CRUCIAL EDGE DETECTION IN SENSOR SYSTEMS UNDER
ENERGY CONSTRAINTS

A Thesis by
Ashok Madhvesh
Bachelor of Engineering, Visvesvaraya Technological University, 2005

Submitted to the Department of Electrical Engineering and Computer Science
and the faculty of the Graduate School of
Wichita State University
in partial fulfillment of
the requirements for the degree of
Master of Science

December 2009
CRUCIAL EDGE DETECTION IN SENSOR SYSTEMS UNDER ENERGY CONSTRAINTS

The following faculty members have examined the final copy of this thesis for form and content, and recommend that it be accepted in partial fulfillment of the requirement for the degree of Master of Science with a major in Electrical Engineering.

Neeraj Jaggi, Committee Chair

M.Edwin Sawan, Committee Member

Hamid Lankarani, Committee Member
ACKNOWLEDGEMENTS

Sincere thanks and gratitude to my research advisor, Dr. Neeraj Jaggi, for his valuable, timely, and appropriate advice throughout my progress with this research work. Being under his guidance as one of his first students, it was a total pleasure to work with and learn from. I wish to carry forward the knowledge and values imparted by him into my career. I also convey my regards to Dr. M.Edwin Sawan, Dr. Ravi Pendse, and Dr. Vinod Namboodiri for their contributions in my development and successful completion of the masters program.

I must mention my research associates, Natarajan, Sandeep, and few others, for sharing their ideas and thoughts selflessly, which definitely has added value. For more than two years of my education I have been very grateful to Wichita State University and all the Shockers who made me feel at home. Other than education, the fun-filled life in Wichita among my many friends has been priceless and forever cherished. My roommates and other friends on Roosevelt street in Wichita made it a very memorable educational experience.

Above all, the love shown and principles taught by my parents and late grandmother helped me complete the masters program on a high note. I am also grateful to my relatives and friends who have always been supportive. Last but not least, I appreciate dear Pooja for the sincere affection, love, and support she has displayed.
ABSTRACT

Wireless sensor nodes are usually deployed in remote locations for various applications that require monitoring of certain interesting events. Due to this remote operational feature the longevity of the sensor node’s lifetime has been a primary concern. Although the sensor nodes available today may be equipped with rechargeable batteries, the minimal energy capacity of such batteries and low recharge rates degrade the sensor’s lifetime and achievable performance. Hence, operational algorithms are needed to guarantee high performance with efficient utilization of energy available. In this thesis, considering temporally correlated event phenomena, the important question answered is: “How long should the sensor sleep, and for how long should the sensor stay active?” To achieve this, a sensor activation/deactivation algorithm has been developed that achieves high performance with efficient energy utilization.

A sensor loses energy predominantly because of redundant transmissions of sensed data. To avoid this, a sensor was modeled to transmit only the changes sensed in the event-occurrence process, referred to as Crucial Edges or Transitions. In addition, the system model allows the transmission of transitions that are detected late. Several intuitive decision-making policies were compared and the results compared in order to determine the best policy for this problem. This policy was later analyzed using Markov chain analysis techniques to derive upper and lower bounds on the achievable performance. The proposed policy achieves high performance under energy balancing constraints, and is deterministic, simple and easy to implement on a sensor node.
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1 INTRODUCTION</strong></td>
<td>1</td>
</tr>
<tr>
<td>1.1 Sensor Characteristics</td>
<td>1</td>
</tr>
<tr>
<td>1.2 Issues with Sensor Operation</td>
<td>3</td>
</tr>
<tr>
<td>1.3 Correlated Nature of Event Occurrence</td>
<td>3</td>
</tr>
<tr>
<td>1.4 Decision-Making Policy/Algorithm</td>
<td>4</td>
</tr>
<tr>
<td>1.5 Contribution</td>
<td>7</td>
</tr>
<tr>
<td><strong>2 RELATED WORK</strong></td>
<td>9</td>
</tr>
<tr>
<td><strong>3 DISCRETE EVENT SIMULATION</strong></td>
<td>12</td>
</tr>
<tr>
<td>3.1 M/M/1 Queue</td>
<td>12</td>
</tr>
<tr>
<td>3.2 Sensor System Model</td>
<td>14</td>
</tr>
<tr>
<td>3.2.1 Simulation Algorithm</td>
<td>15</td>
</tr>
<tr>
<td><strong>4 PROBLEM FORMULATION</strong></td>
<td>16</td>
</tr>
<tr>
<td>4.1 Introduction</td>
<td>16</td>
</tr>
<tr>
<td>4.2 Classification of Sensor State</td>
<td>17</td>
</tr>
<tr>
<td>4.3 Event-Occurrence Process</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Recharge/Discharge Modeling</td>
<td>20</td>
</tr>
<tr>
<td>4.5 Sensor Activation Problem</td>
<td>21</td>
</tr>
<tr>
<td>4.5.1 Key Assumptions</td>
<td>22</td>
</tr>
<tr>
<td>4.5.2 Example of an Intuitive Policy</td>
<td>24</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.5.3</td>
<td>26</td>
</tr>
<tr>
<td>4.5.4</td>
<td>28</td>
</tr>
<tr>
<td>4.5.5</td>
<td>28</td>
</tr>
<tr>
<td>4.6</td>
<td>29</td>
</tr>
</tbody>
</table>

5 EVALUATED POLICIES

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>31</td>
</tr>
<tr>
<td>5.2</td>
<td>31</td>
</tr>
<tr>
<td>5.3</td>
<td>32</td>
</tr>
<tr>
<td>5.4</td>
<td>32</td>
</tr>
<tr>
<td>5.5</td>
<td>32</td>
</tr>
<tr>
<td>5.6</td>
<td>33</td>
</tr>
<tr>
<td>5.7</td>
<td>34</td>
</tr>
<tr>
<td>5.8</td>
<td>35</td>
</tr>
</tbody>
</table>

6 FOUR-TIMER POLICY

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.1</td>
<td>37</td>
</tr>
<tr>
<td>6.2</td>
<td>40</td>
</tr>
</tbody>
</table>
TABLE OF CONTENTS (continued)

<table>
<thead>
<tr>
<th>Chapter</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.2.1</td>
<td>Steady State Probability</td>
</tr>
<tr>
<td>6.2.2</td>
<td>Performance Evaluation</td>
</tr>
<tr>
<td>6.2.3</td>
<td>Upper Bound on $P(\Pi)$</td>
</tr>
<tr>
<td>6.2.4</td>
<td>Lower Bound on $P(\Pi)$</td>
</tr>
<tr>
<td>6.2.5</td>
<td>Discussion on Performance of Policy II</td>
</tr>
<tr>
<td>6.3</td>
<td>Summary</td>
</tr>
<tr>
<td>7</td>
<td>CONCLUSION</td>
</tr>
<tr>
<td>7.1</td>
<td>Future Work</td>
</tr>
</tbody>
</table>

BIBLIOGRAPHY

APPENDIX

viii
# LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>A typical sensor device, courtesy of Crossbow Technologies Inc.</td>
</tr>
<tr>
<td>1.2</td>
<td>Realizing missed transition</td>
</tr>
<tr>
<td>4.1</td>
<td>Evolution of sensor states</td>
</tr>
<tr>
<td>4.2</td>
<td>Example of rainfall as an event-occurrence process ( p_{on}^c = P_{rain} ) and ( p_{off}^c = P_{no} )</td>
</tr>
<tr>
<td>4.3</td>
<td>State transition diagram for ON and OFF periods</td>
</tr>
<tr>
<td>4.4</td>
<td>Model of sensor’s operation</td>
</tr>
<tr>
<td>4.5</td>
<td>An example of change in the system state</td>
</tr>
<tr>
<td>4.6</td>
<td>Sensor activation model correlating with event process</td>
</tr>
<tr>
<td>5.1</td>
<td>Comparison of performance of each policy</td>
</tr>
<tr>
<td>6.1</td>
<td>Sensor activation/deactivation as a Markov chain</td>
</tr>
<tr>
<td>6.2</td>
<td>Expected length of ON and OFF period</td>
</tr>
<tr>
<td>6.3</td>
<td>Performance for ( p_{on}^c = 0.7 ) and ( p_{off}^c = 0.9 )</td>
</tr>
<tr>
<td>6.4</td>
<td>Steady state probability of dead state for ( p_{on}^c = 0.7 ) and ( p_{off}^c = 0.9 )</td>
</tr>
<tr>
<td>6.5</td>
<td>Variation of residual energy level for ( p_{on}^c = 0.7 ) and ( p_{off}^c = 0.9 )</td>
</tr>
<tr>
<td>6.6</td>
<td>Performance for ( p_{on}^c = 0.69 ) and ( p_{off}^c = 0.8 )</td>
</tr>
<tr>
<td>6.7</td>
<td>Steady state probability of dead state for ( p_{on}^c = 0.69 ) and ( p_{off}^c = 0.8 )</td>
</tr>
<tr>
<td>6.8</td>
<td>Variation of residual energy level for ( p_{on}^c = 0.69 ) and ( p_{off}^c = 0.8 )</td>
</tr>
</tbody>
</table>
# LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5.1</td>
<td>Performance for varying $K$</td>
<td>34</td>
</tr>
<tr>
<td>5.2</td>
<td>Performance for varying $Q$</td>
<td>35</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

For many years, technological advancements and their applications have benefited mankind in several ways. One of the latest and most useful applications, yet to be explored completely, has been remote sensing. Remote sensing is used for various applications of which habitat monitoring, military surveillance which seldom encounter human intervention will be considered in this thesis. Wireless sensors are the important devices used in such applications. The concept of wireless sensors has been leveraged from the most sophisticated and advanced sensory system in nature. The sensory system of any organism on earth is the most advanced and plays an important role in day-to-day living by helping the organism make numerous decisions. Similarly, the wireless sensors role is to gather sensory data from the real world and use it for decision making. In this chapter we shall provide an overview about the characteristics, purpose of deployment, and operational mechanism of wireless sensors.

1.1 Sensor Characteristics

Sensors pose great benefits to mankind including increased homeland protection, protection of natural resources, lower catastrophic breakdowns, and superior manufacturing production. These applications demand a combination of efficient data deliverance by a sensor and integration of sensors into th environment, structures, and machines [7]. Some of the other characteristics of a
sensor that make it a dynamic and an advanced device are the following:

- Small and less expensive
- Smart and programmable
- Scalable
- Low power utilization
- Swift data acquirement
- Reliable and accurate over the long term

Figure 1.1 – A typical sensor device, courtesy of Crossbow Technologies Inc.

Figure 1.1 depicts a typical sensor device, courtesy of Crossbow Technologies Inc. [2]. Sensors which may be part of a bigger wireless sensor network are deployed to monitor certain interesting phenomenon occurring in the environment. These sensors sense/detect the occurrence of events and the sensed data is transmitted to a base station either directly or, if required, through a network of sensor nodes. Although these devices have a variety of applications, the main challenge is faced during applications with infrequent human intervention. Since sensors are tiny devices, the components mounted on these devices are also scaled accordingly. One of the components important for the operation of a sensor is the battery, which possesses a finite and small capacity to store charge/energy. The charge accumulated in the battery is consumed gradually during a sensors activity. Some of the applications mentioned earlier demand longevity in a sensors operation and may not be seemingly possible to replace or recharge the battery by humans when the battery dies.
out or loses charge. Hence, the sensor must operate intelligently so that the battery retains energy for a longer duration. One of the lifelines for a sensor is a rechargeable battery that replenishes energy from available natural resources. However, the recharging process is random since several factors like temperature, pressure, and sunlight control the environmental conditions.

1.2 Issues with Sensor Operation

Researchers have taken the problem of energy consumption seriously and are constantly proposing efficient operational algorithms. Some of the highlighted issues in a sensor operation researchers have considered are the following:

• Activation of the sensor when required.

• Activation of the sensor only during the occurrence of an event.

• Deactivation of the sensor when sensing or transmission is not essential.

• Event-driven transmission by a sensor.

• Avoidance of transmission of redundant data.

Each of these issues was considered in this thesis in order to develop an efficient decision-making policy for rechargeable sensor systems.

1.3 Correlated Nature of Event Occurrence

In this section, the nature of events that are being sensed by a sensor is discussed. This is important because the node activation/deactivation question that needs to be answered is very much dependent on the events of interest that show a considerable degree of correlation in their occurrences [1]. Events occurring in nature are random and exhibit two different characteristics, spatially correlated and temporally correlated.
Spatially correlated events:

Demonstrate a substantial degree of spatial correlation. An example would be the occurrence of flood caused by a river which may have high correlation with respect to space and time. If there is a sign of flooding at point A then there is a high probability that point B, downstream of point A, will get flooded later.

Temporally Correlated Events:

Are correlated in time only. As an example if there are signs of smoke accumulating in a room, then there is a high probability that the water sprinkler may switch on after a period of time.

1.4 Decision-Making Policy/Algorithm

A decision-making policy or algorithm is the answer to the question: “How and when must the sensor be activated/deactivated?” The entire operation of a sensor is based on this policy. For the design of such an algorithm or policy it is important to understand the characteristics and issues involved in the operation of sensors, as discussed previously. Since sensors are known to be small devices with small energy reserve and lower computational capabilities, it is necessary to have an efficient, reliable, and less-complex policy. Although each of the issues discussed in Section 1.2 is implicitly addressed, the prime focus for improvement has been to avoid transmission of redundant data, unlike that proposed by Jaggi et al. [1]. As part of this research, several decision-making policies have been evaluated and compared to determine the near-optimal policy “Π”.

Sensors are usually deployed redundantly in a network to detect the occurrence of interesting events in the region being monitored. These events show some degree of temporal correlations, thus making the event-occurrence a random process. In this thesis, a single sensor from the network was considered and evaluated with different policies. The design of a policy for the sensor is challenging, given the event-occurrence process. A sensor loses the available charge in a discrete manner, as long as it is activated. Hence, the sensor is expected to stay activated for an optimal
duration so that the event occurrence is sensed efficiently with optimal energy expenditure. Also, a sensor that is not intended to be active should be deactivated or put to sleep. Throughout the sensor operation, the recharge of batteries occurs at random intervals of time.

As part of this thesis, the event-occurrence process has been modeled in the form of transitions. Transitions refer to the change in the state of event occurrence and are of two distinct types. An active sensor transmits sensed transitions and obtains a reward. This reward is not uniformly distributed for every transmission since some transitions may be missed and transmitted late. An inactive sensor may miss transitions, and at the most one, of the missed transitions may be realized once the sensor activates, as shown in Figure 1.2. The observation made before the sensor has decided to deactivate is compared to the current observation made by the sensor upon activation. If there is a mismatch in the two observations, the sensor easily determines that at least one transition was missed. While being inactive, transitions may have occurred more than once, but the exact number can neither be estimated nor guessed by the sensor. Although the exact number may not be known to the sensor, intelligently sensor can realize that one transition was missed if an odd number of transitions were missed.

![Figure 1.2 – Realizing missed transition](image)

A reward of 1 is obtained whenever the sensor actively senses a transition and transmits. The reward is reduced based on the time taken by the sensor to transmit a transition. Lower rewards are obtained for the late transmission of a missed transition, and the reward decreases exponentially with the time taken. The reward obtained by the sensor after making a transmission may
be represented mathematically as an exponential function, $e^{-t}$, which has the characteristic of a decreasing function with an increase in $t$. Here, $t$ refers to the time taken by the sensor to realize and transmit a transition from the time of its occurrence. In the case where the sensor is active at the time of a transition, then $t = 0$ and the reward is 1. If the sensor realizes and transmits a missed transition $t$ time slots from the time of transition, then the reward will be $e^{-t}$. Over the long term, the sensor will make many transmissions, which include more than one actively transmitted transition and more than one transition missed but later realized. The cumulative reward obtained after these many transitions is referred to as Total Reward.

Performance of a sensor is thus based on the total reward obtained in a time interval of $[0, \cdots , T]$. Let $\Upsilon_o(T)$ denote the number of transitions that occur within this time interval. Of these many transitions, let $\Upsilon_s(T)$ be the number of transitions actively sensed and transmitted and thus receiving a reward of 1. It is clear that a total of $\Upsilon_o(T) - \Upsilon_s(T)$ transitions were missed, of which $l$ transitions are realized and transmitted late. Each of these $l$ late transmitted transitions gets a reward in $[0, \cdots , 1]$, with 1 being the maximum. Let $t_l$ be the time taken by the sensor to realize and transmit the $l^{th}$ transition and the reward equals to $e^{-t_l}$. For a sensor operating under some decision-making policy $\Pi$, the objective is to maximize the performance or reward per transition, $P(\Pi)$.

$$P(\Pi) = \lim_{T \to \infty} \frac{(\Upsilon_s(T) \times 1) + \sum_l e^{-t_l}}{\Upsilon_o(T)}$$  \hspace{1cm} (1.1)

With this definition of the performance, it is easy to understand that the sensor performs best if it actively captures the maximum number of transitions. But the solution to this problem is not as easy as it seems. The main obstacle in the working of a sensor is the limited energy supply which denies the sensor from being active for an unlimited amount of time. The decision-making policy to be developed must consider the nature of the event-occurrence process and ensure that the sensor decides on an optimum sleep/active interval in order to have efficient energy expenditure and to maximize $P(\Pi)$. Generally there exists a trade-off between the energy conservation and the performance achieved by the sensor. Hence the goal is to establish a balance and maximize
performance with efficient energy utilization.

1.5 Contribution

This problem considers a single, partially rechargeable sensor, which may activate when partially recharged and above a certain threshold level. Since the recharge and discharge processes are random processes, the complete sensor system operation is modeled as a semi-Markov process. The important contribution of this research work is the implementation of a novel approach to avoid excess energy expenditure due to redundant transmissions. Redundant transmissions contain information that is already known to the base station by an earlier transmission. Such redundant transmissions not only consume available energy but also remove the available bandwidth available for a network of sensors. Although a single sensor is considered for this work, this problem may be encountered when the sensor is deployed in a network.

The approach taken, to avoid redundant transmissions, is to detect only the change in the event process rather than observing and transmitting the entire event process. For instance, assume the monitored event process is the occurrence of rainfall in a forest to obtain the duration of rainfall in a day. Ideally the sensor may notify the time it starts raining and the time it stops raining. With this information, the duration of rainfall can be estimated and it is not necessary for the sensor to stay active for the entire duration of the rainfall. By suppressing this redundant information, energy is conserved.

It is assumed that an inactive sensor is incapable of observing the event process and eventually may have missed one or more transitions. As explained earlier (Figure 1.2) the sensor possesses intelligence to realize that at least one transition was missed if an odd number of transitions occur while the sensor remains inactive. The sensor system is modeled so that the sensor transmits the missed transition instead of discarding it, since late transmission is believed to be better than no transmission.

To analyze the performance of the sensor operation, a deterministic decision framework is de-
Several simple and intuitive activation policies are developed with the help of simulation to determine the best-performing policy, $\Pi$. Policy $\Pi$ is also analyzed analytically to determine the upper and lower bounds on the achievable performance.
Chapter 2

RELATED WORK

The development of sensor nodes has been followed by incredible research attention in the area of Sensor Networks. A sensor network, or network of several sensor nodes, has several applications in the real world. Some of the applications are pollution control, tracking animal movements, patient monitoring for medical purposes, and smart home. A detailed background on the nature and operation of a sensor in a sensor network with applications is discussed by Lewis in [7]. Issues involving a sensor network and the advances made with the help of different algorithms to deal with these issues have been addressed by Lewis in [8]. Amidst most of the issues of a sensor there has been widespread research in the area of energy management in a sensor.

Non-Rechargeable Sensors: Using the example of object tracking, Fuemmeler and Veeravalli in [6] developed a sleeping policy for the sensor to conserve energy and stay active when required. The trade-off in designing a sleeping policy is the tracking error which results because of inactive sensors. The area of the object’s movement is densely populated with sensors in order to accurately track the movement to the range of the sensor. It is assumed that the path of the objects movement is either known a priori or estimated. Dynamically, the sleeping duration is calculated based on the presence of the object before going to sleep. Thus, sleeping policies are designed to optimize the trade-off in tracking errors. To support this, the authors have assumed the presence of a central unit that determines the sleeping
durations for sensors. These central units are believed to have complete information about the objects movements. The main contribution of this paper is the design of a *Partially Observable Markov Decision Process* (POMDP) framework and the design of sub-optimal sleeping policies since an optimal solution is not possible.

Jaggi and Krishnamurthy [4] worked on connectivity and coverage issues in two different forms of sensors, rechargeable and non-rechargeable. They studied the behavior and improvement techniques of these two forms of sensors deployed in separate sensor networks around the point of interest. These sensor networks were deployed to monitor certain events of interest with spatial or/and temporal correlation. These authors [4], considering non-rechargeable sensors, divided the entire sensor network into several disjointed sets of sensors, mutually exclusively active at a given time. A set of sensors that is activated is referred to as an “active set”. The solution of having disjointed active sets of active sensors at any point of time helps in obtaining greater coverage and connectivity around the point of interest. Good connectivity is established by having appropriate communication methods between sensors, belonging to the active set, and the base station. The other sets of sensors that are not required to be active in the presence of another active set may sleep and conserve energy. The authors further propose an activation policy that activates only one set of sensors at a time, thus providing connectivity and coverage with energy constraints.

**Rechargeable Sensors:** Kar et al. [3] analytically developed a novel threshold activation policy that achieves performance of at least $\frac{3}{4}$ of the optimum over all other policies. Sensors were assumed to overlap completely/partially in their coverage and have an exponential recharge/discharge models. After analyzing a completely overlapping scenario, the threshold activation policy was then extended to the practical scenario where partial overlap may be observed. Through various simulations and analytical studies it was shown that there exists a threshold activation policy that achieves nearly optimal results. Also, the assumption made in this paper is that the sensor may be activated only when completely charged.

Jaggi et al. [5] considered rechargeable sensors, which may be activated although they are
partially recharged. In this paper the authors show that since the recharge and discharge process is random in a realistic sensor network the problem of determining the optimal sensor activation policy is a complex stochastic decision question. The authors argue that to obtain better coverage of the point of interest it may be necessary to make a redundant deployment of sensors. To ensure better coverage it may be necessary to have all the sensors stay active, but the trade-off here is the network lifetime. Also, sensors at different points of deployment exhibit different coverage patterns. Therefore, the utility of the sensors per unit area, and per unit time show decreasing results with an increase in the number of sensors. This utility function seems higher in the short term with all the sensors active, but over time, sensors lose charge and may deactivate to reduce the utility function. In this paper, the objective is to maximize time-average generalized coverage metric.

As part of this research work, a novel technique is proposed to efficiently utilize the available energy in a rechargeable sensor by reducing redundant transmissions. The focus here is on sensing and transmitting transitions or crucial edges and help conserving and efficiently utilizing available energy. Ideally, the sensor is expected to be active during the time of a transition and otherwise, sleep. Just like tracking errors [6], sensors may miss transitions if they remain inactive for a non-optimal duration. Hence, the missed transitions, although realized late by the activated sensor, are transmitted late instead of being discarded.
Chapter 3

DISCRETE EVENT SIMULATION

Simulation is a process through which a system model is evaluated numerically, and the data from this process is used to estimate various quantities of interest. In this thesis, a discrete event simulation of an entire sensor system operation has been performed for each of the intuitively developed policies, [9]. Similar to the analysis of a discrete event system, several components are identified and used for the simulation purposes:

- **State**: list of state variables.
- **Time**: variable holding the simulation time.
- **Scheduled Event List**: list of scheduled events and their occurrence times.
- **Data Registers**: Variables and/or lists to store observed data used for estimation and evaluation.

### 3.1 M/M/1 Queue

To be able to simulate the sensor system, it is necessary to be introduced to the simulation of a queuing system. For this, an M/M/1 queue using C programming and Matlab was simulated. The basic simulation was performed in C, and the results were plotted using Matlab. The simple M/M/1
A queuing system may be viewed as being comprised of entities (e.g., customers, jobs) undergoing service as they flow through the discrete event system. The flow process has a timed structure, which describes the occurrence of events in a sequence. At a given time instance, each entity is waiting for service or being served by the server. Each entity in this system undergoes the following process:

- The entity arrives into the system (the arrival of each entity is separated by a random fraction of time referred to as Inter-Arrival time).
- The entity enters the Queue.
- If the server is found idle, the entity requests service and holds the necessary resources. If the server is not found to be idle, the entity remains in the queue and waits until the server is free to serve.
- Each entity is served for a random interval of time referred to as Service Time.
- The entity departs after it has been served completely by the server and the held resources are released.

In a simple M/M/1 queue, the arrival and departure processes follow the Poisson distribution. In other words, the arrival of a job and the departure of a job is Markovian in nature since both events occur randomly in nature and are independent of past arrivals or departures. The arrival rate is denoted by jobs arriving per unit time, and the service rate is denoted by jobs departing per unit time. In this system we consider the queue capacity to be infinite. Different data variables are used to store data for estimation purposes. The data essential for the estimation and evaluation are current time, arrival time, departure time, waiting time, and system time of the corresponding job, number of jobs in the system, queue length occupancy times, and total number of jobs that entered the system throughout the simulation. With this information it is then easy to evaluate the performance of the system by calculating the following required parameters:

- Expected Average System Time
• Average Waiting Time
• Throughput
• Server Utilization
• Mean Queue Length

The simulation of this queuing system was performed using C language which provided the function drand48() to generate random numbers used to determine the inter-arrival time between two jobs and the service time of a job.

### 3.2 Sensor System Model

By extending the knowledge of a basic M/M/1 queuing system to the sensor node activation/deactivation process, it is easier to analyze the operation of a sensor node as a discrete event system and hence evaluate the performance. As in any queue, the sensor sees the recharge quanta as the arriving entity and discharge quanta as the departing entity. The arrival process of recharge quanta follows an independent and identical distribution and is random in nature. The departure process of the discharge quanta is an event-driven process. The states of the sensor and event process play a key role in the discharge modeling of the sensor. Energy is discharged if and only if the sensor is active. If the active sensor polls the event process while the event is occurring and when a transition is transmitted, then there is additional discharge of energy. However, an inactive sensor never loses energy. The occurrence of a transition in the event processes is based on a certain probability, and because of this, the discharge of energy happens with a certain probability. The randomness of event occurrence is obtained with the help of a random number generator.

Different variables are used to store data that are further used for statistical analysis. Intelligence to realize missed transitions is added to the sensor through simulation techniques by utilizing more data variables. In this thesis, simulation was performed with the help of C programming language in Unix environment.
3.2.1 Simulation Algorithm

An overview of the algorithm governing the operation of the sensor is given below:

- Sensor is recharged by the arriving quanta with a certain probability.

- If the sensor is active in time slot $t$ and a transition occurs, then the sensed information is transmitted and energy gets discharged accordingly.

- If the sensor is active in time slot $t$ and no transition occurs, then only the energy required for the sensor to remain active gets discharged.

- When the sensor is inactive, no sensing is performed and hence no energy is discharged.

- If an odd number of transitions occur while the sensor is inactive, the sensor will be able to realize that at least one transition was missed once it is activated. The missed transition is then transmitted.

- A decision for the next time slot is made at the end of the current time slot. If the sensor is deactivated, a decision is made to sleep for a period of sleep interval. Similarly, an activated sensor decides to remain active for a period of active interval if no transition is sensed and sufficient energy is available.

- If the energy available is lower than a certain threshold level the sensor dies and remains dead for an uncertain period of time until sufficient energy is available for the sensor to be activated.
Chapter 4

PROBLEM FORMULATION

4.1 Introduction

This research considered just only one sensor node and evaluated the performance of various intuitive policies to determine the policy $\Pi$ that achieves the highest reward per transition with efficient energy utilization. The sensor considered in this problem is equipped with a low-cost rechargeable battery, which has the ability to harness energy from the environment with the help of natural resources. This recharge practically happens with a certain probability since the environmental conditions are vulnerable to changes. Although the batteries have the ability to replenish energy from available natural resources, this is not a continuous process, and recharge occurs at random instants of time. The sensor cannot rely only on the recharge quanta for energy in order to stay active for longer durations. Hence, it has become extremely necessary for the sensor node to operate judiciously such that the reserve energy is utilized efficiently in order to maximize achievable performance.

This chapter presents the individual models that comprise the complete sensor system. Section 4.2 describes the different operational states of the sensor. Section 4.3 discusses the occurrence process of the events of interest. Section 4.4 explains the recharge and discharge model of a sensor. An overview of the sensor activation problem is provided in Section 4.5.
4.2 Classification of Sensor State

Since the residual energy of a sensor changes with time, the sensor switches states to conserve and utilize energy when required. The sensor may be found in one of the three states —Active, Sleep or Dead—at any given instant of time. The evolution of sensor states is depicted in Figure 4.1.

- **Active**: The sensor may observe the event process and sense changes in the state of the event process. Also, the sensor may accordingly transmit the observed or sensed information to the base station and lose energy accordingly. Here the sensor decides to remain active for a period of active interval and is recharged by the random arriving quanta. An active sensor decides to sleep if sufficient energy is not available to stay active for at least one time slot.

- **Sleep**: The sensor can neither observe the event process nor transmit in this state. An active sensor decides to sleep if any of the following apply: (i) a transition is sensed, (ii) sufficient energy is not available, or (iii) the sensor was active for a period of active interval and then the timer expired. The sensor decides to sleep for a period of sleep interval and while sleeping the sensor also is recharged.

- **Dead**: The sensor eventually dies if residual energy in the sensor is not sufficient to stay active for at least one time slot. After sleeping for a sleep interval if the sensor has to
activate but does not possess sufficient energy, the sensor dies. The sensor aggressively wakes up once sufficient energy is available after being recharged. The sensor in the dead state will not be able to sense the event process.

4.3 Event-Occurrence Process

The modeling is only concerned about the events demonstrating correlation in time only. Temporally correlated event occurrence process may be understood taking rainfall occurrence in a particular area as an example. Generally, in a forest area that acts as a catchment area, there may be heavy or constant rainfall. In a day, there may be continuous rain for hours, followed by no rain for some time, and then it may rain again. This process happens randomly and at different time intervals. If the forest is receiving heavy rain for one hour constantly then it is easy to predict if there may be rainfall for the next one minute. Here, discrete time slots of 1 minute each are assumed. The prediction comes with a certain probability. Therefore, it is assumed that with a probability of $P_{\text{rain}}$, it will rain in the next one minute and with a probability of $1 - P_{\text{rain}}$, it will stop raining. Similarly, if it is not raining currently, then the probability that it will not rain in the next one minute is $P_{\text{no}}$, and the probability that it will start raining is $1 - P_{\text{no}}$. The correlation probabilities that govern the rainfall are $P_{\text{rain}}$ and $P_{\text{no}}$. Figure 4.2 illustrates this example.

As long as there is rainfall, the event process is in the ON period and similarly, the period when there is no rainfall is known as the OFF period. In either of these two periods for each discrete time interval of one minute, we say the event process is ON if it is raining and OFF if it is not raining. It can be seen that rainfall is observed from $t = 15$ to $t = 25$, which is also called ON period. Similarly, an OFF period is observed until $t = 15$, starting from a point in between $t = 5$ and $t = 10$.

For the problem in this thesis, the correlation probabilities are given by $p_{c}^{\text{on}}$ and $p_{c}^{\text{off}}$ such that $0.5 < p_{c}^{\text{on}}, p_{c}^{\text{off}} < 1$, [1]. Suppose an event occurs in time slot $t$, then in the time slot $t+1$, a
Figure 4.2 – Example of rainfall as an event-occurrence process \( p_{on} = P_{\text{rain}} \) and \( p_{off} = P_{\text{no}} \)

similar event occurs with a probability of \( p_{on} \) or no event occurs with a probability of \( 1 - p_{on} \). Correspondingly, if no event occurs in the current time slot \( t \), then in the next time slot \( t+1 \) no event occurs with a probability \( p_{off} \) or an event occurs with a probability \( 1 - p_{off} \). The state transition diagram of the event process is illustrated in Figure 4.3.

Figure 4.3 – State transition diagram for ON and OFF periods

**ON and OFF PERIODS:** The duration for which events occur consecutively is referred to as the ON period and the duration for which no events occur is referred to as the OFF period. As part of this work the task is to develop a decision-making policy that detects and transmits the transitions occurring between the two ON and OFF periods as illustrated in Figure 4.3. Relating back to the example of rainfall, consider that a transition to have occurred at \( t = 15 \) from the OFF period to the ON period. Similarly a transition occurs at \( t = 25 \) from the ON period to the OFF period. From this, it can be understood that there are two distinct types of transitions, OFF to ON, and vice versa. Thus, sensor is expected to detect these two transitions and transmit for further
analysis. With reference to rainfall occurrence, the transitions may also be seen as the time at which rain started and stopped. This information is sufficient to determine the duration of rainfall or duration of event occurrence, and for this sensor need not remain active for the entire ON period.

A cycle comprises one of each ON and OFF periods, and many such cycles repeat in succession but of random length. Practically, the length of the OFF period is longer than that of the ON period. This implies that \( p_{\text{off}} \geq p_{\text{on}} \). Since the event-occurrence process is a random process, it may be appropriate to mention that the length of such, ON or OFF, periods are of random duration. Let \( On \) denote the random variable representing the length of an ON period. For the length of \( On \) period, i.e., from time \( t \) to \( t+On \), the event occurs with a probability of \( p_{\text{on}} \). A transition from ON to OFF happens with a probability of \( 1 - p_{\text{on}} \). The probability of seeing the length of the ON period equal to \( On \) is \( Pr[On = i] = (p_{\text{on}})^{i-1}(1 - p_{\text{on}}), \forall i \geq 1 \). The expected length of the ON period \([1]\) is thus given by

\[
E[On] = (1 - p_{\text{on}}) \sum_{i=1}^{\infty} i(p_{\text{on}})^{i-1} = \frac{1}{1 - p_{\text{on}}}.
\]  

Thus, \( E[On] \) is the expected length of the ON period and similarly the expected length of the OFF period, \( E[Off] \), is given by \( \frac{1}{1 - p_{\text{off}}} \). Furthermore, the steady-state probability of event occurrence can be evaluated with the help of Markov chain analysis. As \( T \to \infty \), the value of \( \pi_{\text{on}} \) (steady state probability of event occurrence) is \( \frac{1 - p_{\text{off}}}{2 - p_{\text{off}} - p_{\text{on}}} \). Similarly the value of \( \pi_{\text{off}} \) may also be obtained. Intuitively, it is easy to determine \( \pi_{\text{off}} \), since \( \pi_{\text{off}} = 1 - \pi_{\text{on}} \), which further implies that \( \pi_{\text{off}} \geq \pi_{\text{on}} \), if \( p_{\text{off}} \geq p_{\text{on}} \) \([1]\).

### 4.4 Recharge/Discharge Modeling

Rechargeable sensors are capable of harnessing energy from the available natural resources at the place of deployment. However, the recharge of energy is a discontinuous process since it totally depends on environmental factors. The sensor operation is characterized by two distinct features \([5]\):
• **Mutually Exclusive Recharge/Discharge:** An active sensor may get discharged incessantly, devoid of any recharge. On the other hand, an inactive sensor cannot get discharged but may get recharged. The recharge process, modeled as a Bernoulli process with a recharge probability of $Q$ and recharge quanta as $C$, depends on the renewable energy resources, while the discharge process depends on the state of the sensor (active/inactive), state of event process (ON/OFF) and transmissions made by the sensor.

• **Partially Rechargeable Sensors:** Sensors need not be completely recharged to be activated. To activate the sensor, the energy available must be greater than the required threshold level, as discussed in Section 4.5.5.

An active sensor consumes some energy, $\delta_1$, as part of the operational cost. As long as the sensor is active, $\delta_1$ amount of energy is discharged per unit time from the available quanta in the energy bucket. More energy is consumed depending on the state of the event process. If the sensor is active and when it polls the event process, $\delta_2$ amount of energy is spent if the event process is ON. If the event process is OFF, no energy gets discharged by the active sensor. The function of the sensor in this problem is to capture the transitions or changes in the state of the event process and transmit the sensed transitions by consuming $\delta_3$ amount of energy.

The sensor has a high discharge rate since a considerable amount of energy is consumed for the transmission of sensed information. Thus, the objective of this research is to avoid redundant transmissions and thus conserve energy. Frequent and identical data transmissions are referred to as redundant transmissions. Ideally it is sufficient if the sensor is able to communicate to the monitoring station about the change in the event process.

### 4.5 Sensor Activation Problem

As discussed in the previous section, the approach taken to achieve the goal of maximizing performance with efficient energy expenditure is to reduce redundant transmissions. In addition to this, it is known that an active sensor consumes a considerable amount of energy. Hence, the
sensor must be activated and deactivated appropriately. In other words, the sensor must choose an 
opimal duration to remain active or to sleep. The sensor activates itself, if and only if the energy 
is greater than a certain threshold; otherwise, it decides to sleep for a certain duration called the 
sleep interval (SI). By deciding to sleep the sensor ensures that it gains sufficient energy to stay 
active when required. The key observation to be made is that the more the sensor sleeps, the lower 
the chances of capturing the occurring transitions. Hence, there is a need to decide the optimum 
sleeping duration for a sensor in order to maximize the performance of the sensor and capture as 
many transitions as possible.

It appears that no work in the past has considered the duration the sensor needs to stay activate. 
Closely related work [1] has used a similar approach, but the authors designed the model so that the 
active interval (AI) varies with the length of the ON period, and as long as the event is occurring 
the sensor is active and is deactivated if no event has occurred in the previous time slot. The work 
in this thesis considers an optimal active interval, which decides the duration the sensor needs to 
stay active. The sensor decides to deactivate itself by preempting the timer if it captures a transition 
or if the energy level of the sensor is not above the threshold.

The model of the sensor operating under a well-defined decision-making policy is shown in 
Figure 4.4. Now with an example of some event process, exhibiting temporal correlation, a sample 
decision-making policy for the sensor is presented.

4.5.1 Key Assumptions

For analysis and modeling purposes certain assumptions are made. Some of these, mentioned 
below may be easily understood with the given example in the subsequent section:

- The decision for time slot \( t+1 \) is made after observing the event process in time slot \( t \).

- An active sensor may deactivate itself, before the active timer is zero, if a transition occurs 
during this period or if the energy level falls below the energy threshold level. Otherwise,
the sensor must remain active for the entire period determined by the Active Interval.

- Transitions, between the ON and OFF periods, are assumed to occur in the middle of any given time slot.

- The state of the complete system changes only at the end of any given time slot, irrespective of the occurrence of a transition. For example, as shown in Figure 4.5, the state of the system changes at the end of any given time slot. Although a transition occurs from ON to OFF in time slot $t + 1$, the state of the system remains (ON, Active) until the end of the time slot. The state space of the system is explained in detail in Section 6.1.1.

- The performance evaluated depends on the observations made during each time slot $t$.  

\[ \delta_1: \text{Energy consumed per unit time if sensor is Active} \]
\[ \delta_2: \text{Additional energy consumed per unit time for sensing during ON period} \]
\[ \delta_0: \text{No additional energy consumed per unit time for sensing during OFF period} \]
\[ \delta_3: \text{Additional energy consumed per unit time after making a transmission} \]

**Figure 4.4** – Model of sensor’s operation

**Figure 4.5** – An example of change in the system state
4.5.2 Example of an Intuitive Policy

This section presents an example of a decision-making policy controlling the operation of a sensor monitoring an event process as illustrated in Figure 4.6. Here is shown temporal correlations exhibited by the event process within a given interval $[t, \cdots, t+11]$, for some $t$. Accordingly, the sensor, operating on the decision-making policy, senses the occurring transition if it is active during the time slot. Some of the terms used are as follows:

- $AI_{ON}(AI_{OFF})$: Active Interval for Event Process: ON (OFF).
- $SI_{ON}(SI_{OFF})$: Sleep Interval for Event Process: ON (OFF).

Initial conditions, at $t$, are that the sensor is Active and the Event Process is ON (chosen randomly). During time slot,

- $t$: Sensor senses no transition and looses energy accordingly, $\delta_1 + \delta_2$. Sensor decides to sleep for $SI_{ON}$ time slots, equal to 1 in the example.

- $t+1$: Sensor is inactive and no transition occurs. Sensor decides to activate for $AI_{ON} = 1$ time slots.

- $t+2$: Sensor senses no transition and looses energy accordingly, $\delta_1 + \delta_2$. Sensor decides to sleep for $SI_{ON}$ time slots.

- $t+3$: Sensor is inactive and missed the occurrence of a transition. The event process switched to the OFF period as a result of the transition. At the end of the time slot, the sensor decides...
to activate the sensor. This decision is made for the time slot $t+4$. Before deciding the
Active Interval, which is dependent on the event process, the sensor checks if it has missed
any transition while being inactive in the time slot $t+3$. The sensor knows the event process
has changed from ON, when last active, to OFF. This is called as *Late Realization of Missed
Transition*. The sensor transmits this sensed information to the base station, although late,
and loses charge for transmission only, i.e., $\delta_3$. This feature of realizing a missed transition
and making a decision based on the sensed information is the intelligence added to the sensor.

• $t+4$: Sensor has decided to stay active for $AI_{OFF}$ time slots since the event process is OFF.
In this example $AI_{OFF}$ is two time units. Since no transition occurs, the decision made for
the next time slot, $t+5$, is to stay activated. The energy lost during this time slot is only $\delta_1$.

• $t+5$: Sensor is active and senses no transition. By the end of the time slot the active timer
expires, and the decision made for time slot $t+6$ is to sleep for $SI_{OFF} = 2$ time slots. The
energy lost is only $\delta_1$.

• $t+6$: Sensor is inactive and no transition occurs.

• $t+7$: Sensor is inactive and no transition occurs. The decision made for time slot $t+8$ is to
activate since the sleep timer expired. Sensor decides to stay active for $AI_{OFF} = 2$ time
slots.

• $t+8$: Sensor is active and detects the transition from OFF to ON and transmits. Energy
discharged in this process is $\delta_1 + \delta_2 + \delta_3$. Since a transition was sensed and transmitted, the
sensor decides to sleep. Although the sensor decided to stay active for $AI_{OFF}$ time slots, at
the end of $t + 7$, the sensor now decides to sleep for $SI_{ON} = 1$ time slots, since the event
process is ON.

• $t+9$: Sensor is inactive and no transition occurs. The decision made for time slot $t+10$ is to
activate since the sleep timer expired. Accordingly, the decided active interval is $AI_{ON} = 1$. 

25
• $t+10$: Sensor senses no transition and the energy lost is $\delta_1 + \delta_2$. Sensor decides to sleep for $S_{ON} = 1$ time slots.

• $t+11$: Sensor is inactive and misses a transition.

In the detailed example it may be observed that a transition occurs when the sensor was inactive and another transition occurred when the sensor was active. In either case although transition occurred in the middle of the time slot, $t+3$ or $t+8$, the state of the system remained unchanged. For example, the state of the system until the end of time slot $t+3$ will be \{EventProcess = ON; SensorState = Sleep/Inactive; SleepTime = 1; ActiveTime = 0\}. With respect to the system state until the end of time slot $t+3$, (i) the event process is considered to be ON, (ii) the sensor is inactive with residual sleep time as 1, and (iii) the residual active time is 0, i.e., sensor is not active. Section 6.1.1 provides a detailed definition of the state of the system.

### 4.5.3 Handling Missed Transitions

Figure 4.6 shows that a transition occurs in time slot $t+3$ and that the sensor is inactive. Hence the sensor is not able to sense/capture the change in the event-occurrence process. The sensor later activates in $t+4$ to find the change in the event process. The sensor decides to sleep at the end of $t+2$ for one time slot when the event process is ON. The sensor then sleeps during the time slot $t+3$, but there is a transition from ON to OFF. The sensor at the end of time slot $t+3$ decides to activate and then observes the event process to be OFF. The sensor is now in a position to determine at least one transition is missed. In this case, at most one transition can occur/be missed. The sensor is late by one time slot in realizing the occurrence of a transition. This transition is notified to the base station and fetches some reward, but less than one. The reward is reduced with the amount of time taken by the sensor to realize a missed transition. The idea of transmitting late may seem unnecessary since the transition occurs in the past and it may not be significant anymore. This feature is included since late transmission is much better than no transmission.

While the sensor is inactive, there may be more than one transition but this number is neither
known to the sensor nor can be estimated. The sensor will be able to realize a missed transition after being activated, if and only if there is an odd number of transitions during the inactive period. For instance assuming that the sensor decides to sleep at the end of $t$ and the event process to be OFF, let the sensor stay deactivated for ten time slots, i.e., until the end of time slot $t + 10$. During this inactive period it can be assumed that a transition happens from OFF to ON period in the middle of time slot $t + 3$. Another transition occurs from ON to OFF in the middle of $t + 4$. If no more transitions occur before the sensor activates, in $t + 11$, the event process observed by the activated sensor will be OFF which is the same as it was at the end of time slot $t$, when the sensor decided to deactivate. In this case, although there were two transitions during the inactive period of the sensor, the sensor is not able to realize it. Assume that there is another transition in the middle of time slot $t + 7$, i.e., from OFF to ON and then no transition occurs before the sensor activates, in this case, the sensor, when activated in $t + 11$, observes a change in the event process, from OFF (in $t$) to ON (in $t + 11$), and determines at least one transition was missed and transmits the same. In this manner, if there are an odd number of transitions during the inactive period, the sensor can realize and transmit the transition. The reward obtained for transmitting late in the above example is $e^{-(t + 11) - (t + 7)} = e^{-4}$. In an interval of $[0, \cdots, T]$, the sensor makes several transmissions with or without delay. If the transmission faces no delay, the reward fetched is equal to 1 otherwise, the reward is less than 1. Hence, the total reward, $r(N)$, after making $N$ transmissions in an interval $[1, \cdots, T]$ is

$$r(N) = \sum_{i=1}^{N} e^{-(t_i - t'_i)}$$

(4.2)

where $t_i = \text{time at which } i^{th} \text{ transition is transmitted}$, and $t'_i = \text{time at which } i^{th} \text{ transition occurred}$. Transmission without delay will have $t_i - t'_i = 0$, whereas for transmission with delay, $t_i - t'_i > 0$. This feature is modeled in the sensor so that when a decision is made to activate the sensor at the end of any given time slot, the sensor first checks if any transition was missed when inactive and then starts polling the event process. For instance, as given in the example above, at the end of time slot $t + 10$, a decision is made to activate the sensor. After making this decision, the sensor checks for any missed transition. If this is the case, then this transition is transmitted late, and $\delta_3$ amount
of energy is consumed for transmission, and no additional energy is discharged for the sensing performed. After this activity, the sensor now starts polling the event process for any transition in time slot $t+1$.

### 4.5.4 Performance Evaluation

The performance of any sensor is measured on the basis of the number of transitions detected. The performance of a sensor is based on the total reward obtained in a time interval of $[0...T]$. Recollecting the performance evaluation in equation (1.1) in Chapter 1, the reward per transition of any policy $\Pi$ is given by

$$P(\Pi) = \lim_{T \to \infty} \frac{(\gamma_s(T) \times 1) + \sum_t e^{-t(t)}}{\gamma_o(T)}$$  \hspace{1cm} (4.3)

where $\gamma_o(T)$ is the number of transitions that occurs in the given time interval, $\gamma_s(T)$ is the number of transitions actively sensed/transmitted fetching a reward of 1, $l$ is the number of transitions that were missed but realized/transmitted late, and $t_l$ is the time taken by the sensor to realize and transmit the $l^{th}$ missed transition from the time of its occurrence.

The total reward obtained includes the reward for transmissions made with and without delay as given in equation (4.2). For an interval $[1, \cdots, T]$ the total reward is represented as

$$r(T) = (\gamma_s(T) \times 1) + \sum_{\forall l} e^{-t_l}$$  \hspace{1cm} (4.4)

### 4.5.5 Energy Threshold Level

The sensor can activate only when the energy level is above a defined threshold level. If the sensor is active and the residual energy level is not above the threshold level, the sensor decides to deactivate. A deactivated sensor will not be able to perform any sensing. In the problem here, the threshold is equal to the maximum amount of energy consumed when the sensor is active for one time slot. First, once it is activated, the sensor checks for any missed transition. Then it polls the
event process to capture any occurring transition in the current time slot. The sensor then transmits the sensed information to the base station. Hence, the threshold level is chosen to be equal to the energy spent during the activity of the sensor in one time slot. This is equal to the sum of energy in the following instances, (i) required to stay active, (ii) spent to transmit a missed transition, (iii) spent to sense the event process during ON period, and (iv) spent for transmission of a transition. Therefore, the maximum energy consumed during one time slot or energy threshold, $E_{th}$, is given by

$$E_{th} = \delta_1 + \delta_2 + (2 \times \delta_3)$$  (4.5)

where

- $\delta_1$ is the energy spent to stay active.
- $\delta_3$ is the energy spent for transmission after realizing a missed transition. \(^1\)
- $\delta_2$ is the energy spent for sensing during ON period. \(^2\)
- $\delta_3$ is the energy spent for the transmission of sensed transition.

### 4.6 Summary

This chapter focused on the details required to model the sensor operation. The operational states of the sensor were classified and insight into the events of interest and their occurrence pattern were provided. The process of recharge and discharge of energy in a sensor was also explained. With the help of an illustration an example of the activation algorithm was provided followed by the handling of missed transition and the performance criteria.

---

\(^1\)Here the sensor does not consume additional energy for sensing.
\(^2\)No energy is spent for sensing during the OFF period.
A policy is the same as an activation/deactivation algorithm. Any decision-making policy must influence the behavior of the Markovian system as it evolves through time [10] and therefore is called a decision-maker. At any specified point in time, the decision-maker chooses an action or makes a decision, which may fetch a reward, and/or incur a cost, immediately. The system observes a change in the state due to the action chosen or decision taken. This process repeats in succession with a possible change in the state of the system with time. The next state has a distinct probability distribution and is called the state transition probability. The reward awarded and the transition probabilities depend on the choice of action and the current state of the system. Through time the decision-maker receives a sequence of rewards due to the decisions made and corresponding change in states. Hence, a policy prescribes the sequence of decisions or actions. In this chapter, several such decision-making policies for the sensor model described in Chapter 4 are developed and evaluated.

Section 5.1 discusses the various classes of activation policies. Then each of the evaluated policies is described and compared. Section 5.7 compares the performance of each policy with varying system parameters. With this comparison, the best policy for the given problem can be concluded and denoted as policy \( \Pi \).
5.1 Classes of Activation Policies

This thesis has considered three different classes of sensor activation policies:

**Aggressive Wakeup (AW) Policy:** The sensor activates and stays active as long as the residual energy level is above a certain threshold level, equation (4.5). In case the energy level is equal to or less than this threshold level, the sensor switches to the dead state. Until the sensor possesses sufficient energy, the sensor remains in the dead state and replenishes energy with the help of natural resources in the surrounding area. In this policy, the sensor never sleeps, and if sufficient energy is available, then the sensor remains active.

**Deterministic and Preemptive Policy:** This policy is considered deterministic since the choice of active interval or sleep interval is deterministic. It is also preemptive since while the sensor is active, a decision is made to sleep before the timer expires if a transition is detected. This policy involves choices of $SI_{ON}, SI_{OFF}, AI_{ON},$ and $AI_{OFF},$ as discussed in Chapter 4.

**Adaptive Policy:** In a realistic or a practical scenario, a sensor has no knowledge of the global system parameters. An adaptive policy enables the sensor to be decisive with the help of the limited knowledge available. Whenever a sensor needs to make a decision, the current energy level, observable by the sensor, and past decision are taken into consideration. For instance, before going to sleep, the sensor calculates the next sleep interval using the last used sleep interval and current energy level.

The next few sections explain about the different choice of policies. The results of each of the policies are compared and the best performing policy is discussed. It may be observed that the best policy $\Pi$ performs much better than the AW policy.

5.2 Aggressive Wakeup (AW) Policy

Any sensor operating under the AW policy remains active for a longer duration, provided sufficient energy is available. Although the sensors are mounted with rechargeable batteries,
the rate of discharge is much higher than the recharge rate since the sensor is constantly active, 
($\delta_1, \delta_2, \delta_3 > QC$). Eventually the sensor will lose energy and die. From this point on, the sensor 
will spend most of the time in a dead state, thus lowering the reward per transition or performance.

5.3 Two-Timer Policy

This is a type of deterministic and preemptive policy and is very intuitive. Since none of 
the earlier work is related to the problem of transition detection, a simple policy was used to understand the behavior of the system. Two different timers, $AI$ and $SI$, were used to govern the active duration and sleep duration, respectively. The sensor stays active for AI time slots if the following apply: (i) the active timer is running, (ii) no transition occurs, and (iii) residual energy > $E_{th}$. If either of the cases fail, the sensor decides to sleep for SI time slots. Therefore, here $SI_{ON} = SI_{OFF} = SI$ and $AI_{ON} = AI_{OFF} = AI$.

5.4 Three-Timer Policy

In this second type of deterministic and preemptive policy, two different sleep intervals $SI_{ON}$ and $SI_{OFF}$ were implemented, for ON and OFF period, respectively. However, only one active interval, $AI$, is used which is constant for both the ON and OFF periods. The sensor stays active for AI time slots, and once the timer expires the sensor decides to sleep. The sensor preempts the timer and decides to sleep if a transition is sensed or the energy level $\leq E_{th}$. The choice of sleep interval, i.e., $SI_{ON}$ or $SI_{OFF}$, is made depending on the event process while making the decision. Therefore, here $AI_{ON} = AI_{OFF} = AI$.

5.5 Four-Timer Policy

This third type of deterministic and preemptive policy uses four different timers. One timer each is used to track active time during the ON and OFF periods, $AI_{ON}$ and $AI_{OFF}$, respectively.
Similarly, one timer each is used to track sleep time during the ON and OFF periods, $SI_{ON}$ and $SI_{OFF}$, respectively. This is the best policy in its class and demonstrated a distinct property referred to as $TOVO$ (Toggle for ON and Vary for OFF). The sensor performed best when active or sleeping for one time slot during the ON period for different system parameters. In other words, during the ON period, the sensor seemed to toggle between the active and sleep states—hence the name Toggle for ON. During the OFF period, the sensor was active or sleeping for some deterministic and fixed duration, $AI_{OFF}$ and $SI_{OFF}$, respectively. Since this property was consistent with the change in system parameters, only the four-timer policy with $TOVO$ property was considered. However, this property does not hold for the case when $\mu^on < 0.69$. The sensor deactivates itself to a sleep state for a period of $SI_{OFF}(SI_{ON} = 1)$ if the event process is OFF (ON). Once the sensor is activated, it first checks for any missed transition, and based on the latest observations, the sensor decides the active interval, i.e., $AI_{OFF}(AI_{ON} = 1)$, if the event process is OFF (ON). As an example, assume that the sensor deactivated when the event process was OFF but before the sensor activated a transition occurred and the current event process is ON. The observation made after activating is that the event process has been changed to ON. Hence, the sensor decides to stay active for $AI_{ON} = 1$ time slots.

### 5.6 Multiplicative Increase Multiplicative Decrease Policy

The multiplicative increase multiplicative decrease (MIMD) policy is a type of adaptive policy that dynamically performs required calculations. Here the concept of MIMD is used to calculate the required active interval based on the previously used active interval and current energy level. If the energy level is greater than $E_{th}$, the active interval for the current active period is twice that of the last active interval. If the sensor has just been activated from dead state, the active interval is half of the last active interval. The sensor deactivates itself to enter a sleep mode for $SI_{ON}(SI_{OFF})$ period if the event process is ON (OFF). The sensor is deactivated if (i) a transition is sensed, or (ii) the active timer expires, or (iii) the energy level $\leq E_{th}$. 

33
5.7 Comparison of Policies

Each of the above mentioned policies was simulated with the help of C programming language using discrete event simulation [9] and the performance was evaluated. For comparison of the policies, one of the system parameters was varied and the performance results evaluated. System parameters include the following (unless varied): \( K (= 2000) \), \( C (= 2) \), \( Q (= 0.4) \), \( p_{on}^{\text{eff}} (= 0.7) \), \( p_{off}^{\text{eff}} (= 0.9) \), \( \delta_1 (= 1) \), \( \delta_2 (= 4) \), and \( \delta_3 (= 6) \). For the two-, three-, and four-timer and MIMD policies, the performance corresponding to the best choice of active interval and sleep interval was recorded. For instance, for the above mentioned parameters the energy balancing choices of active interval and sleep interval is as follows: 1

- Two-Timer policy: \( SI^\ast = 19 \) and \( AI^\ast = 20 \).
- Three-Timer policy: \( SI_{OFF}^\ast = 19 \), \( SI_{ON}^\ast = 1 \), and \( AI^\ast = 6 \).
- Four-Timer policy: \( SI_{OFF}^\ast = 26 \), \( SI_{ON}^\ast = 1 \), \( AI_{OFF}^\ast = 9 \), \( AI_{ON}^\ast = 1 \).
- MIMD policy: \( SI_{OFF}^\ast = 39 \), \( SI_{ON}^\ast = 1 \), and \( AI^\ast = 15 \).

<table>
<thead>
<tr>
<th>Table 5.1 – Performance for varying ( K )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Policy</td>
</tr>
<tr>
<td>--------</td>
</tr>
<tr>
<td>AW</td>
</tr>
<tr>
<td>Two Timer</td>
</tr>
<tr>
<td>Three Timer</td>
</tr>
<tr>
<td>Four Timer</td>
</tr>
<tr>
<td>MIMD</td>
</tr>
</tbody>
</table>

Tables 5.1 and 5.2 show the performance of each policy with varying bucket size(\( K \)), and probability of recharge (\( Q \)), respectively. In both the cases, correlation probabilities were set to \( p_{on}^{\text{eff}} = 0.9 \) and \( p_{off}^{\text{eff}} = 0.7 \). Table 5.1 compares the performance of each policy with the change in bucket size, \( K \) with \( Q = 0.4 \). It may be observed that for a smaller bucket size, the MIMD policy seems to perform better than the others, due to its adaptive nature. However, with a sufficiently

\[ \text{Performance of each policy for these optimal values is as listed for } K = 2000 \text{ in Table 5.1.} \]
Table 5.2 – Performance for varying $Q$

<table>
<thead>
<tr>
<th>Policy</th>
<th>$Q = 0.2$</th>
<th>$Q = 0.5$</th>
<th>$Q = 0.6$</th>
</tr>
</thead>
<tbody>
<tr>
<td>AW</td>
<td>0.1056</td>
<td>0.2822</td>
<td>0.3479</td>
</tr>
<tr>
<td>Two Timer</td>
<td>0.1195</td>
<td>0.2998</td>
<td>0.3745</td>
</tr>
<tr>
<td>Three Timer</td>
<td>0.1283</td>
<td>0.3396</td>
<td>0.4069</td>
</tr>
<tr>
<td>Four Timer</td>
<td>0.1283</td>
<td>0.3433</td>
<td>0.4112</td>
</tr>
<tr>
<td>MIMD</td>
<td>0.1333</td>
<td>0.3404</td>
<td>0.4073</td>
</tr>
</tbody>
</table>

A large bucket size of $K = 2000$, the four-timer policy performs better than the rest. Similarly, Table 5.2 compares the performance of each policy with the increase in probability of recharge, $Q$, with a fixed bucket size $K = 2000$. With a lower recharge rate, the sensor is bound to sleep more to conserve energy and thus the performance is low. As the recharge rate increases, the sensor spends more time in the active state and is able to detect greater number of transitions. With a particular recharge rate, again the four-timer policy performs better than the other. For $Q = 0.2$, the MIMD policy performs better than the four-timer policy.

Figure 5.1 – Comparison of performance of each policy

Figure 5.1 shows the performance achieved by each of the policies for varying correlation probabilities. For varying $p_c^\text{on}$ with $p_c^\text{off} = 0.9$, it can be observed that the four-timer policy performs better than the other policies, although for lower values of $p_c^\text{on}$, there is a marginal dif-
ference between the four timer and the MIMD policy. Similarly, varying $p_c^{off}$ with fixed $p_c^{on} = 0.7$, the four-timer policy achieves higher performance compared to other policies, although not by a large margin. In general, the performance trend is as follows: $AW < Two-Timer < Three-Timer < MIMD < Four-Timer$.

5.8 Summary

This chapter discusses the different classes of activation policies and compared various policies to determine the best policy $\Pi$ with best performance. Under varying system parameters, the performance of each policy was recorded and compared. After detailed analysis and comparison, it became clear that the four-timer policy performs better in most realistic and practical scenarios.
Chapter 6

FOUR-TIMER POLICY

Chapter 5 discussed the different policies and compared their performances. The four-timer policy exhibiting TOVO property seems to perform better than the other evaluated policies. This chapter discusses in detail, the four-timer policy. In this chapter, the four-timer policy is represented by $\Pi$. Section 6.1 describes the policy’s structure as a Markov decision process (MDP). Section 6.2 discusses the Markov chain model of the policy $\Pi$, which is further used to determine the bounds and evaluate performance of the policy, $P(\Pi)$.

6.1 Structure of Policy $\Pi$

The entire operation of the sensor under the four-timer policy is formulated as an MDP, and the state space, set of actions, state-transition probabilities, and reward function are defined.

6.1.1 State Space and Actions

The 4-tuple state of the system at any time $t$ is defined by, $X_t = (EventProcess, SensorState, SleepTime, ActiveTime)$ where

- **EventProcess** is the current state of the event process, ON or OFF.
• **SensorState** is the current state of the sensor. For analysis purposes, it is assumed that the sensor will always have enough energy to keep it from dying. Hence, for analysis, the sensor is in either the active or sleep state.

• **SleepTime** is the residual sleep time for the sensor before activating (if sensor state is inactive).

• **ActiveTime** is the residual active time for the sensor before deactivating (if sensor state is active).

The decisions made by the sensor are referred to as “actions” taken by the sensor. The action taken by the sensor is either activation or deactivation, thus changing the state of the sensor and the system. Hence, the sensor state mentioned above represents the action taken at the end of time slot \( t-1 \) for time slot \( t \). The possible set of actions taken at the end of time slot \( t \) may be represented as \( u_t \in \{A, I\} \) or \( u_t \in \{1, 0\} \). Here, \( u_t = 1/A \) (\( u_t = 0/I \)) corresponds to sensor activation (deactivation) during time slot \( t \). Throughout the analysis due to the TOVO property, \( S_{I_{ON}} \) and \( A_{I_{ON}} \) have been considered to be equal to one, and denoted \( S_{I_{OFF}}(A_{I_{OFF}}) \) as \( S(I,A) \). The total number of states for any given pair of SI and AI, \((SI,AI)\), will be equal to \((2SI)+AI\). However, this is not true for the case when \( SI=1 \). From the state transition probabilities in Section 6.1.2, it may be understood that when \( SI=1 \), there will be \( SI+AI+2 \) number of states. With these considerations, a suitable state space looks like the following:

\[
X = \{(OFF, I, SI, 0), \cdots , (OFF, I, 1, 0), (ON, I, SI - 1, 0), \cdots , (ON, I, 1, 0), \\
(ON, A, 0, 1), (OFF, A, 0, AI), \cdots , (OFF, A, 0, 1)\}
\]

If in time slot \( t \) the state of the system is \( X_t = (OFF, I, 5, 0) \), then the event process in this time slot is OFF, and the sensor will remain inactive for five time slots. Similarly, \( X_t = (ON, I, 5, 0) \) means that the event process is ON, and the sensor will remain inactive for five more time slots. This may be confusing since due to the TOVO property if the event process is ON, ideally the
sensor must sleep for one time slot only. If the sensor is deactivated when the event process is OFF, and if a transition occurred while the sensor remained inactive, then the system is in such a state. In other words, the state \( X_t = (ON, I, 5, 0) \) means that the event process is OFF when the sensor decided to deactivate, but a transition occurred after some time.

### 6.1.2 State-Transition Probabilities

Each state transits to another state with a certain probability, defined as state-transition probability. Given a state \( x_{t-1} \), the state-transition probability of reaching state \( x_t \) upon a certain action \( u_t \) is given by, \( Pr[x_t|u_t, x_{t-1}] \), where \( x_t, x_{t-1} \in X \) and \( u_t \in [0, 1] \). For the sensor model discussed in this problem, \( p_{c}^{off} \) and \( p_{c}^{on} \) are the correlation probabilities that decide the state transition probability. As an example, assume the sensor decides to deactivate and sleep for five time slots during the OFF period. So at time \( t - 1 \), the state is \( x_{t-1} = (OFF, I, 5, 0) \) and \( u_t = 0 \). The next state \( x_t \) is either \( (OFF, I, 4, 0) \) with a probability of \( p_{c}^{off} \) or \( (ON, I, 4, 0) \) with a probability of \( 1 - p_{c}^{off} \) (transition occurred in time slot \( t - 1 \)). Similarly, if \( x_{t-1} = (ON, A, 0, 1) \) the next state is \( x_t = (ON, I, 1, 0) \), with probability \( p_{c}^{on} \), or \( x_t = (OFF, I, SI, 0) \), with probability \( 1 - p_{c}^{on} \). In the similar lines the complete transition probability matrix is constructed.

### 6.1.3 Reward Function

Let \( E_t \) denote the event process in the time slot \( t \). Thus \( E_t = 1 \) (0) refers to the ON (OFF) period. The decision made or action taken at the end of time slot \( t-1 \) for time slot \( t \) is denoted by \( u_t \in [1, 0] \), where \( u_t=1(0) \) refers to activation (deactivation). The reward function \( r(E_t, u_t) \), as per Section 2.1.1 in [10], denotes the reward obtained for making a decision \( u_t \) with the event process as \( E_t \) state in time slot \( t \). If the decision made at the end of time slot \( t-1 \) is to deactivate the sensor, then it is assumed that the sensor will remain inactive for the next \( t' \) number of time slots, i.e., \( u_t = 0, u_{t+1} = 0, \ldots, u_{t+t'-1} = 0 \) and \( u_{t+t'} = 1 \). The reward function may be mathematically represented as
Rewards are computed similarly for late transmissions. The reward is 1 if the sensor is active at the time of a transition. In case the sensor was inactive at the time of transition, the reward is a maximum of $e^{-1}$ if the transition occurred one time slot before the sensor was activated. The minimum reward is $e^{-t'}$, which means that the transition occurred as soon as the sensor deactivated in time slot $t$. These minimum and maximum values are further used to compute the lower bound and upper bound on the performance, respectively. The steady state expected reward per transition is $r_t = \sum \pi(i)r(i)$, where $r_t$ is the expected reward for one transition in time slot $t$, $\pi(i)$ is the steady-state probability of state $i$, and $r(i)$ is the reward obtained being in state $i$, Section 4.5 [11].

6.2 Markov Chain Modeling of Policy $\Pi$

The analysis of the sensor activation/deactivation process was performed with the help of Markov chain modeling. The concept of discrete time Markov chains was leveraged to help us analyze the sensor operation. Figure 6.1 illustrates the four-timer policy as a Markov chain with the help of a state transition diagram. This Markov chain helps in determining the steady-state probabilities that are further used to calculate the upper bound and lower bound of the performance achieved. Using tools available in Matlab, the Markov chain was solved to obtain the values for steady-state probabilities.

6.2.1 Steady State Probability

The steady-state probability of each state of the Markov chain in Figure 6.1 is denoted by $\pi^i$, where $i \in X$ and $\pi^i$ is the solution to the Markov chain for the best four-timer policy. With the help of discrete time Markov chain analysis technique, the steady state probability of each state is represented as given below. The steady state probability of each state is computed using Matlab.
tools and are further used to compute the upper bound and lower bound of performance.

\[
\begin{align*}
\pi^{(OFF,SI,0)} &= (p_c^{off}) \times \pi^{(OFF,A,1)} + (1 - p_c^{on}) \times \pi^{(ON,A,1)} \\
\pi^{(OFF,SI-1,0)} &= (p_c^{off}) \times \pi^{(OFF,SI,1)}, \text{ if } SI = 1 \\
\pi^{(ON,SI-1,0)} &= (p_c^{off}) \times \pi^{(OFF,SI,0)}, \text{ if } SI! = 1 \\
\pi^{(OFF,SI-2,0)} &= (p_c^{off}) \times \pi^{(OFF,SI-1,0)} + (1 - p_c^{on}) \times \pi^{(ON,SI-1,0)} \\
\pi^{(ON,SI-2,0)} &= (1 - p_c^{off}) \times \pi^{(OFF,SI-1,0)} + (p_c^{on}) \times \pi^{(ON,SI-1,0)} \\
\pi^{(OFF,A,0, AI)} &= (p_c^{off}) \times \pi^{(OFF,A, 1)} + (1 - p_c^{on}) \times \pi^{(ON,1)} \\
\pi^{(OFF,A,0, AI-2)} &= (p_c^{off}) \times \pi^{(OFF,A,0, AI-1)} \\
\pi^{(ON,A,0,1)} &= (1 - p_c^{off}) \times \pi^{(OFF,I,1)} + (p_c^{on}) \times \pi^{(ON,I,1)} \\
\pi^{(ON,I,1,0)} &= (1 - p_c^{off}) \times \pi^{(OFF,I,2)} + (p_c^{on}) \times \pi^{(ON,I,2)} + (p_c^{on}) \times \pi^{(ON,0,1)} + (1 - p_c^{off}) \times \pi^{(OFF,0, AI)} + (1 - p_c^{off}) \times \pi^{(OFF,0, AI-1)} + \cdots + (1 - p_c^{off}) \times \pi^{(OFF,0,1)}, \text{ if } SI > 2 \\
\pi^{(ON,I,1,0)} &= (1 - p_c^{off}) \times \pi^{(OFF,I,0)} + (p_c^{on}) \times \pi^{(ON,A,0,1)} + (1 - p_c^{off}) \times \pi^{(OFF,A,0, AI)} + (1 - p_c^{off}) \times \pi^{(OFF,A,0, AI-1)} + \cdots + (1 - p_c^{off}) \times \pi^{(OFF,A,0,1)}, \text{ if } SI = 2 \\
\pi^{(ON,I,1,0)} &= (1 - p_c^{off}) \times \pi^{(OFF,A,0, AI-1)} + \cdots + (1 - p_c^{off}) \times \pi^{(OFF,A,0,1)}, \text{ if } SI < 2
\end{align*}
\]

Figure 6.1 – Sensor activation/deactivation as a Markov chain
6.2.2 Performance Evaluation

As discussed in Section 4.5.4, the performance of policy $P(\Pi)$ is referred to as reward per transition, which is the ratio of the total reward obtained after transmitting transitions and the total number of transitions that occurred. The performance of policy $\Pi$ is discussed in equation (4.3) and is given by

$$P(\Pi) = \lim_{T \to \infty} \frac{\Upsilon_s(T) \times 1 + \sum_\lambda e^{-\lambda t}}{\Upsilon_o(T)} \quad (6.2)$$

where $\Upsilon_o(T)$ is the number of transitions occurring in the given time interval $\Upsilon_s(T)$ that are actively sensed and transmitted and thus fetching a reward of 1, and $l$ is the number of transitions that are realized and transmitted late. The steady state expected reward per transition is given by $r_t = \sum_i \pi^i r(i)$, where $\pi^i$ is the steady-state probability of state $i$, and $r(i)$ is the reward obtained being in state $i$.

In an interval $[1, \cdots, T]$ there may be many transitions, then the total reward fetched is $r_t \times T$. For analysis purposes one cycle of a complete ON and OFF period is considered as shown in Figure 6.2. The expected duration of the ON period is given by $E[ON] = \frac{1}{(1 - p^{on})}$ and for the OFF period $E[OFF] = \frac{1}{(1 - p^{off})}$. Hence, the duration of one cycle equals the sum of the expected length of the ON period and expected length of the OFF period or $\tau = E[ON] + E[OFF]$.

![Figure 6.2 – Expected length of ON and OFF period](image)

Let $T$ be the time when the system reaches steady state. Let $N = \frac{T}{\tau}$ be the number of times the cycle, with a period of $\tau$, repeats in interval $[1, \cdots, T]$. In one cycle, of period $\tau$, the sensor observes two transitions, one from ON to OFF and the other from OFF to ON. The total number of transitions for $N$ such cycles is given by $2 \times N$. In other words, during the entire period of $T$, the
system witnesses $2N$ number of transitions. Since this cycle forms a renewal interval (Section 2.1 [12]), the reward per transition for capturing the transition ON to OFF by an active sensor in state $(ON, A, 0, 1)$ is

$$\frac{r_t \times T}{2 \times N} = \frac{\pi^{(ON,A,0,1)} \times (1 - p_{on}^{\tau}) \times 1 \times T}{2 \times \frac{\tau}{2}} = \frac{\pi^{(ON,A,0,1)} \times (1 - p_{on}^{\tau}) \times \tau}{2} \quad (6.3)$$

Extending the above example to all possible transitions that occur in the system, the reward per transition is computed, as shown in equation (6.4). For this we consider the probability of sensing and transmitting a transition given the system state $i$, reward obtained after transmission, and total number of transitions that occurred.

$$P(\Pi) = \sum_i \pi^i r(i) \times \frac{\tau}{2} \quad (6.4)$$

With the help of the state space of the Markov chain and equation (6.4) we now determine the upper bound and lower bound on the performance. These two bounds are very useful in analyzing the performance of the sensor, based on the policy $\Pi$ designed in this paper.

### 6.2.3 Upper Bound on $P(\Pi)$

The upper bound (UB) refers to the maximum performance that can be achieved by a system under policy $\Pi$. Since this refers to the best-case scenario, we assume that the sensor never misses a transition, and the reward fetched for any transition transmitted is equal to 1. In other words, the reward for late transmission equals $e^0 = 1$. Accordingly, there are three different equations for the upper bound, for $SI > 2$, $SI = 2$, and $SI < 2$, respectively, depending on the state space and state transition probability matrix:
6.2.4 Lower Bound on \( P(\Pi) \)

Along with the upper bound, the lower bound (LB) helps in bounding the performance of policy \( P(\Pi) \). The lower bound is the lowest possible performance result achieved by policy \( P(\Pi) \). The LB on any performance is considered to be a worst-case scenario and here the worst may happen when the sensor takes a maximum amount of time in realizing a missed transition. The maximum time that may be taken by the inactive sensor to realize a missed transition is the sleep interval itself, which means that a transition occurred as soon as the sensor decided to sleep. Therefore, the reward per transition is \( r_t = \sum_i \pi^i e^{-SI} \). Like the UB, there are three different equations for the LB:

\[
UB(\Pi)_{SI>2} = \pi^{(ON,A,0,1)}(1 - p_c^{on})(1 + p_c^{on}) + \sum_{t=1}^{AI} \pi^{(OFF,A,0,t)}(1 - p_c^{off})(1 + (1 - p_c^{on})) + \{\pi^{(OFF,I,1,0)}(1 - p_c^{off}) + \pi^{(OFF,I,2,0)}(1 - p_c^{off})(1 - p_c^{on}) + \pi^{(ON,I,2,0)}(p_c^{on})^2\} \frac{\tau}{2}
\]

\[
UB(\Pi)_{SI=2} = \pi^{(ON,A,0,1)}(1 - p_c^{on})(1 + p_c^{on}) + \sum_{t=1}^{AI} \pi^{(OFF,A,0,t)}(1 - p_c^{off})(1 + (1 - p_c^{on})) + \{\pi^{(OFF,I,1,0)}(1 - p_c^{off}) + \pi^{(OFF,I,2,0)}(1 - p_c^{off})(1 - p_c^{on})\} \frac{\tau}{2}
\]

\[
UB(\Pi)_{SI<2} = \pi^{(ON,A,0,1)}(1 - p_c^{on})(1 + p_c^{on}) + \sum_{t=1}^{AI} \pi^{(OFF,A,0,t)}(1 - p_c^{off})(1 + (1 - p_c^{on})) + \pi^{(OFF,I,1,0)}(1 - p_c^{off}) \frac{\tau}{2}
\]

\[
(6.5)
\]
\[ LB(\Pi)_{SI<2} = \left[ \pi^{(ON,A,0,1)}(1 - p_{on}^c)(1 + p_{on}^c e^{-1}) \right. + \sum_{t=1}^{AI} \pi^{(OFF,A,0,t)}(1 - p_{off}^c)(1 + (1 - p_{on}^c)e^{-1}) \]
\[ + \left. \pi^{(OFF,I,1,0)}(1 - p_{off}^c)e^{-SI} \right]\left( 1 + (1 - p_{on}^c)e^{-1} \right) \frac{\tau}{2} \]

(6.6)

6.2.5 Discussion on Performance of Policy \( \Pi \)

The four-timer policy was simulated for two different cases: (i) constant \( SI_{OFF} \) but varying \( AI_{OFF} \), and (ii) constant \( AI_{OFF} \) but varying \( SI_{OFF} \). In addition to this, with the help of Matlab, the upper and lower bounds on performance were obtained in all cases. The performance results obtained with the help of simulation were plotted using Matlab along with the two bounds for comparison. The plots were made with system parameters \( K = 2000, C = 2, Q = 0.4, p_{on}^c = 0.7, p_{off}^c = 0.9, \delta_1 = 1, \delta_2 = 4, \delta_3 = 6 \). Figure 6.3 shows the comparison results. From the plots it may be seen that the policy performs best for a particular pair of \( SI_{OFF} \) and \( AI_{OFF} \) denoted by \([SI^*, AI^*] = [26, 9]\). For any sleep interval, \( SI_{OFF} > SI^* \), the sensor will start missing more transitions and thus the performance decreases. On the other hand, an increase in \( AI_{OFF} > AI^* \) with \( SI_{OFF} = SI^* \) does not change the performance much, due to preemptive sleeping caused by a transition detection during the active interval.

![Figure 6.3](image-url)

**Figure 6.3** – Performance for \( p_{on}^c = 0.7 \) and \( p_{off}^c = 0.9 \)

45
Figure 6.4 shows the variation in steady-state probability of the sensor in the dead state, $\pi^{\text{dead}}$ (computed using DES), with respect to $SI_{OFF}$ and $AI_{OFF}$, respectively. With the increase in $SI_{OFF}$ at a constant value of $AI_{OFF}$, $\pi^{\text{dead}}$ appears to be significant until $SI_{OFF} = SI^*$. This is true since the sensor has been over utilizing energy for values of $SI_{OFF} < SI^*$. When $SI_{OFF} = SI^*$, the sensor attains energy balance, and thus the value of $\pi^{\text{dead}}$ becomes almost negligible, and any further increase in $SI_{OFF} > SI^*$ will ensure efficient energy utilization and the value of $\pi^{\text{dead}} = 0$. Similarly, with the increase in $AI_{OFF}$ at a constant value of $SI_{OFF}$, we notice $\pi^{\text{dead}}$ to be zero until $AI_{OFF} = AI^*$. The sensor attains energy balance at the point when $AI_{OFF} = AI^*$, and any increase in $AI_{OFF} > AI^*$ will force the sensor to reach the dead state, thus making $\pi^{\text{dead}} > 0$ since excess energy may be consumed by the longer duration of an active interval. The value of $\pi^{\text{dead}}$ equals zero for values of $AI_{OFF} \leq AI^*$ and $SI_{OFF} \geq SI^*$.  

\[ \text{Figure 6.4} - \text{Steady state probability of dead state for } p_c^{on} = 0.7 \text{ and } p_c^{off} = 0.9 \]

Figure 6.5 shows variation in the residual energy level of the sensor in two different scenarios. When the sensor operates with values of $SI_{OFF}$ and $AI_{OFF}$, equal to $SI^*$ and $AI^*$, respectively, the energy balance condition is observed as sensor operates over time and hence, sensor never

\[ ^1 \text{In the Markov chain analysis, to derive the upper and lower bounds, } \pi^{\text{dead}} \text{ has not been considered. Therefore, the UB and LB hold true if } \pi^{\text{dead}} = 0 \text{ or when } AI_{OFF} \leq AI^* \text{ and } SI_{OFF} = SI^* \text{ or when } SI_{OFF} \geq SI^* \text{ and } AI_{OFF} = AI^*, \text{ as depicted in Figure 6.3.} \]
dies. When the sensor does not operate with optimal values, for instance $SI_{OFF} < SI^*$ and $AI_{OFF} > AI^*$, the sensor may be observed to be unable to operate efficiently. More than half of the energy available is consumed in a short period of time. From this point onwards, the sensor will never recover and needs to operate with the minimal energy available. From this comparison, it may be concluded that the sensor operates efficiently and attains energy balance for the pair, $[SI^*, AI^*]$.

For better understanding of the policy $\Pi$, the performance and the bounds for a different pair of correlation probabilities are plotted. Figure 6.6 shows plots for $p_{\epsilon}^{off} = 0.8$ and $p_{\epsilon}^{on} = 0.69$. It can be observed that the TOVO property holds for values of $p_{\epsilon}^{off} \geq p_{\epsilon}^{on} \geq 0.69$. For this pair of correlation probabilities, the optimal values of sleep interval and active interval were observed to be $[SI^*, AI^*] = [51, 20]$. Figure 6.7 shows the plot of the variation of steady-state probability of the dead state with the change in sleep interval and active interval. Also, Figure 6.8 shows the plot for the residual energy level of the sensor for two different scenarios and shown the energy balancing choices of $SI$ and $AI$. 

Figure 6.5 – Variation of residual energy level for $p_{\epsilon}^{on} = 0.7$ and $p_{\epsilon}^{off} = 0.9$
Figure 6.6 – Performance for $p_{c}^{on} = 0.69$ and $p_{c}^{off} = 0.8$

Figure 6.7 – Steady state probability of dead state for $p_{c}^{on} = 0.69$ and $p_{c}^{off} = 0.8$
In this chapter the entire structure of the four-timer policy was discussed. It was observed that the TOVO property holds good for various system parameters. The policy was analyzed mathematically with the help of Markov chain analysis techniques. Further, the upper bound and lower bound of the achievable performance were determined to evaluate the performance of the four-timer policy. Note that the performance of the policy holds between the derived bounds but possesses a loose upper bound and a tight lower bound.

6.3 Summary

Figure 6.8 – Variation of residual energy level for $p_{c}^{on} = 0.69$ and $p_{c}^{off} = 0.8$
Chapter 7

CONCLUSION

The major problem witnessed in any wireless sensor node is the small energy reserve and expected longevity of its operation. Sensors may be deployed in a network of wireless sensors to monitor certain natural and interesting events occurring over time that exhibit certain degree of temporal correlation. Although these sensors have evolved to include rechargeable batteries, limited energy reserves restrict them from operating continuously and thus lower the performance. Therefore, there is a need for an efficient operational policy that helps the sensor optimally utilize the energy available and perform better. Since there is randomness seen in the event-occurrence process, the discharge of energy in a sensor is random and has a higher rate when compared to the rate of recharge. These issues along with the lower computational capability of the sensor pose a challenging task for developing an efficient and simple decision-making policy. With an understanding of the characteristics of a sensor a novel and intuitive policy was developed to achieve high performance with efficient utilization of the energy available. This policy further provides an answer to the questions “How long should the sensor sleep, and for how long should the sensor stay active?”

This thesis considers the operation of a solitary sensor node that was assigned a job to detect transitions or changes in the state of the event process and transmit this information to a monitoring station or base station. With the help of transition detection, redundant data transmissions are
avoided, thus minimizing energy expenditure. An active sensor transmits an observed transition but an inactive sensor misses the transition, and realizes this only when it is activated and transmits the same. In the latter case, the sensor transmits the missed transition late but, in the process, adds value to the performance since late transmission is better than no transmission. In either case, a suitable reward is obtained after each transmission. If the sensor actively observes and transmits a transition, then the reward is 1, and in the case where the sensor is late in realizing and transmitting a missed transition, the reward is reduced and proportional to the time taken to do so. Hence, the performance of the sensor is defined on the basis of the total reward obtained after several transmissions over a period of time. The goal of this research was to maximize the reward per transition or performance with the help of an appropriate decision-making policy.

Different policies were evaluated, and the best performing policy was concluded to be the four-timer policy. This policy demonstrated a distinct property, known as TOVO (Toggle for ON and Vary for OFF), for a range of system parameters. The policy was mathematically analyzed to determine the bounds on its performance. Part of this research work developed the scope for avoiding redundant data transmissions by sensing only the transitions between the two states of the event process. After comparing several intuitive policies a simple and best decision-making policy for the sensor was proposed, to achieve high performance with efficient utilization of energy available.

7.1 Future Scope

Since the mathematical analysis was performed on the basis of numerical computations, the future work will be to derive appropriate expressions, that may be further solved for the various variables considered, in the four-timer policy. Some of the listed future work related to the mathematical analysis of the entire problem are as follows:

- Equations may be derived to solve for optimal, and energy balancing values of sleep intervals, and active intervals for the four-timer policy.
• Equations may be derived to represent the upper and lower bounds of the achievable performance for any given value of sleep interval, and active interval.

• Each of the evaluated policies may be modeled, and solved as a Markov Decision Process (MDP), in order to determine the optimal decision-making policy that achieves best performance by maximizing the reward per transitions.

Since the complete problem was isolated to a single sensor node, future work may include implementing this algorithm in a network of sensors to solve coverage and connectivity issues. Various other adaptive policies are worth considering since the multiplicative increase and multiplicative decrease policy achieved marginally lower performance than the four-timer policy. Since this work only considered dynamic calculation of the active interval, future work may include the dynamic calculation of the sleep interval.
Bibliography


APPENDIX
Simulation of Sensor Activation/Deactivation Process for the Four-Timer policy

#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <time.h>

double PcOn = 0.7;
double PcOff = 0.9;
/* PcOn and PcOff are the correlation probabilities */
double Q = 0.4; /* Probability of Recharge */
int K = 2000; /* Energy Bucket Size */
double SimulationTime = 1000000;
/* Set the upper limits for the four timers */
int MaxAI_ON = 10;
int MaxSleepWhenON = 10;
int MaxAI_OFF = 10;
int MaxSleepWhenOFF = 10;
int ActiveInterval; /* Duration, sensor decides to stay active */
int SleepInterval; /* Duration, sensor decides to sleep */
int AI_ON, AI_OFF; /* AI_ON/AI_OFF-Active Interval for ON/OFF period. */
int SI_ON, SI_OFF; /* SI_ON/SI_OFF-Sleep Interval for ON/OFF period. */
int ActiveTime; /* Residual active time. */
int SleepTime; /* Residual sleeping time. */
int RechargeQty; /* Amount of charge added during a recharge event */
int EnergyLevel; /* Sensor Energy Level. */
int EnergyThreshold; /* Energy Level required for the sensor to activate. */
int Delta1, Delta2, Delta3;
/* Delta1 is the energy spent by the sensor when Active. */
/* Delta2 is the energy spent when the sensor sees the Event process = 1 (ON). */
/* Delta3 is the energy spent during transmission, of the acquired data, by the sensor. */
int SensorState; /* SensorState=0(Dead),=1(Sleeping),=2(Active) */
double Time; /* Current Simulation Time */
long int TimeOfLastTransition;
int TransitionOccurred;
/* Toggles to 1 if transition occurred. */
/* Toggles to 0 after transmission. */
long int NumOfTransitionOccurred, NumOfTransitionCaptured;
double RewardForLateTransmission;
/* This is the reward for transmitting a missed transition. */
int EventProcess;
/* EventProcess = 0 (OFF period) and EventProcess = 1 (ON period). */
int One2Zero; /* One2Zero = 0 indicates transition from 0 to 1. */
/* One2Zero = 1 indicates that the next possible transition will be from 1 to 0. */
/* Observation made before deactivating */
void initialize();
void recharge();
void FourTimerPolicy();
void checkMissedTransition();
void processing();

main ()
{
  int i, j, k, l;
  int Val_i, Val_j, Val_k, Val_l;
  double Performance, TotalReward, BestPerformance=0.0;
  switch (DecisionPolicy)
  {
  case (3) :
    for (i = 1; i <= MaxAI_ON; i++) {
      for (j = 1; j <= MaxSleepWhenON; j++) {
        for (k = 1; k <= MaxAI_OFF; k++) {
          for (l = 1; l <= MaxSleepWhenOFF; l++) {
            AI_ON = i;
            SI_ON = j;
            AI_OFF = k;
            ...
SI_OFF = 1;
Performance = 0.0;
initialize(); /* Set every variable to respective initial values */
do
{
    recharge(); /* Quantum Recharge */
    processing(); /* Process each time slot to observe transition/transmit */
    ++Time;
    FourTimerPolicy(); /* Decision made for the next time slot */
}while (Time < SimulationTime);
TotalReward = (double) NumOfTransitionCaptured + RewardForLateTransmission;
Performance = (double) (TotalReward)/NumOfTransitionOccurred;
if (BestPerformance < Performance)
{
    BestPerformance = Performance;
    Val_i = i;
    Val_j = j;
    Val_k = k;
    Val_l = l;
} /* (end if) */
} /* (end for loop l) */
} /* (end for loop k) */
} /* (end for loop j) */
} /* (end for loop i) */
printf(" AI_ON = %d; SI_ON = %d; AI_OFF = %d; SI_OFF = %d", Val_i, Val_j, Val_k, Val_l);
printf("Best Performance : %1.4f", BestPerformance);
break;
} /* (end switch case) */
} /* (end main()) */

void initialize()
{
srand48(9); /* srand seeds the random number generator. */
Time = 1; /* Start simulating from first time slot, t=1 */
EnergyLevel = K; /* Energy Bucket is completely charged. */
Delta1 = 1;
Delta2 = 4;
Delta3 = 6;
EnergyThreshold = Delta1+Delta2+(2*Delta3);
RechargeQty = 2;
SensorState = 2; /* SensorState initialized to be Active. */
if (drand48() > 0.5) /* This will determine the initial state of EventProcess. */
{
    EventProcess = 1;
    ActiveTime = AL_ON;
}
else
{
    EventProcess = 0;
    ActiveTime = AL_OFF;
}
One2Zero = EventProcess;
/* Sensor knows the current event process and expects corresponding transition. */
/* Reset all timers and counters before simulation begins. */
TimeOfLastTransition = 0;
TransitionOccurred = 0;
NumOfTransitionOccurred = 0;
NumOfTransitionCaptured = 0;
RewardForLateTransmission = 0.0;
return;
}

/* The sensor will be recharged in the current time slot with a probability Q. */
void recharge()
{
    if (drand48() <= Q)
EnergyLevel += RechargeQty;
if (EnergyLevel > K)
    EnergyLevel = K;
return;
}
return;

/* Sensor is Active during ON/OFF period for AI.ON/AI.OFF duration.
/* Sensor decides to sleep for SI.ON/SI.OFF duration if:
(i) ActiveTime = 0, or (ii) Transition Sensed, or (iii) EnergyLevel < Threshold. */
void FourTimerPolicy()
{
    if (SensorState == 2)
    {
        --ActiveTime;
        if (ActiveTime == 0 || TransitionOccurred == 1 || EnergyLevel <= EnergyThreshold)
        {
            TransitionOccurred = 0;
            ActiveTime = 0;
            SensorState = 1;
            if (One2Zero == 1)
                SleepTime = SI.ON;
            else
                SleepTime = SI.OFF;
        }
    }
    if (EnergyLevel <= EnergyThreshold)
    {
        if (EnergyLevel < 0)
        {
            printf("Error!!(1)\n");
        }
exit(0);
}
if (SensorState == 1)
{
--SleepTime;
if (SleepTime == 0)
SensorState = 0;
else if (SleepTime < 0)
{
printf("Error!!(2)");  
exit(1);
} return;
}
if (SensorState != 0)
{
printf("Error!!(3) Sensor should have been dead...");  
exit(2);
} return;
}
if (SensorState == 1)
{
--SleepTime;
if (SleepTime == 0)
{
SensorState = 2;
checkMissedTransition();
if (One2Zero == 1)
ActiveTime = AL\_ON;
else
ActiveTime = AL\_OFF;
return;
}
if (SleepTime < 0)
printf("Error!!(4)\n");
exit(3);
} return;

/* Aggressive wake up. */
if (SensorState == 0)
{
  SensorState = 2;
  ++NumOfDeadSlots;
  checkMissedTransition();
  if (One2Zero == 1)
    ActiveTime = AI_ON;
  else
    ActiveTime = AI_OFF;
  return;
}

/* Check for any missed transitions during the inactive period of the sensor. */
void checkMissedTransition()
{
  if (SensorState == 2 && One2Zero != EventProcess)
  {
    One2Zero = 1 - One2Zero;
    EnergyLevel = EnergyLevel - Delta3;
    RewardForLateTransmission += exp(-(Time - TimeOfLastTransition));
  } return;
}

/* Transition occurs with a probability of 1-PcOff(1-PcOn) during OFF(ON) period. */
/* Active Sensor transmits the detected transition and loses energy. */
void processing()
{
    double p;
    if (EventProcess == 0)
    { p = 1must-PcOff; }
    else
    { p = 1-PcOn; }
    if (drand48() <= p)
    {
        ++NumOfTransitionOccurred;
        TimeOfLastTransition = Time;
        EventProcess = 1 - EventProcess;
        if (SensorState == 2)
        {
            TransitionOccurred = 1;
            One2Zero = 1 - One2Zero;
            ++NumOfTransitionCaptured;
            if (EventProcess == 1)
                EnergyLevel = EnergyLevel - (Delta1 + Delta2 + Delta3);
            else
                EnergyLevel = EnergyLevel - (Delta1 + Delta3);
        } return;
    }
    if (SensorState == 2)
    {
        if (EventProcess == 1)
            EnergyLevel = EnergyLevel - (Delta1 + Delta2);
    }
else

EnergyLevel = EnergyLevel - (Delta1);

} return;

}