

Ptx_max   = I_tx_max*R^(2) * TransmitPeriod; %Max Transmit Power
Ptx_min   = I_tx_min*R^(2) * TransmitPeriod; %Min Transmit Power
Prx_no_sleep = I_rx*R^(2) * (ReceivePeriod + latency + sleep_Period);%Receive Power
with no sleep
Prx_sleep    = I_rx*R^(2) * (ReceivePeriod + latency);              %Receive Power
with sleep
Ptsleep      = I_sleep*R^(2) * sleep_Period;   %Sleep Power
Ptsleep_OS   = I_sleep*R^(2) * sleep_Period_OS;%Sleep Power minus overshoot time
Ptrx_OS      = I_rx*R^(2) * os_time*timesPerSecond;%Sleep Power minus overshoot time

%% Overshoot overshoot setup and power calculations
os_Persentage = 0.4;
os_Top_max    = Ptx_max * os_Persentage;
os_Top_min    = Ptx_min * os_Persentage;
os_center     = os_time/2;%sec

x = [0:.001:os_time];
f = Ptx_max + os_Top_max * exp(-((x-os_center).^(2)/0.0001));
fun_max = @(x) os_Top_max * exp(-((x-os_center).^(2)/0.0001));
OS_power_max = integral(fun_max,0,os_time);
fun_min = @(x) os_Top_min * exp(-((x-os_center).^(2)/0.0001));
OS_power_min = integral(fun_min,0,os_time);

figure(1)
plot(x,f);
title('Gaussian distribution of a voltage peak after node wakeup')
xlabel('time(sec)') % x-axis label
ylabel('Voltage over 1 Ohm ') % y-axis label
%% Total power and lifetime calculations

P_Total_max_no_sleep = Ptx_max + Prx_no_sleep;%Power without the sleep overshoot
power
```

```
P_Total_min_no_sleep = Ptx_min + Prx_no_sleep;

P_Total_max_sleep = Ptx_max + Prx_sleep + Ptsleep;%Power without the sleep overshoot power
P_Total_min_sleep = Ptx_min + Prx_sleep + Ptsleep;

P_Total_max_sleep_OS = Ptx_max + Prx_sleep + Ptsleep_OS + OS_power_max + Ptrx_OS;%Power without the sleep overshoot power
P_Total_min_sleep_OS = Ptx_min + Prx_sleep + Ptsleep_OS + OS_power_min + Ptrx_OS;

% AA Battery https://en.wikipedia.org/wiki/AA_battery "RAM"
V  = 1.5;
Ah = 2*2600*10^(-3); %mAh for two batteries
BatteryPower = V*Ah;

%Lifetime results PT(Power Time)
PT_Total_max_no_sleep = BatteryPower/P_Total_max_no_sleep;%Lifetime in hours
PT_Total_max_no_sleep = PT_Total_max_no_sleep/2                    %Halfpower Lifetime in hours
PT_Total_min_no_sleep = BatteryPower/P_Total_min_no_sleep;%Lifetime in hours
PT_Total_min_no_sleep = PT_Total_min_no_sleep/2                    %Halfpower Lifetime in hours
PT_Total_max_sleep    = BatteryPower/P_Total_max_sleep;   %Lifetime in hours
PT_Total_max_sleep    = PT_Total_max_sleep/2                    %Halfpower Lifetime in hours
PT_Total_min_sleep    = BatteryPower/P_Total_min_sleep;   %Lifetime in hours
PT_Total_min_sleep    = PT_Total_min_sleep/2                    %Halfpower Lifetime in hours
PT_Total_max_sleep_OS = BatteryPower/P_Total_max_sleep_OS;%Lifetime in hours
PT_Total_max_sleep_OS = PT_Total_max_sleep_OS/2                    %Halfpower Lifetime in hours
PT_Total_min_sleep_OS = BatteryPower/P_Total_min_sleep_OS;%Lifetime in hours
PT_Total_min_sleep_OS = PT_Total_min_sleep_OS/2                    %Halfpower Lifetime in hours


%% Train scenario result, power consumption (no sleep)
Timer_train_between_send = 0.070;%s - Time between each initialisation of a send package
P_Train_send_per_package = (I_tx_min * packagePeriod + I_rx (Timer_train_between_send - packagePeriod))/360;% Power used per package send

% Assuming a cost only for each package send
P_scenario_1 = (29148 + 4066 + 3515) * 4 * P_Train_send_per_package
P_scenario_2 = (29838 + 8155 + 8499) * 4 * P_Train_send_per_package
P_scenario_3 = (29258 + 3706 + 4021) * 4 * P_Train_send_per_package
P_scenario_4 = (30523 + 1232 + 8425) * 4 * P_Train_send_per_package

% Percentage of power used by base station
P_basestation_scenario_1 = (29148 * 4 * P_Train_send_per_package) / Ah
P_basestation_scenario_2 = (29838 * 4 * P_Train_send_per_package) / Ah
P_basestation_scenario_3 = (29258 * 4 * P_Train_send_per_package) / Ah
P_basestation_scenario_4 = (30523 * 4 * P_Train_send_per_package) / Ah
```