การเลือกเส้นทางที่ใช้พลังงานอย่างมีประสิทธิภาพใน
เครือข่ายเคลื่อนที่แบบแอดฮอคโดยใช้วิธีรีอินฟอร์สเมนท์เลิร์นนิ่ง

นางสาววิภาดา นฤพิพัฒน์

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2550

ENERGY-EFFICIENT ROUTING IN MOBILE AD HOC NETWORKS USING REINFORCEMENT LEARNING

Wibhada Naruephiphat

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 2007

ENERGY-EFFICIENT ROUTING IN
MOBILE AD HOC NETWORKS
USING REINFORCEMENT LEARNING

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. Rangsan Tongta)
Chairperson

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

____
(Dr. Paramate Horkaew)
Member

____
(Prof. Dr. Pairote Suttayatham)
Vice Rector for Academic Affairs

____
(Assoc. Prof. Dr. Vorapot Khompis)
Dean of Institute of Engineering

วิภาดา นฤพิพัฒน์ : การเลือกเส้นทางที่ใช้พลังงานอย่างมีประสิทธิภาพในเครือข่าย
เคลื่อนที่แบบแอดฮอคโดยใช้วิธีรีอินฟอร์สเมนท์เลิร์นนิ่ง (ENERGY-EFFICIENT
ROUTING IN MOBILE AD HOC NETWORKS USING REINFORCEMENT LEARNING).
อาจารย์ที่ปรึกษา : ผศ. ดร. วิภาวี หัตถกรรม, 86 หน้า

งานวิจัยนี้นำเสนอ วิธีการเลือกเส้นทางที่ใช้พลังงานอย่างมีประสิทธิภาพในเครือข่ายเคลื่อนที่แบบแอดฮอค โดยการหาจุดสมดุลของวิธีการเลือกเส้นทางที่มีวัตถุประสงค์ขัดแย้งกันระหว่าง การเลือกเส้นทางที่ยืดอายุของเครือข่าย และ การเลือกเส้นทางที่ใช้พลังงานน้อย

รูปแบบทั่วไปของเครือข่ายเคลื่อนที่แบบแอดฮอคประกอบด้วยโหนดซึ่งอาศัยพลังงานแบตเตอรี่สำหรับการใช้งานและติดต่อกับโหนดอื่นในเครือข่าย ดังนั้นการเลือกเพื่อการใช้พลังงานอย่างมีประสิทธิภาพจึงมีความจำเป็นอย่างยิ่ง โดยการเลือกเส้นทางที่พิจารณาพลังงานนั้น สามารถแบ่งได้โดยทั่วไปเป็น 2 วิธีคือ วิธีการเลือกเส้นทางเพื่อยืดอายุของเครือข่าย ซึ่งกระจายการใช้งานยังโหนดต่างๆ และสามารถเพิ่มอายุของเครือข่ายได้นานขึ้นแต่ไม่สามารถลดการใช้พลังงานให้ต่ำลงได้ วิธีการที่สองคือการเลือกเส้นทางที่ใช้พลังงานน้อย สามารถลดการใช้พลังงานลงได้แต่โหนดที่ถูกใช้งานหนัก จะออกจากเครือข่ายเร็วขึ้นเนื่องจากระดับพลังงานแบตเตอรี่หมดลง ดังนั้นจะเห็นว่ามีข้อแลกเปลี่ยนของทั้งสองวิธี วิทยานิพนธ์นี้จึงมีจุดประสงค์ที่จะระบุปัญหาการหาเส้นทางที่มีสมดุลที่เหมาะสมที่สุดร่วมกันระหว่างการใช้พลังงานและอายุเครือข่ายในเครือข่ายเคลื่อนที่แบบแอดฮอคที่มีรูปร่างเครือข่ายพลวัติ วิทยานิพนธ์นี้มีองค์ความรู้หลักสองประการ:

องค์ความรู้ประการแรก คือการกำหนดปัญหาการเลือกเส้นทางในเครือข่ายเคลื่อนที่แบบแอดฮอคให้เป็นกระบวนการการตัดสินใจแบบมาคอฟ (Markov decision process) ซึ่งจุดมุ่งหมายในการปรับปรุงการเลือกเส้นทางเพื่อหาเส้นทางที่ให้ค่าเฉลี่ยมูลค่าที่ต่ำที่สุด โครงสร้างมูลค่านี้กำหนดให้เป็นฟังก์ชันของพลังงานที่ถูกใช้ไปและระดับแบตเตอรี่ที่เหลืออยู่ รวมทั้งจำนวนโหนดที่ยังคงอยู่และสัดส่วนของแพกเก็ตที่ส่งสำเร็จ เพื่อให้ได้นโยบายการเลือกเส้นทางที่ดี และมีสมดุลข้อแลกเปลี่ยน

องค์ความรู้ประการที่สอง คือ การประยุกต์เทคนิครีอินฟอร์สเมนท์เลิร์นนิ่ง (Reinforcement learning) ที่แบ่งการเรียนรู้ออกเป็นเอพพิโซด (episode) ด้วยวิธีการที่เรียกว่า ออนโพลิซี มอนติ คาร์โล (On-policy Monte Carlo หรือ ONMC) เพื่อหาผลคำตอบของกระบวนการการตัดสินใจแบบมาคอฟที่กำหนดขึ้น วิธีการ ออนโพลิซี มอนติ คาร์โลได้ถูกเลือกเนื่องจากการเลือกเส้นในเครือข่ายเคลื่อนที่แบบแอดฮอคมีรอบการทำงานในลักษณะเอพพิโซดโดยธรรมชาติอยู่แล้ว จากผลการทดลองพบว่าวิธีการที่นำเสนอสามารถลดมูลค่าในระยะยาวได้สูงสุด 37% เมื่อเปรียบเทียบกับวิธีการหาเส้นทาง

ที่ใช้พลังงานอย่างมีประสิทธิภาพที่มีอยู่เดิม โดยมูลค่าระยะยาวดังกล่าวคือ ฟังก์ชันวัตถุประสงค์ซึ่งบอกค่าแลกเปลี่ยนที่เหมาะสมในการหาจุดสมดุลของการเลือกเส้นทางในระยะยาว

สาขาวิชาวิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา____________________

ปีการศึกษา 2550 ลายมือชื่ออาจารย์ที่ปรึกษา______________

WIBHADA NARUEPHIPHAT : ENERGY-EFFICIENT ROUTING IN MOBILE AD HOC NETWORKS USING REINFORCEMENT LEARNING. THESIS ADVISOR: ASST. PROF. WIPAWEE HATTAGAM, Ph.D. 86 PP.

MOBILE AD HOC NETWORK (MANET)/ ENERGY-EFFICIENT ROUTING/ REINFORCEMENT LEARNING/ MARKOV DECISION PROCESS (MDP)/ ON-POLICY MONTE CARLO (ONMC)

This research proposes an energy-efficient path selection algorithm which aims at balancing the contrasting objectives of maximum network lifetime routing and minimal energy consumption routing in mobile ad hoc networks (MANETs).

A typical mobile ad hoc network consists of nodes that are usually battery operated. Hence, energy-efficient routing is a critical issue. There are two approaches broadly suggested for energy-aware route selection protocols. Firstly, the maximum lifetime routing protocols balance the load among nodes and can prolong the network lifetime, but do not decrease the total energy consumption. Secondly, the minimum energy consumption routing protocols aim at reducing the network energy consumption, but the nodes exhaustively used along the selected paths *die* very soon. Hence, there exists a tradeoff between the two approaches. The underlying aim of this thesis is to address the problem of jointly optimizing the energy consumption and network lifetime in MANETs with dynamic topology. There are two main contributions in this thesis:

The first contribution is the formulation of the energy-efficient path selecting problem in MANETs as a Markov decision process (MDP), whose goal is to find a sequence of path selection that minimizes the expected accumulated cost for the system. The cost structure is a function of the energy consumed, the residual energy as well as the number of alive nodes and the ratio of successfully delivered packets, so as to achieve a good path selection policy which balances the tradeoffs.

The second contribution is the application of a reinforcement learning method based on sample episodes, called the on-policy Monte Carlo (ONMC) method, to solve for a solution to the formulated MDP. The ONMC method is chosen due to the inherent episodic behavior of the routing process in MANETs. The simulation results show that the proposed algorithm can reduce the long-term cost, which is a function that depicts the optimal tradeoff balance in the long run, by up to 37% when compared to existing well-known energy-efficient routing schemes.

School of Telecommunication Engineering Student's Signature________________

Academic Year 2007 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 wish 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. She provided many insightful ideas and comments throughout my masters degree.

I would like to thank the lecturers in the School of Telecommunication Engineering, Asst. Prof. Dr. Rangsan Wongsan, Asst. Prof. Dr. Chutima Prommak, Asst. Prof. Dr. Chanchai Tongsopa, Asst. Prof. Dr. Prayoth Kumsawatand, Dr. Peerapong and Dr.Monthippa Uthansakul, Dr. Somsak Vanit-Anunchai many others for suggestions and all their help. I would also like to thank Asst. Prof. Dr. Rangsan Tongta, Dr. Paramate Horkaew for accepting to serve in my committee.

I am grateful to the engineers, technicians and personnel at the Telecommunication Laboratory for their suggestions and help. My special thanks goes to Mr. Prapol Jarataku for his full support.

My sincere appreciation goes to Ms. Maneerat Tumpong for her valuable support in documents administration 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.

Wibhada Naruephiphat

TABLE OF CONTENTS

Page

CHAPTER

TABLE OF CONTENTS (Continued)

Page

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

SYMBOLS AND ABBREVIATIONS

MANET	Mobile ad hoc network
RL	Reinforcement learning
ONMC	On-policy Monte Carlo
MP	Markov property
MDP	Markov decision process
CMMBCR	Conditional max-min battery capacity routing
Q^{π}	Action-value function
s	State
a	Action
g	Reward
S	State space
A	Action space
π	Policy
E_{TOT}	Total energy consumption
E_{TX}	Transmitting energy consumption
E_{RX}	Receiving energy consumption
$E_{TX}\left(i\right)$	Energy consumption at node *i*
$Battlevel\left(i\right)$	Residual battery level at node *i*
E_{elec}	Energy expended in the radio electronics

SYMBOLS AND ABBREVIATIONS (Continued)

ε_{fs}	Constant parameter in free space model
P_l	Total energy consumption along path *l*
B_l	Battery level at the bottleneck node along path *l*
B_{init}	Initial battery level
A_0	Action space
l_b	Max-min routing
l_e	Minimum energy routing
l_c	Minimum cost routing
$c(s,a)$	Cost at state-action pair (s, a)
$C(X)$	Long-term cost for algorithm X
E_{avg}	Average network energy consumption per node
R_{DP}	Ratio of successfully delivered packets
E_{DP}	Energy consumed per delivered packet
γ	Threshold value
$\lvert S_i \rvert$	Cardinality of the state space
$\lvert A_i \rvert$	Cardinality of the action space

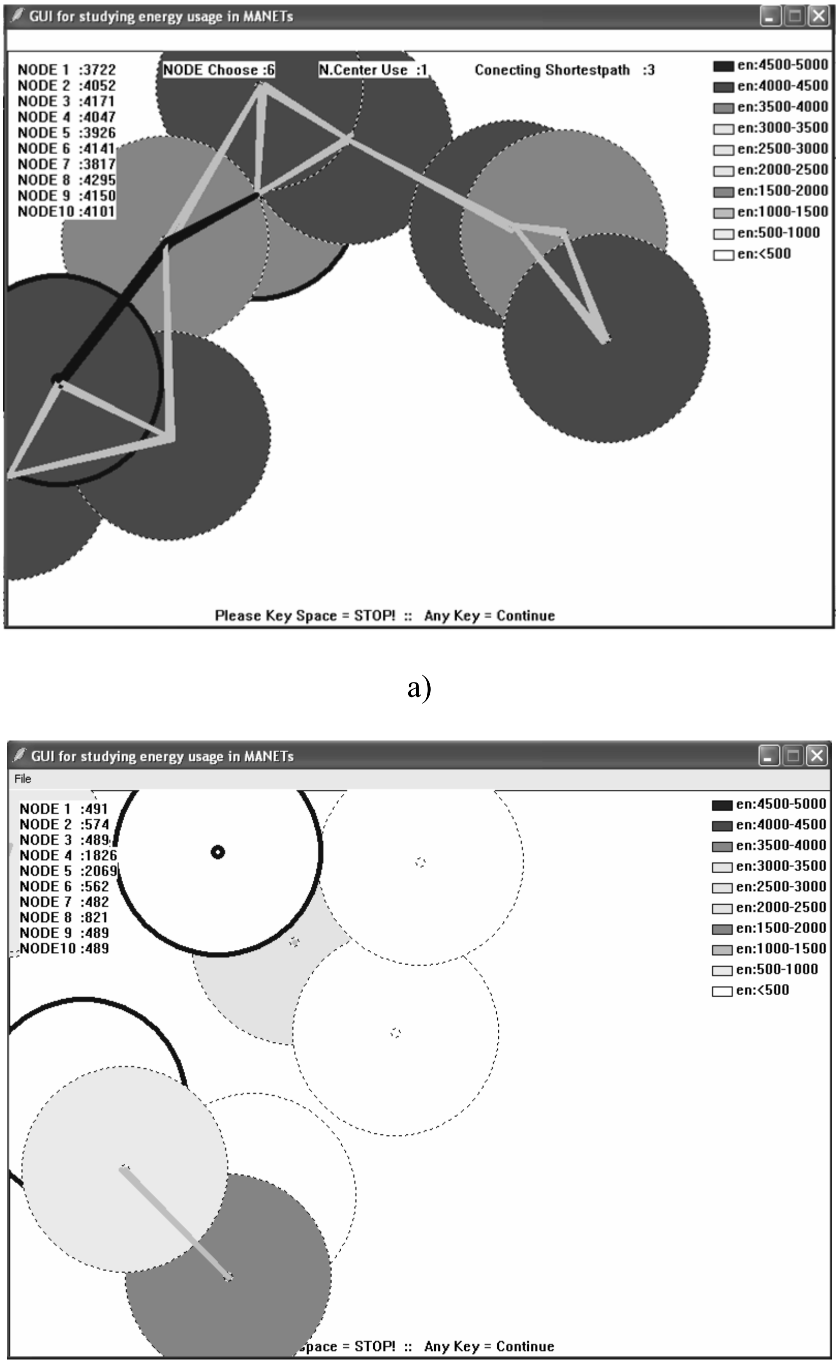
CHAPTER I

INTRODUCTION

This chapter introduces the background of mobile ad hoc networks and highlights the significance of the energy-efficient routing problem in mobile ad hoc networks. It also presents the motivation for applying reinforcement learning to achieve energy-efficient routing which is the main focus of this thesis.

1.1 Significance of the Problem

A mobile ad hoc network (MANET) is a communication network where all nodes cooperatively maintain network connectivity without a centralized infrastructure. Since all nodes in the MANET can move freely, such network is generally characterized by bandwidth-constrained, variable capacity links and unpredictable topology. Each node has a limited transmission range. A source node communicates with a destination node out of its transmission range through intermediate nodes. Thus, every node in the network is capable of functioning as a mobile router which participates in forwarding data packets and as a host which runs applications. Figure 1.1 a)-b) illustrates an example of a MANET and its connectivity at different time instants.



GUI for studying energy usage in MANETs

NODE 1 :3722
NODE 2 :4052
NODE 3 :4171
NODE 4 :4047
NODE 5 :3926
NODE 6 :4141
NODE 7 :3817
NODE 8 :4295
NODE 9 :4150
NODE10 :4101

NODE Choose :6

N.Center Use :1

Conecting Shortestpath :3

■ en:4500-5000
■ en:4000-4500
■ en:3500-4000
□ en:3000-3500
□ en:2500-3000
□ en:2000-2500
■ en:1500-2000
■ en:1000-1500
□ en:500-1000
□ en:<500

Please Key Space = STOP! :: Any Key = Continue

a)

GUI for studying energy usage in MANETs

File

NODE 1 :491
NODE 2 :574
NODE 3 :489
NODE 4 :1826
NODE 5 :2069
NODE 6 :562
NODE 7 :482
NODE 8 :821
NODE 9 :489
NODE10 :489

■ en:4500-5000
■ en:4000-4500
■ en:3500-4000
□ en:3000-3500
□ en:2500-3000
□ en:2000-2500
■ en:1500-2000
■ en:1000-1500
□ en:500-1000
□ en:<500

pace = STOP! :: Any Key = Continue

b)

Figure 1.1 MANET path connectivity between a source-destination pair a) when all 10 nodes have high residual battery levels. The dark lines depicts the selected path. b) when most of the nodes have depleted their batteries level.

1.1.1 Application of MANETs

The essential characteristics of a mobile ad hoc network are infrastructureless, self-organizing and wireless communication. As a result, MANETs are suitable for communication in the following scenarios.

1) Military applications: Operations requiring soldiers, tanks, or battle ships to mobilize freely in the battlefield without any restrictions imposed by wired communication devices. These applications should thus be self-configuring, independent of any centralized control station.

2) Commercial applications: The lack of infrastructure in ad hoc networks is a motivating factor for deployment in commercial applications as it reduces the cost of infrastructure investments. An example of this application would be a conference room with participants communicating with each other.

3) Emergency rescue applications: Since a mobile ad hoc network can be set up at any place, it can substitute the original primary communication for rescue operations in networks which have been destroyed by a disaster. This network is therefore useful for natural disaster scenarios.

1.1.2 Significance of Energy-Efficient Routing Protocols

Since the topology of MANETS change dynamically due to node mobility, routing protocols are necessary to forward data packets. In the literature, routing protocols such as Perkins, Royer, Das and Marina, (2004) summarizes a description of Dynamic Source Routing (DSR) and Ad hoc On-demand Distance Vector routing (AODV), then compares the performance of the two prominent on-demand routing protocols for mobile ad hoc networks. (Geetha, Aithal, and

ChandraSekaran, 2006), studies the Dynamic Source Routing (DSR) , Ad hoc On-demand Distance Vector routing (AODV) , Destination-Sequenced Distance-Vector (DSDV) and Temporally Ordered Routing Algorithm (TORA) which are routing protocols normally used in ad hoc networks. They have analyzed the effect of mobility over two ad hoc routing protocols, namely, AODV and DSDV.

However, the aforementioned works do not explicitly deal with energy utilization in MANETs. A typical mobile ad hoc network consists of nodes that are usually battery operated devices such as laptops, PDAs or sensor nodes. Thus, each node carries out its individual processing as well as acts as a forwarding node (router). Hence, energy consumption is a critical issue. There are routing protocols in recent works which consider power as one of the cost metrics for MANETs. In general, energy-aware routing protocols can be categorized into two approaches, namely, 1) the maximum network lifetime approach, 2) the minimum energy consumption approach.

1.1.2.1 The Maximum Network Lifetime Approach

The maximum network lifetime routing protocols focus on balancing energy usage among the nodes by avoiding overutilized nodes while selecting a routing path. Therefore, nodes are used as intermediate nodes equally, so that no nodes are heavily used and quick depletion of battery level is avoided. Since the network lifetime is defined as the time at which the first node in the network drains out of battery, this routing protocol therefore maximizes the network lifetime.

1) **MME :** The Max-Min Energy algorithm (Venugopal, Bartos, Michael and Sai, 2003) is proposed to balance the Dynamic Source Routing (DSR)

protocol by selecting the route which contains the node with the highest remaining battery level. In particular, the DSR protocol finds the optimal route by searching the node with minimum remaining battery level in each route. Then the minimum remaining battery level of each route is compared. The route with the highest minimum remaining battery level is chosen as the optimal route.

2) **AODV-*energ* :** The Ad-hoc On-Demand Distance Vector with a simple speed-based energy consumption mechanism is presented by Romdhani and Bonnet (2004). The algorithm aims to maximize the network lifetime by selecting the best path with the maximum mean cost, where the mean cost is defined as $cost_{mean} = \frac{\sum cost_{res\_life}}{number_{hops}}$, $cost_{res\_life}$ is the ratio of the remaining battery level over the speed of decreasing battery level within a period of time in each node, and $number_{hops}$ is the number of traversed hops along a given path. Such cost function favors paths with high mean costs since a large summation of $cost_{res\_life}$ and few $number_{hops}$ constitute to short paths that contain nodes with higher battery levels.

3) **PAOD :** The power-aware on-demand routing protocol (Wang, Xu, Chen and Wu, 2004) selects routes based on a cost function which represents the shortest path and the maximum lifetime. In particular, their cost function comprises the number of intermediate nodes along a path and the maximum of the minimum residual battery in a path. In effect, the protocol tries to select a path which maximizes the battery level and has the shortest path.

In the aforementioned works, it can be observed that the maximize network lifetime algorithms either use path costs as a function of residual battery level for all nodes on the routing path, or avoid the route with nodes which have the least battery level. However, these algorithms may not decrease the total energy consumption because they only focus on battery level. Therefore, the path selected may not be the minimum energy consumption path. Furthermore, the algorithms try to use nodes fairly, resulting in a path selection that differs from the previously selected paths, hence making it difficult to control energy consumption.

1.1.2.2 The Minimum Energy Consumption Approach

The minimum energy consumption routing protocols select paths that minimize the energy consumption required to forward a data packet from a source to a destination.

1) **PCR :** The Power Control Routing (PCR) presented by Tsudaka, Kawahara, Matsumoto and Okada (2004) improves the network capacity and decreases the energy consumption. The PCR controls the energy consumption by considering the link weight defined by the number of nodes affected by interfered communication resulted from limited energy consumption. Each intermediate node has a link weight. The weight of the route is the summation of link weights of all intermediate nodes along a path. PCR selects the path with the smallest route weight. Under certain assumptions, the selected path requires minimum energy consumption in each intermediate node. The reason is that the link weight is small when energy consumption is low. In particular, each node limits energy usage by limiting its transmission range which covers its neighbor nodes.

2) **DPC-AODV :** The Distributed Power Control is applied to the AODV routing protocol to achieve energy savings in (Bergamo, Maniezzo and Travasoni, 2003). The algorithm uses a hop-by-hop minimized energy consumption path selection, therefore attaining paths with minimum energy consumption.

3) **MPR :** The Minimum Power Routing protocol (Singh, Woo, and Raghavendra, 1998) is a routing algorithm based on minimizing the amount of power required to send packets from a source to a destination node. The problem is stated as $Minimize \sum_{i \in path} P(i,i+1)$ where *P(i,i+1)* denotes the energy consumption for transmitting between two nodes along some path.

In the aforementioned works, it can be observed that algorithms with minimum energy consumption approach consider intermediate nodes that transmit with low energy consumption, so energy consumption along path is minimized. However, these algorithms do not use nodes fairly as a result. Some nodes lying frequently in a minimum energy consumption path can be used heavily and tend to *die out* very soon because of battery level exhaustion. As a result, the minimum energy consumption approach cannot achieve long network lifetime.

The algorithms in section 1.1.2.1 and 1.1.2.2 show that there is a tradeoff between the maximum network lifetime and the minimum energy consumption approaches. There is no clear consensus that any approach is suitable for all scenarios because the maximum network lifetime approach can maximize the network lifetime but does not decrease the total energy consumption, while the minimum energy consumption approach can save energy consumption but the nodes along the path disconnect very soon.

1.1.2.3 Tradingoff Both Approaches

To address the tradeoff between the two approaches, many works attempt to integrate the advantages of both the maximum network lifetime and the minimum energy consumption approach protocols by reducing the energy consumption and increasing the network lifetime.

1) **CMMBCR :** The Conditional Max-Min Battery Capacity Routing (Toh, 2001), is a conditional strategy power-aware routing protocol. The basic idea is that when all nodes in some possible routes between a source to a destination have sufficient remaining battery capacity *above* a pre-specified threshold, the route with the minimum total energy consumption among these route is selected. Otherwise, the path which maximizes the minimum residual battery level is selected. The value of the threshold parameter (γ) determines the node expiration behavior. If the threshold (γ) is low, the minimum energy consumption is preferred. On the other hand, a high value of threshold (γ) prefers the maximum-minimum residual battery path and gives a longer network lifetime.

2) **Max-Min zP_{min} :** The algorithm in (Aslam, Li, and Rus ,2003) selects a path that uses at most $z * P_{\min}$ energy, where z is a parameter which controls the path selection (i.e. $1 \le z \le \infty$). In particular, the route that maximizes the minimum residual energy fraction (i.e. the ratio of the battery remaining after route selection over the initial battery level) is selected as long as such path consumes no more than zP_{min} energy, where P_{min} is the total energy consumed on the minimum energy route. An important factor in this algorithm is the parameter *z* that measures the tradeoff between max-min path and minimum energy consumption path. If *z* is

low, the minimum energy consumption path is favorable. Increasing *z* implies favoring the maximum-minimum residual battery path.

Note that the CMMBCR and Max-Min zP_{min} both incorporate the benefits of the maximum network lifetime and minimum energy consumption approaches by varying its parameter value. The actual values of *z* and γ could depend on network size and the mobility profile of each node. Hence, it is difficult to determine the value of *z* and γ suitable for each scenario. There exists other algorithms which propose the use of cost functions instead of relying on threshold parameters, to achieve an optimum between the two approaches.

3) **ESDSR :** The Energy Saving Dynamic Source Routing (ESDSR) is proposed in (Tarique, Tepe and Naserian, 2005). The Dynamic Source Routing (DSR) protocol is modified to acheive energy awareness by employing a specific cost function at each node. In particular, node *i* calculates its ratio of current residual battery level and the energy consumption $\left(B_i/e_i\right)$. Some path *l* is selected if it attains the maximum path cost $C\left(R\right)=max\left(R_l\right)$ among all available paths, where $R_l=min\left(B_i/e_i\right)$ and node *i* is an intermediate node in path *l*.

4) **PCSR :** The Power Control Source routing protocol (PCSR) based on DSR (Sheu, Lai and Chao, 2004), selects a path by depending on a minimum cost function and some parameter T_{hold}, where T_{hold} is a parameter value used for comparing the minimum residual battery level of the nodes in each path. In particular, the destination node first checks the minimum remaining battery level of each route. If the least battery level route is greater than T_{hold} , the route that has the minimum cost will be selected. Otherwise, the route that has the maximum residual

battery level will be selected. The cost function for some node *i* is defined as, $cost = \frac{B_i(0)}{B_i(t)} \times e_{ij}$, where *B*$_i$*(0)* is the initial battery level, *B*$_i$*(t)* is the residual battery level at time t and e_{ij} is the energy consumed for transmitting on link (*i* , *j*).

5) **PERRA :** The Power Efficient Reliable Routing protocol for mobile Ad-hoc networks (Kwak, Kim and Yoo, 2004) employs a new cost to select paths based on the minimum residual battery of nodes along the path, the energy consumption along the path and the path's stability in accordance with the node mobility. The algorithm selects the path with the smallest total cost given by

$$\begin{aligned} Total\_cost = {} & w_1 \times \left[\sum\nolimits_{\forall i \in l} E_{TX}(i) + h \times E_{proc} \right] \\ & - w_2 \times min\left[\frac{min(B_i)}{h \times \left(E_{TX}(i) + E_{proc}\right)}, max(B_i) \right] \\ & - w_3 \times min\left[min(PLT), max(PLT) \right], \end{aligned} \tag{1.1}$$

where w$_1$, w$_2$, w$_3$ are weight factors which must sum up to unity. The first term on the right hand side of the equation refers to the energy consumption and hop transfer (h). The second term refers to the residual battery level, and the last term refers to the path lifetime.

Unlike the works in (Toh, 2001) and (Aslam, Li, and Rus ,2003) where path selection decisions are controlled by variation of some threshold parameter, the works in (Tarique, Tepe and Naserian, 2005), (Sheu, Lai and Chao, 2004) and (Kwak, Kim and Yoo, 2004) use cost functions which combine components of the energy consumption and the residual battery level in various formats. However, all of these works have a common feature, that is, they incorporate both the benefits of the maximum lifetime and minimum energy approaches.

However, the actual value of such parameters will depend on network size and the mobility profile of each node. For example, if the network size is large and sparse, the network connectivity decreases because the number of neighbors for relaying is insufficient due to limited transmission range (Bergamo, Maniezzo and Travasoni, 2003). In such scenario, we may favor minimum energy consumption routing to avoid wasting energy in delivering a packet. The parameter value should give priority to minimizing the energy consumption rather than maximizing the network lifetime. In low mobility scenarios, it is possible that some nodes in network are used heavily as intermediate nodes since the position of certain nodes change slowly. In this scenario, the residual battery level should be considered, and the parameter value should be treated to maximize the network lifetime. However, in reality, it is difficult to know the optimal threshold parameter value setting which is suitable for each scenario. On the other hand, algorithms employing the combined cost functions of both energy consumption and residual battery level, can smoothly adjust the policy of path selection and avoid the problem of parameter value settings. Minimum cost routing schemes were also proposed where the sum of the link cost was used to deflect traffic from high cost routes. Link capacity cost of the form $c_{ij} = e_{ij} / B_i$, where B_i is the residual energy at node *i*, and e_{ij} is the communication energy cost for link (*I, j*), has shown good performance in terms of network lifetime (Basagni, Conti, Giordano and Stojmenovic, 2004). The normalized link capacity cost of the form $c_{ij} = e_{ij}\left(B_{init} / B_i\right)$, performed even better than other combined cost metrics (Basagni, Conti, Giordano and Stojmenovic, 2004). Such method appears to perform well in terms of maximizing the network lifetime. However, it is not clear whether there exists an optimal energy tradeoff balance in the long run.

In response to these outstanding issues, this thesis proposes an energy-efficient path selection algorithm which aims at balancing the contrasting objectives of maximizing network lifetime and minimizing energy consumption routing in MANETs with dynamic topology. This thesis applies a reinforcement learning (RL) technique called on-policy Monte Carlo (ONMC) (Sutton and Barto, 1998). In a dynamic environment, the proposed algorithm can learn to select near-optimal decisions to achieve a particular goal. RL consists of states (i.e. information of environment) and actions (i.e. an agent's decision). Before a decision is made to select a path, the agent will consider the state of the environment. In this thesis, the environment state is the information of energy consumption and battery levels of relevant nodes. Such information would help an agent (i.e. source node) to select paths suitable for different scenarios. Once a path is selected, a cost is assigned to the agent. The agent improves its path selection policy with the goal of accumulating the least expected cost in the long run. Under certain assumptions, RL is able to select paths that achieve a suitable tradeoff which balances the maximum network lifetime approach and minimum energy consumption approach.

1.2 Research Objectives

The objectives of this research are as follows:

1.2.1 To study the energy utilization in packet delivery in MANETs.

1.2.2 To apply reinforcement learning (RL) to solve the energy-efficient routing protocol problem in mobile ad hoc networks with dynamic topology.

1.2.3 To compare the reinforcement learning solution with other energy-efficient routing protocols in terms of energy consumption, network lifetime, ratio of the number of successfully delivered packets and the long-term cost.

1.2.4 To compare the tradeoff when RL is used in an energy-efficient routing protocol.

1.3 Assumptions

1.3.1 Energy consumption for transfering data packets depends on the packet size and transmission range. Since a free space radio model is assumed, factors such as noise, fading etc. are ignored.

1.3.2 The state transitions of the environment can be modeled as a Markov process. Consequently, the path selecting problem in MANETs can be modeled as a Markov decision process (MDP).

1.3.3 Reinforcement learning can achieve a near-optimal path selection policy, which can balance the tradeoff between the maximum network lifetime approach and the minimum energy consumption approach.

1.4 Scope of the Thesis

This thesis consists of two main parts. Firstly, we propose the on-policy Monte Carlo (ONMC) reinforcement learning method to deal with the tradeoff between the maximum lifetime and minimum energy consumption approaches. The actions available to the agent in the ONMC method is selected from a set of paths which are optimal according to three different metrics, i.e., the maximum-minimum battery level, the least energy consumption and the least path cost. Therefore, action space contains three types of optimal paths and we aim to find a good path selection policy that balances their tradeoff. We then compare the energy-efficient routing performance in terms of network lifetime, energy consumption, ratio of successfully delivered packets and the long-term cost which is a function that depicts the optimal tradeoff balance in long run.

In the second part, we compare the aforementioned performance metrics of the proposed ONMC method with existing energy-aware routing protocols. These include the Minimum Total Transmission Power Routing (MTPR) which selects a path with minimum energy consumption along path; the Min-Max Battery Cost Routing (MMBR) which prolongs the network lifetime by avoiding the route with nodes having the least battery capacity among all nodes in all possible routes, so that the battery of each node will be used more fairly. The ONMC method is also compared to existing algorithms which integrate the two approaches. These include the conditional max-min battery capacity routing (CMMBCR) which switches from minimum energy consumption routing to maximum network lifetime routing by using a threshold parameter consumption; and an algorithm based on a cost function of both node energy consumption and residual battery level (Chang and Tassiulas ,2004) referred to as the LowCost method.

1.5 Expected Usefulness

1.5.1 To obtain an energy-efficient routing algorithm that balances the tradeoff of both maximum network lifetime and minimum energy consumption approaches by using RL which can discover a near-optimal path in mobile ad hoc networks under the dynamic topology scenario.

1.5.2 To obtain a conclusion about the application of reinforcement learning in energy-efficient routing in mobile ad hoc networks and suggest its possible applications to other routing protocol problems, for example, the mobility prediction, secured routing, etc.

1.6 Organization of the Thesis

The remainder of this thesis is organized as follows. Chapter 2 presents the theoretical background of reinforcement learning (RL) which underlies the contribution of this thesis. Firstly, we give an overview of the Markov decision process (MDP) concept, and introduce reinforcement learning (RL) which provides an approximate solution to the MDP formulated problem. In particular, we employ a RL method called the on-policy Monte Carlo (ONMC) method which learns through experience by interacting with the environment on an episode-by-episode basis. We also provide justification for employing the ONMC method to energy-efficient routing in mobile ad hoc networks.

In Chapter 3, we study the energy-efficient routing protocols in mobile ad hoc networks. Firstly, we present the energy model which is used to calculate the energy consumption in mobile ad hoc networks. We then propose a reinforcement learning technique called the on-policy Monte Carlo (ONMC) method to balance the tradeoff between maximum network lifetime and minimum energy consumption routing. Routing performance is compared in terms of the ratio of successfully delivered packets, and the long-term cost which is a function that depicts the optimal tradeoff balance in long run. The performance of ONMC method is compared with a variety of existing energy-efficient routing protocols in MANETs.

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

CHAPTER II

BACKGROUND THEORY

2.1 Introduction

In this thesis, we study the energy-efficient routing protocol problem in mobile ad hoc networks (MANETs) where the network topology is dynamic. This feature inherent in MANETs is due to node mobility where links are formed whenever nodes are located within the transmission range and are broken otherwise. Furthermore, links may also disappear when certain nodes have exhausted their battery power while participating in the packet forwarding process. Routing protocols determine which nodes the packets are forwarded to. Hence, routing decisions strongly affect the amount of energy consumed in the routing process, the node lifetime and consequently the network lifetime. Good routing protocols therefore should take into account the node mobility behavior, energy consumption and residual node lifetime. Unfortunately, the dynamics between these factors are difficult to capture in MANETs with an explicit mathematical model. In this thesis, we therefore apply reinforcement learning (RL) which can potentially cater the dynamics in mobile ad hoc networks. In RL, no explicit mathematical formulation of a model is needed. Instead, good decisions are discovered through a systematic trial and error interaction with its environment to achieve a particular goal.

RL is a computational approach used to solve a Markov decision process (MDP) problem by identifying how a system in a dynamic environment can learn to choose optimal actions to achieve a particular goal. A RL problem is a problem faced by an agent that must learn good behaviors through trial and error interactions with a dynamic environment. The route discovery in MANETs can be viewed as an episodic task. In particular, an episode starts each time a source node searches for a destination node. If at least one route that reaches the destination node is found, the source node will select a path based on the information of the residual battery level and energy consumption along these paths. Due to the episodic nature of the MANETs, this thesis employs a method for solving reinforcement learning problems with episodic tasks, known as the on-policy Monte Carlo (ONMC) method (Sutton, 1998). The ONMC method learns incrementally on an episode-by-episode basis, meaning that the action-value functions are estimated and policies are improved after each episode. Under certain assumptions, the ONMC method eventually converges to an optimal policy and optimal value function-given only sample episodes and no other knowledge of the environment dynamics.

This chapter presents the ONMC method which is applied to achieve energy-efficient routing in MANETs in this thesis. The next section provides a theoretical background on Markov decision theory. An introduction of reinforcement learning is given in section 2.3. Section 2.4 presents the on-policy Monte Carlo method and the conclusion is presented in the final section

2.2 Markov Decision Theory Background

2.2.1 Markov Property

The Markov property says that anything that has happened so far can be summarized by the current state. Thus, the probability that the next state at time k+1 based on what we have seen can be defined as simply the conditionally probability based on the current state at time k is

$$\Pr\left\{s_{k+1} = s' \middle| s_k = s\right\} = \Pr\left\{s_{k+1} = s' \middle| s_k = s, s_{k-1} = s, ..., s_0 = s\right\}. \tag{2.1}$$

We now formally define the Markov property for the reinforcement learning problem. A state refers to information on the environment that may be useful in making a decision. If the state has the Markov property, then the environment's response at time $k+1$ depends only on the state representation at time *k*. In other words, such state has the Markov property, and is a referred to as a Markov state.

2.2.2 Markov Decision Processes

A reinforcement learning task that satisfies the Markov property is called a Markov decision process, or MDP. Suppose the current time is time step *k*. Based on the current state of the environment (*s*), the agent selects an action (*a*). As a result of taking action a at state s , the environment transits into new state$\left(s'\right)$. The probability of each possible next state is

$$P_{ss'}^{a} = \Pr\left\{s_{k+1} = s' \middle| s_k = s, a_k = a\right\}. \tag{2.2}$$

These quantities are called transition probabilities. Similarly, given any current state and action, s_k and a_k, together with any next state, s_{k+1} and an associated reward g_k is generated and returned back to the agent. The expected value of the next reward is

$$G_{ss'}^{a} = E\left\{ g_k \left| s_k = s, a_k = a, s_{k+1} = s' \right. \right\}. \tag{2.3}$$

Upon receiving this reward signal, the agent assesses how good the action was and seeks to improve its decision in order to maximize the reward gained in the long run.

2.3 Reinforcement Learning

Reinforcement learning (RL) is a computational approach which identifies how a system in a dynamic environment can learn to choose optimal actions to achieve a particular goal. The learner is not told 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.

A form of supervised learning scheme such as neural network require sample input-output pairs from the function to be learned. In other words, supervised learning requires a set of questions with the right answers. For example, we might not know the best way to program a computer to recognize an infrared picture of a tank, but we do have a large collection of infrared picture, and we do know whether each picture contains a tank or not. Supervised learning could look at all the examples with answers and learn how to recognize tank in general. However, there are many situations where we do not know the correct answer that supervised learning requires. For example, in a mobile ad hoc network, the question would be the set of the network topology at a given time, and the answer would be how the routing protocol should find the paths in each network topology. Simple neural networks cannot learn to select intermediate node unless there is a set of known answer. Hence, if we do not predict the network topology in mobile ad hoc network in the first place, simple supervised learning cannot determine the correct routing decision.

Reinforcement learning, on the other hand, differs from the more widely studied problem of supervised learning in several ways. The most important difference is that there is no presentation of input/output pairs. Instead, after choosing an action the agent is told the immediate reward and the subsequent state, but is not told which action would have been in its best long-term interests. It is necessary for the agent to gather useful experience about the possible system states, actions, transitions and rewards actively to act optimally. Another difference from supervised learning is that, for RL, the on-line performance criterion is important.

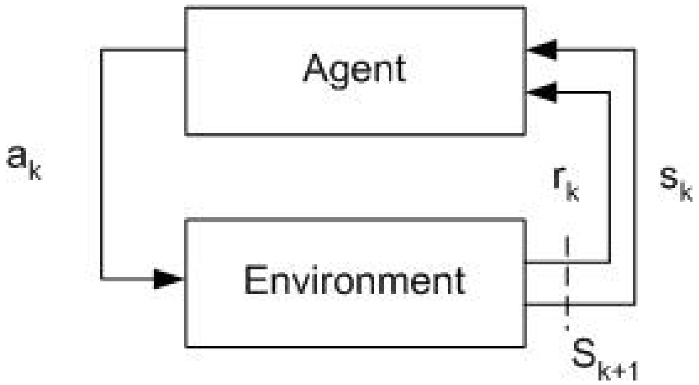


Figure 2.1 Diagram of agent-environment interaction in reinforcement learning

The evaluation of the system is often concurrent with learning. Figure 1 shows the basic idea how RL can learn to solve a complex task through repeated interactions with its environment. Components of RL include an autonomous agent, the environment, associated actions and rewards. The agent is the learner or the decision maker. Everything comprised outside the agent is called the environment. In general, an action refers to a decision that an agent takes, while a state refers to information on the environment that may be useful for the agent to make a decision. An intuitive way to understand the relation between the agent and its environment is given in the following example dialogue.

Environment : You are in state 65. You have 4 possible actions.

Agent : I'll take action 2

Environment : You received a reinforcement of 7 units. You are now in state 15. You have 2 possible actions.

Agent : I'll take action 1

Environment : You received a reinforcement of -4 units. You are now in state 65. You have 4 possible actions.

Agent : I'll take action 2

Environment : You received a reinforcement of 5 units. You are now in state 44. You have 5 possible actions.

The agent's job is to find a policy π that maps a state to actions in such a way that maximizes some long-run measure of reinforcement. In a standard reinforcement learning model, an agent interacts with its environment. This interaction takes the form of the agent sensing the environment, and based on this sensory input choosing an action to perform in the environment. The action changes the environment in some manner and this change is communicated to the agent through a scalar reinforcement signal. There are three fundamental parts of a reinforcement learning problem: the environment, the reinforcement function and the value function.

1) The Environment

Every RL system learns a mapping from states to actions by trial-and-error interactions with a dynamic environment. This environment must at least be partially observable by the reinforcement learning system, and the observations may come in the form of sensor reading, symbolic descriptions, or possibly *mental* situations. If reinforcement learning system can observe perfectly all the information in the environment that might influence the choice of action to perform, then the reinforcement learning system chooses an action based on the true *state* of the environment. This ideal case is the best possible basis for reinforcement learning and, in fact, is a necessary condition for much of the associated theory.

2) The Reinforcement Function

As stated previously, RL systems learn a mapping from states to actions by trial-and-error interactions with a dynamic environment. The goal of the reinforcement learning system is defined using the concept of a reinforcement function, which is the exact function of future reinforcements the agent seeks to maximize. In other words, there exists a mapping from state-action pairs to future reinforcements. That is, after performing an action in a given state, the RL agent will receive some reinforcement (reward) in the form of a scalar value. The agent learns to perform actions that will maximize the sum of the reinforcements it receives when starting from some initial state and proceeding to a terminal state.

3) The Value Function

The value function is mapping from state to state values. Given a policy π, which determines which action should be performed in each state, the value of state

$V^{\pi}(s)$ is defined as the expected sum of the reinforcement received when starting in the state *s* and following some fixed policy to a terminal state

$$V^{\pi}(s) = E_{\pi}\left\{\sum_{n=1}^{\infty} g_{t+n} \left| s_t = s \right.\right\}. \tag{2.4}$$

The optimal policy V^{*} would therefore be the mapping from state to action that maximizes the sum of the reinforcements when starting in an arbitrary state and performing actions until a terminal state is reached, that is,

$$V^{*} = \max_{\pi}\left\{V^{\pi}(s)\right\}. \tag{2.5}$$

In a general setting, we wish to select optimal actions at each time step to maximize the long-term system performance criterion.

2.3.1 Monte Carlo Methods

Among the diverse availability of RL tools, a particular technique called the Monte Carlo method has been selected in this thesis. The reason is because the episodic nature of route search process in mobile ad hoc networks. The path selection decisions are learned directly from experience on an episode-by-episode basis. By estimating the action-value at end of each episode and performing a policy improvement, and repeating the process under the newly improved policy, the policy obtained finally converges to an optimal policy, which aims at balancing the maximum network lifetime and minimum energy consumption approaches.

Monte Carlo methods are ways of solving the reinforcement learning problem based on averaging sample returns. Monte Carlo methods require only experience, i.e., sample sequences of states, actions, and rewards from on-line or simulated interaction with an environment. Learning from on-line experience is striking because it requires no prior knowledge of the environment's dynamics, yet can still attain optimal behavior. To ensure that well-defined returns are available, we define Monte Carlo methods only for episodic tasks. That is, we assume that the experience is divided into episodes, and that all episodes eventually terminate no matter what actions are selected. It is only upon the completion of an episode that value estimates and policies are changed. Monte Carlo methods are thus incremental in an episode-by-episode sense, not in a step-by-step sense (Sutton, 1998).

Let us consider Monte Carlo methods for learning the state-value functions for a given policy, $\pi : S \to A$ π. Recall that the value of a state is the expected return, or in other words, the expected cumulative future discounted reward starting from that state (Sutton, 1998). An obvious way to estimate the state value function from experience, is simply to average the returns (eq.2.4) observed after visits to that state. As more returns are observed, the average should converge to the expected value. The policy evaluation problem for action values is to estimate $Q^{\pi}(s,a)$, the expected return when starting in state s, taking action a, and thereafter following policy π:

$$Q^{\pi}(s,a) = E_{\pi}\left\{ \sum_{n=1}^{\infty} g_{t+n} \middle| s_t = s, a_t = a \right\}. \qquad (2.6)$$

The first-visit Monte Carlo method averages the returns following the first time in each episode that the state was visited and the action was selected. That is,

$$Q^{\pi}\left(s,a\right)=\frac{c\left(s,a,1\right)}{1}, \tag{2.7}$$

where *c(s, a, 1)* is the return of after the *first* occurrence of state action pair *(s, a).* The every-visit Monte Carlo method estimates the value of a state-action pair as the average of the returns that have followed visits to the state in which the action was selected. That is,

$$Q^{\pi}\left(s,a\right)=\frac{\sum_{k=1}^{n}c\left(s,a,k\right)}{n\left(s,a\right)}, \tag{2.8}$$

where *c(s, a, k)* is the return of the state-action pair after the occurrence of each visit to *(s, a)* , *n(s, a)* is the number of visits to *(s, a).*

Both return averaging methods converge quadratically, to the true expected values as the number of visits to each state-action pair approaches infinity. This process is called policy evaluation under a fixed policy π .

After each episode, the observed average returns are used for policy evaluation and the policy is improved at all states visited in the episode. Policy improvement is the process of constructing a new policy that improves over an original policy by making it greedy or ε-greedy. The greedy policy selects the best action with respect to the current action-value estimates. The ε-greedy policy behaves greedily most of the time with respect to the current action-value estimates. But every once in while, with some small probability, the ε-greedy policy selects an action at random and independent of the action-value estimates. This because it is not enough just to select the actions currently estimated to be the best as certain state-action pairs may never be visited. Hence, for these unvisited state-action pairs, there is no return to average, and it may never be learned that these state-action pairs may actually be better than the visited state-action pairs. By using the ε-greedy policy, other (unvisited) state-action pairs have a chance of being visited which may well be better than the visited state-action pairs. Therefore, we need to estimate the value of all the actions available at each state, not just the one we currently favor. The ε-greedy policy helps to explore other actions available in each state.

Consider a reinforcement learning system with finite state of the environment and reinforcement learning agent which has a finite number of actions. Suppose that the initial policy followed by the agent is π_0. By alternating complete steps of policy evaluation and policy improvement, an optimal policy π^* and optimal action-value function Q^* can be achieved :

$$\pi_0 \xrightarrow{E} Q^{\pi_0} \xrightarrow{I} \pi_1 \xrightarrow{E} Q^{\pi_1} \xrightarrow{I} \pi_2 \xrightarrow{E} \ldots \xrightarrow{I} \pi^* \xrightarrow{E} Q^*,$$

where $\xrightarrow{E}$ denotes a complete policy evaluation and $\xrightarrow{I}$ denotes a complete policy improvement.

During policy evaluation many episodes are experienced, with the approximate action-value function approaching the expected value function asymptotically. Let us assume that we do indeed observe an infinite number of episode and that, in addition, the episodes are generated with exploring starts. The latter assumption assigns a non-zero probability to every state-action pair of being the starting pair of an episode. Under these assumptions, the Monte Carlo methods will compute each Q^{π_k} exactly, for an arbitrary policy π_k. In other words, a complete policy evaluation is performed. After each episode, the observed returns are used for policy evaluation, and then the policy is improved at all the states visited in the episode. Policy improvement is done by making the policy greedy with respect to the current value function. This is for any action-value function $Q^{\pi}(s,a)$ under current policy π, the corresponding greedy policy is the one that, for each state s in the state space $(s \in S)$, deterministically chooses an action with maximal action-value function (sometimes referred to as the Q-value)

$$\pi(s) = \arg\max_{a} \{Q(s,a)\}. \tag{2.9}$$

Policy improvement then can be performed by constructing each new policy π_{k+1} as the greedy policy with respect to Q^{π_k}. The policy applies to π_k and π_{k+1} because, for all $s \in S$,

$$\begin{aligned} Q^{\pi_k}\left(s, \pi_{k+1}(s)\right) &= Q^{\pi_k}\left(s, \arg\max_{a}\left\{Q^{\pi_k}(s,a)\right\}\right) \\ &= \max_{a}\left\{Q^{\pi_k}(s,a)\right\} \\ &\geq Q^{\pi_k}\left(s, \pi_k(s)\right). \end{aligned} \tag{2.10}$$

The above relation assures us that each π_{k+1} is uniformly better than π_k, unless it is equal to π_k, in which case they are both optimal policies. This in turn assures us that the overall process converges to an optimal policy and the optimal value function.

2.3.1.1 On-policy and Off-policy Monte Carlo

There are two approaches in the Monte Carlo methods which are the on-policy method and off-policy method. In the on-policy method, the agent commits to always exploring and tries to find the best policy that it still explores. In the off-policy method, the agent also explores, but learns a deterministic optimal policy that may be unrelated to the policy followed. However, only the on-policy method has a sound mathematical proof that it to converges to optimal policy. Convergence proof of the off-policy method remains an open issue (Sutton 1998).

On-policy methods attempt to evaluate or improve the policy that is currently being used to make decisions. The only general way to ensure that all actions are selected infinitely often is for the agent to continue to select them. Let *S* denote the set of all possible states and *A* denote the set of all possible actions. Let the actions selected in episode *t* be governed by policy π_t, where $\pi_t : S \to A$. Denote the state-action value function of (s,a) by $Q^{\pi_t}(s,a)$ which is the expected reward when starting from state-action pair (s,a) and a fixed policy π_t is followed thereafter. Let the initial policy be π_0 and initialize $Q^{\pi_t}(s,a)$ at the beginning of an episode. For each episode *t*, generate the action at a given state according to π_t. At the end of episode *t*, $Q^{\pi_t}(s,a)$ is updated according to

$$Q^{\pi_t}(s,a) = Q^{\pi_{t-1}}(s,a) + \frac{1}{t}\left[\sum_{n=\tau_t(s,a)}^{N_{t-1}} g(s_n,a_n) - Q^{\pi_{t-1}}(s_n,a_n)\right], \tag{2.11}$$

where N_t is the duration or the number of time steps in episode *t*, $\tau_t(s,a)$ where $0 \le \tau_t(s,a) \le N_t$ is the time step when the first visit of state-action pair (s,a) occurs in episode *t*, and $g(s,a)$ is the reward obtained from taking action *a* at state *s*. Note that the summation term is the accumulated reward following only the first occurrence of (s,a).

Furthermore, a new policy for the next episode, π_{t+1}, is improved from the previous policy, π_t, using an ε-greedy policy which is implemented as follows,

$$\pi_{t+1}(s) = \begin{cases} a^* & \text{with probability } 1\text{-}\varepsilon + \dfrac{\varepsilon}{|\mathrm{A}|} \\ a \in A - \{a^*\} & \text{with probability } \dfrac{\varepsilon}{|\mathrm{A}|}, \end{cases} \tag{2.12}$$

where a^* is the greedy policy found by $a^* = \arg\max_{a \in A}\{Q^{\pi_t}(s,a)\}$, $\varepsilon \in [0,1]$ and $|A|$ is the size of the action space. Under specific conditions, for any ε-greedy policy with respect to Q^{π} is guaranteed to be better than or equal to π.

2.4 On-policy Monte Carlo in the Thesis

In this thesis, we propose an energy-efficient routing method for mobile ad hoc networks (MANET) which employs a reinforcement learning method based on sample episodes, called the on-policy Monte Carlo (ONMC) method. This method requires sample episodes for estimating a specific function which quantify what state or action is good in the long run. Such function, so called action-value function is a function of a state-action pair which quantifies the average amount of reward an agent can expect to accumulate in the long run from averaging sample returns received from that state-action pair. In our energy-efficient routing problem (see section 3.3.1 for more details), we define the state to take into account of the amount of energy consumption and residual battery level which are quantized into discrete intervals. Such discretization provides simplification to our problem by partitioning the continuous state space into a discrete state space with a finite number of intervals. The action space in our MANET framework is the subset of all possible paths discovered which connects the source node to the destination node. The process can be viewed as two episodic tasks, one nested in the other. The inner episode starts when the source node (agent) selects an action (path), then it receives a cost signal corresponding to the action selected. The outer episode starts when the distance vector protocols are exchanged periodically. The actions carried out within the inner episode follows a certain fixed policy. Such policy will be evaluated and improved at the end of each outer episode. For each fixed governing policyπ, the action-value functions $Q^{\pi}(s,a)$ are computed from average sample returns received from the environment. The ONMC method learns incrementally on an episode-by-episode basis, meaning that the action-value functions are estimated and policies are improved after each (outer) episode. Under the assumptions, that all state-action pairs are visited an infinite number of times in the limit of an infinite number of episodes, the ONMC method eventually converges to an optimal policy (Sutton, 1998).

2.5 Conclusion

In this chapter, an overview of the Markov decision process (MDP) concept is given. We also introduced the concept of reinforcement learning (RL) to provide an approximate solution to the MDP formulated problem. The MDP framework has been used to formulate routing problems in mobile ad hoc networks (Maneenil and Usaha, 2005), (Chang, Ho, and Kaelbling, 2004). For routing protocol problems, we view them as an episodic task where an episode starts each time a source node searches for a destination node. The episode terminates when at least one route that reaches the destination node is found, or when the maximum hop count to the destination node is reached. For this reason, a RL method based on sample episodes, called the on-policy Monte Carlo (ONMC) method was introduced in this chapter. In the next chapter, an ONMC formulation of the energy-efficient routing in MANETs is presented. Furthermore, we compare the performance of the ONMC method with well-known existing energy-efficient routing protocols such as (Toh, 2001) and (Chang and Tassiulas, 2004).

CHAPTER III

ENERGY-EFFICIENT ROUTING IN MOBILE AD HOC NETWORKS: A RL APPROACH

3.1 Introduction

This chapter presents an energy-efficient path selection algorithm which aims to balance the contrasting objectives of the maximum network lifetime and the minimum energy consumption routing protocols. The proposed algorithm is based on a reinforcement learning (RL) technique called the on-policy Monte Carlo (ONMC) method. This method is suitable for learning good decisions in tasks which are episodic. The routing problem in MANETs may be viewed as an episodic task, where each episode starts when a source node initiates a route search, in order to discover paths that can reach the destination node. An episode ends when at least one path which reaches the destination node is found, or when the number of maximum hop count is reached (and no paths are found). If at least one route is found, the source node will calculate the energy consumption and the residual battery levels along these paths. The source node then selects one of such paths to forward the data packets. At the end of each episode, a cost associated to the selected path is assigned to the decision-maker (i.e., the source node). The expected cost per episode incurred from the source node, evaluates how good the path selection decision was when the source node was in that particular state. The path selection decision of the source node can be improved by systematically selecting the path that minimizes the expected state-action cost per episode. Under the assumptions that every state-action pairs are selected and simulation observe an infinite number of episode (Sutton,1998), the ONMC method eventually converges to a good path selection rule, which aims at balancing the minimum energy consumption and maximum network lifetime routing protocols.

In recent literature, there are many works have been proposed to strike a balance between the contrasting objectives of maximizing the network lifetime and minimizing the energy consumption. The existing protocols which can achieve this objective by varying threshold parameter value include Toh (2001), Aslam, Li and Rus (2003). However in many scenarios, determining the suitable values of parameter values is not straightforward as these values could depend on the network size and the mobility profile of each node etc. Other existing protocols employ the combined cost function of both energy consumption and residual battery level such as Chang and Tassiulas (2004), Kwak, Kim and Yoo (2004). The normalized link capacity cost of the form $c_{ij} = e_{ij}\left(B_{init}/B_i\right)$, where B_i is the residual energy at node *i*, and e_{ij} is the communication energy cost for link (*i, j*), performed better than other combined cost metrics (Basagni, Conti, Giordano and Stojmenovic, 2004). Such method appears to perform well in terms of maximizing the network lifetime in the long run, whereas energy consumption is not emphasized. It is not clear whether there exist an optimal energy tradeoff balance in the long run.

To address the problem of jointly optimizing the energy consumption and residual battery route selection in MANETs with dynamic topology, we present a route selection scheme based on a reinforcement learning technique call on-policy

Monte Carlo (ONMC) method. It should be noted that in recent literature, RL has already been successfully applied in MANETs. For instance, (Usaha, 2004) and (Usaha and Barria, 2004) applied it to control the amount of routing overhead with marginal difference in the path search ability. (Maneenil and Usaha, 2005) integrated RL with an existing reputation scheme to determine a good rule to distinguish malicious nodes in MANETs. (Chang, Ho, and Kaelbling, 2004), applied RL to find good adaptive routing and movement policies in a mobilized ad hoc networking domain and demonstrated some promising empirical results under a variety of different scenarios.

The emphasis of this chapter is focused on the following issues:

1. The introduction of the energy model
2. The MDP formulation for the energy-efficient routing protocol in MANETs, which is the first main contribution of this thesis.
3. Application and performance quantification of the on-policy Monte Carlo (ONMC) method, which is the second main contribution of this thesis.
4. The comparison of routing performance between the best variant of ONMC method and four existing algorithms (i.e. MMBR, MTPR, CMMBCR and Lowcost)

The structure of this chapter is organized as follows. The energy model which describes the energy consumption in each node that is used for transmitting data messages between two nodes is described in section 3.2. Section 3.3 is dedicated to describing the on-policy Monte Carlo (ONMC) formulation to achieve balanced energy tradeoff routing in MANETs. Section 3.4 presents the experimental results and discussion. Finally, section 3.5 concludes the entire chapter.

3.2 Energy Model

An ad hoc network consists of multiple nodes that maintain network connectivity through wireless communications. The connectivity is enabled via radio transmissions generated by a set of cooperating nodes. To model the energy consumption in each node, we use the *radio model* discussed in Muruganathan, S.D.(2005).

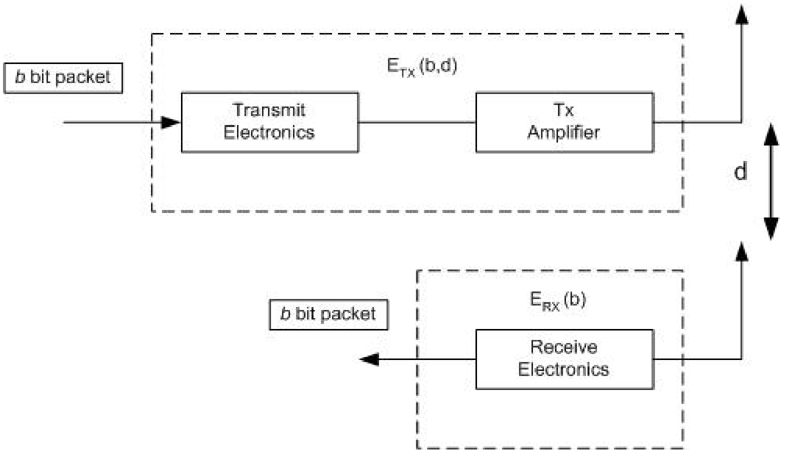


Figure 3.1 Radio model

The transmitting and receiving energy required for transfer of a data message of *b*-bits between two nodes by a transmission range of *d* meters is given by

$$E_{TOT} = E_{TX} + E_{RX}\,, \tag{3.1}$$

where E_{TX} is the energy dissipated in the transmitter of the sending node given by

$$E_{TX}\left(b,d\right) = \left(E_{elec} \times b\right) + \left(\varepsilon_{fs} \times b \times d^{2}\right), \tag{3.2}$$

and $\varepsilon_{fs} = 10\text{pJ}/\text{bit}/m^2$ is the energy consumed in free space at the output transmitter antenna for a transmitting range of one meter.

Consider a *n*-node mobile ad hoc network which makes extensive use of broadcasting (i.e., a message is sent from one node to all other nodes within its transmission range). Suppose that some node *i* has *x*, *y* and *z* as neighboring nodes which are separated by a distance of d_{ix} , d_{iy} and d_{iz} , respectively. When node *i* broadcasts to its neighbors, the energy consumption is calculated from the furthest neighboring node. Hence, all neighboring nodes are reached with one transmission energy usage which is given by

$$E_{TX}\left(i\right) = \max\left\{E_{TX}\left(b, d_{ix}\right), E_{TX}\left(b, d_{iy}\right), E_{TX}\left(b, d_{iz}\right)\right\}. \tag{3.3}$$

The term E_{RX} is the energy consumption at the receiving node given by

$$E_{RX}\left(b\right) = E_{elec} \times b, \tag{3.4}$$

where E_{elec} is the energy expended in the radio electronics which is equal to 50 nJ/bits. The term $E_{elec} \times b$ is assumed negligible. Since the delivered datagram packets have fixed length, then all algorithms use the same amount of energy to receive the packets. The energy consumption is therefore

$$E_{TOT} = E_{TX} + E_{RX} = \left(\varepsilon_{fs} \times b \times d^2\right). \tag{3.5}$$

We define $Battlevel(i)$ as the residual energy in node *i*, which is reduced by a quantity of $E_{TX}(i)$, when node *i* transmits a packet of *b* bits to its neighboring nodes along some path.

3.3 ONMC as An Energy-Efficient Routing Protocol

3.3.1 MDP Formulation

In order to balance the tradeoff between the minimum energy consumption and the maximum network lifetime approaches, the information of the residual battery level and energy consumption at each neighboring node are required. The information of the neighboring nodes, which is referred to as the state, is crucial to the path selection decision. In particular, each source node acts as an agent which decides to select a path depending on the current state. Assuming that each node moves independently from one another and its future movement (position, direction, and velocity) depends only on its current movement, the future topology of the network can depend only on the current topology of the network and is independent of its past. Then it may also be implied that the future state (that is, the energy consumption and the residual battery level of each node) depends only on the current state and not its past, and it is possible to (roughly) model the state transition as a Markov process.

Therefore, we can (roughly) model the path selecting problem in MANETs as a Markov decision process (MDP), whose goal is to optimize some performance criterion in finite horizon. The finite horizon problem is considered here due to the episodic nature of message exchanges between the nodes due to the distance vector protocol. An episode starts immediately after a message exchange and terminates at the subsequent message exchange. Applying the MDP framework with appropriate costs to the route selection process in MANETs permits us to select paths at a given state such that a suitable tradeoff between various energy-efficient paths is achieved. The MDP framework consists of the following components.

1) *State:* Suppose that some source node and destination node are connected by a set of multiple paths, *L*. The state space should encompass both the energy consumption and battery levels of the network since we are interested in finding paths that balance the tradeoff of the two factors. For some path *l*, the energy consumption along path $l \in L$, can be determined by

$$P_l = \sum\nolimits_{\forall i \in l} E_{TX}\left(i\right). \tag{3.6}$$

Denote the minimum energy consumption path by

$$l_e = argmin_{\forall l \in L}\left\{P_l\right\}. \tag{3.7}$$

To account for the battery load distribution among multiple paths connecting the source and destination nodes, we define the *bottleneck* for each path *l* by

$$B_l = \min_{\forall i \in l}\left\{Battlevel\left(i\right)\right\} \tag{3.8}$$

Denote the path with the max-min residual battery level by

$$l_b = argmax_{\forall l \in L}\left\{B_l\right\}. \tag{3.9}$$

The energy consumption and battery bottleneck in each path have continuous values which given rise to a continuous state space. Since the policy improvement (eq.2.10) and evaluation (eq.2.11) are performed for each state-action pair, it is therefore desirable to discretize their values to obtain a finite number of state-action pairs. Furthermore, due to limited onboard processing capability at each node, the discrete state space MDP is thus preferable. Hence, P_{l_e} and B_{l_b} are quantized into discrete intervals.

The quantization of minimum energy consumption is

$$\left\{P_{l_e}(1),.....,P_{l_e}(n)\right\}, \text{ where } P_{l_e}(i), 1 \le i \le n. \tag{3.10}$$

The quantization of bottleneck battery level is

$$\left\{B_{l_b}(1),.....,B_{l_b}(m)\right\}, \text{ where } \mathrm{B}_{l_b}(j), 1 \le j \le m. \tag{3.11}$$

The state space of a source node (i.e., the agent or decision-maker) is given by

$$S = \left\{s : s = \left[P_{l_e}(i), B_{l_b}(j)\right], 1 \le i \le n, 1 \le j \le m\right\}, \tag{3.12}$$

where the size of S is $|S| = n \times m$.

To calculate $P_{l_e}(i)$, we assumed that each node knows the location of its neighbor node's transmission range by means of Global Positioning System (GPS) (Kwak, Kim and Yoo, 2004). Each node in the mobile ad hoc network selects its neighbor nodes from the maximum transmission range and link age[1]. In the route discovery process, the source node broadcasts the RREQ to all its neighbor nodes. The intermediate nodes forward the RREQ packets to their neighbor nodes after having received them from the source node. The process is repeated until the RREQ packets arrives at the destination node or the maximum of hop count is reached. The source node waits for route reply until time out. Once the destination node receives the RREQ packet, it calculates the energy consumption along the path and appends it to a RREP packet and sends it retracing the same path back to the source node. Along the retraced path, the intermediate nodes append data about their residual battery level into the RREP packet. Once the RREP packet arrives, the source node then knows the amount of energy consumed and the battery level of bottleneck nodes in each path connecting the source and destination nodes. The source node can then calculate the quantized level of energy consumption and bottleneck battery level according to (eq.3.12).

2) *Actions:* Given the profile of the quantized energy and battery state of all available paths, the source node must then select a path. We define the set of paths to select from, or action space, based on three commonly-used energy-aware routing mechanisms namely, the minimum energy routing (l_e) in (eq.3.7), the max-min routing (l_b) in (eq.3.9) and the minimum cost routing (l_c) given by Chang and Tassiulas (2004)

[1] The link age determines the stability of a link. The higher the link age, the more likely that link would remain connected.

$$l_c = argmin_{\forall l \in L} \left\{ \left(\sum_{\forall i \in l} E_{TX}(i) \right)^{x_1} \left(B_l\right)^{-x_2} \left(B_{init}\right)^{x_3} \right\}, \tag{3.13}$$

where B_{init} is the initial level of battery which is assumed constant for all nodes, and (x_1, x_2, x_3) are weight factors. Note that the shortest paths can be obtained with the weights (0, 0, 0), whereas (1, 0, 0) and (0, 1, 1) correspond to the minimum energy path (l_e) and the max-min residual battery path (l_b), respectively. Note that the minimum cost route (l_c), which uses a normalized link capacity cost is chosen here due to its outstanding network lifetime performance (Chang and Tassiulas, 2004). Let A be the action space such that

$$A = \left\{ a : a = \left[a(1), a(2), a(3) \right], a(j) \in \{0,1\}, \sum\nolimits_{\forall j} a(j) = 1 \right\}, \tag{3.14}$$

where $a(1) = 1$ refers to the selection of the max-min residual battery path l_b , $a(2) = 1$ selects the minimum energy consumption path l_e , $a(3) = 1$ selects the minimum energy cost path l_c, and $a(j) = 0$ refers to not selecting the corresponding path. We defined the action space in this manner so that it consists of only these paths because we aim to find a *good* path selection policy that balances their tradeoffs.

3) *Cost structure:* Once the source node selects an action, say path $a = l$, at a given state *s*, a cost $c(s,a)$ incurs where

$$c(s,a) = \left(P_l\right)^{x_1} \left(B_l\right)^{-x_2} \left(B_{init}\right)^{x_3}. \tag{3.15}$$

The goal is to find a path selection policy that optimizes the long-term average performance criterion in finite horizon. We apply the method in the following subsection in order to achieve this goal.

3.3.2 ONMC Reinforcement Learning for MANETs

In this thesis, we propose an energy-efficient routing method for mobile adhoc networks (MANETs). Due to the episodic nature of the MANET, we employ a reinforcement learning method based on sample episodes, called the on-policy Monte Carlo (ONMC) method. This method requires sample episodes to estimate the action-value functions $\left(Q(s,a), \forall s \in S, \forall a \in A\right)$ which quantify the average amount of cost an agent can expect to accumulate in the long run from that state-action pair. These action-value functions are computed from average sample returns received from the environment operating within a fixed decision rule called policy $(\pi : S \rightarrow A)$. The ONMC method learns incrementally on an episode-by-episode basis, meaning that the action-value functions are estimated and policies are improved after each episode. The pseudocode of the ONMC method is depicted below:

```
1. Initialize: Return $\{s,a\} \Leftarrow$ empty list $\forall s \in S, \forall a \in A$
2.                    $Q\{s,a\} \Leftarrow$ arbitrary $\forall s \in S, \forall a \in A$
3.                         $\pi \Leftarrow$ arbitrary $\varepsilon - soft$ policy
4. For interval T=1 to forever     //Outer episode loop counting
                                   // distance vector exchange intervals.
5.          For connection request t=1 to end of interval T  //Inner episode
                                   // loop counting number of path search
                                   // connection requests until the end of interval T.
6.               Generate path selection action interval T according to $\pi$.
7.               Get cost from taking action.
8.          For each state-action pair $(s,a)$ appearing in interval T
9.               $R \Leftarrow$ add all costs following the first occurrence of $(s,a)$
10.              Append $R$ to Return $(s,a)$
11.              $Q(s,a) \Leftarrow average(\text{Return}(s,a))$
12.         For each $s$ appearing in interval T
13.              $a^* \Leftarrow \arg\min\{Q(s,a)\}$
14.              $\pi(s,a) \Leftarrow \begin{cases} 1-\varepsilon+\varepsilon/|A| & \text{, if } a = a^* \\ \varepsilon/|A| & \text{, if } a \neq a^*. \end{cases}$
```

The ONMC method can then be mapped into the framework for energy-efficient routing in MANETs as follows. First of all, initialize the return to zero for each state-action pair, at each mobile node (line 1). Arbitrarily initialize the state-action values (line 2) and the starting policy (line 3). The outer episode starts when the distance vectors are exchanged (line 4). The inner episode starts when a source node requests a search for paths that can reach the destination node. A generic multiple path discovery scheme is then executed. The information of the minimum energy consumption and the maximum residual battery levels gathered from the path search defines the state s of the source node. The source node then takes an action by selecting one of such paths (line 6). Once the source node selects a path (takes action

a) at state s, a corresponding cost incurs (line 7). The process for each connection request is repeated until the end of interval T. Then the returns and action-value functions are reevaluated (line 9-11) and ε-greedy policy improvement is performed (line 12-14).

3.4 Simulation Results and Discussion

3.4.1 Parameter Setting

We consider a MANET of 36 nodes randomly distributed in a square area of 1000 m by 1000 m. Each node has an initial battery level of 100 J. A node whose battery is depleted disconnects from the network and cannot recharge from any external power supply. The movement of the nodes follows a random waypoint mobility model. In particular, each node stays in a current location for a period of time called pause time. After this period is over, each node moves to a randomly selected new location with a constant velocity. The node velocity is uniformly chosen between 0 and 15 m/s. In this thesis, we consider two mobility scenarios, i.e. the high mobility scenario with pause time of 0s and the low mobility scenario with pause time 120s. Each node has a transmission range of 200 meters. A link is formed between any pair of nodes within this range. To discover multiple paths between a pair of nodes, flooding is used in each algorithm. Note that the energy usage during the path discovery process is not considered since all algorithms employ the flooding scheme. Since we are focusing on the energy usage for packet routing, the energy used by the flooding scheme is not considered as all algorithms consume the same amount of energy during the path discovery process. Fixed-length datagram packets of 50 Kbytes are transmitted. The path energy consumption, P_{l_e}, is quantized into 5 intervals: [0,0.3), [0.3,0.5), [0.5,0.7), [0.7,0.9), [0.9,∞) J. The residual battery of a path, B_{l_b}, is quantized into 5 intervals: [0,20), [20,40), [40,60), [60,80), [80,100] J. The packet sending rate is 0.2 packets per second. The changing topology is updated every 30-second interval using a distance vector update protocol.

3.4.2 Metrics Used to Compare Routing Performance

Each algorithm is simulated for 20 runs until a precision of 3% is achieved for every performance metric. To assess routing performance, the following metrics are considered:

1) the network lifetime which is the duration until the first node in the network disconnects

2) the average network energy consumption per node,

$$E_{avg} = \frac{1}{N}\sum_{i=1}^{N} E_{TX}\left(i\right), \tag{3.16}$$

where $E_{TX}(i)$ is the total energy consumption of node i, and N is the number of nodes in the network

3) the ratio of successfully delivered packets,

$$R_{DP} = \frac{number\ of\ \ successfully\ delivered\ packets}{Total\ packets\ sent} \tag{3.17}$$

4) the average energy consumed per delivered packet define as the network energy consumption divided by the number of successfully delivered packets accumulated over simulation

5) the long-term cost,

$$C(X) = \frac{R_{EC}(X)}{R_{AN}(X) \times R_{DP}(X) \times R_{LT}(X)}, \tag{3.18}$$

where $R_{EC}(X)$ is the ratio between the network energy consumption of algorithm *X* and the maximum network energy consumption from simulation, $R_{AN}(X)$ is the ratio between number of alive nodes of algorithm *X* and the total number of nodes in the network, $R_{DP}(X)$ is the ratio between the number of successfully delivered packets from algorithm *X* and the total number of packets sent in the simulation, $R_{LT}(X)$ is the ratio between the network lifetime of algorithm *X* and the average duration of simulation.

3.4.3 Impact of Action Space in ONMC

In this subsection, we compare the performance of various sets of action spaces used for the ONMC method. Recall that an action refers to the selection of a path based on three commonly-used energy aware routing as described in (eq. 3.7), (eq.3.9) and (eq.3.13). The minimum energy path in (eq.3.7) minimizes the total energy consumption on a path but suffers short network lifetime. The max-min battery path in (eq.3.9) prolongs the network lifetime but does not guarantee minimum energy consumption. The minimum cost routing in (eq.3.13) reflects both the energy consumption rate and the residual battery levels. By combining different paths to create different action spaces, we compare the performance between the following variants of the ONMC RL method:

- The BERL method selects the min-max routing (l_b) and the minimum energy routing (l_e) as action space, i.e., $A = \{l_b, l_e\}$.

- The BECRL method selects the min-max routing (l_b), the minimum energy routing (l_e) and the minimum cost routing (l_c) as action space, i.e., $A = \{l_b, l_e, l_c\}$.
- The ECRL method selects the minimum energy routing (l_e) and the minimum cost routing (l_c) as action space, i.e., $A = \{l_e, l_c\}$.

The simulations results for the three ONMC variants are shown in Tables 3.1-3.2. To find the best performance in terms of the long-term cost *C(X)* in each method, we investigate the cost function (eq.3.15) by varying the parameters (x_1, x_2, x_3). We observed that (1, *x*, *x*) obtained the best results, which agrees with (Chang and Tassiulas, 2004). In particular, we selected x_1 = 1 and x_2 = x_3 when weighting factors x_2, x_3 equal 1, 5 and 30. Note that (1, 1, 1) gives equal weight to energy usage and residual battery, whereas (1, 5, 5) gives more weight to the residual battery level and shows improved network lifetime (Chang and Tassiulas, 2004). Finally, (1, 30, 30) gives the most weight to the residual battery level and obtained a network lifetime close to min-max battery path (l_b). Tables 3.1-3.2 show the routing performance in the high mobility scenario and low mobility scenario, respectively. From the table, it can be observed that, in terms of network lifetime, BECRL has longer network lifetime than BERL and ECRL. The reason is because BECRL has two actions out of three which favors the maximum network lifetime, i.e., the min-max path (l_b) and the minimum cost path (l_c). Note that in beginning, l_c favors minimum energy consumption routing. Later on, when battery level depletes giving greater weight on the residual battery, l_c will then tend to favor the path with the maximum residual battery level which has the lowest cost and prolongs the network lifetime.

Table 3.1 Weight parameters comparison (x_1, x_2, x_3) at pause time 0s

Parameter (x_1, x_2, x_3)	(1,1,1)			(1,5,5)			(1,30,30)		
Performance	BERL	BECRL	ECRL	BERL	BECRL	ECRL	BERL	BECRL	ECRL
Lifetime (s)	42349	44343	43309	44764	45676	44988	44818	45318	44196
R_{DP}	0.387	0.39	0.398	0.382	0.384	0.391	0.379	0.38	0.386
E_{avg} *(J/node)*	91.27	91.18	90.9	91.43	91.36	91.12	91.55	91.52	91.14
Avg. alive nodes	9.1	9.55	10	9.15	9.15	9.75	8.7	8.75	9.55
C(X)	0.603	0.543	0.52	0.576	0.561	0.523	0.611	0.598	0.551

Table 3.2 Weight parameters comparison (x_1, x_2, x_3) at pause time 120s

Parameter (x_1, x_2, x_3)	(1,1,1)			(1,5,5)			(1,30,30)		
Performance	BERL	BECRL	ECRL	BERL	BECRL	ECRL	BERL	BECRL	ECRL
Lifetime (s)	52770	53606	53065	54738	55810	54827	55870	56180	54601
R_{DP}	0.415	0.417	0.423	0.411	0.413	0.418	0.4098	0.4095	0.414
E_{avg} *(J/node)*	92.03	91.96	91.28	92.4	92.26	91.77	92.68	92.64	92.19
Avg. alive nodes	12.8	13.1	14.3	12.4	12.4	13.25	12.05	11.8	12.6
C(X)	0.326	0.312	0.282	0.329	0.321	0.3	0.334	0.339	0.322

We can observe from the tables that ECRL consumes the least energy. The reason is because ECRL has an action space comprising the minimum energy path (l_e) and the minimum cost path (l_c). Both of these actions favor paths with the minimum energy consumption even though, later on when the battery starts to deplete, l_c tends to prefer paths with more residual battery levels. In addition, ECRL also exhibits the highest ratio of successfully delivered packets, R_{DP}. The reason is because ECRL uses the least energy consumption which in turn increases the number of alive nodes. These remaining nodes give rise to enhanced network connectivity and consequently better chances in successfully discovering paths connecting the source and destination nodes when compared to BECRL and BERL. In addition, the lower average number of alive nodes in BECRL and BERL is caused by the fact that these methods attempt to balance the load among different nodes to extend the network lifetime. As a result, nodes are used uniformly and the battery levels are depleted more or less at the same rate. Therefore, after first node disconnects, other nodes also become exhausted shortly afterwards. Hence, a sharp drop in the number of alive nodes for these methods is observed.

In terms of the long-term cost *C(X)* which is a function of network lifetime, energy consumption, delivered packets and alive nodes, the ECRL method exhibits the least cost of all. The reason is because ECRL can reduce energy consumption, increase network lifetime, higher number of alive nodes and higher ratio of successfully delivered packets. Note that when the weighting factor x_1 = 1 and x_2, x_3 are increased, a longer network lifetime is observed, however with increased long-term cost C*(X)*. Nevertheless, for the three variants of ONMC, the ECRL shows the best performance in terms of the long-term cost. Therefore, in the remaining experiments, ECRL will be used to compare with other existing energy tradeoff balancing schemes.

3.4.4 Comparison of Performance with Existing Schemes

In this subsection the performance of a variant of the proposed ONMC method, called ECRL, will be compared with existing routing algorithms. Since these algorithms use mechanisms to achieve energy tradeoff balance. The first algorithm uses threshold parameters to switch from the minimum energy consumption routing to maximum network lifetime routing. In particular, the conditional max-min battery capacity routing (CMMBCR) scheme in (Toh, 2001) chooses a minimum energy consumption path if all nodes in all possible routes have sufficient battery capacity. When the residual battery for certain nodes fall below a predefined threshold (γ), routing through these nodes will be avoided. As a result, the time until the first node disconnects is extended. The second algorithm is the minimum cost routing scheme from Chang and Tassiulas (2004). This algorithm employs a combined cost function of the energy consumption and residual battery level as described by (eq.3.13) and (eq.3.15).

In this section, the routing performance is compared between five algorithms:

- MMBR (Min-Max Battery cost Routing): selects the path that has the maximized minimum residual battery power of a node in the path, so that the battery of each node is used fairly and maximum network lifetime is achieved (Toh, 2001).
- MTPR (Minimum Total Transmission Power Routing): selects the path that minimizes the total transmission energy consumed per packet, disregarding the lifetime of each node (Toh, 2001).
- CMMBCR (Conditional Min-Max Battery Cost Routing) with threshold 60 (TH-60): selects a path according to MTPR from a set of some possible routes of which the residual battery of each node is above 60J, and switches to MMBR, otherwise (Toh, 2001).
- CMMBCR with threshold 80 (TH-80), similar to CMMBCR with TH-60 but with a threshold value of 80 J (Toh, 2001).

- Lowcost: selects the minimum cost path as described by (eq.3.13) and (eq.3.15). This scheme prefers minimum energy consumption routes when the nodes have plenty of residual battery. As the node's battery depletes, the algorithm prefers paths that maximizes the network lifetime (Chang and Tassiulas, 2004).
- ECRL: selects a path based on the ONMC reinforcement learning decision which is our proposed method.

A. Weight parameters comparison

In this experiment, we determine the weight parameters (x_1, x_2, x_3) which give the best performance for the cost function (eq.3.15). Figure 3.2 compares the long-term cost *C(X)* for different weights between the Lowcost algorithm and the proposed ECRL method. Note that other algorithms did not require use of weight parameters and therefore, are not shown here. Simulation was also run for alternative forms of weights, such as the (x, 0, 0) and (0, x, x) which correspond to minimum energy path and maximum network lifetime. We observed that (1, x, x) parameters obtained the best results. The results observed in high and low mobility scenario, show the best performance is attained when parameters (1, 1, 1) are used for all algorithms. Therefore, in the rest of the experiments, parameters (1, 1, 1) will be used.

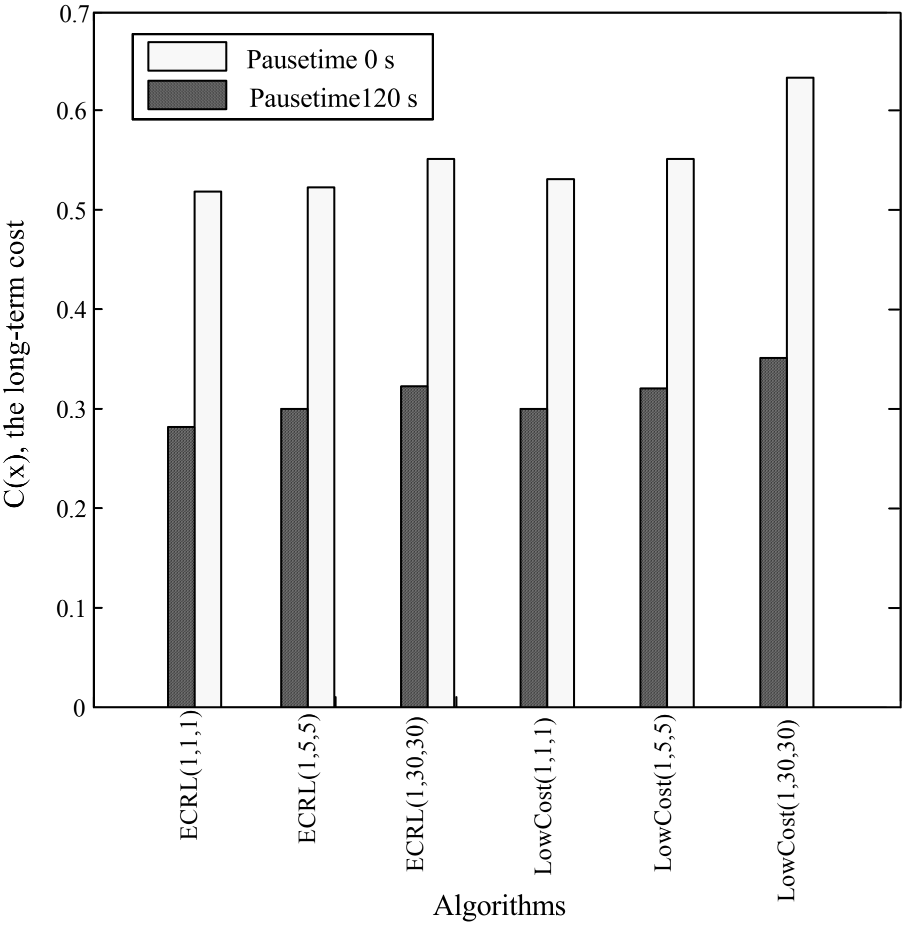


Figure 3.2 Performance comparison of weight parameters (x_1, x_2, x_3)

B. Network lifetime

In Figure 3.3, it can be observed that the LowCost algorithm has the longest network lifetime. Note that the ECRL method is able to attain network lifetimes near that of the LowCost algorithm. The MTPR algorithm has the shortest network lifetime. The reason is because the latter method does not maximize network lifetime. In particular, nodes which frequently find themselves on minimum energy paths experience heavy load forwarding and their battery quickly become exhausted. Other methods have higher network lifetime than MTPR since all take into account the residual battery levels in the path selection. Hence, these algorithms do not suffer early node disconnection as the MTPR. Note that ECRL was able to increase network lifetime by up to 15.1 percent when compared with MTPR.

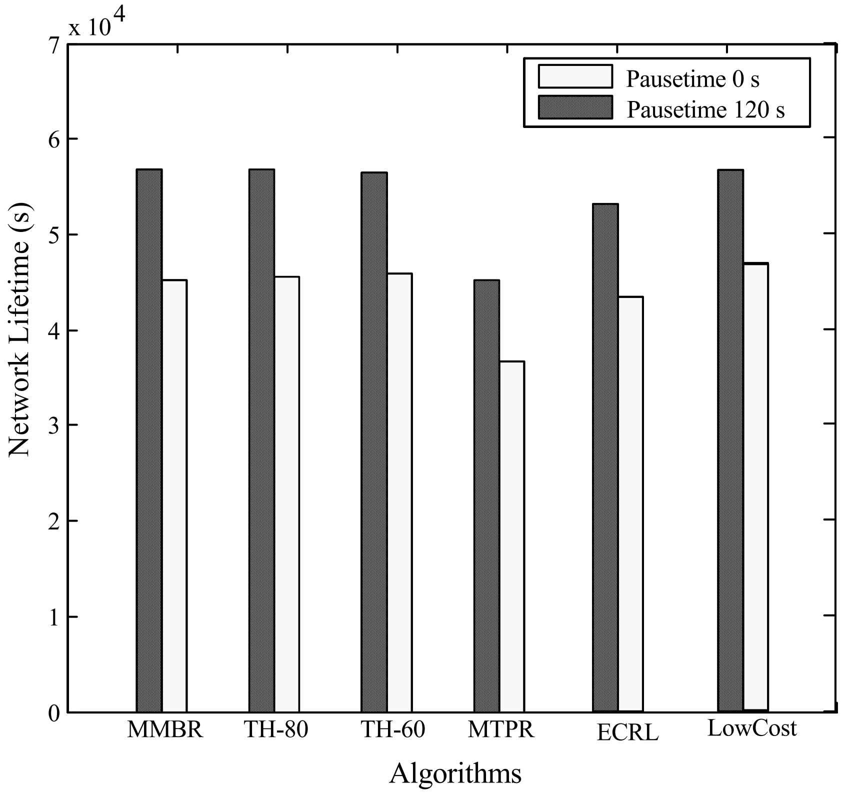


Figure 3.3 Comparison of the network lifetime

C. Network energy consumption per node

Figure 3.4 shows that the MTPR method consumes the least energy in comparison with all other methods. This is, however, at the expense of decreased network lifetime as show in Figure 3.3. Note that apart from MTPR, ECRL consumes less energy than all of the remaining algorithms. This is due to the fact that ECRL action space contains the minimum energy path (l_e). The results show the ECRL algorithm was able to decrease energy consumption by up to 1.8 percent when compared with MMBR while the network lifetimes of ECRL are significantly longer than that of MTPR.

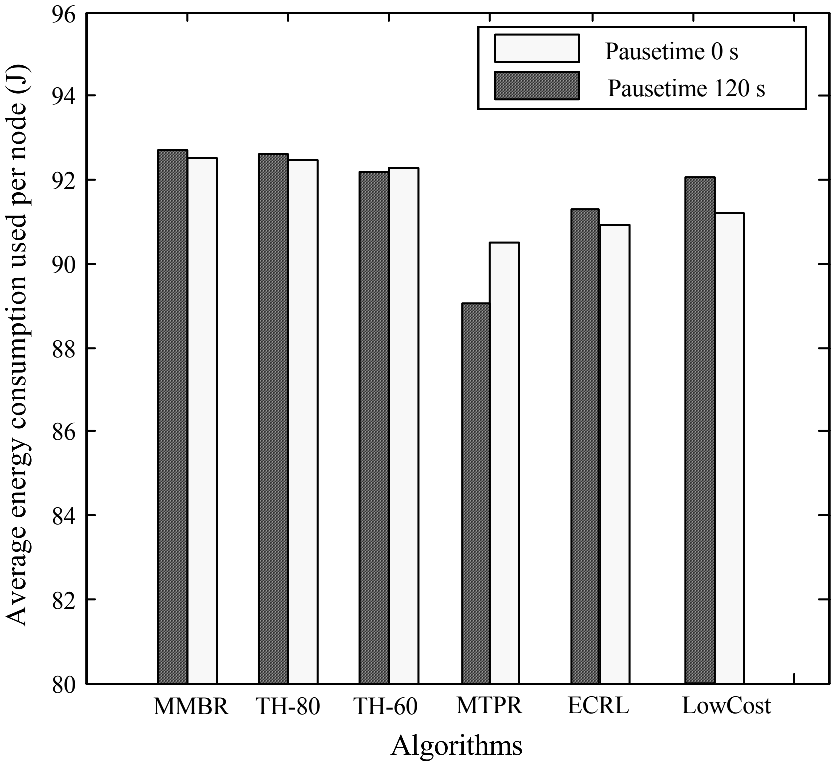


Figure 3.4 Comparison of the network energy consumption used per node

D. Number of alive nodes

Table 3.3, shows the average number of nodes still alive in the MANET. The greater the number of alive nodes, the higher the connectivity opportunity in the network. It can be observed that ECRL shows the highest number of nodes still alive in the high mobility scenario. The MMBR, CMMBCR TH-60 and CMMBCR TH-80 have the least number of nodes still alive in the network than all the remaining algorithms. The reason is because these algorithms balance the load therefore, the batteries at most nodes are exhausted at the same rate. Hence, after first node disconnects other nodes will disconnect soon afterwards. ECRL can attain up to 45 percent higher average number of alive nodes when compared with TH-80.

Table 3.3 Comparison of the number of alive nodes

Pause time	Comparison of alive node in each algorithms					
	MMBR	TH-80	TH-60	MTPR	ECRL	Lowcost
0 s	5.65	5.5	5.7	8.9	10	9.2
120 s	8.75	9.1	10.4	14.8	14.3	12.75

E. Long-term cost versus the network lifetime and energy consumption

Figure 3.5 and 3.6 compare the long-term cost in (eq. 3.18) for all algorithms as a function of network lifetime and network energy consumption. Results show that the MTPR uses the least amount of network energy consumption but has the shortest lifetime since MTPR uses paths with minimum energy consumption. So nodes along such path quickly become exhausted. On the other hand, MMBR distributes the load among nodes according to the residual battery levels. So nodes last longer and the network lifetime is maximized. However, the network energy consumption is highest as MMBR does not take it into account. The preferable location would be near the upper left hand corner of the graph—depicting minimum energy consumption and maximum network lifetime. Note that the LowCost and the ECRL algorithms are closer to this area than MMBR, MTPR, TH-60 and TH-80. This suggests that the combined cost routing such as in (eq. 3.13), can lead to more energy-efficient routing over threshold schemes as MMBR, MTPR, CMMBCR TH-60 and TH-80. Note that the all algorithms exhibit higher long-term cost under the high mobility scenario as it becomes more energy-exhaustive and more difficult to find paths as mobility increases. However, note that the ECRL method achieved the lowest long-term cost over all other methods which depicts a balance among the network lifetime and network energy consumption while attaining high successful packet delivery ratio and number of alive nodes.

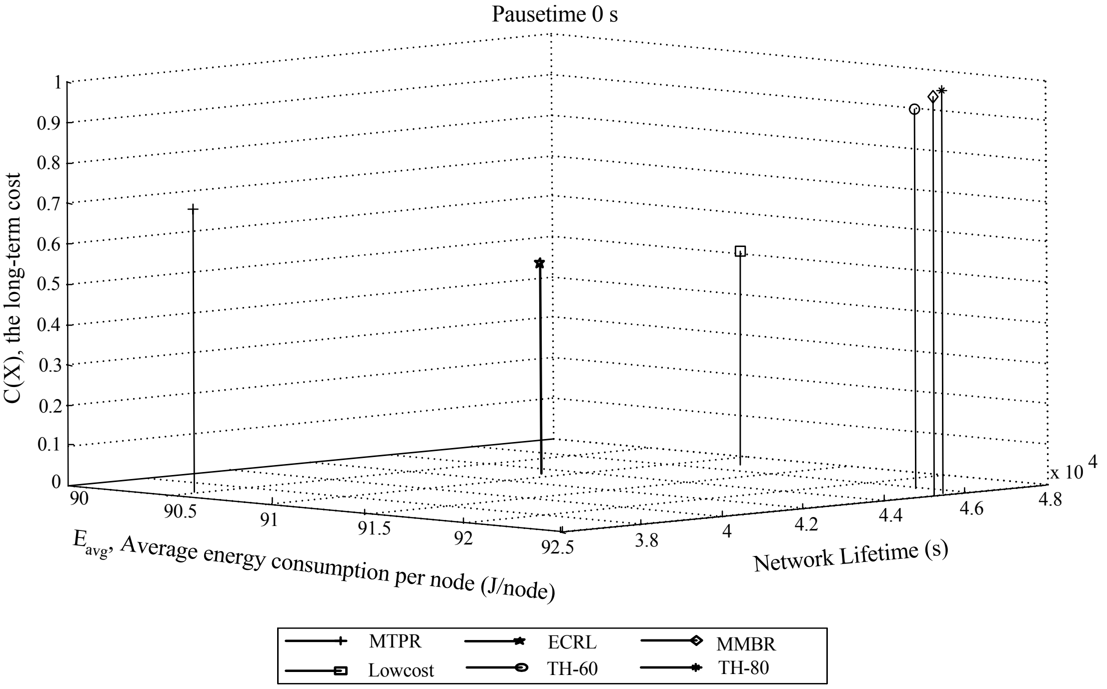


Figure 3.5 Comparison of routing performance in high mobility

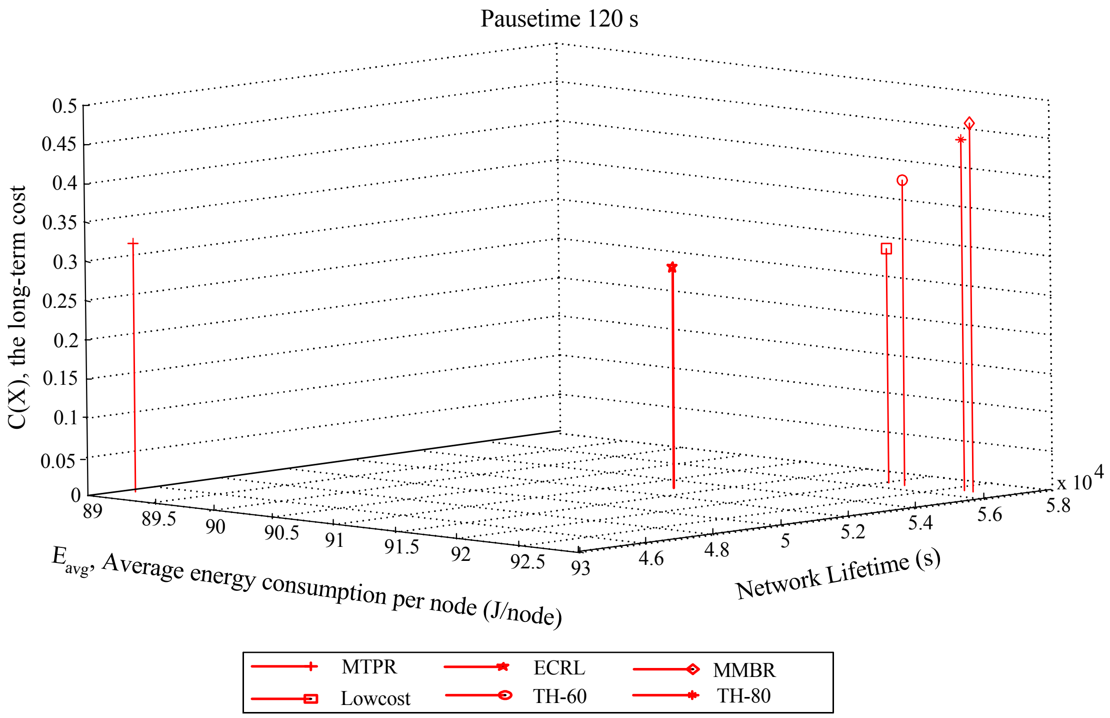


Figure 3.6 Comparison of routing performance in low mobility

F. Long-term cost

Figure 3.7 compares the cost in (eq. 3.18) for all algorithms. Results show that the ECRL algorithm achieved the lowest cost over all other algorithms. The reason is because the source node (agent) is able to learn to select the path which consumes the least energy at the beginning of simulation. In the long run, the ECRL learns to select a path by considering the energy consumption and battery levels of the intermediate nodes. Hence, it can be suggested that the ECRL algorithm can learn to select the paths which best balance the tradeoffs among the four metrics. Note that, the ECRL outperforms all other algorithms even in the high mobility scenario, by achieving up to 37 percent lower long-term cost over all other methods.

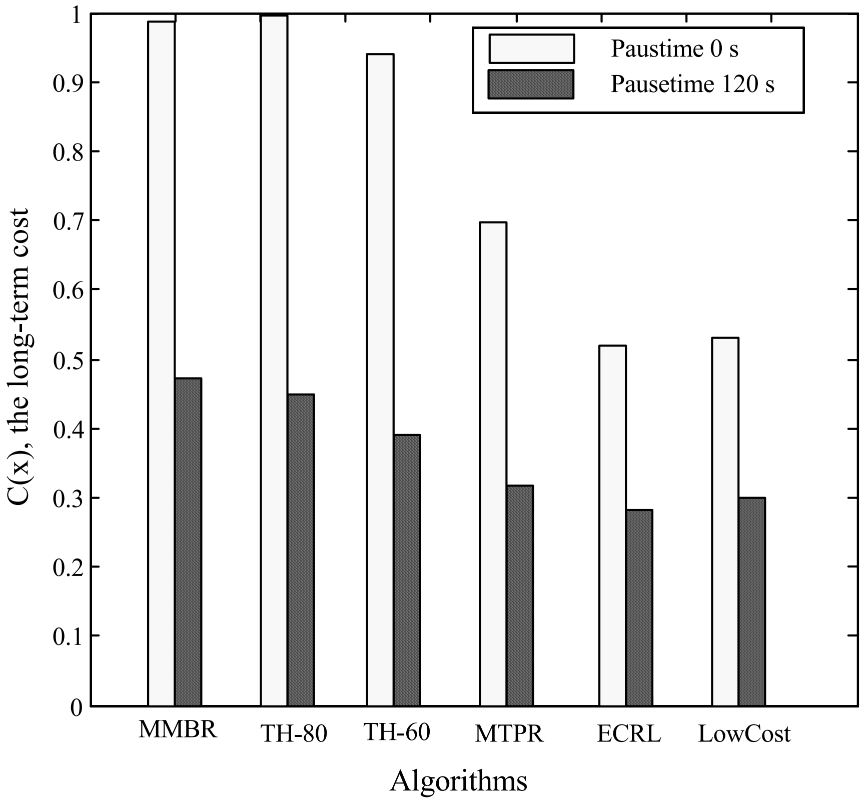


Figure 3.7 Comparison of the long-term cost

G. Ratio of successfully delivered packets

Figure 3.8, compares the ratio of successfully delivered packets. Results show that the ECRL algorithm, exhibits good routing performance in terms of high ratio of successfully delivered packets over all other methods. Results from Table 3.3 suggest that the higher number of alive nodes in the network, as observed in the ECRL algorithm, allow better connectivity and thus successful packet delivery in the network. Note that such ratio for the 0s pause time scenario is less than that of the 120s pause time. The reason is because it becomes more difficult to find paths as mobility increases. Nevertheless, the ECRL algorithm can still perform well even in high mobility environment. The result shows the ECRL algorithm can attain a ratio of successfully delivered packet of up to 5.5 percent higher than all algorithms.

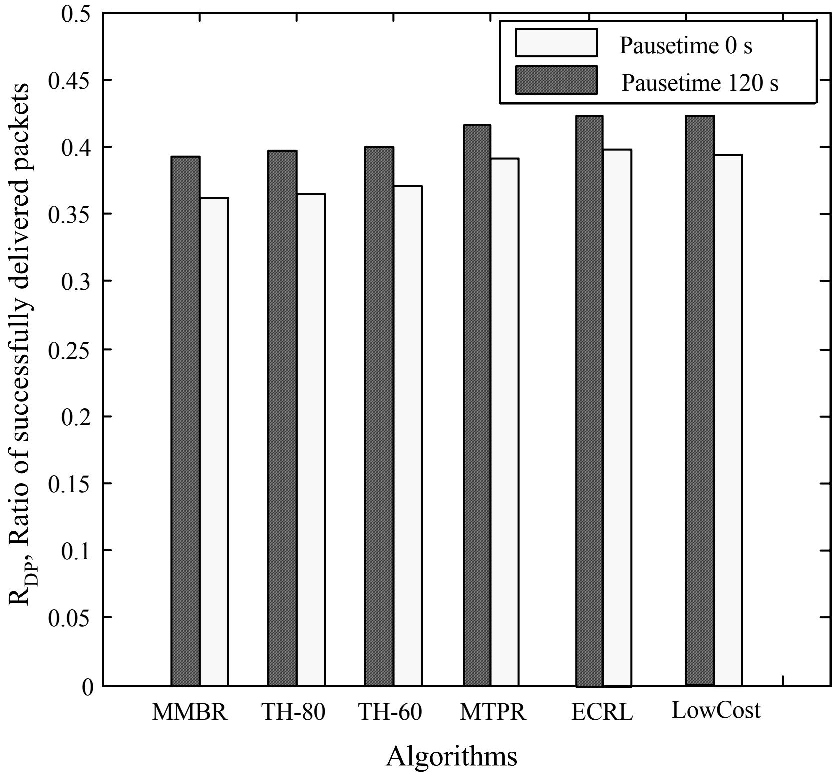


Figure 3.8 Comparison of the ratio of successfully delivered packets

H. Average energy consumed per successfully delivered packet

Figure 3.9 compares average energy consumed per successfully delivered packets. Note that the ECRL algorithm consumes energy as well as the MTPR algorithm in both mobility cases. The reason is because the ECRL algorithm can deliver more packets and consume less energy than other algorithms, owing to its action space.

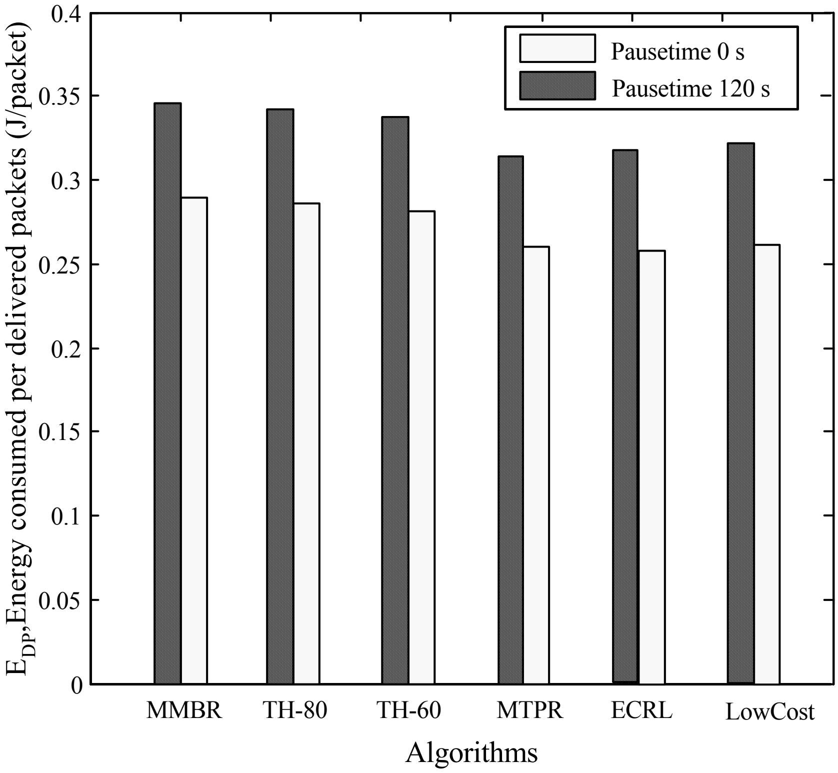


Figure 3.9 Comparison of the energy consumed per successfully delivered packet

I. Maximum node velocity scenarios

So far, the maximum node velocity has been fixed at 15 m/s. In this experiment, we compare the long-term cost *C(X)* for different maximum node velocity scenarios with pause time of 120s. In Figure 3.10, it can be observed that the ECRL algorithm achieved the lowest cost over all algorithms where the maximum node velocity is 15 and 20 m/s. As the nodes increase their maximum velocity the long-term cost *C(X)* is lower. The reason is because at higher node velocity it becomes more difficult to find the paths. Hence, the ratio of successfully delivered packet is decreased. Since fewer packets are delivered the amount of energy consumed by the nodes in the packet forwarding process is also decreased. As the energy consumption at each node is reduced, the battery levels at each nodes are less likely to become exhausted. This gives rise to greater number alive nodes and longer network lifetime. Note that all algorithms exhibit marginal difference in the long-term cost under high node velocity scenario. Hence, it can be suggested that as the maximum node velocity increases, the ability in balancing the energy tradeoffs of the ECRL is indifferent from other methods.

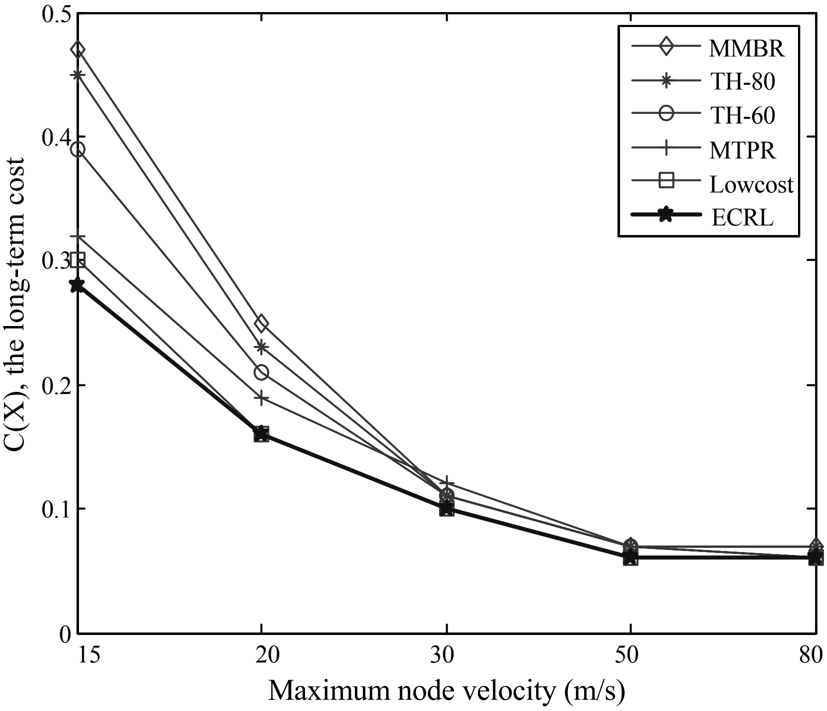


Figure 3.10 The long-term cost and the maximum node velocity

J. Maximum number of nodes in the MANET

So far, the maximum number of nodes in the network has been fixed to 36. In this experiment, we investigate the performance gain of the ECRL method as the network size increases. The maximum node velocity is 15 m/s, the pause time is 120s and the coverage area is 1000x1000 m^2. Figure 3.11 compares the long-term cost *C(X)* as the number of node in network is increased. Results show that all algorithms have lower long-term cost as the size of the network increases. The reason is because the greater number of nodes promotes better the connectivity opportunities in the network. Hence, a higher ratio of successfully delivered packets is observed. The greater number of nodes in the network give rise to higher node density. So nodes which are used to forward data packets are used more distributively. As a result, the network lifetime is increased. Note that apart from MTPR, all other algorithms tend to utilize the nodes fairly. Therefore, we observe that these algorithms have lower long-term cost than MTPR. Note that, however, as the size of the network increases, the ECRL method has marginal difference in the long-term cost. The results suggest that the ability to balance the energy tradeoff of the ECRL method is no different from other algorithms (except the MTPR method) as the network size is increased.

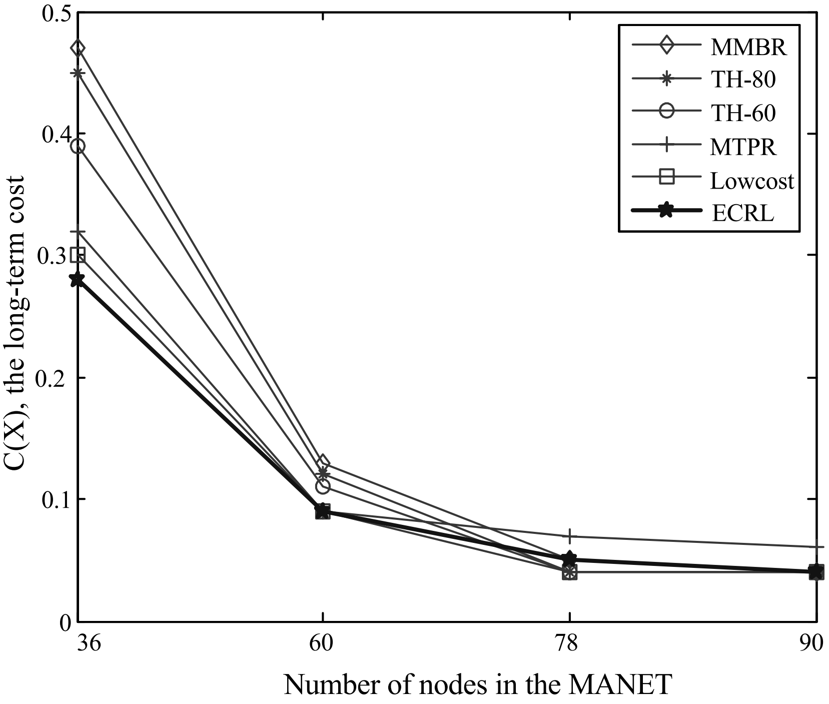


Figure 3.11 The long-term cost and number of nodes in the MANET

K. Implementation

The implementation of the ONMC method requires a reasonable increase in memory storage at each node for storing $Q(s,a)$ which has $|S||A|$ entries. In particular, the setting used in simulation requires memory usage of 600 Bytes $\left(5^2 \times 3 \times 8 Bytes\right)$ which includes the 5 state-discretization of the path energy consumption, the 5 state-discretization of path bottleneck battery level and an action space with 3 actions, assuming that each entry requires 8 Bytes. Furthermore, the packet size of the search message must be increased to store the energy consumption $\left(P_{l_e}\right)$ and residual battery along the path$\left(B_{l_b}\right)$ for the cost calculation for updating of the action-value functions.

The duration of the simulation is 75 s, based on simulation run by Microsoft Visual C++ 6.0 on Microsoft Window XP professional version 2002, run on a 1.8 GHz Intel Pentium 4 processor and 608 MB of RAM. This suggests that if the MANET undergoes significant changes, a new policy can be trained in a timely manner.

3.5 Conclusion

In this chapter, we presented the formulation of the energy-efficient path selecting problem in MANETs as a Markov decision process (MDP), whose goal to find sequence of path selection that minimizes the expected accumulated cost for the system. Furthermore, we presented a reinforcement learning method called the ONMC method to solve the MDP formulated problem for energy-efficient routing in MANETs. The proposed algorithm balances the contrasting objectives between maximizing the network lifetime and minimizing the energy consumption. The routing performances were compared with variations of the conditional Max-Min Battery Capacity Routing (CMMBCR) method which uses threshold values to control of path selection; and the minimum cost routing scheme, called Lowcost, where the cost metric is a function of the energy consumption along a path and the residual battery level.

Simulation results compared the performance metrics in terms of the network lifetime where the proposed method based on the ONMC method called, ECRL, was able to increase network lifetime by up to 15.1 percent when compared with MTPR. In terms of the network energy consumption per node, results show that the ECRL algorithm is able to decrease energy consumption by up to 1.8 percent when compared with MMBR. In addition, the ECRL algorithm can attain a ratio of successfully delivered packets of up to 5.5 percent higher than all algorithms. In terms of the average number of alive nodes remaining in network, ECRL can achieve up to 45 percent more alive nodes than all remaining algorithms. In terms of the long-term cost which takes into account the network lifetime, ratio of successfully delivered packets, network energy consumption and nodes alive in network, the ECRL gives the best tradeoff by achieving a long-term cost of up to 37 percent lower than all other algorithms. However, the performance gain of the ECRL method over other algorithms becomes marginal as the maximum node velocity and the number of nodes in the network are increased.

CHAPTER IV

CONCLUSIONS

4.1 Conclusion

In this thesis, we proposed a reinforcement learning (RL) framework, called the on-policy Monte Carlo (ONMC) method, to solve an energy-efficient routing problem in mobile ad hoc networks. The work carried out in this thesis aims to strike a balance between the contrasting objectives of maximizing the network lifetime and minimizing the energy consumption. The findings of this thesis can be summarized as follows.

4.1.1 Problem Formulation

The problem formulation of the energy-efficient path routing in mobile ad hoc network is a Markov decision process (MDP), whose goal is to find a sequence of path selection that minimizes the expected accumulated cost for the system in the long run. The cost structure is a function of the energy consumed, the residual energy as well as the number of alive nodes and the ratio of successfully delivered packets, so as to achieve a good path selection policy which balances the tradeoffs.

4.1.2 Energy-Efficient Routing in MANET: A RL Approach

In chapter 3, a reinforcement learning technique called the on-policy Monte Carlo (ONMC) method was presented to solve the MDP formulated routing problem. The ONMC method considers the state of the network before selecting a path. The state information includes the energy consumption along a path and the bottleneck battery level of a path. The algorithm then selects a path according to such information. The agent adaptively improves its path selection policy to achieve a balance between the maximum network lifetime approach and minimum energy consumption approach suitable for each scenario.

Simulation results showed that the ONMC with variants of action spaces consisting of the minimum energy path, the max-min residual battery level, and the minimum cost routes, could learn to balance the contrasting objectives by reducing energy consumption and prolonging the network lifetime. To measure the overall of routing performance, we defined a long-term cost as an integrated routing performance metric which is a function of the number of alive nodes, the ratio of successfully delivered packets, energy consumption, and network lifetime. The results showed that the proposed method attained the best tradeoff particularly in the high mobility scenario, by achieving a long-term cost of up to 37 percent higher than all other methods. However, the ability to balance the energy tradeoff of the proposed method is no different from other algorithms (except the MTPR method) as the network size is increased.

These results suggest that the ONMC method can attain good energy-aware routing decisions. However, the tradeoff of using the ONMC method is the requirement of reasonable increase in memory storage for $|S||A|$ entries at the source node where $|S|$ and $|A|$ are the cardinality of the state and action spaces, respectively. In particular, the setting used in our simulation required memory usage of only 600 Bytes $\left(5^2 \times 3 \times 8 Bytes\right)$ assuming that each entry requires 8 Bytes. There is a reasonable tradeoff, however, as the ONMC method requires training time on average of about 75 seconds in order to learn a good path selection policy. Hence, a new policy may be obtained in a timely manner should the MANET undergo any abrupt changes in the network.

4.2 Recommendation for Future Work

4.2.1 Mobility Prediction in Mobile Ad hoc Networks

In this thesis, we focus on the energy-efficient routing problem in mobile ad hoc networks. Since nodes in the network can move freely, this is a challenging task particularly when nodes are highly dynamic. We can extend our framework to predict the position of nodes from a mobility prediction algorithm. Using such prediction, we can prevent route errors due to node mobility and avoid short-lived or unstable paths in the path selection scheme.

4.2.2 Avoiding Malicious Nodes

In this thesis, the main focus is on energy-efficient routing. The fundamental assumption is that all nodes will cooperate and not misbehave. However, in mobile ad hoc networks, communication between nodes out of transmission range greatly relies on intermediate nodes. It is possible that certain intermediate nodes will eventually run out of battery and then misbehave by dropping packets as they try to save their battery level. To secure packet delivery, we can extend our framework to distinguish malicious nodes (Maneenil and Usaha, 2005) in order to achieve a secure and energy-efficient routing protocol in MANETs.

4.2.3 Improved Cost Function

In chapter 1, we referred to cost routing schemes as means to solve energy-efficient routing problems in mobile ad hoc networks (Basagni, Conti, Giordano and Stojmenovic, 2004). The structure of the cost function strongly affects the routing performance. For instance, cost functions which place weight on the residual battery level of a node tend to prolong the network lifetime. However, such cost structure may not decrease the total energy consumption or other performance metrics of interest may not be taken into consideration. Therefore, other forms of cost metrics with additional objectives of interest is another open issue worthwhile investigating.

4.2.4 State Quantization

In chapter 3, our MDP formulation quantizes the battery level and energy consumption into discrete uniform intervals. However, more investigation is needed regarding the suitable quantization levels. The quantization levels linearly affects the memory storage which has $|S||A|$ entries, where $|S|$ and $|A|$ are the size of the state space and action space, respectively. Note that the size $|S||A|$ directly influences the learning process because optimal polices are learned only when all actions and states are visited infinitely often (Sutton, 1998). This is the reason why sampling all possible states and all available actions, by means of exploring starts and action exploration, are crucial to policy improvement in the reinforcement learning process (Sutton, 1998). Therefore, the size of the action space and the state space, the latter of which is governed by how the states are discretized and the number of quantization levels, directly affect how often each state and action may be visited. Thus, the quantization of continuous states is a subject which warrants future investigation.

REFERENCES

Aslam, J., Li *, Q., Rus, D.,(2003). **Three Power-aware Routing Algorithms for Sensor Networks** *Wireless Communications and Mobile Computing*, Volume 3, Issue 2 , March 2003 Page(s) 187 - 208

Basagni, S., Conti, M., Giordano, S., Stojmenovic, I.,(2004). **Mobile Ad Hoc Networking.** IEEE Press and John Wiley & Sons, Inc., New Jersey and New York, April 2004.

Bergamo, P., Maniezzo, D., Travasoni, A., Giovanardi, A., Mazzini, G., Zorzi, M.,(2003).**Performance investigation of distributed power control for AODV routing protocol**, *Personal, Indoor and Mobile Radio Communications, 2003. PIMRC 2003. 14th IEEE Proceedings on Volume 1,* 7-10 Sept. 2003 Page(s):507 - 511

Chang, J.-H., Tassiulas, L.,(2004). **Maximum lifetime routing in wireless sensor networks Networking**, IEEE/ACM Transactions on Volume 12, Issue 4, Aug. 2004 Page(s):609 - 619

Chang, Y.-H., Ho, T., Kaelbling, L.P.,(2004). **Mobilized ad-hoc networks: a reinforcement learning approach**, *Autonomic Computing, 2004. Proceedings. International Conference* on 17-18 May 2004 Page(s):240 – 247

Geetha,V., Aithal, Sridhar, ChandraSekaran, K.,(2006). **Effect of Mobility over Performance of the Ad hoc Networks**, *Ad Hoc and Ubiquitous Computing, 2006. ISAUHC '06 International Symposium* on 20-23 Dec. 2006 Page(s): 138 – 141

Kwak, K.-S., Kim, K.-J., Yoo, S.-J.,(2004). **Power efficient reliable routing protocol for mobile ad-hoc networks**, *Circuits and Systems, 2004. MWSCAS '04. The 2004 47th Midwest Symposium on Volume 2,* 25-28 July 2004 Page(s):II-481 - II-484

Maneenil, K., Usaha, W.,(2005). **Preventing malicious nodes in ad hoc networks using reinforcement learning**, *Wireless Communication Systems, 2005.* 2nd International Symposium on 5-7 Sept. 2005 Page(s):289 – 292

Muruganathan, S.D., Daniel, C.F., Bhasin, R.I., Fapojuwo, A.O., **A centralized energy-efficient routing protocol for wireless sensor networks**, *Communications Magazine, IEEE* Volume 43, Issue 3, March 2005 Page(s):S8 - S13

Perkins, C.E., Royer, E.M., Das, S.R., Marina, M.K., (2001). **Performance comparison of two on-demand routing protocols for ad hoc networks**, *Personal Communications, IEEE* Volume 8, Issue 1, Feb. 2001 Page(s): 16 – 28

Romdhani, L., Bonnet, C.,(2004).**Energy consumption speed-based routing for mobile ad hoc networks**, Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on 2004 Page(s):729 – 734

Sheu, J.-P., Lai, C.-W., Chao, C.-M.,(2004). **Power-aware routing for energy conserving and balance in ad hoc networks**, *Networking, Sensing and Control, 2004 IEEE International Conference on Volume 1,* 21-23 March 2004 Page(s):468 - 473

Singh, S., Woo, M., and Raghavendra, C., **Power-aware routing in mobile ad hoc networks**, *International Conference on Mobile Computing and Networking (MobiCom'98)*, Dallas, TX, Oct. 1998

Sutton, R. S., Barto,A. G.,(1998). **Reinforcement learning: An Introduction**, Massachusetts, The MIT Press, 1998.

Tarique, M., Tepe, K.E., Naserian, M.,(2005). **Energy saving dynamic source routing for ad hoc wireless networks**, *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks, 2005. WIOPT 2005. Third International Symposium* on 3-7 April 2005 Page(s):305 – 310

Toh, C.-K.,(2001). **Maximum battery life routing to support ubiquitous mobile computing in wireless ad hoc networks**, *Communications Magazine, IEEE* Volume 39, Issue 6, June 2001 Page(s):138 – 147

Tsudaka, K., Kawahara, M., Matsumoto, A., Okada, H.,(2001).**Power control routing for multi hop wireless ad-hoc network**, *Global Telecommunications Conference, 2001. GLOBECOM '01. IEEE Volume 5,* 25-29 Nov. 2001 Page(s):2819 - 2824

Usaha ,W.,(2004). **A reinforcement learning approach for path discovery in MANETs with path caching strategy**, *The 1st of International Symposium on Wireless Communication Systems*, September 2004, Page(s):220-224

Usaha,W., Barria, J.A.,(2004). **A reinforcement learning Ticket-Based Probing path discovery scheme for MANETs**, *Ad Hoc Networks Journal*, Vol.2, 2004, Page(s): 319-334.

Venugopal,V., Bartos, R., Michael, J., and Sai, S.,(2003). **Improvement of Robustness for Ad Hoc Networks Through Energy-Aware Routing**, *Proceedings of the Fifteenth IASTED* International Conference on Parallel and Distributed Computing and Systems (PDCS), November 2003, Marina del Rey, CA.

Wang, K., Xu, Y.-L., Chen, G.-L., Wu, Y.-F.,(2004). **Power-aware on-demand routing protocol for MANET**, *Distributed Computing Systems Workshops, 2004. Proceedings. 24th International Conference on* 2004 Page(s):723 – 728

APPENDIX A

CMMBCR Algorithm

A.1 CMMBCR Introduction

Conditional Max-Min Battery Capacity Routing (CMMBCR) is a power-aware routing protocol which aims to satisfy the contrasting objectives between the maximum network lifetime and the minimum energy consumption approaches (Toh, 2001). This algorithm uses a parameter value to protect the nodes which have lower residual battery level than the predefined threshold value.

The basic idea behind CMMBCR is that when all nodes along the routes connecting a source node to a destination node have sufficient remaining battery capacity, i.e., above a predefined protection margin threshold (γ), the route with the minimum total energy consumption among these routes is chosen. However, if these routes consist of nodes with residual battery levels below this threshold, the route with the worst bottleneck nodes, i.e., nodes with the lowest battery capacity in the route, should be avoided to extend the lifetime of these nodes.

Let B_l define the bottleneck battery level for path *l* and *Battlevel(i)* is the residual battery in node *i* where

$$B_l = \min\left\{Battlevel\left(i\right)\right\}, \text{ for all node } i \in \text{ path } l$$

Let *A* be a set containing all possible routes between any two nodes at time *t* which satisfy the following equation:

$$B_l \geq \gamma, \quad \text{for any path } l \in A,$$

where γ is a threshold which ranges between 0 and 100. Note that for γ = 0, the CMMBCR is identical to MTPR. That is, the minimum total energy consumption path will be selected. Furthermore, γ= 100 is always identical to MMBR. That is, the path containing the node with the lowest battery capacity should be avoided so that each node will be used fairly.

Let *Q* denote the set containing all possible paths between the specified source and destination nodes. Then the set $A \cap Q$ defines the set of paths whose bottleneck nodes have remaining battery capacity higher than γ.

If $A \cap Q \neq \phi$, then CMMBCR chooses a path in $A \cap Q$ by applying the MTPR scheme. Otherwise, CMMBCR selects path *l* with the maximum residual battery capacity :

$$l_b = \max\left\{B_l \left| l \in Q\right.\right\}.$$

If the battery capacity of the bottleneck nodes falls below the protection margin threshold (γ), this path will be avoided to prolong its lifetime. The performance of CMMBCR therefore depends on the value of γ.

A.2 Impact of Protection Margin Threshold (γ) in CMMBCR

In this section, the routing performance of CMMBCR with different threshold γ values is investigated under the low mobility scenario with pause time 120s. The parameter setting in section 3.4.1 is used.

Figure A-1 shows a comparison of the network lifetime. It can be observed that higher value thresholds result in longer network lifetime. The reason is because at the higher threshold values, nodes are protected from being used excessively at the early stage before they exhaust their battery level. Therefore, the network lifetime is extended.

However, although high threshold values permit longer network lifetime, they cannot reduce the average energy consumption along a path, as shown in figure A-2. The reason is because the high threshold values tend to select longer paths, which increase the energy consumption. The energy consumption is minimum when $\gamma = 0$, where CMMBCR always selects the route with the minimum energy consumption.

Figure A-3 shows the standard deviation of the energy consumption per node. The results show that when threshold is high, the standard deviation is low. The reason is because the higher threshold values aim at load balancing, so that the energy at each node in the network is consumed fairly. On the other hand, low threshold values give rise to high standard deviation of energy consumption per node. This is because lower threshold values tend to select paths with minimum energy consumption. As a result, some nodes are selected more often than other nodes in the network as minimum energy routing cannot prevent nodes from being overused.

Table A-1 shows the ratio of successfully delivered packets. The results show that at higher threshold values, a lower ratio of successfully delivered packets is obtained. The reason is due to higher threshold values balance the load. Therefore, the batteries at most nodes are exhausted at the same rate. Consequently, the network connectivity is low later on and decreases the chance of discovering paths and successfully delivering packets.

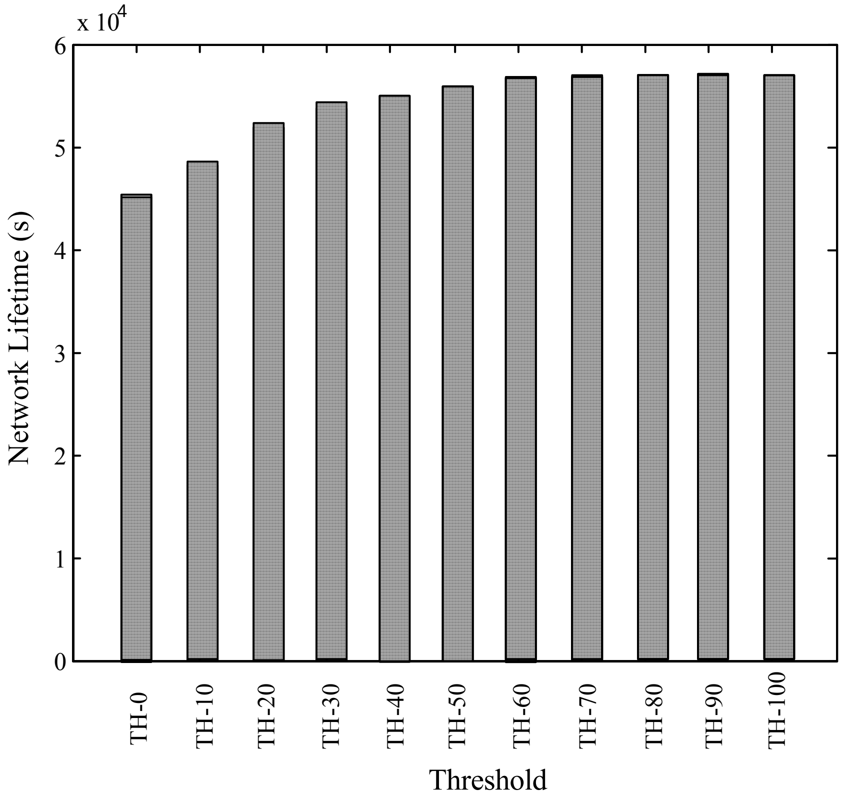


Figure A-1 Comparison of network lifetime with threshold values

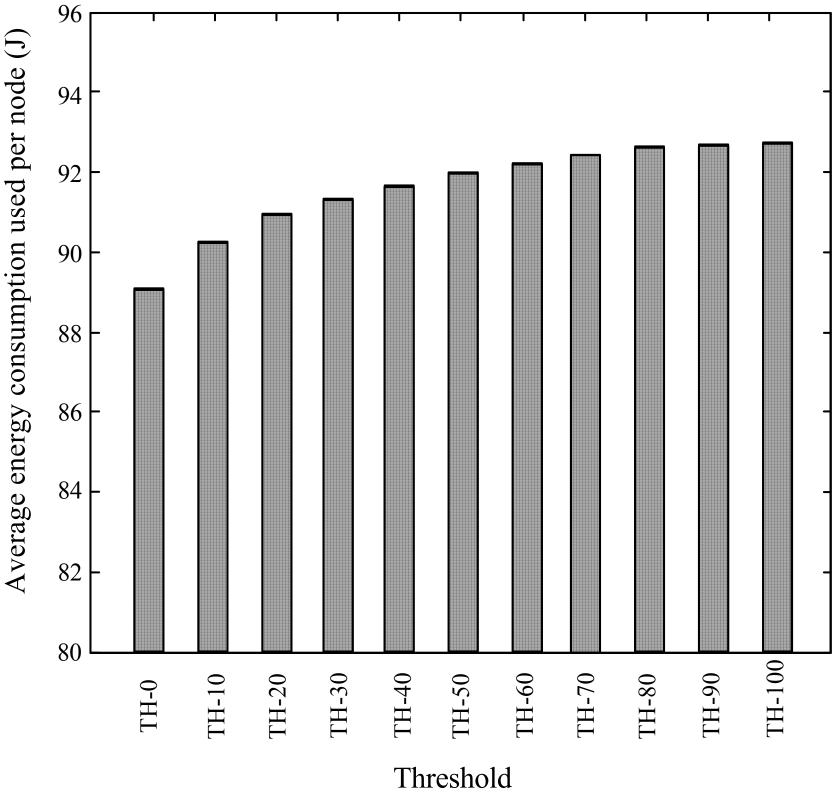


Figure A-2 Comparison of network energy consumption per node with threshold values

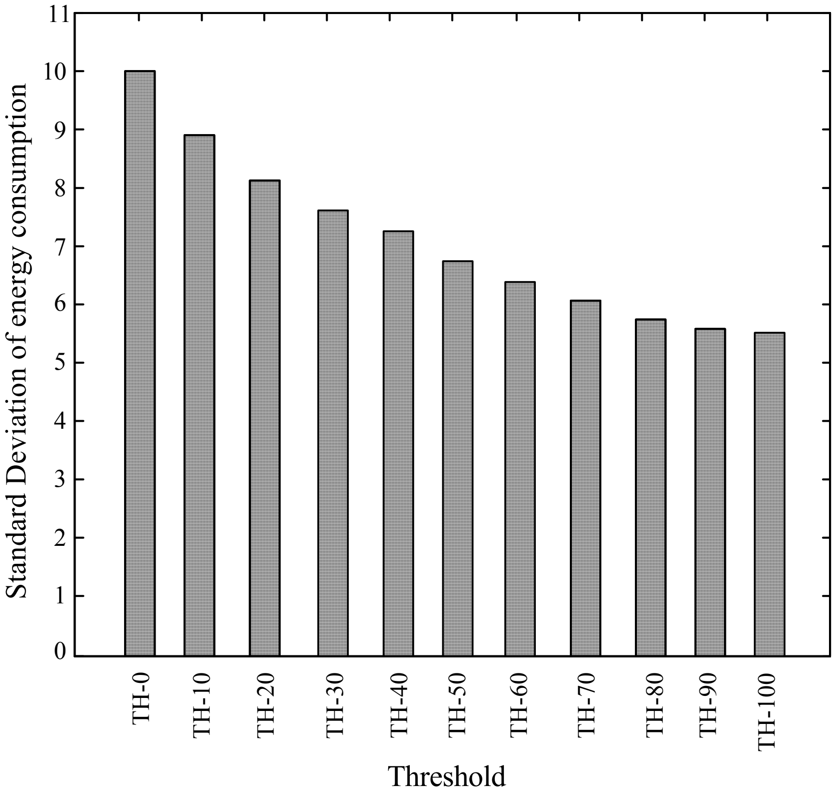


Figure A-3 Comparison of standard deviation with threshold values

Table A-1. Ratio of successfully delivered packets (R_{DP})

Threshold	R_{DP}
0	0.416
10	0.414
20	0.411
30	0.408
40	0.405
50	0.403
60	0.401
70	0.399
80	0.397
90	0.395
100	0.393

A.3 Tradeoff Balancing for Threshold (γ)

Table A-2.Tradeoff between network lifetime and energy consumption

Threshold parameters	**Performance**		
	Network lifetime (s)	Avg. energy consumption	**Tradeoff**
0	45231	89.06	**0.001969**
10	48320	90.22	**0.001867**
20	52203	90.9	**0.001741**
30	54088	91.28	**0.001688**
40	55071	91.62	**0.001664**
50	55811	91.96	**0.001648**
60	56532	92.2	**0.001631**
70	56677	92.42	**0.001631**
80	56827	92.6	**0.001630**
90	56825	92.67	**0.001631**
100	56778	92.69	**0.001632**

Table A-2 compares the tradeoff as the threshold value is varied. Note that the tradeoff is defined as the ratio of energy consumption over network lifetime, so the minimum tradeoff value gives the best performance. In other words, the minimum tradeoff is obtained when energy consumption is minimized and the network lifetime is maximized. From the table, it can be observed that CMMBCR with threshold 80 (TH-80) achieved the smallest tradeoff value. For this reason, we have selected TH-80 as a benchmark for comparison with other algorithms in section 3.4. However, despite TH-80 achieved the best tradeoff when compared with the rest of the threshold values, it still consumes high energy. Other values of thresholds such as TH-60 consumes less energy with a tradeoff value comparable to the TH-80. For this reason, we have selected TH-60 as another algorithm for comparison in section 3.4.

APPENDIX B

List of Publications

List of Publications

Naruephiphat, W. and Usaha, W. Energy-Efficient Routing in Mobile Ad hoc Networks as a Reinforcement Learning Problem. **29th Electrical Engineering Conference 2006.** Page: 749-752.

Naruephiphat, W. and Usaha, W. Balanced Energy-Efficient Routing in MANETs using Reinforcement Learning. **The International Conference on Information Networking 2008.** Page: 1-5.

Naruephiphat, W. and Usaha, W. Balancing Tradeoffs for Energy-Efficient Routing in MANETs based on Reinforcement Learning. **The IEEE 67th Vehicular Technology Conference, Spring 2008.**

BIOGRAPHY

Ms. Wibhada Naruephiphat was born on July 4, 1978 in Pakthongchai District, Nakhon Ratchasima Province. She graduated studying for her primary education at Pakthongchai Chunhawan Vitayakarn School, in 1989. In 1990-1996 she began studying for her secondary and high school education at Pakthongchai Prachaniramit School. In 1997, she began studying for her Bachelors degree at the School of Telecommunication Engineering, Institute of Engineering at Suranaree University of Technology, Nakhon Ratchasima Province. In 2002-2005, she was a teaching assistant and later in 2005 she began studying for her Masters degree at the same school.