SECURE COVERAGE CONTROL IN SHORT-RANGE WIRELESS SENSOR NETWORKS USING MULTI-AGENT SYSTEMS

Akkachai Phuphanin

A Thesis Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Engineering in Telecommunication Engineering

Suranaree University of Technology

Academic Year 2011

การควบคุมพื้นที่ครอบคลุมอย่างปลอดภัยในเครือข่ายเซ็นเซอร์ไร้สายระยะสั้น โดยใช้ระบบมัลติเอเจนท์

นายอรรคชัย ภูผานิล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2554

SECURE COVERAGE CONTROL IN SHORT-RANGE WIRELESS SENSOR NETWORKS USING MULTI-AGENT SYSTEMS

Suranaree University of Technology has approved this thesis submitted in partial fulfillment of the requirements for a Master's Degree.

Thesis Examining Committee

_____
(Asst. Prof. Dr. Peerapong Uthansakul)
Chairperson

_____
(Asst. Prof. Dr. Wipawee Hattagam)
Member (Thesis Advisor)

_____
(Asst. Prof. Flt.Lt. Dr. Prayoth Kumsawat)
Member

_____
(Prof. Dr. Sukit Limpijumnong)
Vice Rector for Academic Affairs

_____
(Assoc. Prof. Gp.Cpt. Dr. Vorapot Khompis)
Dean of Institute of Engineering

อรรคชัย ภูผานิล : การควบคุมพื้นที่ครอบคลุมอย่างปลอดภัยในเครือข่ายเซ็นเซอร์ไร้สายระยะสั้นโดยใช้ระบบมัลติเอเจนท์ (SECURE COVERAGE CONTROL IN SHORT-RANGE WIRELESS SENSOR NETWORKS USING MULTI-AGENT SYSTEMS) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม, 91 หน้า

เครือข่ายเซ็นเซอร์ไร้สายประกอบด้วยอุปกรณ์เซ็นเซอร์ที่ทำงานอย่างกระจายและอัตโนมัติ โดยแต่ละเซ็นเซอร์สามารถติดต่อสื่อสารกับเซ็นเซอร์อื่น ๆ เพื่อทำการตรวจรู้และประมวลข้อมูลร่วมกัน อย่างไรก็ตามเนื่องจากข้อจำกัดของทรัพยากรบนบอร์ดเซ็นเซอร์ จึงมีความจำเป็นต้องลดการใช้พลังงาน รูปแบบการควบคุมพื้นที่ครอบคลุมซึ่งมักอาศัยความร่วมมือระหว่างโหนดจึงเป็นสิ่งจำเป็นในการเฝ้าระวังสิ่งแวดล้อมอย่างมีประสิทธิภาพ อย่างไรก็ตามโหนดอาจถูกโจมตีจากโหนดที่เป็นอันตรายซึ่งจะมีผลกระทบโดยตรงต่อพื้นที่สังเกตการณ์ ดังนั้นงานวิจัยนี้จึงมุ่งเน้นไปในการพัฒนาการควบคุมพื้นที่ครอบคลุมที่ไม่ซับซ้อนและทำงานอย่างกระจายตัว และยังสามารถป้องกันการโจมตีในโครงข่ายเซ็นเซอร์ไร้สายได้ งานวิจัยนี้นำเสนอวิธีการของการควบคุมพื้นที่ครอบคลุมที่ใช้การเรียนรู้แบบมัลติเอเจนท์รีอินฟอร์สเมนท์ร่วมกับโปรโตคอลรักษารูปร่างเครือข่าย เพื่อที่จะได้กลยุทธ์การจัดการพื้นที่ครอบคลุมที่ปลอดภัยและใกล้เคียงกลยุทธ์ที่เหมาะสมที่สุด โดยวิธีการที่นำเสนอถูกออกแบบให้โหนดหนึ่ง ๆ ทำการตัดสินใจ โดยพิจารณาข้อมูลจากโหนดข้างเคียงหลายโหนด เพื่อรับมือกับข้อมูลที่ผิดพลาดจากโหนดที่เป็นอันตราย

ผลการทำลองชี้ให้เห็นว่าวิธีการที่นำเสนอมีความทนทานและมีประสิทธิภาพกว่าโดยได้รับพื้นที่ครอบคลุมต่อพลังงานหนึ่งหน่วยที่มีค่าสูงและได้รับเปอร์เซ็นต์ของพื้นที่ครอบคลุมมากกว่าวิธีการดีวีเอฟ (DVF) เดิมถึง 6 - 12% ภายใต้การโจมตีแบบไม่ให้หลับและการโจมตีแบบให้หลับ นอกจากนี้จากการศึกษาการโจมตีแบบยึดครองโหนดในเครือข่ายซึ่งทำให้โหนดแลกเปลี่ยนข้อมูลที่คลาดเคลื่อนระหว่างโหนด พบว่าวิธีการที่นำเสนอได้รับพื้นที่ครอบคลุมที่มากกว่าวิธีการเดิมถึง 19% 32% และ 37% เมื่อเกิดการยึดครองโหนดอย่างเดียวหรือเกิดการยึดครองโหนดร่วมกับการโจมตีแบบไม่ให้หลับ และการโจมตีแบบให้หลับตามลำดับ ผลการทดลองชี้ให้เห็นว่าวิธีการที่รวมโปรโตคอลรักษารูปร่างเครือข่ายนั้นสามารถลดความเสี่ยงของการโดนโจมตีได้อย่างมีประสิทธิภาพ

สาขาวิชาวิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2554 ลายมือชื่ออาจารย์ที่ปรึกษา_____

AKKACHAI PHUPHANIN : SECURE COVERAGE CONTROL IN SHORT-RANGE WIRELESS SENSOR NETWORKS USING MULTI-AGENT SYSTEMS. THESIS ADVISOR : ASST. PROF. WIPAWEE HATTAGAM, Ph.D., 91 PP.

WIRELESS SENSOR NETWORKS/MULTI-AGENT/MALICIOUS NODE/ COVERAGE CONTROL

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous sensory devices that can communicate with each other to perform sensing and data processing cooperatively. However, due to limited onboard resources, power consumption must be reduced. Coverage control schemes, where nodes typically cooperate with each other, are therefore essential for effective condition monitoring of the environment. However, nodes are vulnerable to malicious attacks which directly affect the area under observation. Therefore, the underlying aim of this thesis is to develop a distributed light-weight coverage control scheme which countermeasures against malicious attacks in WSNs. This thesis proposed a coverage control algorithm based on multi-agent reinforcement learning integrated with a topology maintenance protocol in order to obtain a secure and near-optimal coverage allocation strategy. The proposed algorithm was designed in such a way that a node makes its decisions by considering inputs from multiple neighboring nodes in order to tolerate false messages created by malicious nodes.

Simulation results showed that our algorithm was more robust and efficient by consistently attaining higher coverage per unit energy consumed, and achieving 6-12% of coverage greater than the original DVF algorithm under sleep deprivation

and snooze attacks. Furthermore, the network substitution attack was studied where inaccurate information was exchanged between nodes. The proposed algorithm gained up to 19%, 32% and 37% of coverage higher than the DVF algorithm for the network substitution attack alone, and network substitution paired with sleep deprivation and snooze attacks, respectively. By integrating the secure topology maintenance protocol, our results suggested that vulnerability to such attacks can efficiently be reduced.

School of Telecommunication Engineering Student's Signature _____

Academic Year 2011 Advisor's Signature _____

ACKNOWLEDGEMENT

I am grateful to all those, who by their direct or indirect involvement have helped in the completion of this thesis.

First and foremost, I would like to express my sincere thanks to my thesis advisors, Asst. Prof. Dr. Wipawee Hattagam for her invaluable help and constant encouragement throughout the course of this research. I am most grateful for her teaching and advice, not only the research methodologies but also many other methodologies in life. I would not have achieved this far and this thesis would not have been completed without all the support that I have always received from her.

I would also like to thank Asst. Prof. Dr. Peerapong Uthansakul and Asst. Prof. Flt.Lt. Dr. Prayoth Kumsawat for accepting to serve in my committee.

I would also like to thank the Telecommunication Research Industrial and Development Institute (TRIDI), National Telecommunication Commission Fund, Thailand for financially the financial support received throughout my postgraduate studies. I gratefully acknowledge Prof. Dr. Ivan Andonovic, Dr. Kae Hsiang Kwong and for their technical advice, assistance, during my visit at the Centre for Dynamic Intelligent Communication (CIDCOM), University of Strathclyde, Scotland.

My sincere appreciation goes to Ms. Pranitta Arthan for their valuable administrative support during the course of my dissertation.

Finally I am most grateful to my parents and my friends both in both masters and doctoral degree courses for all their support throughout the period of this research

Akkachai Phuphanin

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

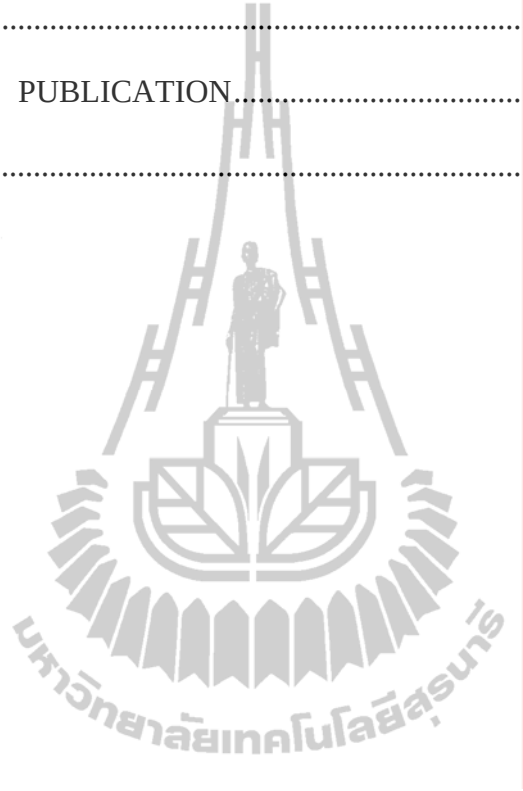
Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page



LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

SYMBOLS AND ABBREVIATIONS

WSNs	=	Wireless sensor networks
MAS	=	Multi-agent system
RL	=	Reinforcement learning
MARL	=	Multi-agent reinforcement learning
TMP	=	Topology maintenance protocols
FMQ	=	Frequency maximum Q-learning
DVF	=	Distributed value function
MDP	=	Markov decision process
t	=	Time step index
s_t	=	State of the process at time t
S	=	State space
s	=	Current state
s'	=	Next state
A	=	Action space
a	=	Action
$E[\cdot]$	=	Expectation operator
β	=	Discount factor
$R(s,a,s')$	=	Expected reward given any current state s and an action a with any next state s'
r	=	Reward

SYMBOLS AND ABBREVIATIONS (Continued)

P	=	State transition probability matrix
π	=	Policy
π^*	=	Optimal policy
$P[A]$	=	Distribution over the action space
$Q_t^{\pi}(s,a)$	=	The action-value function of a given policy π associated to each state-action pair (s,a)
R_t	=	Expected discounted return of the agent
$E^{\pi}\{\cdot\}$	=	Expectation operator under the policy π
$V^{\pi}(s)$	=	Value function of a state (s) under policy π
$V^*(s)$	=	Value function of a state (s) under optimal policy π^*
α	=	Learning rate
$Q^*(s,a)$	=	The action-value function of a given optimal policy π^* associated to each state-action pair (s,a)
i	=	Agent index
ξ	=	degree of inaccuracy

CHAPTER I

INTRODUCTION

This chapter introduces the background on coverage control in multi-agent wireless sensor networks (WSNs) and presents the possible malicious attacks on WSNs. It also presents the motivation for applying reinforcement learning (RL) to achieve maximum coverage, maximize trade-off between area coverage with energy consumption and how to handle such malicious node attacks which is the main focus of this thesis.

1.1 Significance of the problem

A wireless sensor network (WSNs) is a wireless network consisting of spatially distributed autonomous sensory devices that can communicate with each other to perform sensing and data processing cooperatively (Stankovic, A.J., 2008). The overall objective of WSNs is to provide a low-cost solution to gather physical data from the environment, such as noise, pressure, light, vibration or temperature, at different locations, observation and transmit it to a base station. The most common energy storage device used in a sensor node is a battery which is suitable for a micro-sensor with very low power consumption (Yick, J., et al., 2008). Therefore, WSNs promises unlimited potential for numerous application areas including environmental (Chitnis, L., et al., 2009), medical, military, transportation, entertainment, crisis management, homeland defense, and smart space (Han, X., et al., 2010), (Yu, L., et al., 2007), (Li, M., et al., 2006).

Since WSNs are based on limited power sources and must be extremely small, their battery supplies are much more constrained. Therefore, processing power, memory and wireless communication abilities are very limited in order to reduce power consumption.

Most research works consider WSNs which consist of sensor nodes that cooperate with one another. For example, Qiu, W., et al. (2008); Liu, Z., et al. (2008); Chen, W., et al. (2007); Wang, C., and Wu, W. (2009) investigated routing protocols, which require energy-awareness at all layers of the protocol stack. At the network layer, the aim is to cooperatively set up energy-efficient routes and reliably relay data from sensor nodes to the sink so that the lifetime of the network is maximized. The authors in (Seah, M.W.M., et al., 2007); (Munir, S.A., et al., 2007) proposed coordination algorithms between wireless sensor networks node which aim at maximizing the coverage of the sensing field while minimizing the total energy consumption, thereby increasing the lifetime of network.

To encourage coordination between sensor nodes, multi-agent systems (MAS) have been applied in WSNs. MAS have potential to tackle the resource constraints inherent in these networks by efficiently coordinating the activities among the nodes. MAS are made up of a number of cooperative agents, each with its own set of states and actions, which must coordinate with one another in order to maximize the overall benefit for all agents. Seah, M.W.M., et al. (2007); Tham, C.K., and Renaud, J.C. (2005); Guestrin, C., et al. (2002), showed that cooperation between sensors in the same area can be achieved by a distributed learning algorithm based on distributing information of their value functions (a function that gives an estimate of how well an agent has performed so far at a given state) among agents. In particular, each agent

achieves cooperative decisions by exchanging the value of the state each agent lands in with its neighbors.

However, cooperative behaviors between sensor nodes may not always be readily available. This may be caused by selfish behaviors of certain sensor nodes to conserve their energy. Furthermore, Vaz de Melo, P.O.S., et al. (2008); Wu, M.Y., et al. (2005) studied the conditions of node cooperation in packet forwarding in overlay WSNs where two WSNs under different authorities coexist in the same region. They showed that there was no guarantee that node cooperation will be beneficial to both WSNs. In particular, node cooperation depended on a number of factors, such as, network density, hostility of the environment, and network configuration, etc.

Apart from dealing with non-cooperative behavior from sensor nodes under different network authorities, sensor nodes may also experience non-cooperative behavior from attacks by other malicious nodes inside or outside of the network. These attacks may be used to reduce the lifetime of the sensor network, or to degrade the functionality of the sensor application by reducing the network connectivity and the sensing coverage that can be achieved. Three types of attacks are common in WSNs (Gabrielli, A., et al., 2011). Firstly, the *sleep deprivation* attack is an attack which the adversary tries to induce a node in a specific area to remain active. This attack has two effects, (i) it increases the energy expenditure of sensor nodes and reduces the estimated lifetime of the network. (ii) in the case of a densely populated area, it can lead to increased energy consumption due to congestion and contention at the data link layer. Secondly, the *snooze attack* is an attack which the adversary forces the nodes to remain in the sleeping state. This type of attack can be applied to the whole network or to a subset of nodes. In the latter case, the adversary can launch an

attack to jeopardize the connectivity of the network or to reduce the sensing coverage in a region. For example, an adversary can selectively turn off nodes that are monitoring an intruder's path through an area in which a sensor field has been deployed for surveillance. Thirdly, the *network substitution attack* is an attack which the adversary deploys some nodes, which are in a set elected by the TMP, to gain control of part of or the entire network. Once these nodes are under control, the adversary can carry out other attacks such as sharing false or inaccurate information or readings with other nodes. This type of attack is difficult to detect since the compromised node can still maintain connectivity and appear as it were operating as normal.

To prevent such attacks, topology maintenance protocols (TMPs), such as SPAN (Chen, B., et al., 2002), ASCENT (Cerpa, A., and Estrin, D., 2004), PEAS (Ye, F., et al., 2003), and CCP (Wang, X., et al., 2003) were critical to the operation of wireless sensor networks. These protocols aimed to increase the lifetime of the sensor network by maintaining only a subset of nodes in an active or awake state, while turning off redundant nodes. There had to be enough active nodes to maintain the connectivity of the network as well as to obtain sensing coverage in the area where the sensor network was deployed. One research was Karlof, C., and Wagner, D. (2003) work that pointed out the security issues on topology maintenance protocols. However, Karlof, C., and Wagner, D. (2003) only described the snooze attack against GAF (Xu, Y., et al., 2001), SPAN (Chen, B., et al., 2002) authentication protocols. They did not discuss effects of the snooze attack on the sensing coverage. Moreover, they did not take sleep deprivation and network substitution attacks into consideration either. In another related work, Stajano, F., and Anderson, R. (1999) introduced the

problem of the sleep deprivation attack. However, they neither considered this attack in topology maintenance protocols nor described any countermeasures. Gabrielli, A., et al. (2011) proposed a meta-protocol (Meta-TMP) for countering malicious nodes by including authentication mechanisms that can be used to prevent outsider attacks and certain insider attacks.

So far, several works on coverage control in the literature have assumed that all nodes in the WSN were cooperative (Schneider, J., et al., 1999); (Lauer, M., et al., 2000); (Renaud, J.C., and Tham, C.K., 2006). Some applications of these networks are critical, and sensors are deployed in a hostile environment. For this reason, it is mandatory to develop solutions that make WSNs resilient to malicious behaviors. Otherwise, it is possible that an adversary can compromise the network functioning. For example, the wireless nature of the communication medium makes the data exchange vulnerable to eavesdropping attacks. The lack of protection mechanism in the network devices makes the sensors vulnerable to physical attacks and compromises the data stored inside the sensor. Moreover, the sensor hardware constraints such as the memory size and the energy supply can make the security techniques used in traditional networks unaffordable in WSNs.

A simple, adaptive coverage control method which is not computationally or resource demanding is therefore needed. Reinforcement learning (RL) techniques (Sutton, R., and Barto, A., 1998); (Kaelbling, L., et al., 1996) have been a common approach to coordinately and cooperatively improve the network performance in WSNs (Tham, C.K., and Renaud, J.C., 2005). RL is defined as the problem faced by a learner of how to take actions, or make optimal decisions, through trial and error interactions with a dynamic environment. A common RL method called Q-learning is

an algorithm which directly approximates the optimal action-value function (a function that describes how good an action was, given that the agent is at a particular state). Each learning agent takes an action, receives a reward, updates local information with an input from the environment, and repeats the process by learning its own optimal strategy. Renaud, J.C., and Tham, C.K. (2006) proposed the Frequency maximum Q-learning (FMQ) to encourage cooperative coverage control in WSNs. FMQ was based on Q-learning, which enabled autonomous self learning, adaptive applications with inherent support for efficient resource or task management. Their results when compared with that of (Tham, C.K., and Renaud, J.C., 2005) showed that the FMQ algorithm consumed more energy and received more average rewards than the coverage control approach in (Tham, C.K., and Renaud, J.C., 2005).

The Multi-agent reinforcement learning (MARL) approach called Distributed value function (DVF) in (Tham, C.K., and Renaud, J.C., 2005) was promising and warranted potential use for coverage control in WSNs. However, preliminary results in this thesis showed that when malicious nodes were present in the system, such as in the case when nodes were under sleep deprivation, snooze attacks and network substitution attacks, the performance of the WSN was affected by an energy consumption increase and average reward reduction. This suggested that the DVF algorithm alone strongly relied on cooperation between nodes. To deal with malicious node behaviors and vulnerability to attacks, we proposed to solve this problem by using a topology maintenance protocol (Gabrielli, A., et al., 2011) in order to obtain a secure and near-optimal coverage allocation strategy under energy constraints. Such topology maintenance protocol should be designed in such a way that a node makes its decision whether to sleep or remain active, based on inputs from multiple neighboring

nodes in order to be resilient to false messages or non-cooperative behaviors by malicious nodes.

1.2 Research objectives

1.2.1 To study cooperative coverage control schemes in wireless sensor networks and study mechanisms which enhance secure operation for three types of attacks on sensors, i.e. sleep deprivation, snooze and network substitution attacks.

1.2.2 To study how such attacks affect the performance of the DVF algorithm and investigate the integration of TMP and DVF algorithm to deal with such malicious node behaviors.

1.3 Assumptions

1.3.1 Cooperative coverage control is beneficial when the network is sparse or when the environment is hostile.

1.3.2 A secure topology maintenance control mechanism integrated with the DVF algorithm to deal with such non-cooperative behavior can provide a more reliable and secure coverage control than the normal DVF approach.

1.4 Scope of the Research

1.4.1 The coverage of the wireless sensor network consisted of 5 different agents and 40 different agents which were located in the same region.

1.4.2 Decision methods for choosing the maximum coverage strategy in WSNs have been studied.

1.4.3 Non-cooperative sensor node behavior and multi-agent reinforcement learning (MARL) methods have been studied.

1.4.4 The performance of the Distributed Value Function algorithm for coverage control in a WSN have been evaluated under three types of attacks on a sensor node i.e., sleep deprivation, snooze and network substitution attacks.

1.4.5 Simulations have been carried out by Visual C++. Two algorithms have been compared, namely, the Distributed Value Function (DVF) and the proposed integrated TMP with DVF algorithm which is the algorithm proposed in this thesis to handle non-cooperative behavior. The experimental results have been analyzed to find secure and optimal coverage strategies under energy constraints.

1.4.6 Compared metrics include:

- **Average reward**

In the MARL framework, an action taken by agent changes the state of the environment and of the agent. A scalar reward is then returned to the agent from the environment. The agent should behave so as to maximize the received rewards, or more specifically, the long-term average reward. This metric is to measure the amount of incentives which encourage sensor nodes to cooperate.

- **Coverage**

A coverage is the area being monitored by the sensor nodes. This metric is to measure the coverage area among sensor nodes.

- **Average energy consumption**

The energy consumption is the energy spent on communication and computation. This metric is to verify which algorithm provides the best coverage with minimal energy consumption. The average energy consumption is the total energy consumption divided by the total duration of simulation.

- **Trade-off**

The trade-off represents the number of illuminated cells (i.e. the coverage) per unit energy consumed. The trade-off is defined as the ratio of achieved coverage over the average energy consumption.

1.5 Expected Usefulness

1.5.1 To conceptually show that the MARL can be applied to find a suitable policy for secure coverage control in WSNs with malicious nodes.

1.5.2 To develop a coverage control algorithm in wireless sensor networks which can handle non-cooperative behavior and use energy efficiently.

1.6 Synopsis of Thesis

The remainder of this thesis is organized as follows.

Chapter 2 presents the theoretical background which is the foundation for the contributions of this thesis. In this chapter, the definition of single agent and multi-agent RL are presented. Then, the concept of the Markov decision process formulation is reviewed. A brief introduction to an existing tools used for solving the coverage control problem called the Distributed Value Function algorithms is then provided.

Chapter 3 investigates the coverage control problem in multi-agent WSNs. Studied secure coverage control in wireless sensor networks with malicious nodes using multi-agents. In this chapter, we propose to alleviate malicious node attacks by using a topology maintaenance protocol in order to obtain a secure and near-optimal coverage allocation strategy under energy constraints.

Chapter 4 summarizes all the findings and contributions in this thesis and points out possible future research directions.

CHAPTER II

BACKGROUND THEORY

2.1 Introduction

This thesis proposed a multi-agent secure coverage control scheme for wireless sensor networks. A wireless sensor network is a wireless network consisting of spatially distributed autonomous devise using sensor that can communicate with each other to perform sensing and data processing cooperatively. The overall objective of WSN is to provide a low-cost solution to gather physical data from the environment, such as noise, pressure, light, observation and transmit power sources and must be extremely small, hence their battery capacity constraints are much higher. Therefore, processing power, memory and wireless communication abilities are very limited to reduce power consumption.

Due to scarce battery supply, maintaining and maximizing coverage control has become a challenging issue in WSNs. Distributed self-adaptive coverage control schemes are of particular interest as WSN are typically deployed in dynamically changing environment which may be difficult to access and manually configure. Such autonomous coverage control can be achieved by multi-agent systems (MAS). The implementation of MAS in a WSN require sensor-actuator node with processing capability which enable these node to perform tasks in a coordinated manner to achieve some desired system-wide objective.

This thesis proposed the application of multi-agent reinforcement learning (RL) to address the coverage control scheme in WSNs. Reinforcement learning (Sutton, R., and Barto, A., 1998) is a computational approach for autonomous goal-directed learning and decision-making. RL is different from other computational approaches in that RL emphasizes on learning by the agent itself from direct interaction with its environment, without relying on any supervision or a complete model of the environment. In a distributed learning and decision-making system such as a multi-agent system, the system's behavior is influenced by a team of agents acting simultaneously and independently (Tham, C.K., and Renaud, J.C., 2005). Thus, the state dynamics of the environment are likely to change more frequently than in the single agent case. Because it is a learning method that does not need any prior model of the environment and can perform online learning, RL is well-suited for cooperative multi-agent systems.

Multi-agent systems (MAS) differ from single-agent systems in that there are many different agents that learn a task. Furthermore, all of the agents' actions affect the state of the environment. Thus, the optimal policy not only relies on only one agent, but on all agents. There are works which directly applied a commonly used RL method called Q-learning to multi-agent systems whereby each agent disregards other agents in the system and takes action to maximize its own benefit. By neglecting the presence of other agents, suboptimal decisions are likely to be achieved. Therefore, an individual agent should consider the effect of joint actions from other agents as well to achieve better decisions in MAS.

To promote cooperation between sensor nodes, multi–agent systems (MAS) have been applied in WSNs. MAS has the potential to tackle the resource constraints inherent in these networks by efficiently coordinating the activities among the nodes. MAS is made up of a number of agents, each with its own set of states and actions. Each agent must coordinate with one another in order to maximize the overall benefit for all agents. Seah, M.W.M., et al. (2007) examined how coordination between the wireless sensor nodes could lead to maximization of coverage of the sensing field as well as minimization of the total energy consumption, thereby increasing the life time of network. They tested three algorithms, i.e. the fully distributed Q-learning, the Distributed Value Function (DVF) and the coordinated algorithm (COORD). Guestrin, C., et al. (2002) presented an algorithm for multi-agent reinforcement learning called Coordinated reinforcement learning. In this algorithm, agents coordinate both their action selection and their parameter update. Within the limits of their parametric representation, the agent determines a joint action without explicitly considering every possible action in their exponentially large joint action space. Tham, C.K., and Renaud, J.C. (2005) implemented a multi-agent system on a wireless sensor network comprising sensor-actuator nodes with processing capability. Their approach enabled these nodes to perform tasks in a coordinate manner to achieve maximum coverage. Tham, C.K., and Renaud, J.C. (2005) considered the implementation of several algorithms including the Indlearners algorithm, the Distributed value function (DVF) algorithm and the Opt DRL algorithm. The optimal algorithm for coverage control in multi-agent system was found to be the Distributed value function (DVF) algorithm, in terms of trade-off between the achieved area coverage and energy consumption. The Distributed value function (DVF) algorithm extended a commonly

used RL method called, Q-learning, to encourage cooperative behavior between agents in multi-agent systems to achieve maximum coverage area in the network. In this thesis, this algorithm was used as a benchmark for coverage control comparison in presence of malicious node attacks in the WSN.

This chapter serves as an introductory to the fundamental theory of reinforcement learning which the basis of the contribution of this thesis. The next section explains the concept of single-agent and multi-agent RL. The following section provides a theoretical background on Markov decision process (MDP). A description of reinforcement learning is given in section 2.4. Section 2.5 presents the multi-agent Q-learning. Section 2.6 presents the distributed reinforcement learning and a summary is presented in the final section.

2.2 Single-agent and multi-agent systems

2.2.1 Single-agent systems

Before studying and categorizing MAS, we first consider the most obvious centralized, single-agent systems. Centralized systems have a single agent which makes all the decisions. A single-agent system may have multiple entities, several actuators, or several physically separated components. However, if each entity sends its perceptions to and receives its actions from a single central process, then there is only a single agent in the central process. The central agent models all of the entities as a single “self”.

In general, the agent in a single-agent system models itself, the environment, and the interactions between the agent and environment. The agent is an independent entity with its own goal, action, and domain knowledge. In a single-agent system, other agents are not recognized by the agent. Thus, even if there are other agents in the system, they are not modeled as having a goal. That is, they are just considered part of the environment. The point being emphasized is that although agents are also a part of the environment, they are explicitly modeled as having their own goals, actions, and domain knowledge can be shown in Figure 2.1.

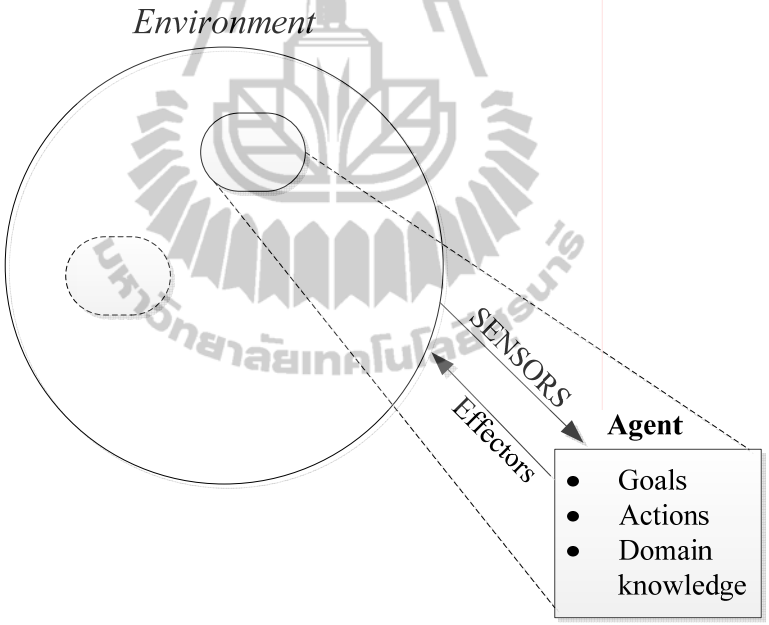


Figure 2.1 A general-agent framework. The agent models itself, the environment, and their interaction. If other agents exist, they are considered part of the environment.

2.2.2 Multi-agent systems

Multi-agent systems differ from single-agent systems in that several agents co-exist, each with their own goals and actions. From an individual agent's point of view, multi-agent systems differ from single-agent systems in that the environment's dynamics can be affected by other agents. Thus, all multi-agent systems can be viewed as having dynamic environments. Figure 2.2 depicts a multi-agent system where each agent is both part of the environment and modeled as a separate entity. There may be any number of agents, with different degrees of heterogeneity and with or without the ability to communicate directly.

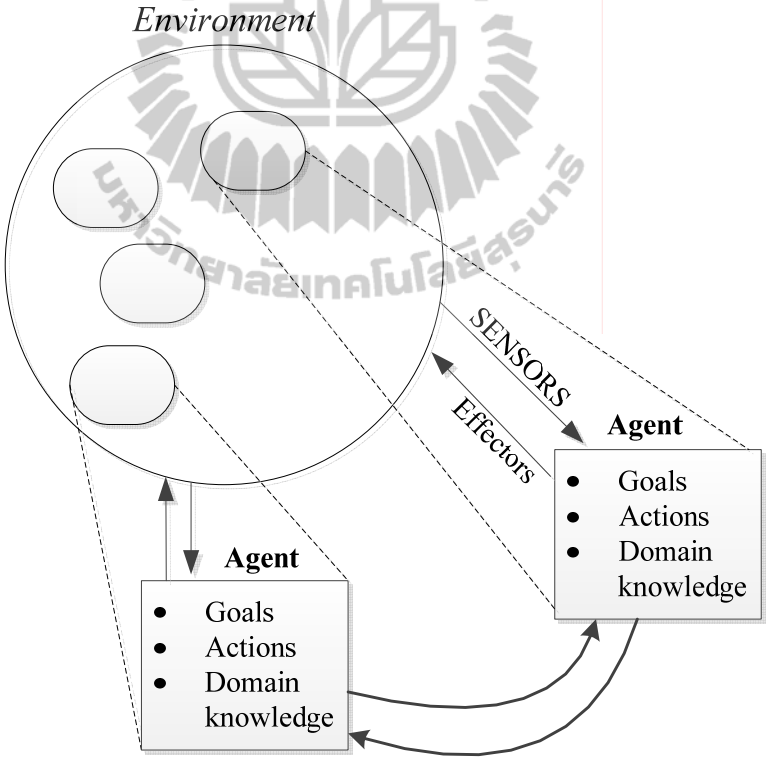


Figure 2.2 A multi-agent scenario. Each agent models each other's goals, actions, and domain knowledge. Direct interaction (communication) are indicated by the arrows between the agent.

2.3 Markov decision process theory

A Markov decision process (MDP) is a model of a decision-maker interacting sequentially with the environment. If the decision-maker sees the environment's true state, it is referred as a *completely observable MDP (COMDP)*. Otherwise, it is referred as a *partially-observable MDP (POMDP)*. In this thesis, it is assumed that the environment the multi-agent systems is a COMDP. The foundation of Markov decision process is presented as follows.

2.3.1 Markov Decision Process

A MDP is a discrete-time random decision process defined by a set of states, actions and the one-step state transition of the environment. Given any state *s* and action *a*, the probability of occurrence of each possible next state *s*' is

$$P(s' \mid s, a) = P(S_{t+1} = s' \mid S_t = s, a_t = a) \tag{2.1}$$

This equation is called the state transition probability. Similarly, given any current state and action, *s* and *a*, together with any next state, *s*', the expected value of the incurred reward is

$$R(s, a, s') = E\left[r_{t+1} \middle| S_{t+1} = s', S_t = s, a = a \right] \tag{2.2}$$

where $E[\cdot]$ is the expectation operator and r_{t+1} is the reward received at time $t+1$. Equation (2.1) and (2.2), completely specify the most important aspects of the dynamics of the MDP. A MDP model can be shown in Figure 2.3.

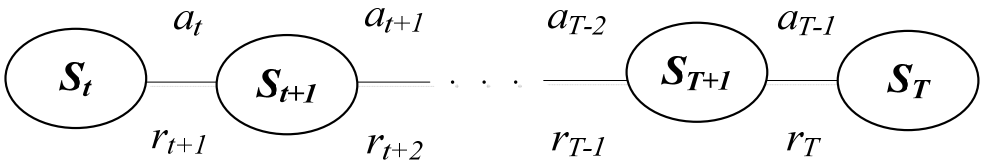


Figure 2.3 A MDP model.

A tuple (*S, A, P, R*) can describe the MDP characteristics, where *S*is a discrete set of states, *A*isa discrete set of actions available in each state, *P: S* x *A*→ *S* is a mapping from the state-action space to a probability distribution over the state space. The function of *P* is called a state transition probability matrix where each element is defined in (2.1).*R*: *S* x *A*→*R* is a mapping of the state-action space which returns the reward for taking a particular action in a given state as presented in (2.2).

The objective of solving a MDP is to find a policy, π, defined as a mapping of the state space to the action space, $\pi : S \to P[A]$, where *P[A]* is the distribution over the action space. The action-value function $Q_t^{\pi}(s,a)$ of a given policy π associates all state-action pairs (s,a) with an expected reward for performing action *a* in state *s* at time step *t* and following π thereafter;

$$\begin{aligned} Q^{\pi}(s,a) &= E^{\pi}\left[R_t \mid s_t = s, a_t = a\right] \\ &= E^{\pi}[\sum_{k=0}^{\infty} \beta r_{t+k+1} \mid s_t = s, a_t = a] \end{aligned} \tag{2.3}$$

where $R_t = r_{t+1} + \beta r_{t+2} + \beta^2 r_{t+3} + ... = \sum_{k=0}^{\infty} \beta^k r_{t+k+1}$ is the expected discounted return of the agent, β is the discount factor and $E^{\pi}[\cdot]$ is the expectation operator under policy π. The quantity Q^{π} is called the action-value function for policy π. The objective of the learning task is to find a policy π^* such that the expected value of the return is maximized, i.e. find π^* such that

$$Q^{\pi^*}(s,a) = \max_{\pi} Q^{\pi}(s,a) \forall (s,a) \in S \times A \tag{2.4}$$

In other words, the objective of MDP is to find a policy to select actions at a given state such that the long term average reward is maximized. To achieve this, particularly in scenarios where the dynamics of the environment is difficult to model (such as in WSNs), a technique called reinforcement learning can be used to solve MDPs.

2.4 Reinforcement learning

In reinforcement learning (RL), an agent(s) can learn how to map a situation to an action so as to maximize a numerical reward signal. RL is computation approach which identifies how a system in a dynamic environment can learn to choose optimal actions to achieve a particular goal. The learner is not taught which action to take, but instead must discover which action achieves the most reward by trial-and-error interactions with its environment (Sutton, R., and Barto, A., 1998).

In a RL model, the learner or decision maker is called the agent. Everything outside the agent is called environment. The interaction between a learning agent and its environment can be described in terms of states (s_t), actions (a_t) and rewards (r_t). The agent selects actions and the environment responds to those actions. Furthermore, the environment also feedback rewards to the agent, as a consequence of the action selection at a given state in terms of a reward signal which the agent tries to maximize over time. More specifically, the agent and environment interact with each other in a sequence of discrete time steps. At each time step (t), the agent detects some representation of the environment's state (s_t) and selects an action (a_t). One time step later, the agent receives a numerical reward (r_{t+1}) and finds itself in a new state (s_{t+1}). Figure 2.4 shows the agent-environment interaction in reinforcement learning.

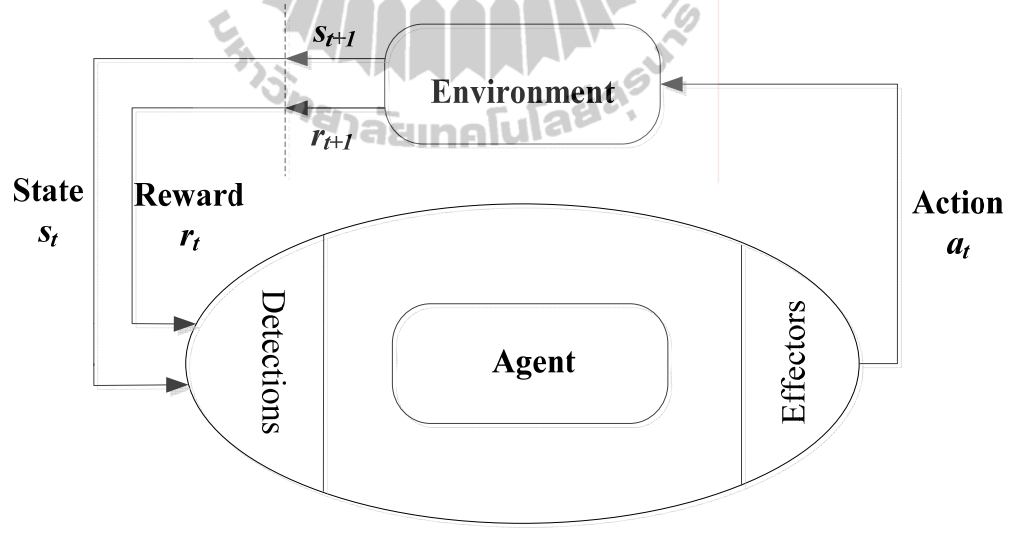


Figure 2.4 Diagram of agent-environment interaction in reinforcement learning.

2.4.1 The value function

Reinforcement learning algorithms are based on estimating value functions. A value function is the expected sum of rewards received from starting in state *s*. The value functions quantify how well the decision which the agent has taken at a given state was. Since the rewards to be received by the agent depend on the actions taken, value functions are thus defined with respect to each particular policy. Therefore, we can define the value function of a state under a policy π, $V^{\pi}(s)$ by

$$V^{\pi}(s) = E^{\pi}\{R_t \mid s_t = s\} = E^{\pi}\left\{\sum_{k=0}^{\infty} \beta^k r_{t+k+1} \mid s_t = s\right\}. \quad (2.5)$$

2.4.2 The optimal value function

The aim of solving a MDP is to find an optimal policy that achieves the maximum discounted reward over the long run. The optimal state-value function, denoted as $V^*(s)$, is therefore the state-value function at state *s* that is the maximum over all possible policies,

$$V^*(s) = \max_{\pi} V^{\pi}(s), \quad (2.6)$$

for all $s \in S$.

The optimal action-value function, denoted by $Q^*(s)$ is defined in a similar manner by

$$Q^*(s) = \max_{\pi} Q^{\pi}(s,a). \quad (2.7)$$

To solve the problem above for an optimal policy, one possible solution is through an iterative search method (Puterman, M.L., 1994) that searches for a fixed point of the following *Bellman* equation:

$$V^*(s) = \max_a \left\{ R_t + \beta \sum_{s'} P\left(s' \middle| s, a\right) V^{\pi}\left(s'\right) \right\}. \tag{2.8}$$

Equation (2.8) is a form of the Bellman optimality equation for $V^*(s)$. The Bellman optimality equation is actually a system of equations, one for each state, so if there are *N* states, then there are *N* equations in *N* unknowns. If the dynamics of environment are known, then in principle one can solve this system of equations for $V^*(s)$ using any one of variety of methods for solving systems of nonlinear equations. One can solve a related set of equations for $Q^*(s)$. The Bellman optimality equation for $Q^*(s)$ defined by

$$Q^*(s) = R_t + \beta \sum_{s'} P\left(s' \middle| s, a\right) \max_{a'} Q^*\left(s', a'\right). \tag{2.9}$$

2.4.3 Q-learning

Q-learning (Sutton, R., and Barto, A., 1998) is a RL method for MDPs which are controlled by a single agent. Q-learning is an algorithm that does not need a model of the environment and can directly approximate the optimal action-value function (Q-value) through online learning.

In Q-learning process, the agent starts with an arbitrary initial Q-value at time step 0. Upon selecting action *a* at state *s*, the agent obtains an immediate reward *r* from the environment which then transits to a new state. The agent updates

the new Q-value with the feedback from the environment. The update rule at time step *t+1* of the Q-value is given by:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha\left[r + \beta \max_{a'} Q_t(s',a')\right] \tag{2.6}$$

where $\alpha \in [0,1)$ is the learning rate. The process is repeated iteratively so that the agent can learn its own optimal policy. Note that the Q value in equation (2.10) can converge to $Q^*(s,a)$ under the assumption that all states and actions have been visited infinitely often (Sutton, R., and Barto, A., 1998).

2.5 Multiple agent Q-learning algorithm

Multi-agent systems differ from single-agent systems in that there are many different agents that learn a task and that all of the agents' actions affect the environment. Thus, the optimal policy does not rely on only one agent, but conditions on all agents. There are works which directly applied Q-learning to multi-agent systems where an individual agent maximizes its own benefit. By doing so, their works neglect the presence of the other agents. As a result, suboptimal decisions may be reached. Therefore, an individual agent should take account of the effect of joint actions as a more suitable strategy for multi-agent systems.

2.6 Distributed reinforcement learning

In recent years, several extensions to RL and Q-learning for distributed systems have been proposed. Many interesting problems which require solving with reinforcement learning (RL) also have properties that make distributed solutions desirable. In scenarios where the state and/or action space are large, a distributed approach to perform the computation is desirable because it speeds up computation. In many systems such as WSNs, access to sensors and actuators is inherently distributed, thus making a distributed solution method an attractive alternative to implementing a global high bandwidth communication network. Potential applications include control of power grids (or any other distribution of a resource such as water, gas, etc.), automobile traffic control, electronic network routing, control of robot teams, and communication networks. Figure 2.5 illustrates a distributed RL framework.

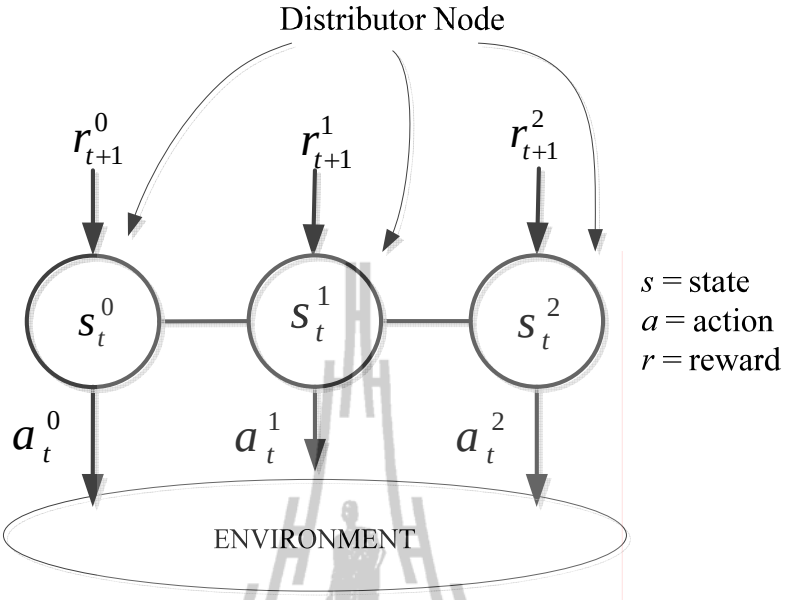


Figure 2.5 Distributed RL diagram representing logical nodes in the distributed RL formulation. Each node senses its own state of the environment, takes its own action, and receives its own reward signal.

2.6.1 Distributed value functions

In this section, we present an algorithm for distributed reinforcement learning based on distributing the representation of the value function between nodes. Each node in the system only has the ability to sense state locally, choose actions locally, and receive rewards locally. The goal of the system is to maximize the sum of the discounted rewards over all nodes and over time. However, each node is allowed to give its neighbors the current estimate of its value function for the states it transits through. A value function learning rule (described in the next section) uses information that allows each node to learn a value function that is an estimate of a weighted sum of future rewards for all the nodes in the network. With this representation, each node can choose actions to improve the performance of the overall system.

2.6.2 Distributed value function (DVF) algorithm

Usually, in MAS, agents only have local state information since the global state of the system is not fully observable from each agent's point of view. Hence, Schneider, J., et al. (1999) proposed a Q-learning based algorithm for the DVF algorithm. This approach allowed each node to compute its local value function based only on available local information. Hence, agents only need to transmit the estimated value of the current state they land in, i.e. $V^i(s_t^i)$ for agent i at time t at each iteration. The update rule at time step t for agent i is given by

$$Q_{t+1}^i\left(s_t^i, a_t^i\right) = (1-\alpha)Q_t^i\left(s_t^i, a_t^i\right) + \alpha\left[r_{t+1}^i\left(s_{t+1}^i\right) + \gamma \sum_{j \in Neigh(i)} f^i(j) V_t^j\left(s_{t+1}^j\right)\right] \quad (2.11)$$

$$V_{t+1}^i(s_t^i) = \max_{a \in A^i} Q_{t+1}^i\left(s_t^i, a\right), \quad (2.12)$$

where α is the learning rate, $f^i(j)$ are factors that weigh the value functions of the neighbors of agent i such that

$$f^i(j) = \begin{cases} \dfrac{1}{|Neigh(i)|} & \text{, if } Neigh(i) \neq 0 \\ 1 & \text{, otherwise,} \end{cases} \quad (2.13)$$

where $j \in Neigh(i)$ is the set of neighbors of node i .

2.7 Summary

In this chapter, an overview of the multi-agent Q-learning algorithm called Distributed Value Function has been given. This algorithm was used to determine maximum coverage control in WSNs in this thesis. The reason for selecting this method was that of the algorithm allowed the agent to rationally determine the near-optimal policy and receive maximum coverage and maximum trade-off between achieved coverage with energy consumption in WSNs.

However, the DVF algorithm was not designed to handle malicious nodes which may be present in the system. In the case when nodes are under sleep deprivation and snooze attacks, the performance of the WSN will be affected by an energy consumption increase and average reward reduction. This suggests that the DVF algorithm alone strongly relies on the cooperation between nodes. To deal with malicious node behaviors and vulnerable attacks, we proposed to solve this problem by using a topology maintenance protocol in order to obtain a secure and near-optimal coverage allocation strategy under energy constraints.

In the next chapter, asecure multi-agent coverage control scheme for wireless sensor networks with malicious nodes is presented.

CHAPTER III

A SECURE MULTI-AGENT COVERAGE CONTROL SCHEME FOR WIRELESS SENSOR NETWORKS WITH MALICIOUS NODES

3.1 Introduction

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous device using sensors that can communicate with each other to perform sensing and at processing cooperatively. The overall objective of a WSN is to provide a low-cost solution to gather physical data from the environment, such as noise, pressure, light, observation and transmit it to a base station.

Due to scarce battery supply, topology maintenance and coverage control has become a challenging issue in WSNs. Works in (Chen, B., et al., 2002); (Cerpa, A., and Estrin, D., 2004); (Ye, F., et al., 2003); (Wang, X., et al., 2003) aimed at increasing the lifetime of the network by keeping only a subset of sensing nodes active and turning off the remaining redundant nodes. While Chen, B., et al. (2002); Cerpa, A., and Estrin, D. (2004) attempted to maintain connectivity but not guarantee sensing coverage, Ye, F., et al. (2003); Wang, X., et al. (2005) addressed both network connectivity and coverage requirement.

Distributed self-adaptive coverage control schemes are attractive as WSNs are typically spatially-distributed and deployed in dynamically changing environments which may be difficult to access and manually reconfigure. Such autonomous coverage control can be achieved by multi-agent systems (MAS) (Tham, C.K., and Renaud, J.C., 2005). Such distributed approach is also more scalable and compatible with resource-constrained sensor nodes. One of such system called the DVF algorithm has been investigated in (Tham, C.K., and Renaud, J.C., 2005) where all sensor nodes act as agents that cooperate to achieve a common goal of maximum coverage and minimum energy consumption.

However, all of the aforementioned works were designed for trusted and cooperative environments. With scarce onboard resources, it is possible that sensor nodes may act selfishly by declining to service other nodes (Vaz de Melo, P.O.S., et al., 2008); (Singsanga, S., et al., 2010). Sensor nodes may also encounter attacks by other malicious nodes inside or outside of the network. These attacks may be used to reduce the lifetime of the sensor network, or to degrade the functionality of the sensor application by reducing the network connectivity and the sensing coverage that can be achieved. We study *three* types of attacks that can be launched in WSNs: *sleep deprivation* are attacks which the adversary tries to induce a node in a specific area to remain active thereby wasting energy and reduce the sensor network lifetime; *snooze attack* which the adversary forces the nodes to remain in the sleeping state thereby reducing sensing coverage or network connectivity; and *network substitution attack* which an adversary controls some nodes which were elected to maintain the connectivity. Once the adversary takes control of a portion of the network, it can carry out other attacks such as sending false information to other nodes. To the best of our

knowledge, only Gabrielli, A., et al. (2011) presented countermeasures against these types of security attacks in topology maintenance and in WSNs. However, Gabrielli, A., et al. (2011) only aimed at maintaining coverage by using a subset of nodes in an active or awake state, their objective was not to maximize coverage area per unit energy consumed.

This thesis therefore proposed a coverage control scheme which aimed at maximizing the coverage per unit energy consumption and was designed to operate in an adversarial malicious environment. In particular, we proposed to integrate the DVF algorithm which is an adaptive and distributed multi-agent coverage control scheme (Tham, C.K., and Renaud, J.C., 2005) with a secure topology maintenance protocol (TMP) (Gabrielli, A., et al., 2011) against malicious node attacks in WSNs. More specifically, we incorporated a TMP countermeasure i.e. *probing* to verify the local states and active nodes within a neigh boring area before increasing or reducing its coverage to allow tolerance against attacks from multiple nodes within a node's transmission range. Our contribution centers on the integration of the probing mechanism to the DVF scheme and its performance evaluation against sleep deprivation, snooze and network substitution attacks.

3.2 Multi-agent coverage control

A multi-agent coverage control scheme called the Distributed Value Function (DVF) has been a common approach to coordinately and cooperatively improve the coverage control performance in wireless sensor networks (Tham, C.K., and Renaud, J.C., 2005); (Seah, M.W.M., et al., 2007); (Renaud, J., and Tham, C.K., 2006). In this method, each node communicates and exchanges information about its value function.

A value function is a function that quantifies how well the agent at a node performs at a given state $s \in S$ where S is a discrete set of all possible states of the sensor network. Let $a \in A$ be the action selected by an agent, where A is the discrete set of all possible actions available at each state. The rule, so called policy π, is defined as a rule which the agent selects actions as a function of states. In other words, it is the mapping from a state $s \in S$ and action $a \in A$ to the probability of selecting action a at state s. The *value function* of state s under a given policy π is formally defined by $V^{\pi}(s) = E^{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s\right\}$, where r_{t+1} is the reward of taking a particular action in a given state s at time t, γ is the discount factor and $E^{\pi}\{\ \}$ is the expectation operator. Similarly, we define the *action value function* of taking action a at a given state under policy π by $Q^{\pi}(s,a) = E^{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a\right\}$. The objective is to find a policy π^* such that $\pi^* = \arg\max_{\forall \pi} Q^{\pi}(s,a)$. To achieve this objective, each agent i (node) in the DVF algorithm performs an update of its own action value function. The update rule at time step t for agent i is given by (Tham, C.K., and Renaud, J.C., 2005):

$$Q_{t+1}^{i}(s_t^i, a_t^i) = (1-\alpha) Q_t^i(s_t^i, a) + \alpha\left(r_{t+1}^i(s_{t+1}^i) + \gamma \sum_{j \in Neigh(i)} f^i(j) V_t^j(s_{t+1}^i)\right) \quad (3.1)$$

$$V_{t+1}^{i}(s_t^i) = \max_{a \in A^i} Q_{t+1}^i(s_t^i, a), \quad (3.2)$$

where α is the learning rate, $f^i(j)$ are factors that weigh the value functions of the neighbors of agent *i* such that:

$$f^i(j) = \begin{cases} \dfrac{1}{|Neigh(i)|} & \text{, if } Neigh(i) \neq 0 \\ 1 & \text{, otherwise,} \end{cases} \tag{3.3}$$

where $j \in Neigh(i)$ is the set of neighbors of node *i* (Tham, C.K., and Renaud, J.C., 2005). Hence, in the DVF algorithm, nodes cooperate not only with their direct neighbors but with all the nodes since the value function captures information about other nodes which are not direct neighbors as well. Therefore, the DVF algorithm is strongly dependent on cooperation from other nodes in the network through the last summation term on the right hand side of equation (3.1). The information exchange (i.e. the value functions of other nodes) in the DVF algorithm is vulnerable to malicious nodes attacks; as such information may be falsely exchanged by a compromised node. This has motivated us to improve the resilience of the DVF algorithm to malicious nodes.

3.3 Malicious node environment

So far the DVF algorithm has been studied under the assumption that all nodes in the WSN are cooperative (Tham, C.K., and Renaud, J.C., 2005). Hence, like other topology maintenance and coverage control schemes assuming this condition, DVF is vulnerable to security attacks where malicious nodes send spoofed or false messages to defeat the objective of the algorithm. This section describes the types of attacks that could occur in a WSN (Gabrielli, A., et al., 2011). These attacks could

potentially be used to reduce the lifetime of the sensor network, or reduce the achievable network connectivity and sensing coverage.

3.3.1 Sleep deprivation attack

In this type of attack, the adversary tries to induce a node in a specific area to remain active. This attack has two effects. First, by increasing the energy expenditure of sensor nodes, it reduces the estimated lifetime of the network. Second, in the case of a densely populated area, it can lead to increased energy consumption due to congestion and contention at the data link layer.

3.3.2 Snooze attack

In this type of attack, the adversary forces the nodes to remain in the sleeping state. This kind of attack can be applied to the whole network or to a subset of nodes. In the latter case, the adversary can launch an attack to jeopardize the connectivity of the network or to reduce the sensing coverage in a region. For example, an adversary can selectively turn off nodes that are monitoring an intruder's path through an area in which a sensor field has been deployed for surveillance.

3.3.3 Network substitution attack

In this type of attack, the adversary deploys some nodes, which are in a set elected by the TMP, to gain control of part of or the entire network. Once these nodes are under control, the adversary can carry out other attacks such as sharing false or inaccurate information or readings with other nodes. This type of attack is difficult to detect since the compromised node can still maintain connectivity and appear as it were operating as normal.

3.4 Secure multi-agent coverage control: Part 1

In order to make the DVF coverage control scheme more robust to malicious node attacks, TMP probing procedures were integrated into the DVF framework. The probing mechanism verifies whether there are active or inactive nodes in a node's transmission range before decision take any action (i.e. changing the size of coverage area). In particular, in our proposed algorithm, a node can tolerate attacks by up to t nodes within its transmission range.

To study the coverage control performance of a WSN under attack, a lighting control application of a room represented by a 10 x 10 grid was studied as shown in Figure 3.1. This room contained a group of 5 agents deployed on 5 nodes with light sensing capabilities, labeled Ml to M5. Each of them had a light source that can illuminate the part of the room surrounding the agent. The objective was for the agents to learn to cooperate with one another, in presence of malicious nodes, in order to completely illuminate the room in an energy-efficient way, i.e. minimize the number of lights turned on. The area of agent i, denoted as a^i, refers to the 5 x 5 grid square centered on agent i.

We defined three modes for each sensor node, i.e., sleep mode, work mode and probe mode. In sleep mode, a node becomes inactive. In probe mode, a node has just awoken from the sleep mode but is checking on other nodes in its transmission range whether they are active or not, prior to taking any decision. In work mode, nodes are active and can decide to become inactive (since their cells may already be covered by other nodes) or to use low or high coverage. On the other hand, in the original DVF algorithm, a node collects and exchanges data with its neighboring nodes, and immediately decides whether to be in an active (awake) or to be inactive

(asleep) state. In our proposed integrated DVF and TMP algorithm, before a node enters work mode, it enters probe mode to check whether or not there exists other nodes within its transmission range already in work mode or not.

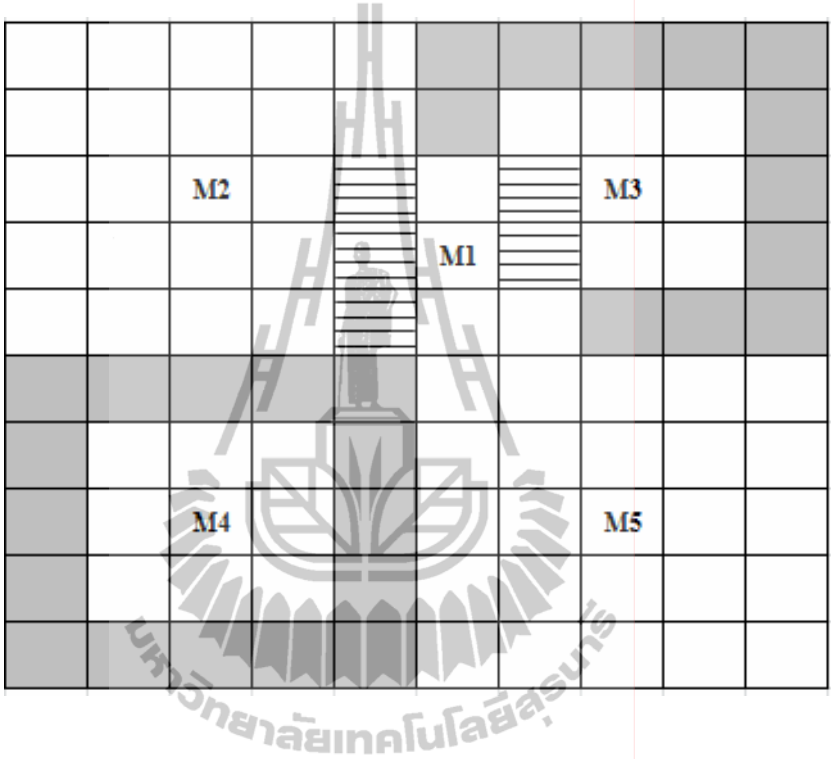


Figure 3.1 A 10 x 10 grid room representation. Grey cells are not illuminated, white cells are illuminated by one node and striped cells are illuminated by two nodes.

Local agent state: Each agent i can sense the level of light in its area. Its local state s^i is state of each agent based on its mode and coverage. The state of mode consists of three modes i.e. sleep mode, work mode and probing mode. In sleep mode, there is 1 possible state (i.e. no lit cells). In probing and work modes, each has 26 possible states. Therefore, there are 53 possible states (1+26+26) for the system given by:

$s^i = (state\ of\ mode, state\ of\ coverage) = (s_m^i, s_c^i),$

where $s_m^i \in \{sleep\ mode, probe\ mode, work\ mode\}$ and $s_c^i \in \{0,...25\}$.

Local agent actions: Each agent i has the ability to take one of the following three actions in any state it lands in. The action space A^i is the set of all possible actions for each state A^i = {*Action 0* (Turn off the light), *Action 1* (Turn on the light in LOW coverage. This illuminates 9 cells around the agent, as shown by Ml, M3 and M5 in Fig.1), *Action 2* (Turn on the light in HIGH coverage. This illuminates the 25 cells around the agent, as shown by M2 and M4 in Figure 3.1), and *Action 3* (Send probe to neighbors and wait for a response within in a finite time. This action allows the agent to sense the illuminated cells within its range prior to deciding to take *Action 0, Action 1 or Action 2*)}. Once an action is taken, the current local state of the agent transits to a new local state accordingly as shown in Figure 3.2.

Note that in the distributed learning schemes such as the DVF algorithm, the agents use only information that is locally available to make their decisions. The reward for agent i, denoted as $r^i\left(s_t^i\right)$ is a function of agent i's state s^i at time t and is defined by:

$$r^i(s_t^i) = G^i\left(s_t^i\right) - C^i, \tag{3.4}$$

where $G^i(s_t^i)$ is a function of the number of cells illuminated in the area of agent i such that

$$G^i(s_t^i) = nb\_cell\_bright\left(a^i\right) \times GAIN\_CELL\_BRIGHT, \tag{3.5}$$

and C^i is the energy consumption resulting from the action taken by agent *i* at time *t* such that

$$C^i = \begin{cases} 0 & \text{, if Action 0 was taken} \\ COST\_LOW & \text{, if Action 1 was taken} \\ COST\_HIGH & \text{, if Action 2 was taken} \\ COST\_PROBE & \text{, if Action 3 was taken.} \end{cases} \tag{3.6}$$

The reward functions and state transitions for the proposed DVF+TMP algorithm are shown in Figure 3.2. Note that when the agent decides to take Action0, a reward *G* is obtained since the local cells could still be lit by neighboring nodes and the cost is zero since no energy is consumed if the agent becomes inactive. For other actions, the agent is rewarded with G subtracted by a non-zero cost in (3.6).

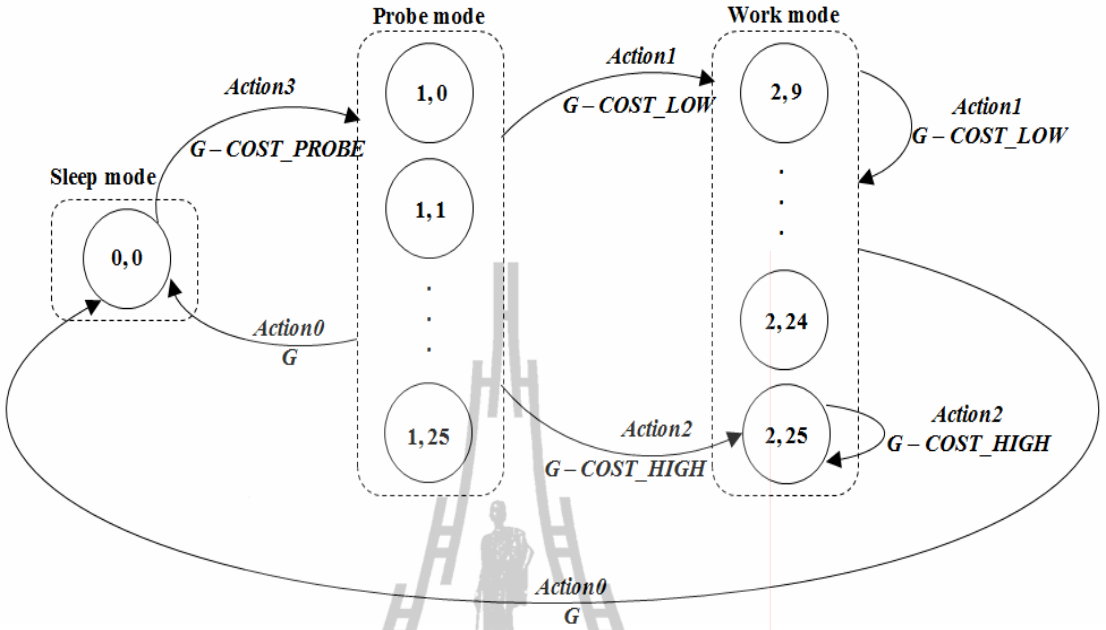


Figure 3.2 State transition diagram of the probing mechanism in the DVF+TMP algorithm.

The probe mode was designed in such a way that a node makes its state-transition decisions, e.g. a decision whether to sleep or remain active, based on inputs from multiple neighboring nodes in order to be resilient to false message created by malicious nodes. Probing was also performed each instant a node is awaken from an inactive state. For Without probing a node's eligibility to be in sleeping or active state, an adversary can launch a resource consumption attack that results in a node staying in a active state until its energy is depleted. Four performance metrics were considered: (1) the average reward per time step defined by:

$$average\ reward = \frac{\sum_{t=0}^{T} r^i\left(s_t^i\right)}{T}, \tag{3.7}$$

where T is the total number of time steps, and $r^i\left(s_t^i\right)$ is given by (3.4); (2) the average energy consumption; (3) the percentage of coverage area from good (uncompromised) nodes defined as the total number of cells illuminated by all good nodes at a time step divided by the total number of cells in the system; and (4) the trade-off which is defined by:

$$Trade\text{-}off = \frac{\sum_{t=0}^{T}\sum_{\forall i} number\ of\ lit\ cells\ by\ agent_i(t)}{\sum_{t=0}^{T}\sum_{\forall i} energy\ consumption\ by\ agent_i(t)} \tag{3.8}$$

Note that the trade-off in the above equation reflects the total number of illuminated cells throughout the simulation over the total amount of energy consumption. The trade-off represents the number of illuminated cells (coverage) per unit energy consumed. It is expected that the better the system can deal with malicious nodes, the better (and more efficient) the uncompromised sensor nodes can decide and therefore the higher the trade-off.

In the simulation, we used *GAIN_CELL_BRIGHT* = 0.5, *COST_LOW* = 0.8, *COST_HIGH* = 3, *COST_PROBE* = 0.5 the learning rate α = 0.4 and the discount factor γ = 0.7. The values of the learning rate and discount factor were obtained from experimenting a range of values and selecting the parameters which received the best performance in terms of average reward per time step. The run length of each simulation was T = 20,000 time steps and the results were averaged over 10 runs to achieve the desired accuracy.

3.5 Results and analysis: Part 1

3.5.1 Sleep deprivation and snooze attack

We assigned each agent to encounter sleep deprivation attack and snooze attack and then analyze the results in Figures 3-6. Note also the percentages of coverage in the normal situation (no attack) are also shown in Figures 3 and 5 (represented by DVF and DVF+TMP). Denote “M1-w”, “M2-w”, “M3-w”, “M4-w”, “M5-w” for cases when agent 1, 2, 3, 4 and 5 were each attacked by sleep deprivation, respectively. Similarly, denote “M1-s”, “M2-s”, “M3-s”, “M4-s”, “M5-s” for cases where agent 1, 2, 3, 4 and 5 were each attacked by snooze attack, respectively. As a benchmark to compare the effects from malicious nodes, i.e., sleep deprivation and snooze attacks, we observe how each agent operates in the normal situation. We compared our proposed integrated DVF and TMP algorithm (abbreviated by DVF+TMP) with the original DVF algorithm (abbreviated by DVF).

Figure 3.3 depicts the sleep deprivation effect on the percent coverage of the DVF algorithm compared with the proposed DVF+TMP algorithm. In case of sleep deprivation attack on agents 2, 3, 4, and 5, we can see that the percentage of coverage for each case differs only slightly. Furthermore, for each case, convergence to a policy which obtained the most coverage was achieved. However, in the case when agent 1 was attacked by sleep deprivation, the original DVF algorithm attained zero coverage. Note that agent 1 was located in the center of the area and its coverage overlapped those of agent 2, 3, 4, and 5 (see Figure 3.1). Such result showed that when agent 1 was under sleep deprivation attack, all the other good (uncompromised) agents in the system were falsely led to converge to sleep mode thereby attaining zero good node coverage with the original DVF scheme. On the other hand, when agent 1 was

attacked by sleep deprivation, the proposed DVF+TMP algorithm can attain 75% percentage of coverage. Figure 3.4 shows the trade-off results. In the cases of sleep deprivation attack on agent 2, 3, 4, 5 trade-off values gradually increased to the maximum tradeoff that could be achieved, thereby agreeing with the percentage of coverage results. Note that in the case when agent 1 was attacked, we can see that the average coverage per energy consumption unit of DVF+TMP was significantly better than the original DVF algorithm.

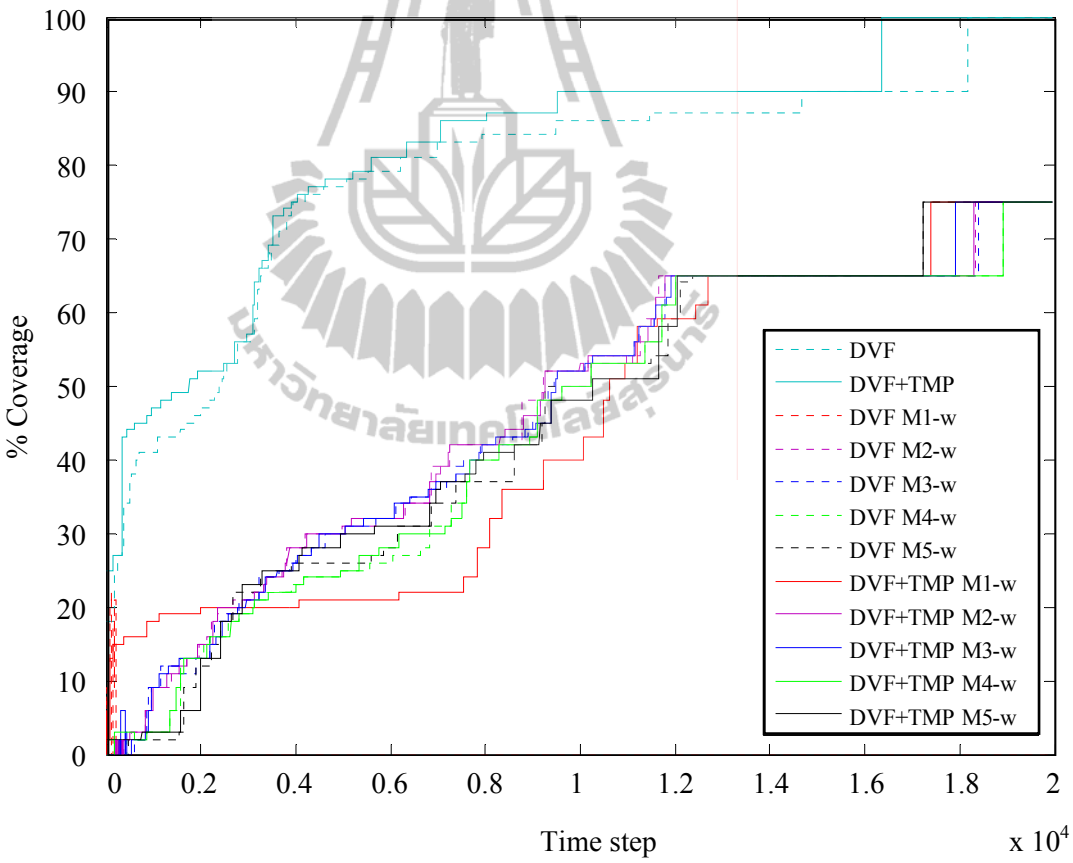


Figure 3.3 Sleep deprivation effect on the percentage of coverage.

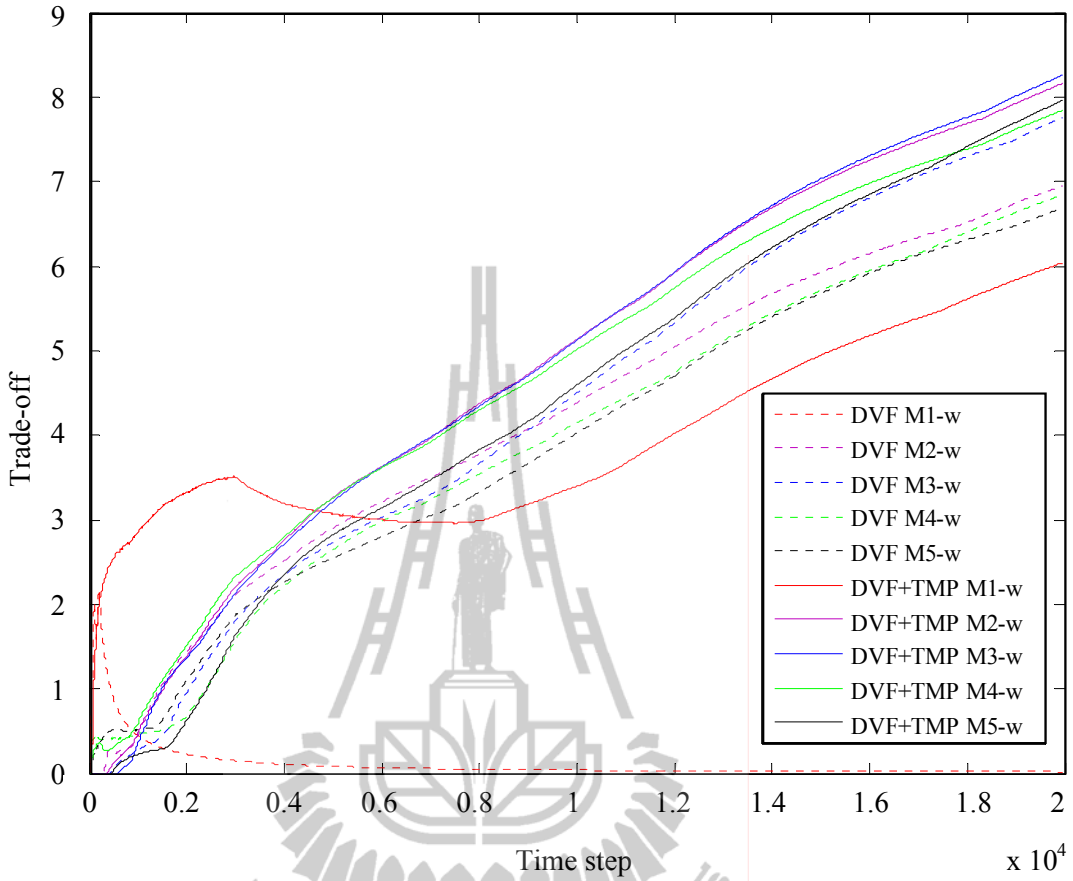


Figure 3.4 Sleep deprivation effect on the trade-off.

Figures 3.5, 3.6 illustrate the snooze attack effect on the percentage of coverage and the trade-off in both algorithms. In Figure 3.5, when agent 2, 3, 4, 5 were attacked by snooze attack, the DVF algorithm learned slower than the DVF+TMP algorithm. Furthermore, the final coverage results of the DVF+TMP algorithm obtained were up to 10% more than those of the DVF algorithm (i.e. for agent M4). Note that when agent 1 was under snooze attack, both algorithms eventually attained 100% coverage. This was because all agents must work in HIGH mode when agent 1 was attacked, in which case was the optimal policy for the system. When considering the trade-off in Figure 3.6, we can see that all agents

depicted similar patterns though the DVF+TMP algorithm consistently gave a better trade-off (i.e. more cells illuminated per unit energy) than the DVF algorithm.

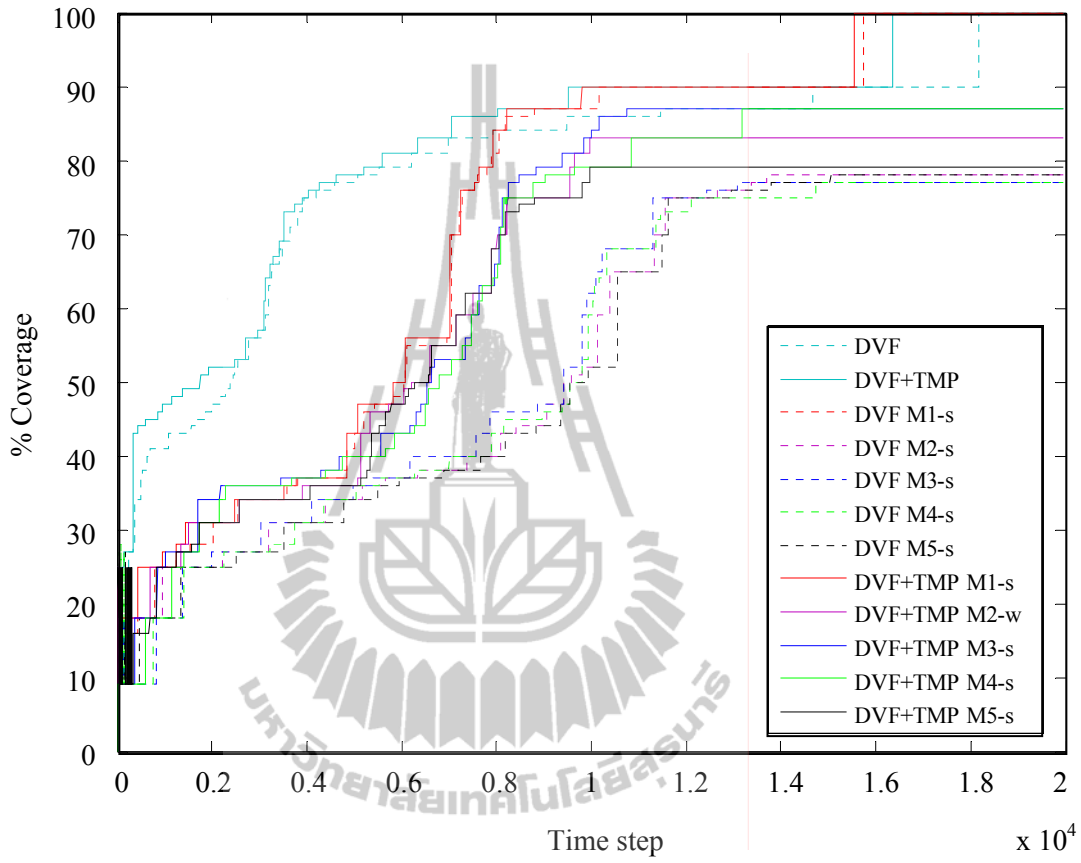


Figure 3.5 Snooze effect on the percentage of coverage.

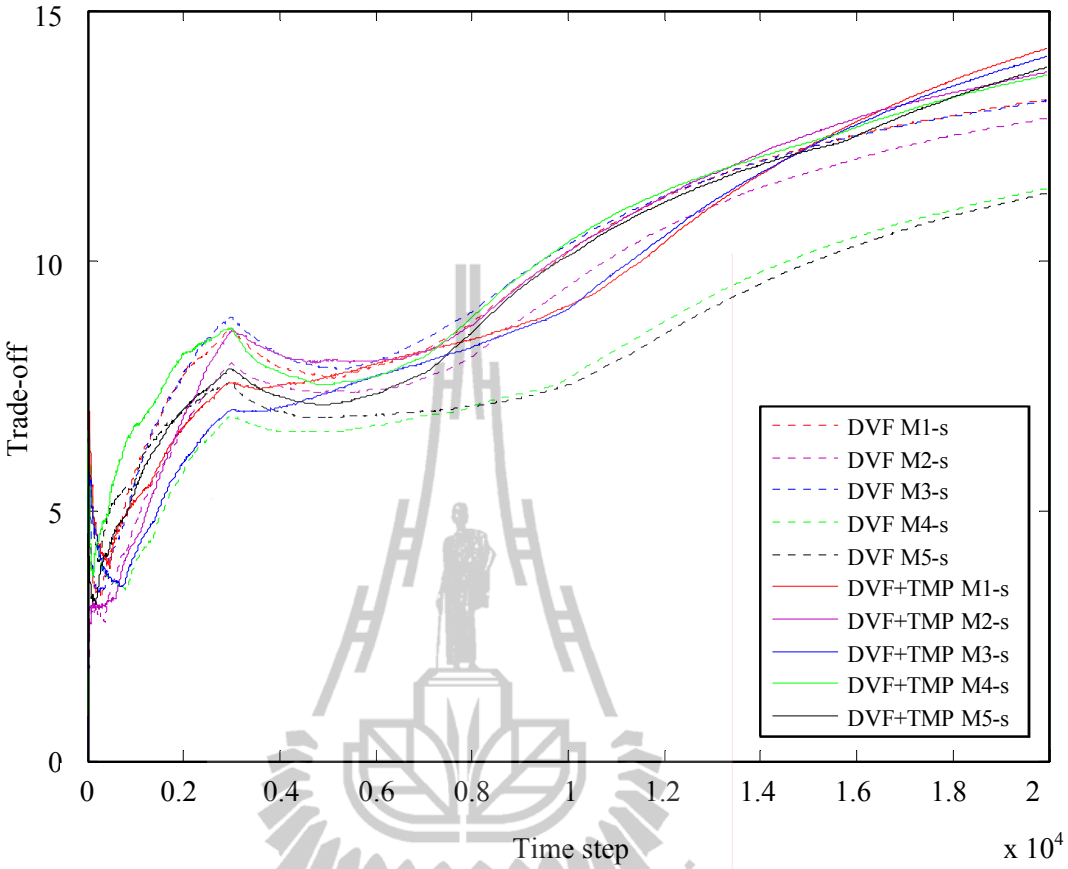


Figure 3.6 Snooze effect on the trade-off.

Table 3.1 Randomly generated sleep deprivation attack results.

No. of attacked nodes	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward/ Avg energy consumed	Avg reward	Avg energy consumed	Avg reward/ Avg energy consumed
0	6.19	1.18	5.23	6.59	1.21	5.45
1	8.57	1.6	5.35	8.98	1.64	5.47
2	9.33	1.8	5.18	9.65	1.8	5.36
3	9.85	1.95	5.04	10.35	2.04	5.07
4	10.21	2.09	4.88	11.13	2.12	5.26

Table 3.2 Randomly generated snooze attack results.

No. of attacked nodes	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward/ Avg energy consumed	Avg reward	Avg energy consumed	Avg reward/ Avg energy consumed
0	6.19	1.18	5.23	6.59	1.21	5.45
1	3.03	0.75	4.06	5.39	0.88	6.15
2	1.87	0.62	3.03	4.39	0.8	5.5
3	1.26	0.49	2.58	3.34	0.55	6.06
4	1.03	0.44	2.36	2.98	0.54	5.49

So far the nodes under attack have been predetermined. In the next experiment, we evaluated the performance of the DVF and DVF+TMP algorithms when the malicious nodes were randomly generated. Tables 3.1 and 3.2 show the results as the number of malicious nodes were increased from 0 (normal situation) to 4 (worst case scenario). Table 3.1 shows results from DVF and DVF+TMP algorithms under the sleep deprivation attack. Although the average rewards were increased as a result of the attack, the average energy consumption was high. However, the DVF+TMP algorithm achieved higher average reward per unit energy consumed than

the DVF algorithm alone. On the contrary, Table 3.2 shows that as the number of nodes under snooze attack increased, the average reward dropped significantly along with the energy consumption. Once again, our algorithm attained more average reward than the DVF alone. Furthermore, under this attack, the average reward per unit energy consumed by the DVF+TMP scheme was also higher than the DVF algorithm. The results in Tables 3.1 and 3.2 agreed with the trade-off in Figures 3.3 and 3.5, indicating that our method can achieve more coverage per unit energy consumed than the DVF alone.

3.5.2 Network substitution attacks

In this subsection, we study the performance of the two algorithms in presence of network substitution attacks. Under this type of attack, the adversary takes control of a node in the system. The compromised node can still maintain connectivity and appear to operate normally. However, since it is completely controlled by the adversary, it can carry out other types of attacks such as selective or complete packet dropping, traffic analysis, send false or inaccurate readings or information. We assume that the compromised node exchanges inaccurate information with other agents. We assume that the degree of inaccuracy is inserted by multiplying the value function in the last summation term in (3.1) by a parameter ξ randomly chosen from the interval $[\ 1-\xi_{max},\ 1+\xi_{max}]$ where ξ_{max} = 0.25, 0.5, 0.75 and 1. Each agent encountered such attack and the results obtained were averaged over all agents. Figure 3.7 depicts the percentage of achievable coverage obtained from different degrees of inaccurate value functions on each algorithm under normal situation (with no sleep deprivation or snooze attacks), sleep deprivation and snooze attacks, respectively. The higher the degree of inaccuracy, the less the coverage achieved. This confirms our motivation

that distributed coverage control schemes rely on node cooperation and thereby are vulnerable to malicious node attacks. Furthermore, for each scenario, the DVF+TMP algorithm consistently outperforms the DVF algorithm alone by gaining up to 12%, 25% and 8% more coverage in the normal situation (with no other attacks), sleep deprivation and snooze attacks, respectively.

The effects of inaccurate value functions on the average reward, average energy consumption, and the ratio of the two parameters are shown in Tables 3.3, 3.4 and 3.5 for the normal situation (with no sleep deprivation or snooze attacks), sleep deprivation and snooze attacks, respectively. From the three tables it can be seen that as the degree of inaccuracy increases, the average reward from both algorithms decreased accordingly. However, the DVF+TMP algorithm consumed less average energy than the DVF algorithm alone therefore achieved higher efficiency in terms of average reward per unit energy consumed for all cases.

All of these results suggested that the DVF alone relies strongly on the cooperation among agents and is vulnerable to security attacks. The proposed DVF+TMP scheme can enhance the security and can cope with sleep deprivation, snooze and network substitution attacks, thereby improving the resilience of the distributed coverage control scheme.

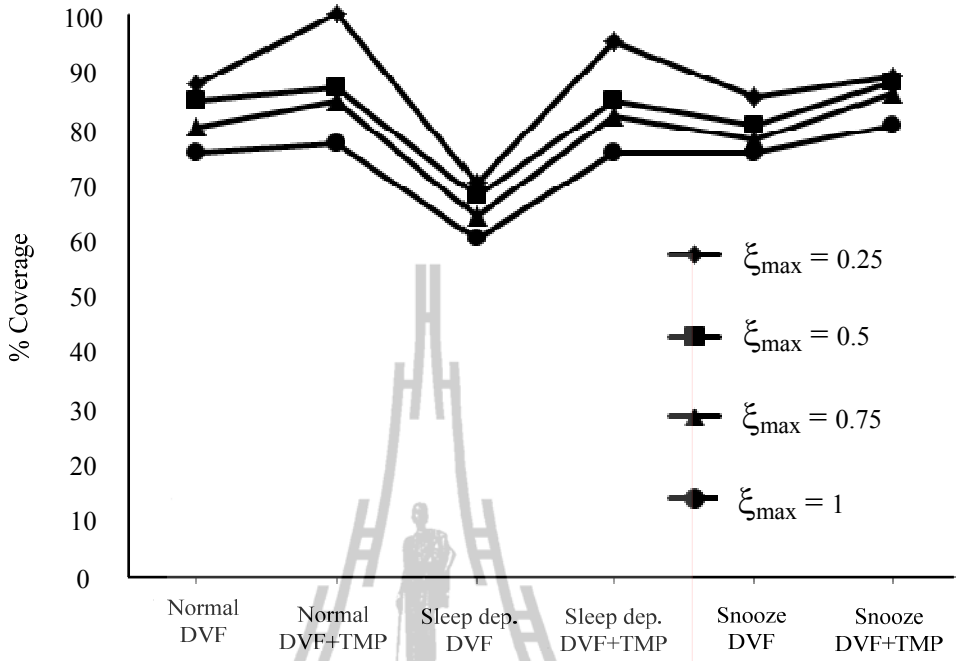


Figure 3.7 Effect of inaccuracy on the percentage of coverage.

Table 3.3 Effect of inaccuracy in normal scenario.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0	6.19	1.18	5.23	6.59	1.21	5.45
0.25	7.14	1.54	4.63	7.9	1.55	5.11
0.5	7.11	1.56	4.57	7.84	1.54	5.11
0.75	7.08	1.55	4.58	7.76	1.53	5.09
1	7.02	1.55	4.55	7.67	1.51	5.07

Table 3.4 Effect of inaccuracy in sleep deprivation scenario.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0.25	7.2	1.56	4.63	7.91	1.52	5.2
0.5	7.1	1.55	4.6	7.58	1.49	5.09
0.75	6.9	1.54	4.49	7.21	1.5	4.81
1	6.7	1.53	4.37	6.87	1.49	4.61

Table 3.5 Effect of inaccuracy in snooze scenario.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0.25	6.89	1.58	4.36	7.76	1.57	4.95
0.5	6.86	1.57	4.32	7.57	1.56	4.86
0.75	6.77	1.56	4.3	7.28	1.52	4.8
1	6.63	1.54	4.29	7.01	1.51	4.65

3.5.3 Summary: Part 1

In this thesis, we proposed the DVF+TMP coverage control scheme based on the integration of a distributed learning scheme for multi-agent systems called the DVF algorithm and a secure topology maintenance protocol (TMP) to countermeasure sleep deprivation, snooze and network substitution attacks in WSNs. To evaluate its performance, a lighting control application was studied. The results showed that in the presence of malicious nodes in the system, the original DVF algorithm was directly affected suggesting that the DVF algorithm alone strongly relies on the cooperation between nodes. However, results showed that the proposed

DVF+TMP algorithm was more resilient to malicious node attacks by achieving up to 75% and 10% of coverage more than the DVF algorithm alone under sleep deprivation and snooze attack, respectively. The proposed algorithm also attained a better trade-off in terms of the number of cells illuminated per unit energy consumed. Similar results were achieved when the presence of malicious nodes in the system were increased. Furthermore, in the network substitution attack where various degrees of inaccurate information of value functions were exchanged, the DVF+TMP algorithm gained up to 12%, 25% and 8% of coverage than the DVF algorithm alone for the normal, sleep deprivation and snooze attack cases, respectively. The proposed algorithm also consistently achieved more average reward per unit energy consumed than the DVF algorithm.

3.6 Secure multi-agent coverage control: Part 2

To ensure that performance obtained in part 1 did not owe to a fixed topology of agents or the placement of agents. In this section, we extended to a lighting control application of a room represented by a 30 x 30 grid and with 40 agents placed randomly on the grid was studied as shown in Figure 3.8. This room contained a group of 40 agents deployed on 40 nodes with light sensing capabilities, labeled Ml to M40. The agents, action, and state space description remain the same as in section 3.4.

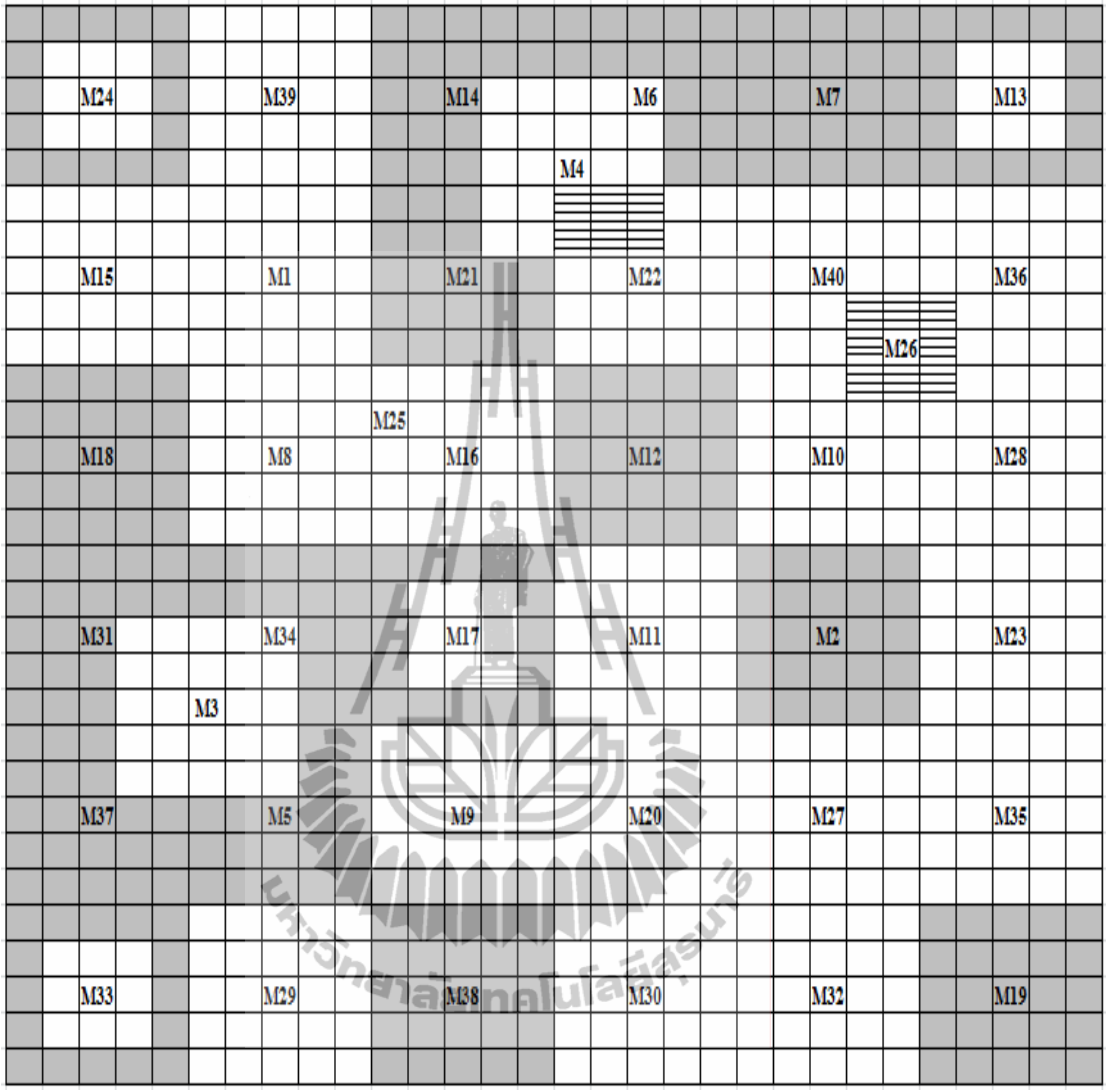


Figure 3.8 An example 30 x 30 grid room representation. Grey cells are not illuminated, white cells are illuminated by one node and striped cells are illuminated by two nodes.

3.7 Results and analysis: Part 2

3.7.1 Sleep deprivation and snooze attack

We randomly assigned 5, 10 and 15 nodes to encounter sleep deprivation attacks and snooze attacks and then analyze the results. Denote "Sleep dep. *n* nodes" ("Snooze *n* nodes") for the case when the system was under attacked by *n* sleep deprived (snooze) nodes. As a benchmark to compare the effects from these malicious nodes, we observe how each agent operates in the normal situation (without any attacks). We compared our proposed integrated DVF and TMP algorithm (abbreviated by DVF+TMP) with the original DVF algorithm (abbreviated by DVF).

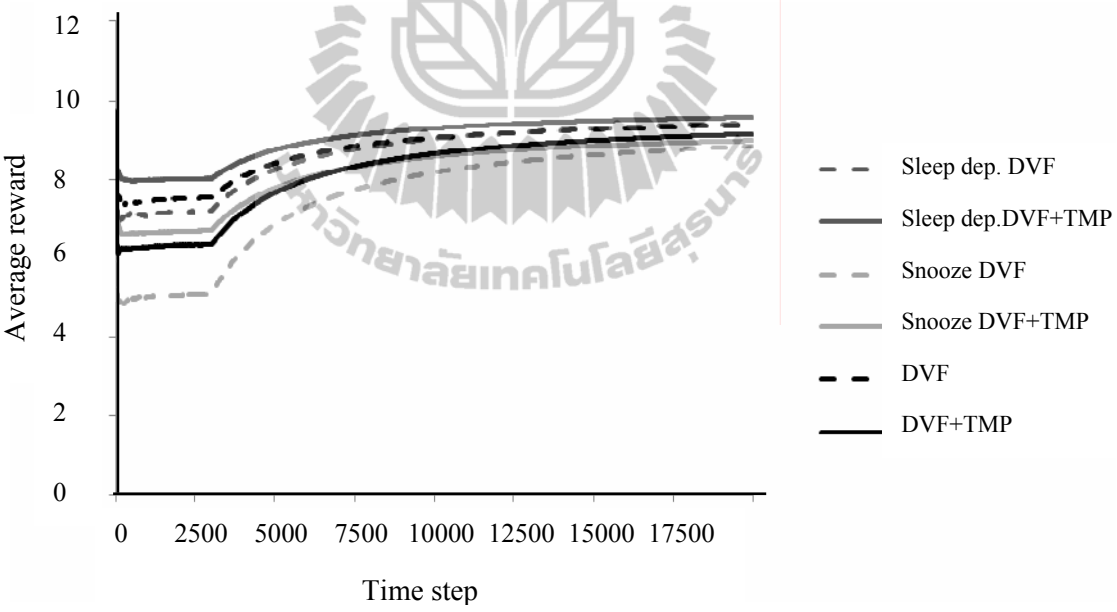


Figure 3.9 Average rewards for sleep deprivation and snooze attacks from 15 malicious nodes in the normal situation.

Figure 3.9 depicts the learning progress in terms of the average reward at each time step for the DVF algorithm and the DVF+TMP algorithm under normal situation, and the cases when 15 nodes were attacked by each type of attack. As time progressed, the agents in both algorithms were able to learn to take better decisions as depicted by the gradually increasing average reward for all situations. However, our algorithm consistently outperformed the original DVF algorithm.

The effect of the number of malicious nodes on the coverage is shown in Figure 3.10 for the sleep deprivation. In the normal situation where no nodes were attacked, the percentage of coverage achieved by the DVF and DVF+TMP algorithm were 98%, 100% respectively. When the network was attacked by 5, 10, and 15 sleep deprivation nodes, the percentage of coverage of the original DVF algorithm reduced to 76, 64 and 52%, respectively. On the other hand, the DVF+TMP outperformed the DVF algorithm by attaining 88%, 70%, and 59% of coverage. Although it would first seem reasonable to expect the coverage to increase since sleep deprived nodes were forced to operate under this attack, it should be noted that only the coverage obtained from good nodes were considered here. DVF+TMP achieved greater coverage from uncompromised nodes since it employed the probing mechanism to check the status of the surrounding nodes prior to taking any decisions.

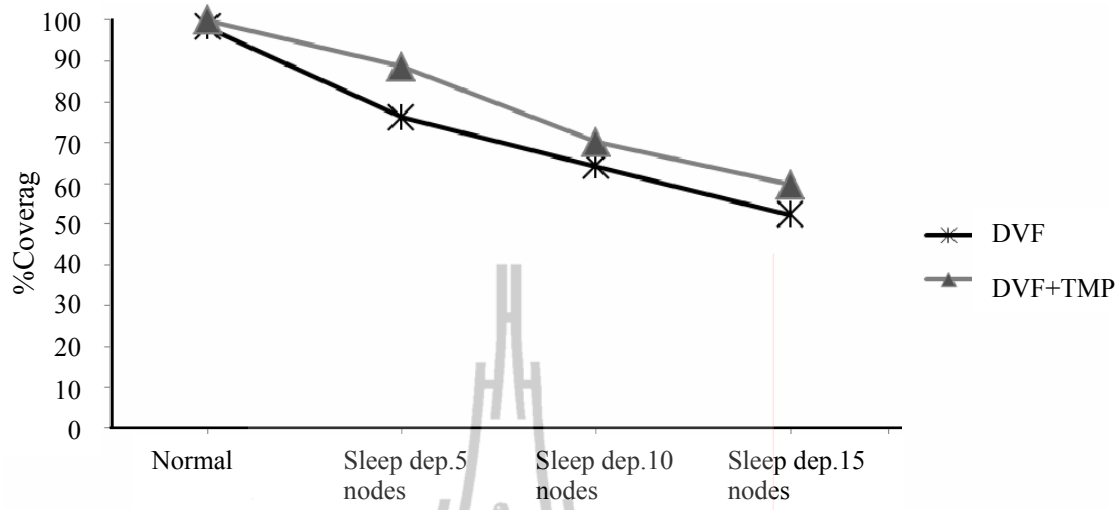


Figure 3.10 Effect of the number of sleep deprived nodes on the percentage of coverage.

In order to show that the amount of coverage our algorithm gained over the original DVF algorithm consumed energy efficiently, the trade-off is illustrated as a function of the number of malicious nodes in Figure 3.11. Results show that under normal situation, both algorithms performed indifferently. However, as the number of sleep deprivation attacked nodes increased, the DVF algorithm was significantly affected whereas that of the proposed DVF+TMP algorithm gradually declined. Although the higher coverage of the DVF+TMP algorithm was attained by an increase in energy consumption, each unit of energy consumed achieved greater coverage therefore better efficiency than the original DVF algorithm.

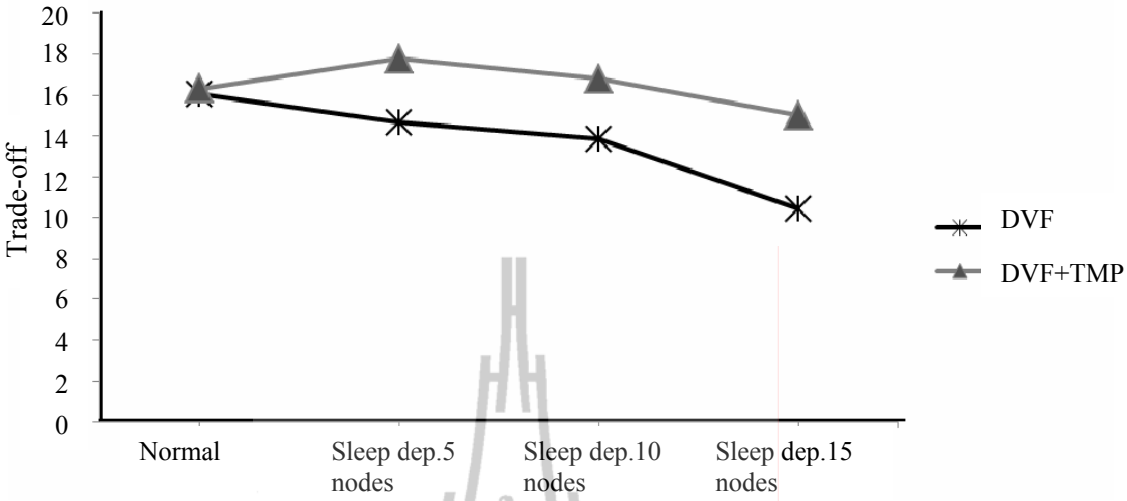


Figure 3.11 Effect of the number of sleep deprived nodes on the trade-off.

As for snooze attack results, Figure 3.12 depicts the effect of the number of the snooze attacked nodes on the percentage of coverage. With no malicious nodes, the percentages of coverage attained by the DVF and DVF+TMP algorithm were 98%, 100% respectively. But as the number of malicious nodes increased, the percentage of coverage achieved by the uncompromised nodes reduced. In particular, the DVF was able to attain up to 74%, 62% and 50% of coverage whereas our proposed algorithm attained up to 86%, 68% and 57% of coverage for 5, 10 and 15 malicious nodes, respectively. Note that prior to taking any actions, each node verified the state of the neighboring nodes by using the probing mechanism of the DVF+TMP algorithm. As a result, the algorithm was able to identify the snooze nodes and increase its coverage accordingly.

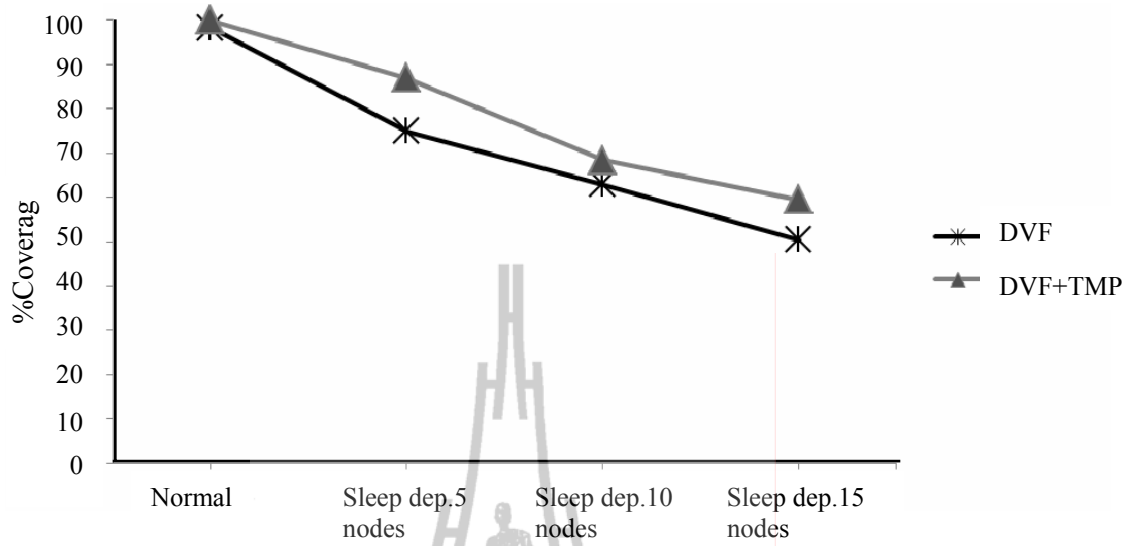


Figure 3.12 Effect of the number of snooze attacked nodes on the percentage of coverage.

Figure 3.13 shows the trade-off as a function of the number of malicious nodes. Note that as the number of snooze nodes increased, the trade-off incurred showed how efficient each algorithm utilized its energy. Once again, a significant drop in trade-off was observed for the DVF algorithm as opposed to the DVF+TMP algorithm where the trade-off was gradually reduced. Therefore, the snooze attack results agree with those of the sleep deprivation attacks, showing that our algorithm can achieve higher coverage, despite the increase in energy consumption due to probing, thereby attaining better trade-off than the DVF algorithm.

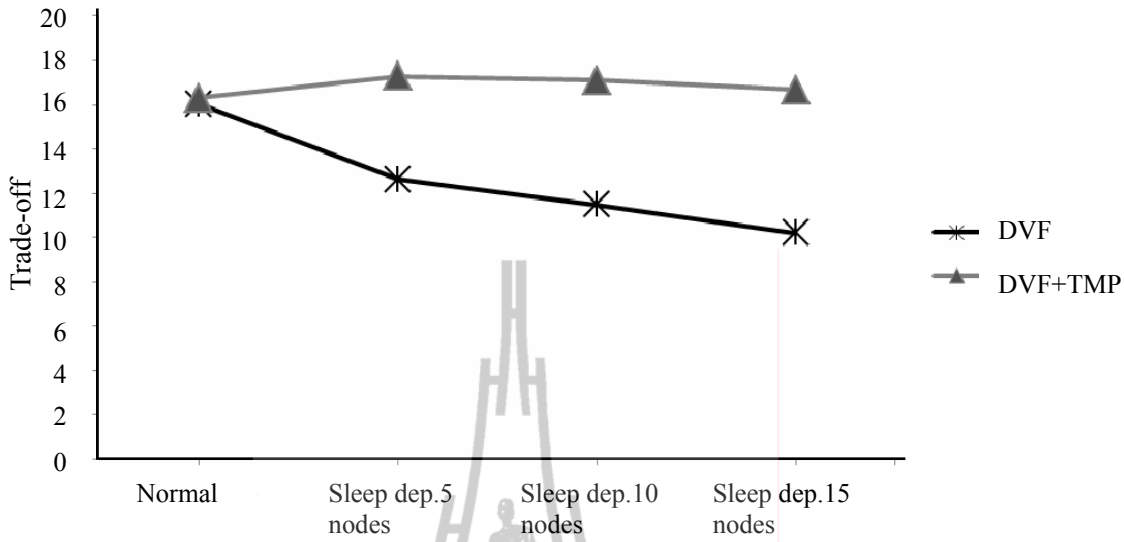


Figure 3.13 Effect of the number of snooze attacked nodes on the trade - off.

3.7.2 Network substitution attacks

In this subsection, we studied the performance of the two algorithms in presence of network substitution attacks. Under this type of attack, the adversary takes control of a node in the system. The compromised node can still maintain connectivity and appear to operate normally. However, since it is completely controlled by the adversary, it can carry out other types of attacks such as selective or complete packet dropping, traffic analysis, sending false or inaccurate readings or information. In this experiment, we assumed that the compromised node exchanges inaccurate information with other agents. We assumed that the degree of inaccuracy was inserted by multiplying the value function in the last summation term in (3.1) by a parameter ξ randomly chosen from the interval [$1-\xi_{max}$, $1+\xi_{max}$] where ξ_{max} was varied from 0.25 to 1, 2.5, 5, 7.5 and 10. Note that ξ_{max} was varied from 0.25 to 10 to cater the larger network (40 agents randomly placed in the area). For this reason, the number of neighbors increased, thereby increasing the value functions updated at each

node. As a result, a small value of ξ_{max} may not clearly affect the system performance. A number of agents were randomly selected to encounter such attack and the results obtained were averaged over all agents.

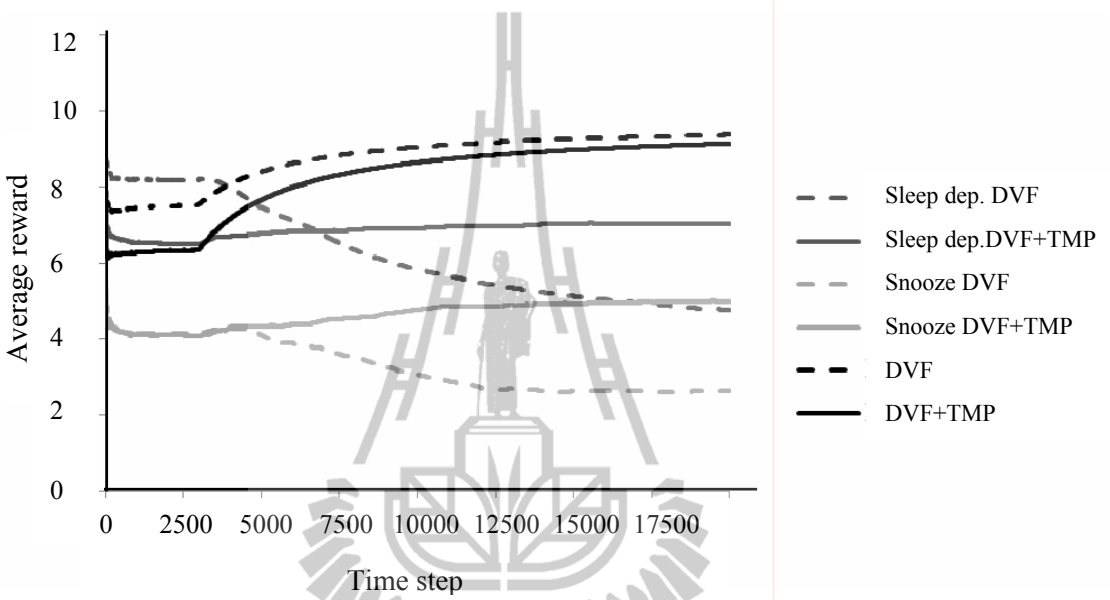


Figure 3.14 Average rewards for sleep deprivation and snooze attacks from 15 malicious nodes and the normal situation, all cases with $\xi_{max} = 10$.

Figure 3.14 illustrates the learning progress of both algorithms in presence of an inaccuracy degree of ξ_{max} = 10, under scenarios of a 15-node sleep deprivation attack, a 15-node snooze attack and normal situation (with neither sleep deprivation nor snooze attacks). It can be observed that as time progressed, the DVF+TMP was able to learn to take better decisions as depicted by the gradually increasing average reward. Although the learning rate of our algorithm was noticeably slow under the sleep deprivation and snooze attack scenarios, the original DVF

algorithm was not able to learn improved decisions at all as the observed from the breakdown in average reward.

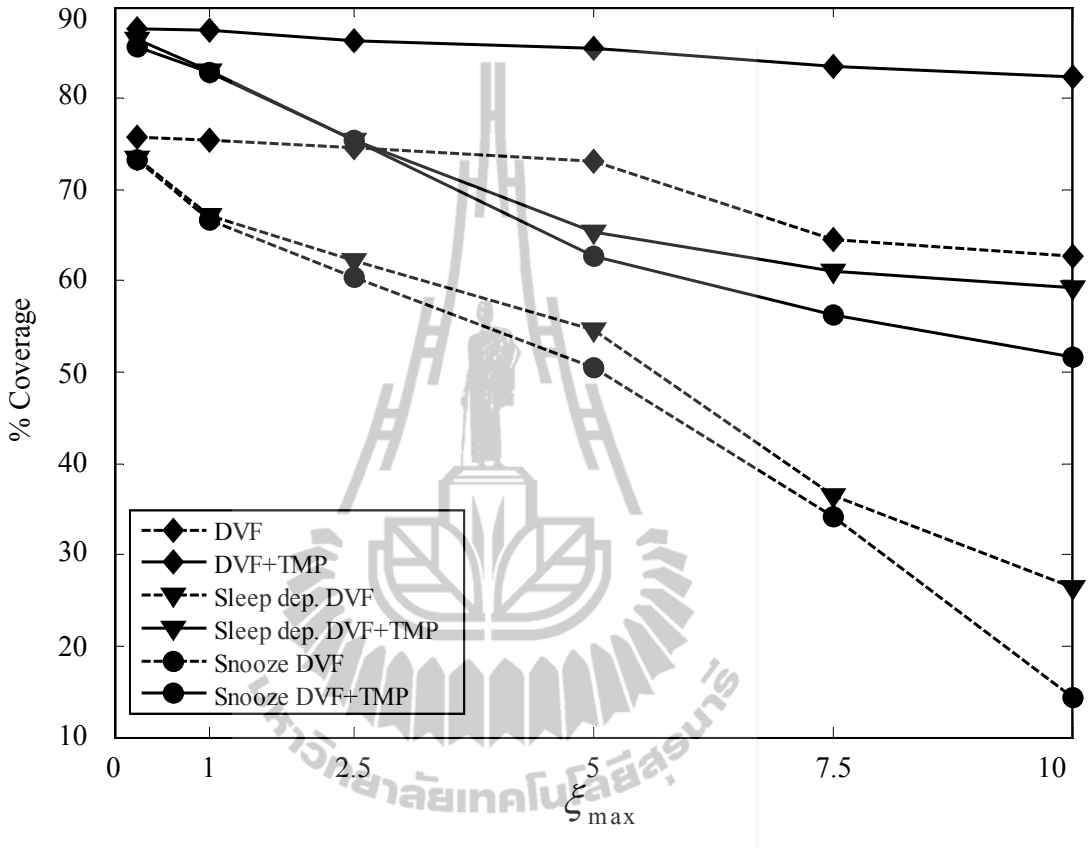


Figure 3.15 Effect of inaccuracy on the percentage of coverage with 5 malicious nodes.

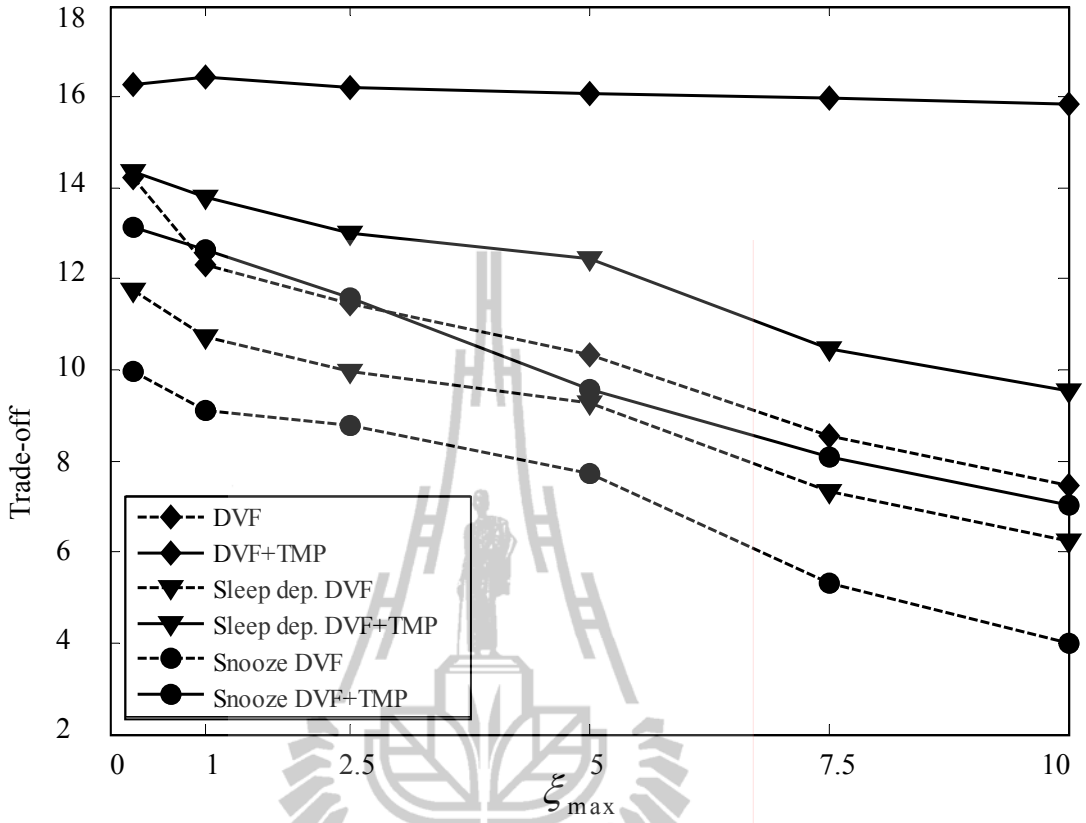


Figure 3.16 Effect of inaccuracy on the trade-off with 5 malicious nodes.

Figure 3.15 depicts the percentage of achievable coverage obtained from different degrees of inaccurate value functions with 5 malicious nodes. Three scenarios were compared i.e. under the normal situation (with no sleep deprivation or snooze attacks), sleep deprivation and snooze attacks. We observed that the higher the degree of inaccuracy, the less the coverage achieved. This confirmed our motivation that distributed coverage control schemes rely on node cooperation and thereby are vulnerable to malicious node attacks. Furthermore, for each scenario, the DVF+TMP algorithm consistently outperformed the DVF algorithm alone by gaining up to 19%, 32% and 37% more coverage in the normal situation, sleep deprivation and snooze attacks, respectively. Figure 3.16 shows the trade-off obtained from different degrees

of inaccurate value functions on each algorithm under normal situation, sleep deprivation and snooze attacks, respectively. Results show that the DVF+TMP algorithm can gain up to 3-8 lit cells per unit energy consumed over the DVF algorithm. Similar to the percentage of coverage, as the degree of inaccuracy increased, the trade-off decreased because decisions were based on inaccurate information exchanged between nodes in the network.

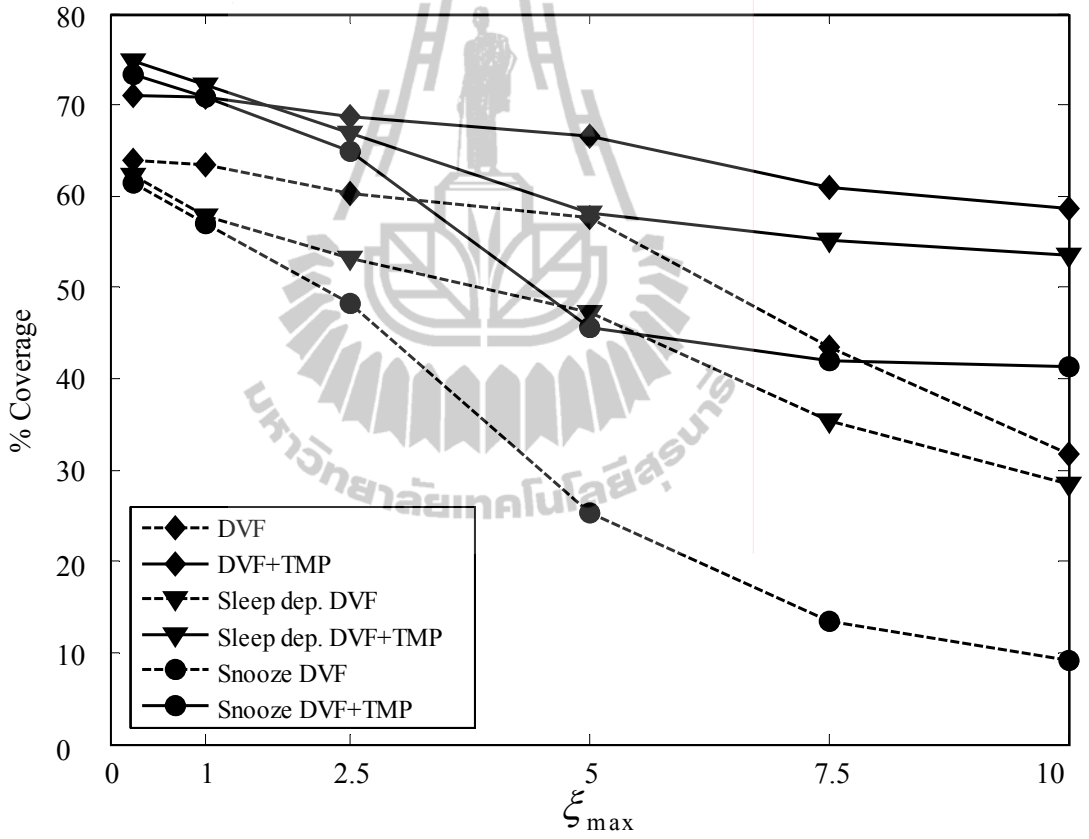


Figure 3.17 Effect of inaccuracy on the percentage of coverage with 10 malicious nodes.

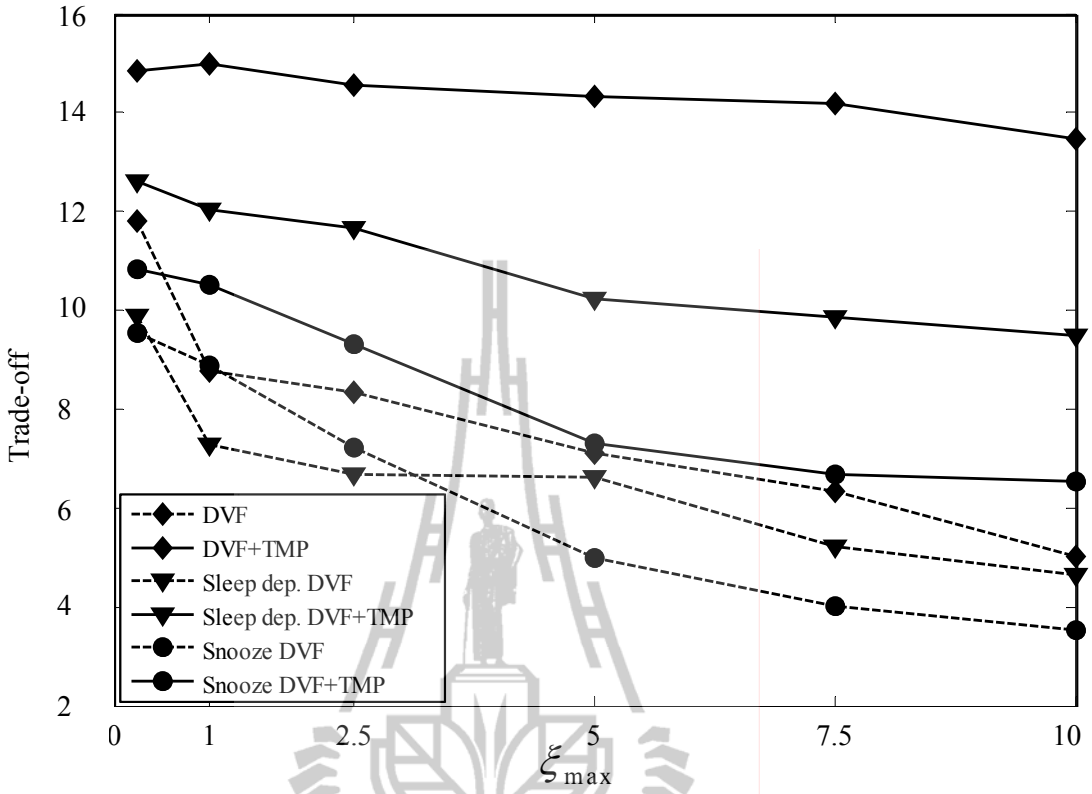


Figure 3.18 Effect of inaccuracy on the trade-off with 10 malicious nodes.

Figure 3.17 shows the percentage of coverage attained in presence of 10 malicious nodes. The DVF+TMP algorithm consistently outperformed the DVF algorithm alone by gaining up to 27%, 25% and 32% more coverage in the normal situation, sleep deprivation and snooze attacks, respectively. However, ξ_{max} within the interval [0-1] did clearly not affect the percentage of coverage since it was too small compared with the value function. Figure 3.18 presents the trade-off against different degrees of inaccurate value functions under normal situation, sleep deprivation and snooze attacks, respectively. Results show that the DVF+TMP algorithm can gain up to 3-8 lit cells per unit energy consumed over the DVF algorithm. Therefore, the results of 10 malicious nodes were in accord with the 5 malicious nodes case, although with less coverage and trade-off attained.

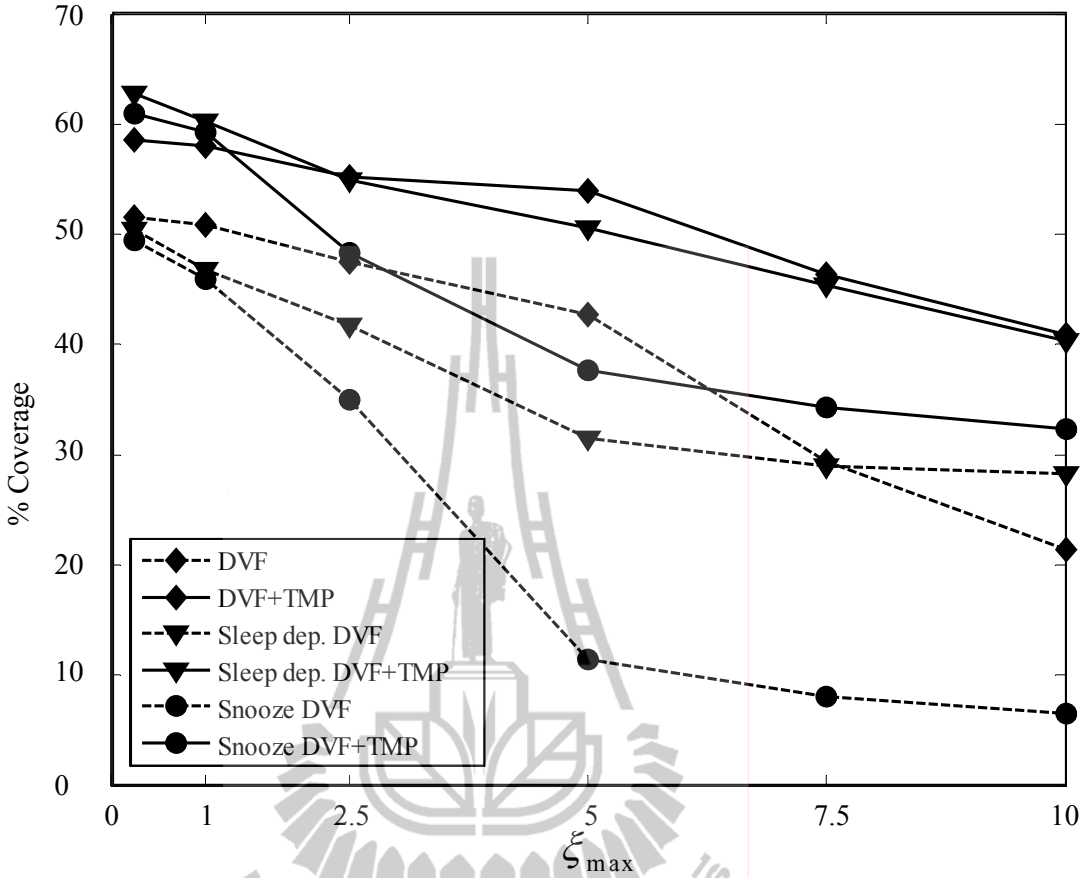


Figure 3.19 Effect of inaccuracy on the percentage of coverage with 15 malicious nodes.

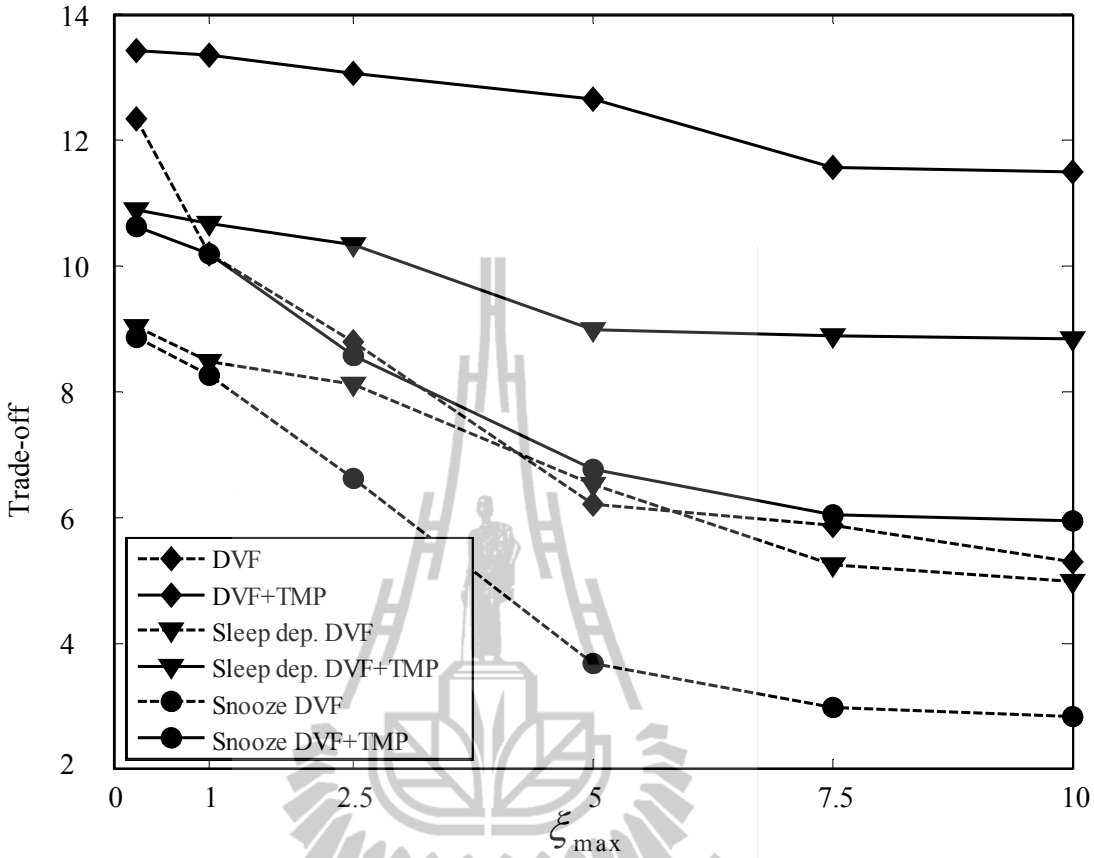


Figure 3.20 Effect of inaccuracy on the trade-off with 15 malicious nodes

Results of the percentage of coverage with 15 malicious nodes are depicted in Figure 3.19. The DVF+TMP algorithm consistently outperformed the DVF algorithm alone by gaining up to 19%, 12% and 26% more coverage in the normal situation, sleep deprivation and snooze attacks, respectively. Figure 3.20 compares the trade-off obtained from different degrees of inaccurate value functions on each algorithm under normal situation, sleep deprivation and snooze attacks, respectively. The results were in agreement with the previous cases of 5 and 10 malicious nodes with ,the DVF+TMP gaining 3-8 lit cells per unit consumed energy over the DVF algorithm, though achieving the least coverage and trade-off results.

3.7.3 Summary: Part 2

In this thesis, we proposed the DVF+TMP coverage control scheme based on the integration of a distributed learning scheme for multi-agent systems called the DVF algorithm and a secure topology maintenance protocol (TMP) to countermeasure sleep deprivation, snooze and network substitution attacks in WSNs. Results showed that the proposed DVF+TMP algorithm was more resilient to malicious node attacks by achieving 6-12% of coverage greater than the original DVF algorithm under sleep deprivation and snooze attacks. Furthermore, in the network substitution attack where various degrees of inaccurate information of value functions were exchanged, the DVF+TMP algorithm gained 19 to 37% of coverage more than the DVF algorithm alone for the normal, sleep deprivation and snooze attack cases, in presence of 5 malicious nodes. When the number of malicious nodes was increased to 10 and 15 nodes, the DVF+TMP algorithm achieved, respectively, 25% to 32% and 12% to 26% more coverage than the original DVF algorithm. In terms of trade-off, for 5-15 malicious nodes, the DVF+TMP algorithm can consistently provide coverage of 3-8 cells per unit energy consumed greater than the DVF algorithm. By integrating the secure topology maintenance protocol, our results suggested that vulnerability to these attacks can efficiently be reduced.

3.8 Implementation

The implementation of the Distributed value function integrated with the Topology maintenance protocol (DVF+TMP) algorithm requires message information exchange between nodes. In terms of memory requirement for storing entries in the Q-table, DVF+TMP algorithm requires memory storage for storing all values of $Q^i(s,a)$ which has $\left(\left|S_{probe\ \mathrm{mode}}\right|+\left|S_{sleep\ \mathrm{mode}}\right|+\left|S_{work\ \mathrm{mode}}\right|\right)\times\left|A\right|$ entries at each agent. Suppose that each entry requires 8 Bytes, a reasonable amount of memory of 1272 Bytes ((1+26+26) x 3 x 8 *Bytes*) was required. The parameters in the learning process such as the learning rate (α) and discount factor (γ) were 0.4 and 0.7, respectively. The learning rate determines to what extent the newly acquired information will override the old information. The discount factor determines the importance of future reward. Their values influence the learning process of algorithm. However, such values may be tuned later for other environment settings.

3.9 Summary

This chapter proposed a coverage control scheme which aimed at maximizing the coverage per unit energy consumption, and was designed to operate in an adversarial malicious environment. In particular, we proposed to integrate the DVF algorithm which was an adaptive and distributed multi-agent coverage control scheme with a secure topology maintenance protocol (TMP) against malicious node attacks in WSNs. More specifically, we incorporated a TMP countermeasure, i.e., *probing,* to verify the local states and active nodes within a neighboring area before increasing or reducing its coverage to allow tolerance against attacks from multiple nodes within a node's transmission range. In this chapter, we divided the experiment into two parts.

The first part included 5 agents with fixed topology in a 10x10 grid room. The results showed that the proposed DVF+TMP algorithm was more resilient to malicious node attacks by achieving up to 75% and 10% of coverage more than the DVF algorithm alone under sleep deprivation and snooze attack, respectively. In network substitution attack where various degrees of inaccurate information of value functions were exchanged, the DVF+TMP algorithm gained up to 12%, 25% and 8% of coverage than the DVF algorithm alone for the normal, sleep deprivation and snooze attack cases, respectively. To ensure that the results obtained in part 1 was not caused by fixed topology, we extended the network to 40 agents in the second part when each agent was randomly placed in a 30x30 grid room. The results showed that DVF+TMP algorithm was more resilient to malicious node attacks by achieving 6-37% of coverage greater than the original DVF algorithm under attacked by malicious nodes.

Simulation results from both parts showed that the proposed DVF+TMP algorithm was more resilient to sleep deprivation and snooze attacks by achieving 6-75 % coverage more than the original DVF algorithm. Furthermore, in the network substitution attack, the DVF+TMP algorithm obtained coverage more than the DVF algorithm alone for the network substitution attack only, and network substitution attack paired with sleep deprivation and snooze attack scenarios. By integrating the secure topology maintenance protocol, our results suggested that vulnerability to these attacks can efficiently be reduced.

CHAPTER IV

CONCLUSIONS AND FUTURE WORK

4.1 Conclusions

In multi-agent system applications in wireless sensor networks, information exchange and cooperation between the agents are required to achieve the objective of maximum coverage and maximum coverage per unit energy consumption. However, it is possible that sensor nodes may act selfishly by declining to service other nodes. Sensor nodes may also encounter attacks by other malicious nodes inside or outside of the network. These attacks may be used to reduce the lifetime of the sensor network, or to degrade the functionality of the sensor application by reducing the network connectivity and the sensing coverage that can be achieved. This thesis proposed a secure multi-agent coverage control scheme for wireless sensor networks with malicious nodes. The research work carried out in this thesis was divided into two parts. The first part studied the coverage control performance of a WSN under attack by means of a lighting control application of a room represented by a 10 x 10 grid. This part contained a group of five agents deployed by fixed on 5 nodes with light sensing capabilities, labeled Ml to M5. In the second part, we extended the lighting control to a room represented by a 30 x 30 grid and with 40 agents placed randomly on the grid. In both parts we studied and compared the performance of 2 algorithms namely the existing Distributed Value Function method and the proposed Distributed Value Function+Topology Maintenance Protocol algorithm under 3 types malicious

attacks namely the sleep deprivation, snooze, network subsitution attacks. The original contributions and findings in this thesis can be summarized as follows.

4.1.1 Secure multi-agent coverage control: Part 1

The purpose of this section is to conceptually show that the Distributed value function algorithm can be integrated with the Topology maintenance protocol to deal with malicious node behaviors i.e. sleep deprivation, snooze, network substitution attacks. Results of the existing multi-agent RL so called the DVF algorithm were compared with the proposed DVF+TMP algorithm under the presence of malicious nodes. In this part, a lighting control system in a 10 x 10 grid room was studied. The room contained a group of five agents deployed on five fixed nodes with light sensing capabilities, labeled Ml to M5. In this part, the following contribution and findings were made here:

1) It was found that the DVF coverage control scheme was directly affected by the presence of malicious nodes in the network. This is due to the fact that DVF relies on cooperation from all nodes to achieve optimal coverage.

2) The proposed algorithm, which is the integrated DVF with a secure Topology maintenance protocol (DVF+TMP) can handle such malicious node attacks by introducing a probing mechanism to verify local states and neighboring nodes.

The performance of the DVF coverage control scheme was evaluated under three types attacks on sensor nodes i.e., sleep deprivation, snooze and network substitution attacks. The original DVF algorithm can learn the optimal policy i.e. the policy which attained the maximum area coverage when full cooperation among nodes was available. However, once the system was attacked by three types of malicious nodes, the original DVF algorithm failed to learn to achieve the optimal policy.

Results showed that in an extreme case when the most critical agent (M1) was attacked in by sleep deprivation, the original DVF algorithm attained zero coverage. Under snooze attack, the percentage of coverage of the original DVF reduced by 10 % when compared with the normal situation. When under the network substitution attack, the percentage of coverage reduced by 17% when compared with the normal situation. This result suggested that the original DVF method alone relied strongly on the cooperation among agents and was vulnerable to security attacks. We proposed to integrate DVF with a secure Topology Maintenance Protocol (DVF+TMP) with can handle such malicious node attacks by introducing a probing mechanism to check node's eligibility to be in sleeping or active state. When the system encountered sleep deprivation attack, the percentage of coverage the proposed DVF+TMP algorithm reduced only by 25% in comparison to 100% loss in DVF. Under snooze attack, the percentage of coverage did not reduced when compared with the normal situation. Finally, for the network substitution attack the percentage of coverage reduced only by 5% when compared with the normal situation. From all above results, we can see the proposed DVF+TMP algorithm is a promising approach to deal with the malicious node attacks.

4.1.2 Secure multi-agent coverage control: Part 2

To ensure that the performance obtained in part 1 was not caused by a particular fixed topology or the placement of agents, the second part of the experiment was conducted. In this part, the network of agents was increased to 40 and the agents were randomly placed in an enlarged 30 x 30 grid room. Once again, the coverage area achieved and the amount energy consumption when a number of agents were under the 3 types of attacks were measured. The results in the second part showed that

when agent was increased to 40 and agents were located randomly, the affected the number of neighbor nodes increased. For this reason, a small value of ξ_{max} (i.e. ξ_{max} = 0.25) affected the system performance slightly. But as ξ_{max} increased ($\xi_{max} \in [1, 10]$), the network substitution attack affected percentage of coverage and trade-off significantly.

This section was performed to show that when the number of agents were increased and agents were placed randomly, the 3 types of malicious node attacks i.e., the sleep deprivation, snooze and network substitution attacks can still affect the performance of the MAS. The simulation results show that, under normal situation (no attack) the DVF and the DVF+TMP algorithms were able to learn to take better decisions as depicted by the gradually increasing average reward. However, when under the 3 types of attacks, that our algorithm was more resilient by consistently attaining higher coverage per unit energy consumed, and achieving 6-12% of coverage greater than the original DVF algorithm under sleep deprivation and snooze attacks. Furthermore, under the network substitution attack the DVF+TMP algorithm gained up to 19%, 32% and 37% of coverage higher than the DVF algorithm for the network substitution attack only, and network substitution attack paired with sleep deprivation and snooze attack, respectively. By integrating the secure topology maintenance protocol, our results suggest that vulnerability to such attacks can efficiently be reduced.

4.2 Future work

4.2.1 Weighting factors of DVF algorithm

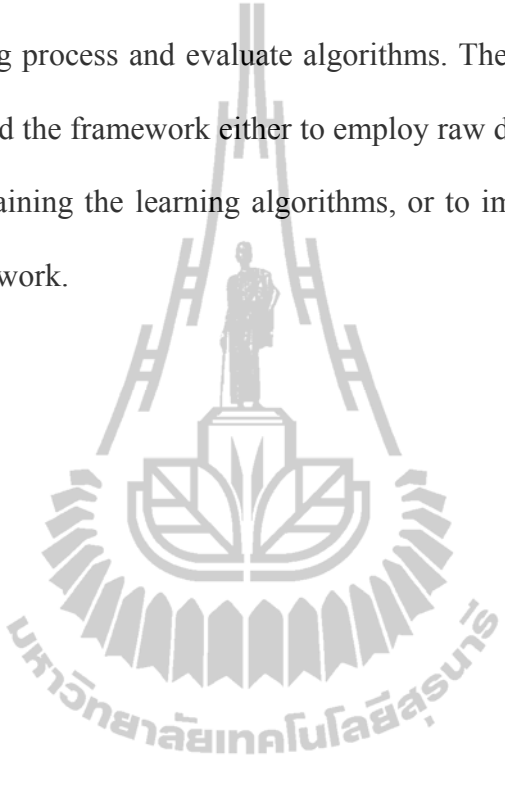
The choice of the weighting factor in the DVF algorithm can influence its overall performance (Schneider, J., et al., 1999). Additional weighting functions need to be studied. These should take into consideration the major constraints of WSNs. For example, the section of the neighboring nodes which a node exchanges, its value function with may be decided based on communication cost parameter. That is, an agent can decide to transmit its value function to its neighbors only when the communication cost incurred is less than the expected gain obtained by the exchange of the value function.

4.2.2 Apply DVF+TMP algorithm to radio model

In thesis, to visually study the coverage control performance of a WSN under attack, we assumed that the agent functioned as a lighting control illuminating a room represented by a 10 x 10 grid and 30 x 30 grid. Therefore, the coverage of each agent was considered in terms of the number of cells lit. However, in practice, the coverage of nodes must be considered in the form of a radio model, which should consider the power received from neighboring nodes to calculate the percentage of coverage area. Hence, in the future work, the algorithm should be extended to cater the radio model.

4.2.3 Performance evaluation of testbed

The main objective of this thesis was to show that coverage control in multi-agent systems in WSNs can be governed by using DVF and DVF+TMP algorithms. The coverage control was simulated by Visual C++ programming to perform the learning process and evaluate algorithms. Therefore, an important future direction is to extend the framework either to employ raw data collected from the field measurement for training the learning algorithms, or to implement the framework in an actual sensor network.



REFERENCES

Stankovic, A.J. (2008). **Wireless Sensor Networks., Chapter in Handbook of Real-Time and Embedded Systems.** CRC Press.

Yick, J., Mukherjee, B., and Ghosal, D. (2008). Wireless Sensor Network Survey. **Journal of Computer Networks,** Vol. 52, No. 12, pp. 2292-2330.

Chitnis, L., Dobra, A., and Ranka, S. (2009). Fault Tolerant Aggregation in Heterogeneous Sensor Networks. **Journal of Parallel and Distributed Computing,** Vol. 69, No. 2, pp. 210-219.

Han, X., Cao, X., Lloyd, E.L., and Shen, C.C. (2010). Fault-Tolerant Relay Node Placement in Heterogeneous Wireless Sensor Networks. **IEEE Transactions on Mobile Computing,** Vol. 9, No. 5, pp. 643-656.

Yu, L., Wang, N., Zhang, W., and Zheng, C. (2007). Deploying a Heterogeneous Wireless Sensor Network. **Proceedings of IEEE International Conference on Wireless Communications, Networking and Mobile Computing.**

Li, M., Lu, Y., and Wee, L. (2006). Target Detection and Identification with a Heterogeneous Sensor Network by Strategic Resource Allocation and Coordination. **Proceedings of IEEE International Conference on ITS Telecommunications.**

Qiu, W., Pham, H., and Skafidas, E. (2008). Routing and localization for extended lifetime in data collection wireless sensor networks. **Proceedings of IEEE International Conference on Communications and Networking in China.**

Liu, Z., Guan, X., and Chen, C. (2008). Energy-efficient optimal scheme based on mixed routing in wireless sensor networks. **Proceedings of IEEE International Conference on Chinese Control Conference.**

Chen, W., Mei, T., Li, Y., Liang, H., Liu, Y., and Meng, M.Q.H. (2007). An Auto-Adaptive Routing Algorithm for Wireless Sensor Networks. **Proceedings of IEEE International Conference on Information Acquisition.**

Wang, C., and Wu, W. (2009). A Load-Balance Routing Algorithm for Multi-sink Wireless Sensor Networks. **Proceedings of IEEE International Conference on Communication Software and Networks.**

Seah, M.W.M., Tham, C.K., Srinivasan, V., and Xin, A. (2007). Achieving Coverage through Distributed Reinforcement Learning in Wireless Sensor Networks. **Proceedings of IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing.**

Munir, S.A., Ren, B., Jiao, W., Wang, B., Xie, D., and Ma, M. (2007). Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing. **Proceedings of IEEE International Conference on Advanced Information Networking and Applications Workshops.**

Tham, C.K., and Renaud, J.C. (2005). Multi-Agent Systems on Sensor Networks: A Distributed Reinforcement Learning Approach. **Proceedings of IEEE International Conference on Intelligent Sensors, Sensor Networks and Information Processing.**

Renaud, J.C., and Tham, C.K. (2006). Coordinated Sensing Coverage in Sensor Networks using Distributed Reinforcement Learning. **Proceedings of IEEE International Conference on Networks.**

Vaz de Melo, P.O.S., Cunha, F.D., Almeida, J.M., Loureiro, A.A.F., and Mini, R.A.F. (2008). The Problem of Cooperation Among Different Wireless Sensor Networks. **Proceedings of the 11th International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems.**

Singsanga, S., Hattagam, W., and Tat, E.H. (2010). Packet Forwarding in Overlay Wireless Sensor Networks using NashQ Reinforcement Learning. **Proceedings of IEEE International Conference on Intelligent Sensors, Sensor networks and Information Processing.**

Wu, M.Y., and Shu, W. (2005). InterSensorNet : Strategic Routing and Aggregation. **Proceedings of IEEE Global Telecommunications Conference.**

Chen, B., Jamieson, K., Balakrishnan, H., and Morris, R. (2002). Span : An Energy-Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks. **Journal of ACM Wireless Networks,** Vol.8, No.5, pp. 481-494.

Cerpa, A., and Estrin, D. (2004). ASCENT : Adaptive Self-Configuring Sensor Networks Topologies. **IEEE Transactions on Mobile Computing,** Vol. 3, No. 3, pp. 272-285.

Ye, F., Zhong, G., Lu, S., and Zhang, L. (2003). PEAS : A Robust Energy Conserving Protocol for Long-Lived Sensor Networks. **Proceedings of IEEE International Conference on Distributed Computing Systems.**

Wang, X., Xing, G., Zhang, Y., Lu, C., Pless, R., and Gill, C. (2003). Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks. **Proceedings of the 1st International Conference on Embedded Networked Sensor Systems.**

Karlof, C., and Wagner, D. (2003). Secure Routing in Wireless Sensor Networks: Attacks and Countermeasures. **Proceedings of IEEE International Workshop on Sensor Network Protocols and Applications.**

Xu, Y., Heidemann, J., and Estrin, D. (2001). Geography-Informed Energy Conservation for Ad Hoc Routing. **Proceedings of the 7th Annual International Conference on Mobile Computing and Networking.**

Stajano, F., and Anderson, R. (1999). The Resurrecting Duckling: Security Issues for Ad-hoc Wireless Networks. **Proceedings of the 7th International Workshop on Security Protocols.**

Gabrielli, A., Mancini, L.V., Setia, S., and Jajodia, S. (2011). Securing Topology Maintenance Protocols for Sensor Networks. **IEEE Transactions on Dependable and Secure Computing,** Vol.8, No.3, pp. 450-465.

Schneider, J., Wong, W. K., Moore, A., and Riedmiller, M. (1999). Distributed Value Functions. **Proceedings of the 16th International Conference on Machine Learning.**

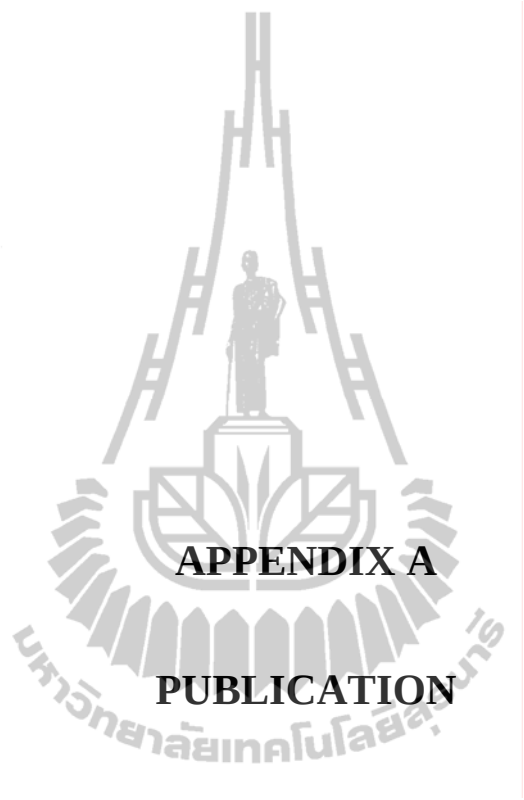
Lauer, M., and Riedmiller, M., (2000). An Algorithm for Distributed Reinforcement Learning in Cooperative Multi-Agent Systems. **Proceedings of the 17th International Conference on Machine Learning.**

Guestrin, C., Lagoudakis, M., and Parr, R. (2002). Coordinated Reinforcement Learning. **Proceedings of the 19th International Conference on Machine Learning.**

Sutton, R., and Barto, A. (1998). **Reinforcement Learning: An Introduction.** The MIT Press.

Kaelbling, L., Littman, M., and Moore, A. (1996). Reinforcement Learning: A Survey. **Journal of Artificial Intelligence Research,** Vol. 4, pp. 237-285.

Puterman, M.L. (1994). **Markov Decision Processes: Discrete Stochastic Dynamic Programming.** Wiley-Interscience.



APPENDIX A

PUBLICATION

Publication

Phuphanin, A., and Usaha, W. (2011). **A Secure Multi-Agent Coverage Control Scheme for Wireless Sensor Networks with Malicious Nodes**. Proceedings of IEEE/ IFIP International Conference on Embedded and Ubiquitous Computing and Information Processing. Melbourne, Australia (Accepted for publication)

Secure Coverage Control in Wireless Sensor Networks with Malicious Nodes using Multi-agents

A. Phuphanin
School of Telecommunication Engineering
Suranaree University of Technology
NakhonRatchasima, Thailand 30000
a.phuphanin@gmail.com

W. Usaha
School of Telecommunication Engineering
Suranaree University of Technology
NakhonRatchasima, Thailand 30000
wusaha@ieee.org

***Abstract*—In this paper, a multi-agent coverage control scheme in wireless sensor networks called the Distributed Value Function was integrated with a secure Topology Maintenance Protocol (DVF+TMP). The objective was to achieve a coverage control scheme which maximizes the coverage per unit energy consumed and countermeasures sleep deprivation, snooze and network substitution attacks in WSNs. Simulation results showed that our algorithm was more resilient by consistently attaining higher coverage per unit energy consumed, and achieving up to 75% and 10% of coverage greater than the original DVF algorithm under sleep deprivation and snooze attacks, respectively. Furthermore, the network substitution attack was studied where inaccurate information was exchanged between nodes. The proposed algorithm gained up to 12%, 25% and 8% of coverage than the DVF algorithm for the normal, sleep deprivation and snooze attack, respectively. The proposed algorithm also consistently achieved more average reward per unit energy consumed than the existing algorithm.**

Keywords-multi-agent; malicious node; control covverage;

I. INTRODUCTION

A wireless sensor network (WSN) is a wireless network consisting of spatially distributed autonomous device using sensors that can communicate with each other to perform sensing and ta processing cooperatively. The overall objective of a WSN is to provide a low-cost solution to gather physical data from the environment, such as noise, pressure, light, observation and transmit it to a base station.

Due to scarce battery supply, topology maintenance and coverage control has become a challenging issue in WSNs. Works in [1-4] aimed at increasing the lifetime of the network by keeping only a subset of sensing nodes active and turning off the remaining redundant nodes. While [1, 2] attempt to maintain connectivity but not guarantee sensing coverage, [3, 4] addressed both network connectivity and coverage requirement.

Distributed self-adaptive coverage control schemes are attractive as WSNs are typically spatially-distributed and deployed in dynamically changing environments which may be difficult to access and manually reconfigure. Such autonomous coverage control can be achieved by multi-agent systems (MAS) [7]. Such distributed approach is also more scalable and compatible with resource-constrained sensor nodes. One of such system called the DVF algorithm has been investigated in [7] where all sensor nodes act as agents that cooperate to achieve a common goal of maximum coverage and minimum energy consumption.

However, all of the aforementioned works were designed for trusted and cooperative environments. With scarce onboard resources, it is possible that sensor nodes may act selfishly by declining to service other nodes [5, 6]. Sensor nodes may also encounter attacks by other malicious nodes inside or outside of the network.These attacks may be used to reduce the lifetime of the sensor network, or to degrade the functionality of the sensor application by reducing the network connectivity and the sensing coverage that can be achieved. We study *three* types of attacks that can be launched in WSNs: *sleep deprivation*are attacks which the adversary tries to induce a node in a specific area to remain active thereby wasting energy and reduce the sensor network lifetime; *snooze attack*which the adversary forces the nodes to remain in the sleeping state thereby reducing sensing coverage or network connectivity; and *network substitution attack* which an adversary controls some nodes which were elected to maintain the connectivity. Once the adversary takes control of a portion of the network, it can carry out other attacks such as sending false information to other nodes. To the best of our knowledge, only [10] presented countermeasures against these types of security attacks in topology maintenance and coverage control schemes in WSNs. However, [10] only aimed at maintaining coverage by using a subset of nodes in an active or awake state, their objective was not to maximize coverage area per unit energy consumed.

This paper therefore proposes a coverage control scheme which aims at maximizing the coverage per unit energy consumption and is designed to operate in an adversarial malicious environment. In particular, we

proposed to integrate the DVF algorithm which is an adaptive and distributed multi-agent coverage control scheme [7] with a secure topology maintenance protocol (TMP) [10] against malicious node attacks in WSNs. More specifically, we incorporated a TMP countermeasure i.e. *probing* to verify the local states and active nodes within a neighboring area before increasing or reducing its coverage to allow tolerance against attacks from multiple nodes within a node's transmission range. Our contribution centers on the integration of the probing mechanism to the DVF scheme and its performance evaluation against sleep deprivation, snooze and network substitution attacks.

II. MULTI-AGENT COVERAGE CONTROL

A multi-agent coverage control scheme called the Distributed Value Function (DVF) has been a common approach to coordinately and cooperatively improve the coverage control performance in wireless sensor networks [7-9]. In this method, each node communicates and exchanges information about its value function. A value function is a function that quantifies how well the agent at a node performs at a given state $s \in S$ where S is a discrete set of all possible states of the sensor network. Let $a \in A$ be the action selected by an agent, where A is the discrete set of all possible actions available at each state. The rule, so called policy π, is defined as a rule which the agent selects actions as a function of states. In other words, it is the mapping from a state $s \in S$ and action $a \in A$ to the probability of selecting action a at state s. The *value function* of state s under a given policy π is formally defined by $V^{\pi}(s) = E^{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s\right\}$, where r_{t+1} is the reward of taking a particular action in a given state s at time t, γ is the discount factor and $E^{\pi}\{\ \}$ is the expectation operator. Similarly, we define the *action value function* of taking action a at a given state under policy π by

$$Q^{\pi}(s,a) = E^{\pi}\left\{\sum_{k=0}^{\infty} \gamma^k r_{t+k+1} \middle| s_t = s, a_t = a\right\}.$$

The objective is to find a policy π^* such that $\pi^* = \underset{\forall \pi}{argmax} Q^{\pi}(s,a)$. To achieve this objective, each agent i (node) in the DVF algorithm performs an update of its own action value function. The update rule at time step t for agent i is given by [7]:

$$Q^i_{t+1}(s^i_t, a^i_t) = (1-\alpha) Q^i_t(s^i_t, a^i_t) + \alpha(r^i_{t+1}(s^i_{t+1}) + \gamma \sum_{j \in Neigh(i)} f^i(j) V^j_t(s^j_{t+1})) \quad (1)$$

$$V^i_{t+1}(s^i_t) = \max_{a \in A^i} Q^i_{t+1}(s^i_t, a) \quad (2)$$

where α is the learning rate, $f^i(j)$ are factors that weigh the value functions of the neighbors of agent i such that:

$$f^i(j) = \begin{cases} \frac{1}{|Neigh(i)|} & \text{, if } Neigh(i) \neq 0 \\ 1 & \text{, otherwise} \end{cases} \quad (3)$$

where $j \in Neigh(i)$ is the set of neighbors of node i [7]. Hence, in the DVF algorithm, nodes cooperate not onlywith their direct neighbors but with all the nodes since the value function captures informationabout other nodes which are not direct neighbors as well. Therefore, the DVF algorithm is strongly dependent on cooperation from other nodes in the network through the last summation term on the right hand side of equation (1). Theinformation exchange (i.e. the value functions of other nodes) in the DVF algorithm is vulnerable to malicious nodes attacks, as such information may be falsely exchanged by a compromised node. This has motivated us to improve the resilience of the DVF algorithm to malicious nodes.

III. MALICIOUS NODE ENVIRONMENT

So far the DVF algorithm has been studied under the assumption that all nodes in the WSN are cooperative [7]. Hence, like other topology maintenance and coverage control schemes assuming this condition, DVF is vulnerable to security attacks where malicious nodes send spoofed or false messages to defeat the objective of the algorithm. This section describes the types of attacks that could occur in a WSN [10]. These attacks could potentially be used to reduce the lifetime of the sensor network, or reduce the achievable network connectivity and sensing coverage.

*Sleep deprivation attack:*In this type of attack, the adversary tries to induce a node in a specific area to remain active. This attack has two effects. First, by increasing the energy expenditure of sensor nodes, it reduces the estimated lifetime of the network. Second, in the case of a densely populated area, it can lead toincreased energy consumption due to congestion and contention at the data link layer.

*Snooze attack:*In this type of attack, the adversary forces the nodes to remain in the sleeping state. This kind of attack can be applied to the whole network or to a subset of nodes. In the latter case, the adversary can launch an attack to jeopardize the connectivity of the network or to reduce the sensing coverage in a region. For example, an adversary can selectively turn off nodes that are monitoring an intruder's path through an area in which a sensor field has been deployed for surveillance.

Network substitution attack: In this type of attack, the adversary deploys some nodes, which are in a set elected by the TMP, to gain control of part of or the entire network. Once these nodes are under control, the

adversary can carry out other attacks such as sharing false or inaccurate information or readings with other nodes. This type of attack is difficult to detect since the compromised node can still maintain connectivity and appear as it were operating as normal.

IV. SECURE MULTI-AGENT COVERAGE CONTROL

In order to make the DVF coverage control scheme more robust to malicious node attacks, TMP probing procedures were integrated into the DVF framework. The probing mechanism verifies whether there are active or inactive nodesin a node's transmission rangebefore decision take any action (i.e. changing the size of coverage area). In particular, in our proposed algorithm, a node can tolerate attacks by up to t nodes within its transmission range.

To study the coverage control performance of a WSN under attack, a lighting control application of a room represented by a 10 x 10 grid was studied as shown in Figure 1. This room contains a group of five agents deployed on five nodes with light sensing capabilities, labeled Ml toM5. Each of them has alight source that can illuminate the part of the room surrounding the agent.The objective is for the agents to learn to cooperatewith one another, in presence of malicious nodes, in order to completely illuminate the room in an energy-efficient way, i.e. minimize the number of lights turned on. The area of agent i, denoted as a^i, refers to the 5 x 5 grid square centered on agent i.

We define three modes for each sensor node, i.e., sleep mode, work mode and probe mode. In sleep mode, a node becomes inactive. In probe mode, a node has just awoken from the sleep mode but is checking on other nodes in its transmission range whether they are active or not, prior to taking any decision. In work mode, nodes are active and can decide to become inactive (since their cells may already be covered by other nodes) or to use low or high coverage. On the other hand, in the original DVF algorithm, a node collects and exchanges data with its neighboring nodes, and immediately decides whether to be in an active (awake) or to be inactive (asleep) state. In our proposed integrated DVF and TMP algorithm, before a node enters work mode, it enters probe mode to check whether or not there exists other nodes within its transmission range already in work mode or not.

Local agent state: Each agent i can sense the level of light in its area. Its local state s^i is state of each agent based on its mode and coverage. The state of mode consists of three modes i.e. sleep mode, work mode and probing mode. In sleep mode, there is 1 possible state (i.e. no lit cells). In probing and work modes, each has 26 possible states. Therefore, there are 53 possible states (1+26+26) for the system given by:

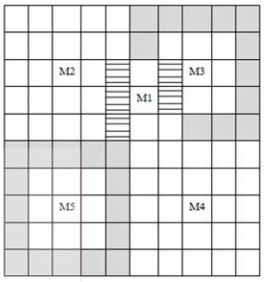


Figure 1.A 10x10 grid room representation. Grey cells are not illuminated, white cells are illuminated by one node and striped cells are illuminated by two nodes.

$$s^i = (state\ of\ mode, state\ of\ coverage) = (s_m^i, s_c^i)$$

where $s_m^i \in \{sleep\ mode, probe\ mode, work\ mode\}$ and $s_c^i \in \{0, \ldots 25\}$.

Local agent actions: Each agent i has the ability to take one of the following three actions in any state it lands in. The action space A^i is the set of all possible actions for each state $A^i = \{$*Action 0* (Turn off the light), *Action 1* (Turn on the light in LOW coverage. This illuminates 9 cells around the agent, as shown by Ml, M3 and M5 in Fig.1), *Action 2* (Turn on the light in HIGH coverage. This illuminates the 25 cells around the agent, as shown by M2 and M4 in Fig.1), and *Action 3* (Send probe to neighbors and wait for a response within in a finite time. This action allows the agent to sense the illuminated cells within its range prior to deciding to take *Action 0, Action 1 or Action 2*)$\}$. Once an action is taken, the current local state of the agent transits to a new local state accordingly as shown in Figure 2.

Note that in the distributed learning schemes such as the DVF algorithm, the agents use only information that is locally available to make their decisions. The reward for agent i, denoted as $r^i\left(s_t^i\right)$ is a function of agent i's state s^i at time t and is defined by:

$$r^i(s_t^i) = G^i(s_t^i) - C^i \quad (4)$$

where $G^i(s_t^i)$ is a function of the number of cells illuminatedin the area of agent i such that

$$G^i(s_t^i) = nb\_cell\_bright(a^i) \times GAIN\_CELL\_BRIGHT, \quad (5)$$

and C^i is the energy consumption resulting from the action taken by agent i at time t such that

$$C^i = \begin{cases} 0 & \text{, if Action 0 was taken} \\ COST\_LOW & \text{, if Action 1 was taken} \\ COST\_HIGH & \text{, if Action 2 was taken} \\ COST\_PROBE & \text{, if Action 3 was taken} \end{cases} \quad (6)$$

The reward functions and state transitions for the proposed DVF+TMP algorithm are shown in Figure 2. Note that when the agent decides to take Action0, a reward G is obtained since the local cells could still be lit by neighboring nodes and the cost is zero since no energy is consumed if the agent becomes inactive. For other actions, the agent is rewarded with G subtracted by a non-zero cost in (6).

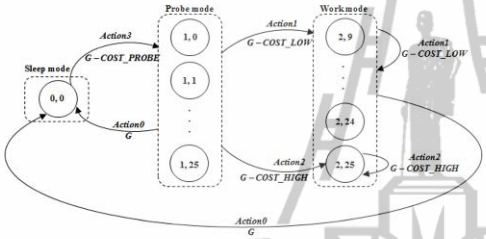


Figure 2.State transition diagram of the probing mechanism in the DVF+TMP algorithm.

Four performance metrics were considered: 1) the average reward per time step defined by:

$$average\ reward = \frac{\sum_{t=0}^{T} r^i\left(s_t^i\right)}{T}, \quad (7)$$

where T is the total number of time steps, and $r^i\left(s_t^i\right)$ is given by (4); 2) the average energy consumption; 3) the percentage of coverage area from good (uncompromised) nodes defined as the total number of cells illuminated by all good nodes at a time step divided by the total number of cells in the system; and 4) the trade-off which is defined by:

$$Trade\text{-}off = \frac{\sum_{t=0}^{T}\sum_{\forall i} number\ of\ lit\ cells\ by\ agent_i(t)}{\sum_{t=0}^{T}\sum_{\forall i} energy\ consumption\ by\ agent_i(t)} \quad (8)$$

Note that the trade-off in the above equation reflects the total number of illuminated cells throughout the simulation over the total amount of energy consumption. The trade-off represents the number of illuminated cells (coverage) per unit energy consumed. It is expected that the better the system can deal with malicious nodes, the better (and more efficient) the uncompromised sensor nodes can decide and therefore the higher the trade-off.

In the simulation, we used $GAIN\_CELL\_BRIGHT$ = 0.5, $COST\_LOW$ = 0.8, $COST\_HIGH$ = 3, $COST\_PROBE$ = 0.5 the learning rate α = 0.4 and the discount factor γ = 0.7.The values of the learning rate and discount factor were obtained from experimenting a range of values and selecting the parameters which received the best performance in terms of average reward per time step. The run length of each simulation was T = 20,000 time steps and the results were averaged over 10 runs to achieve the desired accuracy.

V. Results and Analysis

A. Sleep Deprivation and Snooze Attacks

We assigned each agent to encounter sleep deprivation attack and snooze attack and then analyze the results in Figures 3-6. Note also the percentages of coverage in the normal situation (no attack) are also shown in Figures 3 and 5 (represented by DVF and DVF+TMP). Denote "M1-w", "M2-w", "M3-w", "M4-w", "M5-w" for cases when agent 1, 2, 3, 4 and 5 were each attacked by sleep deprivation, respectively. Similarly, denote "M1-s", "M2-s", "M3-s", "M4-s", "M5-s" for cases where agent 1, 2, 3, 4 and 5 were each attacked by snooze attack, respectively. As a benchmark to compare the effects from malicious nodes, i.e., sleep deprivation and snooze attacks, we observe how each agent operates in the normal situation. We compared our proposed integrated DVF and TMP algorithm (abbreviated by DVF+TMP) with the original DVF algorithm (abbreviated by DVF).

Figure 3.depicts the sleep deprivation effect on the percent coverage of the DVF algorithm compared with the proposed DVF+TMP algorithm. In case of sleep deprivation attack on agents 2, 3, 4, and 5, we can see that the percentage of coverage for each case achieved at 75%. Furthermore, for each case, convergence to a policy which obtained the most coverage was achieved. However, in the case when agent 1 was attacked by sleep deprivation, the original DVF algorithm attained zero coverage. Note that agent 1 was located in the center of the area and its coverage overlapped those of agent 2, 3, 4, and 5 (see Figure 1). Such result showed that when agent 1 was under sleep deprivation attack, all the other good (uncompromised) agents in the system were falsely led to converge to sleep mode thereby attaining zero good node coverage with the original DVF scheme. On the other hand, when agent 1 was attacked by sleep deprivation, the proposed DVF+TMP algorithm can attain 75% percentage of coverage. Figure 4 shows the trade-off results. In the cases of sleep deprivation attack on agent 2, 3, 4, 5 achieved the maximum tradeoff that

could be achieved, thereby agreeing with the percentage of coverage results. Note that in the case when agent 1 was attacked, we can see that the average coverage per energy consumption unit of DVF+TMP was significantly better than the original DVF algorithm.

Figures 5, 6 illustrate the snooze attack effect on the percentage of coverage and the trade-off in both algorithms. In Figure 5, when agent 2, 3, 4 were attacked by snooze attack, the final coverage results of the DVF+TMP algorithm obtained were up to 5% 10% 10% respectively is more than those of the DVF algorithm . Note that when agent 1 was under snooze attack, both algorithms eventually attained 100% coverage. This was because all agents must work in HIGH mode when agent 1 was attacked, in which case was the optimal policy for the system. When considering the trade-offin Figure 6, we can see that all agents depicted similar patterns though the DVF+TMP algorithm consistently gave a better trade-off (i.e. more cells illuminated per unit energy) than the DVF algorithm.

So far the nodes under attack have been predetermined. In the next experiment, we evaluated the performance of the DVF and DVF+TMP algorithms when the malicious nodes were randomly generated. Tables 1 and 2 show the results as the number of malicious nodes were increased from 0 (normal situation) to 4 (worst case scenario). Table 1 shows results from DVF and DVF+TMP algorithms under the sleep deprivation attack. Although the average rewards were increased as a result of the attack, the average energy consumption was high. However, the DVF+TMP algorithm achieved higher average reward per unit energy consumed than the DVF algorithm alone. On the contrary, Table 2 shows that as the number of nodesunder snooze attack increased, the average reward dropped significantly along with the energy consumption. Once again, our algorithm attained more average reward than the DVF alone. Furthermore, under this attack, the average reward per unit energy consumed by the DVF+TMP scheme was also higher than the DVF algorithm. The results in Tables 1 and 2 agreed with the trade-off in Figures 3 and 5, indicating that our method can achieve more coverage per unit energy consumed than the DVF alone.

B. Network Substitution Attacks

In this subsection, we study the performance of the two algorithms in presence of network substitution attacks. Under this type of attack, the adversary takes control of a node in the system. The compromised node can still maintain connectivity and appear to operate normally. However, since it is completely controlled by the adversary, it can carry out other types of attacks such as selective or complete packet dropping, traffic analysis, send false or inaccurate readings or information. We assume that the compromised node exchanges inaccurate information with other agents. We assume that the degree of inaccuracy is inserted by multiplying the value function in the last summation term in (1) by a parameter ξrandomly chosen from the interval$[1-\xi_{max}, 1+\xi_{max}]$ where ξ_{max}= 0.25, 0.5, 0.75 and 1. Each agent encountered such attack and the results obtained were averaged over all agents. Figure 7 depicts the percentage of achievable coverage obtained from different degrees of inaccurate value functions on each algorithm under normal situation (with no sleep deprivation or snooze attacks), sleep deprivation and snooze attacks, respectively. The higher the degree of inaccuracy, the less the coverage achieved. This confirms our motivation that distributed coverage control schemes rely on node cooperation and thereby are vulnerable to malicious node attacks. Furthermore, for each scenario, the DVF+TMP algorithm consistently outperforms the DVF algorithm alone by gaining up to 12%, 25% and 8% more coverage in the normal situation (with no other attacks), sleep deprivation and snooze attacks, respectively.

The effects of inaccurate value functions on the average reward, average energy consumption, and the ratio of the two parameters are shown in Tables 3, 4 and 5 for the normal situation (with no sleep deprivation or snooze attacks), sleep deprivation and snooze attacks, respectively. From the three tables it can be seen that as the degree of inaccuracy increases, the average reward from both algorithms decreased accordingly. However, the DVF+TMP algorithm consumed less average energy than the DVF algorithm alone therefore achieved higher efficiency in terms of average reward per unit energy consumed for all cases.

All of these results suggested that the DVF alone relies strongly on the cooperation among agents and is vulnerable to security attacks. The proposed DVF+TMP scheme can enhance the security and can cope with sleep deprivation, snooze and network substitution attacks, thereby improving the resilience of the distributed coverage control scheme.

VI. CONCLUSION

In this paper, we proposed the DVF+TMP coverage control scheme based on the integration of a distributed learning scheme for multi-agent systems called the DVF algorithm and a secure topology maintenance protocol (TMP) to countermeasure sleep deprivation, snooze and network substitution attacks in WSNs. To evaluate its performance, a lighting control application was studied. The results showed that in the presence of malicious nodes in the system, the original DVF algorithm was directly affected suggesting that the DVF algorithm alone strongly relies on the cooperation between nodes. However, results showed that the proposed DVF+TMP algorithm was more resilient to malicious node attacks byachieving up to 75% and 10% of coveragemore than the DVF algorithm alone under sleep deprivation and snooze attack, respectively. The proposed algorithm also

attained a better trade-off in terms of the number of cells illuminated per unit energy consumed. Similar results were achieved when the presence of malicious nodes in the system were increased. Furthermore, in the network substitution attack where various degrees of inaccurate information of value functions were exchanged, the DVF+TMP algorithm gained up to 12%, 25% and 8% of coverage than the DVF algorithm alone for the normal, sleep deprivation and snooze attack cases, respectively. The proposed algorithm also consistently achieved more average reward per unit energy consumed than the DVF algorithm.

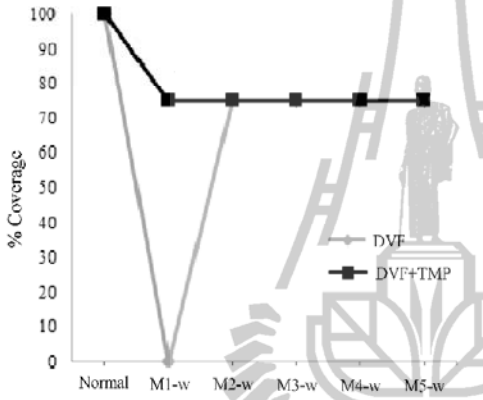


Figure 3.Sleep deprivation effect on the percentage of coverage.

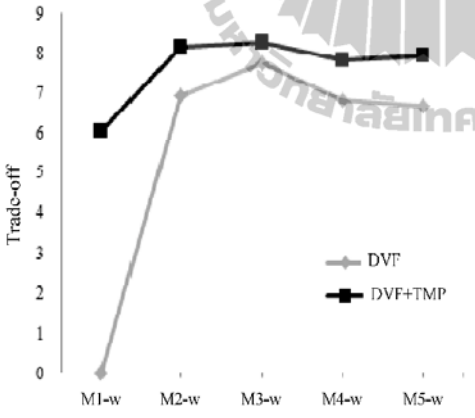


Figure 4.Sleep deprivation effect on the trade-off.

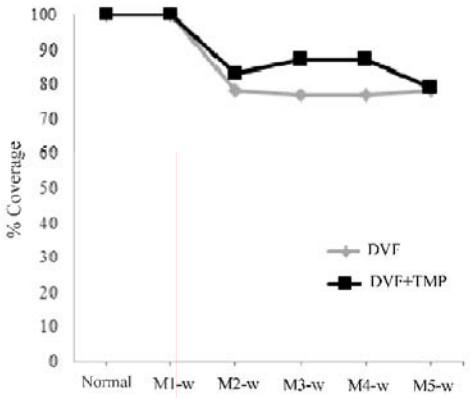


Figure 5.Snooze effect on the percentage of coverage.

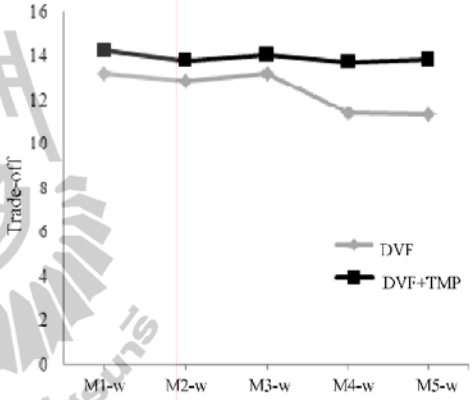


Figure 6.Effect of the number of snooze attacked nodes on the percentage of coverage.

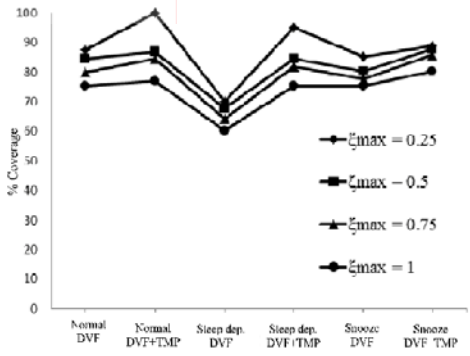


Figure 7.Effect of inaccuracy on the percentage of coverage.

TABLE I. RANDOMLY GENERATED SLEEP DEPRIVATION ATTACK RESULTS.

No. of attacked nodes	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avgreward	Avg energy consumed	Avgreward/ Avg energy consumed	Avg reward	Avg energy consumed	Avgreward/ Avg energy consumed
0	6.19	1.18	5.23	6.59	1.21	5.45
1	8.57	1.60	5.35	8.98	1.64	5.47
2	9.33	1.80	5.18	9.65	1.80	5.36
3	9.85	1.95	5.04	10.35	2.04	5.07
4	10.21	2.09	4.88	11.13	2.12	5.26

TABLE II. RANDOMLY GENERATED SNOOZE ATTACK RESULTS.

No. of attacked nodes	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avgreward/ Avg energy consumed	Avg reward	Avg energy consumed	Avgreward/ Avg energy consumed
0	6.19	1.18	5.23	6.59	1.21	5.45
1	3.03	0.75	4.06	5.39	0.88	6.15
2	1.87	0.62	3.03	4.39	0.80	5.50
3	1.26	0.49	2.58	3.34	0.55	6.06
4	1.03	0.44	2.36	2.98	0.54	5.49

TABLE III. EFFECT OF INACCURACY IN NORMAL SCENARIO.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0.25	7.14	1.54	4.63	7.90	1.55	5.11
0.5	7.11	1.56	4.57	7.84	1.54	5.11
0.75	7.08	1.55	4.58	7.76	1.53	5.09
1	7.02	1.55	4.55	7.67	1.51	5.07

TABLE IV. EFFECT OF INACCURACY INSLEEP DEPRIVATION SCENARIO.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0.25	7.20	1.56	4.63	7.91	1.52	5.20
0.5	7.10	1.55	4.60	7.58	1.49	5.09
0.75	6.90	1.54	4.49	7.21	1.50	4.81
1	6.70	1.53	4.37	6.87	1.49	4.61

TABLE V. EFFECT OF INACCURACY INSNOOZE ATTACK SCENARIO.

ξ_{max}	Coverage control schemes					
	DVF algorithm			DVF+TMP algorithm		
	Avg reward	Avg energy consumed	Avg reward/ Avg energy consumed	Avg reward	Avg energy consumed	Avg reward / Avg energy consumed
0.25	6.89	1.58	4.36	7.76	1.57	4.95
0.5	6.86	1.57	4.32	7.57	1.56	4.86
0.75	6.77	1.56	4.30	7.28	1.52	4.80
1	6.63	1.54	4.29	7.01	1.51	4.65

ACKNOWLEDGEMENT

This work is financially supported by the Telecommunication Research Industrial and Development Institute (TRIDI), National Telecommunication Commission Fund, Thailand.

REFERENCES

[1] B. Chen, K. Jamieson, H. Balakrishnan, R. Morris, "Span: An energy efficient coordination algorithm for topology maintenance in ad hoc wireless networks", ACM Wireless Networks Journal, Vol. 8 (5), Sep 2002.

[2] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies", IEEE Trans. on Mobile Computing, Vol. 3 (3), Jul-Aug 2004.

[3] F.Ye, G. Zhong, S. Lu, L. Zhang, "PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks.", Proc. of the 23rd IEEE Intl. Conf. on Distributed Computing System (ICDCS'03), May 2003.

[4] X. Wang, G. Xing, Y. Zhang, C. Lu, R. Pless, C. Gill, "Integrated Coverage and Connectivity Configuration in Wireless Sensor Networks", ACM Trans. on Sensor Network (TOSN), Vol. 1 (1), Aug 2005.

[5] P. Vaz de Melo, F. da Cunha, J. Almeida, A. Loureiro, and R. Mini, "The Problem of Cooperation Among Different Wireless Sensor Networks", Proceedings of the International Symposium on Modelling, Analysis and Simulation of Wireless and Mobile Systems, 2008.

[6] S. Singsanga, W. Hattagam, H.T Ewe, "Packet Forwarding in Overlay Wireless Sensor Networks Using NashQ Reinforcement Learning", Intelligent Sensors, Sensor Networks and Information Processing (ISSNIP'10), Dec 2010.

[7] C.K. Tham, J.C. Renaud, "Multi-agent systems in sensor networks: A distributed reinforcement learning approach", Proceedings of the 2005 International Conference on Intelligent Sensors, Sensor Networks and Information Processing, December 2005.

[8] M.W.M. Seah, C.K Tham, V. Srinivasan and A. Xin "Achieving Coverage through Distributed Reinforcement Learning in Wireless Sensor Network", International Conference on Intelligent Sensors, SensorNetworks and Information, pp. 425-430, Dec, 2007.

[9] J.C. Renaud and C.K. Tham "Coordinated sensing coverage in sensor network using distributed reinforcement learning", IEEE International Conference on Networks, ICON '06, Vol. 1, pp. 1-6, Feb. 2007.

[10] A. Gabrielli, L. V. Mancini, S. Setia, and S. Jajodia, "Securing Topology Maintenance Protocols for Sensor Network", IEEE Trans. on Dependable and Secure Computing, Vol.8 (3), pp.450-465, May-Jun.2011.

BIOGRAPHY

Mr. Akkachai Phuphanin was born on October 29, 1986 in Kalasin province, Thailand. He finished high school education from Kalasinpittayasan School, Kalasin province. He received his Bachelor's Degree in Engineering (Telecommunication) from Suranaree University of Technology in 2009. For his post-graduate, he continued to study for a Master's degree in the Telecommunication Engineering Program, Institute of Engineering, Suranaree University of Technology of which he was granted a scholarship form Telecommunication Research Industrial and Development Institute (TRIDI). During his Master's degree education, he was a visiting researcher at Centre for Dynamic Intelligent Communication (CIDCOM), Department of Electrical and Electronic Engineering, University of Strathclyde, Scotland in the topic of wireless sensor networks.