OPTIMISATION OF POWER CONSUMPTION IN WIRELESS NETWORK

A Project Report Submitted
in Partial Fulfilment of the Requirements
for the Degree of

BACHELOR OF TECHNOLOGY

in

Mathematics and Computing

by

Vishal Keshav

(Roll No. 11012341)

to the

DEPARTMENT OF MATHEMATICS

INDIAN INSTITUTE OF TECHNOLOGY GUWAHATI

GUWAHATI - 781039, INDIA

April 2015

CERTIFICATE

This is to certify that the work contained in this project report entitled "**Optimisation of power consumption in wireless network**" submitted by **Vishal Keshav** (**Roll No.: 11012341**) to Indian Institute of Technology Guwahati towards partial requirement of **Bachelor of Technology** in Mathematics and Computing has been carried out by him under my supervision and that it has not been submitted elsewhere for the award of any degree.

Guwahati - 781 039 (Dr. N. Selvaraju)

April 2015 Project Supervisor

ABSTRACT

The main aim of the project is to study and apply Queueing Theory in modelling the wireless network node and create a power consumption function along with user specified constraints. Power consumption function is minimized satisfying the constraint using a numerical optimization technique. Optimised parameters are then tested on performance metrics developed.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

A wireless system can be differentiated into three segments which are Wireless LANs, Wireless PANs and Wireless Sensor Network(WSN). While LANs and PANs are equally important, much of the research focus has been put into Wireless Sensor Networks where a large number of tiny low cost, low power sensing devices are spreaded over the area used for extracting, processing and transmitting data to neighbour nodes, to further route it to central node. Wide applications of WSNs, from environmental monitoring to data logging, makes it a important subject to study and analyse. Recent inclination towards Internet of Thing(IoT) where most of the things are given computational capabilities along with connectedness has a close resemblance with WSN, which further makes it important area to approach.

One of the drawbacks associated with WSN is the limited availability of of power. Generally, nodes in the network are battery powered with limited lifetime and it has become a difficult task to make a replacement. For this reason, increasing the operational lifetime of sensor nodes has becomes one of the main design issues for WSN.

With the objective to minimize the energy consumption of a wireless sensor node, in this project we model the node mathematically and try to obtain optimal parameters that minimize the power cost function. In the second chapter, we start with the overview of M/M/1 queue and further extend it to describe N-policy M/M/1 queue, on which nodes in WSN are modelled[1, 3, 6]. We describe three approach to solve steady state probability equations for M/M/1 queue, putting more emphasis on moment generating function approaches. The moment generating function approach is used to solve steady state probability equation for N-policy M/M/1 queue. Important parameters of system such as average number of packets in queue and average time spent in the queue have been derived[1].

The chapter 3 deals with the formulation of energy consumption cost function and a related Quality of Service(QoS) constraint[3, 5]. Numerical techniques have been employed to optimize the cost function under the constraint[2] in order to find optimal value of associated variables of the system parameters. We take a particular example and show the percentage improvement over ordinary method. The last chapter presents the conclusion and future improvement possibility in this topic.

Chapter 2

Background

In order to analyse a node in wireless network, its important to construct a mathematical model that mimics its working. In addition to that, the model gives us the properties of a node in terms of parameter that becomes an integral part to further study the node in greater detail. We have modelled our node on queuing system. So, in this Chapter we give a brief description of queuing theory.

2.1 M/M/1 Queue

A queuing system is basically characterised by Arrival process, Departure Process and the number of servers that can serve at a time. A mathematical representation of queuing system is given in Kendall's notation as follow:

$$A/S/c \tag{2.1}$$

where **A** denotes the arrival process **S** denotes the service time distribution and **c** is the number of servers

Arrival Process and Servicing Process describes the probability density distribution that determines the packet arrivals in the system and service time in the system respectively. **A** and **S** can be Markov(**M**) i.e. describing exponential probability density, Deterministic(**D**) i.e. all packet takes same amount of time or General(**G**).

In this section, we will analyze **M/M/1** queue having exponential inter-arrival times with mean λ, exponential service times with mean μ and a single server. Packets are processed in order of arrival time. This queueing system can be applied to a wide variety of problems as any system with a very large number of independent customers can be approximated as a Poisson process.

We say the system is in state k if $N(t) = k$ where $N(t)$ is the number of packets in the system at time t. Since all the random variables involved is exponentially distributed, $N(t)$ have memory less property and is continuous time Markov chain with state space 0,1,2... . We denote $p_k(t)$ as probability that system has k number of packets at time t and $p_{ij}(\Delta t)$ as transition probability from state i to j in time Δt.

$$\begin{aligned} p_{k,k+1}(\Delta t)) = & (\lambda \Delta t + O(\Delta t))(1 - (\mu \Delta t + O(\Delta t))) \\ & + \sum_{k=2}^{\infty} (\lambda \Delta t + O(\Delta t))^k (\mu \Delta t + O(\Delta t))^{k-1} \end{aligned} \tag{2.2}$$

Neglecting higher order terms in the above equation, transition probability is re-written as

$$p_{k,k+1}(\Delta t)) = (\lambda \Delta t + O(\Delta t)) \tag{2.3}$$

Similarly,

$$p_{k,k-1}(\Delta t)) = (\mu \Delta t + O(\Delta t)) \tag{2.4}$$

Based on memory less property, state probability is given by

$$p_0(t + \Delta t) = (1 - \lambda \Delta t) p_0(t) + \mu \Delta t p_1(t) + O(\Delta t)$$

$$p_k(t + \Delta t) = \lambda \Delta t p_{k-1}(t) + (1 - (\lambda + \mu)\Delta t) p_k(t) + \mu \Delta t p_{k+1}(t) + O(\Delta t)$$

The above equation in differential form is given by

$$p_0'(t) = -\lambda p_0(t) + \mu p_1(t)$$
$$p_k'(t) = \lambda p_{k-1}(t) - (\lambda + \mu)p_k(t) + \mu p_{k+1}(t)$$

Since we are interested in the system at equilibrium condition i.e. at $t \to \infty$, differential terms becomes zero and we obtain the system equation at equilibrium condition given by

$$-\lambda p_0(t) + \mu p_1(t) = 0 \tag{2.5}$$

$$\lambda p_{k-1}(t) - (\lambda + \mu)p_k(t) + \mu p_{k+1}(t) = 0 \tag{2.6}$$

Before solving (2.5) and (2.6), we describe some assumptions on system at equilibrium condition. Traffic intensity ρ defined by $\rho = \frac{\lambda}{\mu}$ must be smaller than 1 otherwise system would be overloaded. Also, p_k should satisfy normalization condition given by $\sum_{k=0}^{\infty} p_k = 1$.

2.1.1 Equilibrium state solution

There are different approach to solve (2.5) and (2.6), but we describe three of them.

1. Direct Approach: Since the equation (2.6) is a second order recurrence relation, the general solution will be of the form $p_n = c_1 x_1^n + c_2 x_2^n$ where x_1 and x_2 are the roots of quadratic equation $\lambda - (\lambda + \mu)x + \mu x^2 = 0$. Substituting the roots $x = 1$ and $x = \frac{\lambda}{\mu}$ in general solution of p_n, we get $p_n = c_1 + c_2\rho^n$. Applying normalization condition and equation (2.5), solution we get is $p_n = (1 - \rho)\rho^n$ where ρ is traffic intensity as defined earlier.

2. Recursion: p_1 can be obtained in terms of p_0 using equation (2.5) as $p_1 = p_0\rho$. Using equation (2.6) and p_1, p_2 can be obtained as $p_2 = p_0\rho^2$. In general, p_n is given by $p_n = p_0\rho^n$. Using normalization condition, we get the solution $p_n = (1-\rho)\rho^n$.

3. Generating function approach: Probability generating function of number of packets in the system denoted by L is given by

$$P_L(z) = \sum_{n=0}^{\infty} p_n z^n \tag{2.7}$$

where $|z| \leq 1$ We multiply n^{th} state equilibrium equation to z^n and sum the equation over n, the equilibrium equation can be transformed in terms of $P_L(z)$ as given below

$$\mu p_0(1 - z^{-1}) + (\lambda z + \mu z^{-1} - (\lambda + \mu))P_L(z) = 0 \tag{2.8}$$

The solution of equation is given by

$$P_L(z) = \sum_{n=0}^{\infty} (1-\rho)\rho^n z^n \tag{2.9}$$

Equating the coefficient of moment generating function with p_n yields us the equilibrium solution.

2.1.2 Performance measures of the system

In this subsection, we derive some important performance parameters of the system.

1. Mean number of packets in the system: It is the average number of packets waiting to be processed and packets under processing. We

denote it by $\overline{N}$.

$$\overline{N} = E(L) = \sum_{n=0}^{\infty} np_n = \frac{\rho}{1-\rho} \tag{2.10}$$

2. Mean time spent in the system: It is the average time spent by a packet in the system once entered.

$$E(S) = \frac{E(L)}{\lambda} = \frac{1/\mu}{1-\rho} \tag{2.11}$$

3. Mean number of waiting packets: This excludes the packets under process in the system. We denote it by $\overline{Q}$.

$$\overline{Q} = E(L - \rho = \sum_{n=0}^{\infty} n - 1p_n = \frac{\rho^2}{1-\rho} \tag{2.12}$$

4. Mean waiting time: It is the average time a packet spends in the system waiting to be precessed.

$$E(W) = E(S) - 1/\mu = \frac{\rho/\mu}{1-\rho} \tag{2.13}$$

5. Server utilization: It gives the fraction of time server remains busy processing packets in the system. We denote it by U_s.

$$U_s = 1 - p_0 = \rho \tag{2.14}$$

2.2 N-policy M/M/1 Queue

Up till now, we have discussed the basic M/M/1 queue. But our analysis of a sensor network is based on N-policy M/M/1 queue which can thought of as an extension of basic M/M/1 queue. The main difference between the two lies on how server process data packets present in the buffer. In N-policy M/M/1 queue, server stops processing data packets until the number of packets reaches a specified value N. Once it starts processing, server serves the packets exhaustively i.e. keeps on processing packets until there is no packet left in the system. Once, there is no packet left to serve, server node start to accumulate N packets before going to busy state.

So basically, a sensor node switches between two state which are Idle state and Busy state. A node remain in one of the two states i.e. Idle state and Busy state in addition to a state described by the number of packets in the system. Mathematically, a state of the system is described by (i, n) where $i = 0, 1$ represents Idle and Busy state respectively and $n = 0, 1, 2, 3, ...$ represents the number of packets present in the system. Similar to M/M/1 queue, we assume packet arrival follow a Poisson process with mean arrival rate λ for a generic sensor node and the service times are exponentially distributed with mean $1/\mu$.

For a fixed N, we denote $P_0(n)$ as probability of being in Idle state and with n number of packets in the system. Value of n in Idle state can range from 0 to $N - 1$. Similarly, we denote probability of system of being in Busy state with n number of packets by $P_1(n)$. In case of Busy state, n can range from 1 to ∞.

Similar to equation (2.5) and (2.6) as in the case of M/M/1 queue, steady state equation for $P_0(n)$ and $P_1(n)$ can be derived and is given as follow:

$$\lambda P_0(0) = \mu P_1(1) \tag{2.15}$$

$$\lambda P_0(n) = \lambda P_0(n-1), 1 \leq n \leq N-1 \tag{2.16}$$

$$(\lambda + \mu) P_1(1) = \mu P_1(2) \tag{2.17}$$

$$(\lambda + \mu) P_1(n) = \lambda P_1(n-1) + \mu P_1(n+1), 2 \leq n \leq N-1 \tag{2.18}$$

$$(\lambda + \mu) P_1(N) = \lambda P_0(N-1) + \lambda P_1(N-1) + \mu P_1(N+1) \tag{2.19}$$

$$(\lambda + \mu) P_1(n) = \lambda P_1(n-1) + \mu P_1(n+1), n \geq N+1 \tag{2.20}$$

The above set of equation is difficult to solve using recursive approach. So, we use Generating function approach to obtain a closed form expression for probability of being in a state. Let us denote I, B and L the number of packets in the system during Idle state, Busy state and under N-policy M/M/1 queue respectively. We define moment generating function for these three different case as follow:

$$G_I(z) = \sum_{n=0}^{N-1} z^n P_0(n), |z| \leq 1 \tag{2.21}$$

$$G_B(z) = \sum_{n=1}^{\infty} z^n P_1(n), |z| \leq 1 \tag{2.22}$$

$$G_L(z) = G_I(z) + G_B(z) \tag{2.23}$$

Using equation (2.16), $G_I(z)$ can be re-written in terms of $P_0(0)$.

$$G_I(z) = P_0(0) \sum_{n=0}^{N-1} z^n = \frac{1 - z^N}{1 - z} P_0(0) \tag{2.24}$$

Multiplying z to (2.16) and z^n to (2.17) - (2.20) for $n = 2, 3, 4, ...$ and adding all equation term by term for all possible value of n, we obtain the following:

$$G_B(z) = \frac{\rho z(1 - z^N)}{\rho z^2 - (1 + \rho)z + 1} P_0(0) \tag{2.25}$$

Combining $G_I(z)$ and $G_B(z)$, we obtain

$$G_L(z) = G_I(z) + G_B(z) = \frac{1 - z^N}{(1 - \rho z)(1 - z)} P_0(0) \tag{2.26}$$

With $z = 1$ in $G_L(z)$, we get normalization condition and L'Hospital's rule

$$G_L(1) = G_I(1) + G_B(1) = \sum_{n=0}^{N-1} P_0(n) + \sum_{n=1}^{\infty} P_1(n) = \lim_{z \to 1} G_L(z) = \frac{N}{1 - \rho} P_0(0) = 1 \tag{2.27}$$

Thus the probability of having no packet in system and system being in Idle state is given by $P_0(0) = \frac{1-\rho}{N}$.

We denote P_I and P_B as probability of being in Idle and Busy state respectively. These probabilities can be derived easily using moment generating function. $P_I = G_I(1) = \lim_{z\to 1} \frac{1-z^N}{1-z} P_0(0) = NP_0(0) = 1 - \rho$ and $P_B = G_B(1) = \lim_{z\to 1} \frac{\rho z(1-z^N)}{(1-\rho)(1-z)} P_0(0) = \rho$.

2.2.1 Performance measure of the system

In this subsection, we derive some important performance parameters of N-policy M/M/1 queue.

1. Expected number of packets in Idle state:

$$E(I) = \sum_{n=0}^{N-1} nP_0(n) = \frac{1 - \rho}{N} \frac{N(N - 1)}{2} = \frac{(N - 1)(1 - \rho)}{2} \tag{2.28}$$

2. Expected number of packets in Busy state:

$$E(B) = \sum_{n=0}^{\infty} nP_1(n) = G_B(1)' = \lim_{z \to 1} G_B(1)' = \frac{N\rho(1-\rho+\rho(1+\rho))}{2(1-\rho)} \tag{2.29}$$

3. Expected number of packets in N-policy M/M/1 queue: Combining the two expressions above we obtain:

$$E(L) = E(I) + E(B) = \frac{N-1}{2} + \frac{\rho}{1-\rho} \tag{2.30}$$

4. Idle Period: It is the length of time per cycle when server is Idle with waiting packet number is less than N. We denote it by T_I. It is the sum of N exponential random variable with mean $1/\lambda$ and thus given by:

$$E(T_I) = N/\lambda \tag{2.31}$$

5. Busy Period: It is the length of time per cycle when server is busy and transmitting data packet. We denote it by T_B. Since $E(T_B)/E(T) = 1-\rho$ where $E(T) = E(T_I) + E(T_B)$, $E(T_B)$ can be derived and is given by following

$$E(T_B) = \frac{N}{\mu(1-\rho)} \tag{2.32}$$

6. Busy Cycle: It is the time difference between two consecutive starts of Idle period. We denote it by T.

$$E(T) = \frac{N}{\lambda(1-\rho)} \tag{2.33}$$

Chapter 3

The Power Function

In the previous chapter, we have discussed the mathematical model for sensor node in wireless networks in great detail. This chapter deals with the construction of Energy consumption function using the performance measure derived for N-policy M/M/1 queue. We then try to obtain optimal value of N and service rate μ by minimizing the cost function under a constraint. We then describe a performance measurement metrics PCIF which is essential in order to measure the optimality achieved with optimal value of N and μ.

3.1 Cost Function formulation

We make following assumptions before constructing energy consumption function.

1. Fixed energy consumption is incurred per busy cycle in switching between Idle and Busy state and vice verse.

2. Energy is consumed for retaining the data packets present in the system.

3. Constant amount of energy is consumed per Busy and Idle period in one Busy cycle.

4. Server has the flexibility of increasing servicing rate but at the expense of increased energy consumption[8]. Linear model is taken into consideration for simplicity.

Lets us denote C_h, C_s, C_{id}, C_b and P_s as holding power for each packet in the system, setup energy per busy cycle, energy consumed to keep server in idle state, energy consumed to keep server in busy state and data processing power in one busy cycle respectively.

Using the above notation, we construct average cost function as follow:

$$F(N,\mu) = C_h E(L) + \frac{C_s}{E(T)} + C_{id}\frac{E(T_I)}{E(T)} + C_b\frac{E(T_B)}{E(T)} + P_s g(\mu) \tag{3.1}$$

The above function has two variables, N which is the number of packets the server accumulate before processing exhaustively and μ which is the mean service rate or average data processing rate of the server. Using equation (2.30), (2.33), (2.31), (2.32) and linear dependency on power consumption over service rate, equation (3.1) can be re-written as follow:

$$F(N,\mu) = C_h\left(\frac{N-1}{2} + \frac{\rho}{1-\rho}\right) + C_s\frac{\lambda(1-\rho)}{N} + C_{id}(1-\rho) + C_b\rho + P_s\mu \tag{3.2}$$

where $\rho = \frac{\lambda}{\mu}$.

We need to minimize the cost function in order to find optimal value of N and μ. But we also need to consider the user's Quality of Service needs. Since a bound on mean waiting time in the queue, denoted by W_q is an important QoS requirement, we take it as a constraint while minimizing power cost function.

Derivation of mean waiting time W_q at steady state is given as follow:

$$W_q = \frac{E(L')}{\lambda} \tag{3.3}$$

where $E(L')$ is expected number of packets waiting in the system.

$$E(L') = E(L) - \rho = \frac{N-1}{2} + \frac{\rho}{1-\rho} - \rho = \frac{N-1}{2} + \frac{\rho^2}{1-\rho} \tag{3.4}$$

Using equation (3.4) in (3.3), we obtain average delay per packet

$$W_q = \frac{N-1}{2\lambda} + \frac{\rho}{\mu - \lambda} \tag{3.5}$$

3.2 Constrained Power Optimization

Having constructed the cost function, we want to minimize it under the the constraint of mean delay of a packet in the system. Mathematically, we want to solve the following constrained problem

$$\begin{aligned} &\text{minimize} \quad F(N, \mu) \\ &\text{subject to} \quad W_q \leq D \end{aligned} \tag{3.6}$$

where D is the delay.

We could use KKT to solve above of inequality constrained 2 variable function, but obtaining a closed form solution is not easy. This is because of the complexity and non-linearity of cost function. So, we approach the solution of problem through numerical optimization techniques[7].

To obtain the optimal value (N^*, μ^*), we break the problem in two steps. In step 1, we minimize the cost function without constraint in an iterative way.

$$x_{k+1} = x_k + \alpha_k p_k \tag{3.7}$$

where x_0 is an initial guess, α_k is the step length and p_k is the direction of movement in search for the minimum $F(.)$. Here, we represent x_k as a vector $[N, \mu]^T$ at k^{th} iterate. We can choose $p_k = -\nabla F_k / \|\nabla F_k\|$ which is steepest decent direction. We can use Backtracking Line Search algorithm to find an appropriate value of α at every iterate. We give the algorithm below.

Result: return α
Choose $\alpha > 0$, $\rho \in (0, 1)$, $c \in (0, 1)$;
while $F(x_k + \alpha \cdot p_k) \leq F(x_k) + c \cdot \alpha \nabla F_k \cdot p_k$ **do**
    $\alpha \leftarrow \rho \cdot \alpha$;
end

Algorithm 1: Backtracking Line Search

Number of iterates depend on precision required. Once we arrive at an optimal (N^*, μ^*), we check if it lies in the feasible region i.e. $W_q(N, \mu) \leq D$. If it lies in the feasible region, we terminate the process with optimal value (N^*, μ^*), otherwise, we move to step 2.

In step 2, we include inequality constraint $W_q \leq D$ as an equality constraint $W_q = D$ in our optimization problem. Observing that N can be represented in terms of D and μ, we could use elimination, substituting $N(D, \mu)$ in $F(N, \mu)$ to get a new cost function $f(D, \mu)$ which we need to minimize. Here we fix D and minimize the new cost function $f(\mu)$ using steep decent minimization. The optimal value μ^* is used to find out corresponding N^* thus giving us a complete optimal solution.

We describe below the complete algorithm:

Result: Output N^* and μ^*

Choose C_h, C_s, C_{id}, C_b, P_s, λ and D;

Take initial guess $x_0 = [N_0, \mu_0]^T$;

$p = -\nabla F_0 / \|\nabla F_0\|$, $\alpha = BacktrackingLineSearch(p)$;

$x_n = x_0 + \alpha \cdot p$;

while $\|\nabla F(x_n) - F(x_0)\| \geq tol$ **do**

  $x_0 = x_n$;

  $p = -\nabla F_k / \|\nabla F_k\|$;

  $\alpha = BacktrackingLineSearch(p)$;

  $x_n = x_0 + \alpha \cdot p$;

end

if $W_q(x_n) \leq D$ **then**

  $N^* = x_n(1)$, $\mu^* = x_n(2)$;

else

  Make initial guess μ_0;

  Choose α;

  $\mu_n = \mu_0 - \alpha \cdot \frac{f_0'}{\|f_0'\|}$;

  while $\|\nabla f(\mu_n) - f(\mu_0)\| \geq tol$ **do**

    $\mu_0 = \mu_n$;

    $\mu_n = \mu_0 - \alpha \cdot \frac{f_k'}{\|f_k'\|}$;

  end

  $N^* = N(\mu_n)$, $\mu^* = \mu_n$;

end

Algorithm 2: Algorithm to find optimal solution

We demonstrate the working of the above algorithm by setting the system parameters to a fixed value. We set C_h = 2, C_s = 20, C_{id} = 4, C_b = 5, P_s = 20 and λ = 2.5. With an initial guess of $[N_0, \mu_0]^T = [5, 5]$ and for five different values of Delay D, we present the surface plot of power function, contour plot along with constraint curve and results tabulated below.

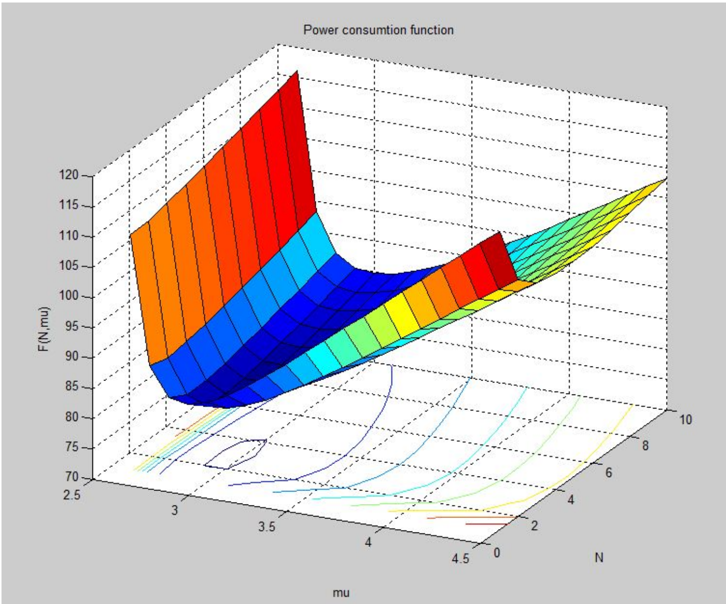

Figure 3.1: Surface plot of $F(N, \mu)$

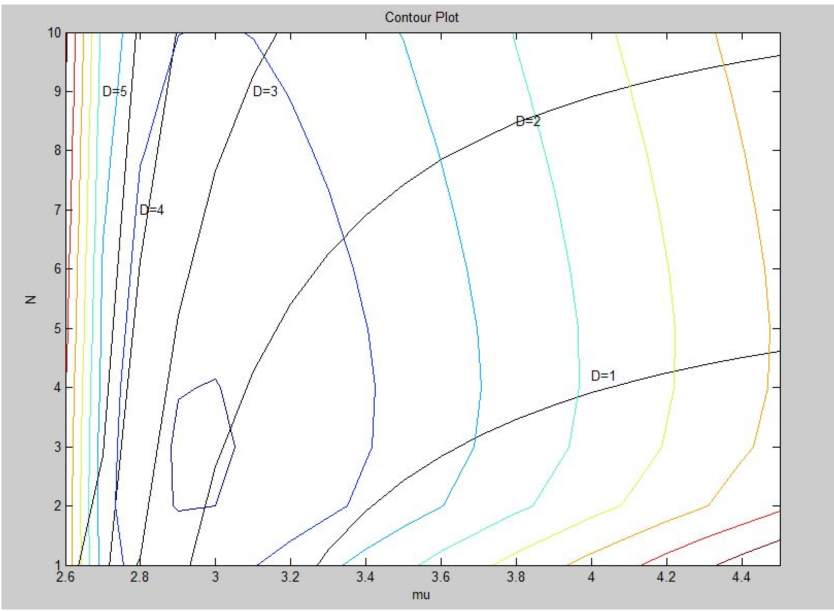

Figure 3.2: Contour plot between N and μ

	N^*	μ^*	$F(N^*, \mu^*)$
$D = 1$	1.7611	3.3740	86.0574
$D = 2$	2.5843	2.9958	79.6215
$D = 3$	2.7588	2.9484	79.4818
$D = 4$	2.7588	2.9484	79.4818
$D = 5$	2.7588	2.9484	79.4818

Table 3.1: Optimal parameters for different delay

3.3 Performance Measurement Metrics

In this section, we describe a way to measure the improvement over power consumption by substituting the optimal output from the algorithm given in previous section. We define Power Consumption Improvement Factor(PCIF) given by the expression:

$$PCIF = \frac{F(1, \mu_{min}) - F(N^*, \mu^*)}{F(1, \mu_{min})} * 100\% \tag{3.8}$$

where $F(1, \mu_{min})$ represents the ordinary cost function with $N = 1$ and minimum average servicing rate. $F(N^*, \mu^*)$ is the cost value obtained after minimization.

Using the same set of constants given in previous section, we tabulate the improvement percentage below

	$F(N^*, \mu^*)$	$F(1, \mu_{min})$	PCIF
$D = 1$	86.0574	108.8846	20.9646%
$D = 2$	79.6215	108.8846	26.8753%
$D = 3$	79.4818	108.8846	27.0037%
$D = 4$	79.4818	108.8846	27.0037%
$D = 5$	79.4818	108.8846	27.0037%

Table 3.2: Percentage improvement over energy minimization

We now do analysis of our method of improving power consumption. We vary Delay and λ keeping one fixed at a time to show the effect of restriction and load on system with Power consumption function and corresponding improvement percentage.

Below is the graph between cost functions with varying delay constraint at $\lambda = 2.5$. In the graph below, we observe that as we lower the user's QoS need, we achieve lower power consumption.

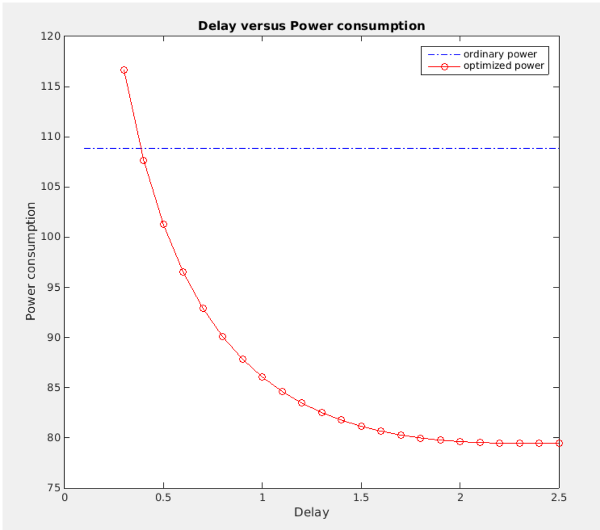

Figure 3.3: Plot between Delay D and cost function $F(.)$

To conclude, we show graph between D and improvement % below

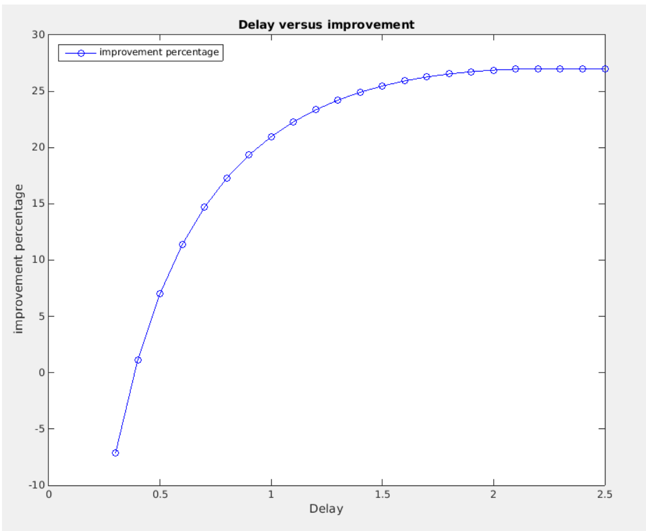

Figure 3.4: Plot between Delay D and improvement % $PCIF$

Next, we demonstrate the dependency between system load λ and cost function $F(.)$ at delay $D \leq 2$ when power function remain constrained.

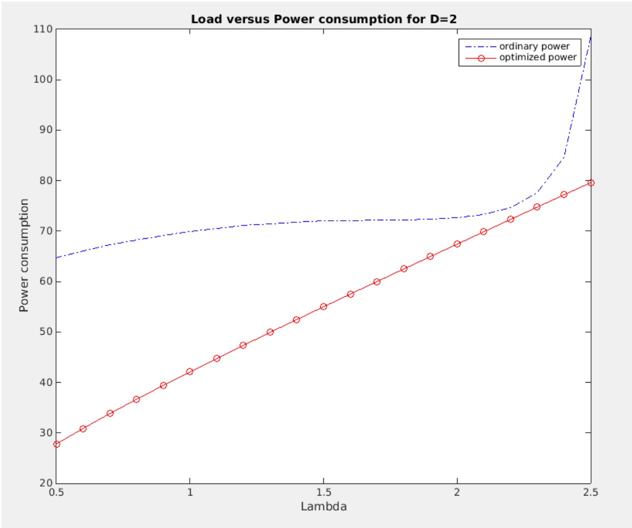

Figure 3.5: Plot between λ and cost function $F(.)$ at $D \leq 2$

With $D \leq 4$, optimized solution of cost function remain in feasible region, hence constraint have no meaning. Corresponding graph is given below

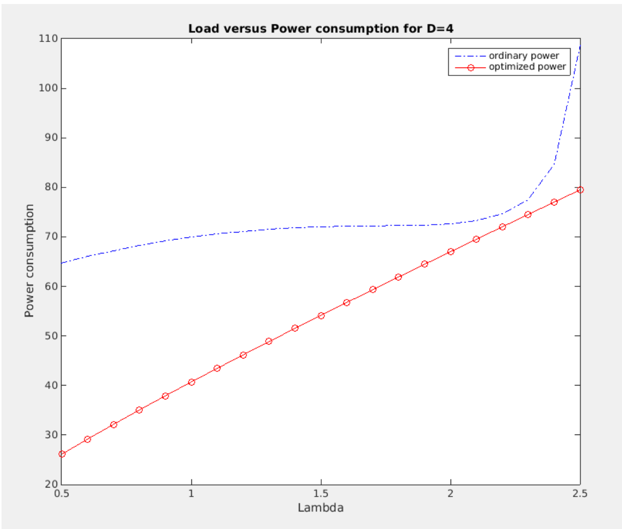

Figure 3.6: Plot between λ and cost function $F(.)$ at $D \leq 4$

Corresponding load versus improvement % graph is shown below

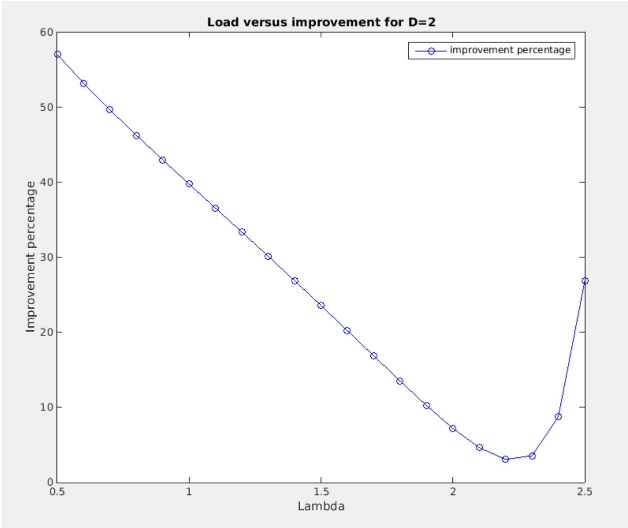

Figure 3.7: Plot between λ and improvement % $PCIF$ at $D \leq 2$

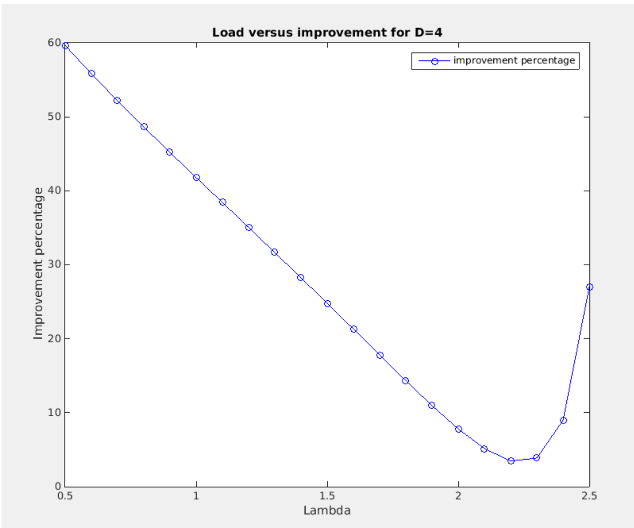

Figure 3.8: Plot between λ and improvement % $PCIF$ at $D \leq 4$

From the two plots above, it can be readily seen that, on increasing λ

i.e. increasing the average number of incoming packet rate, improvement % decreases up to a certain level and then again starts to increase till the system reaches in unstable condition. But, here we note that, even it decreases, it remains positive. From this we conclude that, there is an optimal point during system load when improvement goes to its lowest level.

Chapter 4

Conclusion

In this project, we used the N-policy M/M/1 queue as one of the approach to model and minimize energy consumption in wireless sensor network node. We included an important QoS constraint which is mean delay per packet along with the cost function. We stated all the assumptions on which we have constructed the multi dimensional cost function. We proposed numerical optimization techniques to solve out constrained problem. Later, the improvements factor over ordinary M/M/1 queue was given. With particular constants in energy consumption function, we tabulated the improvement percentage. At the end, further analysis was done varying other system parameters such as λ and Delay D.

Further research work in this direction could be inclusion of a different yet important QoS constraint such as temporal accuracy in data processing and energy efficiency. Better numerical techniques with guaranteed and faster convergence rate such as solution based on quadratic model can be employed as a future work.

Bibliography

[1] Fuu-Cheng Jiang, Der-Chen Huang, Chao-Tung Yang, and Fang-Yi Leu. Lifetime elongation for wireless sensor network using queue-based approaches. *The Journal of Supercomputing*, 59(3):1312–1335, 2012.

[2] Fuu-Cheng Jiang, Chao-Tung Yang, Kuo-Hsiung Wang, and Der-Chen Huang. Design framework to optimize power consumption and latency delay for sensor nodes using min (n, t) policy m/g/1 queuing models. In *Future Information Technology (FutureTech), 2010 5th International Conference on*, pages 1–8. IEEE, 2010.

[3] R Maheswar and R Jayaparvathy. Optimal power control technique for a wireless sensor node: a new approach. *Int. J. Comput. Electric. Eng*, 3(1):1793–8163, 2011.

[4] Jorge Nocedal and Stephen Wright. Numerical optimization, series in operations research and financial engineering. *Springer, New York, USA*, 2006.

[5] Aladdin Sleem, Ming Bao, and Wei W Li. Analyzing the impact of the n-policy on quality of service in wireless sensor networks. In *Networking, Sensing and Control (ICNSC), 2014 IEEE 11th International Conference on*, pages 150–154. IEEE, 2014.

[6] Kuo-Hsiung Wang, Shu-Lung Chuang, and Wen-Lea Pearn. Maximum entropy analysis to the n policy m/g/1 queueing system with a removable server. *Applied Mathematical Modelling*, 26(12):1151–1162, 2002.

[7] Stephen J Wright and Jorge Nocedal. *Numerical optimization*, volume 2. Springer New York, 1999.

[8] Dequan Yue, Yan Zhang, and Wuyi Yue. Optimal performance analysis of an m/m/1/n queue system with balking, reneging and server vacation. *International Journal of Pure and Applied Mathematics*, 28(1):101–115, 2006.