ADAPTIVE ENERGY MANAGEMENT FOR SOLAR ENERGY HARVESTING WIRELESS SENSOR NODES

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF ENGINEERING AND SCIENCE

OF BILKENT UNIVERSITY

IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR

THE DEGREE OF

MASTER OF SCIENCE

IN

ELECTRICAL AND ELECTRONICS ENGINEERING

By

Abdul Kerim Aydin

September 2018

ADAPTIVE ENERGY MANAGEMENT FOR SOLAR ENERGY HARVESTING WIRELESS SENSOR NODES

By Abdul Kerim Aydin

September 2018

We certify that we have read this thesis and that in our opinion it is fully adequate, in scope and in quality, as a thesis for the degree of Master of Science.

Nail Akar(Advisor)

Sinan Gezici

Bülent Tavlı

Approved for the Graduate School of Engineering and Science:

Ezhan Karaşan
Director of the Graduate School

ABSTRACT

ADAPTIVE ENERGY MANAGEMENT FOR SOLAR ENERGY HARVESTING WIRELESS SENSOR NODES

Abdul Kerim Aydin
M.S. in Electrical and Electronics Engineering
Advisor: Nail Akar
September 2018

Wireless Sensor Networks (WSN) will have a key role in the upcoming era of the Internet of Things (IoT) as they will be forming the basis of communication infrastructure. Energy harvesting has been a widely used instrument for prolonging the battery life and enhancing the quality of service (QoS) of sensor nodes (SN). In this study, we investigate adaptive transmission policies for a solar-powered wireless sensor node which is tasked with sending status updates to a gateway as frequently as possible with energy-neutral operation constraints. On the basis of empirical data, we model the daily variations of the solar energy harvesting process with a Discrete Time Markov Chain (DTMC). When the number of states of the DTMC is increased, the harvesting process is modeled more accurately. Using the DTMC model, we formulate the energy management problem of the WSN node as a Markov Decision Process (MDP); and based on this model, we use the policy iteration algorithm to obtain optimal energy management policies so as to minimize the average Age of Information (AoI) of the corresponding status update system. We validate the effectiveness of the proposed approach using datasets belonging to two different locations with 20 years of solar radiance data.

Keywords: wireless sensor nodes, solar energy harvesting, Markov Decision Process, battery management, Discrete-Time Markov Chain, age of information, duty cycling.

ÖZET

GÜNEŞ ENERJİSİ HARMANLAYAN KABLOSUZ ALGILAMA DÜĞÜMLERİ İÇİN UYARLAMALI ENERJİ YÖNETİMİ

Abdul Kerim Aydin
Elektrik ve Elektronik Mühendisliği, Yüksek Lisans
Tez Danışmanı: Nail Akar
Eylül 2018

Kablosuz Algılayıcı Ağları (WSN) yaklaşan Nesnelerin İnterneti (IoT) çağında iletişim altyapısının temelini oluşturacağı için önemli bir rol oynayacaktır. Enerji hasadı pil ömrünü uzatmak ve algılayıcı düğümlerinin (SN) hizmet kalitesini (QoS) arttırmak için yaygın olarak kullanılan bir araç olmuştur. Bu çalışmada, enerji-nötr çalışma kısıtlamaları ile bir ağ geçidine mümkün olduğunca sık bir şekilde durum güncellemeleri göndermekle görevlendirilmiş, güneş enerjisiyle çalışan bir kablosuz algılayıcı düğümü için uyarlamalı iletim politikalarını araştırdık. Güneş enerjisi harmanlama işleminin günlük varyasyonlarını deneysel verilere dayanarak Ayrık-Zamanlı Markov Zinciri (DTMC) ile modelledik. DTMC'nin durumlarının sayısı arttığında hasat sürecinin daha doğru modellendiğini gözlemledik. DTMC modelini kullanarak, WSN düğümünün enerji yönetimi problemini Markov Karar Süreci (MDP) olarak tanımladık; ve bu modelden yola çıkarak, durum güncelleme sisteminin ortalama bilgi yaşını (AoI) en aza indirmek için en uygun enerji yönetim politikalarını elde etme amacıyla politika yineleme algoritmasını kullandık. İki farklı lokasyona ait 20 yıllık güneş ışınımı verilerini kullanarak önerilen yaklaşımın etkinliğini doğruladık.

Anahtar sözcükler: kablosuz algılayıcı düğümler, güneş enerjisi harmanlama, Markov Karar Süreci, batarya yönetimi, Ayrık-Zaman Markov Zinciri, bilgi yaşı, görev çevrimi.

Acknowledgement

First and foremost, I would like to express my most sincere gratitude to my supervisor Prof. Nail Akar for his continuous support, patience and encouragement throughout this research.

I would like to thank Prof. Sinan Gezici and Prof. Bülent Tavlı for taking part in my thesis committee and reviewing my work.

I would like to thank Dr. Gökhan Kahraman for his motivational talks and guidance he has provided me throughout this study.

I appreciate the financial support of the Scientific and Technological Research Council of Turkey (TÜBİTAK) through BİDEB 2228-A Scholarship Program.

I would like to express my appreciation to my parents for their unconditional love and continuous support.

Finally, I would like to express my deepest gratitude to my beloved wife. Her endless love and encouragement during my research was invaluable.

Contents

1 Introduction **1**

1.1 Overview 1

1.2 Literature Review 7

1.3 Thesis Outline 8

2 Background Information **10**

2.1 Markov Decision Processes 10

2.2 Dynamic Programming 14

2.2.1 Policy Evaluation 15

2.2.2 Policy Improvement 16

2.2.3 Policy Iteration 16

2.2.4 Value Iteration 18

3 Energy Management in EH-WSNs with MDP **20**

3.1 Solar Energy Harvester Model with Discrete Time Markov Chain 20

3.2 Energy Management Problem with Markov Decision Process . . . 23

4 Numerical Results 25

4.1 Model Parameters 25

4.2 Building the Model of Solar Harvesting Process 26

4.3 Model Verification 29

4.3.1 Results for Different Harvesting Models 29

4.3.2 Results Against Benchmark Policies 34

4.3.3 Solutions for Some Engineering Problems 36

5 Conclusions 40

List of Figures

1.1 General structure of a single-hop Wireless Sensor Network. 2

1.2 General form of an energy harvesting sensor node. 3

1.3 Change of age of information with time (circles shows the time of status updates). 4

2.1 Agent-environment interaction in MDP. 11

3.1 DTMC model of the harvesting process. 21

3.2 Mapping between time-slots and states. 22

4.1 Hourly average of solar radiation in the used datasets. 27

4.2 Distribution of harvested energy for $h = 1$. 28

4.3 Probability distributions of *Day* states for both locations. 29

4.4 AAoI for packet transmission energy of 1 mWh. 30

4.5 AAoI for packet transmission energy of 2 mWh. 30

4.6 AAoI for packet transmission energy of 5 mWh. 31

4.7 AAoI for packet transmission energy of 10 mWh. 31

4.8 AAoI for packet transmission energy of 20 mWh. 32

4.9 AoI and battery level variations through a test case with 500mWh battery capacity, 2mWh packet transmission energy, 30cm^2 solar panel area and 8-state harvesting model. 32

4.10 AoI and battery level variations through a test case with 500mWh battery capacity, 5mWh packet transmission energy, 20cm^2 solar panel area and 4-state harvesting model. 33

4.11 AoI and battery level variations through a test case with 500mWh battery capacity, 5mWh packet transmission energy, 30cm^2 solar panel area and 4-state harvesting model. 33

4.12 AoI and battery level variations through a test case with 800mWh battery capacity, 5mWh packet transmission energy, 30cm^2 solar panel area and 4-state harvesting model. 34

4.13 AoI-threshold based policy results for different values of battery capacity. 35

4.14 Comparison of MDP approach with a threshold-based policy. . . . 36

4.15 AAoI for different battery capacities. 37

4.16 AAoI for different values of ratio of battery capacity to packet transmission energy. 38

4.17 AAoI vs. solar panel area. 38

4.18 Change of AAoI with panel size and battery capacity. 39

Chapter 1

Introduction

1.1 Overview

New communication needs have emerged along with advancements in technology, and the Internet of Things (IoT) is one of those concepts that has arisen in recent years to meet those needs. IoT is a new paradigm in which various objects that are equipped with sensing, processing and networking capabilities, are communicating with each other or with other devices without human intervention over a public communications infrastructure to achieve a particular goal [1]. There are many different applications of the IoT concept, such as smart homes, smart cities, precision agriculture etc., all of which depend on a Wireless Sensor Network (WSN) infrastructure [2]. Figure 1.1 illustrates the general structure of a single-hop WSN. WSNs are composed of sensor nodes which can be described as low-cost, low-power devices with sensing, computation and wireless communication capabilities [3]. As wireless sensor nodes are generally relatively small-sized devices, they are equipped with a very limited power source [4]. Figure 1.2 presents the general form of an energy harvesting wireless sensor node. These nodes can be deployed in a human-controlled local area scenario or more typically in a wide area setting where human control is not likely. Therefore, in the latter situation, the nodes must be self-sufficient, i.e., not requiring any human

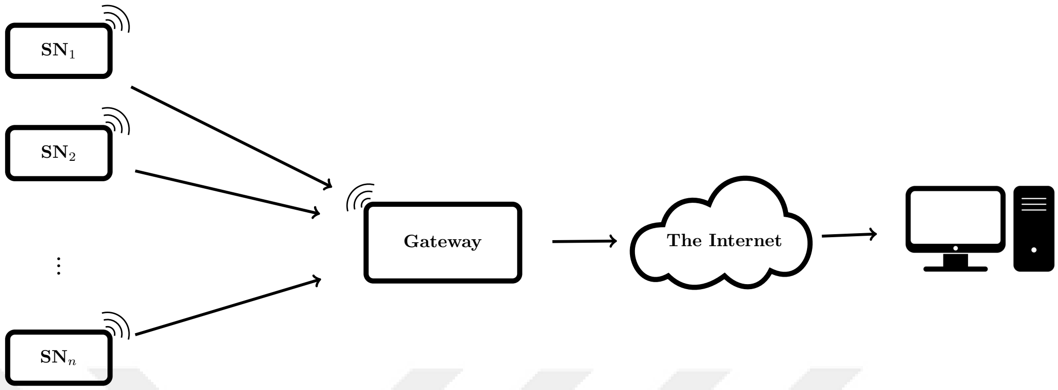

Figure 1.1: General structure of a single-hop Wireless Sensor Network.

intervention, for a relatively long time [5]. Using environmental energy sources is a promising solution that is used to build more sustainable WSNs by extending the lifetime of sensor nodes. In this solution, sensor nodes are equipped with energy harvesters that can produce energy using environmental energy sources like wind, solar radiance, vibration, etc. and store the harvested energy in the rechargeable batteries of sensor nodes [6–9].

Incorporation of energy harvesting on sensor nodes not only extends the lifetime of the node but also it provides the opportunity to make use of the harvested excessive energy in the peak hours [10]. This brings up the problem of optimal energy management in energy harvesting wireless sensor nodes (EH-WSN). For the WSN to run without any interruptions, sensor nodes must be running in the energy-neutral state, i.e., the amount of consumed energy should not exceed the amount of energy gathered [11]. Sensor nodes also need to meet certain Quality of Service (QoS) requirements. For different WSN infrastructures, various QoS requirements have been proposed, such as maintaining a reasonable age of information, lengthening the timespan until the first battery depletion, increasing the network lifetime duration, reducing network delays etc. [12]. While preserving energy-neutral operation constraints, the node also needs to take into account such QoS requirements and maximize the utilization. However, this is not an easy task due to the dynamism and randomness in the nature of renewable energy sources [13]. Various energy management algorithms have been developed in

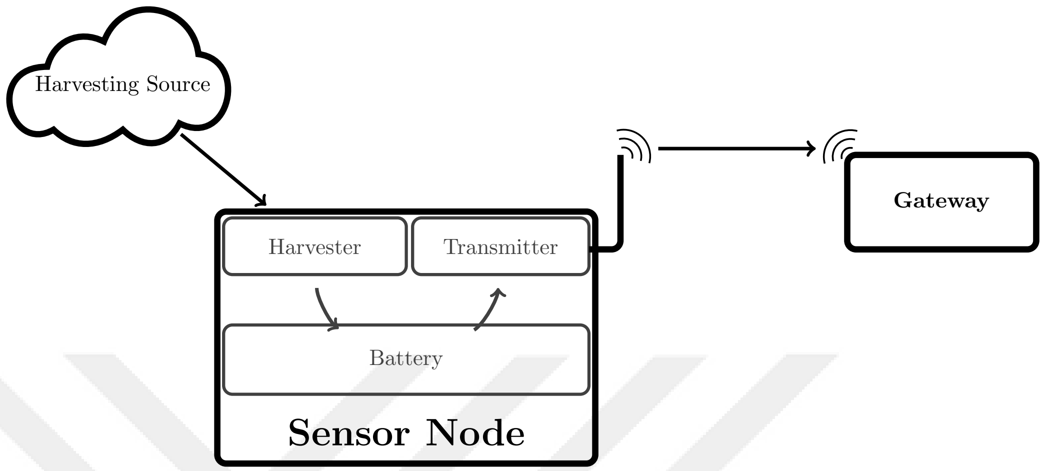

Figure 1.2: General form of an energy harvesting sensor node.

the literature for different types of energy sources and QoS requirements. In particular, adaptive duty cycling methods that adaptively adjust the inter-sensing times have been developed to maximize system performance while avoiding power failures [14–20].

Energy harvesting sources usually have a stochastic nature which can be represented with a stochastic model [20]. A subset of these models is built upon Markov Chains (MC). The tool Markov Decision Processes (MDP) is a commonly used approach for modeling the energy management problem of EH-WSNs when the underlying model uses MCs [21]. Particularly, MDPs are used for modeling WSNs to optimize sensor nodes for different objectives, e.g., sensor coverage and object detection, security, topology formulation or power optimization. In [21], benefits of using MDPs for modeling WSNs are listed as below:

- Adaptively managing power consumption to increase energy utilization,
- Balanced optimization against different objectives,
- Predicting the effects of mobility,
- Applicability of derived policies to resource-limited nodes,
- The flexibility of different variants of MDPs that can fit into different WSN applications.

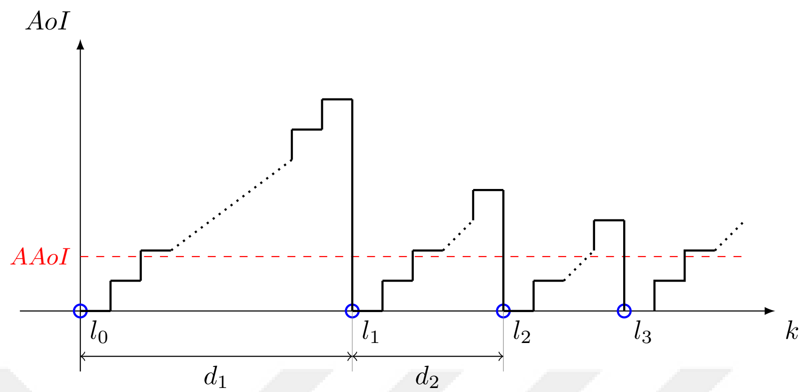

Figure 1.3: Change of age of information with time (circles shows the time of status updates).

Age of Information (AoI) is a commonly used metric for status update systems and it shows the freshness of the status update [22–25]. It is a powerful metric that includes the packet formation delay and transmission time. It is used to model the effects of all system delays in a simplistic and effective manner which makes it a popular status update metric. In this thesis, we focus on only one of the sensor nodes in Fig. 1.1 which is trying to send status updates as frequently as possible to a gateway. Definitely, a status update requires sensing, processing, and communication. We define the Age of Information (AoI) as the age of the most recent status update at the gateway for that particular sensor node. We envision a time-slotted system with slot length T. Since T would typically be much longer compared to the transmission and network delay, we will assume throughout the thesis that the data packets carrying status update messages are delivered immediately upon the sensing event. In Figure 1.3, the AoI function is illustrated as a function of the integer index k that keeps track of the number of slots in an example scenario. In this figure, l_i represents the status update instances, and d_i represents the time interval between status updates. Every time a status update decision is made, the AoI is brought down to zero since the update would fresh at the gateway at that instant. However, the AoI increases uniformly until the next status update.

Average Age of Information (AAoI) is the time average of the AoI function, which can be written as

$$AAoI = \lim_{K\to\infty} \frac{\sum_{k=0}^{K} AoI(k)}{K}.$$

The AAoI metric is indicative of the average status update rate. Battery depletion is an undesired phenomenon in sensor nodes as it not only affects the area covered by the node but also diminishes the performance of the WSN due to shortened network lifetime. A system that aims to minimize the AAoI also needs to prevent battery depletions as an empty battery will prevent the node from sending status updates when it is supposed to and the AoI would consequently rise to extreme levels. Therefore, the objective of minimizing AAoI also helps reducing the number of battery depletions that may occur until a time horizon. In this thesis, we use the AAoI metric for optimization purposes due to its effectiveness and simplicity.

In this thesis, we propose a solution to the battery management problem of an EH-WSN in a status update system. For this purpose, we have modeled the energy management process of a single sensor node using an MDP model. The packet transmission event is assumed to be immediate, and it includes sensing, processing and communication phases. This assumption is sensible as the node will sense and process data only when it decides to transmit, and also the order of the waiting time between two transmissions is much higher than the duration of the transmission event. As the sensing depends on the transmission decision, the system does not use a data buffer. We assume a single-hop WSN, i.e. the node does not carry transit traffic. We also omit the effect of energy leakage caused by the battery imperfections.

In our system model, the node wakes up with a predetermined system period denoted by T which is the slot length [1]. If the node decides to make a transmission, it senses the environment, processes the data, transmits the packet and goes back to sleep. Otherwise, the node goes back to sleep immediately without sensing. In the former case, the transmission process is assumed to be immediate. This assumption is valid as packet transmissions are not very frequent, i.e., T is much larger than the transmission times. For the sake of simplicity, we assume

[1] In the numerical examples of the thesis, we fix the slot length T to 10 minutes

that the communication channel is error free and each data packet is transmitted only once with a predetermined energy level appropriately set for the deployment area. The main goal of the thesis is to find an optimal policy for transmission decision (when to transmit or not) at each slot so as to minimize the performance metric AAoI.

In this thesis, sensor nodes that use particularly solar light as their harvesting source are studied. However, we believe that this research is extendible to different harvesting sources as well. We use a varying size Discrete Time MC (DTMC) to model the harvesting process and construct the model based on the time of day (ToD) information with a data-driven approach. In this manner, we aim to reflect the effect of daily variations of solar irradiance data on the transmission decisions of the wireless node. Dynamic programming (DP) techniques are subsequently adopted to solve the system model with the objective of obtaining a transmission policy that minimizes the AAoI metric. In fact, DP has been the most popular approach for solving reasonably-sized MDPs [26]. There are many different approaches that have been developed for solving MDPs such as Reinforcement Learning (RL) with reduced computational complexity in order to tackle MDPs with far larger state spaces [27]. In this thesis, only the DP approach is used stemming from tractable state space of the underlying MDPs.

With this motivation, our main concern in this thesis is to model the energy management process of an EH-WSN as an MDP and solve for optimal transmission policies to minimize AAoI. To the best of our knowledge, this is the first study that uses MDPs for minimizing AAoI in a solar energy harvesting sensor node scenario with a data-driven harvesting model. We show that daily variations of solar data can be used to learn smart policies for a wireless node to minimize AAoI in the long run. Besides, the effect of increasing the number of DTMC model of the harvesting process on the AAoI results is presented. Lastly, we compare the performance of generated transmission policies with commonly used threshold-based transmission policies for benchmarking.

1.2 Literature Review

Energy management problem of energy harvesting wireless sensor nodes has been a topic of research interest in recent years. They are many works focused on how to properly use the battery of the sensor node to maximize the utilization of harvested energy. [28], [29] and [30] present surveys on the recent works conducted on this area. From a general perspective, these energy management algorithms may be categorized into two groups: algorithms which are trying to solve (i) offline problem and (ii) online problem. The offline problem assumes the availability of knowledge on energy arrivals prior to the system execution; then, the energy management task turns into a planning problem. [31] focuses on a system where energy and data packet sizes are known and the aim is to minimize the transmission completion time. The authors in [32] present optimal solutions for throughput maximization for the offline case with considering storage losses due to the battery imperfections. A group size based approach to adaptively manage the duty cycle of sensor nodes is proposed to prolong network lifetime in [33].

In the online scenario, data and energy arrivals are not known a prior to the planning problem, but they arrive according to a statistical model. In [34], a threshold-based policy is proposed to maximize an average reward function in the long-term in a setting where the harvesting policy is modeled as a two-state Markov Chain. The study in [35] also considers the number of packets waiting in the queue as another dimension to the Markov Chain. Some prediction based energy managers [10] use predictions of the future amount of harvested energy on a finite time horizon for deciding on the energy consumptions of the node. There are also some studies which focus on optimizing the energy consumption of sensor node with the objective of energy-neutral operation with modeling the harvesting process as a Markov Decision Process [36, 37]. While the main focus in these energy-oriented schemes is to prolong the network lifetime by controlling node duty cycles, QoS requirements are neglected [38].

For the online energy management schemes, adaptive duty cycling is a commonly used strategy for optimizing the energy consumption of a sensor node according to the state of the energy and data sources, by making the node sleep or wake at proper times to use the energy of the node efficiently. The authors in [22] focus on age of information minimization in a status update scheme with finite and infinite battery cases and they offer a threshold based approach for minimizing AoI; however, they prove the finite battery case asymptotically optimal as battery size goes to infinity. In [37], optimal energy management algorithms are proposed to maximize the data rate of the node in the long run for several energy storages and harvesting models. [14] proposes a mixed approach that assumes a periodic discrete model for harvested energy to decide on the duty cycles based on this model and adjusts the duty cycles online with deviations of real harvested energy from the estimate.

There are also many approaches for modeling particularly the harvesting process. In [39], a day is divided into many time slots. It is assumed that the harvesting power and consumption in each slot are known and constant. A power consumption planning scheme is constructed in [40] with the assumption of a deterministic energy arrival process. [41] extends this approach with the objective of maximizing the amount of data transmitted until a specified finite time horizon. In [42], node battery is represented with a Multi-Regime Markov Fluid Queue, and the Continuous Time Markov Chain (CTMC) is used to model the harvesting process. The authors in [14] model the harvesting and consumption as two independent bounded random processes. [43] assumes a two-state (active and passive) continuous-time Markovian model with independent exponential variables to represent the duration of stay in each state.

1.3 Thesis Outline

The thesis is organized as follows: In Chapter 2, we present a brief survey about Markov Decision Processes and various Dynamic Programming techniques for solving MDPs, namely Policy Iteration and Value Iteration. Modeling of the harvesting process and the energy management process are explained in Chapter

3 along with system parameters and theoretical background. In Chapter 4, numerical examples and simulation results are presented for the validation of the effectiveness of proposed approach. We explain the derivation of the harvesting model parameters using solar data. We also demonstrate the use of our model for obtaining solutions for some engineering problems. In Chapter 5, we conclude with final remarks and provide future research directions.

Chapter 2

Background Information

In this chapter, we provide information required to understand the system model and techniques to propose solutions for the designed system model. We use a discrete-time MDP to model the energy management problem of the EH-WSN and Dynamic Programming is used to derive optimal transmission algorithms for minimizing AAoI. Therefore, in Section 2.1, we explain the theoretical background of Markov Decision Processes. In Section 2.2, we provide information about Dynamic Programming, a conventional approach that is used for solving MDPs. Theoretical information presented in this chapter and their notations are gathered from [27] and [44].

2.1 Markov Decision Processes

Markov Decision Processes can be defined as sequential decision-making problems, in which the agent has to select an action in each visited state. MDPs are systems that are run based on Markov Chains, in which the system changes its *state* randomly at each time instance. State transition probabilities of the system are based on the current state only. The path that the system followed until the current state does not affect the probability of transition to a next state. This

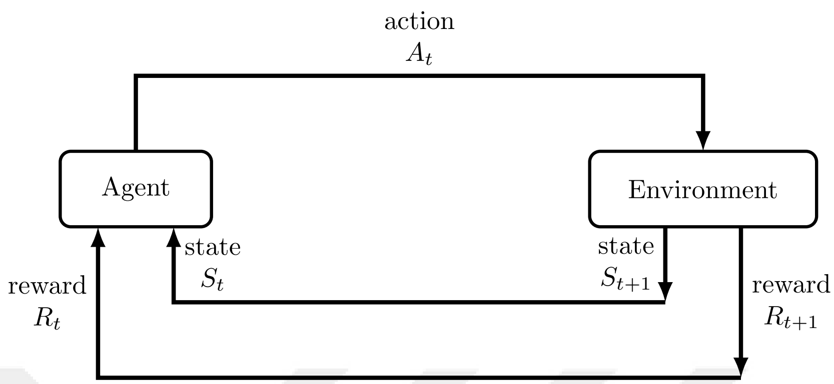

Figure 2.1: Agent-environment interaction in MDP.

characteristic of Markov Chains are defined as Markov property [45]. In an MDP, the system has to make decisions and choose an *action* from a set of possible actions in each state. Actions to be taken in each state are specified by a *Policy*. The system gets a *reward* in response to passing from one state to another by taking an action. Performance of policies are evaluated based on a performance metric, which is usually a function of rewards collected until a finite or an infinite time horizon [46]. The main objective of MDP is to find an optimal policy for the predefined set of states, actions, rewards and transition probabilities to maximize the performance metric [44]. It must be taken into consideration that the chosen actions not only affect the immediate reward, but also it will have an effect on the future set of states that will be transitioned, and so the future rewards that will be collected. Therefore, the policy must plan the trade-off between immediate rewards and future rewards, and sometimes it must choose the action with a lesser immediate reward for being able to collect higher rewards in subsequent states.

The main philosophy behind Markov Decision Processes is that there is a learner or *agent* that interacts with its environment and learns from these interactions to achieve a goal [27]. This interaction occurs in a sequence of discrete time steps, $t = 0, 1, 2, 3, \cdots$. At each time-step t, the agent observes the *state* of the environment S_t, and it selects an action A_t based on the state of the environment. Consequently, it receives a reward R_{t+1} in the next time step and transits to a state S_{t+1}. In a similar manner, it continues to take actions, getting rewards and changing states, eventually forming a trajectory like:

$S_0, A_0, R_1, S_1, A_1, R_2, S_2, A_2, R_3, ...$

In a finite MDP, number of states, actions and rewards are finite. In such cases, for each pair of S_t and A_t, discrete probability distribution of S_t and R_t in the next state are specified. These distributions constitute the state transition probabilities, which define the environment dynamics completely:

$$p(s', r|s, a) = \Pr\{S_t = s', R_t = r|S_{t-1} = s, A_{t-1} = a\} \tag{2.1}$$

The learning task may be split into independent learning sequences, called as *episodes*. In such cases, the agent starts an episode at an arbitrary state in the state space and continues to travel until it reaches to the goal or *terminal state*. Subsequently, it starts another episode on a state picked from a set of starting states. This kind of tasks is called as *episodic tasks*. In some cases, the interaction between the agent and the environment cannot be divided into separate episodes, rather it goes on continually. Such tasks are called *continuing tasks*.

Formulation of a performance metric for a policy is easier for episodic tasks as the learning sequence will include a finite number of steps. However, for continuing tasks, the sequence of obtained rewards through learning process may have infinite members. In such cases, there are two different metrics that have been used mostly: *average reward* and *discounted reward*. According to the discounted reward approach, the aim of the agent is to maximize the sum of discounted rewards it will receive over the future steps. In other words, the agent chooses A_t in order to maximize the expected discounted reward. The expected total discounted reward the agent will get if it follows policy π starting from state s is:

$$J_\pi(s) = \lim_{k\to\infty} \mathbb{E}\left[\sum_{t=0}^{k} \gamma^{t-1} R(S_t, \pi(S_t), S_{t+1}) \middle| S_0 = s\right] \tag{2.2}$$

where γ denotes a parameter called *discount rate* with $0 \leq \gamma \leq 1$. The discount rate is the parameter that evaluates the current value of a future reward. In other words, the value of a reward would worth γ^{k-1} times its immediate value if it would be received k time steps in the future. If γ is chosen as 0, the agent only considers immediate rewards it will receive when it is deciding on actions. As γ gets larger values closer to 1, the agent becomes more far-sighted considering the effect of future rewards more and more when it is picking an action. Another approach that is used as a performance metric of a policy is the average reward method. The average reward agent will receive if it follows policy π starting from state s is:

$$\rho_\pi(s) = \lim_{k\to\infty} \frac{\mathbb{E}\left[\sum_{t=0}^{k} R(S_t, \pi(S_t), S_{t+1}) \middle| S_0 = s\right]}{k} \tag{2.3}$$

where k is the number of time-steps. In the average reward approach, the agent aims at maximizing the average reward that will be obtained until an infinite horizon.

Value functions are defined to denote how good it is for the agent to be in a state, or to take an action in a state. This value is a representation of the future rewards that can be received, and it is calculated with the assumption of following a specific policy. *State-value function* ($v_\pi(s)$) of a state s when policy π is followed is defined as:

$$v_\pi(s) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s\right], \quad \forall s \in \mathcal{S} \tag{2.4}$$

Note that this value is calculated using the discounted reward metric. For the average reward case, the state-value function becomes:

$$v_\pi(s) = \frac{\mathbb{E}_\pi\left[\sum_{k=0}^{\infty} R_{t+k+1} \middle| S_t = s\right]}{k}, \quad \forall s \in \mathcal{S} \tag{2.5}$$

State value is a representation of the worth of being in a state. Another type of value functions is *action-value function* which shows the value of taking an action in a specific state. It can be described as the value of following policy π after starting from state s and taking action a and can be formulated for discounted reward as follows:

$$q_\pi(s,a) = \mathbb{E}_\pi\left[\sum_{k=0}^{\infty} \gamma^k R_{t+k+1} \middle| S_t = s, A_t = a\right], \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s) \tag{2.6}$$

For the average reward case, the action-value expression can be written as:

$$q_\pi(s,a) = \frac{\mathbb{E}_\pi\left[\sum_{k=0}^{\infty} R_{t+k+1} \middle| S_t = s, A_t = a\right]}{k}, \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s) \tag{2.7}$$

Then, solving an MDP basically boils down to the problem of finding the policy that will maximize the value functions for all states, or for all state-action pairs. That is:

$$v_*(s) = \max_\pi v_\pi(s), \quad \forall s \in \mathcal{S} \tag{2.8}$$

$$q_*(s,a) = \max_\pi q_\pi(s,a), \quad \forall s \in \mathcal{S}, \forall a \in \mathcal{A}(s) \tag{2.9}$$

where v_* and q_* show the *optimal state-value* and *optimal action-value* functions.

Ultimately, an MDP is a model of a system in which an agent travels between states based on the actions it takes and transition probabilities between the current and future states. In consequence of its actions and traveled states, it also collects rewards through its trajectory. Solving an MDP basically means to find the optimal policy that will maximize the performance utilization of the system using a metric calculated with a function of rewards. In the next section, we present Dynamic Programming, a method to reach the optimal policy for a predefined system.

2.2 Dynamic Programming

Dynamic Programming (DP) is a set of algorithms that are to find optimal policies for finite MDPs. There are different techniques for solving MDPs, and DP can be considered as the core understanding behind all of these methods. In DP, a perfect model of the system, i.e. complete definitions of states, actions, rewards, and transition policies, is required. Having the complete system model, the main aim of DP is to derive optimal policies by finding optimal value functions using the Bellman optimality equations:

$$\begin{aligned} v_*(s) &= \max_a \mathbb{E}\left[R_{t+1} + \gamma v_*(S_{t+1}) \middle| S_t = s\right] \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \gamma v_*(s')\big] \end{aligned} \tag{2.10}$$

$$\begin{aligned} q_*(s,a) &= \max_a \mathbb{E}\left[R_{t+1} + \gamma q_*(S_{t+1},a') \middle| S_t = s, A_t = a\right] \\ &= \max_a \sum_{s',r} p(s',r|s,a)\big[r + \max_{a'} \gamma q_*(s',a')\big] \end{aligned} \tag{2.11}$$

for all $s \in \mathcal{S}$ and $a \in \mathcal{A}(s)$.

2.2.1 Policy Evaluation

Policy Evaluation is the task of calculating the value function v_π for a policy π. This is an iterative process based on the Bellman equation and each set of value function approximations are calculated based on the previous one, by using the following update rule:

$$\begin{aligned} v_{k+1}(s) &= \mathbb{E}_\pi\left[R_{t+1} + \gamma v_k(S_{t+1}) \middle| S_t = s\right] \\ &= \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\left[r + \gamma v_k(s')\right], \quad \forall s \in \mathcal{S} \end{aligned} \tag{2.12}$$

for the discounted reward scheme, where v_k shows the kth set of approximations. For average reward, the update rule becomes:

$$v_{k+1}(s) = \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a) \frac{r + k v_k(s')}{k+1}, \quad \forall s \in \mathcal{S} \tag{2.13}$$

Algorithm 1 Policy Evaluation

Input: π
Output: $v_\pi(s)$

```
Parameter: Small threshold value of ϵ > 0 for checking convergence
Initialize v(s) arbitrarily for all s ∈ S
loop
    Δ ← 0
    k ← 0
    for all s in S do
        v_old ← v(s)
        v(s) ← Σ_a π(a|s) Σ_{s',r} p(s',r|s,a)[r + kv(s')]/(k + 1)
        Δ ← max(Δ, |v_old − v(s)|)
        k ← k + 1
    end for
    if Δ < ϵ then
        break
    end if
end loop
```

The sequences in eqs. (2.12) and (2.13) start with arbitrary value assignments and converge to v_π as $k \to \infty$. Value of each state is updated using the values of its next states. This procedure is called the *iterative policy evaluation* and an example implementation of this algorithm is given Algorithm 1 [27] for average reward. In the discounted reward case, only line with number 8 of Algorithm 1 changes as:

$$v(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma v(s')\big]$$

2.2.2 Policy Improvement

Once we evaluate a policy by finding the value functions, the next step is to improve the policy using those value functions. The policy update is performed by acting greedily based on the state or action values:

$$\pi'(s) = \operatorname*{argmax}_{a \in \mathcal{A}} q_\pi(s,a) \tag{2.14}$$

The approach here is to make the policy choose whichever action provides the highest expected reward in the future. Updating the policy, the state-values are improved:

$$q_\pi(s,\pi'(s)) = \max_{a \in \mathcal{A}} q_\pi(s,a) \geq q_\pi(s,\pi(s)) = v_\pi(s) \tag{2.15}$$

If the state-values are not improved, that is, the new policy is only as good as the previous policy, then,

$$v_{\pi'}(s) = v_\pi(s) = \max_{a \in \mathcal{A}} q_\pi(s,a), \quad \forall s \in \mathcal{S} \tag{2.16}$$

which shows that the Bellman optimality equation is satisfied. Therefore,

$$v_\pi(s) = v_*(s), \quad \forall s \in \mathcal{S} \tag{2.17}$$

and v_π is an optimal policy.

2.2.3 Policy Iteration

Policy Iteration is the process of iteratively repeating policy evaluation and policy improvement steps until the optimal policy is obtained. For a finite MDP, this process must converge to an optimal policy. A pseudo-code of the resulting algorithm is given in Algorithm 2 for the average reward scheme:

Algorithm 2 Policy Iteration

1: Parameter: Small threshold value of $\epsilon > 0$ for checking convergence
2: Initialize $v(s)$ and $\pi(s) \in \mathcal{A}(s)$ arbitrarily for all $s \in \mathcal{S}$
▷ POLICY EVALUATION
3: **loop**
4: $\quad \Delta \leftarrow 0$
5: $\quad k \leftarrow 0$
6: $\quad$ **for all** s in $\mathcal{S}$ **do**
7: $\qquad v_{old} \leftarrow v(s)$
8: $\qquad v(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)[r + kv(s')]/(k+1)$
9: $\qquad \Delta \leftarrow \max(\Delta, |v_{old} - v(s)|)$
10: $\qquad k \leftarrow k+1$
11: $\quad$ **end for**
12: $\quad$ **if** $\Delta < \epsilon$ **then**
13: $\qquad$ **break**
14: $\quad$ **end if**
15: **end loop**
▷ POLICY IMPROVEMENT
16: $policy - converged \leftarrow true$
17: **for all** $s \in \mathcal{S}$ **do**
18: $\quad old - action \leftarrow \pi(s)$
19: $\quad \pi(s) \leftarrow \text{argmax}_a \sum_{s',r} p(s',r|s,a)[r + kv(s')]/(k+1)$
20: $\quad$ **if** $old - action \neq \pi(s)$ **then**
21: $\qquad policy - stable \leftarrow false$
22: $\quad$ **end if**
23: **end for**
24: **if** $policy - stable \neq true$ **then**
25: $\quad$ Go to Step 3
26: **end if**
27: **return** $\pi(s), \forall s \in \mathcal{S}$

In the discounted reward case, Algorithm 2 is updated as:

$$v(s) \leftarrow \sum_a \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma v(s')\big]$$

at line number 8, and

$$\pi(s) \leftarrow \underset{a}{\text{argmax}} \sum_{s',r} p(s',r|s,a)[r + \gamma v'(s)]$$

at line number 19.

2.2.4 Value Iteration

Value iteration can be considered as a technique that combines the policy evaluation and policy improvement phases into a single step to reduce computational complexity. Policy evaluation is done for finding the value functions for a specific policy, and policy improvement aims to update policy so as the policy chooses actions greedily based on the value functions. At each step of value iteration, value functions are updated considering the best action that maximizes state value using the update formula in Equation 2.18 for the discounted reward scheme:

$$\begin{aligned} v_{k+1}(s) &= \max_a \mathbb{E}\left[R_{t+1} + \gamma v_k(S_{t+1}) \middle| S_t = s, A_t = a)\right] \\ &= \max_a \sum_{s',r} p(s', r|s, a)\left[r + \gamma v_k(s)\right] \end{aligned} \tag{2.18}$$

For the average reward case, the update formula becomes:

$$v_{k+1}(s) = \max_a \sum_{s',r} p(s', r|s, a) \frac{r + k v_k(s')}{k+1} \tag{2.19}$$

Algorithm 3 Value Iteration

```
Parameter: Small threshold value of ϵ > 0 for checking convergence
Initialize v(s) arbitrarily for all s ∈ S
loop
    Δ ← 0
    k ← 0
    for all s in S do
        v_old ← v(s)
        v(s) ← max_a Σ_{s',r} p(s',r|s,a)[r + kv(s')]/(k + 1)
        Δ ← max(Δ, |v_old − v(s)|)
        k ← k + 1
    end for
    if Δ < ϵ then
        break
    end if
end loop
π(s) ← argmax_a Σ_{s',r} p(s',r|s,a)[r + kv(s')]/(k + 1)
return π(s), ∀s ∈ S
```

Using a similar approach with policy iteration, one can prove that v_k converges to v_* in an infinite number of steps. However, in practice, value iteration is stopped when the change of value function in a single step is below a threshold value. A pseudo-code of an application in this form is shown in Algorithm 3. For the discounted reward scheme, Algorithm 3 only differs in lines with numbers 8 and 16 as:

$$v(s) \leftarrow \sum_{a} \pi(a|s) \sum_{s',r} p(s',r|s,a)\big[r + \gamma v(s')\big]$$

and

$$\pi(s) \leftarrow \underset{a}{\mathrm{argmax}} \sum_{s',r} p(s',r|s,a)[r + \gamma v'(s)]$$

In this chapter, we have presented the basic concepts of MDPs, and DP techniques to solve MDPs. In our work, policy iteration algorithm is used with average reward scheme and continuing tasks.

Chapter 3

Energy Management in EH-WSNs with MDP

In this chapter, we describe our system model developed for an EH-WSN using MDPs. In Section 3.1, we demonstrate the DTMC representation of the harvesting process which is one of the components of the WSN model. In Section 3.2, we construct the overall MDP model of the sensor node.

3.1 Solar Energy Harvester Model with Discrete Time Markov Chain

In this study, we focus on wireless nodes that harvest solar energy. An explicit characteristic of solar energy source, i.e., daylight, is that it exhibits fluctuations at different time scales. The statistical behavior of these fluctuations typically vary from one time period to another, i.e., monthly or seasonal variations. It can also exhibit random variations within a day due to clouding. In this research, we aim to reflect the effects of daily variations of daylight on the harvester model. We leave the incorporation of fluctuations at longer time scales for future research.

In the literature, there are many studies that use DTMCs to model the energy harvester [47–52]. In our work, varying sized DTMCs (with size H) are used for comparative assessment. In the proposed model, states of the DTMC represent the time of day, and in each state, we have a different distribution of the amount of harvested energy. Figure 3.1 shows the general form of the model of harvesting process.

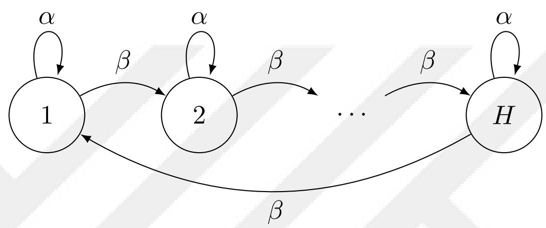

Figure 3.1: DTMC model of the harvesting process.

We fix the slot length T to 10 minutes in our example. A single day is then partitioned into 144 slots. For the DTMC model of size H, each state of the DTMC represents a certain collection of consecutive slots with overall $(\frac{144}{H})$ slots per collection. For ensuring divisibility, we use values of 1, 2, 4, 6, 8 and 24 for the size parameter H. For example, when $H = 2$, the state of the MC will represent whether there is daylight or not. For the general H case, the transition probabilities α and β can be written as:

$$\beta = \frac{H}{144}, \quad \alpha = 1 - \beta = \frac{144 - H}{144}. \tag{3.1}$$

In each state h, the energy harvesting distribution $p_h(k)$ is defined as the probability of harvesting k units of energy throughout a single slot. [1] These probabilities are extracted from the solar radiance data of 20 years. For this purpose, we need a mapping between time-slots indexed by $\tau \in \{1, 2, \ldots, 144\}$ and the states $h \in \{1, 2, \ldots, H\}$. Here, $\tau = 1$ corresponds to the time slot 00:00 - 00:10, $\tau = 2$ corresponds to the time slot 00:10 - 00:20, and so on, and finally $\tau = 144$ corresponds to the time slot 23:50 - 00:00. Figure 3.2 illustrates an example case of this mapping. In the figure, q_h denotes the index of time-slot that state h starts and $L = \frac{144}{H}$ shows the slot counts in each state. Values of q_h are

[1] In the numerical examples, the unit of energy is fixed to 1 mWh.

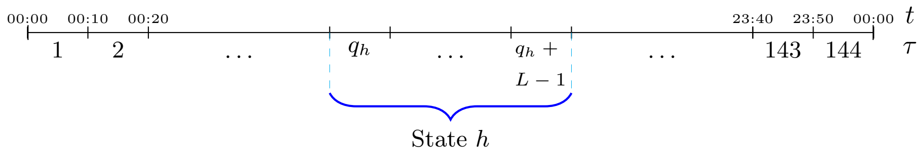

Figure 3.2: Mapping between time-slots and states.

determined based on the general pattern of the solar data for different locations. How these values are determined is numerically explained in the next chapter. Having done this mapping, the solar data is averaged for each time-slot(τ):

$$Z(\tau) = [\frac{\sum_{d=0}^{D-1} X_d(\tau)}{D}], \quad \forall \tau \in \{1, 2, ..., 144\} \tag{3.2}$$

where D is the total number of days in solar dataset, $X_d(\tau)$ is the solar radiance value in time-slot τ of day d, and $Z(\tau)$ is the average solar radiance value for each τ rounded to the multiples of 1 mWh. Harvesting distributions of all states are obtained from $Z(\tau)$ as following:

$$p_h(k) = \sum_{\tau \longleftrightarrow h} \frac{H}{144} \delta(k - Z(\tau)), \quad \forall h \in \{1, 2, ..., H\} \tag{3.3}$$

where δ is the Dirac delta function.

Let's consider the case with a 2 state DTMC. In this problem, a day is divided into two parts: hours that the node can harvest energy and hours that the node can harvest almost nothing. The two-state case basically describes the day and night modes for the harvester state. In both modes, the harvesting source is modeled with a stochastic process following a discrete random variable. We have produced the DTMC representation of the harvesting source for 1, 2, 4, 6, 8 and 24 cases. The process of obtaining the probability distributions in each case from the solar data will be explained in detail with numerical examples in Section 4.2.

3.2 Energy Management Problem with Markov Decision Process

We assume a battery with a finite capacity C. At each time step, the node harvests some energy based on the state of the harvesting process and stores the harvested energy in its battery. The node also decides to whether or not to make a transmission and based on this decision a predetermined amount of energy is spent. We assume that the sensing, processing, and transmission tasks are performed immediately. This assumption is reasonable as the duration of these tasks will be negligibly small compared to the time-step duration of the node, which is the duration of sleep time between transmission decisions. As we have defined in Chapter 1, AoI is defined as the elapsed time-steps the since last status update of the node and our QoS requirement is to minimize AAoI.

To model this system, we have built a three dimensional MDP composed of these parts: the discretized value of remaining energy in the battery in units of mWh, $\mathcal{B} = \{0, 1, ..., N_B\}$, the value of the age of information in terms of time-steps, $\mathcal{M} = \{0, 1, ..., N_M\}$, and the state of the harvester model, $\mathcal{H} = \{0, 1, ..., N_H\}$, where N_B, N_M, N_H represents the maximum values of respective states [53]. This set of variables forms our state space as $\mathcal{S} = \mathcal{B} \times \mathcal{M} \times \mathcal{H}$. The action space for each state can be written as $\mathcal{A} = \{0, 1\}$, where 1 and 0 represents whether or not to make a status update in the corresponding state. While the transition probability of the battery state dependents on the harvester state, state transition of age of information state is deterministic, depending only on the action taken by the agent in that state. Then the transition probability from state $(S_B, S_M, S_H) = (b, m, h)$ to state $(S_B, S_M, S_H) = (b', m', h')$ in the k^{th} time instance can be written as:

$$\begin{aligned}
&\Pr\Big\{\big(S_B(k+1), S_M(k+1), S_H(k+1)\big) = (b', m', h')\Big| \\
&\quad \big(S_B(k), S_M(k), S_H(k)\big) = (b, m, h)\Big\} \\
&\quad = P\Big(S_B(k+1) = b'\Big|\big(S_B(k), S_H(k)\big) = (b, h)\Big) \\
&\qquad P\big(S_M(k+1) = m'\big|S_M(k) = m\big)P\big(S_H(k+1) = h'\big|S_H(k) = h\big)
\end{aligned} \tag{3.4}$$

Transition probability from $S_M(k) = m$ to $S_M(k+1) = m'$ is dependent on the action value $A(k)$. When $A(k) = 1$, this relation can be written as:

$$P\big(S_M(k+1) = m' \big| S_M(k) = m\big) = \begin{cases} 1, & \text{if } m' = 0 \\ 0, & \text{otherwise} \end{cases} \tag{3.5}$$

When $A(k) = 0$ and $m < N_M$, transition probability for becomes:

$$P\big(S_M(k+1) = m' \big| S_M(k) = m\big) = \begin{cases} 1, & \text{if } m' = m+1 \\ 0, & \text{otherwise} \end{cases} \tag{3.6}$$

If $m = N_m$ when $A(k) = 0$, then $S_M(k+1) = S_M(k) = m$ with probability 1.

State of the harvester changes according to a Bernoulli distribution, and the duration of stay in each harvester state forms a geometric distribution. This transition probability can be shown as:

$$P\big(S_H(k+1) = h' \big| S_H(k) = h\big) = \begin{cases} \beta, & h' \neq h \\ \alpha, & h' = h \end{cases} \tag{3.7}$$

Transition probabilities of the battery level depend on the probability distribution of the harvester. This relation can be represented with the following equation:

$$P\Big(S_B(k+1) = b' \Big| \big(S_B(k), S_H(k)\big) = (b, h)\Big) = p_h(k), \quad \text{for } b' = b + k - \theta_k \tag{3.8}$$

$\forall k \in \{0, 1, ...\}$, where θ shows the amount of energy spent for the transmission of a status update and,

$$\theta_k = \begin{cases} \theta, & A(k) = 1 \\ 0, & \text{otherwise} \end{cases} \tag{3.9}$$

Rewards of the model are assigned for the objective of minimizing AAoI in the long run. For this purpose, the agent gets a negative reward $-m$, or a cost, when it resides in a state with $S_M(k) = m$. Reward assignments of each state can be described as $\mathcal{R} = -S_M$. This completes the definition of the MDP model with descriptions of state space, action space, transition probabilities and reward assignments.

Chapter 4

Numerical Results

In this chapter, we present the outputs of our work. Section 4.1 explains the parameters that we use in our model. In Section 4.2, details of the extracting DTMC model from solar data is presented. Lastly, in Section 4.3, we demonstrate the results that we obtained using the model depicted in the previous chapter.

4.1 Model Parameters

In this study, we use a 3.7V Lithium-Polymer rechargeable battery. Unless otherwise specified, we assume a capacity of 500 mWh for most of the test cases. But we also show the effects of changing battery capacity in the engineering problems section. We assume 5% leakage rate per month; however, this does not show any effects on the results. As the system time-step is 10 minutes, battery's self discharge amount would be around 6uWh in a time-step, assuming a constant rate discharge model. Our battery model is discrete with 1mWh resolution and decreasing this value would cause the number of system states increase drastically. Therefore, we prefer to omit the effect of leakage reasonably. There should be an upper bound on the value of age of information to limit the number of system states. We assume this upper bound as 50, since 500 minutes can be considered as a catastrophic value of AoI that the system will never reach.

Solar cell efficiency can be defined as the ratio of electrical energy generated by solar cell to the total solar energy that arrives to the cell. Solar cell technology has been developing rapidly over the recent years. However, solar harvesters still has a long way to go before reaching the theoretical limits of energy efficiency. Most solar cells on the market has an efficiency rate between 10 to 30%. In our work, we assume a mid-level solar harvester with 20% efficiency. For the cell size, we assume a 30 cm^2 photo-voltaic cell. We also consider various solar cell sizes in the engineering problems framework.

We assume that the packet transmission process occurs immediately without any retransmissions. The transmission event is considered as a bundle of successive events including data sensing, data processing, and transmission of the generated packet. These tasks will not be performed independently as the node will only sense data when it decides to transmit a status update. Therefore, we assume that the node spends energy only for transmission events. Considering Expected Retransmission Count (ETX) for different channel conditions and distance to the gateway, in our simulations, we assume several values of packet transmission energy varying from 1 mWh to 20 mWh per packet.

4.2 Building the Model of Solar Harvesting Process

To build the model of harvesting process and test the generated policies, we make use of solar data obtained from National Solar Radiation Database [54]. We used the solar radiance data of 20 years (1991-2010) with 1 hour resolution for two places. These locations are chosen so that the 24 hour daily cycles and seasonal effects can be more clearly observed. Therefore, we preferred locations in middle latitudes, as our model is more suitable for those areas. These locations are Atlanta Hartsfield-Jackson International Airport ($33°38'N$) and Seattle-Tacoma

International Airport ($47°27'N$) and they are mentioned as *Location 1* and *Location 2* in the remainder of the paper. You can see the hourly averages of solar radiance data for two locations in Figure 4.1. We can see that *Location 1* gets more solar radiation compared to *Location 2* due to the latitudinal effect as expected.

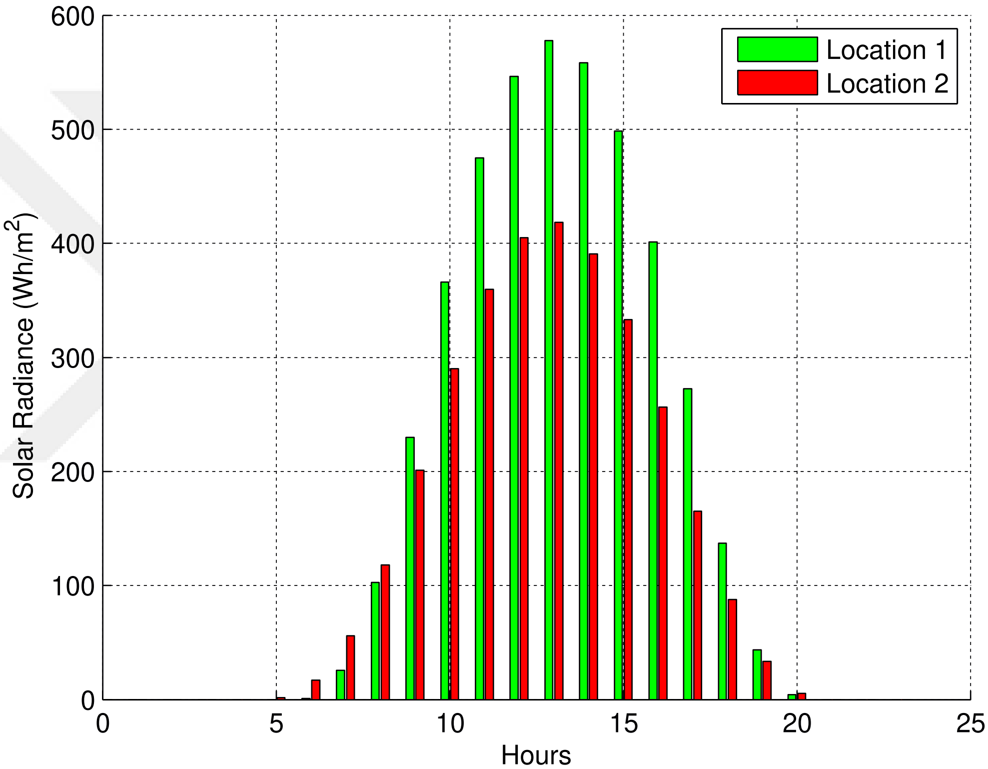

Figure 4.1: Hourly average of solar radiation in the used datasets.

To construct our Markov Chain model, we make use of the hourly average values presented in Figure 4.1. Using the hourly averages, we deduce the proper times to divide a day into the number of states of our Markov Chain model. For the case with a single state, the harvester model is only composed of a random variable stating the probabilities of amount of energy in terms of milliwatt-hours that can be harvested in a single time-step. Since the solar data is at 1 hour resolution, we assume that the value of solar radiation is constant through an hour. With this assumption, the histogram of the harvested energy in terms of mWhs is calculated as shown in Figure 4.2. Note that these probability distributions are what determines the state transition probabilities of the MDP.

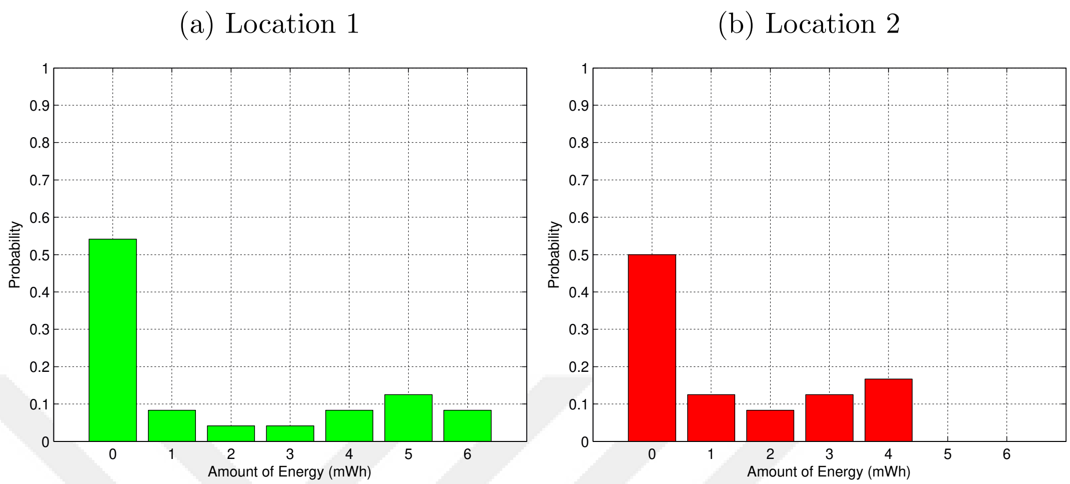

Figure 4.2: Distribution of harvested energy for $h = 1$.

To construct the 2-state DTMC model of the harvesting process, we first divide the day into two sets of hours using the graph in Figure 4.1. We split the day in such a way that the hours that the most solar energy arrives are grouped together. We ensure this by preserving the symmetry of the graph, i.e., the 12 hours that surrounds the most sunny hour is taken into a group, and the rest composes the other group. With this approach, the first set of hours is from 7am to 7pm and the rest makes the second group, creating a Markov Chain as in 3.1. The transition probabilities between the states of the Markov Chain is calculated according to the system time-step, 10 minutes. For this case, expected duration of each state is 72 time-steps. Therefore, states change with probability $\frac{1}{72}$ and stay otherwise. Then, for the 2-state case, calculated energy harvesting probability distributions are given for Day states of both locations in Figure 4.3. Amount of energy harvested in $Night$ states are 0 with hundred percent probability.

Following the same approach, energy harvesting probability distributions are obtained for 4, 6, 8 and 24 state cases. For instance, in the 4-state DTMC model, states are divided as: 10am-4pm, 4pm-10pm, 10pm-4am, and 4am-10am. Note that the division is done based on the same strategy to keep 1pm as the center hour of one of the states. For various values of DTMC state count same procedure is applied and energy harvesting probability distributions are obtained for each state. Using the constructed harvester models, optimal transmission policies are

(a) Location 1 (b) Location 2

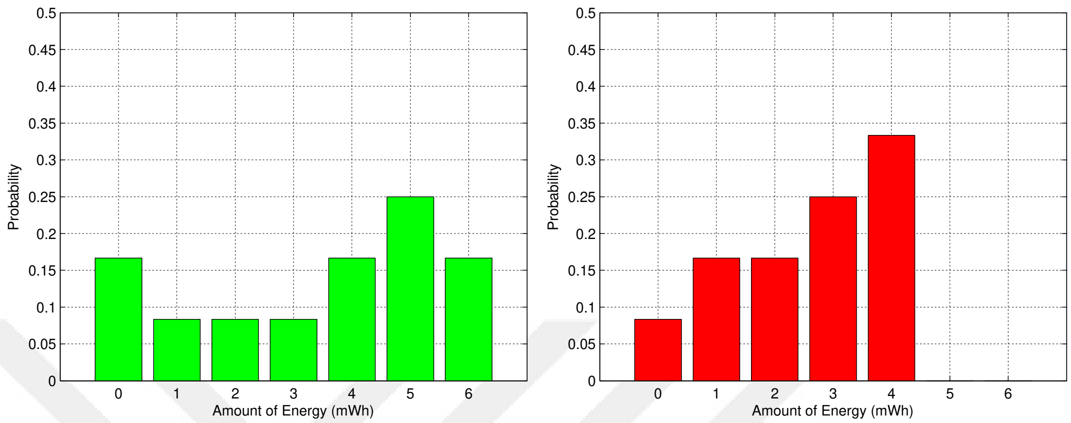

Figure 4.3: Probability distributions of *Day* states for both locations.

calculated for several system parameters. In the next section, we present the results obtained for different cases.

4.3 Model Verification

In this section, we demonstrate the performance results of the policies that are extracted from our system model.

4.3.1 Results for Different Harvesting Models

In the previous section, we described how we obtain different solar energy harvesting models by using DTMCs with different number of state counts. In this section, we present the effect of number of states of the harvester model to the average age of information values. Note that we fix battery capacity as $C = 500mWh$, solar panel area as $30cm^2$ for these tests. The results in Figures 4.4 to 4.8 demonstrates AAoI values obtained for cases for different packet transmission energies. These results can be interpreted as the performance obtained for different channel conditions. The case that the packet transmission is 20 mWh represents the performance of a sensor node in the harshest channel conditions.

(a) Location 1 (b) Location 2

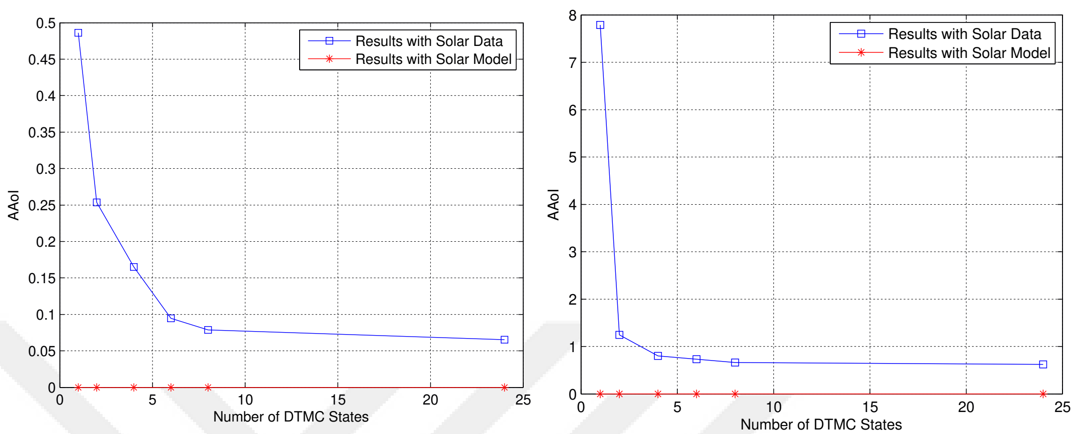

Figure 4.4: AAoI for packet transmission energy of 1 mWh.

(a) Location 1 (b) Location 2

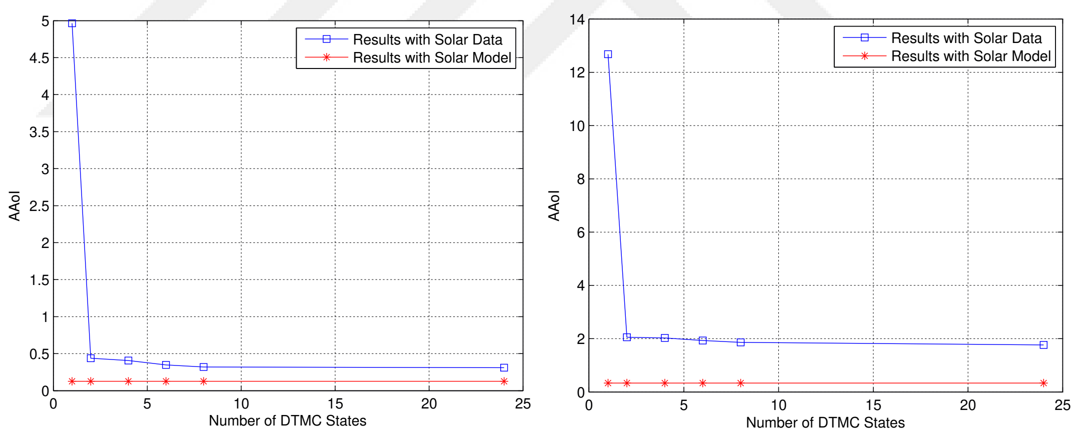

Figure 4.5: AAoI for packet transmission energy of 2 mWh.

There are two separate results shown in each figure. One of those is the performance of the policy when it is tested against the real solar data obtained from NSRDB. The other result is the performance of the policy when it is tested with the solar model that is used to derive the policy. As expected, the performance of the policy is optimized for the system model, and the AAoI when it is tested against the real data is higher. However, we can see that the performance difference between two results is decreased as we increase the number of DTMC states. This shows that the we get a more realistic model with the increasing number of harvester states in the model. The gap between the blue and red lines is quite normal as we omit some critical parameters in our approximations, e.g. seasonal and yearly variations of solar radiance. If we had a perfect model of the system, we could observe the convergence of these lines.

(a) Location 1 (b) Location 2

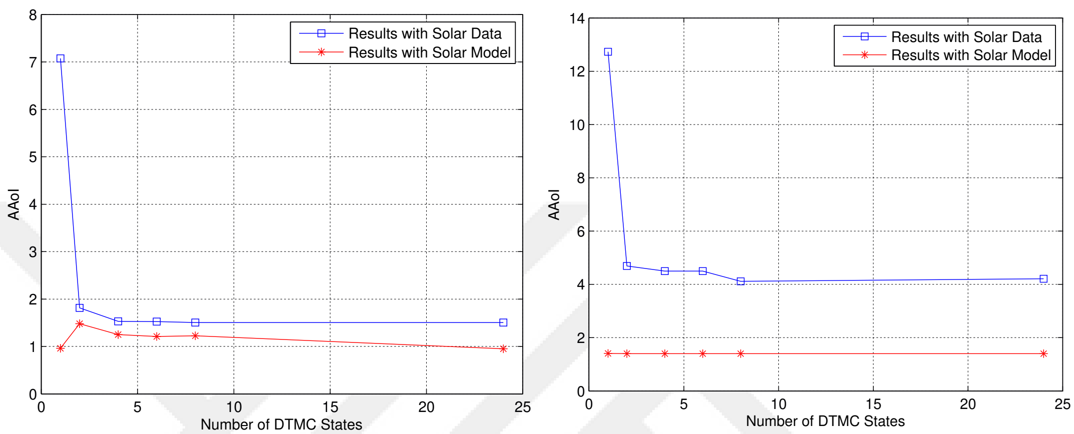

Figure 4.6: AAoI for packet transmission energy of 5 mWh.

(a) Location 1 (b) Location 2

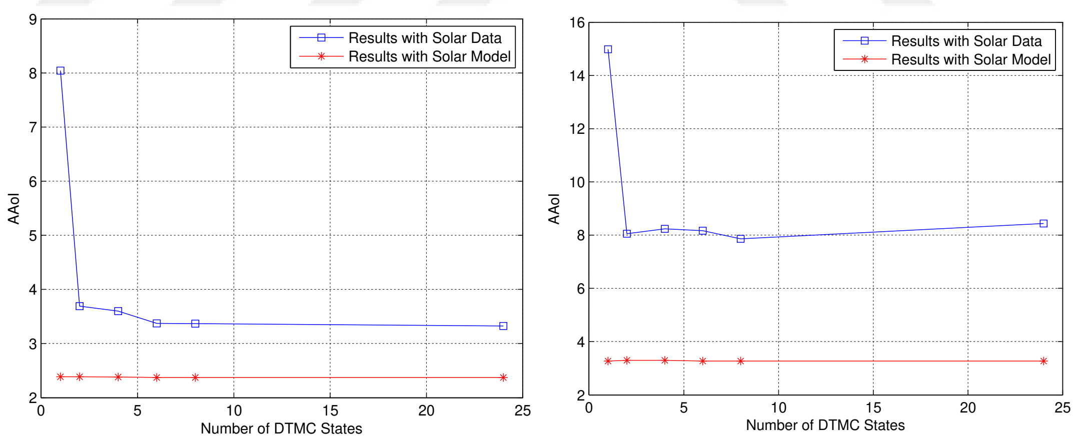

Figure 4.7: AAoI for packet transmission energy of 10 mWh.

Comparing the results for two locations, we can also observe the response of generated policies in environments with different insolation durations. Recall that the *Location 1* is a place getting more solar radiation compared to *Location 2*. Consequently, the sensor node in *Location 1* can be more robust against usage of higher transmission powers. Note that there are some cases in presented results in which we see a slight decrease in the performance of the policy despite of the increase in the number of DTMC states. Conceivably, these errors are caused by some simulation inaccuracies.

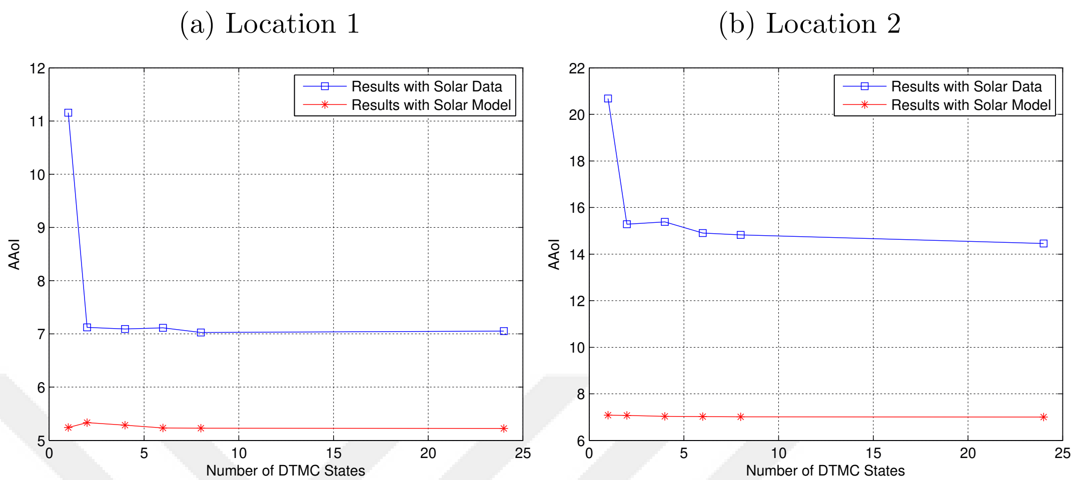

Figure 4.8: AAoI for packet transmission energy of 20 mWh.

From the results in Figures 4.4 to 4.8, we can also deduce that 2-state DTMC is providing a good performance, and increasing the state count beyond 2 doesn't affect the performance of the policy considerably. However, in several cases, there are still more than 10% difference between the AAoIs obtained with 2 and 4-state models. Therefore, we will continue with using the 4-state DTMC model for the rest of the test scenarios.

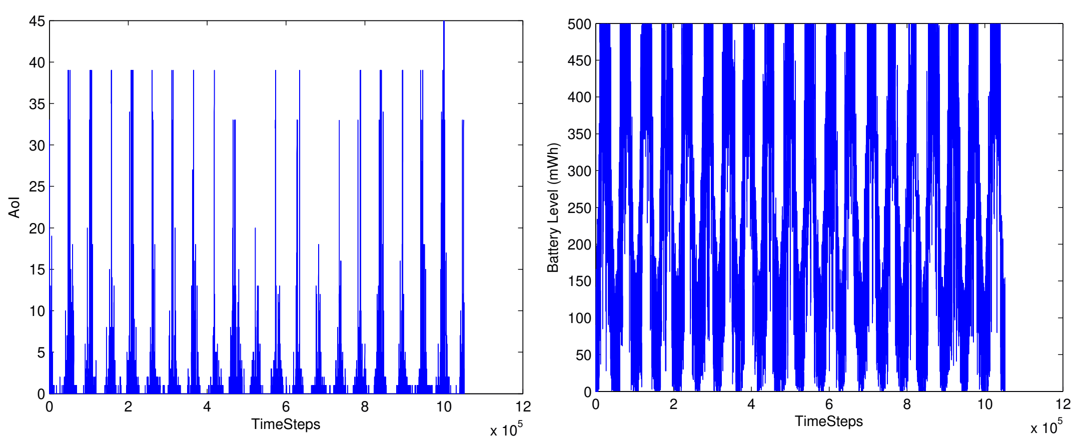

Figure 4.9: AoI and battery level variations through a test case with 500mWh battery capacity, 2mWh packet transmission energy, 30cm^2 solar panel area and 8-state harvesting model.

Figures 4.9 to 4.12 show the records of four example test cases. In figures, we can see the change of battery level and age of information through tests with solar dataset. From variations of the battery level, we can observe the seasonal

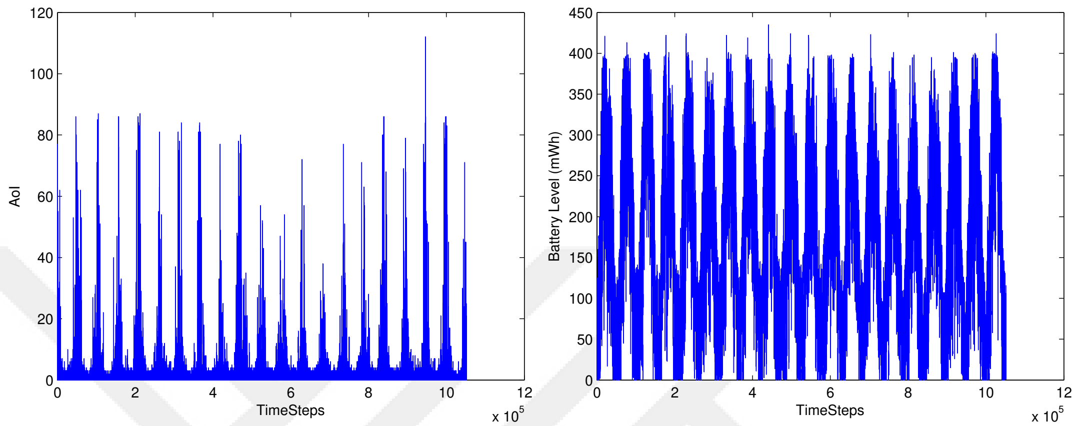

Figure 4.10: AoI and battery level variations through a test case with 500mWh battery capacity, 5mWh packet transmission energy, 20cm^2 solar panel area and 4-state harvesting model.

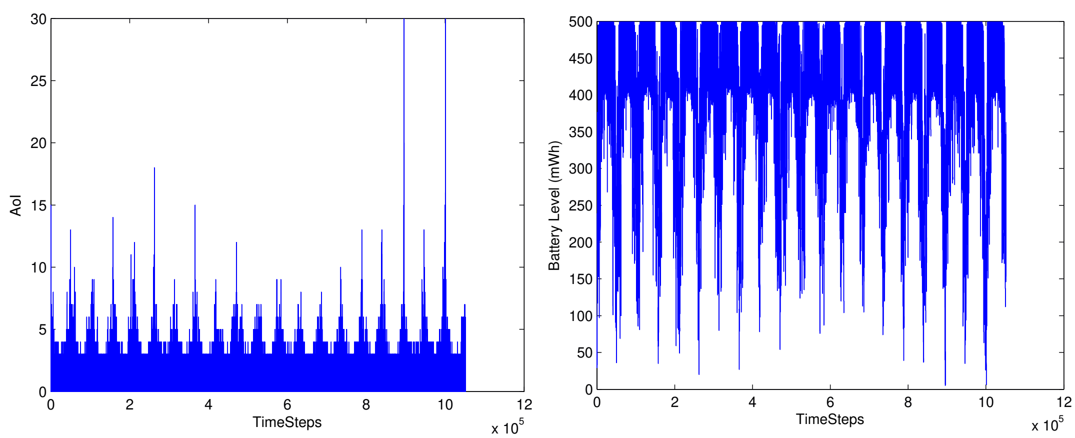

Figure 4.11: AoI and battery level variations through a test case with 500mWh battery capacity, 5mWh packet transmission energy, 30cm^2 solar panel area and 4-state harvesting model.

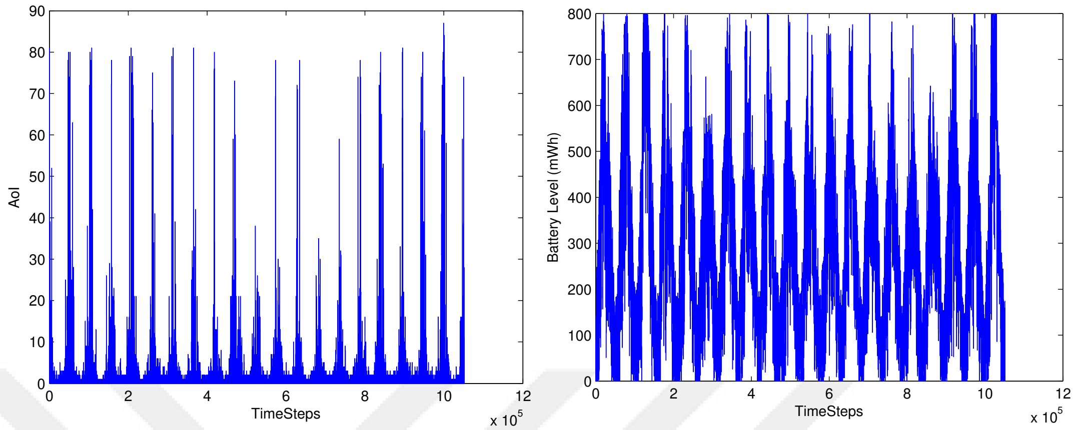

Figure 4.12: AoI and battery level variations through a test case with 800mWh battery capacity, 5mWh packet transmission energy, 30cm^2 solar panel area and 4-state harvesting model.

variations of the solar radiance. The node battery is almost full in summers, while it is very close to depletion in winter times. And the transmission policy fulfills its task by finding a balance in between. In one of the cases, the node battery depletes just a few times through a 20 years of solar data test, while the performance of other cases are relatively worse. In Figures 4.9 and 4.11, we can see that the age of information never hits to the catastrophic level, and it is mostly held below a threshold level. However, AoI value goes to the peak levels in the times of scarcity of solar radiance. The transmission policy allows the AoI to rise for some time, in order to prevent the battery depletion. By taking this precaution, the policy reduces the AAoI in the long term.

4.3.2 Results Against Benchmark Policies

In this section, performance of the generated transmission policies are compared with basic threshold-based policies. Threshold based policies are widely preferred for satisfying QoS requirements with a very simplistic transmission policy [22,34]. A threshold-based policy can be defined as a transmission algorithm in which the node decides to make transmission whenever a system variable is above or below a predefined threshold. First, we consider an AoI based transmission policy, which makes transmission whenever age of information rises above a threshold value.

We name this algorithm BP, as the first benchmark policy. Using our solar dataset, we test possible values of threshold value to find the ideal threshold that minimizes the AAoI in the long term, fixing the battery capacity and packet transmission energy. Figure 4.13 displays performance of BP for different values of battery capacity.

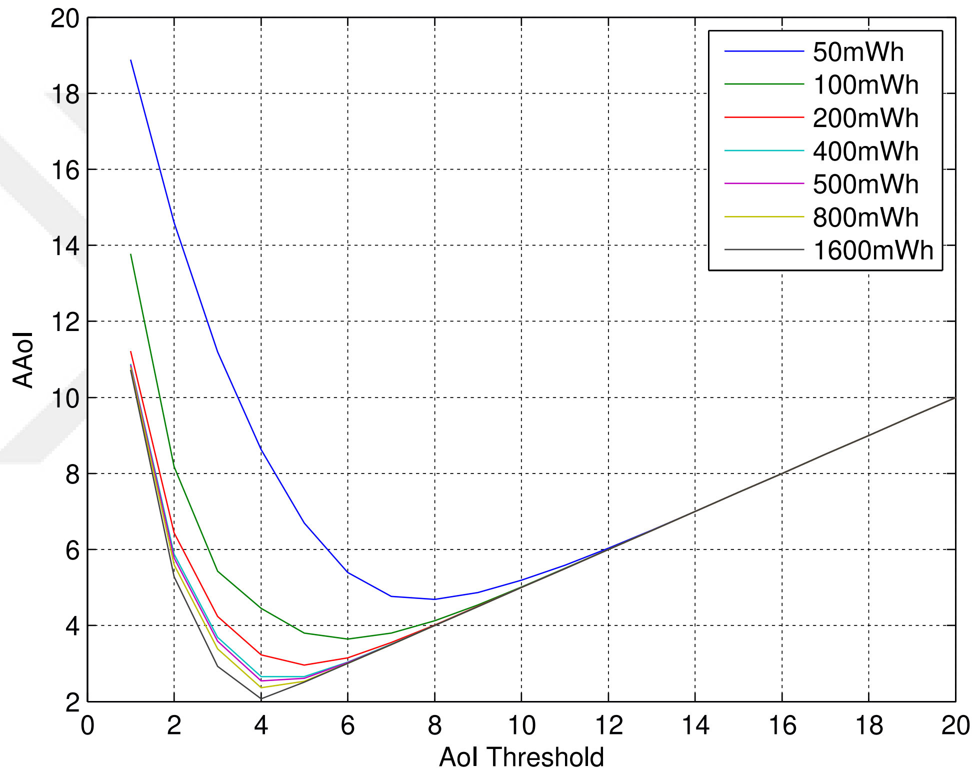

Figure 4.13: AoI-threshold based policy results for different values of battery capacity.

We define BP^* as the optimal threshold-based policy that uses the optimal threshold values for different values of battery capacity. In Figure 4.14, we demonstrate the comparison of mean AoI results of BP^* and the transmission policy obtained with our MDP model. It can be seen that our algorithm performs better than the threshold-based policies that uses the current AoI information.

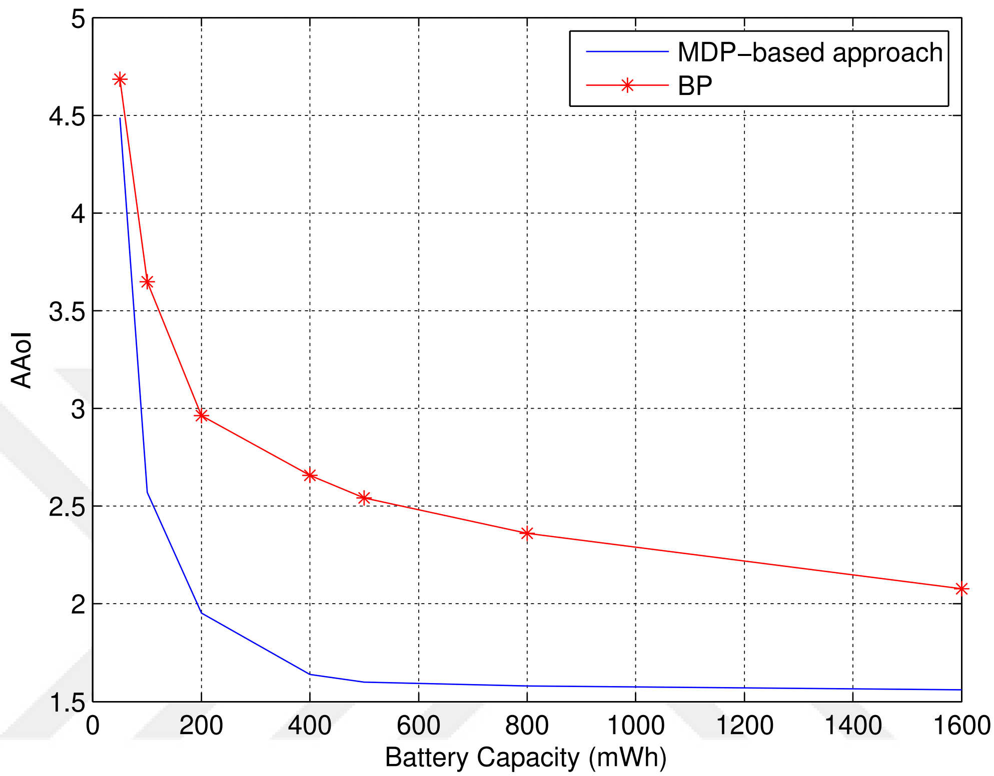

Figure 4.14: Comparison of MDP approach with a threshold-based policy.

4.3.3 Solutions for Some Engineering Problems

Up to this point, we have shown that our approach can be used to minimize the AAoI by finding an optimal policy for a given set of system parameters. In this part, we demonstrate that we can also use this technique to obtain solutions for some engineering problems while building a WSN system. We show that the ideal system parameters can be derived in the sense of a cost-optimal sensor node. EH-WSNs are sensor devices with a limited battery and energy harvester unit. These nodes are designed to be small, self-sufficient and cheap, as a large number of these nodes are used to build a WSN. Therefore, containing excessive amount of resources on these nodes is not a good practice for price competition of the SN with other products on the market. In cost-optimality perspective, a sensor node must include resources that is just adequate to fulfill its QoS requirements. Putting resources below that level will lead the node not to carry out its duties. For these reasons, finding the optimal values for the system parameters are vital for the usability of the sensor node.

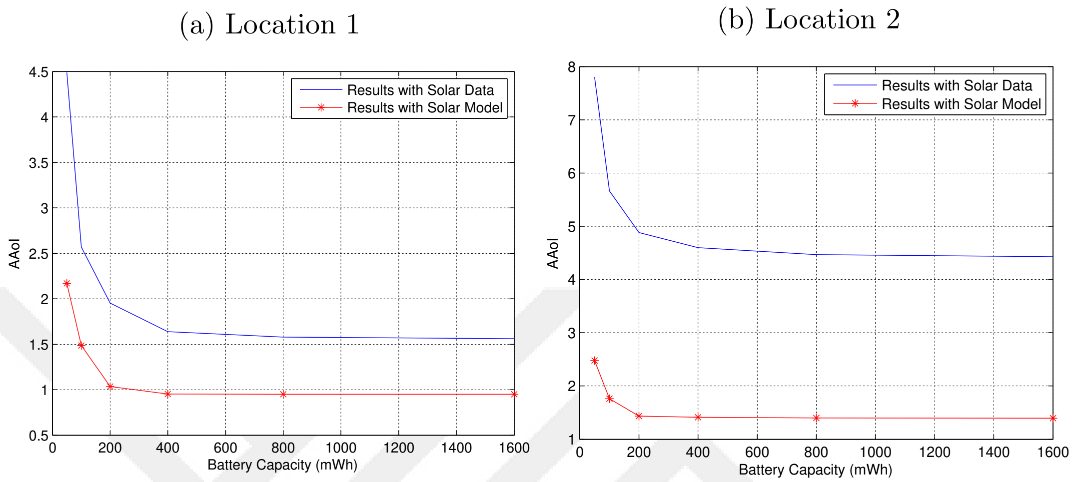

Figure 4.15: AAoI for different battery capacities.

In this section, we focus on two system parameters that can be optimized with our approach, battery capacity and solar panel size. These are the two main entries that determines the price of the sensor node, as they are usually the most expensive components of the node. To find the optimal values for these variables, we have found the AAoI results of the system with different values of these resources. In Figure 4.15, change of AAoI is demonstrated for various values of battery capacity, with packet transmission energy fixed at 5 mWh and solar panel size fixed at 30 cm^2. We can see that mean AoI decreases up to a value of B, but it doesn't change considerably beyond some level. This is expected as the battery can store energy only as much as the solar cell can harvest. After some point, solar panel size becomes the bottleneck for system performance, and increasing the battery capacity beyond that point will have no effect on the average AoI, as the excessive part of the battery will not be filled. Note that the optimum battery capacity is also dependent on the energy spent for transmission of a single packet. In Figure 4.16, we show the change of mean AoI with different values of battery capacity in terms of transmission energy of single data packet. For a fixed sized of solar panel, required battery capacity can be found for a requirement of average AoI level using the plots in Figure 4.16. For instance, a battery that can store energy for 50 packet transmissions will be enough to

(a) Location 1 (b) Location 2

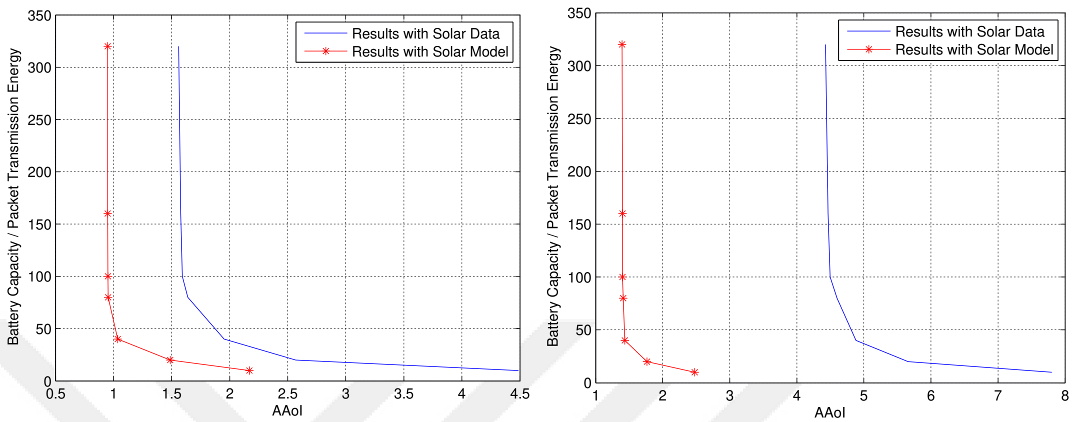

Figure 4.16: AAoI for different values of ratio of battery capacity to packet transmission energy.

obtain a system that will have 20 minutes average AoI, in *Location 1*. However, *Location 2* cannot reach to this mean AoI level, as the 30 cm^2 solar cell area is not sufficient in this location.

(a) Location 1 (b) Location 2

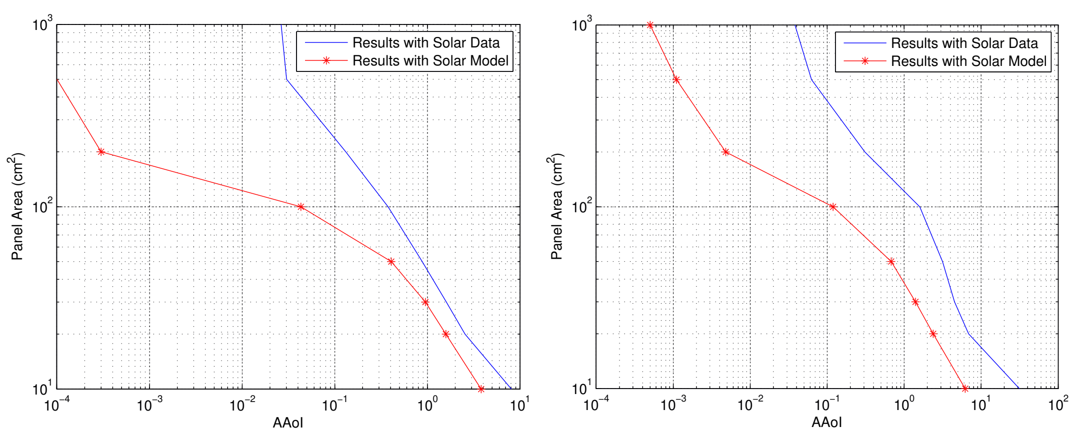

Figure 4.17: AAoI vs. solar panel area.

As the second step, we fix the battery capacity at 500 mWh and packet transmission energy at 5 mWh to show the effect of solar panel size on the AAoI. Figure 4.17 shows the change of mean AoI for several values of solar panel size from 10 cm^2 to 1000 cm^2. In this case, battery capacity becomes the bottleneck for the system performance while the area of solar cell is increased.

Above results have shown how battery capacity and solar panel size are interconnected. To obtain a better system performance, both must be enhanced. Therefore, their effect on QoS should be observed together. Combining the results above, Figure 4.18 shows the change of AAoI while changing the size of solar cell and battery capacity for *Location 1*.

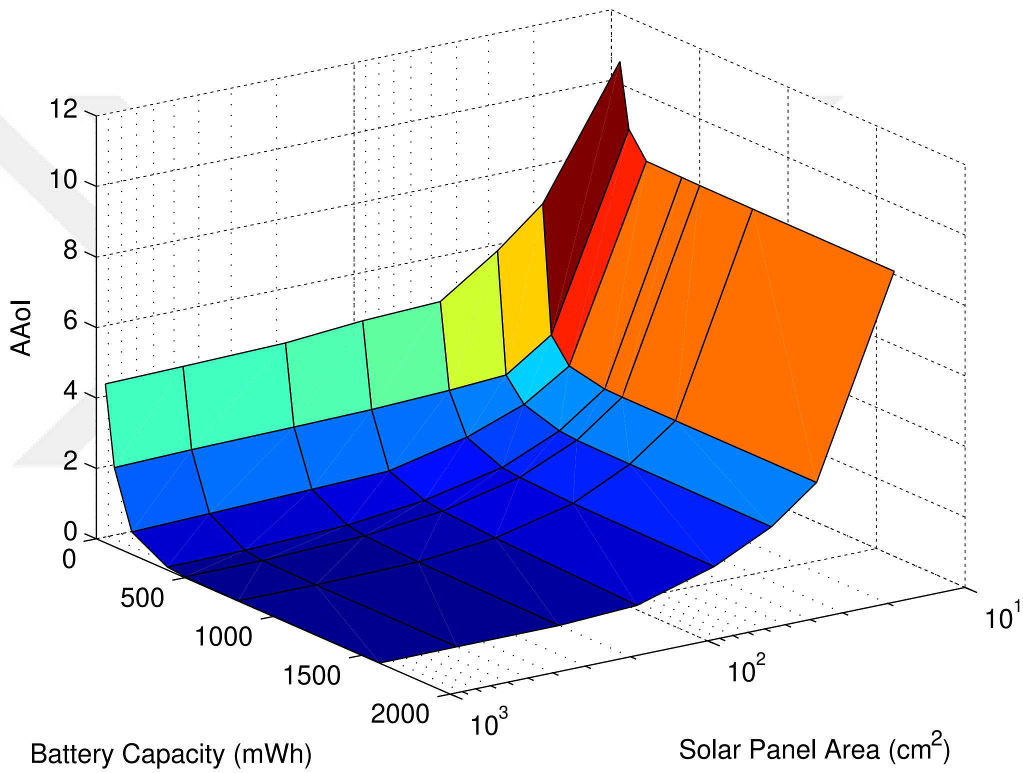

Figure 4.18: Change of AAoI with panel size and battery capacity.

Chapter 5

Conclusions

In this thesis, we propose an MDP model for the battery management problem of a solar energy harvesting wireless sensor node and we present optimal transmission policies with the objective of minimization of the AAoI metric. For this purpose, we model the daily variations in the solar energy harvesting process with a varying size DTMC and subsequently we show that increasing the number of states of the DTMC up to a certain value enhances the AAoI performance of the transmission policy. We use the average cost Policy Iteration technique of the Dynamic Programming framework to construct optimal policies. In this work, we focus on modeling solar energy harvesting. Moreover, only the hourly variations of the solar radiance data is reflected on the harvesting model without considering monthly changes or seasonal effects. Even though the main theme was built upon a solar energy harvesting device, it is quite possible to extend this work to other energy harvesting sources. We also show that the proposed approach outperforms a benchmark threshold-based policy in terms of AAoI. The MDP model can also be used to derive system parameters to build cost-optimal devices by finding the minimum amount of resources that can provide a desired level of QoS requirements. Use of the single average AoI metric for optimization has made it possible to obtain transmission policies that maximize the transmission rate of the sensor node while also preventing battery depletions to the extent possible.

There are a number of possible future research directions. Firstly, a more advanced system model can be built to obtain better average AoI. We have shown that there is a performance difference of our policies that we see by comparing the results obtained from our solar model and the real solar data, due to the imperfections of our model. An initial step for a better harvester model could be including the effects of the seasonal changes of solar data in the model of the solar harvesting process. Effects of battery leakage and spent energy during stand-by can also be reflected on the battery model. As a secondary potential research direction, DP algorithms require a perfect system model and large state spaces in such models makes it infeasible to use Policy Iteration techniques, also known as the "curse of dimensionality" problem for large MDPs. Advanced harvester models would probably require larger state spaces and DP methods would probably collapse in these cases. The use of modern RL techniques in such scenarios to solve advanced system models appears to be a legitimate future research direction.

Bibliography

[1] A. Whitmore, A. Agarwal, and L. Da Xu, "The Internet of Things—a survey of topics and trends," *Information Systems Frontiers*, vol. 17, pp. 261–274, Apr 2015.

[2] A. Ali, Y. Ming, S. Chakraborty, and S. Iram, "A comprehensive survey on real-time applications of WSN," *Future Internet*, vol. 9, no. 4, 2017.

[3] F. Xia, "Wireless sensor technologies and applications," *Sensors*, vol. 9, no. 11, pp. 8824–8830, 2009.

[4] D. Bhattacharyya, T.-h. Kim, and S. Pal, "A comparative study of Wireless Sensor Networks and their routing protocols," *Sensors*, vol. 10, no. 12, pp. 10506–10523, 2010.

[5] C. Buratti, A. Conti, D. Dardari, and R. Verdone, "An overview on Wireless Sensor Networks technology and evolution," *Sensors (Basel, Switzerland)*, vol. 9, no. 9, p. 6869—6896, 2009.

[6] S. Swapna Kumar and K. Kashwan, "Research study of energy harvesting in Wireless Sensor Networks," *International Journal of Renewable Energy Research*, vol. 3, pp. 745–753, 01 2013.

[7] G. Xu, W. Shen, and X. Wang, "Applications of Wireless Sensor Networks in marine environment monitoring: A survey," *Sensors*, vol. 14, no. 9, pp. 16932–16954, 2014.

[8] H. Jawad, R. Nordin, S. Gharghan, A. Jawad, and M. Ismail, "Energy-efficient Wireless Sensor Networks for precision agriculture: A review," *Sensors (Switzerland)*, vol. 17, 8 2017.

[9] G. Zhou, L. Huang, W. Li, and Z. Zhu, "Harvesting ambient environmental energy for Wireless Sensor Networks: A survey," *Journal of Sensors*, vol. 2014, pp. 1–20, 06 2014.

[10] A. Castagnetti, A. Pegatoquet, C. Belleudy, and M. Auguin, "A framework for modeling and simulating energy harvesting WSN nodes with efficient power management policies," *EURASIP Journal on Embedded Systems*, vol. 2012, p. 8, Oct 2012.

[11] F. A. Aoudia, M. Gautier, and O. Berder, "RLMan: An energy manager based on reinforcement learning for energy harvesting Wireless Sensor Networks," *IEEE Transactions on Green Communications and Networking*, vol. 2, pp. 408–417, June 2018.

[12] D. Chen and P. Varshney, "QoS support in Wireless Sensor Networks: A survey," *Proceedings of the International Conference on Wireless Networks, ICWN'04*, vol. 1, pp. 227–233, 01 2004.

[13] S. Sojan and R. Kulkarni, "A comprehensive review of energy harvesting techniques and its potential applications," *International Journal of Computer Applications*, vol. 139, pp. 14–19, 04 2016.

[14] A. Kansal, J. Hsu, S. Zahedi, and M. B. Srivastava, "Power management in energy harvesting sensor networks," *ACM Trans. Embed. Comput. Syst.*, vol. 6, Sept. 2007.

[15] J. Hsu, S. Zahedi, A. Kansal, M. Srivastava, and V. Raghunathan, "Adaptive duty cycling for energy harvesting systems," in *Proceedings of the 2006 International Symposium on Low Power Electronics and Design*, ISLPED '06, (New York, NY, USA), pp. 180–185, ACM, 2006.

[16] C. M. Vigorito, D. Ganesan, and A. G. Barto, "Adaptive control of duty cycling in energy-harvesting Wireless Sensor Networks," in *2007 4th Annual*

IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, pp. 21–30, June 2007.

[17] Y. Gu, T. Zhu, and T. He, "Esc: Energy synchronized communication in sustainable sensor networks," in *2009 17th IEEE International Conference on Network Protocols*, pp. 52–62, Oct 2009.

[18] T. Zhu, Z. Zhong, Y. Gu, T. He, and Z.-L. Zhang, "Leakage-aware energy synchronization for Wireless Sensor Networks," in *Proceedings of the 7th International Conference on Mobile Systems, Applications, and Services*, MobiSys '09, (New York, NY, USA), pp. 319–332, ACM, 2009.

[19] V. Sharma, U. Mukherji, V. Joseph, and S. Gupta, "Optimal energy management policies for energy harvesting sensor nodes," *IEEE Transactions on Wireless Communications*, vol. 9, pp. 1326–1336, April 2010.

[20] C. Moser, J.-J. Chen, and L. Thiele, "An energy management framework for energy harvesting embedded systems," *J. Emerg. Technol. Comput. Syst.*, vol. 6, pp. 7:1–7:21, June 2008.

[21] M. A. Alsheikh, D. T. Hoang, D. Niyato, H. Tan, and S. Lin, "Markov Decision Processes with applications in Wireless Sensor Networks: A survey," *IEEE Communications Surveys Tutorials*, vol. 17, pp. 1239–1267, thirdquarter 2015.

[22] X. Wu, J. Yang, and J. Wu, "Optimal status update for age of information minimization with an energy harvesting source," *IEEE Transactions on Green Communications and Networking*, vol. 2, pp. 193–204, March 2018.

[23] B. T. Bacinoglu, E. T. Ceran, and E. Uysal-Biyikoglu, "Age of information under energy replenishment constraints," in *2015 Information Theory and Applications Workshop (ITA)*, pp. 25–31, Feb 2015.

[24] M. Costa, M. Codreanu, and A. Ephremides, "On the age of information in status update systems with packet management," *IEEE Transactions on Information Theory*, vol. 62, pp. 1897–1910, April 2016.

[25] A. Valehi and A. Razi, "Maximizing energy efficiency of cognitive Wireless Sensor Networks with constrained age of information," *IEEE Transactions on Cognitive Communications and Networking*, vol. 3, pp. 643–654, Dec 2017.

[26] M. L. Littman, T. L. Dean, and L. P. Kaelbling, "On the complexity of solving Markov Decision Problems," in *Proceedings of the Eleventh Conference on Uncertainty in Artificial Intelligence*, UAI'95, (San Francisco, CA, USA), pp. 394–402, Morgan Kaufmann Publishers Inc., 1995.

[27] R. S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning*. Cambridge, MA, USA: MIT Press, 2nd ed., 2018.

[28] S. Ulukus, A. Yener, E. Erkip, O. Simeone, M. Zorzi, P. Grover, and K. Huang, "Energy harvesting wireless communications: A review of recent advances," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 360–381, March 2015.

[29] J. A. Khan, H. K. Qureshi, and A. Iqbal, "Energy management in Wireless Sensor Networks: A survey," *Computers and Electrical Engineering*, vol. 41, pp. 159 – 176, 2015.

[30] K. S. Adu-Manu, N. Adam, C. Tapparello, H. Ayatollahi, and W. Heinzelman, "Energy-harvesting Wireless Sensor Networks (EH-WSNs): A review," *ACM Trans. Sen. Netw.*, vol. 14, pp. 10:1–10:50, Apr. 2018.

[31] J. Yang and S. Ulukus, "Optimal packet scheduling in an energy harvesting communication system," *IEEE Transactions on Communications*, vol. 60, pp. 220–230, January 2012.

[32] K. Tutuncuoglu, A. Yener, and S. Ulukus, "Optimum policies for an energy harvesting transmitter under energy storage losses," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 467–481, March 2015.

[33] M. Cha, M. Kim, M. Kim, and H. Choo, "Adaptive duty-cycling based on group size for energy balance of sensor nodes in Wireless Sensor Networks," in *Proceedings of the 2011 ACM Symposium on Research in Applied Computation*, RACS '11, (New York, NY, USA), pp. 135–140, ACM, 2011.

[34] N. Michelusi, K. Stamatiou, and M. Zorzi, "Transmission policies for energy harvesting sensors with time-correlated energy supply," *IEEE Transactions on Communications*, vol. 61, pp. 2988–3001, July 2013.

[35] H. Liu, A. Chandra, and J. Srivastava, "eSENSE: Energy efficient stochastic sensing framework scheme for wireless sensor platforms," in *Proceedings of the 5th International Conference on Information Processing in Sensor Networks*, IPSN '06, (New York, NY, USA), pp. 235–242, ACM, 2006.

[36] A. Sinha and A. Chandrakasan, "Dynamic power management in Wireless Sensor Networks," *IEEE Design Test of Computers*, vol. 18, pp. 62–74, March 2001.

[37] M. Gorlatova, A. Wallwater, and G. Zussman, "Networking low-power energy harvesting devices: Measurements and algorithms," *IEEE Transactions on Mobile Computing*, vol. 12, pp. 1853–1865, September 2013.

[38] W. B. Heinzelman, A. L. Murphy, H. S. Carvalho, and M. A. Perillo, "Middleware to support sensor network applications," *IEEE Network*, vol. 18, pp. 6–14, Jan 2004.

[39] S. Reddy and C. R. Murthy, "Profile-based load scheduling in wireless energy harvesting sensors for data rate maximization," in *2010 IEEE International Conference on Communications*, pp. 1–5, May 2010.

[40] K. Tutuncuoglu and A. Yener, "Optimum transmission policies for battery limited energy harvesting nodes," *IEEE Transactions on Wireless Communications*, vol. 11, pp. 1180–1189, March 2012.

[41] B. Devillers and D. Gündüz, "A general framework for the optimization of energy harvesting communication systems with battery imperfections," *Journal of Communications and Networks*, vol. 14, pp. 130–139, April 2012.

[42] C. Tunc, "Energy management in energy harvesting wireless sensor nodes with lifetime contraints," Master's thesis, Bilkent University, June 2016.

[43] S. Zhang, *Modeling, Analysis and Design of Energy Harvesting Communication Systems*. PhD thesis, University of Rochester, 2013.

[44] A. Gosavi, "Reinforcement learning: A tutorial survey and recent advances," *INFORMS Journal on Computing*, vol. 21, pp. 178–192, 05 2009.

[45] D. P. Bertsekas, *Dynamic Programming and Optimal Control*. Athena Scientific, 2nd ed., 2000.

[46] O. Alagoz, H. Hsu, A. J. Schaefer, and M. S. Roberts, "Markov Decision Processes: A tool for sequential decision making under uncertainty," *Medical Decision Making*, vol. 30, no. 4, pp. 474–483, 2010. PMID: 20044582.

[47] A. Susu, M. Magno, A. Acquaviva, D. Atienza, and G. De Micheli, "Reconfiguration strategies for environmentally powered devices: Theoretical analysis and experimental validation," *Transactions on High-Performance Embedded Architectures and Compilers*, vol. 1, pp. 341–360, 01 2007.

[48] A. E. Susu, A. Acquaviva, D. Atienza, and G. D. Micheli, "Stochastic modeling and analysis for environmentally powered wireless sensor nodes," in *2008 6th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops*, pp. 125–134, April 2008.

[49] P. Poggi, G. Notton, M. Muselli, and L. Alain, "Stochastic study of hourly total solar radiation in Corsica using a Markov model," *International Journal of Climatology*, vol. 20, pp. 1843 – 1860, 11 2000.

[50] C. E. Kement, "Design and optimization of a rate adaptation algorithm for energy harvesting transmitters," Master's thesis, Middle East Technical University, September 2014.

[51] L. Guntupalli, M. Gidlund, and F. Y. Li, "An on-demand energy requesting scheme for wireless energy harvesting powered IoT networks," *IEEE Internet of Things Journal*, vol. PP, pp. 1–1, 06 2018.

[52] L. Guntupalli, F. Y. Li, and M. Gidlund, "Energy harvesting powered packet transmissions in duty-cycled WSNs: A DTMC analysis," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, Dec 2017.

[53] M. Ku, Y. Chen, and K. J. R. Liu, "Data-driven stochastic models and policies for energy harvesting sensor communications," *IEEE Journal on Selected Areas in Communications*, vol. 33, pp. 1505–1520, Aug 2015.

[54] National Renewable Energy Laboratory, "National solar radiation database." `https://rredc.nrel.gov/solar/old_data/nsrdb`. [Online; accessed 15-August-2018].