การเลือกเส้นทางโดยใช้ค่าตอบแทนสำหรับเครือข่ายตัวตรวจรู้ไร้สายเคลื่อนที่ต่างประเภทที่ไม่ทำงานร่วมกัน

นายชานนท์ ฤทธิ์ทอง

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2554

INCENTIVE-BASED ROUTING FOR NON-COOPERATIVE HETEROGENEOUS MOBILE WIRELESS SENSOR NETWORKS

Chanon Rittong

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

INCENTIVE-BASED ROUTING FOR NON-COOPERATIVE HETEROGENEOUS MOBILE WIRELESS SENSOR

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. Dr. Piyaporn Krachodnok)
Member

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

(Assoc. Prof. Flt. Lt. Dr. Kontorn Chamniprasart)
Dean of Institute of Engineering

ชานนท์ ฤทธิ์ทอง : การเลือกเส้นทางโดยใช้ค่าตอบแทนสำหรับเครือข่ายตัวตรวจรู้ไร้สายเคลื่อนที่ต่างประเภทที่ไม่ทำงานร่วมกัน (INCENTIVE-BASED ROUTING FOR NON-COOPERATIVE HETEROGENEOUS MOBILE WIRELESS SENSOR NETWORKS) อาจารย์ที่ปรึกษา : ผู้ช่วยศาสตราจารย์ ดร.วิภาวี หัตถกรรม, 100 หน้า.

เครือข่ายตัวตรวจรู้ไร้สายได้ถูกพัฒนาและประยุกต์ใช้ใช้งานอย่างกว้างขวางในหลายๆ ด้าน หนึ่งในการประยุกต์ใช้ที่มีความสำคัญเป็นอย่างมากคือการนำไปประยุกต์ใช้กับการเฝ้าระวังด้านสุขภาพ อย่างไรก็ตามเครือข่ายตัวตรวจรู้ไร้จะต้องสามารถส่งข้อมูลข่าวสารที่มีความสำคัญสูงได้ นอกจากนี้แล้วโหนดจะต้องติดไปกับตัวผู้ป่วย และมีความสามารถในการรองรับชนิดของข้อมูลในการส่งผ่านที่มีความแตกต่างกัน ข้อมูลที่ส่งผ่านไปยังศูนย์เฝ้าระวังทางการแพทย์นั้นจำเป็นต้องมีความน่าเชื่อถือสูง

วัตถุประสงค์ของงานวิจัยนี้ก็คือนำเสนอการปรับปรุงวิธีการหาเส้นทางโดยใช้ค่าตอบแทนสำหรับเครือข่ายตัวตรวจรู้ไร้สายต่างประเภทที่ไม่ทำงานร่วมกันโดยใช้อัลกอริธึมเรียนรู้แบบรีอินฟอร์สเมนท์ (reinforcement learning; RL) เรียกว่าวิธีการเรียนรู้แบบคิว (Q-learning) เมื่อเปรียบเทียบกับวิธีการเดิมที่มีอยู่แล้วซึ่งเรียกว่าอัลกอริธึมซีวีซีพี (continuous value cooperation protocol; CVCP) เพื่อใช้ในการรับมือร่วมกับกับเครือข่ายตัวตรวจรู้ไร้สายต่างประเภทที่ไม่ทำงานร่วมกัน งานวิจัยชิ้นนี้ได้ศึกษาความแตกต่างในเชิงของชนิดข้อมูลที่ปรากฏในเครือข่ายและความแตกต่างในเชิงของอัตราการประมวลผลของโหนด

ผลการทดลองแสดงให้เห็นว่าอัลกอริธึมวิธีการเรียนรู้แบบรีอินฟอร์สเมนท์ที่นำเสนอสามารถให้ประสิทธิภาพสูงกว่าอัลกอริธึมซีวีซีพีที่มีอยู่แล้วในเทอมของค่าตอบแทนระยะยาวเฉลี่ยมูลฐาน (normalized average reward) สูงถึง 14% อย่างไรก็ตามค่าเปอร์เซ็นต์ของความสามารถในอัตราการประมวลผลของโหนดนั้นจะไม่ขึ้นอยู่อัลกอริธึมใดๆ แต่จะขึ้นอยู่กับสัดส่วนของชนิดของโหนดตามอัตราการประมวลผล ผลการทดลองดังกล่าว แสดงให้เห็นถึงความไม่ลำเอียงในการเลือกโหนด ทั้งยังคงรักษาความได้เปรียบของค่าตอบแทนระยะยาวเฉลี่ยมูลฐาน ซึ่งมีค่าสูงกว่าวิธีซีวีซีพี ดังนั้นความแตกต่างของความสามารถในการประมวลผลของโหนดจึงไม่ส่งผลที่มีนัยสำคัญต่อผลการทดลอง แต่อย่างไรก็ตามสำหรับความแตกต่างในเชิงของชนิดของข้อมูลข่าวสารในเครือข่ายนั้นวิธีการเรียนรู้แบบรีอินฟอร์สเมนท์มีความได้เปรียบวิธีการซีวีซีพีแบบดั้งเดิมอยู่ 2-14% อย่างสม่ำเสมอในค่าของค่าตอบแทนระยะยาวเฉลี่ยมูลฐานซึ่งขึ้นอยู่กับคุณลักษณะค่าตอบแทนตามชนิดของข้อมูลข่าวสาร ผลการทดลองในการทดลอง

ของเราชี้ให้เห็นว่าวิธีการเรียนรู้แบบรีอินฟอร์สเมนท์สามารถนำมาประยุกต์ใช้เพื่อปรับปรุงความร่วมมือระหว่างโหนดในเส้นทางเมื่อเปรียบเทียบกับอัลกอริธึมที่มีอยู่แล้วอย่างเช่นวิธีการซีวีซีพีได้

สาขาวิชา<u>วิศวกรรมโทรคมนาคม</u> ลายมือชื่อนักศึกษา____________________

ปีการศึกษา 2554 ลายมือชื่ออาจารย์ที่ปรึกษา______________

CHANON RITTONG : INCENTIVE-BASED ROUTING FOR NON-COOPERATIVE HETEROGENEOUS MOBILE WIRELESS SENSOR NETWORKS. THESIS ADVISOR : ASST. PROF. WIPAWEE HATTAGAM, Ph.D., 100 PP.

MOBILE WIRELESS SENSOR NETWORKS/ REINFORCEMENT LEARNING/ NON-COOPERATIVE/ Q-LEARNING/ ROUTING COOPERATION

Wireless Sensor Networks (WSNs) have been developed and extensively applied in many fields. One of the most important applications is healthcare monitoring. However, wireless sensor networks must be able to transmit messages with high priority. In addition, nodes are attached to patients and should have the ability to handle different types of data transmission. Forwarding critical data to the medical surveillance center must be highly reliable.

The underlying aim of this research is therefore to propose an enhancement to an incentive-based routing scheme for non-cooperative heterogeneous mobile wireless sensor networks by using reinforcement learning (RL) algorithm, called Q-learning, in comparison to an existing scheme which has been used to deal non-cooperative heterogeneous mWSNs, called the continuous value cooperation protocol (CVCP) algorithm. The heterogeneity studied in this research covered two aspects, *i.e.*, heterogeneity in terms of traffic or message classes present in the network and heterogeneity in terms of node processing rate capabilities.

The experiments results showed that proposed RL algorithm can outperform existing CVCP algorithms in terms of normalized average reward by up to 14%.

However, the percentage of node processing rate did not depend on any algorithm but only on the proportion of nodes of each type of node processing rate. Such result suggests that the advantage of the proposed method ensures a certain degree of fairness in node selection while maintaining the advantage of achieving higher normalized average reward that the CVCP method. Therefore, the heterogeneity in node processing rate did not significantly affect our experiment results. However, in presence of diverse message class heterogeneity in the network, RL consistently gained 2-14% of normalized average reward, depending on the reward regime of the message classes, over the original CVCP method. The results in our experiment suggest that RL can be applied to improve cooperation among routing nodes in comparison to an existing incentive-based algorithm like CVCP.

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 advisor, 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.

In addition, I am grateful for the lecturers in School of Telecommunication Engineering for their suggestion and all their help. I would also like to express my thanks to Assoc. Prof. Dr. Mohd Fadlee A Rasid of Universiti Putra Malaysia, Malaysia, for granting me the opportunity to do research in Malaysia.

I would also like to thank Asst. Prof. Dr. Piyaporn Krachodnok and Asst. Prof. Dr. Peerapong Uthansakul for accepting to serve in my committee.

My sincere appreciation goes to Ms. Pranitta Arthans for her 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.

Chanon Rittong

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

Figure **Page**

LIST OF FIGURES (Continued)

Figure **Page**

SYMBOLS AND ABBREVIATIONS

WSNs	=	Wireless sensor networks
mWSN	=	Mobile wireless sensor network
MANETs	=	Mobile Ad-hoc Networks
ARAMA	=	Ant routing algorithm for mobile ad-hoc networks
CVCP	=	Continuous Value Cooperation Protocol
DVCP	=	Discrete Value Cooperation Protocol
RL	=	Reinforcement learning
I_v	=	Vital credits
NT	=	Network traffic load
P	=	Message priority level
C	=	Criticality of the routing device
MDP	=	Markov decision process
t	=	Time step index
α	=	Learning rate
S_t	=	State of the process at time t
S	=	State space
s	=	Current state
s'	=	Next state
A	=	Action space

SYMBOLS AND ABBREVIATIONS (Continued)

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
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 associates to a state-action pair (s,a) at time t
R_t	=	Expected discounted return of the agent at time t
$E^{\pi}[\cdot]$	=	Expectation operator under 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^*
$Q^*(s,a)$	=	The action-value function of a given optimal policy π^* associates to state-action pair (s,a)
i	=	Class of message

SYMBOLS AND ABBREVIATIONS (Continued)

n	=	State number
I_{store}	=	Stored credits
$T_{current}$	=	Current time
T	=	Time period
$r_{average}$	=	Average reward
λ_i	=	Message arrival rate of message class *i*
μ_i	=	Message departure rate of message class *i*
B_j	=	Rejection probability of message class *i*
r_i	=	Reward of message in class *i*
r_{accept_all}	=	Average reward incurred when all message classes can be accepted with no capacity constraints
r_{norm}	=	Normalized average reward
SR	=	Success ratio of the message delivery
RS	=	Reward scheme
L, M, N	=	Constants that can be chosen to emphasize certain factors in vital credits
λ_{state}	=	State dependent node capacity usage rate

CHAPTER I

INTRODUCTION

This chapter introduces a background on routing problems in non-cooperative heterogeneous mobile wireless sensor networks and highlights the significance of improving router cooperation in such networks. It also presents the motivation for applying reinforcement learning which can provide a good routing solution which is the main focus of this thesis.

1.1 Significance of the Problem

In recent years, wireless sensor networks (WSNs) are used in many applications (Romer and Mattern, 2004; Bonivento *et al*., 2006; Zhang *et al*., 2008) such as military applications, smart home, environment monitoring, inventory tracking as well as industrial sensors and healthcare monitoring. Healthcare monitoring is an interesting research that poses challenges in our daily life. Due to the growing number of population, senior people and people with disability are also on the rise, aggravated by the dramatic increase of healthcare costs, a new technology such as wireless sensors which are attached to patients requiring close care may help limit costs and human resources. Such healthcare monitoring wireless sensors must cover both indoor and outdoor areas such as in their homes, hospitals, nursing homes, or even in public areas like parks and supermarkets. For example, (Baldus, *et al*., 2004) proposed the use of wireless sensors to monitor vital signs of patients in a hospital environment.

A WSN usually consists of numerous sensor nodes deployed in the area of interest. Each node is able to collect and process data with neighboring devices. There are many reasons for its popularity, including low costs, flexibility and ease of deployment. However, WSNs have some constraints, such as limited power supply, storage, bandwidth and computation capability. Such constraints combined with a typical deployment of large number of sensor nodes have posed may challenges to the design and management of sensor networks. These challenges necessitate energy-awareness at all layers in the networking protocol stack. At the network layer, the aim is to set up energy-efficient routes and reliably relay data from sensor nodes to the sink so that the lifetime of the network is maximized. There are many researches which aim at solving these routing problems in WSNs (Wanming *et al*., 2007; Liu *et al*., 2008; Wanzhi *et al*., 2008; Chunping and Wei, 2009).

Most of current researches assume WSNs to be stationary and homogeneous. A wireless sensor network is said to be homogeneous if its sensors have the same storage, processing power, battery power, sensing and communication capabilities (Koucheryavy and Salim, 2009; Puccinelli and Haenggi, 2009). However, in some scenarios WSNs must be mobile and may even have heterogeneous sensor nodes. In many prototype systems available today, sensor networks consist of a variety of different devices. Nodes may differ in the type and number of attached sensors. Some nodes may be computationally more powerful than others and thereby collect, process, and route sensory data from many more limited sensing nodes. Some sensor nodes may be equipped with special hardware such as a GPS receiver (Bevly *et al*., 2006) to act as beacons for other nodes to infer their location. Some nodes may act as gateways to long-range data communication networks (*e.g.*, GSM networks, satellite networks, or the Internet). The degree of heterogeneity in a sensor network is an important factor since it affects the complexity of the software executed on the sensor nodes and also the management of the whole system. Apart from sensor node heterogeneity, sensor nodes are mobile in many applications therefore creating a mobile wireless sensor network (mWSN). For instance, for wild life monitoring, sensor nodes are cast into the region of interest as well as equipped on animals to be monitored. The self-organized WSN is mobile as animals move around. In a telemedicine application (Field, 1996), sensor nodes attached to moving patients also form a mWSN.

In mobile wireless networks, such as mobile ad hoc networks, path breakage occurs more frequently due to channel fading, shadowing, interference, node mobility as well as power failure. When a path breaks, rerouting should be carried out promptly to avoid packet loss and large delay.

One main reason why mWSNs immediately resemble mobile ad hoc networks is because both are distributed wireless networks (*i.e.*, there is not a significant network infrastructure in place) and the fact that routing between two nodes may involve the use of intermediate relay nodes (also known as multi-hop routing). Besides, there is also the fact that both ad hoc and sensor nodes are usually battery-powered and therefore there is a major concern on minimizing power consumption. Both networks use a wireless channel placed in an unlicensed spectrum that is prone to interference by other radio technologies operating in the same frequency. Recent advances in WSNs have led to many new protocols specifically designed for sensor networks where energy awareness is an essential consideration. Most of the attention, however, has been given to the routing protocols since they might differ depending on the application and network architecture. Routing in mWSNs is very challenging due to several characteristics that distinguish them from contemporary communication and mobile ad hoc networks (MANETs).

First of all, it is not possible to build a global addressing scheme for the deployment of sensor nodes in mWSNs whereas MANETs can. Therefore, classical IP-based protocols cannot be applied to mWSNs like MANETs. However, there is an ongoing development standard from IETF for mWSNs which defines encapsulation and header compression mechanisms that allow IPv6 packets to be sent to and received from over IEEE 802.15.4-based networks called 6LoWPAN (Xin and Wei, 2008).

Secondly, contrary to typical communication networks and MANETs, almost all applications of mWSNs require the flow of sensed data from multiple regions (sources) to a particular sink. In contrast, MANETs can communicate directly with all other devices within its transmission range without a centralized administrator.

Thirdly, the generated data traffic in most WSNs are significantly redundant and are highly correlated since multiple sensors may generate same data with in the vicinity of a phenomenon. Such redundancy may be exploited by the routing protocols to improve energy and bandwidth utilization. By contrast, MANETs consist of standalone nodes communicating with others via multi-hop connection, so there are no data-redundant nodes like mWSNs.

And finally, in mWSNs, sensor nodes have tightly constrained resources in terms of transmission power, on-board energy supply, processing capacity and storage and thus require carful resource management. As for MANETs, the types of nodes include notebooks, handheld PCs, and so on. Thus, each such node has less constraint on energy, processing capability or storage than sensor nodes.

There are many existing researches related to routing in MANETs (Shah *et al*., 2008) and mWSNs (Xiaoxia *et al*., 2008). (Xiaoxia *et al*., 2006) proposed back up nodes and cooperative caching is proposed to enhance the robustness in routing against path breakage in mWSNs. Guangcheng and Xiaodong (Guangcheng and Xiaodong, 2008) proposed an opportunistic routing for mWSNs based on receive signal strength indicator (RO-RSSI). Their approach outperformed traditional TinyAODV (Pham *et al*., 2006) in terms of successful delivery ratio for sparse mWSNs.

Some researches such as Iyengar, Hsiao-Chun *et al*. (Hsiao-Chun *et al*., 2007) investigated routing based on biologically inspired mechanisms and the associated techniques for resolving routing in mWSNs and MANETs, including ant-based and genetic approaches. Hussein and Saadawi (Hussein and Saadawi, 2003) proposed the ant routing algorithm for mobile ad-hoc networks (ARAMA), which is also a biologically-based routing algorithm.

Most routing or packet forwarding schemes in the aforementioned literature assume that nodes function properly, are trustworthy and cooperative. However, in realistic scenarios, nodes may fail to operate due to lack of resources, hardware failure or malicious behaviors. Varshney (Varshney, 2008) proposed a reliable packet forwarding scheme in non-cooperative mWSNs for wireless health monitoring applications with spotty coverage areas. A node cooperation based on earned or offered incentives was proposed to encourage devices cooperate as router thereby improving message reliability.

There are many algorithms which are used to deal with non-cooperative routing in mWSNs. The incentive-based concept has been applied in many algorithms such as reputation-based routing mechanism (Lewis and Foukia, 2008), Nash-Q (Hu and Wellman, 2003), reinforcement learning (RL) (Sutton and Barto, 1998), game theory (Machado and Tekinay, 2008). Varshney (Varshney, 2008) proposed an incentive-based mechanism mWSNs for healthcare monitoring to improve the routing cooperation of mobile wireless sensor nodes which are attached to patients. Forster (Forster *et al.*, 2008) proposed an efficient implementation of reinforcement learning based routing on real mWSNs which consist of ScatterWeb (Schiller *et al.*, 2005) sensor nodes.

So far, the above works assume homogeneous mWSNs where sensor nodes are identical. However, in many applications like healthcare monitoring, sensor nodes are typically heterogeneous. Huang *et al*. (Huang *et al*., 2009) proposed a pervasive secure access to a hierarchical sensor based healthcare monitoring architecture in wireless heterogeneous networks where nodes have different data collection abilities, such as, electrocardiogram (ECG) and body temperature. Similarly, Varshney (Varshney, 2008) proposed heterogeneous sensor nodes in the terms of data gathering which included blood pressure, electrocardiographic activity, pulse, body core temperature, and oxygen saturation as well as alerting (emergency) signals when one or more vital signs exceed some predefined threshold. Jurik and Weaver (Jurik and Weaver, 2008) described heterogeneous sensors as those which come from different shapes and sizes and offering different functionalities and accommodating different constraints. Typical medical applications for sensors include monitoring pulse, temperature, motion acceleration, blood pressure, and pulse oximeter.

Heterogeneity in mWSN is a challenge to all researchers. Many recent literature related to routing assume homogeneous, non-cooperative mWSNs (Munir *et al*., 2007; Agah *et al*., 2004) although many applications require heterogeneous mWSNs. Only Varshney (Varshney, 2008) and Forster *et al*. (Forster *et al*., 2008) considered routing problems in heterogeneous mWSN for routing. The significance and advantages of heterogeneous, non-cooperative mWSNs are that they are more realistic for healthcare monitoring application. This is the motivation for the problem which this thesis aims to solve. The incentive-based concept is the one of the effective tools for solve the routing problem in heterogeneous, non-cooperative mWSNs. Many algorithms such as reputation-based routing mechanism, Nash-Q, reinforcement learning (RL), game theory, are all based on the incentive-based concept. However, reputation-based routing mechanisms are typically used to enhance security in ad-hoc networks by identifying and avoiding malicious nodes in the network. Thus, reputation-based methods may not be suitable for promoting cooperation among nodes.

Game theory has been used extensively to deal with uncooperative wireless sensor networking resource allocation problems (Michiardi and Molva, 2003) where different players may have different strategies to compete for resource usage within the network. Game theory is a formal way to analyze interactions among a group of rational players who behave strategically. A game is the interactive situation, specified by the set of players (*i.e.* sensor nodes), the possible actions of each node, and the set of all possible payoffs. Games in which the actions of the players are directed to maximize the profit without subsequent subdivision of the profit among the player are called cooperative games. In cooperative games, the outcome arises as a result of an agreement among players. These games are compared with respect to the preferred ability of payoffs. In other words, in a cooperative game, different players form alliance with each other in a way to influence the outcome of the game in their favor. Hence, such game is not defined as a game in which players actually do cooperate, but as a game in which any cooperation is enforced by an outside party. Cooperative games have been applied to their wireless sensor networks in (Liqiang *et al*., 2008; Gharehshiran and Krishnamurthy, 2009). In a non-cooperative game, unlike cooperative ones, no outside authority assures that players stick to the same predetermined rules and binding agreements are not feasible. In the early 50's John Nash recognized that in non-cooperative games, there exist sets of optimal strategies (called Nash equilibrium) used by the players in a game such that no players can benefit by unilaterally changing his or her strategy if the strategies of the other players remain unchanged. Felegyhazi *et al*. (Felegyhazi *et al*., 2006) and Chengnian *et al*. (Chengnian *et al*., 2007) proposed non-cooperative games with Nash equilibrium applied to their wireless sensor networks.

Reinforcement learning (RL) is the study of how animals and artificial systems can learn to optimize their behavior in the face of rewards and punishments. Reinforcement learning algorithms have been developed to approximate solutions to problems that are closely related to dynamic programming, which is a general approach to determine optimal control in a sequential decision problem. Reinforcement learning phenomena have been observed in psychological studies of animal behavior, in neurobiological investigations and some works applied to WSNs (Egorova-Forster and Murphy, 2007; Shah and Kumar, 2007; Busoniu *et al*., 2008). In the terminology of RL, the network represents the environment whose state is determined by the number and relative position of sensor nodes, the status of links between them and the dynamics of packets. The destination of handled packets and the status of local links form the sensor node's observation. Each node is an agent who has a choice of actions. It decides where to send the packet according to a policy. A reinforcement learning method, called Q-learning, which directly approximates the optimal action-value function (Q-value), is commonly applied in the literature. Each learning agent takes an action, receives a reward, updates local information with input from the environment, and repeats the process by learning its own optimal strategy. RL has low complexity and computational requirement and no limitation on the number of agents (sensor nodes). However, RL requires training time during the learning curve in order to learn to optimize the agent's behavior. It also requires a certain amount of memory usage to store the Q-values for the learning process.

Nash-Q is an algorithm which is a mixture of game theory and reinforcement learning. Nash-Q uses the framework of a general sum stochastic game, whereby each agent's reward depends on the joint action of all agents and the current state of the environment. The agent attempts to learn its Nash equilibrium Q-values, which are defined by the Q-values received in Nash equilibrium. Moreover, the agent not only learns to find its own optimal policy, but it also learns actions and rewards of the other agent to find the other agent's optimal strategy. Therefore, each agent acts rationally with respect to this expectation and eventually fairness can be achieved. However, the theory and convergence proof of Nash-Q applies to two players only. As a result, this algorithm is inappropriate for mWSNs which typically consist of many sensor nodes. However, there is only one ongoing work which applied Nash-Q to enhance packet forwarding in non-cooperative multi-domain wireless sensor networks controlled by two different authorities. In Nash-Q, the computational complexity is high and global network information (*i.e.* information on the other player) is required.

The similarities of game theory and RL can summarized as follows. Firstly, both game theory and RL can support a large number of players (Minh Hanh and Krishnamurthy, 2005). Secondly, both game theory and RL can be applied to both centralized and distributed operations. However, several key differences between game theory and RL are as follows. First, if we consider in terms of opponent players, RL does not have competitors like game theory. The reason is because, in RL, we consider the agent's self-interest and the surrounding agents as the environment, whereas game theory requires knowledge of the other players in the game. Second, RL is more robust than game theory in changing environments. However, the long-term optimality of the behavior or strategy depends on whether the environment is static or dynamic. RL is more suitable in a dynamic environment scenario such as in mWSNs where nodes can move around. Game theory, on the other hand, is more suitable for a static environment where strategies can be determined by exhaustive search from each possible situation. Third, game theory can achieve the optimal strategy while RL may only achieve near-optimality. This is because the agent in RL only requires knowledge of local information of neighboring agents whereas game theory requires knowledge the global information from all players in the network. Fourth, RL needs more training time than game theory because the agents in RL must take time to learn good behaviors.

Of the above algorithms, a comparison can be made to determine their suitability to the routing problem in heterogeneous, non-cooperative mWSNs.

Although, both game theory and RL have the ability to cater a large number of players, distributed operation with reasonable memory requirement, game theory requires knowledge of the all other opponent’s strategies. Hence, game theory may not be as scalable as RL, especially in a dynamic environment such as in mWSNs because RL requires only local information from the neighbor nodes. Xuedong, Balasingham *et al*. (Balasingham *et al*., 2008) and Ping and Ting (Ping and Ting, 2006) used RL to solve routing problems in static WSNs. However, to the best of our knowledge, there are no works which proposed RL to promote cooperative routing in mWSNs yet. Reinforcement learning (RL) therefore warrants further investigation for its potential use for routing in heterogeneous, non-cooperative mWSNs.

Therefore, the underlying objective of the thesis proposal is to solve the routing problem for heterogeneous, non-cooperative mWSNs using a scalable, distributed incentive-based mechanism with reasonable resource requirements such as RL. We also study their effects on the efficiency in heterogeneous, non-cooperative mWSN and propose a good-optimal routing strategy under energy-constrained conditions.

1.2 Research Objectives

1. To study routing problems in heterogeneous, non-cooperative mWSNs.
2. To apply RL to solve the routing problem in heterogeneous, non-cooperative mWSNs and compare with other existing incentive-based routing algorithms.

1.3 Research Hypothesis

1. RL can provide a good routing solution in heterogeneous, non-cooperative mWSNs.
2. In realistic applications, many types of sensor nodes will be used. We therefore consider heterogeneous sensor nodes.
3. Some sensor nodes are uncooperative due to various reasons, *e.g.*, nodes may drop packets from other nodes in order to conserve their energy.

1.4 Basic Agreements

1. Visual C++ was used to simulate the routing protocols in heterogeneous, non-cooperative mWSNs.
2. Some data in the experiments were normalized to facilitate analysis and obtain a conclusion.

1.5. Scope and Limitation

1. Heterogeneous, non-cooperative mWSNs were studied to model to realistic applications.
2. Incentive-based methods for choosing a good routing strategy in heterogeneous, non-cooperative mWSNs were studied.
3. RL methods were studied and compared with the Continuous Value Cooperation Protocol (CVCP) for the good routing strategy in heterogeneous, non-cooperative mWSNs.
4. Simulations were carried out by Visual C++. The experimental results were analyzed to find a good routing strategy under energy constraints.

1.6. Research Procedures

1.6.1 Progressions

1. Review of literature and related theories.
2. Study the existing routing methodologies in non-cooperative heterogeneous mobile wireless sensor networks and their effects.
3. Test the proposed RL algorithm by simulation using Visual C++ to solve routing problems in mWSNs.
4. Analyze and conclude results.
5. Prepare publication.
6. Write thesis.

1.6.2 Research Methodology

Objective 1: To study routing problems in heterogeneous, non-cooperative mWSNs.

1. Review literature and related works about routing in heterogeneous, non-cooperative mWSNs.
2. Determine the advantages and disadvantages of the routing methods chosen as benchmark for this thesis.
3. Apply simulation tools such as Visual C++ to evaluate routing non-cooperative, heterogeneous mWSNs under special conditions.
4. Design the experiment scenario to compare with existing incentive-based algorithm (Varshney, 2008) which is used of incentives called vital credits (I_v)

$$I_v = NT^L \times P^M \times C^N$$

where *L, M* and *N* represent constant that can be chosen to emphasize certain factors in vital credits. This vital credit is the function of network traffic load (*NT*), message priority level (*P*) and criticality of the routing device (*C*). Two algorithms have been proposed in Varshney (Varshney, 2008) work namely the Continuous Value Cooperation Protocol (CVCP) and the Discrete Value Cooperation Protocol (DVCP).The difference between CVCP and DVCP was the network size, *i.e.*, CVCP was designed for large networks while DVCP was designed for smaller networks.

5. Under various network scenarios, we measured the following parameters to evaluate the performance of CVCP: the average normalized reward, success ratio, and percentage of node processing rate.

<u>Objective 2:</u> To apply RL to solve the routing problem in heterogeneous, non-cooperative mWSNs and compare with other existing incentive-based routing algorithms.

1. Survey various RL methods and type of RL which are suitable for heterogeneous, non-cooperative mWSNs. RL can be typically classified into 3 types: Actor Only methods (Vazquez-Abad and Krishnamurthy, 2002) which learning rates are slow but with performance improvement guaranteed; Critic Only methods (Makarevitch, 2000) which learning rates are fast with performance improvement not always guaranteed; Actor-Critic

(Usaha and Barria, 2007) which learning rates are fast with performance improvement guaranteed.

2. Implement the selected RL method for heterogeneous, non-cooperative mWSNs and compare with CVCP algorithms.
3. Compare performance metrics of RL algorithm procedure with CVCP incentive-based algorithm by considering the following parameters, the average normalized reward, success ratio, and percentage of node processing rate.

1.6.3 Research Location

1. Wireless Communication Research and Laboratory, Factory Building 4 (F4) 111 University Avenue, Muang District, Nakhon Ratchasima 30000, Thailand.
2. Computer and Communication Systems Engineering, Faculty of Engineering, University Putra Malaysia, 43400 Serdang, Selangor Darul Ehsan, Malaysia.

1.6.4 Research Equipments

1. Personal Computer
2. Visual C++ software

1.6.5 Data Collection

1. Information collected by reviewing literatures and related works.
2. Data collected from Visual C++ simulations.

1.6.6 Data Analysis

The simulation collected data from the sensor node were analyzed, compared and concluded in terms of graphs and tables.

1.7 Expected Benefit

1. A good routing strategy for non-cooperative heterogeneous mobile wireless sensor networks.
2. Improved routing reliability in non-cooperative, heterogeneous mobile wireless sensor networks.

1.8 Organization of Thesis

The remainder of this thesis is organized as follows. **Chapter 2** presents the theoretical background which underlies the contribution of this thesis. Firstly, an introduction of related works followed by the introduction of Markov Decision Process theory, the birth and death process and reinforcement learning (RL). Finally, the basic theory of Q-learning is presented which is the RL tool used to enhance routing cooperation in this thesis.

In the first part of **Chapter 3**, we studied the existing algorithm CVCP and formulated the Q-learning algorithm to evaluate the routing performance results in homogeneous mWSNs. The Q-learning and CVCP tools were compared in terms of the average normalized reward, success ratio, and percentage of node processing rate. The advantages and disadvantages of these two algorithms were then explained. In the latter part of the chapter, routing cooperation in non-cooperative heterogeneous mWSNs was presented. The routing performance results were evaluated and compared between the CVCP and the Q-learning algorithms.

Chapter 4 This chapter summarizes all findings and original contribution in this thesis and points out possible future research directions.

CHAPTER II

BACKGROUND THEORY

2.1 Introduction

In this thesis, we study incentive-based routing for non-cooperative heterogeneous mobile wireless sensor networks (mWSNs). Typically, wireless sensor networks contain of a large number of sensor nodes that are deployed in the interested area. Sensor nodes may differ in types thus creating a heterogeneous WSN. These nodes may or may not cooperate with each other in terms of routing messages for one another due to several reasons as presented in the previous chapter. Furthermore, these sensor nodes may be able to move around as they are attached to the observation object such as human or animals. The routing problem in mWSNs is the one of an important issue required to send messages reliably through the network. Therefore, the main focus in this thesis is to investigate means to enhance routing cooperation among heterogeneous sensor nodes in mWSNs.

This thesis proposed the application of reinforcement learning (RL) to address the issue of incentive-based routing for non-cooperative heterogeneous mWSNs. Reinforcement learning (Sutton and Barto, 1998) is the study of how animals or machines can learn to optimize their behavior to obtain rewards and to avoid punishments. This learning scheme can permit a decision maker to learn its optimal decisions (actions) through series of trial-and-error interactions with a dynamic environment. Its main idea is to reinforce good behaviors of the decision maker while discouraging bad behaviors through a scalar reward value returned by the environment. RL relies on the assumption that the dynamics of the system satisfies a Markov decision process (MDP).

Q-learning (Watkins, 1989) is a reinforcement learning technique that approximates the optimal action-value function which is a function that gives the expected reward for taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment.

Therefore, this chapter introduces the basic theory of the reinforcement learning. It also serves as an introduction to Q-learning algorithm which is the basis of this thesis. The next section provides a background theory of Markov decision process (MDP), followed by the birth-death process, reinforcement learning (RL) and its elements. A summary is presented in the final section.

2.2 Markov Decision Process Theory

Markov decision processes (MDPs) is a model of a decision-maker interacting synchronously with the environment. Since the decision-maker sees the environment's true state, it is referred as a completely observable Markov decision process. The basis of Markov decision process is presented as follows.

2.2.1 Markov Property

Markov property refers to the memory-less property of a stochastic process. A stochastic process has the Markov property if the conditional probability distribution of future states of the process depends only upon the present state, not on the sequence of events that preceded it. A process with this property is called a *Markov* *process*. The Markov property states that anything that has happened so far can be summarized by the current state S_t. Therefore, the probability of being in the next state at time t+1 based on the past history of state changes can be defined simply as the conditional probability based on the current state at time t by;

$$P(S_{t+1} = s_{t+1} \mid S_t = s_t, \ldots, S_0 = s_0) = P(S_{t+1} = s_{t+1} \mid S_t = s_t). \tag{2.1}$$

This equation is referred to as the Markov property. In other words, a stochastic process has Markov property if the probability distribution of future states of the process time t+1, given the present state at time t and all past states, depends only upon the present state and not on any past states.

2.2.2 Markov Decision Process

The probability that the process chooses s' as its new state is influenced by the chosen action. Specifically, it is given by the state transition probability function. Thus, the next state s' depends on the current state s and the decision maker's action a. But given s and a, it is conditionally independent of all previous states and actions. In other words, the state transitions of an MDP possess the *Markov property*. This state transition probability function equation is defined by;

$$P(s' \mid s, a) = P(S_{t+1} = s' \mid S_t = s, a_t = a). \tag{2.2}$$

Similarly, given any current state and action, s and a, together with any next state, s', the expected value of the incurred reward is;

$$R(s, a, s') = E[r_{t+1} \mid S_t = s, a_t = a, S_{t+1} = s'] \tag{2.3}$$

where $E[.]$ is the expectation operator and r_{t+1} is the reward received at time $t+1$. Equation (2.2) and (2.3), completely specify the most important aspects of the dynamics of the MDP. The simulation programming requires the exact knowledge of these two functions in order to determine the optimal policy. A MDP model can be shown in Fig. 2.1.

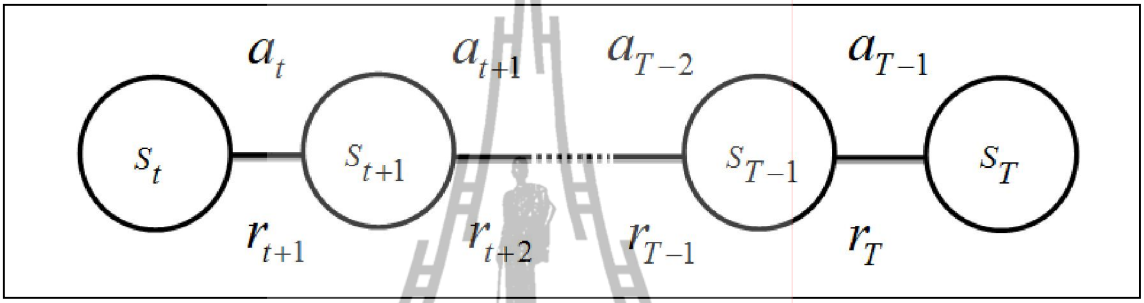

Figure 2.1 A MDP model.

A Markov decision process is a 4-tuple (*S, A, P, R*) which can describe the MDP characteristics, where *S* denotes the set of states, *A* is a finite set of actions, *P* is the probability that action *a* in state *s* at time *t* will lead to state *s'* at time *t + 1*, *R* is the immediate reward (or expected immediate reward) received after transition to state *s'* from state *s* after having taken action $a \in A$. Let $P(s' \mid s, a) \in P$ be the state transitioning model that denotes the probability of transiting to the next state $s' \in S$ after an agent takes action $a \in A$ at the current state $s \in S$.

2.2.3 Policy

A policy, π is a description of the behavior of a decision-maker, or a function mapping states to actions, π: *S* →*A.* There are two types of policies. A *stationary policy* is a situation-action mapping, *i.e.*, it specifies an action to be taken at each state. The choice of action depends only on the state and is independent of the time step. A *non-stationary policy*, on the other hand, is a sequence of situation-action mappings, indexed by time. In this thesis, we focus on stationary policies since our data acquisition problem is based on models of sensor readings which are obtained in a particular time frame, such as in the mornings, afternoons, etc. Hence, within such period, the model maybe considered stationary hence the policy is also assumed stationary.

The objective of solving a MDP is to find a policy, π, defined as a mapping of the state space to the action space, $\pi : S \rightarrow P[A]$, 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 a state-action pair (s,a) with an expected reward for performing action *a* in state *s* at time step *t* and policy π.

To achieve this objective, particularly in scenarios where the dynamics of the environment is difficult to model (such as in mWSNs), a technique called reinforcement learning can be used to solve MDPs.

2.3 Reinforcement Learning

Reinforcement learning (RL) is a computational approach which is concerned with how an agent ought to take actions in an environment so as to maximize some notion of cumulative reward. In machine learning, the environment is typically formulated as a Markov decision process (MDP), and many reinforcement learning algorithms for this context are highly related to dynamic programming techniques. The main difference from these classical techniques is that reinforcement learning algorithms do not need the knowledge of the MDP and they target large MDPs where exact methods become infeasible. The learner is not taught which action to take, as in most forms of machine learning, but instead must discover which actions yield the most reward by trial-and-error interactions with its environment (Sutton and Barto, 1998).

A reinforcement learning agent interacts with its environment in discrete time steps. At each time *t*, the agent receives an observation, which typically includes the reward r_t. It then chooses an action a_t from the set of actions available. The environment then moves to a new state s_{t+1} and the reward r_{t+1} associated with the transition (s_t, a_t, s_{t+1}) is determined. The goal of a reinforcement learning agent is to collect as much reward as possible. Figure 2.3 shows the agent-environment interaction in reinforcement learning.

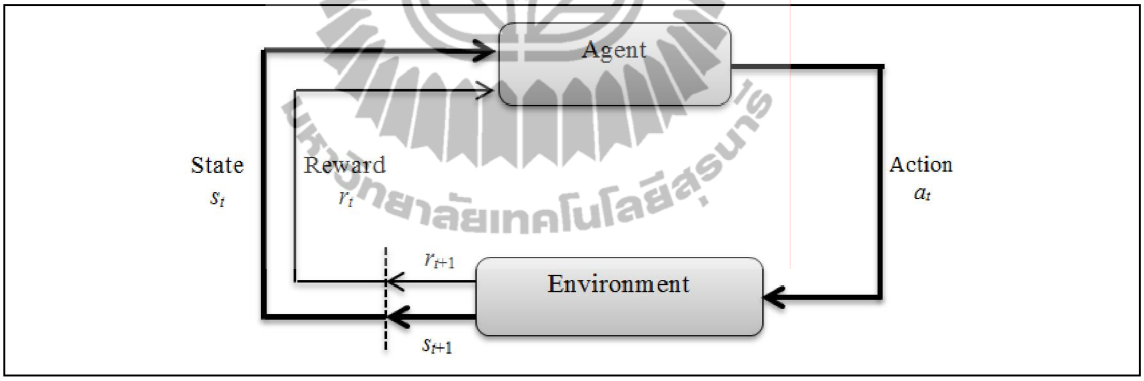

Figure 2.2 Diagram of agent-environment interaction in reinforcement learning.

2.3.1 The Value Function

Define the value function $V^{\pi}(s)$ of a policy π by;

$$V^{\pi}(s) = E^{\pi}\left[R_t \mid s_t = s\right]$$

$$= E^{\pi}\left[\sum_{k=0}^{\infty}\beta^{k} r_{t+k+1} \mid s_t = s\right] \tag{2.4}$$

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 which $0 \le \beta \le 1$ and $E^{\pi}[\cdot]$ is the expectation operator under policy π. Similarly, the action-value function $Q_t^{\pi}(s,a)$ of a given policy π associates a state-action pair (s,a) with an expected reward for performing action a in state *s* at time step *t* and following π thereafter;

$$\begin{aligned} Q_t^{\pi}(s,a) &= E^{\pi}\left[R_t \mid s_t = s, a_t = a\right] \\ &= E^{\pi}\left[\sum_{k=0}^{\infty}\beta r_{t+k+1} \mid s_t = s, a_t = a\right]. \end{aligned} \tag{2.5}$$

2.3.2 The Optimal Value Function

Solving a reinforcement learning task means, roughly, finding a policy that achieves the maximum reward over the long run. The optimal value function denoted as $V^*(s)$ which is defined as the maximum state value function over all possible policies, at state *s*.

$$V^*(s) = \max_{\pi} V^{\pi}(s). \tag{2.6}$$

Optimal policies also share the same optimal action-value function, denoted $Q^*(s)$, and defined by;

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

The standard solution to the problem above is through an iterative search method (Puterman 1994) that searches for a fixed point of the following *Bellman* equation;

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

The equation (2.9) is a form of the Bellman optimality equation for $V^*(s)$. The Bellman optimality equation for $Q^*(s)$ is;

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

2.4 Q-learning

Q-learning is a reinforcement learning technique that works by learning an action-value function that gives the expected utility of taking a given action in a given state and following a fixed policy thereafter. One of the strengths of Q-learning is that it is able to compare the expected utility of the available actions without requiring a model of the environment. Q-learning (Sutton and Barto, 1998) defines a learning method within a MDP that is employed in single-agent RL systems. 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. Assume that the learning agent exists in an environment described by some set of possible states $s \in S$. It can perform any of the possible actions $a \in A$. The interaction between the agent and the environment at each instant consists of the following sequence;

- The agent senses the state $s_t \in S$.
- Based on s_t, the agent performs an action $a_t \in A$.
- As a result, the environment makes a transition to the new state $s_{t+1} = s' \in S$.
- The agent receives a real-valued reward (payoff) r_t that indicates the immediate reward value of this state-action transition.

The task of the agent is to learn a policy, $\pi : S \rightarrow A$, for selecting its next action $a_t = \pi(s_t)$ based only on the current state s_t. For a policy π, the Q-value $Q^{\pi}(s,a)$ (or state-action value) is the expected discounted cost for executing action a at state s and then following policy π thereafter. The optimal policy $\pi^*(s)$ is the policy that maximizes the total expected discount reward which received over an infinite time. The Q-learning process tries to find $Q^*(s,a) = Q^{\pi^*}(s,a)$ in a recursive manner using available information (s_t, a_t, s', a', r_t) where s_t and s' are the states at time t and $t+1$ respectively, a_t and a' are the actions at time t and $t+1$, respectively, and r_t is the immediate reward due to a_t. The Q-learning rule at time step $t+1$ is given by;

$$Q_{t+1}(s_t, a_t) = (1-\alpha)Q_t(s_t, a_t) + \alpha \left[r_t + \beta \max_{a'} Q_t(s', a') \right] \tag{2.10}$$

where $0 \le \beta \le 1$ is a discount factor, $0 \le \alpha \le 1$ is the learning rate and $Q_t(s', a')$ is the action-value function for next state s' and next action a'.

2.4.1 Exploration

One of the most important issues for Q-learning algorithm is maintaining a balance between exploration and exploitation. Normally, the convergence theorem of Q-learning requires that all state-action pairs (s,a) are tried infinitely (Sutton and Barto, 1998). Such a balanced condition is satisfied by selecting a good action according to some probability ε and exploring new actions, otherwise. Note that ε is the probability that a greedy action is selected *i.e.*;

$$a^* = \arg\max_{\forall a \in A} Q(s,a). \tag{2.11}$$

This probability termed $\varepsilon - greedy$, significantly speeds up the convergence of the Q-value function. If the Q-value of each admissible (s,a) pair is visited infinitely often, and if the learning rate is decreased to zero in suitable way, then as $t \to \infty$, $Q_t(s,a)$ converges to $Q^*(s,a)$ with probability 1 (Sutton and Barto, 1998). The optimal policy is defined by;

$$\pi^*(s) = \arg\max_{a \in A(s)} Q^*(s,a). \tag{2.12}$$

2.5 Summary

In this chapter, an overview of Q-learning which is a reinforcement learning method has been introduced. Furthermore, we also provided a concise background on theories related to reinforcement learning including the Markov decision process. In the next chapter, an incentive-based routing mechanism proposed for non-cooperative homogeneous and heterogeneous mobile wireless sensor networks using Q-learning will be presented and its routing performance compared with an existing algorithm.

CHAPTER III

INCENTIVE-BASED ROUTING FOR NON-COOPERATIVE MOBILE WIRELESS SENSOR NETWORKS

3.1 Introduction

A wireless sensor network (WSN) usually consists of numerous sensor nodes deployed in the area of interest. Each node is able to collect and process data with neighboring devices. There are many reasons for its popularity, including low costs, flexibility and ease of deployment. However, WSNs have some constraints, such as limited power supply, storage, bandwidth, and computation capability. Such constraints combined with a typical deployment of large number of sensor nodes have posed may challenges to the design and management of sensor networks. These challenges necessitate energy awareness at all layers of networking protocols stack. At the network layer, the aim is to set up energy-efficient routes and reliably relay data from sensor nodes to the sink so that the lifetime of the network is maximized. There are many researches which aim at solving these routing problems in WSNs.

Most current researches assume WSNs to be stationary. However, in many scenarios WSNs must be mobile. For instance, for wild life monitoring, sensor nodes are cast into the region of interest as well as equipped on animals to be monitored. The self-organized WSN is mobile as animals move around. In a telemedicine application (Field, 1996) sensor nodes attached to moving patients also form a mobile WSN (mWSN).

Furthermore, most routing schemes assume that nodes function properly, are trustworthy and cooperative. However, in realistic scenarios, nodes may fail to operate due to lack of resources, hardware failure or malicious behaviors. There are many algorithms which are used to deal with non-cooperative routing in mWSNs. The incentive-based concept has been applied in many algorithms such as reputation-based routing mechanism (Lewis and Foukia, 2008) Nash-Q (Hu and Wellman, 2003) reinforcement learning (RL) (Sutton and Barto, 1998) Game theory (Machado and Tekinay, 2008).Nodes decide whether to cooperate or not based on incentives stored or earned. Varshney (Varshney, 2008) proposed an incentive-based mechanism called continuous value cooperation protocol (CVCP) for healthcare monitoring to improve the routing cooperation of mobile wireless sensor nodes which are attached to patients. Forster, Murphy *et al.* (Forster *et al.,*2008) proposed an efficient implementation of RL-based routing on real mWSNs.

Routing related literature mostly assume homogeneous mWSNs where sensor nodes are identical. For instance, assume homogeneous, non-cooperative mWSNs (Munir *et al.,* 2007; Agah *et al.,* 2004). However, in many applications like healthcare monitoring, sensor nodes are typically heterogeneous. Heterogeneity in mWSN is a challenge. This is the motivation for the problem which this thesis aims to solve.

The incentive-based concept is the one of the effective tools for solving the routing problem in non-cooperative mWSNs. Reputation mechanisms are typically used to enhance security by identifying and avoiding malicious nodes, but not promote node cooperation. Game theory requires knowledge of the other opponents' strategy, thereby may not be scalable especially in dynamic environments as mWSNs. On the other hand, RL can cater a large number of nodes with distributed operation using only local information from the neighboring nodes.

In this chapter, we apply a RL method called Q-learning to promote packet forwarding in a periodic sleep cycle *homogeneous* and *heterogeneous* mWSN. We compare its performance with an existing sleep cycle incentive-based routing algorithm (Varshney, 2008) under various message arrival rates and traffic scenarios and node processing capability.

Therefore, the underlying objective of this chapter is to show that RL can be applied to enhance the routing problem for non-cooperative homogeneous and heterogeneous mWSNs in comparison with the existing CVCP routing algorithm.

This chapter is focused on the following issues:

1. The formulation of the packet forwarding problem under the RL framework in non-cooperative mWSNs.
2. The simulation of RL and the existing CVCP algorithm in non-cooperative mWSNs.
3. The comparison of performance between the proposed RL algorithm and the existing CVCP algorithm.

The rest of this chapter is organized as follows. Section 3.2 describes the CVCP algorithm followed by RL and Q-learning in sections 3.3 and 3.4, respectively. In section 3.5, we formulate the packet forwarding problem using the RL framework using CVCP. We then present a homogeneous mWSN experiment with its simulation results and conclusion in section 3.6. Finally, section 3.7 presents a heterogeneous mWSN experiment with its simulation results and conclusion.

3.2 Continuous Value Cooperation Protocol (CVCP)

The continuous value cooperation protocol (CVCP) has been proposed to promote router cooperation in ad hoc networks deployed to supplement infrastructure-oriented wireless health monitoring systems (Varshney, 2008). The protocol used an incentive called vital credit which is a function of message priority level (P), network traffic loads (NT), and criticality of the message delivery (C). Vital credits (I_v) are defined as:

$$I_v = NT^L \times P^M \times C^N \tag{3.1}$$

where *L*, *M* and *N* represent constants that can be chosen to emphasize certain factors in vital credits. For example, the message priority level may be assigned to nodes transmitting emergency messages or alerts in such a way that vital credits are greater than symptoms monitoring message to encourage delivery. The network traffic level may depend on the frequency of monitoring, the number of packets per message, and the number of monitored patients. The node criticality may rely on the location of the routing node and routing scheme.

Figure 3.1 shows an individual routing node using CVCP to decide whether to forward a particular message to a destination node based on the number of vital credits offered by the source node, and the routing node’s already earned vital credits. The source device uses an *incentive estimator* to determine the vital credits it will offer to a routing node to forward its message. The routing node stores already earned vital credits. If the offered vital credits exceed its stored credits, the more likely a routing node will cooperate. On the other hand, a routing node with a large number of stored credits might not cooperate even if the offered number of credits is high. If a routing node decides to cooperate, it receives the offered vital credits from the source node and adds them to its stored credits.

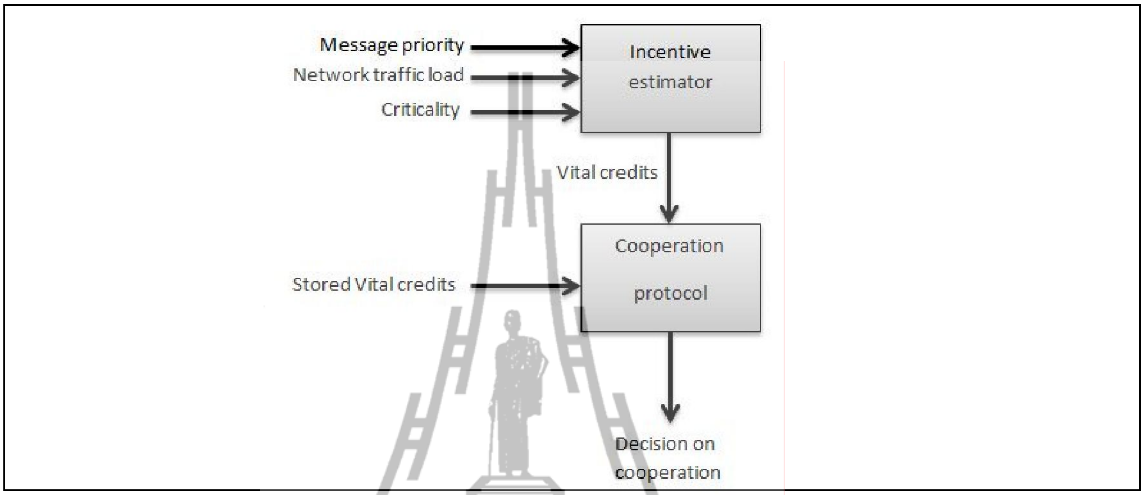

Figure 3.1 Cooperation protocol (Varshney, 2008) in which a routing node makes a decision based on vital credits the source node offers and its stored credits.

Nodes with the most vital credits can receive higher sleep-cycle priority, thereby promoting energy saving. However, nodes that have used up their vital credits for a recent sleep cycle are more likely to cooperate to increase their earned vital credits. Figure 3.2 depicts the CVCP vital credit checking procedure at a routing node. Furthermore, a routing node also checks whether a sleep cycle will be initiated soon and opts to cooperate accordingly.

Furthermore, apart from checking the offered and stored credits and sleep cycles, decisions to cooperate or not may also be dependent on state conditions of the routing node other than shown in Figure 3.2. For instance, different states of network loads or residual battery levels may provide different decisions in order to achieve an optimal long term benefit for a particular routing node. A scalable, distributed self-learning scheme with reasonable computation requirements described in the next section warrants potential use for finding long term benefit decisions at each routing node in a mWSN.

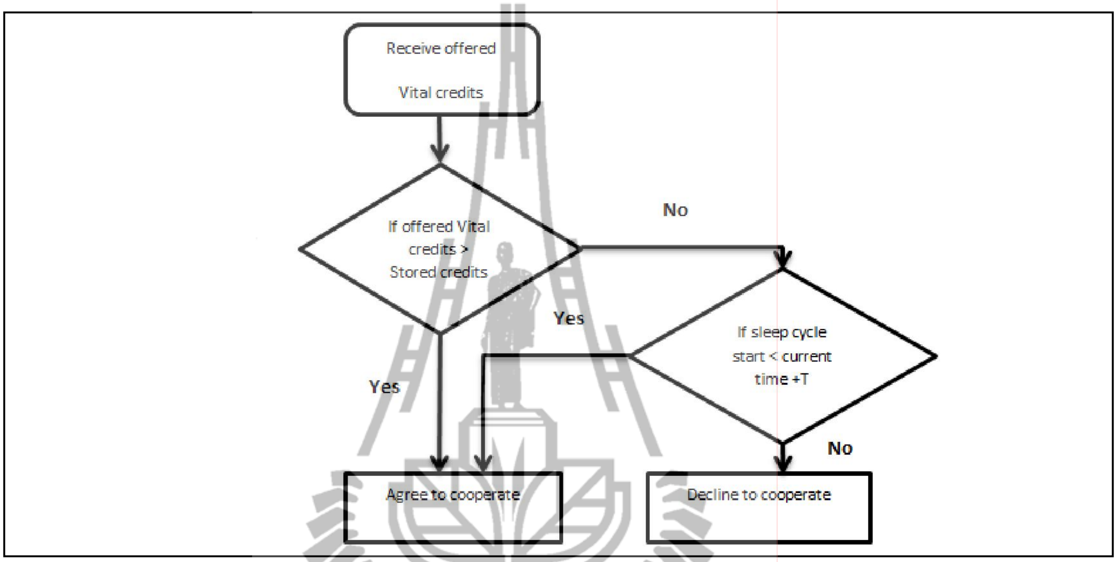
Receive offered
Vital credits
If offered Vital credits > Stored credits
No
Yes
If sleep cycle start < current time +T
Yes
No
Agree to cooperate
Decline to cooperate

Figure 3.2 Diagram of CVCP checking procedure performed at routing node (Varshney, 2008)

3.3 Reinforcement Learning

Reinforcement learning (RL) (Sutton and Barto, 1998) is a machine learning scheme which can permit a decision maker to learn its optimal decisions (actions) through a series of trial-and-error interactions with a dynamic environment. Its main idea is to reinforce good behaviors of the decision maker while discouraging bad behaviors through a scalar reward value returned by the environment. In RL, the decision maker is called the agent whereas everything outside the agent is called the environment. Upon an action taken, the environment responds to the action by transiting to a new state.

Furthermore, the environment also feedbacks the agent the corresponding reward as a consequence of the action selection at a given state, which the agent tries to maximize overtime. More specifically, the agent and environment interact with each other in a sequence of discrete time steps. At each time step (*t*), the agent receives some representation of the environment's state (s_t) and select and action (a_t). On time step later, the agent receives a numerical reward (r_{t+1}) and finds itself in a new state (s_{t+1}). The agent should behave so as to maximize the long term benefit or the received reward, or more specifically, the average amount of accumulated rewards the agent receives over time.

3.4 Q-learning Strategy

Among the popular RL algorithm, Q-learning (Sutton and Barto, 1998) has been well investigated. Q-learning is a model-free algorithm which learns the values of the function *Q(s,a)* which quantifies how good it is to perform a certain action in a given state. With its ease of use, Q-learning has seen wide applications in resource allocation and is promising for dynamic environments such as mWSNs. Since Q-learning requires no prior model of the environment and can perform online learning, it is suitable for learning in non-cooperative mWSNs where little information is known among nodes.

In a MDP, the tuple (*S, A, P, r*) is defined to describe their characteristics, where *S* denotes the set of all possible states, *A* denotes the set of all possible actions, *P* is the state transition probability matrix such that $P(s' \mid s, a) \in P$ is the probability of transiting to the next state $s' \in S$ after an agent takes action $a \in A$ at state $s \in S$. *r* is a function of the reward expected from the environment as a result of taking action $a \in A$. The objective is to find a policy, π , defined as a mapping from the state space to the probability distribution, $\pi : S \rightarrow P[A]$, where $P[A]$ is the distribution over the action space. To determine the optimal policy, π^*, Q-learning requires the knowledge of a quantification of future benefits (or returns) at a given condition called the action-value function. The action-value function of a given policy π, denoted by $Q_t^\pi(s,a)$, associates a state-action pair (s,a) with an expected reward for performing action a in state s at time step t and following π thereafter;

$$
\begin{aligned}
Q_t^{\pi}(s,a) &= E^{\pi}\{R_t \mid s_t = s, a_t = a\} \\
&= E^{\pi}\{\sum_{k=0}^{\infty} \beta r_{t+k+1} \mid s_t = s, a_t = a\}
\end{aligned}
$$

where $R_t = r_{t+1} + \beta r_{t+2} + \beta^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \beta^k r_{t+k+1}$ is the expected discounted return at the time t of the agent, β is the discount factor and $E^{\pi}[\cdot]$ is the expectation operator of a given policy π. The goal of the Q-learning agent is to determine a policy to select actions so that its expected discounted future reward is maximized.

3.5 Problem Formulation

In this section, we propose an alternative RL approach to enhance routing cooperation among in mWSNs and present the details of how to formulate the problem. Based on the conjecture that different states of network loads may affect cooperation decisions for a particular routing node, we define the state s in our model as the quantized level of the network load experienced at a routing node where $s \in S$, S is the state space of the environment which is divided into 5 states, *i.e.* from low (0) to high (4) network level load. Each agent can independently decide its own action whether or not to cooperate with the other agent. The set of all the possible actions for a routing node is defined by $A = \{a_0, a_1\}$ where a_1 refers to agreeing to cooperate and a_0, otherwise.

During the learning process, the agent starts with an arbitrary initial Q-value. After executing action a at state s, the agent receives an immediate reward r and then transits to a new state and updates the new Q-value. The update rule at time step $t+1$ of Q-learning is given by;

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha[r + \beta \max_{a'} Q_t(s',a')], \tag{3.2}$$

where $0 \le \alpha < 1$ is the learning rate, $0 \le \beta \le 1$ is the discount factor, and $Q_t(s',a')$ is action-value function for the next state s' and next action a'. In this framework, the reward function for node i defined by:

$$r_i = I_v \tag{3.3}$$

where I_v is vital credit as shown in equation (3.1). The process is repeated iteratively to learn the agent's own optimal policy. The condition for Q-learning to converge is that all states and actions must be visited infinitely often (Sutton and Barto, 1998).

Figure 3.3 describes the procedure for applying Q-learning algorithm to CVCP. Suppose a source node sends a message, it first uses the incentive estimator to estimate the vital credit I_v to offer the routing nodes. Upon receiving the message, each routing node compares the offered vital credit to its stored credits I_{store}. If this I_v is greater than its I_{store}, this routing node will check the state of its network load (NT) and choose an action between *random action* and *greedy action*. The decision to choose *random action* or *greedy action* depends on the ε *-greedy probability*. Note that $\varepsilon \in [0,1]$ is the probability that a *greedy action* is selected. Note that ε can be set to zero in the training phase so that the agent can randomly explore all possible actions. On the other hand, ε can be set to unity to allow the selection of the greedy action which refers to an action such that $a^* = \arg\max_{\forall a \in A} Q(s,a)$. The ε *-greedy probability* is required to satisfy the convergence condition for Q-learning which is that all states and actions must be visited infinitely often (Sutton and Barto, 1998). Upon each decision taken at each node, *Q(s,a)* is updated according to (3.2).

However, if the value of I_v is less than I_{store}, the routing node then considers whether or not it will initiate the sleep cycle soon, by comparing the sleep cycle start time T_{start} with the system time $T_{current}$+ T. If T_{start} is less than $T_{current}$+T, it still stays active. It continues to operate by checking its state and select either a *random action* or *greedy action* according to the ε *-greedy probability*. If the routing node is about to initiate the sleep cycle; it will compare its I_{store} with a predefined threshold which is the vital credit required for entering a sleep cycle. If the I_{store} of this routing node is greater than such threshold, the node will subtract this amount from the current I_{store} and then checks it state. On the other hand, if T_{start} is greater than $T_{current}$+ T, this routing node will decline to cooperate and enter sleep mode where it will remain inactive for a finite period of sleep cycle.

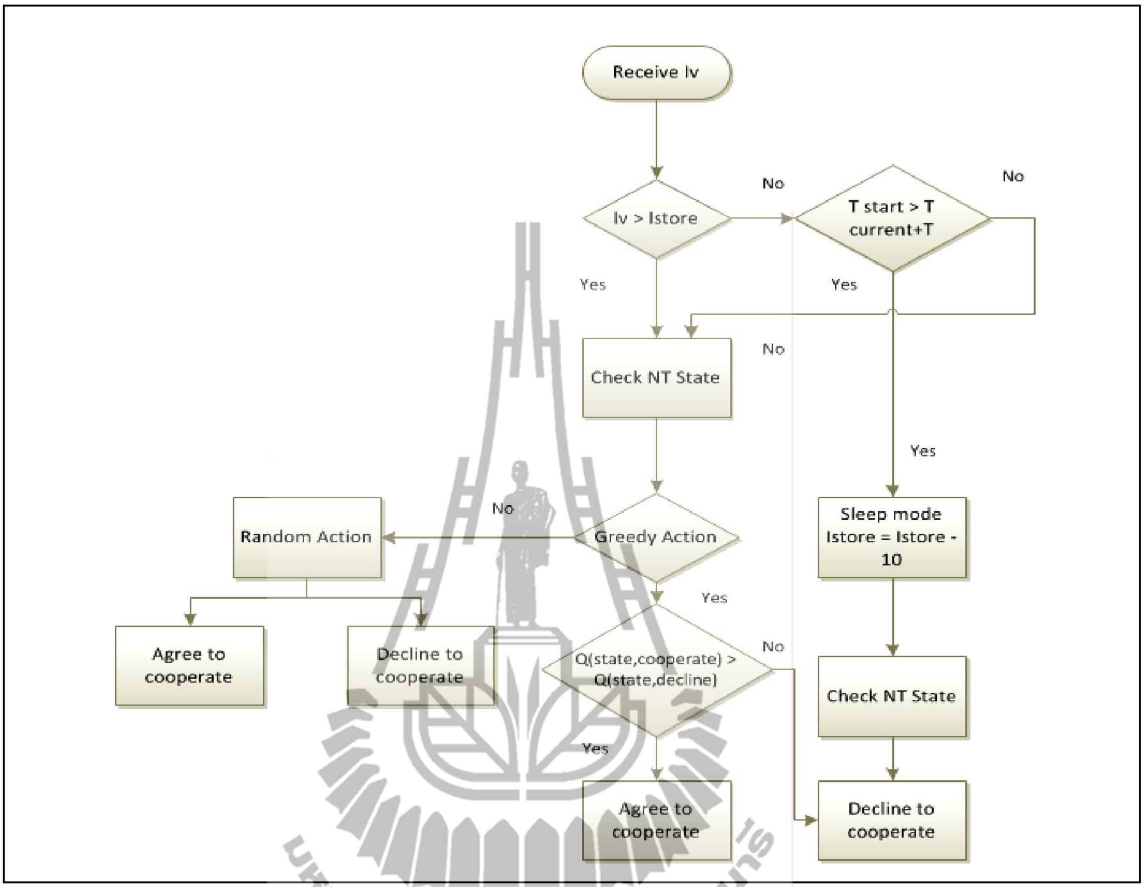

Figure 3.3 Diagram of the proposed algorithm which integrates Q-learning with CVCP performed at a routing node.

3.6 Homogeneous mWSN

In this section, we evaluated the proposed integrated CVCP and Q-learning algorithm and compared it with the original CVCP. Visual C++ was used to simulate a homogeneous mWSN under various conditions according to Table 3.1. All the 36 nodes within the mWSN followed the random way point mobility model and had equal initialized stored credits, while offered incentives I_v were based on (3.1). Packets were sent from an origin node to a destination node. Intermediate routing nodes then decided whether to cooperate or not depending on incentives they received and their sleep cycle period. Each node along a path received an offered vital credit from the origin node. If the offered vital credits were more than their stored credits, then they agreed to cooperate. Otherwise, the nodes declined to cooperate. The sleep cycle period also affected a node's decision. In particular, if a node had enough credits to sleep but the sleep cycle would not be initiated any time soon within a certain window, such node can agree to cooperate. The reward scheme, message arrival rate and message departure rate were varied to evaluate the performance under different types of message classes. In a healthcare scenario (Varshney, 2008) these message classes may present the significance or urgency of the vital sign measurements transmitted from a patient such as ECG signal, blood pressure, and oxygen saturation. Hence, in our simulation we classified the arrival packets into message classes which signify the importance and characteristics of each message class. The remaining simulation parameters are shown in Table 3.1. Note that the homogeneous node processing rate was equal to 0.1 message/sec for all nodes.

3.6.1 Performance Metrics

In order to evaluate the benefits of our RL algorithm, the following performance metrics were measured.

3.6.1.1 Average reward

This metric is the average normalized reward obtained over the course of simulation. Two different types of arrival message classes were evaluated as shown in the schemes presented in Table 3.2-3.5. Let $r_{average}$ be the average reward function generated from accepting the message classes under a particular policy of each algorithm, λ_i(message/sec) be the message arrival rate and μ_i(message/sec) be the message departure rate and P_i be the priority of the reward of message (r_i) according to (3.3) of class i. Let B_i be the rejection probability of message class i given by;

$$B_i = \frac{num\_rej\_unsat(i) + num\_rej\_sat(i)}{num\_arr\_msg(i)}, \tag{3.4}$$

where $num\_rej\_unsat(i)$ is the number of rejection messages in class i as a result of a node's (an agent's) decision when such node is unsaturated, $num\_rej\_sat(i)$ is the number of rejection messages in class i as a result of node's decision when such node is saturated, and $num\_arr\_msg(i)$ is the number of all messages arrival requests. We divided the states of a node into 5 states according to its processing capacity status as shown in Figure 3.4 with 0 being unsaturated and 4 being fully saturated capacity.

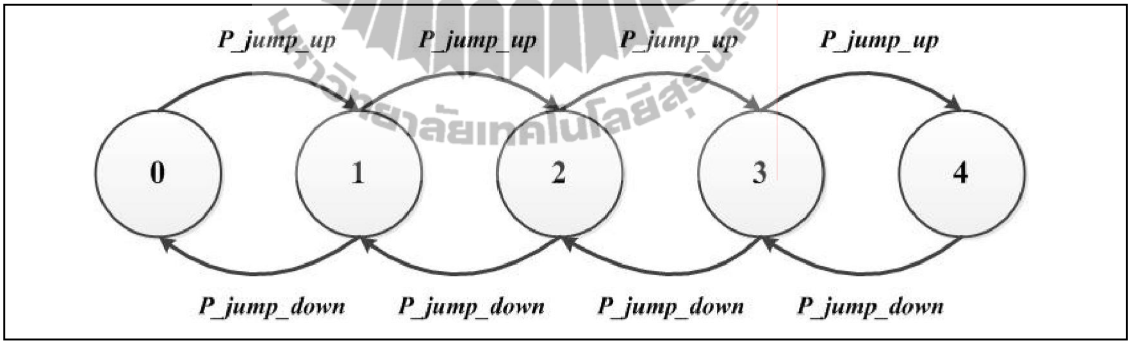

Figure 3.4 Birth – death diagram of node’s state.

Table 3.1 Simulation Parameters.

Parameters	**Value**
Number of sensor nodes	36
Node mobility	Random way point
Node velocity (m/s)	Min. = 0.3 Max = 0.7
Area size	13x13m^2
Transmission range	3m
Run length (number of route requests)	200000
Routing scheme	Shortest path
Sleep, wake cycle period (s)	30, 30
Credits spent per sleep cycle	10
C, M, N, L in (3.1)	1
P in (3.1)	See Table 3.2-3.7 for values of P_i

Note that *P_jump_up* is the probability of jumping up to the next upper state and *P_jump_down* is the probability of jumping down to the next lower state. *P_jump_up* is defined by;

$$P\_jump\_up = \frac{\lambda_i + \lambda_{state}}{(\lambda_i + \lambda_{state} + \mu_i)},$$

where λ_{state} is a state dependent node capacity usage rate. *P_jump_down* equals to 1 all states except for state 0 which is 0. If a node is in a saturated state and it still decides to cooperate in a message forwarding request for class i, then $num\_rej\_sat(i)$ is incremented. The average reward is given by;

$$r_{average} = \sum_{i=1}^{k} \frac{\lambda_i}{\mu_i} P_i (1 - B_i), \tag{3.5}$$

where P_i is the priority level of message class i, B_i is the rejection probability defined in (3.4), λ_i is the message arrival rate of message class i, μ_i is the message departure rate of message class i, and k is the number of all message classes.

Let r_{accept_all} be the average reward incurred when all message classes can be accepted when the node has no capacity saturation, *i.e.,* when the rejection probability B_i equals to zero, defined by;

$$r_{accept\_all} = \sum_{i=1}^{k} P_i \frac{\lambda_i}{\mu_i} . \tag{3.6}$$

From equation (3.5) and (3.6), we defined r_{norm} as the normalized average reward given by;

$$r_{norm} = \frac{r_{average}}{r_{accept\_all}} . \tag{3.7}$$

Figure 3.5, 3.6 and 3.7 illustrate that the RL method gave better performance results than CVCP in terms of normalized average reward. Figure 3.5 show results for the reward scheme setting in Table 3.2. Results show that accepting more messages in class 2 resulted in higher average rewards for RL, while the normalized average reward for each scheme from CVCP was indifferent. In terms of message arrival rate, we found that in Figure 3.6 when the message arrival rate was increased following the settings in Table 3.3, the rejection probability also increased as a result of an increase in the probability of jumping up so nodes landed in the saturated state (*i.e.,* state 4) more frequently. As a result the rejection probability increased thereby decreasing the normalized average reward. On the other hand, Figure 3.7 depicts the results when the message departure rate was increased according to Table 3.4. Results show that the rejection probability decreased due to faster message departure rate effect, consequently leading to fewer node and decreasing the rejection probability, and eventually increasing the normalized average reward. All schemes have shown that the proposed RL algorithm obtained a normalized average reward significantly higher than the existing CVCP algorithm.

Table 3.2 Varying reward scheme parameters.

Scheme	**Class 1**			**Class 2**		
	λ_1	μ_1	P_1	λ_2	μ_2	P_2
1	0.02	0.1	1	0.02	0.1	5
2	0.02	0.1	1	0.02	0.1	30
3	0.02	0.1	1	0.02	0.1	100

Table 3.3 Varying message arrival rate scheme parameters.

Scheme	**Class 1**			**Class 2**		
	λ_1	μ_1	r_1	λ_2	μ_2	P_2
1	0.01	0.1	1	0.01	0.1	5
2	0.05	0.1	1	0.05	0.1	30
3	0.10	0.1	1	0.10	0.1	100

Table 3.4 Varying message departure rate scheme parameters.

Scheme	**Class 1**			**Class 2**		
	λ_1	μ_1	r_1	λ_2	μ_2	P_2
1	0.02	0.01	1	0.02	0.01	5
2	0.02	0.05	1	0.02	0.05	30
3	0.02	0.10	1	0.02	0.10	100

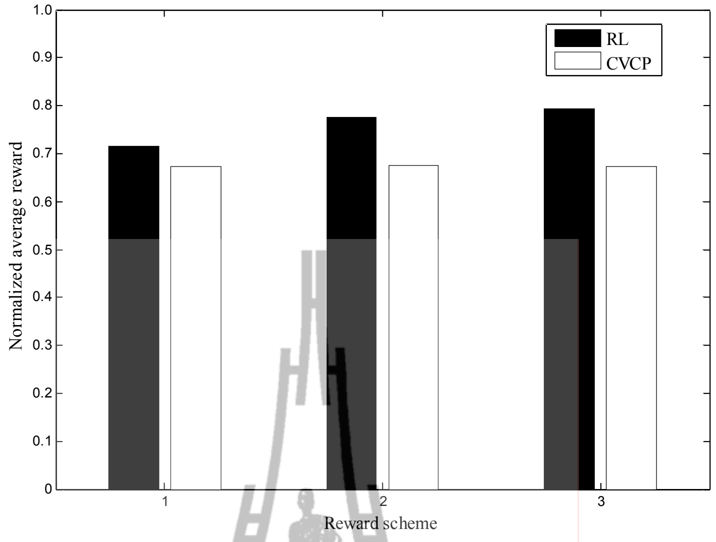

Figure 3.5 Normalized average reward of RL and CVCP in the reward scheme setting in Table 3.2.

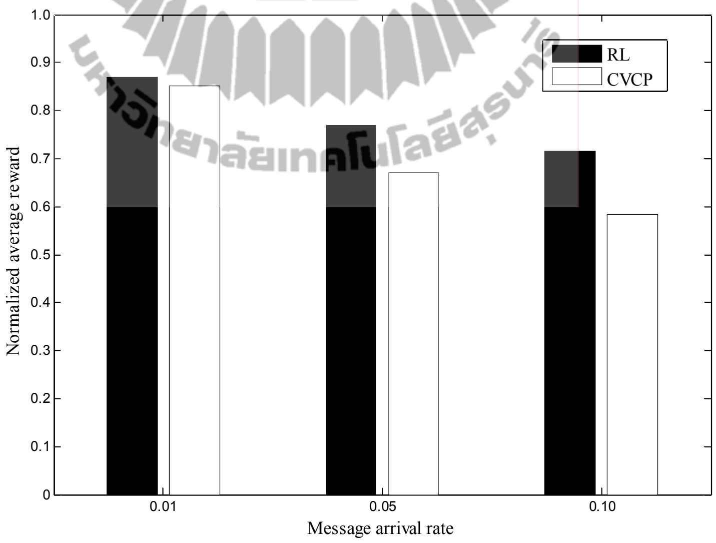

Figure 3.6 Normalized average reward of RL and CVCP in the message arrival rate setting in Table 3.3.

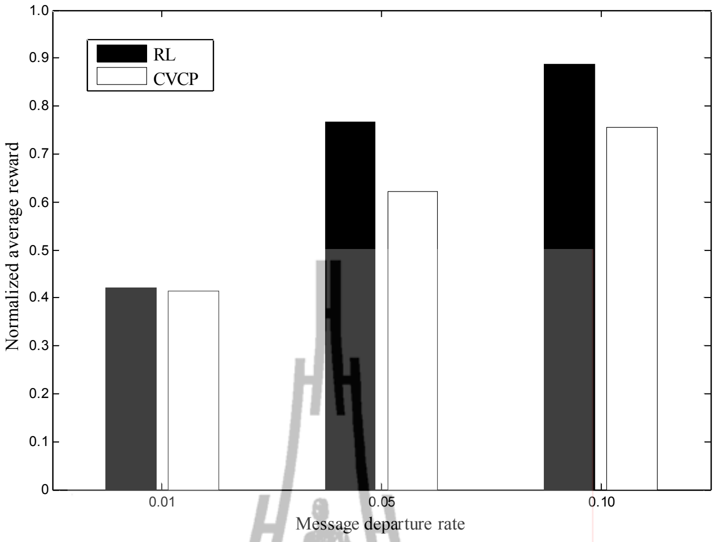

Figure 3.7 Normalized average reward of RL and CVCP in the message departure rate setting in Table 3.4.

3.6.1.2 Success ratio

This metric was determined by the number of class *i* messages successfully delivered to the destination node over selected paths. Let SR be the success ratio of the message delivery given by ;

$$SR = 1 - B_i . \tag{3.8}$$

Figure 3.8, 3.9 and 3.10 illustrate the success ratio versus various schemes in Table 3.2, Table 3.3 , Table 3.4 , respectively. Figure 3.8 depicts the success ratio of RL and CVCP reward scheme setting in Table 3.2 which show that the rejection probability of RL class 2 is less than RL class 1. It can be seen that the CVCP results do not depend on the weight of reward in any setting at all as evidently shown in the unvaried *SR*. Figure 3.9 illustrates the successful ratio obtained from varying the message arrival rate. We found that when the message arrival rate was increased following Table 3.3, the rejection probability increased due to the increase in the number of node satutation thereby decreasing the success ratio in (3.8). Figure 3.10 depicts the results when the message departure rate was increased according to Table 3.4. Results show that the rejection probability decreased from a lower number of node saturation and therefore the success ratio increased. Once again, the result from these two figures showed that the average reward from CVCP was indifferent. We noted that the ability to selectively accept more class 2 messages (which has higher priority) than class 1 messages under these settings was an advantage for RL over CVCP which accepted both classes equally.

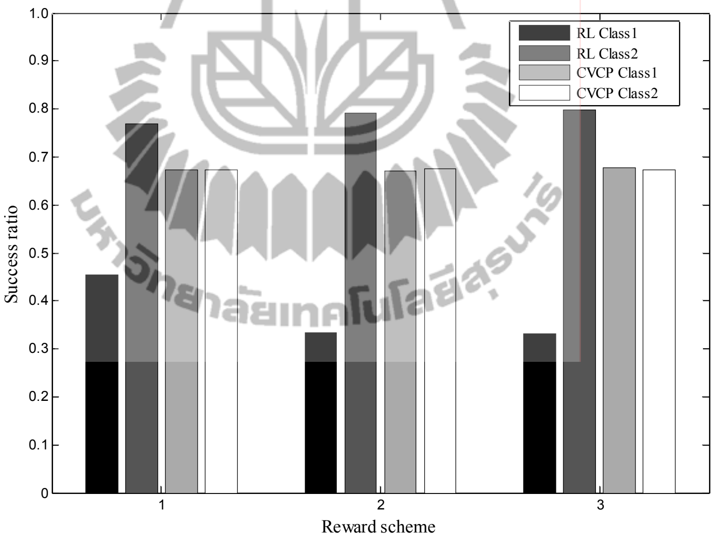

Figure 3.8 Success ratio of RL and CVCP under the reward scheme settings in Table 3.2.

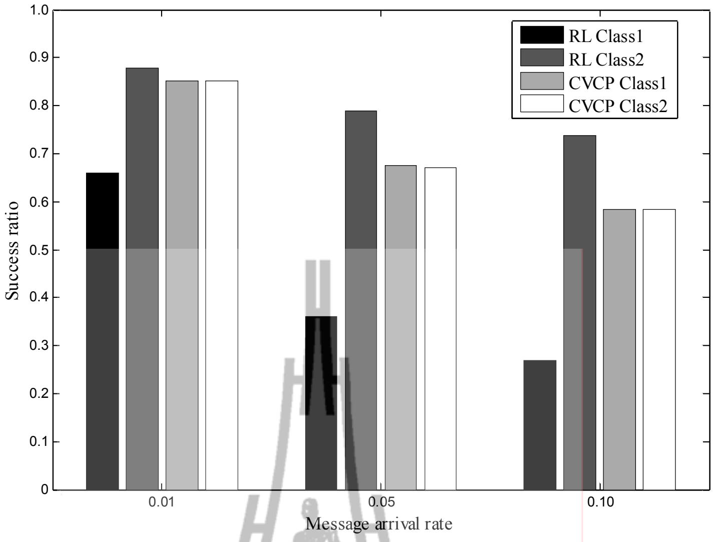

Figure 3.9 Success ratio of RL and CVCP under the message arrival rate scheme settings in Table 3.3.

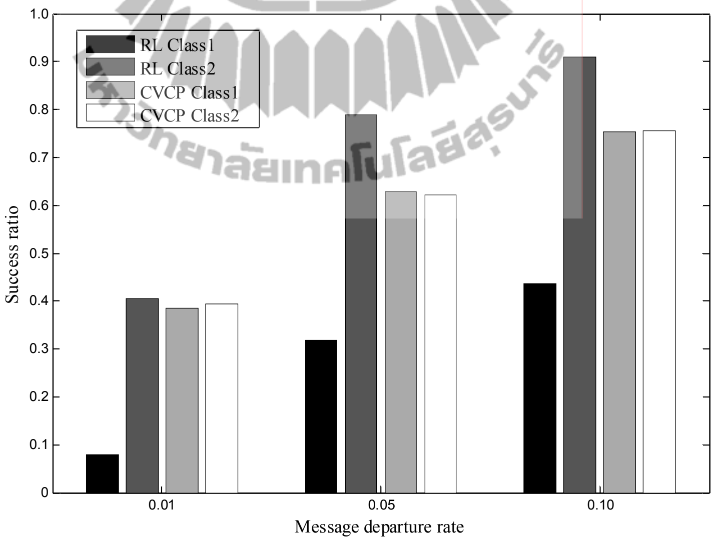

Figure 3.10 Success ratio of RL and CVCP under the message departure rate scheme settings in Table 3.4.

3.6.2 Homogeneous mWSN Conclusion

In this section, we proposed an incentive-based routing scheme for non-cooperative homogeneous mWSNs. The proposed method incorporates the RL into CVCP to solve the routing problem. Its performance was evaluated by means of simulation in terms of normalized average reward and success ratio. We compared the proposed RL and the existing CVCP algorithms. We found that the RL method consistently outperformed CVCP in terms of success ratio and normalized average reward under various reward, message arrival rate and message departure rate scheme settings. The preliminary results suggest that the proposed RL approach based on Q-learning algorithm can achieve better cooperation among nodes for high priority messages than the CVCP algorithm and that RL can be applied to improve cooperation among routing nodes in comparison to the existing incentive-based algorithm like CVCP.

In the next section, we extend the framework to address the heterogeneity of the sensor and routing nodes to cater more realistic scenarios which has been the main focus of this thesis.

3.7 Heterogeneous mWSN

In this section, we evaluated the proposed integrated CVCP and Q-learning algorithm and compared it with the original CVCP algorithm. Visual C++ was used to simulate a heterogeneous mWSNs under various conditions according to Table 3.1. Note that the heterogeneous mWSN may contain a mix of various types of nodes, different data collection abilities, different shapes and sizes and offering different functionalities and accommodating different constraints. Heterogeneity in terms of data collection has been considered so far in this work because of the various types of data collection in a healthcare scenario, ECG signal, blood pressure and oxygen saturation. In particular, certain vital signals require more reliability and priority over others. For instance, ECG signals require more frequent measurements than blood pressure from patients. A consistent stream of ECG measurements reliably delivered to a healthcare professional is necessary to assess the well-being of a patient. These different requirements led us to model vital signal measurements with messages of different classes, each with different arrival rates, service rates and reward weights. These were preliminarily investigated in the homogeneous mWSN section of this chapter and were found to affect the performances of CVCP and our proposed method. However, in this section we further consider heterogeneity in terms of node processing rates. We consider the heterogeneity in the terms of node processing rates because it represents a node's ability to forward a message in healthcare applications which is a major task for nodes in mWSNs. Apart from the node processing rates, all the 36 nodes in the mWSN followed the random way point mobility model and had equal initialized stored credits. The offered incentives I_v were based on (3.1). Packets were sent from an origin node to a destination node. Intermediate routing nodes then decide whether to cooperate or not depending on incentives they receive and their sleep cycle period. Each node along a path receives an offered vital credit from the origin node. If the offered vital credits are more than their stored credits, then they will agree to cooperate. Otherwise, the nodes decline to cooperate. The sleep cycle period affects a node's decision when that node has enough credits to sleep but the sleep cycle will not be initiated any time soon (within a certain window), such node can agree to cooperate. The reward, message arrival and message departure rates were varied to evaluate the performance under different scheme of message classes according to Table 3.5-3.7. In (Varshney, 2008), these message classes represent the vital signs of the patient such as ECG signal, blood pressure, and oxygen saturation. Hence, in our simulation we classified the messages into classes signify the importance of each message class. In the next section, each performance metric obtained from the simulation experiments in terms of normalized average reward, success ratio and the percentage of node usage for each type of node processing rate will be discussed.

Table 3.5 Varying reward scheme parameters for 2 traffic classes.

Scheme	**Class 1**			**Class 2**		
	λ_1	μ_1	P_1	λ_2	μ_2	P_2
1	0.02	0.1	1	0.02	0.1	5
2	0.02	0.1	1	0.02	0.1	20
3	0.02	0.1	1	0.02	0.1	80

Table 3.6 Varying reward scheme parameters for 3 traffic classes.

Scheme	**Class 1**			**Class 2**			**Class 3**		
	λ_1	μ_1	P_1	λ_2	μ_2	P_2	λ_3	μ_3	P_3
1	0.02	0.1	1	0.02	0.1	5	0.02	0.1	10
2	0.02	0.1	1	0.02	0.1	5	0.02	0.1	30
3	0.02	0.1	1	0.02	0.1	5	0.02	0.1	100

Table 3.7 Varying reward scheme parameters for 4 traffic classes.

Scheme	Class 1			Class 2			Class 3			Class 4		
	λ_1	μ_1	P_1	λ_2	μ_2	P_2	λ_3	μ_3	P_3	λ_4	μ_4	P_4
1	0.02	0.1	1	0.02	0.1	5	0.02	0.1	10	0.02	0.1	30
2	0.02	0.1	1	0.02	0.1	5	0.02	0.1	10	0.02	0.1	100
3	0.02	0.1	1	0.02	0.1	5	0.02	0.1	10	0.02	0.1	300

Table 3.8 Varying the number of node processing rates.

Scheme	Types of processing rate	0.1 message/sec	0.5 message/sec	0.7 message/sec	0.9 message/sec
1	2	18	-	18	-
2	2	12	-	24	-
3	3	12	12	-	12
4	3	4	12	-	20

3.7.1 Two heterogeneous processing rates (18:18)

This section we present the simulation results of 2 heterogeneous processing rates which have been symmetrically assigned to the mWSN, *i.e.,* 18 nodes with a processing rate equal to 0.1 message/sec and the other 18 nodes with a processing rate of 0.7 message/sec according to Table 3.6. Under this setting, we investigated three reward schemes presented in Table 3.5-3.7. Note that the message arrival rates and message departure rates were not varied, because these two parameters were already discussed in the homogeneous experiments in section 3.5. Instead, we focused on the effects of the node processing rates and increased message classesand discuss the results as follows.

3.7.1.1 Two traffic classes

This sub-section presents simulation results for 2 traffic classes of arrival messages with rewards varied according to Table 3.5. Figure 3.11 illustrates that RL method showed a steady increase in normalized average reward as the weight of rewards increased. On the other hand, that of the existing CVCP method remained unchanged regardless of the reward scheme used. The reason is because CVCP decisions were not dependent on the long term rewards, but rather the short term or immediate reward given by I_v. Therefore CVCP always accepts the decision to cooperate regardless of the message class whereas RL selectively accepts to cooperate mostly in presence high priority message classes. Such feature is most relevant in healthcare applications where the significance of vital signals should be emphasized. For example, ECG signal is the most significant message to be forwarded to the physician. This result suggests that RL can select decisions to decline or cooperate according to the characteristics of the vital signs of the patient, while CVCP can handle only with same priority. Figure 3.12 shows the success ratio of RL compared to CVCP, RL with class 2 can obtain more success ratio than CVCP. However, RL with class 1 obtained the lowest success ratio confirmed how RL decisions are based on long term rewards instead of immediate rewards. Figure 3.13 depicts the percentage of nodes with each type of processing rate used in RL and CVCP under scheme 1 in Table 3.8. The percentage of node processing rate for a message class is given by;

$$\%\ \text{node processing rate of type j} = \frac{\text{Number of nodes with type j processing rate which cooperated in a course of simulation}}{\text{Number of nodes of all types of processing rate which cooperated in a course of simulation}} \times 100\%$$

RS1 in Figure 3.13 refers to reward scheme 1 in Table 3.5 and so on. Note that both algorithms used each type of node equally for all reward schemes. Hence, the percentage of node processing rates did not depend on the algorithm. The reason is because our simulation consisted of two types of symmetrically allocated node processing rates and the topology of the network is random way point. All nodes moved randomly for both slow and fast processing nodes, so all of the nodes in our simulation were used with equally probability. Thus, this proportion agreed with the nodes processing rate proportion, 18:18.

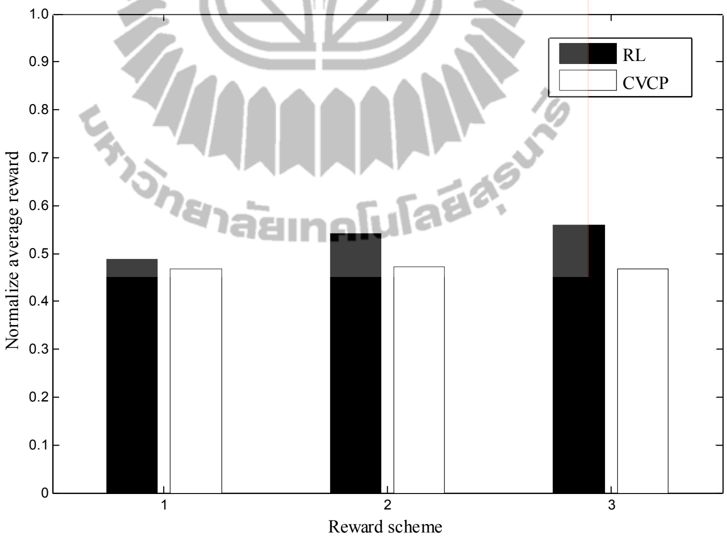

Figure 3.11 Normalized average reward of RL and CVCP for 18:18 heterogeneous processing rates with 2 traffic classes.

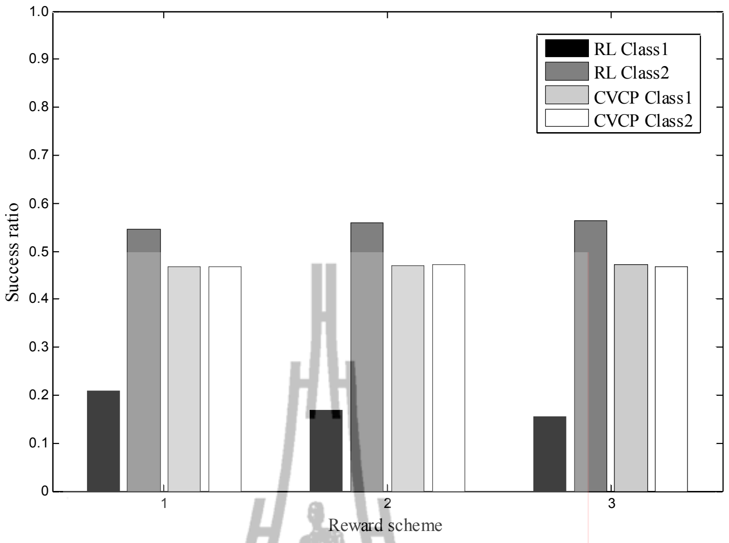

Figure 3.12 Success ratio of RL and CVCP for 18:18 heterogeneous processing rates with 2 traffic classes.

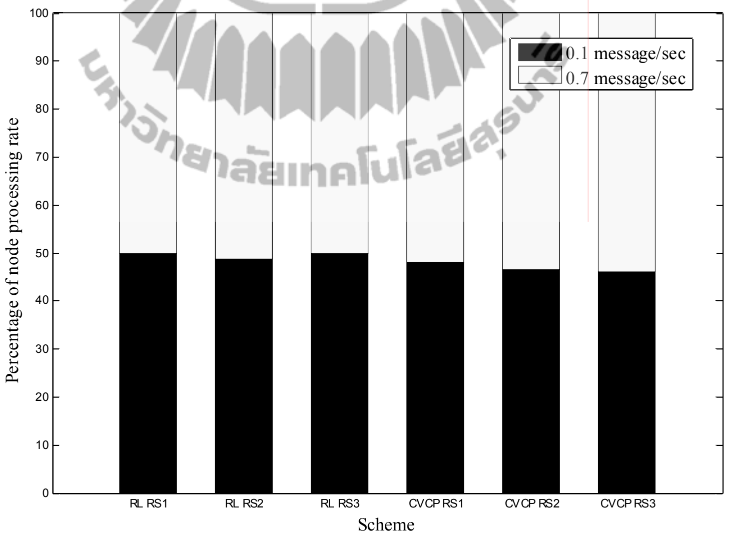

Figure 3.13 Percentage of node processing rates of RL and CVCP for 18:18 heterogeneous processing rates with 2 traffic classes.

3.7.1.2 Three traffic classes

In this experiment, simulation results are shown for 3 traffic classes of messages under the 18:18 heterogeneous processing rate nodes regime with rewards varying according to Table 3.6. We expanded the experiment from 2 traffic classes to investigate if results improved as the traffic message classes became more diverse. A network with more traffic classes in the view of healthcare application may represent more types of vital signs present in the network. For this reason, we expected the cooperation among nodes to increase giving rise to higher normalized average reward and success ratio in the simulation. From Figure 3.14, we observed that RL can get a better normalized average reward than CVCP in our experiment. Figure 3.15 shows the success ratio of RL compared to CVCP. Note that RL with class 3 which had the most normalized average reward achieved the most success ratio gain over the CVCP method. Similar to the 2 traffic classes, Figure 3.16 illustrates that both RL and CVCP equally used each type of node processing rate under scheme 1 in Table 3.8. Note that RS1 refers to reward scheme 1 in Table 3.6 and so on. Hence, the result in terms of percentage of node processing rate did not depend on the algorithm. The reason was because all nodes moved randomly, so all nodes had a chance to forward messages equally. To clearly demonstrate the RL performance trend, the next experiment in the following subsection extends to a scenario with 4 message classes.

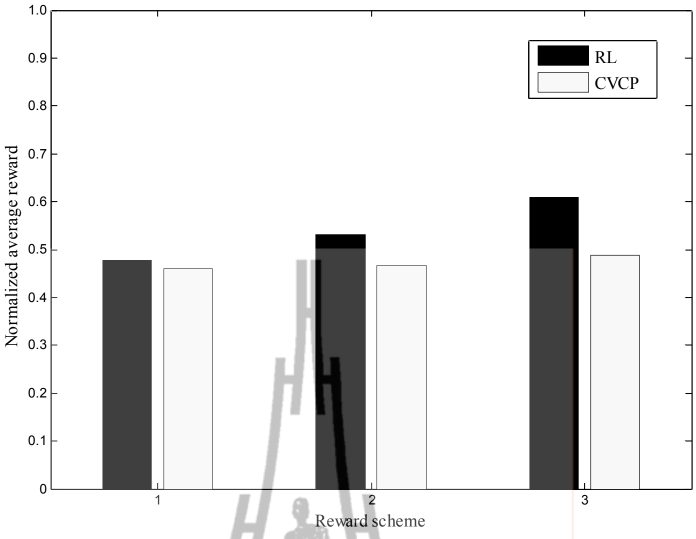

Figure 3.14 Normalized average reward of RL and CVCP for 18:18 heterogeneous processing rates with 3 traffic classes.

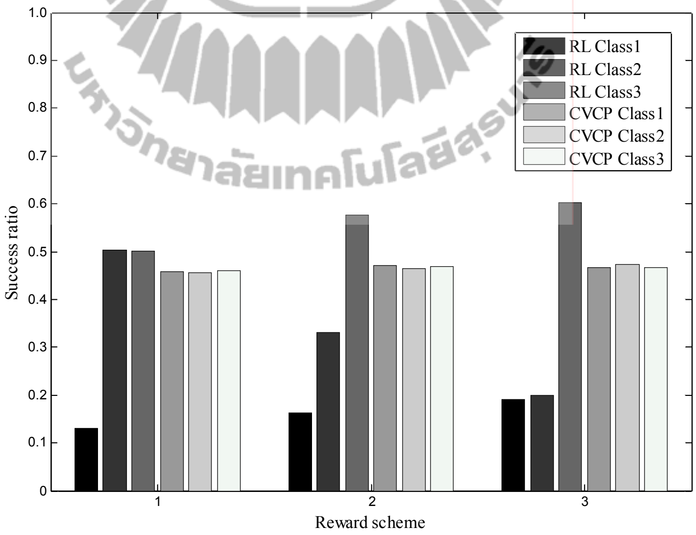

Figure 3.15 Success ratio of RL and CVCP for 18:18 heterogeneous processing rates with 3 traffic classes.

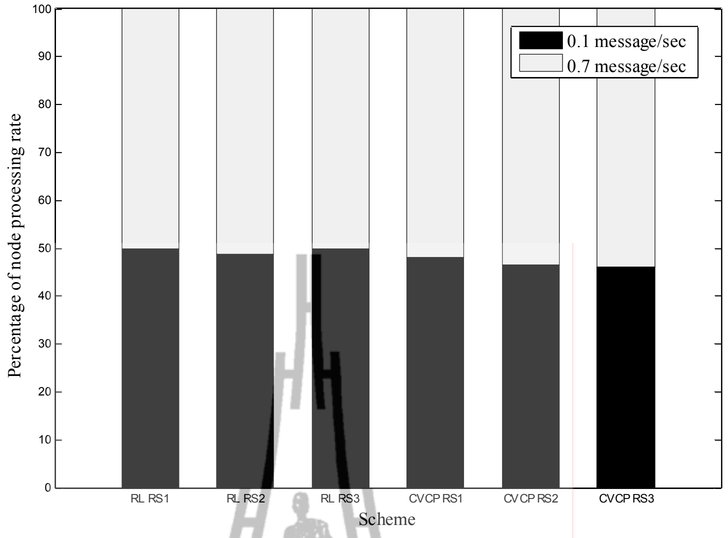

Figure 3.16 Percentage of node processing rates of RL and CVCP for 18:18 heterogeneous processing rates with 3 traffic classes.

3.7.1.3 Four traffic classes

In this subsection, 4 traffic classes of messages with varying rewards according to Table 3.7 was studied to show that the RL algorithm can support multi-classes of arrival messages and that the advantages of RL become even more evident in presence of diverse types of traffic. Figure 3.17 illustrates that the normalized average reward increased accordingly with the last reward scheme, RS3, attaining the most gain against the CVCP. Figure 3.18 shows that the success ratio of RL class 4 was consistently greater than CVCP for all reward schemes. Figure 3.19 illustrates the percentage of node processing rate according to scheme 1 in Table 3.8. Once again, the node usage was equally distributed for both RL and CVCP. The proportion of each type of processing rate was equal to 18:18 due to the random way point movement, so this led to the same result as the 2 and 3 message classes' scenarios.

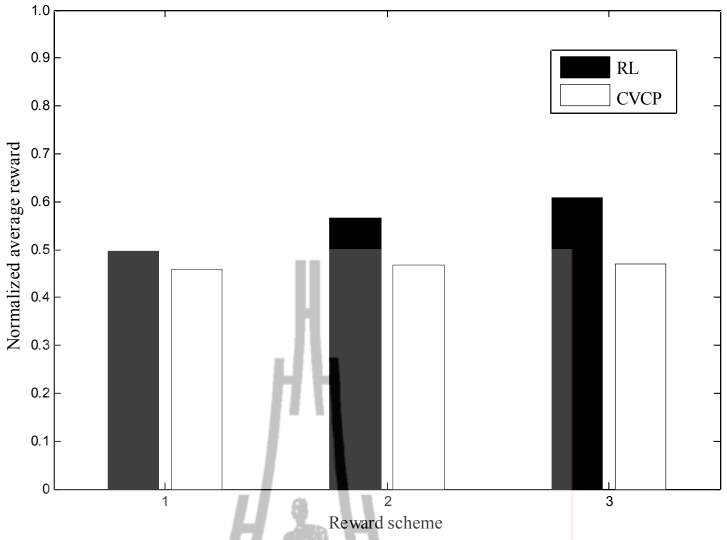

Figure 3.17 Normalized average reward of RL and CVCP for 18:18 heterogeneous processing rates with 4 traffic classes.

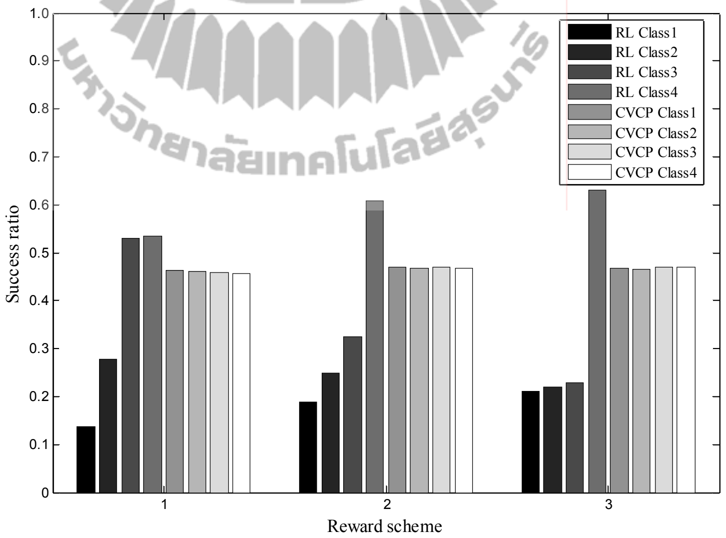

Figure 3.18 Success ratio of RL and CVCP for 18:18 heterogeneous processing rates with 4 traffic classes.

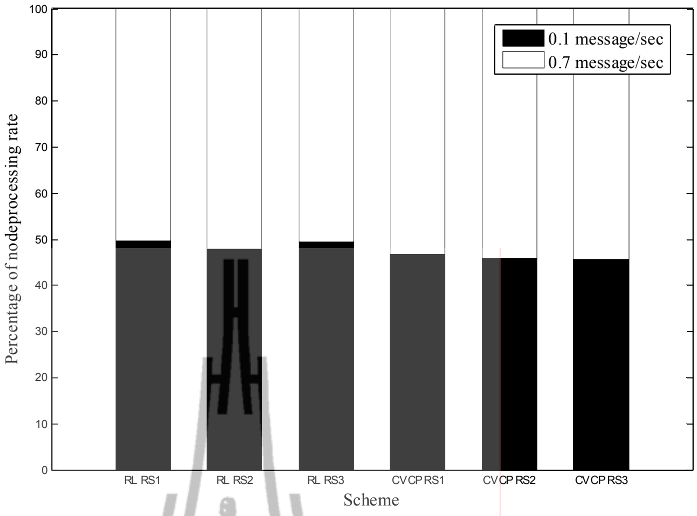

Figure 3.19 Percentage of node processing rates of RL and CVCP for 18:18 heterogeneous processing rates with 4 traffic classes.

3.7.2 Three heterogeneous processing rates (12:12:12)

In this section, we present the simulation results under scheme 3 in Table 3.8 where 3 heterogeneous processing rates were assigned symmetrically to 12 nodes in the 36 node mWSN, with processing rates equal to 0.1 message/sec, 0.5 message/sec and 0.9 message/sec, respectively according to Table 3.8. The purpose of this study was to investigate the performance under more types of node processing rates and to gauge the benefit of RL in presence of a higher degree of heterogeneity.

3.7.2.1 Two traffic classes

We initially studied two traffic classes for the sake of simplicity, where rewards were varied according to Table 3.5. Results in Figures 3.20 and 3.21 agreed with the scenario of 2 message classes with 18:18 node proportion, with higher normalized average rewards and success ratios for the message class with the most reward. Figure 3.22 which shows the percentage of node usage, also agreed with Figure 3.12 where each node type was equally used.

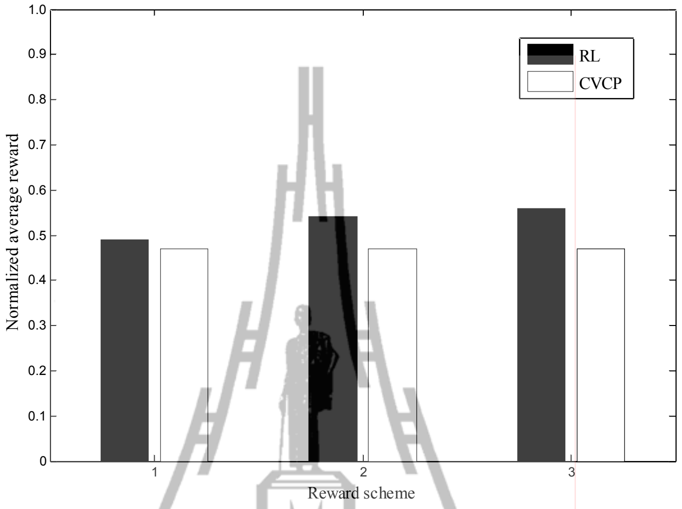

Figure 3.20 Normalized average reward of RL and CVCP for 12:12:12 heterogeneous processing rates with 2 traffic classes.

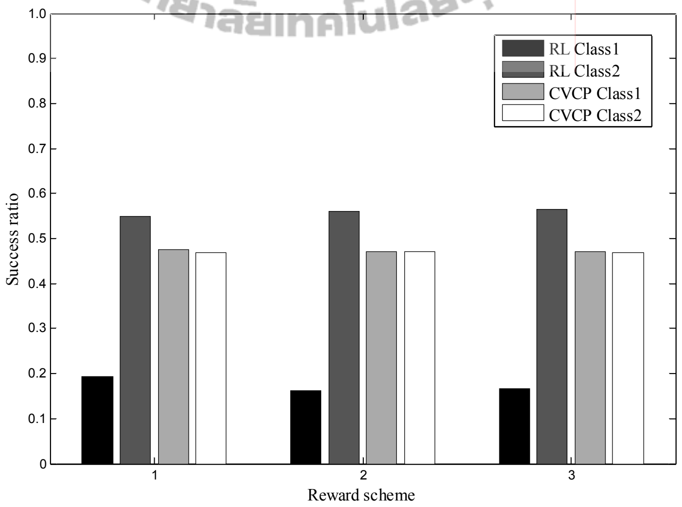

Figure 3.21 Success ratio of RL and CVCP for 12:12:12 heterogeneous processing rates with 2 traffic classes.

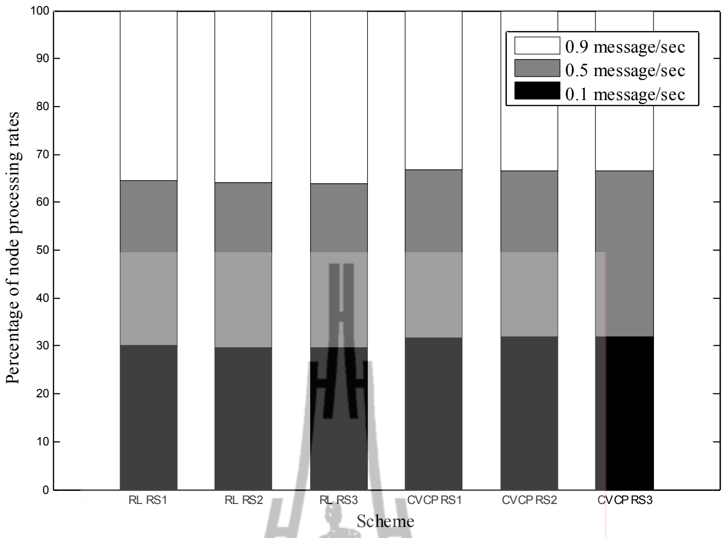

Figure 3.22 Percentage of node processing rates of RL and CVCP for 12:12:12 heterogeneous processing rates with 2 traffic classes.

3.7.2.2 Three traffic classes

In this sub-section, 3 traffic classes have been investigated by varying the reward according to Table 3.6. Figure 3.23 and 3.24 depict the normalized average reward and success ratio in this scenario, respectively. It can be observed that the success ratio for RL was dependent on the reward assigned to each message class whereas such dependency was absent in CVCP. Figure 3.25 illustrates the percentage of node processing rate according to scheme 3 in Table 3.8. This result shows that both CVCP and RL used 33% of each node processing rate. This was because the setting node processing rate was 12:12:12 thus agreeing with the 18:18 heterogeneous node processing rate case in Figure 3.15.

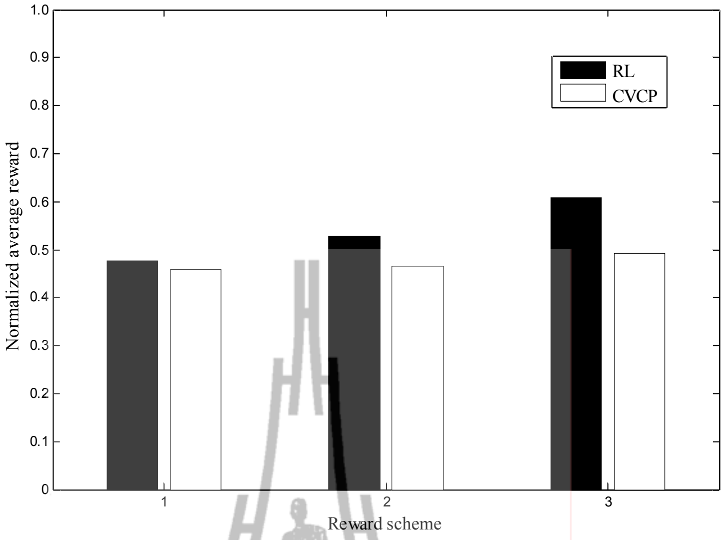

Figure 3.23 Normalized average reward of RL and CVCP for 12:12:12 heterogeneous processing rates with 3 traffic classes.

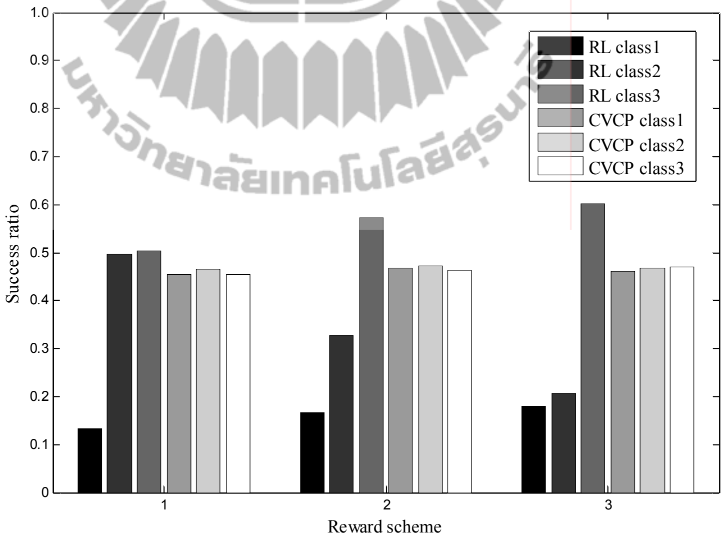

Figure 3.24 Success ratio of RL and CVCP for 12:12:12 heterogeneous processing rates with 3 traffic classes.

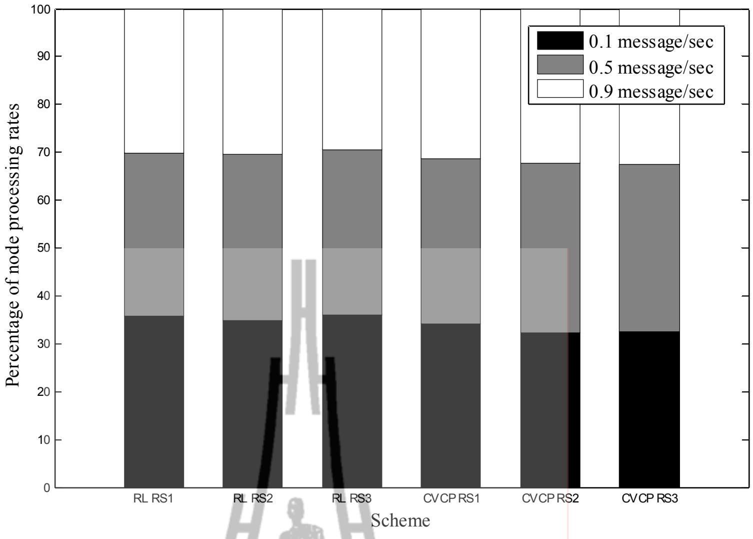

Figure 3.25 Percentage of node processing rates of RL and CVCP for 12:12:12 heterogeneous processing rates with 3 traffic classes.

Figure 3.26, 3.27 and 3.28 present results for the 4 message classes under the reward regime in Table 3.7, in a 12:12:12 heterogeneous mWSN scenario in scheme 3 in Table 3.8. As the reward of the fourth message class increased, so did its success ratio with a trade off in decrease of the success ratio in other classes. Fairness can be guaranteed by introducing a penalty parameter (Tong and Brown, 2000) to ensure that the rejection probability of each message class does not fall below a predetermined threshold. This would be a constrained optimization problem which can be extended from this work (See Section 4.2). The percentage of node processing rate usage in Figure 3.28 was also proportional to the number of each type of node in the network.

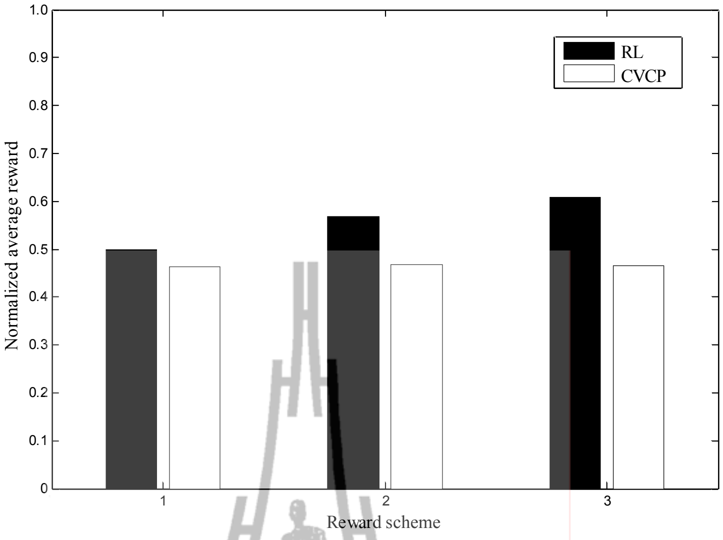

Figure 3.26 Normalized average reward of RL and CVCP for 12:12:12 heterogeneous processing rates with 4 traffic classes.

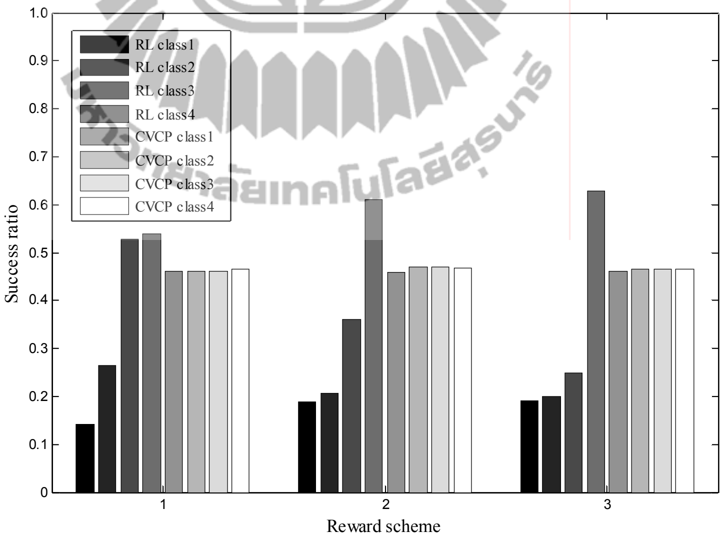

Figure 3.27 Success ratio of RL and CVCP for 12:12:12 heterogeneous processing rates with 4 traffic classes.

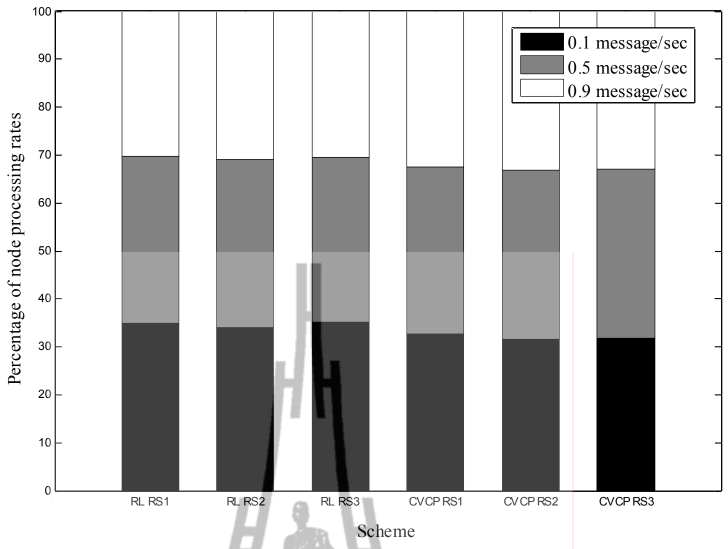

Figure 3.28 Percentage of node processing rates of RL and CVCP for 12:12:12 heterogeneous processing rates with 4 traffic classes.

3.7.3 Alternative node processing rates

In this subsection, we studied the effects of asymmetrical node processing rates assignment in the heterogeneous mWSN under scheme 2 (12:24) and 4 (4:12:20) in Table 3.8. In particular, we studied the gain in normalized average reward of RL over the existing CVCP method as the degree in heterogeneity in messages classes and node processing rate increases in the mWSN. It was found that the normalized average rewards and success ratio demonstrated similar patterns as presented in section 3.7.1 and 3.7.2, we do not show them in this subsection for the sake of redundancy. However, a complete presentation of results in this subsection can be found in Appendix B. In this section, we focus the percentage of node processing rates in Figure 3.29 and 3.30 for 4 message classes under schemes 2 and 4 in Table 3.8, respectively. Once again, the percentage of node usage for both the RL and CVCP algorithms were found to be proportional to the amount of nodes of each type of processing rates, irrespective of the reward schemes. Such results suggest that both algorithms utilized nodes in a similar manner, *i.e.,* according to the node availability within the network. Nodes with faster processing rates were not used to forward messages more often than the slower ones, thereby ensuring a certain degree of fairness in node utilization within the network.

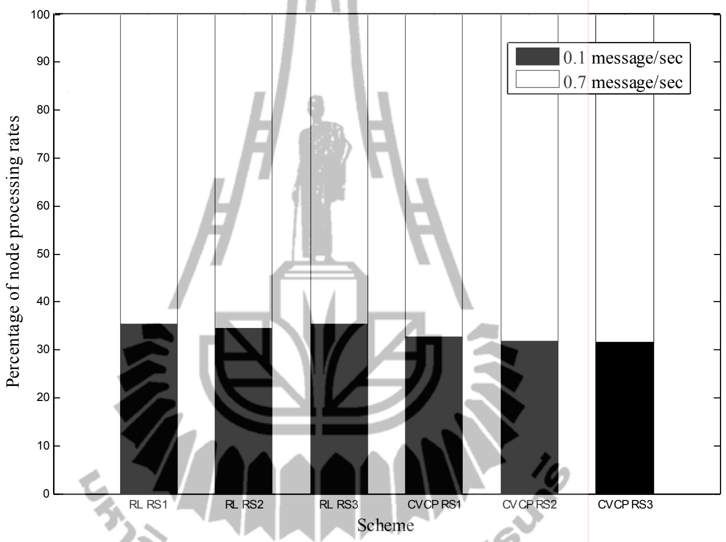

Figure 3.29 Percentage of node processing rates of RL and CVCP for 12:24 heterogeneous processing rates with 4 traffic classes.

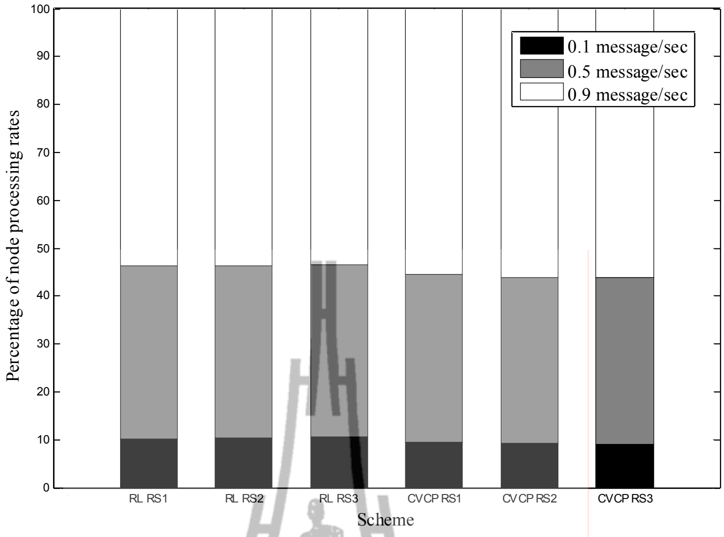

Figure 3.30 Percentage of processing rates of RL and CVCP for 4:12:20 heterogeneous processing rates with 4 traffic classes.

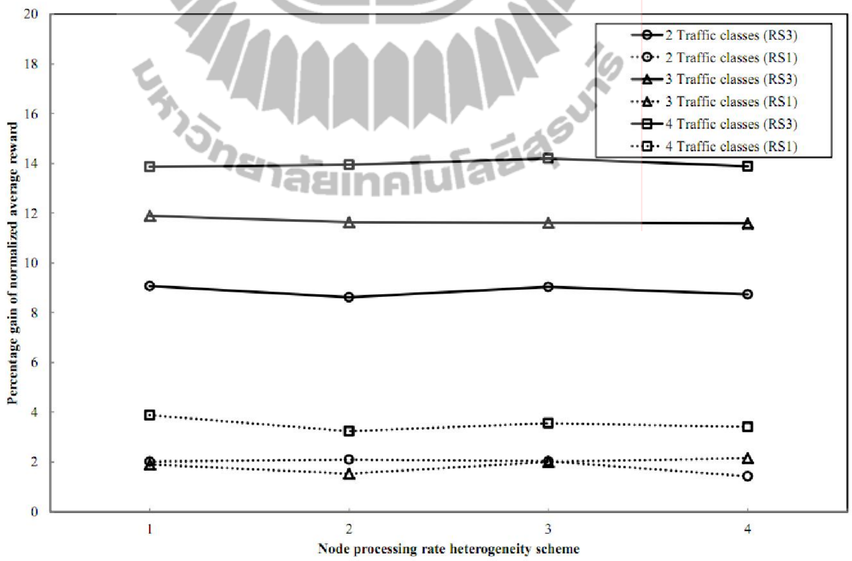

Figure 3.31 Percentage gain of normalized average reward for all node processing rate heterogeneity schemes in Table 3.8.

Figure 3.31 illustrates the gain of normalized average reward for all node processing rate heterogeneity schemes according to Table 3.8. The gain in normalized average reward refers to the normalized average reward of RL subtracted by the normalized average reward of CVCP. Results are shown for the maximum reward scheme setting (RS3) and the minimum reward scheme setting (RS1) according to Table 3.5-3.7. From the figure, it can be observed that the gain in normalized average reward was affected by reward setting of the message class schemes being used, irrespective of the node processing rate heterogeneity regime. In particular, RS3 which was the maximum reward scheme used in the experiments consistently obtained up to 9-14% gain in normalized average reward over the CVCP method, whereas RS1 gained about 2-4% gain, depending on the number of traffic classes present in the network. The fact that the gain in normalized average reward was invariant to the diverse node processing rates available in the network suggested that intermediate nodes' decisions on whether or not to cooperate are based on the future expected return for a particular message class alone. The level of node processing rate nor the amount of nodes in each type of processing rate did not have any significant impact on the performance of the proposed algorithm and the original CVCP algorithm. Therefore, some degree of fairness can be observed in the routing decisions at the intermediate nodes for both algorithms. That is, nodes with faster processing capabilities were not inclined to cooperate any more frequent than slower processing rate nodes. However, the RL framework proposed in this work allowed the nodes in the heterogeneous mWSN to selectively cooperate in forwarding a particular message class and achieve a better normalized average reward than the CVCP algorithm.

3.7.4 Heterogeneous mWSN Conclusion

In this section, we studied the proposed incentive-based routing in a non-cooperative heterogeneous mWSNs. The proposed method incorporates a RL method called Q-learning into an existing incentive-based scheme called CVCP to solve the routing problem. The main focus of the proposed method was on enhancing routing cooperation in heterogeneous mWSNs, particularly for high priority message classes which require critical and reliable handling from intermediate nodes within the network. The problem was formulated under the RL framework using vital credits as incentives. However, instead of basing decisions on the vital credits alone as the original CVCP method, our proposed method took into consideration the future expected benefits of agreeing or declining to cooperate in the packet forwarding process. Moreover, we studied symmetric and asymmetric node processing rates in mWSNs operating under various message reward regimes in an order to cater a more realistic scenario for healthcare applications which require more complexity in terms of node and traffic heterogeneity. Simulation results showed that for all multi-traffic class regimes, RL outperformed CVCP in terms of normalized average rewards by up to 14%. However, the percentage of node processing rate did not depend on any algorithm but only on the proportion of nodes of each type of node processing rate. Such result suggests that the advantage of the proposed method ensures a certain degree of fairness in node selection, *i.e.,* faster nodes were not used more frequently than slower nodes. Nodes only cooperated based on the incentives or vital credits as well as the future benefits of their decisions at a particular state. In the final subsection, results also showed that heterogeneity in node processing rate did not affect our experiment results, In particular, the normalized average rewards and success ratio did not show any significant changes as the node processing rate heterogeneity changed, although RL consistently gained 2-14% of normalized average reward, depending on the reward regime, over the original CVCP method. The percentage of node usage in each type of node processing rate only depended on the proportion of each type of nodes for both algorithms. The results in our experiment suggest that RL can be applied to improve cooperation among routing nodes in comparison to an existing incentive-based algorithm like CVCP.

CHAPTER IV

CONCLUSION AND FUTURE WORK

4.1 Conclusion

In this thesis, we proposed a RL method called Q-learning to enhance routing cooperation among nodes in non-cooperative heterogeneous mobile wireless sensor networks (mWSNs). The work carried out in this thesis was divided into two parts which were homogeneous and heterogeneous node processing rate non-cooperative mWSNs. We first simulate the homogeneous node processing rate scenario to compare the results with an existing algorithm Continuous Value Cooperation Protocol (CVCP) to analyze the effects of traffic or message class heterogeneity alone on the routing performance within the network. In a subsequent experiment, we then extend the heterogeneity to encompass a broader case of different node processing rate scenario. These two parts were presented in Chapter 3.The original contributions and findings in this thesis can be summarized as follows.

4.1.1 Homogeneous mWSNs

The purpose of this section was to demonstrate that the Q-learning algorithm can be applied to promote routing cooperation in non-cooperative homogeneous mWSNs in comparison with an existing CVCP algorithm. Two contributions were made here:

1) The simulation result comparison between RL and the existing CVCP algorithm in non-cooperative mWSNs.

The proposed experiment results showed that the RL method consistently outperformed CVCP in terms of success ratio and normalized average reward for all reward, message arrival rate and message departure rate scheme settings. These elementary results suggested that the proposed RL approach based on Q-learning algorithm can be applied to improve cooperation among routing nodes with a homogeneous node processing rate under the presence of different traffic classes in comparison to the existing CVCP incentive-based algorithm.

4.1.2 Heterogeneous mWSNs

The purpose of this section was to extend the framework from homogeneous mWSNs to address many challenges associated with an incentive-based routing for non-cooperative heterogeneous mWSNs. Heterogeneity of the sensor and routing nodes was applied to cater more realistic scenarios. One contribution was made here:

1) The simulation result comparison of the proposed method and the existing CVCP algorithm in non-cooperative mWSNs in presence of heterogeneous node processing rates under different traffic regimes.

The significance of our work was centered on proposing means to enhance routing cooperation among nodes in heterogeneous mWSNs, particularly, in the presence of high priority message classes which require critical and reliable handling from intermediate nodes within the network. Moreover, we studied symmetric and asymmetric node processing rates in mWSNs operating under various message class schemes to cater a more realistic scenario for healthcare applications in terms of node and traffic heterogeneity. The results showed that, RL algorithm can promote more robust performance than CVCP algorithm in terms of success ratio and normalized average reward. We also evaluated the percentage of node processing rate which only depended on the proportion of each type of nodes for both algorithms. This suggests that RL approach based on Q-learning algorithm can obtain the better performance than the CVCP.

4.2 Future Works

4.2.1 mWSNs with Transmission Cost Function

Each sensor node should employ a radio model (Naruephiphat and Usaha, 2008) to compute the transmission and receiving cost required for transmitting a packet. To study the effect of this radio model, many challenges associated with incentive-based routing for non-cooperative mWSNs with the transmission cost function should be addressed.

4.2.2 mWSNs with Energy Consumption Condition

Energy consumption in mWSNs is one of the most important issues. To manage the energy problems in mWSNs, a possible future direction is to study how to manage the energy consumption to achieve the optimal solution with incentive-based routing for non-cooperative mWSNs.

4.2.3 Performance Evaluation of Test Bed

The main objective of this thesis was to compare packet forwarding strategies in incentive-based routing for non-cooperative mWSNs governed by using both RL and CVCP algorithms. This experiment 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 algorithm, or to implement the framework in an actual mWSN.

4.2.4 Extend the State Space of RL

Larger state spaces should be investigated, particularly, the impact of this larger state space to our performance metrics including the normalized average reward, success ratio and percentage of node processing rate.

4.2.5 Guarantee the Fairness for Message Rejection Probability

According to the results from our work, the success ratios of message classes with lower priority were not guaranteed. To ensure that the rejection probability of each message class does not fall below a predetermined threshold, a penalty parameter can be introduced to guarantee fairness of each message class (Tong and Brown, 2000). This would be a constrained optimization problem which can be extended from this work.

REFERENCES

Agah, A., Das, S. K., & Basu, K. 2004. A non-cooperative game approach for intrusion detection in sensor networks. **Vehicular Technology Conference**, 2004. VTC2004-Fall. 2004 IEEE 60th.

Baldus, H., Klabunde, K., & Msch, G. 2004. Reliable Set-Up of Medical Body-Sensor Networks, **Wireless Sensor Networks**: 353-363.

Bevly, D. M., Ryu, J., & Gerdes, J. C. 2006. Integrating INS Sensors With GPS Measurements for Continuous Estimation of Vehicle Sideslip, Roll, and Tire Cornering Stiffness. **Intelligent Transportation Systems, IEEE Transactions**, 7(4): 483-493.

Bonivento, A., Carloni, L. P., & Sangiovanni-Vincentelli, A. 2006. Platform-Based Design of Wireless Sensor Networks for Industrial Applications. **Design, Automation and Test in Europe**, 2006. DATE '06. Proceedings.

Busoniu, L., Babuska, R., & De Schutter, B. 2008. A Comprehensive Survey of Multiagent Reinforcement Learning. **Systems, Man, and Cybernetics, Part C: Applications and Reviews, IEEE Transactions**, 38(2): 156-172.

Chengnian, L., Qian, Z., Bo, L., Huilong, Y., & Xinping, G. 2007. Non-cooperative power control for wireless ad hoc networks with repeated games. **Selected Areas in Communications, IEEE Journal** , 25(6): 1101-1112.

Chunping, W., & Wei, W. 2009. A Load-Balance Routing Algorithm for Multi-Sink Wireless Sensor Networks. **Communication Software and Networks,** 2009. ICCSN '09. International Conference.

Egorova-Forster, A., & Murphy, A. L. 2007. Exploiting Reinforcement Learning for Multiple Sink Routing in WSNs. **Mobile Adhoc and Sensor Systems,** 2007. MASS 2007. IEEE Internatonal Conference.

Felegyhazi, M., Hubaux, J. P., & Buttyan, L. 2006. Nash equilibria of packet forwarding strategies in wireless ad hoc networks. **Mobile Computing, IEEE Transactions** , 5(5): 463-476.

Field, M. J. 1996. **Telemedicine:A Guide to Assessing Telecommunications for Health Care**: The National Academies Press.

Forster, A., Murphy, A. L., Schiller, J., & Terfloth, K. 2008. An Efficient Implementation of Reinforcement Learning Based Routing on Real WSN Hardware. **Networking and Communications,** 2008. WIMOB '08. IEEE International Conference on Wireless and Mobile Computing.

Gharehshiran, O. N., & Krishnamurthy, V. 2009. Dynamic coalition formation for efficient sleep time allocation in wireless sensor networks using cooperative game theory. **Information Fusion,** 2009. FUSION '09. 12th International Conference.

Guangcheng, H., & Xiaodong, W. 2008. An Opportunistic Routing for Mobile Wireless Sensor Networks Based on RSSI. **Wireless Communications, Networking and Mobile Computing,** 2008. WiCOM '08. 4th International Conference.

Hu, J., & Wellman, M. P. 2003. **Nash q-learning for general-sum stochastic games**. J. Mach. Learn. Res., 4: 1039-1069.

Huang, Y. M., Hsieh, M. Y., Chao, H. C., Hung, S. H., & Park, J. H. 2009. Pervasive, secure access to a hierarchical sensor-based healthcare monitoring architecture in wireless heterogeneous networks. **Selected Areas in Communications, IEEE Journal**, 27(4): 400-411.

Hussein, O., & Saadawi, T. 2003. Ant routing algorithm for mobile ad-hoc networks (ARAMA). **Performance, Computing, and Communications Conference,** 2003. Conference Proceedings of the 2003 IEEE International.

Iyengar, S. S., Hsiao-Chun, W., Balakrishnan, N., & Shih Yu, C. 2007. Biologically Inspired Cooperative Routing for Wireless Mobile Sensor Networks. **Systems Journal, IEEE,** 1(1): 29-37.

Jurik, A. D., & Weaver, A. C. 2008. Remote Medical Monitoring. **Computer**, 41(4): 96-99.

Koucheryavy, A., & Salim, A. 2009. Cluster head selection for homogeneous Wireless Sensor Networks. **Advanced Communication Technology,** 2009. ICACT 2009. 11th International Conference.

Lewis, N., & Foukia, N. 2008. An Efficient Reputation-Based Routing Mechanism for Wireless Sensor Networks: Testing the Impact of Mobility and Hostile Nodes. **Privacy, Security and Trust,** 2008. PST '08. Sixth Annual Conference.

Liqiang, Z., Hailin, Z., & Jie, Z. 2008. Using Incompletely Cooperative Game Theory in Wireless Sensor Networks. **Wireless Communications and Networking Conference**, 2008. WCNC 2008. IEEE.

Liu, Z., Guan, X., & Chen, C. 2008. Energy-efficient optimal scheme based on mixed routing in wireless sensor networks. **Control Conference,** 2008. CCC 2008. 27th Chinese.

Machado, R., & Tekinay, S. 2008. A survey of game-theoretic approaches in wireless sensor networks. **Computer. Networks,** 52(16): 3047-3061.

Makarevitch, B. 2000. Application of reinforcement learning to admission control in CDMA network. **Personal, Indoor and Mobile Radio Communications,** 2000. PIMRC 2000. The 11th IEEE International Symposium.

Michiardi, P., & Molva, R. 2003. A Game Theoretical Approach to Evaluate Cooperation Enforcement Mechanisms in Mobile Ad hoc Networks. **In Modeling and Optimization in Mobile, Ad Hoc and Wireless Networks,** 2003, 3-5**.**

Minh Hanh, N., & Krishnamurthy, V. 2005. Game Theoretic Optimal Transmission Strategies in Multipacket Reception Sensor Networks. **Signals, Systems and Computers,** 2005. Conference Record of the Thirty-Ninth Asilomar Conference.

Munir, S. A., Biao, R., Weiwei, J., Bin, W., Dongliang, X., & Man, M. 2007. Mobile Wireless Sensor Network: Architecture and Enabling Technologies for Ubiquitous Computing. **Advanced Information Networking and Applications Workshops,** 2007, AINAW '07. 21st International Conference.

Naruephiphat, W., & Usaha, W. 2008. Balancing Tradeoffs for Energy-Efficient Routing in MANETs Based on Reinforcement Learning. **Technology Conference,** 2008. VTC Spring 2008. IEEE.

Pham, N. N., Youn, J., & Won, C. 2006. A Comparison of Wireless Sensor Network Routing Protocols on an Experimental Testbed. **Sensor Networks, Ubiquitous, and Trustworthy Computing,** 2006. IEEE International Conference.

Ping, W., & Ting, W. 2006. Adaptive Routing for Sensor Networks using Reinforcement Learning. **Computer and Information Technology,** 2006. CIT '06. The Sixth IEEE International Conference.

Puccinelli, D., & Haenggi, M. 2009. Lifetime Benefits through Load Balancing in Homogeneous Sensor Networks. **Wireless Communications and Networking Conference**, 2009. WCNC 2009. IEEE.

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

Romer, K., & Mattern, F. 2004. The design space of wireless sensor networks. **Wireless Communications, IEEE**, 11(6): 54-61.

Schiller, J., Liers, A., Ritter, H., Winter, R., & Voigt, T. 2005. ScatterWeb - Low Power Sensor Nodes and Energy Aware Routing. **System Sciences,** 2005. HICSS '05. Proceedings of the 38th Annual Hawaii International Conference.

Shah, K., & Kumar, M. 2007. Distributed Independent Reinforcement Learning (DIRL) Approach to Resource Management in Wireless Sensor Networks. **Mobile Adhoc and Sensor Systems,** 2007. MASS 2007. IEEE Internatonal Conference.

Shah, N., Qian, D., & Iqbal, K. 2008. Performance evaluation of multiple routing protocols using multiple mobility models for mobile ad hoc networks. **Multitopic Conference,** 2008. INMIC 2008. IEEE International.

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

Tong, H., & Brown, T. X. 2000. Adaptive call admission control under quality of service constraints: a reinforcement learning solution. **Selected Areas in Communications, IEEE Journal,** 18(2): 209-221.

Usaha, W., & Barria, J. A. 2007. Reinforcement Learning for Resource Allocation in LEO Satellite Networks. **Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions,** 37(3): 515-527.

Varshney, U. 2008. Improving Wireless Health Monitoring Using Incentive-Based Router Cooperation. **Computer,** 41(5): 56-62.

Vazquez-Abad, F. J., & Krishnamurthy, V. 2002. Self learning call admission control for multimedia wireless DS-CDMA systems. **Discrete Event Systems,** 2002. Proceedings. Sixth International Workshop.

Wanming, C., Tao, M., Yangming, L., Huawei, L., Yumei, L., & Meng, M. Q. H. 2007. An Auto-adaptive Routing algorithm for Wireless Sensor Networks. **Information Acquisition,** 2007. ICIA '07. International Conference.

Wanzhi, Q., Pham, M., & Skafidas, E. 2008. Routing and localization for extended lifetime in data collection wireless sensor networks. **Communications and Networking,** 2008. ChinaCom 2008. Third International Conference.

Watkins, C. 1989. **Learning from Delayed Rewards**. University of Cambridge,England.

Xiaoxia, H., Hongqiang, Z., & Yuguang, F. 2008. Robust cooperative routing protocol in mobile wireless sensor networks. **Wireless Communications, IEEE Transactions** , 7(12): 5278-5285.

Xiaoxia, H., Hongquiang, Z., & Yugang, F. 2006. Lightweight Robust Routing in Mobile Wireless Sensor Networks. **Military Communications Conference,** 2006. MILCOM 2006. IEEE.

Xin, M., & Wei, L. 2008. The Analysis of 6LowPAN Technology. **Computational Intelligence and Industrial Application,** 2008. PACIIA '08. Pacific-Asia Workshop.

Xuedong, L., Balasingham, I., & Sang-Seon, B. 2008. A multi-agent reinforcement learning based routing protocol for wireless sensor networks. **Wireless Communication Systems,** 2008. ISWCS '08. IEEE International Symposium.

Zhang, K., Li, Y., Xliao, W.-h., & Suh, H. 2008. The Application of a Wireless Sensor Network Design Based on ZigBee in Petrochemical Industry Field. **Intelligent Networks and Intelligent Systems,** 2008. ICINIS '08. First International Conference.

APPENDIX A

PUBLICATION

Publication

Rittong, C., and Hattagam, W. (2011). **Improving Router Cooperation in Mobile Wireless Sensor Networks Using Reinforcement Learning.** The 9th Embedded and Ubiquitous Computing (EUC), Australia, December 2011.

2011 IFIP Ninth International Conference on Embedded and Ubiquitous Computing

Improving Router Cooperation in Mobile Wireless Sensor Networks using Reinforcement Learning

Chanon Rittong
School of Telecommunication Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand 30000
E-mail: M5140671@g.sut.ac.th

Wipawee Usaha
School of Telecommunication Engineering
Suranaree University of Technology
Nakhon Ratchasima, Thailand 30000
E-mail: wusaha@ieee.org

***Abstract*— This paper proposes to promote cooperative routing for homogeneous mobile wireless sensor networks (mWSNs) using a scalable, distributed incentive-based mechanism with reasonable resource requirements using reinforcement learning (RL). In particular, Q-learning which is a well-known RL method was integrated an existing Continuous Value Cooperation Protocol (CVCP). We also studied their effects on the efficiency in non-cooperative mWSNs and propose a good routing strategy under constrained conditions such as network traffic load, degree of mobility and path loss exponent.**

Keywords- Reinforcement Learning, Mobile Wireless Sensor Network, Routing Cooperation

I. INTRODUCTION

A wireless sensor network (WSN) usually consists of numerous sensor nodes deployed in the area of interest. Each node is able to collect and process data with neighboring devices. There are many reasons for its popularity, including low costs, flexibility and ease of deployment. However, WSNs have some constraints, such as limited power supply, storage, bandwidth, and computation capability. Such constraints combined with a typical deployment of large number of sensor nodes have posed may challenges to the design and management of sensor networks. These challenges necessitate energy-awareness at all layers of networking protocols stack. At the network layer, the aim is to set up energy-efficient routes and reliably relay data from sensor nodes to the sink so that the lifetime of the network is maximized. Therefore, there are many researches which aim at solving these routing problems in WSNs.

Most current researches assume WSNs to be stationary. However, in many scenarios WSNs must be mobile. For instance, for wild life monitoring, sensor nodes are cast into the region of interest as well as equipped on animals to be monitored. The self-organized WSN is mobile as animals move around. In a telemedicine application [2], sensor nodes attached to moving patients also form a mWSN. Furthermore, most routing schemes assume that nodes function properly, are trustworthy and cooperative. However, in realistic scenarios, nodes may fail to operate due to lack of resources, hardware failure or malicious behaviors. There are many algorithms which are used to deal with non-cooperative routing in mWSNs. The incentive-based concept has been applied in many algorithms such as reputation-based routing mechanism [3], Nash-Q [4], reinforcement learning (RL) [5], Game theory [6]. Nodes decide whether to cooperate or not based on incentives stored or earned. In [7], the authors proposed an incentive-based mechanism called continuous value cooperation protocol (CVCP) for healthcare monitoring to improve the routing cooperation of mobile wireless sensor nodes which are attached to patients. In [8], the authors proposed an efficient implementation of RL-based routing on real mWSNs.

The incentive-based concept is the one of the effective tools for solving the routing problem in non-cooperative mWSNs. Reputation mechanisms are typically used to enhance security by identifying and avoiding malicious nodes, but not promote node cooperation. Game theory requires knowledge of the other opponents' strategy, thereby may not be scalable especially in dynamic environments as mWSNs. On the other hand, RL can cater a large number of nodes with distributed operation using only local information from the neighboring nodes.

The objective of this paper is therefore to solve the routing problem for non-cooperative mWSNs using a scalable, distributed incentive-based mechanism. In particular, we apply a RL method called Q-learning to promote packet forwarding in a periodic sleep cycle homogeneous mWSN. We compare its performance with an existing sleep cycle incentive-based routing algorithm [7] under various mobility and traffic scenarios.

II. CONTINUOUS VALUE COOPERATION PROTOCOL (CVCP)

The continuous value cooperation protocol (CVCP) [7] has been proposed to promote router cooperation in ad hoc

978-0-7695-4552-3/11 $26.00 © 2011 IEEE
DOI 10.1109/EUC.2011.45

networks deployed to supplement infrastructure-oriented wireless health monitoring systems. The protocol used incentives called vital credit which is a function of message priority level (P), network traffic loads (NT), and criticality of the message delivery (C). Vital credits (I_v) are defined as:

$$I_v = NT^L \times P^M \times C^N \tag{1}$$

where L, M and N represent constants that can be chosen to emphasize certain factors in vital credits. For example, nodes transmitting emergency messages or alerts can be assigned vital credits greater than symptoms monitoring nodes to encourage delivery; the network traffic level depends on frequency of monitoring, number of packets per message, number of monitored patients; node criticality may rely on the location of the routing node and routing scheme.

Figure 1 shows an individual routing node using CVCP to decide whether to forward a particular message to a destination node based on the number of vital credits offered by the source node, and the routing node's already earned vital credits. The source device uses an *incentive estimator* to determine the vital credits it will offer to a routing node to forward its message. The routing node stores already earned vital credits. If the offered vital credits exceed its stored credits, the more likely a routing node will cooperate. On the other hand, a routing node with a large number of stored credits might not cooperate even if the offered number of credits is high. If a routing node decides to cooperate, it receives the offered vital credits from the source node and adds them to its stored credits.

Nodes with the most vital credits can receive higher sleep-cycle priority, thereby promoting energy saving. However, nodes that have used up their vital credits for a recent sleep cycle are more likely to cooperate to increase their earned vital credits. Figure 2 depicts the CVCP vital credit checking procedure at a routing node. Furthermore, it also checks whether a sleep cycle will be initiated soon and opts to cooperate accordingly.

However, cooperation also incurs certain energy consumption costs as a result of agreeing to forward a packet. Such costs may vary according to the hostility of the mobile environment. Decisions should also consider the long term average reward which is a trade-off between earned credits and energy consumption costs. Furthermore, apart from checking the offered and stored credits and sleep cycles, decisions to cooperate or not may also be dependent on state conditions of the routing node other than shown in Figure 2. For instance, different states of network loads or residual battery levels may provide different decisions in order to achieve an optimal long term average reward for a particular routing node. A scalable, distributed self-learning scheme with reasonable computation requirements described in the next section warrants potential use in highly dynamic systems such as mWSNs.

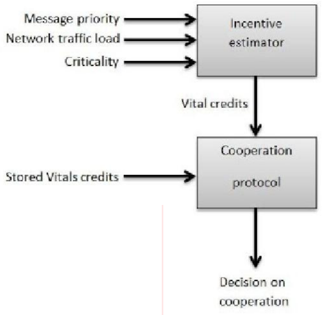

Figure 1. Cooperation protocol [7] in which a routing node makes a decision based on vital credits the source node offers and its stored credits.

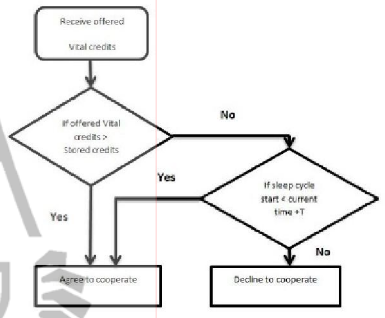
Receive offered Vital credits
If offered Vital credits > Stored credits
No
Yes
If sleep cycle start < current time +T
Yes
No
Agree to cooperate
Decline to cooperate

Figure 2. Diagram of CVCP checking procedure performed at a routing node [7].

III. Reinforcement Learning

Reinforcement learning (RL) [5] is a machine learning scheme which can permit a decision maker to learn its optimal decisions (actions) through a series of trial-and-error interactions with a dynamic environment. Its main idea is to reinforce good behaviors of the decision maker while discouraging bad behaviors through a scalar reward value returned by the environment. In RL, the decision maker is called the agent whereas everything outside the agent is call the environment. Upon an action taken, the environment responds to the action by transiting to a new state. Furthermore, the environment also feedbacks the agent the corresponding reward as a consequence of the action selection at a given state, which the agent tries to maximize overtime. More specifically, the agent and environment interact with each other in a sequence of discrete time steps. At each time step (t), the agent receives some representation of the environment's state (s_t) and select and action (a_t). On time step later, the agent receives a numerical reward (r_{t+1}) and finds itself in a new state (s_{t+1}). The agent should behave so as to maximize the received reward, or more specifically, the amount of accumulated rewards the agent receives over time.

IV. Q-Learning

Among the popular RL algorithm, Q-learning [5] has been well-investigated and relies on a Markov decision theory (MDP). Q-learning is a model-free algorithm which learns the values of the function *Q(s,a)* which quantifies how good it is to perform a certain action in a given state. With its ease of use, Q-learning has seen wide applications in resource allocation and is promising for dynamic environments such as mWSNs. Since Q-learning requires no prior model of the environment and can perform online learning, it is suitable for learning in non-cooperative mWSNs where little information is known among nodes.

In a MDP, the tuple (*S, A, P, R*) is defined to describe their characteristics, where *S* denotes the set of all possible states, *A* denotes the set of all possible actions, *P* is the state transition probability matrix such that $P(s' \mid s, a) \in P$ is the probability of transiting to the next state $s' \in S$ after an agent takes action $a \in A$ at state $s \in S$. *R* is a function of the reward expected from the environment as a result of taking action $a \in A$. The objective of solving a MDP is to find a policy, π , defined as a mapping from the state space to the probability distribution, $\pi : S \rightarrow P\left[A\right]$, where $P\left[A\right]$ is the distribution over the action space. To determine the optimal policy, π^* , RL requires the knowledge of a quantification of future benefits (or returns) at a given condition called the action-value function. The action-value function of a given policy π , denoted by $Q_t^{\pi}(s,a)$, 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;

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

where $R_t = r_{t+1} + \beta r_{t+2} + \beta^2 r_{t+3} + \ldots = \sum_{k=0}^{\infty} \beta^k r_{t+k+1}$ is the expected discounted return of the agent, β is the discount factor and $E^{\pi}\{\cdot\}$ is the expectation operator of a given policy π . The goal of the RL agent is to determine a policy to select actions so that its expected discounted future reward is maximized.

Based on the conjecture that different states of network loads or residual battery levels may affect cooperation decisions for a particular routing node, we define the state *s* in our game as the quantized level of the network load experienced at a routing node where $s \in S$, *S* is the state space of the environment which is divided into 5 states, i.e. from low (*1*) to high *(5)*. Each agent can independently decide its own action whether or not to cooperate with the other agent. A set of all the possible actions is defined by $A = \{a_0, a_1\}$ where a_1 refers to agreeing to cooperate and a_0, otherwise.

Suppose that the agent at time step *t* executes action *a* in state *s*, then transits to the next state *s'* and obtains a scalar reward *r*. During the learning process, the agent starts with an arbitrary initial Q-value. After executing action *a* at state *s*, the agent receives an immediate reward *r* and then updates both the new state and new Q-value with input from the environment. The update rule at time step t+1 of Q-learning is given by:

$$Q_{t+1}(s,a) = (1-\alpha)Q_t(s,a) + \alpha[r + \beta \max_{a'} Q_t(s',a')], \tag{2}$$

where $\alpha \in [0,1]$ is the learning rate, $\beta \in [0,1]$ is the discount factor, and *Q(s',a')* is action-value function for the next state *s'* and next action *a'*. In this work, *r* is a reward function for node *i* defined by:

$$r_i = I_v - C_i(b,d) \tag{3}$$

where C_i is the transmission cost function of node *i* which comes from the path loss exponent model given by:

$$C_i(b,d) = E_{elec} \times b + (\varepsilon_{amp} \times b \times d^{\sigma}), \tag{4}$$

where E_{elec} = 50 nJ/bit is the expended cost in the radio electronics and we assume that *b* = 250 Kbits is the size of the measurement packet transmitted, σ is the path loss exponent and ε_{amp} = 10pJ/bit/m$^{\sigma}$ is the energy consumed at the output transmitter antenna for transmitting one meter and *d* is the distance between node. The process is repeated iteratively to learn the agent's own optimal policy. The condition for Q-learning to converge is that all states and actions must be visited infinitely often [5].

Figure 4. describes the procedure of applying Q-learning algorithm to CVCP. Suppose a source node sends a message, it first uses the incentive estimator to estimate the vital credit I_v to offer the routing nodes. Upon receiving the message, each routing node compares the offered vital credit to its stored credits I_{store}. If this I_v is greater than its I_{store}, this routing node will check the state of its network load (NT) and chooses an action between *Random action* and *Greedy action*. The decision to choose *Random action* or *Greedy action* depends on the ε *-greedy probability*. Note that $\varepsilon \in [0,1]$ is the probability that a *Greedy action* is selected which is defined as:

$$\varepsilon = 0.9 + \left(\frac{0.9}{runlength}\right) \times current\_num\_event \qquad (5)$$

where *Greedy action* refers to $a^* = \arg\max_{\forall a \in A} Q(s,a)$. Otherwise, a *Random action,* which refers to the action being randomly selected in a uniform fashion, is selected. The ε -*greedy probability* is required so as to satisfy the convergence condition for Q-learning which is that all states and actions must be visited infinitely often [5]. Note that the ε -*greedy probability* increases with time in order to encourage more greedy action selection as the agent progressively learns. Upon each decision taken at each node, *Q(s,a)* is updated according to (2) for the state action pair at that particular node.

However, if the value of I_v is less than I_{store}, the routing node then considers whether it will intiate the sleep cycle soon or not, by comparing the sleep cycle start time T_{start} with the system time $T_{current}$+ T. If T_{start} is less than $T_{current}$+T, it still stays active. It continues to operate by checking its state and select either a *Random action* or *Greedy* action according to the ε -*greedy probability*. If the routing node is about to initiate the sleep cycle; it will compare its I_{store} with a predefined threshold which is the vital credit spent for entering a sleep cycle. If I_{store} of this routing node is greater than this threshold, the node will subtract this amount from the current I_{store} and then checks its state. Finally, this node will then decline to cooperate and enter sleep mode where it remains inactive for a finite period of sleep cycle.

V. SIMULATION RESULTS

In this section, we evaluated the proposed integrated CVCP and Q-learning algorithm and compared it with the original CVCP. Visual C++ was used to simulate a homogeneous, mWSNs under various conditions according to Table 1. All the 36 nodes of mWSNs which followed the random way point mobility model had equal initialized stored credits, while offered incentives I_v were based on (1). Packets were sent from an origin node to a destination node. Intermediate routing nodes then decide whether to cooperate or not depending on incentives they receive and their sleep cycle period. Each node along a path receives an offered vital credits from the origin node. If the offered vital credits are more than their stored credits, then they will agree to cooperate. Otherwise, the nodes decline to cooperate. The sleep cycle period affects a node's decision when that node has enough credits to sleep but the sleep cycle will not be initiated any time soon (within a certain window), such node can agree to cooperate. The pause time was varied to evaluate the performance under different degrees of mobility. The remaining simulation parameters are shown in Table 1.

A. *Performance metrics*

- **Percentage of cooperative nodes:** the evaluation of this metric is given by:

$$\%coop = \left(\frac{all\_coop\_node}{all\_node\_in\_path} \times 100\right) \qquad (6)$$

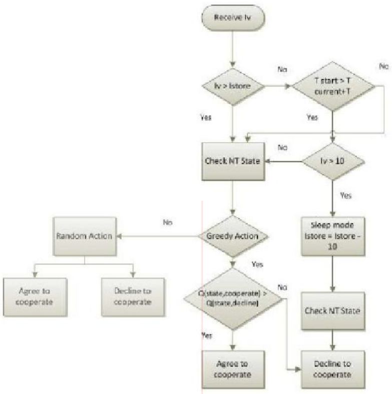

Figure 3. Diagram of the proposed algorithm which integrates Q-learning with CVCP performed at a routing node.

TABLE 1. SIMULATION PARAMETERS.

Parameters	Value
Number of sensor nodes	36
Node mobility	Random way point
Node pause time (PT)	0, 30 ,60 ,180, 300 s.
Node velocity (m/s)	Min. = 0.3 Max = 0.7
Area size	13x13m^2
Transmission range	3m
Runlength (number of route requests)	200000
Routing scheme	Shortest path
Path loss exponent (σ)	2-6
Sleep, wake cycle period (s)	30, 30
Credits spent per sleep cycle	10

From (6), this metric is the proportion of all cooperative nodes in the path over all nodes in the path. In this paper, cooperative nodes are nodes which agree to cooperate along a route from an origin node to a destination node. of RL in comparison with CVCP in Figure 4 compares RL and CVCP results as the pause time (PT) is varied. The percentage of cooperative nodes for RL is 97% while that of CVCP is around 80%. The constant trend was because of the number of all cooperative nodes and all nodes in the path were both increasing at the same rate as the pause time increased. As the pause time increased, more paths were discovered which gave rise to an increase in the number of nodes. Figure 5 depicts the results of the two algorithms as a function of λ, the message demand, which governed the network load (NT). In Figure 5, the percentage of cooperative nodes for RL was consistently higher than CVCP.

- **Successful path ratio:** This metric was given by the proportion of successful paths (i.e. message successfully delivered to destination node) over all available paths. Figure 6 illustrates the successful path ratio for RL and CVCP as the pause time (PT) was varied. Note that when nodes were static, the best results were obtained. Furthermore, RL outperformed CVCP as a result of the online learning. This result was in agreement with the percentage of cooperative nodes, since more paths were available therefore more successful message deliveries as the pause time increased. Figure 7 depicts the successful path ratio of RL was still greater than CVCP, though constant, as the network load increased. This was because the proportion of number of successful path and all paths were constantly growing. However the result in Figure 8, the cost function *Ci (b, d)* from equation (4) was added and the path loss exponent (σ) varied. Results showed that successful path ratio was decreased for RL. This was because increasing σ increased the cost, hence a reduction in reward r_i in equation (3).
- **Average reward:** This metric is the average reward obtained over the course of simulation. In Figure 9, the average reward is shown against a varying pause time (PT). Both RL and CVCP obtained the best performance when pause time equal to 300s, this is because the effect from slow node movement. Note that when all nodes are moving slowly, a more diverse range of nodes were available from the slowly changing topology when compared with the static case. For, in static topology obtained less average reward than at pause time equal to 300s. Figure 10 depicts the average reward with varying *λ*. Both RL and CVCP achieved more average reward with greater *λ* since the vital credit is a function of network load (see equation (1)). Figure 11 shows the average reward when the cost function *Ci (b, d)* was included and the path loss exponent (σ) was varied. Results show that RL and attained less average reward as the path loss exponent (σ) increased for all pause times.

VI. CONCLUSIONS

We proposed an incentive-based routing to improve router cooperation in a homogeneous mWSN. The proposed method incorporates the RL and the CVCP to solve the routing problem. We compared these two algorithms to and found that the RL method can achieve up a percentage of cooperative nodes up to 17% more than CVCP. In terms of the successful path ratio and average reward, RL also consistently outperformed CVCP. The results in our experiment suggest that RL can be applied to improve cooperation among routing nodes in comparison to an existing incentive-based algorithm like CVCP.

Our future work will address many challenges associated with incentive-based routing for non-cooperative mWSN, including energy consumption condition and heterogeneity of the sensor and routing nodes to cater more realistic scenarios.

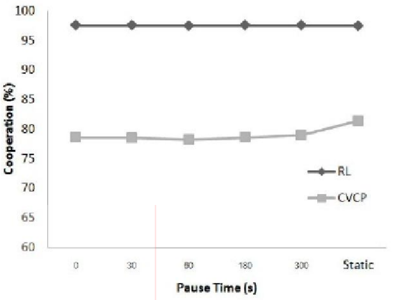

Figure 4. Percentage of cooperative nodes vs pause time.

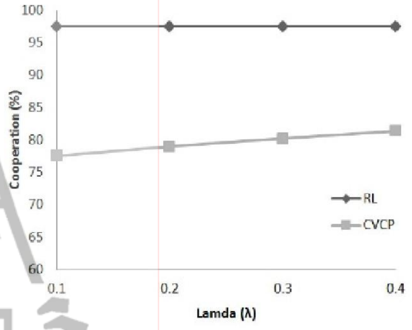

Figure 5. Percentage of cooperative nodes vs network load.

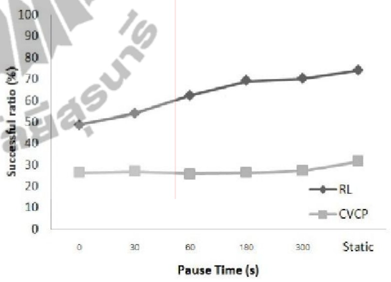

Figure 6. Successful path ratio vs pause time.

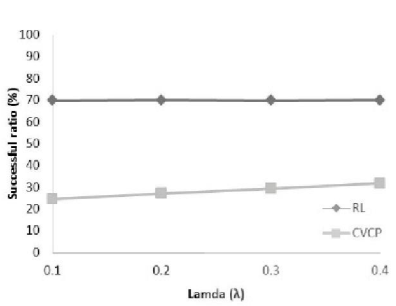

Figure 7. Successful path ratio vs network load.

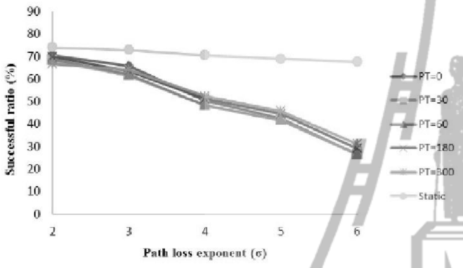

Figure 8. Successful path ratio vs path loss exponent.

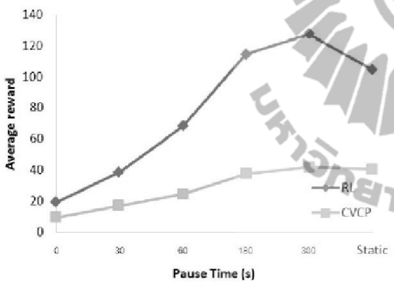

Figure 9. Average reward vs pause time.

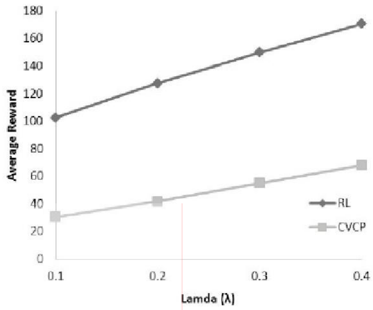

Figure 10. Average reward vs network load.

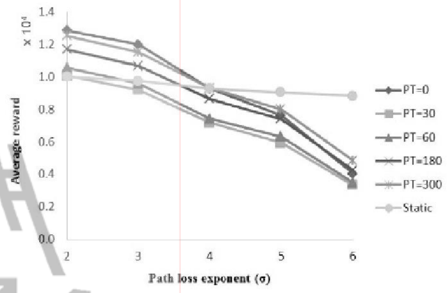

Figure 11. Average reward vs path loss exponent.

References

[1] H. Baldus, K. Klabunde, G. Muesch, "Reliable Set-Up of Medical Body-Sensor Networks", *Proc. EWSN*, vol. 2920, 2004, pp. 353-363.

[2] Field, Marilyn J. (Eds.), Telemedicine: A Guide to Assessing Telecommunication in Health Care, National Academy Press, Washington, D.D., 1996, pp. 26.

[3] N. Lewis, N. Foukia , "An Efficient Reputation-Based Routing Mechanism for Wireless Sensor Networks: Testing the Impact of Mobility and Hostile Nodes", *Privacy, Security and Trust Sixth Annual Conference*, 2008, pp. 151 – 155.

[4] J. Hu, M.P. Wellman, "Nash Q-Leaning for General-Sum Stochastic Games", *Journal of Machine Learning Research 4*, 2003, pp. 1039-1069.

[5] R.S. Sutton and A. G. Barto, *Introduction to Reinforcement Learning,* The MIT Press, Massachussetts, 1998.

[6] R. Machado, S. Tekinay, "A survey of game-theoretic approaches in wireless sensor networks", *Journal of Computer Networks 52*, 2008, pp. 3047–3061.

[7] U. Varshney, "Improving Wireless Health Monitoring Using Incentive-Based Router Cooperation", *IEEE Computer Magazine*, Vol. 41, 2008, pp. 56 – 62.

[8] A. Forster, A.L. Murphy, J. Schiller, K. Terfloth, "An Efficient Implementation of Reinforcement Learning Based Routing on Real WSN Hardware", *Wireless and Mobile Computing*, 2008, pp. 247 – 252.

APPENDIX B

FIGURES FOR ALTERNATIVE NODE PROCESSING RATES

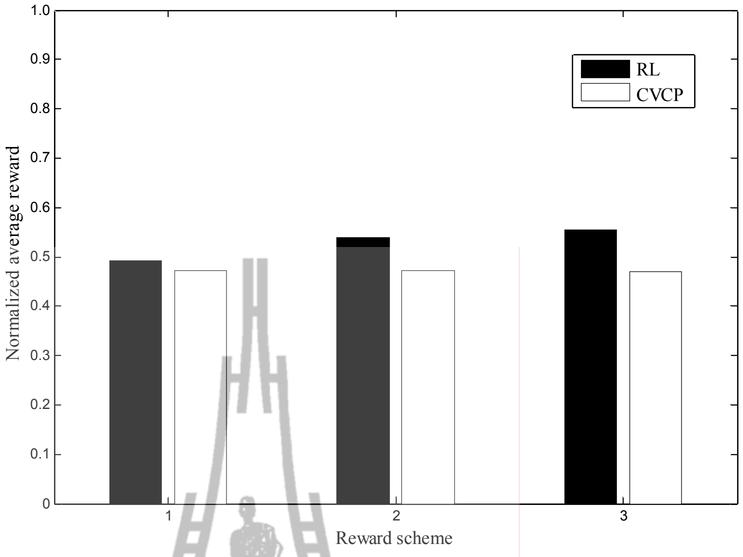

Figure B.1 Normalized average reward of RL and CVCP for 12:24 heterogeneous processing rates with 2 traffic classes.

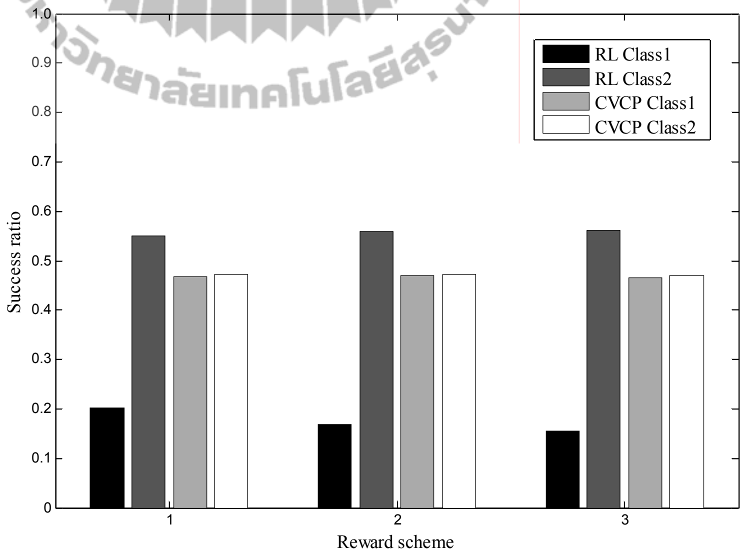

Figure B.2 Success ratio of RL and CVCP for 12:24 heterogeneous processing rates with 2 traffic classes.

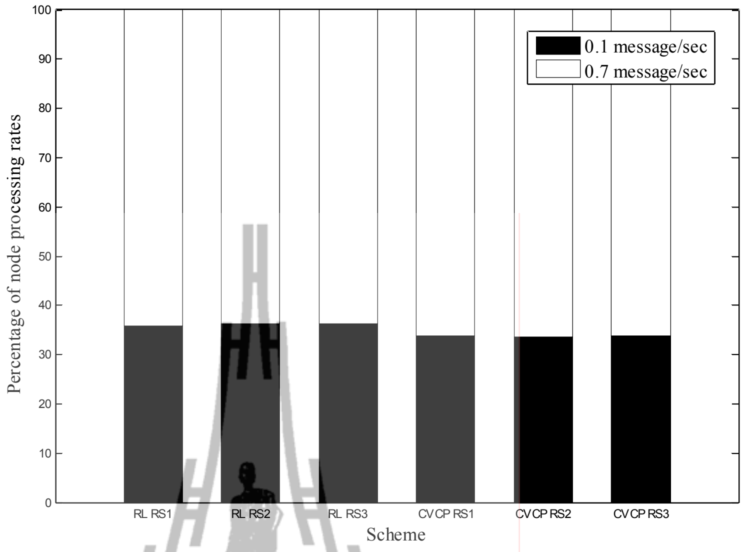

Figure B.3 Percentage of node processing rates of RL and CVCP for 12:24 heterogeneous processing rates with 2 traffic classes.

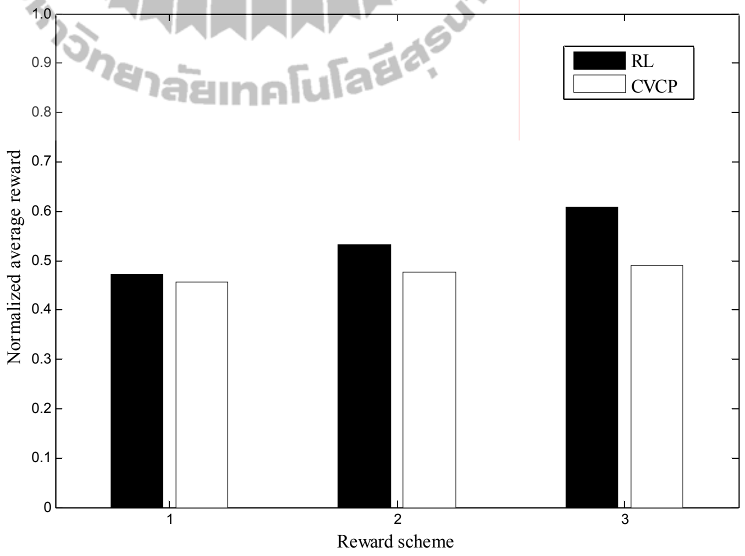

Figure B.4 Normalized average reward of RL and CVCP for 12:24 heterogeneous processing rates with 3 traffic classes.

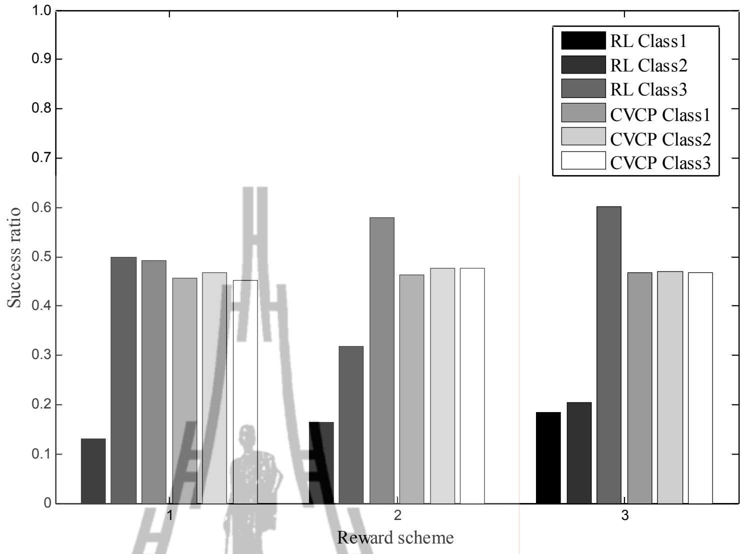

Figure B.5 Success ratio of RL and CVCP for 12:24 heterogeneous processing rates with 3 traffic classes.

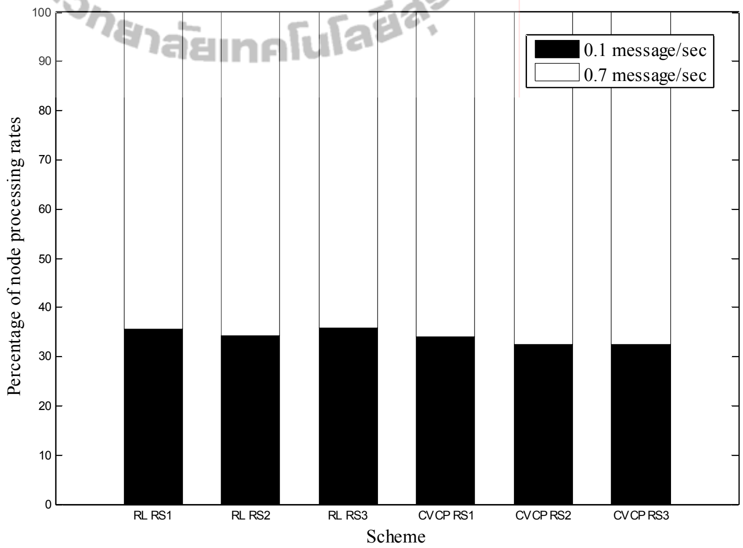

Figure B.6 Percentage of node processing rates of RL and CVCP for 12:24 heterogeneous processing rates with 3 traffic classes.

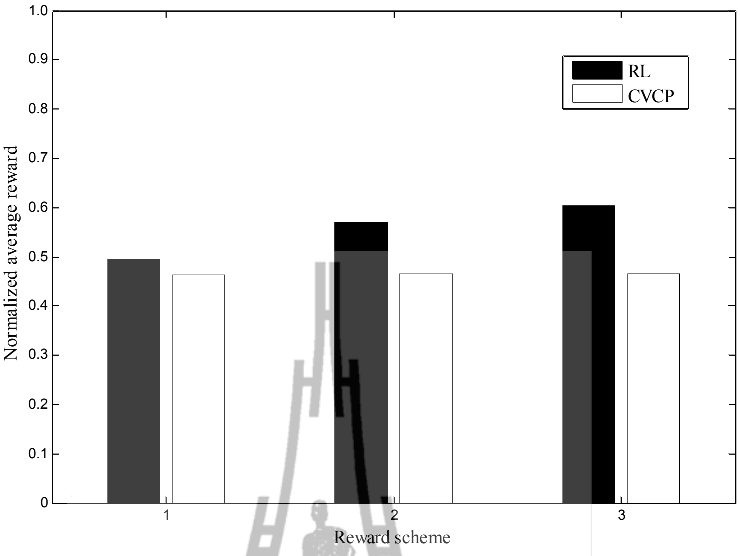

Figure B.7 Normalized average reward of RL and CVCP for 12:24 heterogeneous processing rates with 4 traffic classes.

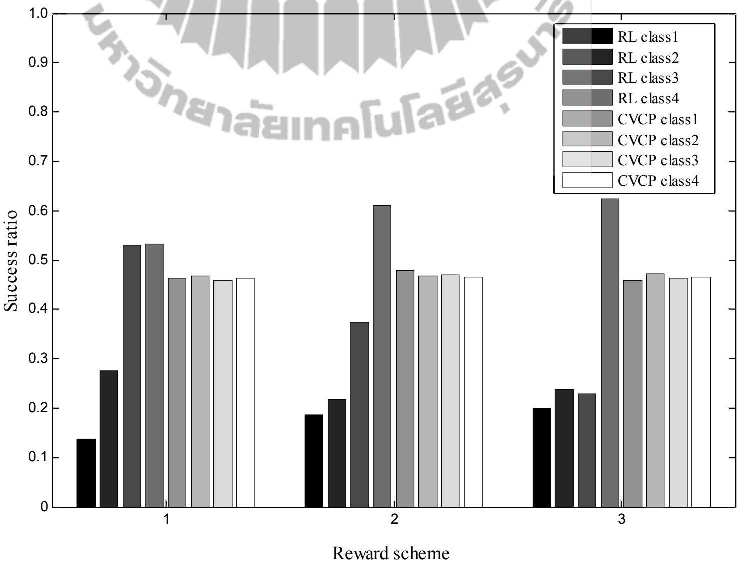

Figure B.8 Success ratio of RL and CVCP for 12:24 heterogeneous processing rates with 4 traffic classes.

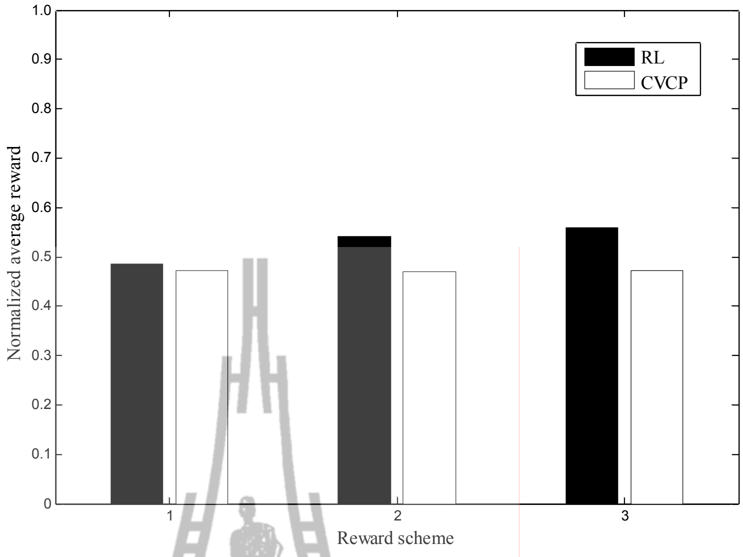

Figure B.9 Normalized average reward of RL and CVCP for 4:12:20 heterogeneous processing rates with 2 traffic classes.

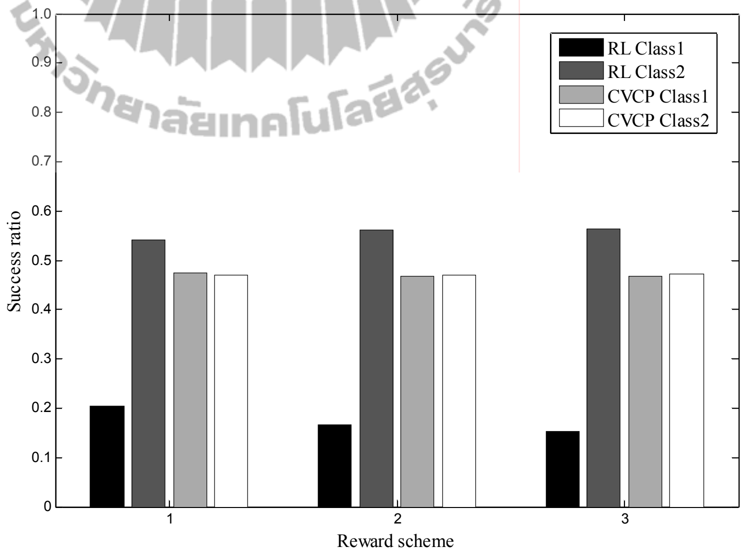

Figure B.10 Success ratio of RL and CVCP for 4:12:20 heterogeneous processing rates with 2 traffic classes.

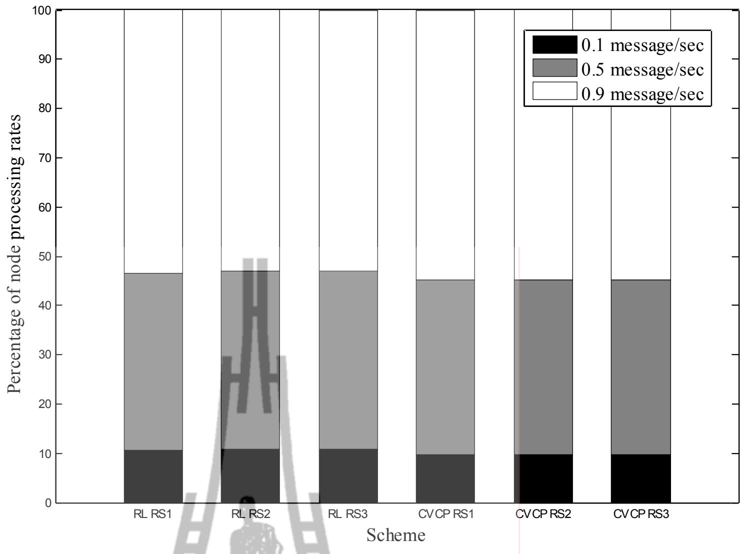

Figure B.11 Percentage of node processing rates of RL and CVCP for 4:12:20 heterogeneous processing rates with 2 traffic classes.

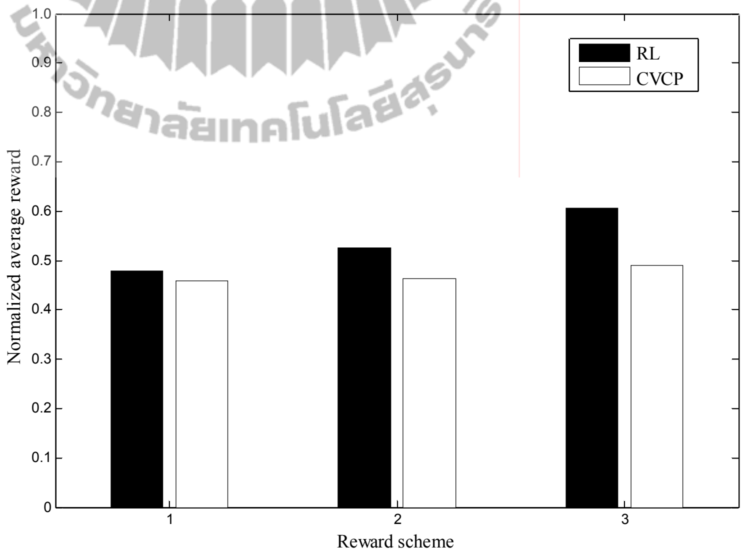

Figure B.12 Normalized average reward of RL and CVCP for 4:12:20 heterogeneous processing rates with 3 traffic classes.

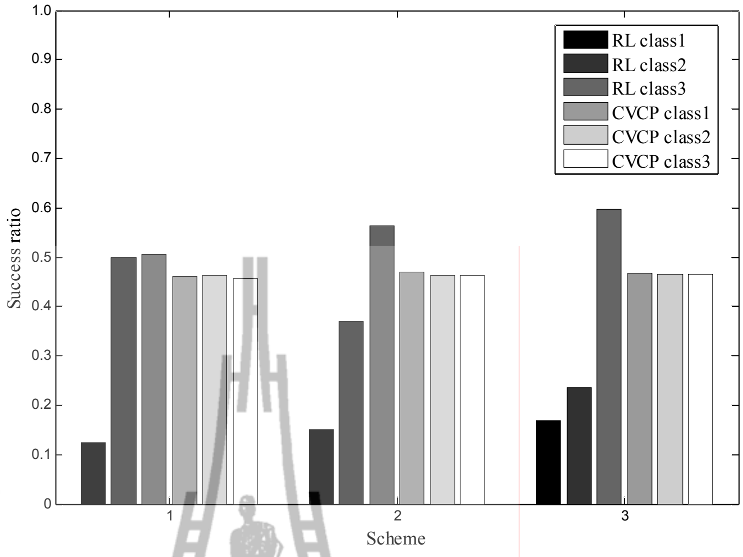

Figure B.13 Success ratio of RL and CVCP for 4:12:20 heterogeneous processing rates with 3 traffic classes.

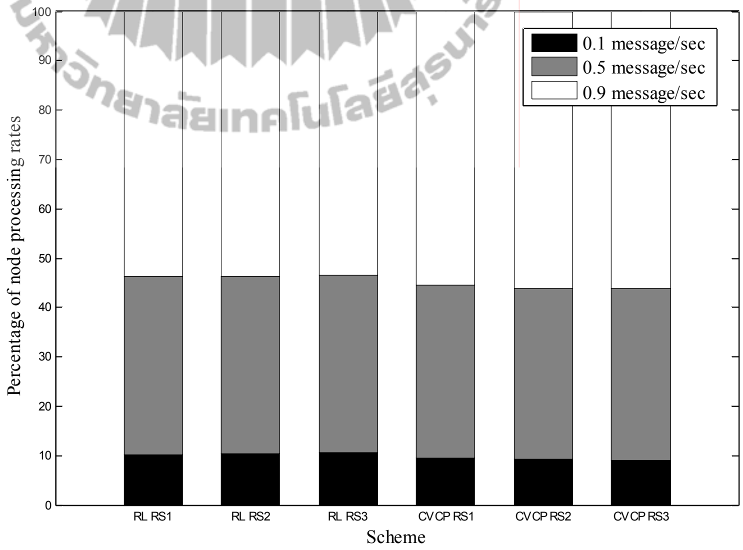

Figure B.14 Percentage of node processing rates of RL and CVCP for 4:12:20 heterogeneous processing rates with 3 traffic classes.

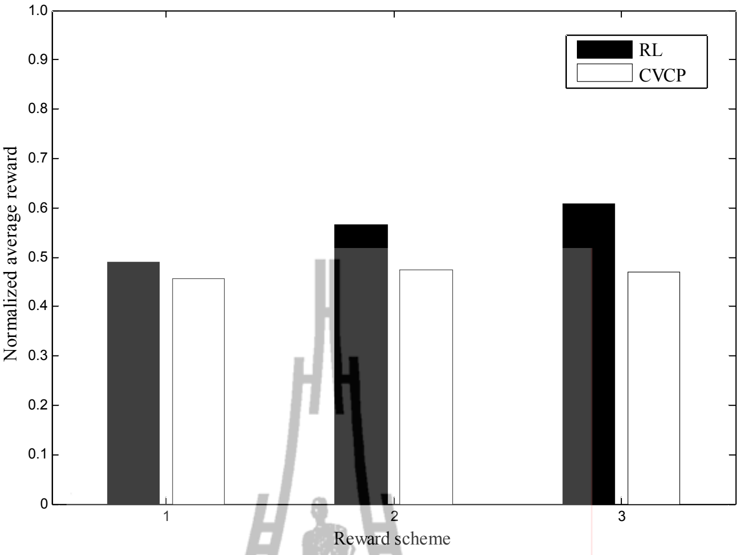

Figure B.15 Normalized average reward of RL and CVCP for 4:12:20 heterogeneous processing rates with 4 traffic classes.

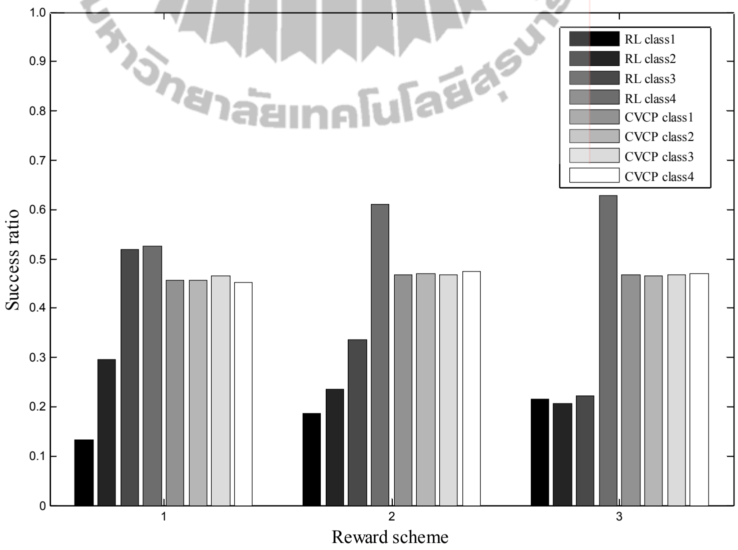

Figure B.16 Success ratio of RL and CVCP for 4:12:20 heterogeneous processing rates with 4 traffic classes.

BIOGRAPHY

Mr. Chanon Rittong was born on January 16, 1982 in Phatthalung province, Thailand. He finished high school education from Phatthalung School, Phathalung province. He received his Bachelor’s Degree in Engineering (Telecommunication) from Suranaree University of Technology in 2005. After graduating, he worked for SONY Technology Thailand for 3 years. In 2008, he began studying for a Master’s degree in the Telecommunication Engineering Program, Institute of Engineering, Suranaree University of Technology. During Master’s degree education, he was a visiting researcher and studied at Universiti Putra Malaysia, Malaysia.