การเลือกเส้นทางที่ปลอดภัยในโครงข่ายเคลื่อนที่แบบแอดฮอคด้วยการเรียนรู้แบบรีอินฟอร์สเมนท์

นางสาวกาญจน์กมล มณีนิล

วิทยานิพนธ์นี้เป็นส่วนหนึ่งของการศึกษาตามหลักสูตรปริญญาวิศวกรรมศาสตรมหาบัณฑิต
สาขาวิชาวิศวกรรมโทรคมนาคม
มหาวิทยาลัยเทคโนโลยีสุรนารี
ปีการศึกษา 2549

ISBN 974-533-601-7

A REINFORCEMENT LEARNING APPROACH FOR SECURE ROUTING IN MOBILE AD HOC NETWORKS

Karnkamon Maneenil

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 2006

ISBN 974-533-601-7

A REINFORCEMENT LEARNING APPROACH FOR SECURE ROUTING IN MOBILE AD HOC NETWORKS

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

Thesis Examining Committee

_____

(Dr. Rangsan Tongta)

Chairperson

_____

(Dr. Wipawee Hattagam)

Member (Thesis Advisor)

_____

(Asst. Prof. Dr. Arthit Srikaew)

Member

_____ _____

(Assoc. Prof. Dr. Saowanee Rattanaphani) (Assoc. Prof. Dr. Vorapot Khompis)

Vice Rector for Academic Affairs Dean of Institute of Engineering

กาญจน์กมล มณีนิล : การเลือกเส้นทางที่ปลอดภัยในโครงข่ายเคลื่อนที่แบบแอดฮอคด้วยการเรียนรู้แบบรีอินฟอร์สเมนท์ (A REINFORCEMENT LEARNING APPROACH FOR SECURE ROUTING IN MOBILE AD HOC NETWORKS) อาจารย์ที่ปรึกษา : ดร.วิภาวี หัตถกรรม, 60 หน้า. ISBN 974-533-601-7

วัตถุประสงค์ของงานวิจัยคือ การหานโยบายที่เหมาะสมเพื่อหลีกเลี่ยงโหนดที่ประสงค์ร้าย (malicious nodes) ในการส่งต่อแพ็กเกตข้อมูลและการเลือกโหนดข้างเคียงที่สามารถส่งต่อแพ็กเกตข้อมูลไปยังโหนดปลายทางที่ต้องการในโครงข่ายเคลื่อนที่แบบแอดฮอคได้ ซึ่งนโยบายนี้สามารถเปลี่ยนแปลงได้ตามรูปร่างของโครงข่ายเคลื่อนที่แบบแอดฮอค โดยองค์ความรู้ใหม่ที่ได้รับมี 2 ส่วนคือ

ส่วนที่หนึ่ง ศึกษาวิธีการกำหนดค่าจุดเริ่มเปลี่ยนของเร็พพิวเทชัน (fixed-threshold reputation scheme) เพื่อระบุความน่าเชื่อถือของโหนดในการส่งต่อแพ็กเกตข้อมูล แต่การกำหนดค่าจุดเริ่มเปลี่ยนของเร็พพิวเทชันในระดับคงที่นั้น อาจไม่เหมาะสมกับการทำงานในโครงข่ายเคลื่อนที่แบบแอดฮอค เนื่องจากโครงข่ายมีการเปลี่ยนแปลงรูปร่างตลอดเวลา ดังนั้น งานวิจัยนี้เสนอการเรียนรู้แบบรีอินฟอร์สเมนท์ร่วมกับวิธีเร็พพิวเทชัน เพื่อหานโยบายที่เหมาะสมสำหรับการระบุความน่าเชื่อถือของโหนดในการส่งต่อแพ็กเกตข้อมูลในโครงข่ายเคลื่อนที่แบบแอดฮอค เนื่องจากวิธีการนี้เรียนรู้จากจุดมุ่งหมายโดยตรงแล้วจึงตัดสินใจในการทำตามจุดมุ่งหมายนั้น โดยค่าเร็พพิวเทชันในส่วนนี้ขึ้นอยู่กับแบบจำลองมาร์คอฟเชน (Markov chain) ซึ่งใช้ระบุพฤติกรรมของโหนด และเป็นการทดสอบจุดประสงค์ของการวิจัย โดยไม่คำนึงถึงรูปแบบการส่งต่อแพ็กเกตข้อมูล จากผลการทดลองพบว่าปริมาณงานของโครงข่ายสูงขึ้นถึง 89% เมื่อเปรียบเทียบกับการกำหนดค่าจุดเริ่มเปลี่ยนของเร็พพิวเทชัน

ส่วนที่สอง มีการนำวิธีการเรียนรู้แบบรีอินฟอร์สเมนท์ร่วมกับวิธีเร็พพิวเทชันไปใช้ในโครงข่ายเคลื่อนที่แบบแอดฮอคที่จำลองให้ใกล้เคียงสถานการณ์จริงยิ่งขึ้น โดยกำหนดให้แต่ละโหนดมีบัพเฟอร์ขนาดจำกัด ซึ่งการกำหนดขนาดของบัฟเฟอร์นั้นส่งผลต่อค่าเร็พพิวเทชัน จากผลการทดลองพบว่าปริมาณงานของโครงข่ายสูงขึ้นถึง 71% เมื่อนำไปเปรียบเทียบกับการกำหนดค่าจุดเริ่มเปลี่ยนของเร็พพิวเทชัน

สาขาวิชาวิศวกรรมโทรคมนาคม ลายมือชื่อนักศึกษา_____

ปีการศึกษา 2549 ลายมือชื่ออาจารย์ที่ปรึกษา_____

KARNKAMON MANEENIL : A REINFORCEMENT LEARNING APPROACH FOR SECURE ROUTING IN MOBILE AD HOC NETWORKS

THESIS ADVISOR : WIPAWEE HATTAGAM, Ph.D. 60 PP.

ISBN 974-533-601-7

MALICIOUS NODE/ REINFOURCEMENT LEARNING/ MOBILE AD HOC NETWORKS

The underlying aim of this research is to determine a good rule to distinguish malicious nodes and select cooperative nodes for packet forwarding to target nodes in mobile ad hoc networks (MANETs) which is adaptive to ad hoc environments. The contributions in this research can be classified into two categories.

Firstly, an enhancement to an existing fixed-threshold reputation scheme is proposed. Reputation schemes are used to promote cooperation among nodes through establishment of trust and confidence among nodes in terms of reputation values. However, static reputation values may not be suitable for every ad hoc environment. Hence, we proposed an integration of a reinforcement learning technique with an existing reputation scheme. The rule is adaptive to the network dynamics because it is learned by interacting directly with the environment. In this part, the reputation value of each node is directly obtained from a Markov chain model which allows us to test the proposed approach without complication of packet traffic generation. Numerical studies show that up to 89% of throughput increase can be achieved over the fixed threshold reputation scheme.

Secondly, we extend the previous contribution to a more realistic scenario by generating packet traffic and employing a finite buffer queueing model to characterize the reputation value among the MANET nodes. Numerical studies show a throughput increase of up to 71% over existing fixed-threshold reputation scheme.

School of Telecommunication Engineering Student's Signature_____

Academic Year 2006 Advisor's Signature_____

ACKNOWLEDGEMENT

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

In addition, I am grateful for the teachers of telecommunication engineering: Asst. Prof. Dr. Rangsan Wongsan, Dr. Rangsan Tongta, Dr. Chutima Prommak, Dr. Chanchai Tongsopa and others person for suggestions and all their help.

Finally, I most gratefully acknowledge my parents and my friends for all their support throughout the period of this research.

Karnkamon Maneenil

TABLE OF CONTENTS

Page

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

Page

LIST OF FIGURES

Figure **Page**

SYMBOLS AND ABBREVIATIONS

MANETs	mobile ad hoc networks
RL	reinforcement learning
$\{X(t)\}$	discrete-valued stochastic process
$X(t)$	state of the process at time t
pmf	probability mass function
s	state
a	action
r	reward
MDP	Markov decision process
MC	Monte Carlo method
π	policy
$Q^{\pi}(s,a)$	action value of state-action pair
$r(s,a)$	reward of state-action pair
Rec	recommendation
R	reputation value
R_{thresh}	threshold reputation
λ	arrival rate
μ	service rate
ρ	traffic intensity or offered load
QoS	quality-of-service

SYMBOLS AND ABBREVIATIONS (Continued)

n_m	number of maximum allowed packets

CHAPTER I

INTRODUCTION

1.1 Significance of the Problem

Mobile ad hoc networks (MANETs) are comprised of mobile computing devices which use wireless transmission for communication. MANETs do not have a central administration infrastructure such as base stations in cellular wireless networks or access points in wireless local area networks. Due to the limited range of wireless transmission these mobile devices, so-called nodes, also serve as routers. Therefore, these nodes may need to participate in routing or relaying packets to the designated node (Murty, 2004). MANETs can be deployed widespread because they circumvent the complexity of infrastructure setup. These networks support several applications. For example, MANETs are established communication among a group of soldiers for tactical operations. Furthermore, MANETs can also be used for emergency and rescue operations, by establishing communication among rescue personnel in disaster areas.

A number of issues must be addressed in order to realize the practical benefits of MANETs. These include security of communication in MANETs. The lack of any central coordination makes them more vulnerable to attacks than centralized networks. The major security threats that exist in MANETs are as follows.

1.1.1 Resource consumption

The scarce availability of resources in MANETs makes it an easy target for internal attacks, particularly attacks which aim at consuming resources

available in the network. The major types of resource consumption attacks include:

1.1.2.1 Energy depletion

Since nodes in MANETs are highly constrained in energy source, this type of attack is basically aimed at depleting the battery power of critical nodes by directing unnecessary traffic through them. For example, Yan and Lowenthal (2005) propose a fine-grain cooperation coefficient scheme to quantify the cooperation contribution in order to build an ad hoc network which bandwidth sharing is fair. Luo, Cheng and Lu (2004) propose the Maximize-Local-Minimum Fair Queueing (MLM-FQ) to save energy by allowing sender nodes to schedule multiple packets once it grabs a channel and other nodes to remain in sleep mode during this period.

1.1.2.2 Buffer overflow

The buffer overflow attack is carried out either by filling the routing table with unwanted routing entries or by consuming the data packet buffer space with unwanted data. Such attacks can lead to a large number of data packets being dropped, leading to loss of critical information. Routing table attacks can lead to many problems such as preventing a node from updating route information for important destinations and filling the routing table with routes for nonexisting destinations (Basagni, Conti, Giordano and Stojmenovic, 2004).

1.1.2 Host impersonation

A compromised internal node can act as another node and respond with appropriate control packets to create wrong route entries and can receive the traffic meant for the intended destination node. For example, Rebahi and Sisalem (2005) provide authentication and encryption for detecting incorrect packet forwarding

attacks and denial of service problems

1.1.3 Information disclosure

A compromised node can act as an informer by deliberately disclosing confidential information to unauthorized nodes. Information such as the amount and the periodicity of traffic between a selected pair of nodes and pattern of traffic changes can be very valuable for military applications (Basagni et al., 2004).

1.1.4 Interference

A common attack in defense applications is to jam the wireless communication by creating a wide-spectrum noise. This can be done by using a single wide-band jammed, sweeping across the spectrum. The MAC and the physical layer technologies should be able to handle such external threats. For example, Bianchi (2000) present MAC technique of 802.11 is called distributed coordination function for carrier senses multiple accesses with collision avoidance.

In this thesis, we are interested in a security threat which involves nodes that avoid participating in regular routing and packet forwarding (Buchegger, 2005), which will be referred to hereon as malicious nodes. Malicious nodes arise for several reasons such as to save battery power, bandwidth and processing power and create wrong route entries. The effects of malicious nodes are decreased network throughput and deteriorated network performance such as packets loss, denial of service (Buchegger, 2005). Therefore, this thesis places emphasis on methods that avoid malicious nodes and select good nodes for secure routing in MANETs.

Since selecting good nodes for forwarding packets to designated nodes while avoiding malicious nodes consequently result in high network throughput, there are several current researches on node selection mechanisms for secure routing in

MANETs. Such mechanism should be able to weed out compromized nodes and establish a certain level of node trust. The reputation method is one method which is used to promote cooperation among nodes through establishment of trust and confidence among nodes (Dewan and Dasgupta, 2003). As a result, network throughput is increased because nodes are trusted and cooperate in forwarding packets to target nodes. For example, Liu et al. (2003) proposed a reputation method for MANETs in order to stimulate cooperation among mobile nodes. Vassilaras et al. (2005) present a reputation method for detecting non-cooperating nodes during packet forwarding in clustered MANETs which operate under the coordination and supervision of a central entity. Wang et al (2005) propose a reputation method for detecting and punishing selfish behaving nodes that drop data packets in MANETs. Dewan, Dasgupta and Bhattacharya (2004) show that high network performance in MANETs can be achieved by using a reputation method to identify malicious nodes and find suitable routes for relaying packets that ensure packets will be relayed by cooperative nodes. Although empirical evaluations in the aforementioned works show that reputation schemes can identify misbehaving nodes and improve the performance in MANETs, all of these schemes employ fixed-threshold reputation values for identifying trustworthy nodes. For example, Dewan et al. (2004) use fixed-threshold reputation of 0.5. If a node's reputation value is higher than the fixed-threshold, nodes are considered trustworthy and should be included in the packet forwarding process. On the other hand, nodes should be weeded out when their reputation value is lower than the fixed-threshold. However, fixed-threshold reputation may not be suitable for every ad hoc environment. MANETs change topology frequently therefore fixed-threshold reputation may not be suitable for selecting cooperative nodes. Hence,

reputation thresholds should be adaptive for various ad hoc scenarios.

In this thesis, we study methods that avoid malicious nodes and select well-behaving nodes for forwarding packets based on an adaptive reputation threshold. In particular, we integrate a reinforcement learning (RL) technique with an existing reputation scheme to determine a good rule to distinguish malicious nodes. The advantage of this approach is that the rule is adaptive to the network dynamics because it is learned by interacting directly with the environment. Hence, the underlying aim of this thesis is to determine a good rule to distinguish malicious nodes and select cooperative nodes for forwarding packets to target nodes which is adaptive to ad hoc environments.

Finally, it should be noted that such approach can indeed learn good rules to identify and avoid malicious nodes, resulting in increased network throughput over the fixed-threshold reputation scheme (Karnkamon Maneenil and Wipawee Usaha, 2005). Furthermore, an extension to a more realistic scenario by employing a finite buffer queueing model to characterize the reputation scheme in the MANETs also confirms the advantage of the approach (Wipawee Usaha and Karnkamon Maneenil, 2006).

1.2 Research Objective

The objective of this research is organized as follows:

1.2.1 To select a suitable path(s) for forwarding packets which improve network throughput in MANETs.

1.2.2 To select cooperative and trustworthy neighboring nodes as well as avoid malicious nodes in MANETs.

1.2.3 To reduce the number of loss packets which arrive at the destination node as the number of malicious nodes is increased.

1.2.4 To increase the network relative throughput as the number of malicious nodes is increased.

1.3 Assumptions

1.3.1 Reinforcement learning can increase the relative throughput in MANETs when the number of malicious nodes is increased.

1.3.2 Reinforcement learning can find secure paths from the source node to the destination node when the number of malicious nodes is increased.

1.3.3 Reinforcement learning is used to forward packets from the source node to the destination node when the number of malicious nodes is increased.

1.4 Scope

The experiment is separated into two parts. The first part involves a study of secure network functionalities that are necessary to defend attacks from malicious nodes. In this part, we study a reputation scheme combined with a reinforcement method to learn good rules to identify and therefore select behaving nodes as well as avoid malicious nodes. From the numerical study, four metrics are compared, namely, the accumulated reward per episode, the number of packets arrived at the destination, relative throughput, the number of packets arrived at the destination when the number of maximum allowed packets is decreased. We compare these metrics among three

routing schemes, namely, a reputation scheme with fixed threshold (Dewan, 2004), a reputation scheme combined with the reinforcement learning method and the shortest path scheme which disregards the reputation values. Experiments are conducted under both static and dynamic topology cases. In the dynamic topology case, we generate the topology using a random connectivity model where link between nodes are formed probabilistically.

The second part extends the study from the first part to a more realistic scenario by employing a finite buffer queueing model to characterize the reputation scheme among the MANET nodes. In this part, malicious nodes are characterized by the size of node buffer. The experiments are conducted with the same metrics, topology dynamics and routing schemes as in part one.

1.5 Expected Usefulness

1.5.1 To obtain an algorithm that can avoid malicious nodes in MANETs.

1.5.2 To obtain an algorithm program that can increase network throughput and improve the ability to find secure paths.

1.5.3 To obtain a conclusion about the application of reinforcement learning in secure routing and suggest possible application for other resource allocation problems in MANETs.

1.6 Synopsis of Thesis

This thesis is organized as follows. **Chapter 2** gives a brief introduction to the

reinforcement learning technique and the queueing model used for secure routing in this thesis.

Chapter 3 proposes an enhancement to an existing reputation method for indicating and avoiding malicious hosts in MANETs. The proposed method combines a simple reputation scheme with a reinforcement learning technique called the on-policy Monte Carlo method (ONMC) where each mobile host distributively learns a good policy for selecting neighboring nodes in a path search.

Chapter 4 extends the contribution of the previous chapter to a more realistic scenario by employing a finite buffer queueing model to characterize the reputation value among the MANET nodes. The advantage of approach is that the rule is adaptive to a more realistic network dynamics.

Finally, **Chapter 5** summarizes all the original contributions in this thesis and provides recommendation for possible further work.

CHAPTER II

BACKGROUND THEORY

MANETs are comprised of nodes which use wireless transmission for communication. These nodes may need to participate in routing or relaying packets to the destination node. Such a network needs external motivation to make the nodes cooperate for forwarding packets. Threshold reputation can be used to promote cooperation for packet forwarding and detect malicious nodes (Dewan and Dasgupta, 2003). Reputation values can be used to quantify the behavior of such nodes. In particular, if a node has reputation value that is higher than a certain threshold, it is considered trustworthy for forwarding packets. Note that the reputation value of each node is characterized by the number of packets it has received and forwarded, as well as the latest reputation value attained (Dewan et al., 2004). In such scenario, changes of the reputation value at each node may be viewed as a Markov process where the updated (future) value of the reputation depends on the present reputation value only and independent of the past values.

2.1 Markov Processes

Let $\{X(t)\}$ be a discrete-valued stochastic process where $X(t)$ refers to the state of the process at time t. If the future of the process, given that the process is presently in state $X(t_k)$, is independent of the past, then $\{X(t)\}$ is called Markov

process. That is $\{X(t)\}$ is a Markov process if

$$\begin{aligned} P[X(t_{k+1}) = x_{k+1} \mid X(t_k) = x_k, ..., X(t_1) = x_1] \\ = P[X(t_{k+1}) = x_{k+1} \mid X(t_k) = x_k], \end{aligned} \tag{2.1}$$

where $t_1 < t_2 < \cdots < t_k < t_{k+1}$, t_k is the present and t_{k+1} is the future. We refer to Eq. (2.1) as the Markov property.

A discrete-valued Markov process is called a Markov chain. If $\{X(t)\}$ is a Markov chain, then the joint probability mass function (pmf) for three arbitrary time instants is

$$\begin{aligned} &P[X(t_3) = x_3, X(t_2) = x_2, X(t_1) = x_1] \\ &\quad = P[X(t_3) = x_3 \mid X(t_2) = x_2, X(t_1) = x_1] \times P[X(t_2) = x_2, X(t_1) = x_1] \\ &\quad = P[X(t_3) = x_3 \mid X(t_2) = x_2] \times P[X(t_2) = x_2, X(t_1) = x_1] \\ &\quad = P[X(t_3) = x_3 \mid X(t_2) = x_2] \times P[X(t_2) = x_2, \mid X(t_1) = x_1] \times P[X(t_1) = x_1], \end{aligned}$$

where we have used the definition of conditional probability and the Markov property. In general, the joint pmf for $k+1$ arbitrary time instants is

$$\begin{aligned} P[X(t_{k+1}) = x_{k+1}, X(t_k) = x_k, ..., X(t_1) = x_1] \\ = P[X(t_{k+1}) = x_{k+1} \mid X(t_k) = x_k] \times P[X(t_k) = x_k \mid X(t_{k-1}) = x_{k-1}] \cdots \\ \times P[X(t_2) = x_2, \mid X(t_1) = x_1] \times P[X(t_1) = x_1]. \end{aligned} \tag{2.2}$$

2.1.1 Discrete-Time Markov Chain

Let $\{X_n\}$ be a discrete-valued Markov chain that starts at $n = 0$ with pmf

$$p_j(0) = P[X_0 = j], \tag{2.3}$$

where $j = 0, 1, 2, \ldots$.

From Eq. (2.2), the joint pmf for the first $n+1$ values of the process is

$$\begin{aligned} &P[X_n = i_n, \ldots, X_0 = i_0] \\ &\quad = P[X_n = i_n \mid X_{n-1} = i_{n-1}] \cdots \times P[X_1 = i_1 \mid X_0 = i_0] \times P[X_0 = i_0]. \end{aligned} \tag{2.4}$$

Thus, the joint pmf for a particular sequence is simply the product of the probability for the initial state and the probabilities for the subsequent one-step state transitions.

We will assume that the one-step state transition probabilities are fixed and do not change with time, that is,

$$P[X_{n+1} = j \mid X_n = i, a_n = a] = p_{ij}^a, \tag{2.5}$$

for all n. Now suppose that, in order to transit into a new state, an action from a set of all possible actions must be taken. If an environment has the Markov property, then its one-step dynamic enables us to predict the next state and expected next reward given the current state and action. This is referred to as Markov Decision Process.

2.1.2 Markov Decision Process (MDP)

If the state and action spaces are finite, the MDP is called a finite MDP. A particular finite MDP is defined by its state and action sets and by the one-step dynamics of the environment. Given a state s and action a, the probability of each transiting into the next state s' is given by

$$p_{ss'}^{a} = P[X_{n+1} = s' \mid X_n = s, a_n = a]. \tag{2.6}$$

These quantities are called transition probabilities. Similar, given a current state s and action a, the expected value of the next reward is given by

$$r_{ss'}^{a} = E[r_{n+1} \mid X_n = s, a_n = a, X_{n+1} = s']. \tag{2.7}$$

The quantities, $p_{ss'}^{a}$ and $r_{ss'}^{a}$, completely specify the most important aspects of the dynamics of a finite MDP.

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

2.2 Reinforcement Learning

Reinforcement learning (RL) is a computational approach for goal-directed learning and decision-making (Sutton, 1998). The learner or decision maker is called the agent. Everything outside the agent is called environment. It uses a formal framework defining the interaction between a learning agent and its environment in terms of states (s_t), actions (a_t) and rewards (r_t). The agent selects actions and the environment response to those actions. Furthermore, the environment also gives rise to rewards that agent tries to maximize over time. More specifically, the agent and environment interact with each in a sequence of discrete time steps. At each time step

(t), the agent receives some representation of the environment's state (s_t) and selects an action (a_t). One time step later, the agent receives a numerical reward (r_{t+1}) and finds itself in a new state (s_{t+1}). Figure 2.1 shows the agent-environment interaction in reinforcement learning.

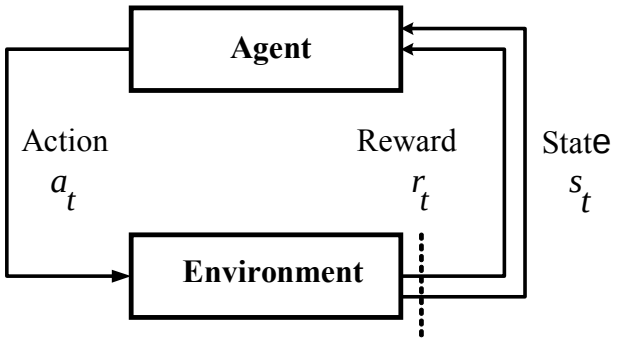

Figure 2.1 Diagrams of agent-environment interaction in reinforcement learning.

Furthermore, the agent implements a mapping from state to probabilities of selecting each possible action. This mapping is called the agent's policy. RL methods specify how the agent changes its policy as a result of its experience. The agent's objective is to maximize the total amount of reward it receives over the long run.

The function of future rewards that the agent seeks to maximize is called return. It has several different definitions depending upon the nature of the task and whether one wishes to discount delayed reward. The undiscounted formulation is appropriate for episodic tasks, in which the agent-environment interaction breaks naturally into episodes. The discounted formulation is appropriate for continuing tasks, in which the interaction does not naturally break into episodes but continues

without limit.

An environment satisfies the Markov property if its current state signal compactly summarizes the past without degrading the ability to predict the future state. This is rarely exactly true, but often nearly so. The state signal should be chosen or constructed so that the Markov property holds as nearly as possible. If the Markov property does hold, then the interaction with the environment defines a Markov decision process (MDP). A finite MDP is an MDP with finite state and action sets. Most of the current theory of reinforcement learning is restricted to finite MDPs, but the methods and ideas can be to continuous state and action sets generally (Sutton, 1998).

The expected return from the state (or state-action pair) is called the value function of the state (or state-action pair) under a given policy. The optimal value functions assign to each state (or state-action pair) the largest expected return achievable by any policy. A policy whose value functions are optimal is an optimal policy. Whereas the optimal value functions for states and state-action pairs are unique for a given MDP, there can be many optimal policies. Any policy that is greedy with respect to the optimal value functions must be an optimal policy.

RL framework has proved to be widely useful and applicable. For example, in Wipawee Usaha (2004) applied reinforcement learning for path discovery in MANETs that can indeed achieve message overhead reduction with marginal difference in the path search ability with reasonable computational and storage requirements. Cheng (2004) showed that reinforcement learning method can be used

to control both packet routing decisions and node mobility, dramatically improving the connectivity of the ad

hoc networks. The RL tool which use to apply in this thesis is Monte Carlo Method.

2.2.1 Monte Carlo Method (MC)

Monte Carlo (MC) methods are ways of solving the reinforcement learning problem based on averaging sample returns. To ensure that well-defined returns are available, MC methods are defined only for episodic tasks. Hence, 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. MC methods are thus incremental in an episode-by-episode sense. The term "MC" is often used more broadly for any estimation method whose operation involves a significant random component. Here we use it specifically for methods based on averaging complete returns.

2.2.2 Monte Carlo Estimation of Action Values

The expected return when starting in state (s), taking action (a) and thereafter following policy (π) is called the action value of state-action pair (s,a) under policy π, $Q^{\pi}(s,a)$. There are two methods which used to estimate the value of state-action pair. The every-visit MC method estimates the value of a state-action pair as the average of the returns that have followed visit to the state in which the action was selected. The first-visit MC method averages the returns following the first time in each episode that the state was visited and the action was selected.

If π is a deterministic policy, then in following π return will be available only for one of the action from each state. With no returns to average, the MC estimates of the other actions will not improve with experience. This is a serious problem because the purpose of learning action values is to help in choosing among the actions available in each state. To compare alternatives we need to estimate the value of all the actions from each state, not just the one we currently favor.

This is the general problem of maintaining exploration. For policy evaluation to work for action values, continual exploration must be assured. One way to do this is by specifying that the first step of each episode starts at a state-action pair, and that every such pair has nonzero probability of being selected as the start. This guarantees that all state-action pairs will be visited an infinite number of times in the limit of an infinite number of episodes. Hence, call this the assumption of exploring starts (Sutton, 1998).

2.2.3 Monte Carlo Control

In generalized policy iteration one maintains both an approximate policy and an approximate the value function. The value function is repeatedly altered to a closer approximate value function for the current policy, and the policy is repeatedly improved with respect to the current value function.

Consider a MC version of classical policy iteration. In this method, we alternate complete steps of policy evaluation and policy improvement, beginning with an arbitrary policy (π_0) and ending with the optimal policy and optimal action-value function as follows

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

where $\xrightarrow{E}$ denotes a complete policy evaluation and $\xrightarrow{I}$ denotes a complete policy improvement. Policy evaluation is done exactly as described in the preceding section. In addition, the episodes are generated with exploring starts. Under these assumptions, the MC methods will compute each Q^{π_k} exactly, for arbitrary π_k.

Policy improvement is done by making the policy greedy with respect to the current action-value function. For any action-value function (Q), the corresponding greedy policy is the one that, for each $s \in S$, deterministically chooses an action with maximal Q value as follows

$$\pi(s) = \arg\max_a Q(s,a). \tag{2.8}$$

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

$$\begin{aligned} Q^{\pi_k}(s, \pi_{k+1}(s)) &= Q^{\pi_k}(s, \arg\max_a Q^{\pi_k}(s,a)) \\ &= \max_a Q^{\pi_k}(s,a) \\ &\geq Q^{\pi_k}(s, \pi_k(s)) \\ &= V^{\pi_k}(s). \end{aligned} \tag{2.9}$$

Note that each π_{k+1} is uniformly better than π_k, unless it is equal to π_k which is the case only when they are both optimal policies. This is turn assures us that the overall

process converges to an optimal policy and the optimal value function. In this way MC methods can be used to find optimal policies given only sample episodes and no other knowledge of the environment's dynamics. In this thesis, we employ a MC method that is called On-Policy Monte Carlo method.

2.3 On-Policy Monte Carlo Method (ONMC)

In this thesis, we employ a learning approach based on sample episodes, called the on-policy Monte Carlo (ONMC) method (Sutton, 1998). This method uses sample episodes for specify what is good in the long run. The ONMC method learns incrementally on an episode-by-episode basis meaning that the values are estimated and policies are improved after each episode. Under certain assumptions, the ONMC method eventually converges to an optimal policy and optimal value functions—given only sample episodes and no other knowledge of the environment's dynamics.

Let sets of states and actions be denoted S and A, respectively. We consider each episode that the state-action pair (s,a) was visited where $s \in S$ and $a \in A$.

Let π_0 be the initial policy. For each episode *t*, let the action be generated according to π_t. At the end of episode *t*, the estimated state-action value function of (s,a) is updated according to

$$Q^{\pi_t}(s,a) = Q^{\pi_{t-1}}(s,a) + \frac{1}{t}\left(\sum_{n=\tau_t(s,a)}^{N_{t-1}} r(s_n,a_n) - Q^{\pi_{t-1}}(s,a) \right), \tag{2.10}$$

where N_t is the number of time steps in episode t, $\tau_t(s,a)$ is the time step when the first visit of (s,a) occurred and $r(s,a)$ is the reward for taking action a at state s. Note that the summation term is the accumulative reward following only the first occurrence of (s,a). Thus the greedy policy is found by

$$a^* = \arg\max\left\{Q^{\pi_t}(s,a)\right\}, \tag{2.11}$$

and the $\varepsilon - soft$ on-policy, $\varepsilon \in [0,1]$ is implemented as follows

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

where $|A|$ is the size of action space.

The ONMC method is selected in this thesis because the episodic nature of route search process in mobile ad hoc networks. An episode starts immediately when a source node initiates a route search to a destination node, and terminates when the target node is found or the maximum number of hop count is reached. Each time the search is successful, a reward is to every node along all paths found. The goal is to find a rule that selects neighboring nodes which optimizes the average returns in the long run. The ONMC method is integrated with an existing secure route discovery scheme in the next chapter.

CHAPTER III

SECURE ROUTING IN MANETS : A REINFORCEMENT LEARNING PROBLEM

3.1 Introduction

In mobile ad hoc networks (MANETs), each host has a limited transmission range. Successful delivery of packets between hosts outside transmission range of each other therefore relies on cooperation of intermediate nodes. The fundamental assumption for such networks is that the nodes will cooperate and not misbehave. However, hosts join the network on the fly creating a dynamic topology network. The lack of a centralized network management leads ad hoc networks vulnerable to attacks by misbehaving nodes. Consequently, packets are dropped or even misdirected therefore resulting in low network throughput.

Figure 3.1 illustrates an ad hoc network which contains malicious nodes in the shortest paths. With some quantification of node misbehavior, malicious nodes can be identified and the source node is able to send packets along an alternative path such as through nodes 4, 5 and 6.

Recently, reputation schemes have been employed to identify and avoid malicious nodes. The reputation of a node is a function of only the number of data packets that have been previously relayed by the node (Dewan et al., 2004). Hence, nodes have high reputation when they successfully forward packets they receive.

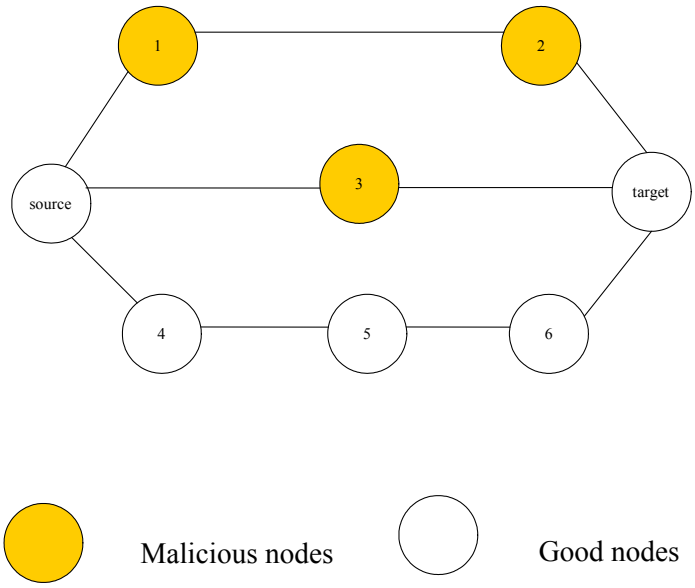

Figure 3.1 An ad hoc network with malicious nodes.

If the nodes have low reputation, it is subsequently weeded out from the ad hoc network. Reputation is an average of recommendations received by a node. Suppose node A receives 100 packets and routed 60 packets but dropped 40 packets. Hence total reputation of node A becomes (60-40)/100 = 0.2. Reputation values vary stochastically depending on the traffic load and behavior of nodes themselves. More details on the reputation method is given in section 3.2.

Liu et al. (2003) propose a reputation method for MANETs in order to stimulate cooperation among mobile nodes. Vassilaras et al. (2005) present a reputation method for detecting noncooperative nodes during packet forwarding in clustered MANETs which operate under the coordination and supervision of a central entity. Wang et al (2005) propose a reputation method for detecting and punishing selfish behavior nodes in ad hoc networks where reputation values are evaluated by direct observation. Dewan et al. (2004) show that high network throughput can be achieved when using their reputation method to ensure packets will be relayed by

cooperative nodes in the MANET.

So far the aforementioned works have employed fixed values of reputation thresholds to discriminate cooperative nodes from noncooperative nodes. However, such static values may not be suitable for every ad hoc environment. In this chapter, we integrate a reinforcement learning technique with an existing reputation scheme (Dewan et al., 2004) and determine a good rule to distinguish malicious nodes. The advantage of our approach is that the rule is adaptive to the network dynamics because it is learned by interacting directly with the environment.

3.2 Reputation Method

Reputation methods can be used to detect various types of misbehaving nodes. These methods monitor and rate the behavior of other nodes in the routing and packet forwarding process so that the node under consideration can respond according to its opinion about other nodes. The opinion a node has of another node is called reputation. The goal of a reputation system is to enable nodes to make informed decisions about which nodes to cooperate with or exclude from the network (Buchegger, 2005). Reputation systems can be used to cope with any kind of misbehavior as long as it is observable.

In this thesis, the reputation of a node is a function of the number of packets that have been previously relayed by the node. In the proposed reputation scheme, the source node finds a set of paths to the destination by using a MANET routing protocol. The source node sends the packet to the adjacent node with the highest reputation. This node then forwards the packet to the next hop with the highest reputation and the process is repeated until the packet reaches its destination. If there

is a malicious node in the route, the packet does not reach its destination.

The advantages of the reputation scheme include:

3.2.1 Circumvent of malicious nodes.

3.2.2 Injection of motivation to cooperate among nodes.

3.2.3 Decentralized collection and storage of reputations

3.2.4 Subsequent increase in the average throughput of the ad hoc network.

3.2.1 Dewan's Reputation Scheme (Dewan et al., 2004)

In this section, we define some of more commonly used term in this thesis and introduce components of Dewan's reputation scheme.

3.2.1.1 Recommendation

Recommendation is the value assigned to the service provider by the service seeker during a transaction. All the recommendations of the service provider are combined to evaluate its reputation. For example, a node can obtain recommendations from its availability, accuracy and efficiency. In this thesis, we consider only one context of recommendation that is the number of forwarded packets, which directly relates to nodes' the resource availability. In most cases, a good node routes the received packets to the next hop even if it has no vested interest in the packet. A bad node maliciously drops the packets or tampers with the contents of the packet or routes it in the wrong direction. For example, consider a network that consists of three nodes as follows $A \rightarrow B \rightarrow C$. If node A wants to send a packet to node C and it finds out that the only way which it can send the packet to node C is via node B. It then sends the packet to node B, which in turn routes it to node C. If node C acknowledges receiving the packet to node A, node A can then deduce that

node B routed the packet properly. In such case, suppose that node A gives node B a recommendation of +1. Mathematically, the recommendation will appear as follows

$$Rec_{AB} = +1 .$$

3.2.1.2 Reputation Value

Reputation value is the mean of the recommendations received by a node. Suppose node B received 100 packets and routed 90 packets but dropped 10 packets, the sender of the routed packets give node B a recommendation of +1 and the sender of the dropped packets give node B a recommendation of -1. Hence, the total reputation of node B is given by

$$R_B = \frac{(90-10)}{100} = \frac{80}{100} = 0.8 .$$

3.2.1.3 Node Identifier

Each node possesses a certificate which was issued to it when the network was established. Each node possesses a single unique identity. The reputation is assigned to the node identity. The nodes in the network can easily verify the identity of a particular node in the network by using challenge response mechanisms (Dewan et al., 2004). If a node gets compromised and does not cooperate, its reputation decreases quickly and soon it is weeded out of the system, even if it possesses an authentic identity.

3.2.1.4 Threshold Reputation

The threshold reputation, R_{thresh} , is the minimum reputation a node expects from a possible next hop node in a path. If the next hop node does not

possess the required reputation, it will not be included in the packet forwarding process. Note that only the source node can send a packet to a node whose reputation is less than the threshold. In this method, all nodes in the network use the reputations of their neighboring nodes to find out the best node which the packet should be forwarded to.

3.3 Reputation as a Reinforcement Learning Problem

Reinforcement learning (RL) is a computational approach for goal-directed learning and decision-making (Sutton, 1998). The learner or decision maker is called the agent. Everything outside the agent is called environment. It uses a formal framework defining the interaction between a learning agent and its environment in terms of states (s_t), actions (a_t) and rewards (r_t). The agent selects actions and the environment responses to those actions. Furthermore, the environment also gives rise to rewards of which the agent tries to maximize over time. More specifically, the agent and environment interact in a sequence of discrete time steps. At each time step t, the agent receives some representation of the environment's state (s_t) and selects an action (a_t). One time step later, the agent receives a numerical reward (r_{t+1}) and finds the environment in a new state (s_{t+1}).

Furthermore, the agent implements a mapping from environment states to probabilities of selecting each possible action. This mapping is called the agent's policy. RL methods specify how the agent changes its policy as a result of its experience. The agent's objective is to maximize the total amount of reward it receives over the long run.

In this chapter, we employ a learning approach based on sample episodes, called the on-policy Monte Carlo (ONMC) method (Sutton, 1998) that is explained in the previous chapter. This method uses sample episodes to specify how well an action at a given state is in the long run. The ONMC method learns incrementally in an episode-by-episode basis which means the value functions are estimated (section 2.2.2) and policies are improved (section 2.2.3) after each episode. Under certain assumptions, the ONMC method eventually converges to an optimal policy and optimal value functions (Sutton, 1998), given only sample episodes and no other knowledge of the environment's dynamics.

The ONMC method is selected because the episodic nature of route search process in mobile ad hoc networks. An episode starts immediately when a source node initiates a route search to a target node, and terminates when the target node is found or the maximum number of hop count is reached. As the route search is executed, the intermediate nodes are selected hop-by-hop based on their reputation values. Each time the route search is successful, a reward is assigned to every node along all paths discovered. The goal is to find a rule that selects neighboring nodes based on their reputation values which optimizes some performance criterion in finite horizon. Note that the finite horizon problem is considered here due to the episodic nature of route search process.

3.4 Problem Formulation

The reputation scheme based on the ONMC method is applied to MANETs in order to obtain a trustworthy neighboring node selection policy. Consider a N-node ad hoc networks. Each node maintains reputation values of all its neighboring nodes.

Suppose that nodes A, B, C and D are neighbors of node S. The reputation value of a node is derived from the number of packets forwarded by a node divided by the number of packets which this node receives.

Let R_A, R_B, R_C and R_D be the reputation values of nodes A, B, C and D, respectively where

$$0 \le R_A, R_B, R_C, R_D \le 1. \tag{3.1}$$

Since the reputation values are real numbers, we quantize the state space at node S as

$$X_S = \{x_S : x_S = [q_A, q_B, q_C, q_D]\}, \tag{3.2}$$

where q_A, q_B, q_C and q_D are quantized reputation values of nodes A, B, C and D, respectively. For example, node S with n neighboring nodes and quantize reputation values into l subintervals, $|X_S| = l^n$ where $|X_S|$ is size of state space.

The actions are the choices made by the agent. Let the action space at node S be given by

$$A_S = \{a_S : a_S = [\delta_A, \delta_B, \delta_C, \delta_D]\}, \tag{3.3}$$

where $\delta_A, \delta_B, \delta_C$ or δ_D is the unity if node S selects node A, B, C or D in the route search and zero, otherwise. Therefore, node S with n neighboring nodes has $|A_S| = 2^n$ entries, that is, $[0,0,0,0], \cdots, [1,1,1,1]$. The process is repeated at every selected node until the destination node is found or the maximum number of hop counts is reached.

If the route search is successful, then a reward of 1 is assigned to every node on all successful paths. Otherwise, no reward is assigned to all nodes involved in the route search.

By using the ONMC method in this scenario, we are able to determine an optimal neighboring node selection policy based on reputation values.

3.5 Experimental Results

We consider a MANET of 23 nodes which includes a number of misbehaving nodes as shows in figure 3.2. Two cases of topologies have been considered, i.e., the static and dynamic topology. In the latter case, the topology of the network is generated by a random connectivity model. Reputation values between 0 and 1 at each node reflect how trustworthy of a node is—the higher reputation values, the more reliable the nodes are. Since reputation values are continuous values, the state space is quantized into 5 subintervals, $[0,0.2),[0.2,0.4),[0.4,0.6),[0.6,0.8)$ and $[0.8,1.0]$ which are represented by integers 1, 2, 3, 4 and 5, respectively.

We assume that each node has four neighboring nodes, so the state space of node S has total of $5^4 = 625$ possible states. For example, suppose some node S has nodes A, B, C and D as neighboring nodes with reputation values R_A, R_B, R_C and R_D, respectively. The state $x_S = [4,2,1,3]$ refers to the state of node S which has neighbors with reputation values in intervals 4, 2, 1 and 3, that is, $R_A \in [0.6,0.8)$, $R_B \in [0.2,0.4)$, $R_C \in [0.0,0.2)$ and $R_D \in [0.4,0.6)$ respectively.

In this chapter, we compare three schemes, namely, Dewan's reputation scheme with a fixed reputation threshold of 0.5 (Dewan et al., 2004), and Dewan's

reputation scheme combined with the ONMC method and a shortest path scheme which disregards the reputation values.

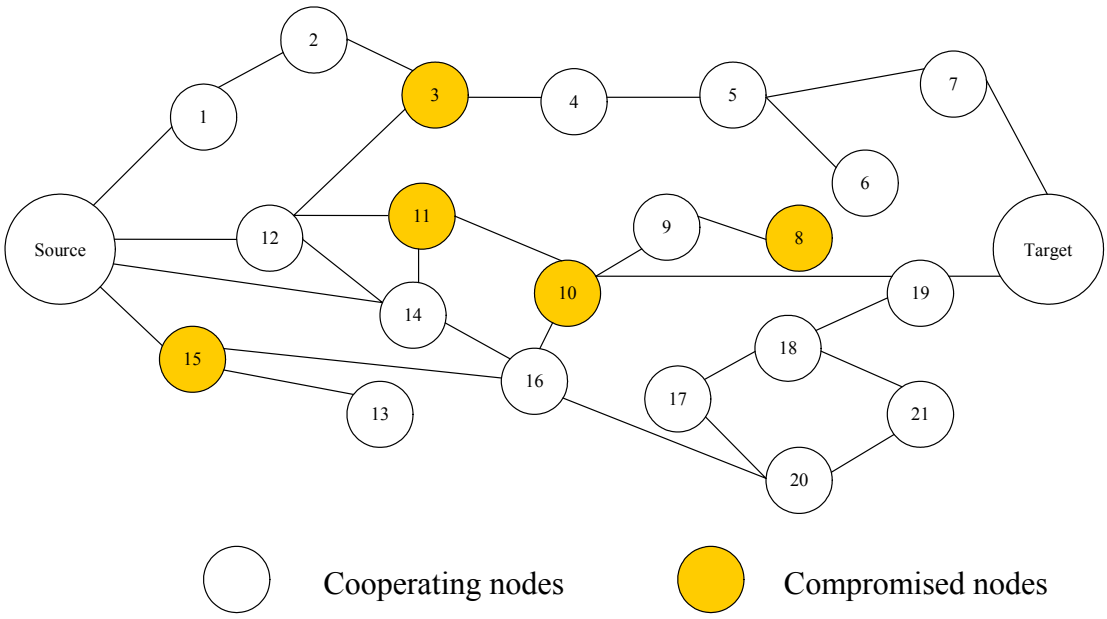

Figure 3.2 Mobile ad hoc networks

To assess the performance, we use the following metrics, namely, the accumulated reward per episode, the number of packets arrived at the destination and the relative throughput[1]. In addition, we also study the effect of reducing the maximum allowed packets of node.

3.5.1 Accumulated Reward per Episode

Figure 3.3 shows the accumulated reward per episode as the number of malicious nodes in the network increases for the static and dynamic topology cases. The maximum number of allowed packets (n_m) broadcasted in the network is 1000.

[1] The relative throughput = $\frac{throughput}{throughput_{reputation\ only}}$

Under both topology cases, the ONMC scheme outperforms the other two schemes as the number of malicious nodes increases.

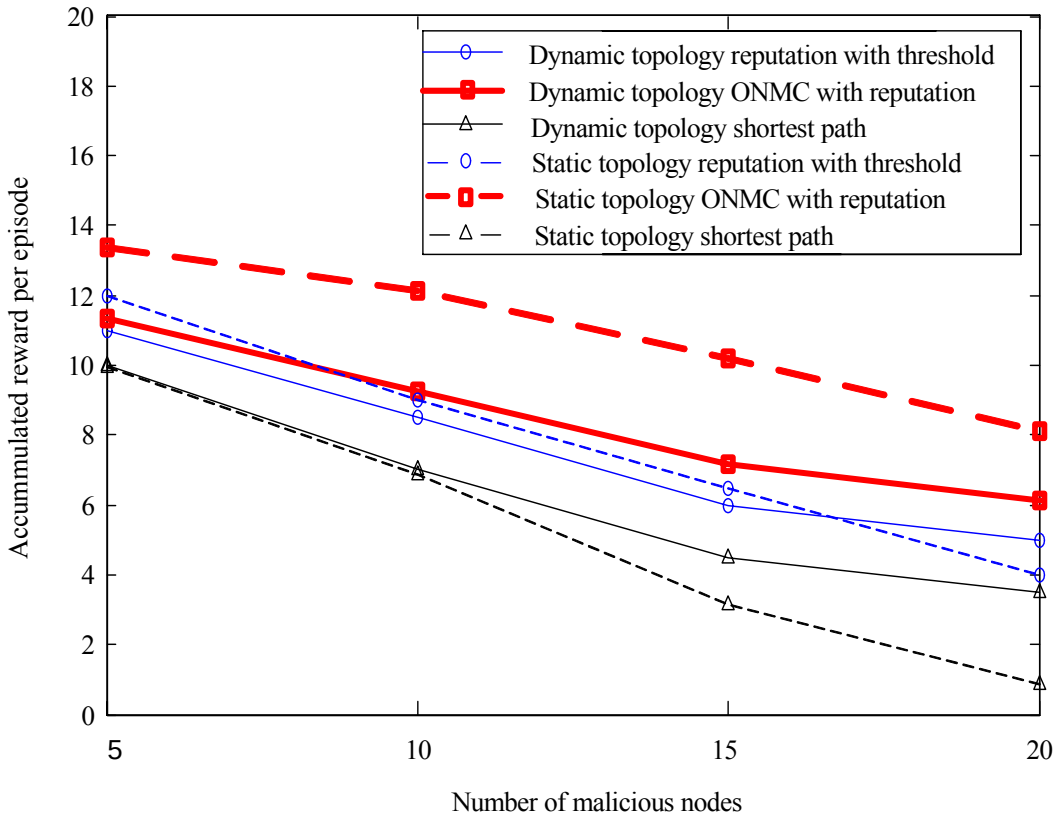

Figure 3.3 Accumulated reward per episode.

The reason is because the ONMC method can attain good policies for avoiding malicious nodes and is able to find more successful routes when compared to other schemes. As multiple successful paths are found, a reward of +1 is assigned to every node on all successful paths. Therefore, the accumulated reward per episode of the ONMC method is the highest among the schemes. On the other hand, the accumulated reward per episode of the reputation scheme with threshold of 0.5 is consistently lower than that of the ONMC scheme for both topology cases because fix-valued threshold may not be suitable for every ad hoc environment. The accumulated reward

per episode of the shortest path scheme is the lowest of all because it does not consider any reputation values in avoiding malicious nodes.

3.5.2 Number of Packets Arrived at the Destination

Figure 3.4 shows the number of packets arrived at the destination for static and dynamic topologies as the number of malicious nodes in the network increases. The maximum number of allowed packets (n_m) broadcasted in the network is 1000. Results show that the reputation with ONMC scheme consistently gives the highest number of packets under both topologies. The reason is because the ONMC method learns its decision through direct interaction with the environment and can eventually learn to select suitable neighboring nodes to forward the packets. On the other hand, packets are dropped more in the fixed-threshold reputation and shortest path schemes as they cannot identify malicious nodes as effective as the ONMC scheme.

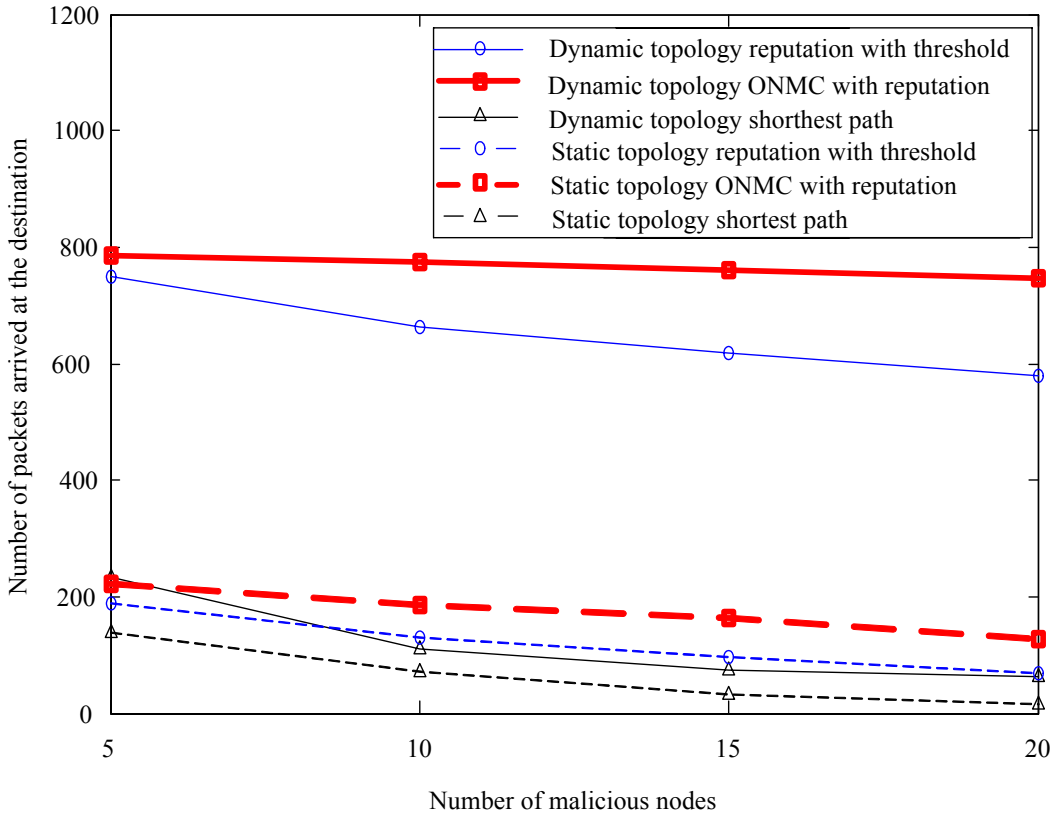

Figure 3.4 The number of packets arrived at the destination with n_m =1000.

3.5.3 Relative Throughput

Figure 3.5 shows the relative throughput as the number of malicious nodes in the network increases. Results show that the reputation with ONMC scheme can achieve up to 89% and 29% increase in throughput over the fixed-threshold reputation scheme for static and dynamic topologies cases, respectively. Furthermore, the relative throughput of the ONMC scheme is the highest for both topology cases because it can deliver the most number of packets to the destination. For, the shortest path scheme we observe upto 75% and 90% reduction in throughput when compared of the fixed- threshold reputation scheme for static and dynamic topologies cases, respectively.

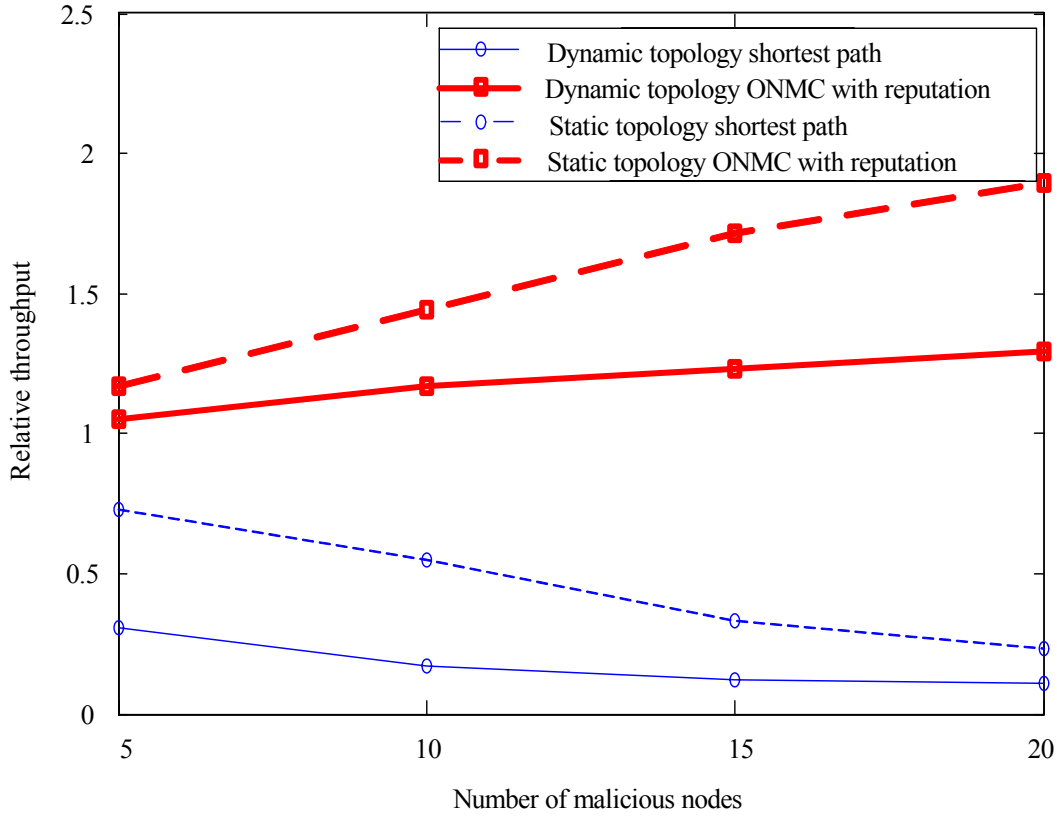

Figure 3.5 The relative throughput.

3.5.4 Effect of Varying the Maximum Allowed Packets

So far, the number of maximum allowed packets (n_m) is fixed at 1000. Figure 3.6 shows the performance in terms of the number of packets arrived at the destination as we reduce the maximum allowed packets. The number of malicious nodes is fixed at 5. Results show that the ONMC scheme still gives a significantly higher number of packets arrivals compared to the other scheme. The results of the shortest path scheme are not shown here as it performed the worst compared to other schemes as evidently shown in previous figures.

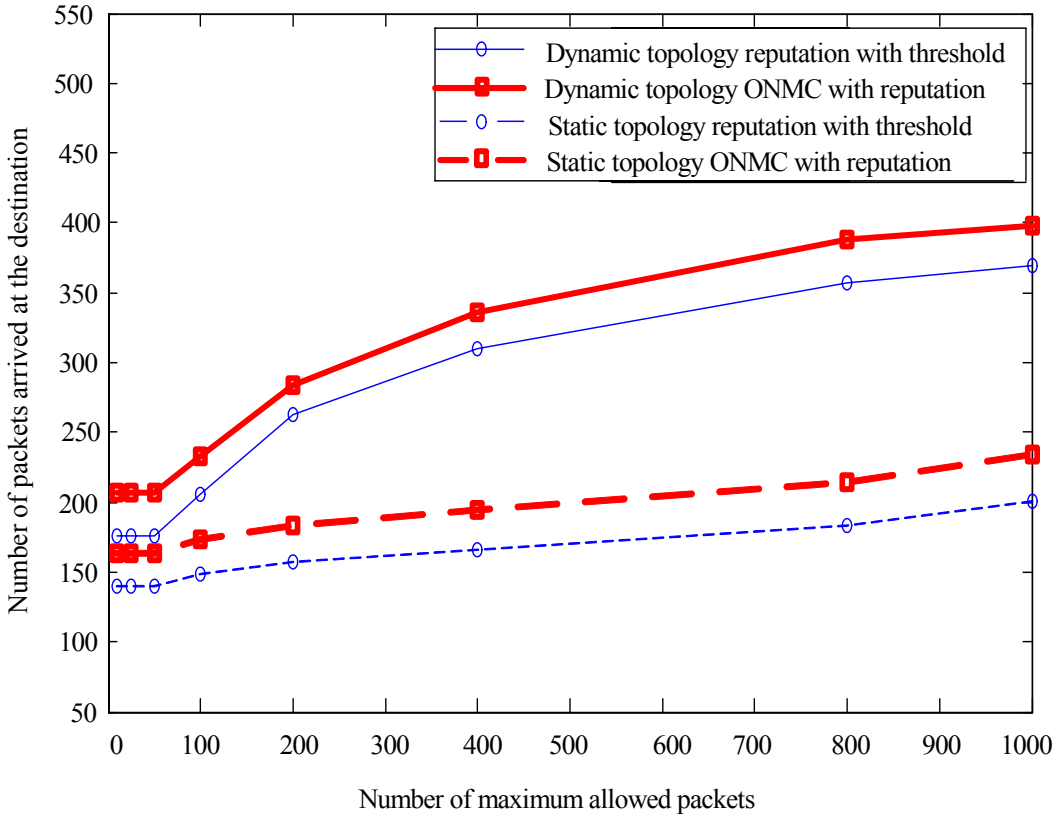

Figure 3.6 Effect of varying the maximum allowed packets.

3.6 Conclusion

The ability to join the network on the fly without a centralized infrastructure exposes MANETs to major security vulnerabilities. Secure network functionalities are therefore necessary to defend attacks from malicious nodes. In this chapter, we study a reputation scheme combined with the ONMC method to learn good rules to identify and therefore select behaving nodes as well as avoid malicious nodes. Numerical studies show that up to 89% of throughput increase can be achieved over the fixed threshold reputation—showing that learning through direct interaction with the network can lead to better reputation decision rules.

In next chapter, we extend the findings in this chapter to a more realistic scenario by generating actual packet traffic and employing a finite buffer queueing model to characterize the reputation values among the MANET nodes.

CHAPTER IV

PERFORMANCE STUDY OF RL—BASED SECURE ROUTING IN MANETS UNDER M/M/1/K MODEL

4.1 Introduction

Chapter 3 presents a reputation scheme combined with reinforcement learning to determine a good rule to select trustworthy nodes. In this chapter, we extend our study in the previous chapter to a more realistic scenario by employing a finite buffer M/M/1/K queueing model to produce packet drops that in turn characterize the reputation values at each MANET node. The purpose of this chapter is to demonstrate the performance of our approach in a more realistic network dynamics.

4.2 Reputation as a Reinforcement Learning Problem

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 (Sutton, 1998). RL can learn to solve a complex task through repeated interaction with its environment in term of states (s_t), actions (a_t) and rewards (r_t). The agent is the learner or decision maker. Everything outside the agent is called the environment. The agent selects an action and the state of environment changes according to those actions. The environment also gives rise to a reward in which the agent tries to maximize over time.

In this chapter, we employ a RL method based on sample episodes, called the on-policy Monte Carlo (ONMC) method (Sutton, 1998). This method is selected because the episodic nature of the route search process in wireless ad hoc networks. An episode starts immediately when a source node initiates a route search to a destination node, and terminates when the target node is found or the maximum number of hop count is reached. As the route search is executed, the intermediate nodes are selected hop-by-hop based on their reputation values. Upon a successful path search, a reward is assigned to every node along all discovered paths. A recent work in Karnkamon Maneenil and Wipawee Usaha (2005) propose a path discovery algorithm in MANETs based on a reputation scheme combined with the ONMC method. Their results showed that such combination can achieve significant increase in throughput over the reputation only scheme for static and dynamic topology cases. However, a Markov model is used to characterize the state of the reputation values' of MANETs nodes. Thus, the contribution in this chapter is to extend their work to a more realistic scenario by employing a finite buffer queueing model to produce packet drops that in turn characterize the reputation values at each MANET node. The goal of this chapter is to find a rule for selecting trustworthy neighboring nodes based on reputation values obtained from finite buffer M/M/1/K queuing model, which optimizes some performance criterion in finite horizon.

4.3 M/M/1/K Queueing Model

In this section, we describe the M/M/1/K queueing model which models a queueing system that has Poisson arrivals with rate λ and exponentially distributed service with rate μ as shown in figure 4.1. The M/M/1/K has a single server with

finite buffer capacity of K. This means that the M/M/1/K method can hold at most a total of K customers including the customer in service. If the system already holds K customers, newly arriving customers will in fact be refused entry to the system and will depart immediately without service. Only those who find the system with strictly less than K customers will be allowed entry.

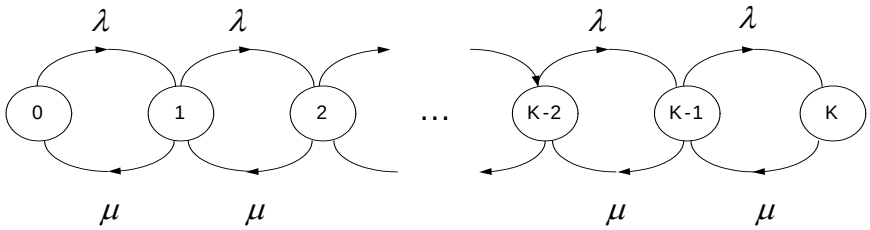

Figure 4.1 M/M/1/K queueing diagram.

In the case of single server queueing systems without state dependent arrival and service rates, the quantity $\frac{\lambda}{\mu}$ is called the traffic intensity and it is usually designated by

$$\rho = \frac{\lambda}{\mu}. \tag{4.1}$$

Let $N(t)$ denote the number of customers in the system, T denote the total customer delay in the system and τ denote the service time. It can be readily shown that the steady state probabilities are (Garcia, 1994)

$$P[N(t) = j] = \frac{(1-\rho)\rho^j}{1-\rho^{K+1}}, \tag{4.2}$$

where $j = 0,1,2,...,K$. The mean number of customers in the system is given by (Garcia, 1994)

$$E[N(t)] = \sum_{j=0}^{K} jP[N(t) = j]$$
$$= \begin{cases} \dfrac{\rho}{1-\rho} - \dfrac{(K+1)\rho^{K+1}}{1-\rho^{K+1}} & for\ \rho \neq 1 \\ \dfrac{K}{2} & for\ \rho = 1. \end{cases} \tag{4.3}$$

Let the proportion of time when the system turns away customers be denoted by

$$P[N(t) = K] = p_K. \tag{4.4}$$

Thus, the system turns away customers at the rate

$$\lambda_b = \lambda p_K. \tag{4.5}$$

The actual arrival rate into the system is given by

$$\lambda_a = \lambda(1 - p_K). \tag{4.6}$$

By applying Eq. (4.3), we obtain the mean total time spent by customers in the system from

$$E[T] = \frac{E[N(t)]}{\lambda_a}$$
$$= \frac{E[N(t)]}{\lambda(1-p_K)}. \tag{4.7}$$

In finite capacity systems, it is necessary to distinguish between the traffic load offered to a system and the actual carried by the system. The offered load or traffic intensity is a measure of the demand made on the system and is defined as

$$\lambda \frac{customers}{second} \times E[\tau] \frac{seconds\, of\ service}{customer}. \tag{4.8}$$

The carried load is the actual demand met by the system as follows

$$\lambda_a \frac{customers}{second} \times E[\tau] \frac{seconds\, of\ service}{customer}. \tag{4.9}$$

A comparison of the carried load versus the offered load ρ for two values of K is shown in figure 4.2. It can be seen that increasing the capacity K results in an increase in carried load since more customers can be accommodated into the system.

A comparison of the mean delay as a function of offered load is shown in figure 4.3. It can be seen that increasing K results in increased delays. Once again, this is because more customers are allowed into the system.

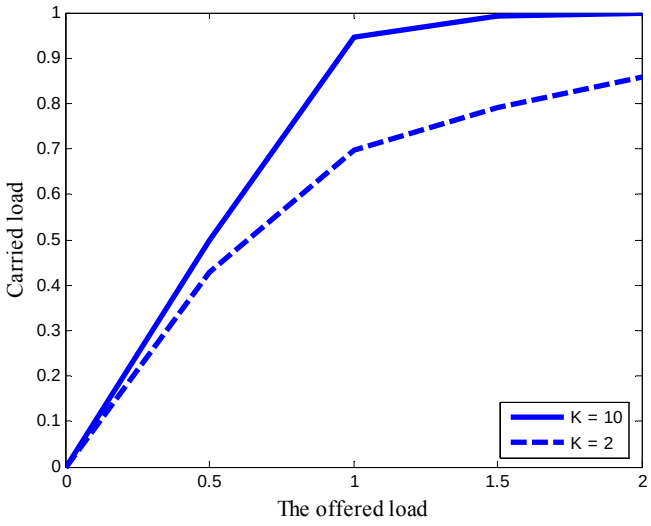

Figure 4.2 Carried load versus offered load for M/M/1/K

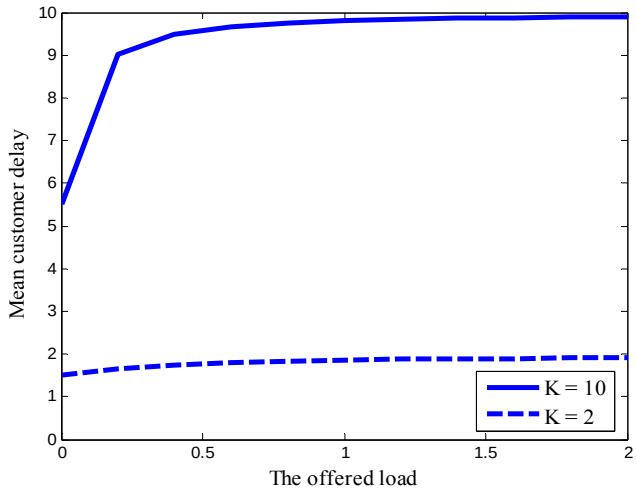

Figure 4.3 Mean customer delay versus offered load in M/M/1/K

4.4 Problem Formulation

In this section, Dewan's reputation scheme based on the ONMC method is applied to MANETs modeled by a M/M/1/K model in order to obtain a trustworthy neighboring node selection policy (Wipawee Usaha and Karnkamon Maneenil, 2006).

Consider a *N*-node ad hoc networks. Each node maintains reputation values of all its neighboring nodes. Suppose that nodes A, B, C and D are neighbors of node S. The reputation value of node is derived from the number of packets forwarded by a node divided by the number of packets which this node receives.

Let R_A, R_B, R_C and R_D be the reputation values of nodes A, B, C and D, respectively where

$$0 \le R_A, R_B, R_C, R_D \le 1. \tag{4.10}$$

Since the reputation values are real numbers, we quantize the state space at node S as

$$X_S = \{x_S : x_S = [q_A, q_B, q_C, q_D]\}, \tag{4.11}$$

where q_A, q_B, q_C and q_D are quantized reputation values of nodes A, B, C and D, respectively.

For example, node S with n neighboring nodes and quantize reputation values into l subintervals, $|X_S| = l^n$ where $|X_S|$ is size of state space.

The actions are the choices made by the agent. Let the action space at node S be given by

$$A_S = \{a_S : a_S = [\delta_A, \delta_B, \delta_C, \delta_D]\}, \tag{4.12}$$

where $\delta_A, \delta_B, \delta_C$ or δ_D is the unity if node S selects node A, B, C or D in the route search and no reward, otherwise. Therefore, node S with n neighboring nodes has

$|A_S| = 2^n$ entries, that is, $[0,0,0,0], \cdots, [1,1,1,1]$. The process is repeated at every selected node until the destination node is found or the maximum number of hop counts is reached.

If the route search is successful, then a reward of +1 is assigned to every node on all successful paths. Otherwise, no reward is assigned to nodes involved in the route search. By using the ONMC method described in Chapter 2, we are able to determine an optimal policy over $\varepsilon - soft$ policies for neighboring node selection based on reputation values.

In this thesis, it is assumed that each node in the MANET operates as a M/M/1/K queueing model. In particular, each node has a single server whose service time is exponentially distributed with mean $\frac{1}{\mu}$. Assume that the packets arrive according to a Poisson process with a mean arrival rate λ. It is also assumed that the nodes have a finite buffer to store arriving packets which have not yet been processed. Under such assumptions, the node follows a M/M/1/K queueing discipline. Nodes with large buffers are assumed to be trustworthy nodes because they are able to receive and forward more packets. On the other hand, malicious nodes have smaller buffers which result in packets being dropped more frequently.

4.5 Experimental Results

We consider a MANET of 23 nodes which includes a number of misbehaving nodes as shown in figure 4.4. Both static and dynamic topology cases are considered. In the dynamic topology case, we generate the topology using a random connectivity model where links between nodes are formed probabilistically. Each node maintains

its own reputation value and announces it to its neighboring nodes. Each node has finite capacity so that packets are dropped if they arrive at a node when the buffer is full. In this chapter, we use buffer size of 6 MB and 4 MB for cooperative and malicious nodes, respectively. The maximum number of allowed packets in the networks (n_m) is 1000.

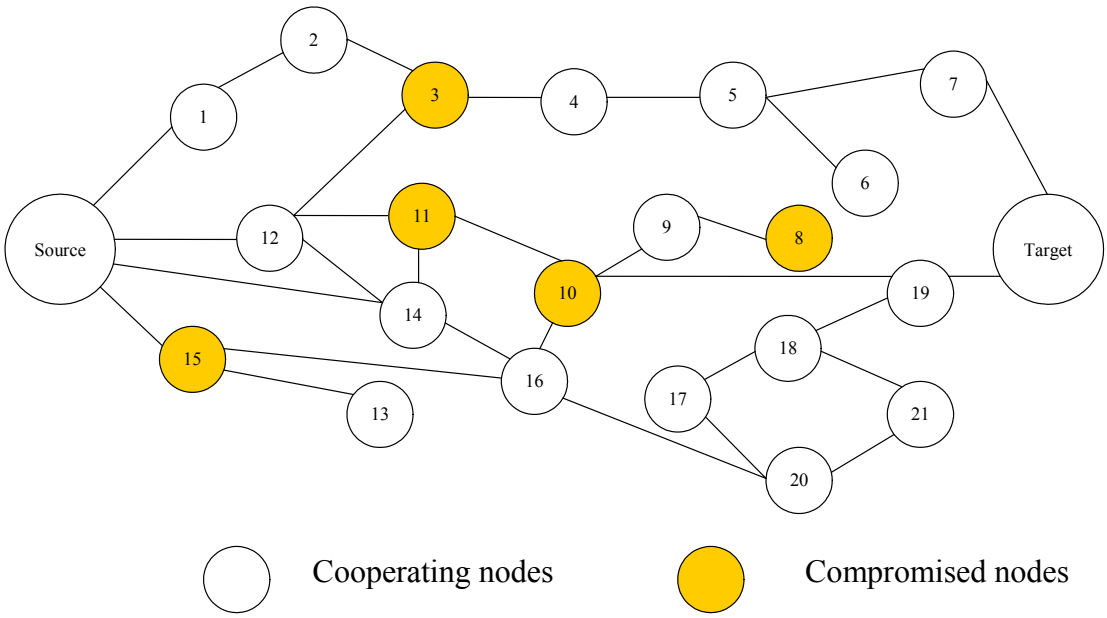

Figure 4.4 Mobile ad hoc networks

Since reputation values are continuous values, the state space is quantized into 5 subintervals, $[0,0.2),[0.2,0.4),[0.4,0.6),[0.6,0.8),[0.8,1.0)$ which are represented by integers 1, 2, 3, 4 and 5, respectively. Each node is assumed to have a maximum connectivity of four nodes. Thus, the state space of some node S has total of $5^4 = 625$ possible states. For example, $x_S = [1,4,2,3]$ refers to the state of node S which has neighbors (A, B, C and D) with reputation values $R_A \in [0,0.2)$, $R_B \in [0.6,0.8)$, $R_C \in [0.2,0.4)$ and $R_D \in [0.4,0.6)$ respectively.

To assess the performance, we use the following metrics, namely, the accumulated reward per episode, the number of packets arrived at the destination and the relative throughput[2]. Furthermore, we compare these metrics among three reputation schemes, namely, a reputation scheme with threshold of 0.5 (Dewan et al., 2004), a reputation scheme combined (Dewan et al., 2004) with the ONMC method and the shortest path scheme which disregards the reputation values.

4.5.1 Accumulated Reward per Episode

Figure 4.5 shows the accumulated reward per episode as the number of malicious nodes in the network increases for the static and dynamic topology cases. Under both topologies, the ONMC scheme outperforms the other two schemes consistently. The reason is because the ONMC scheme can attain good node selection policies for avoiding malicious nodes and is therefore able to find more successful routes when compared to other schemes. Note that when multiple successful paths are found, a reward of +1 is assigned to every node on all successful paths. Therefore, the accumulated reward per episode of the ONMC scheme is the highest among the schemes for both topology cases. On the other hand, that of the reputation scheme with threshold of 0.5 is consistently lower than the ONMC scheme for both topology cases because fix-valued threshold may not be suitable for every ad hoc environment. The accumulated reward per episode of the shortest path scheme is the lowest of all because it does not consider any reputation values in avoiding malicious nodes.

---

[2] The relative throughput = $\frac{throughput}{throughput_{reputation\ only}}$

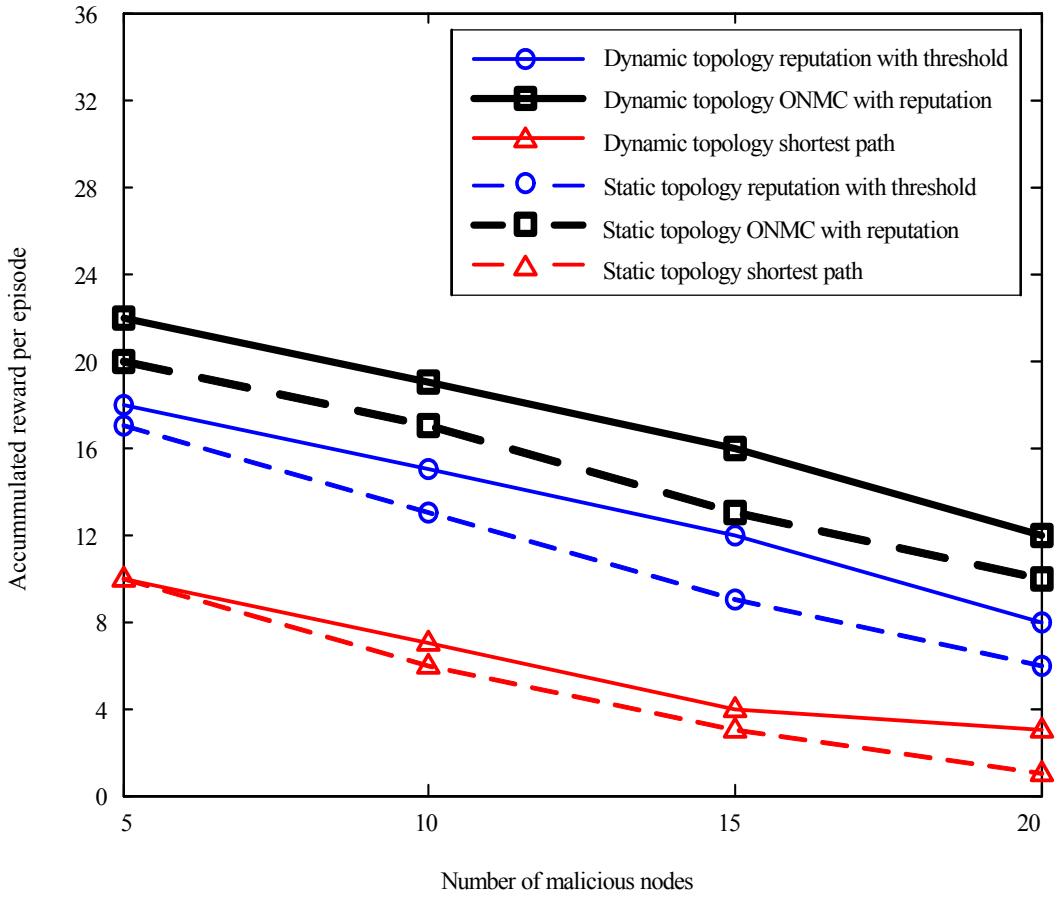

Figure 4.5 Accumulated reward per episode.

4.5.2 Number of Packets Arrived at the Destination

Figure 4.6 shows the number of packets arrived at the destination as the number of malicious nodes in the networks increases for both topology cases. Results show that the reputation scheme combined with the ONMC scheme consistently gives the highest number of packets under both topologies. The reason is because the ONMC method learns its decision through direct interaction with the environment and can eventually learn to select suitable to forward the packets. On the other hand, packets are dropped more in the fixed-threshold reputation and shortest path schemes as they cannot identify malicious nodes as effective as the ONMC scheme.

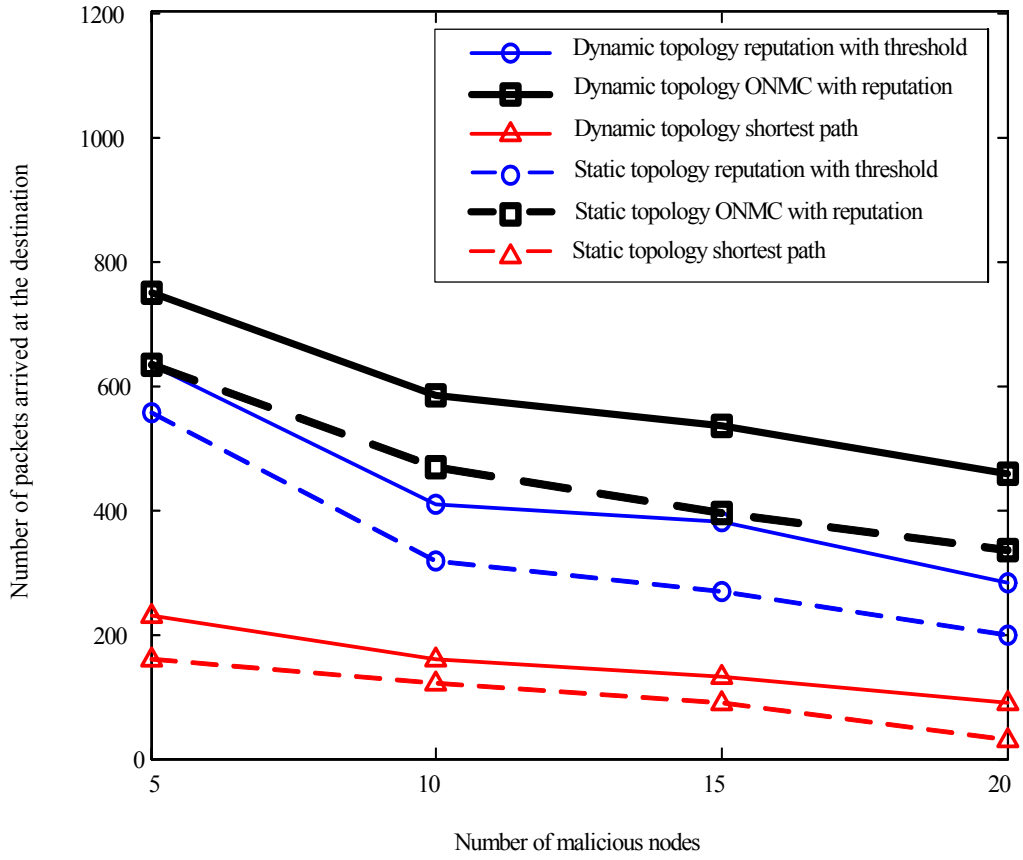

Figure 4.6 The number of packets arrived at the destination with n_m=1000.

4.5.3 Relative Throughput

Figure 4.7 shows the relative throughput is a function of the reputation scheme as the number of malicious nodes in the network. Results show that the reputation with ONMC scheme can achieve up to 71% and 61% increase in throughput over the fixed-threshold reputation scheme for static and dynamic topologies cases, respectively. Note that, the relative throughput of the ONMC scheme is the highest for both topology cases because it can deliver the most number of packets to the destination. On the other hand, the shortest path scheme can only achieve up to 37% and 39% of throughput compared of the fixed-threshold reputation scheme under both topologies case.

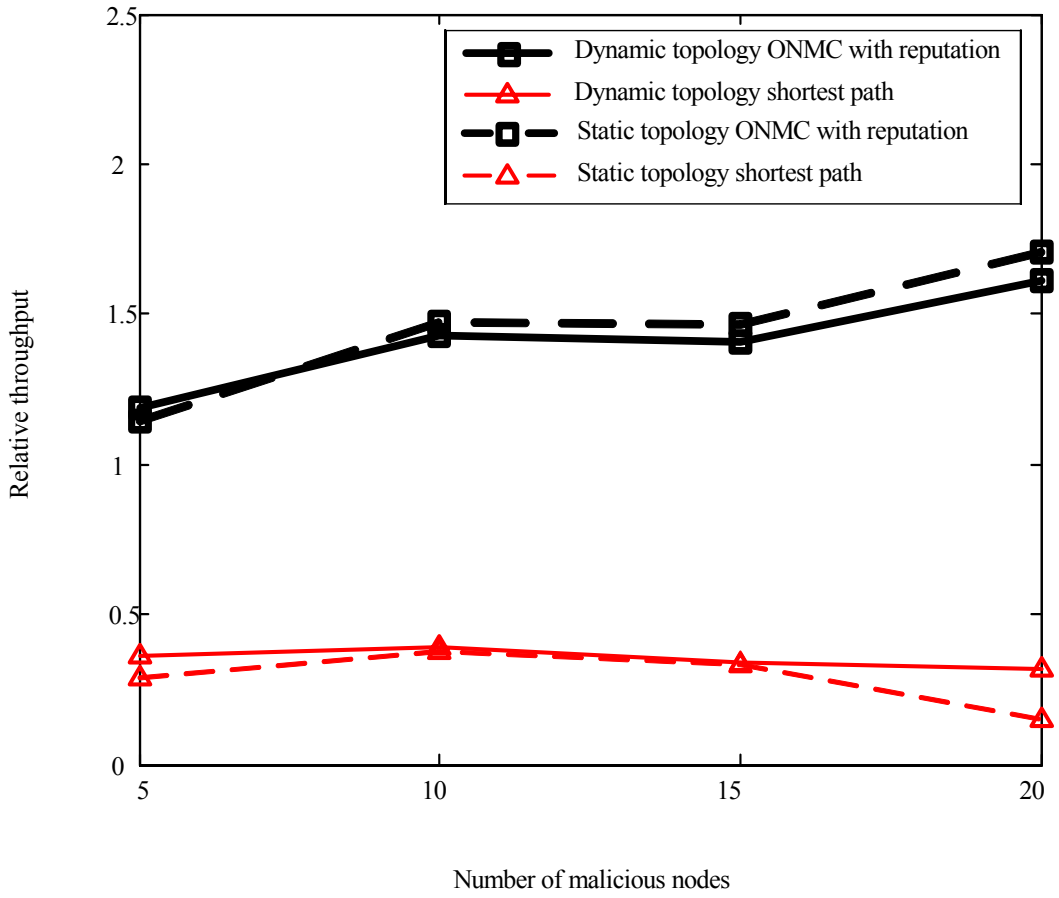

Figure 4.7 The relative throughput.

4.5.4 Effect of Varying the Maximum Allowed Packets

So far, the number of maximum allowed packets is fixed (n_m) at 1000. Figure 4.8 shows the performance in terms of the number of packets arrived at the destination as we reduce the maximum allowed packets. The number of malicious nodes is fixed at 5. Results show that the ONMC scheme still gives a significantly higher number of packets arrivals compared to the other scheme. The results of the shortest path scheme are not shown here as it performed the worst compared to other schemes as evidently, shown in previous figures.

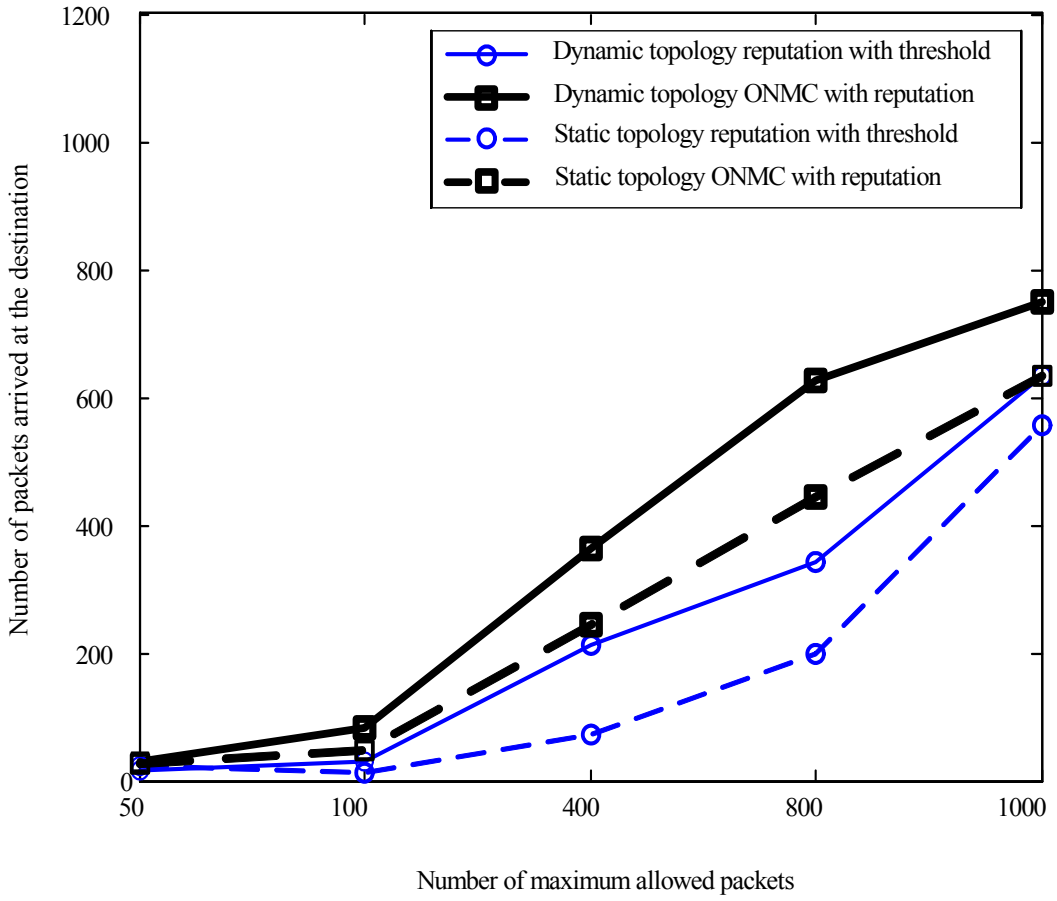

Figure 4.8 Effect of varying the maximum allowed packets.

4.5.5 Effect of Varying the Buffer Size of Good Nodes

In addition, we also study the effect when the buffer size of cooperative nodes is gradually reduced. In this scenario, the buffer size of malicious nodes is fixed at 2 MB. The number of malicious nodes is fixed at 10. The number of maximum allowed packets is fixed at 1000. Figure 4.9 illustrates the performance in terms of the number of packets arrived at the destination as we reduce the buffer size of good nodes. Results show that the ONMC scheme still gives a significantly higher number of packets arrivals compared to the other scheme. The results of the shortest path scheme are not show here as it performed worst of all.

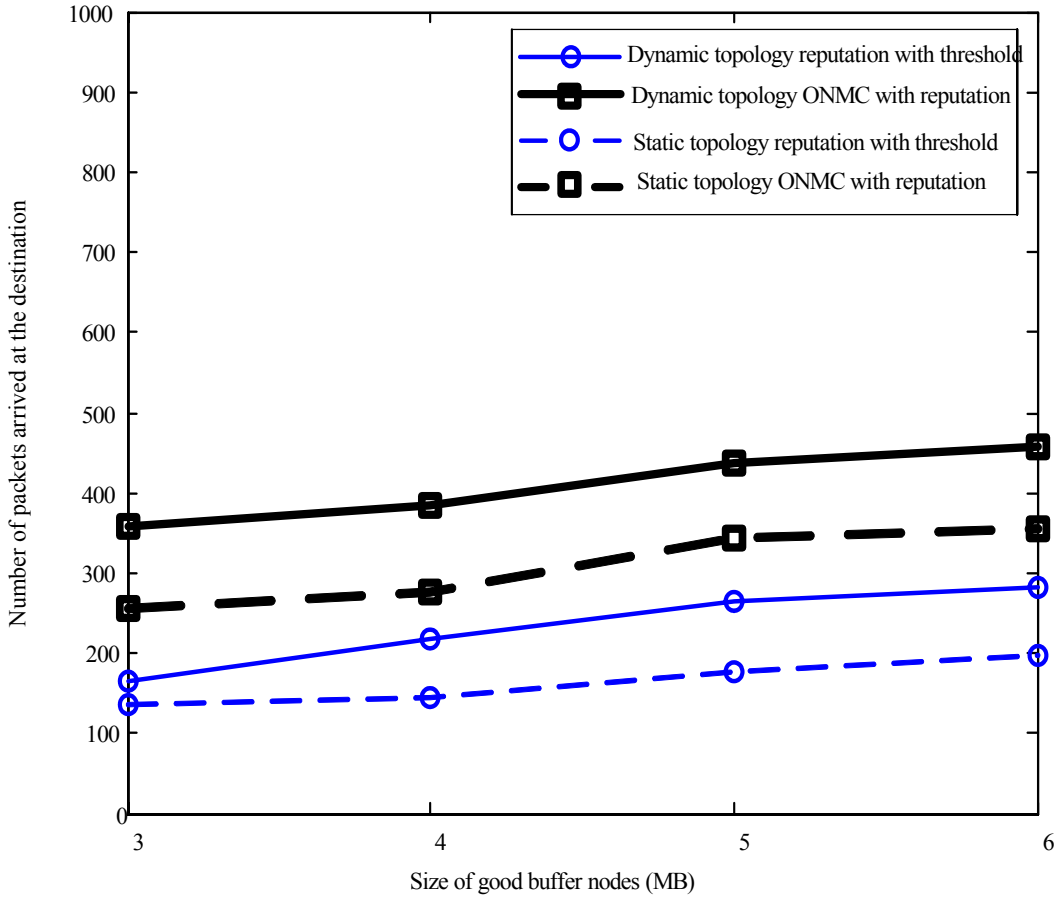

Figure 4.9 Effect of varying the buffer size of good nodes with n_m=1000.

4.6 Conclusion

In this chapter, we study a reputation scheme combine with the ONMC method to learn good rules to identify and therefore select behaving nodes as well as avoiding malicious nodes. This allows us to extend of the findings in (chapter 3) to a more realistic scenario by employing a finite buffer M/M/1/K queueing model to produce actual packet dropping which in turn varies the reputation value at each node in the MANET. Numerical studies show throughput increase of up to 71% over the fixed threshold reputation scheme. The results suggest that reinforcement learning can

lead to better decision rules for neighboring node selection based on reputation values.

CHAPTER V

CONCLUSION AND FUTURE WORK

5.1 Conclusion

In MANETs each host has a limited transmission range. Successful delivery of packets between hosts outside transmission range of each other therefore relies on cooperation of intermediate nodes. The fundamental assumption for such networks is that the nodes will cooperate and not misbehave. However, hosts join the network on the fly creating a dynamic topology network. The lack of a centralized network management leads ad hoc networks vulnerable to attacks by misbehaving nodes. Consequently, packets are dropped or even misdirected therefore resulting in low network throughput. Hence, we proposed an integration of a reinforcement learning technique with an existing reputation scheme, which determines a good rule to distinguish malicious nodes and select cooperative nodes for packet forwarding to the destination node. The contribution in this research can be classified into two parts.

5.1.1 Chapter 3

In this part, we proposed an integration of a reinforcement learning technique with an existing reputation scheme. In particular, the reputation value of each node is directly obtained from a Markov chain model which allows us to test the proposed approach without complication of actual packet traffic generation. Numerical studies show throughput increase of up to 89% over the fixed threshold reputation scheme.

5.1.2 Chapter 4

In this part, we extend the previous contribution to a more realistic scenario by generating actual packet traffic and employing a finite buffer queueing model to characterize the reputation value among the MANET nodes. Numerical studies show throughput increase of up to 71% over the fixed threshold reputation scheme.

5.2 Future Work

Reinforcement learning can be applied to deal with other challenges in MANETs.

5.2.1 Energy Consumption

MANETs are cooperative forms of networks which do not rely on any fixed base station infrastructure. Hence, energy management is a critical issue for deployment of these networks. A routing scheme based on energy efficiency management by reinforcement learning is studied in (Wibhada Naruephiphat and Wipawee Usaha, 2006). Since energy usage is also an important factor which characterizes node behavior, an extension to incorporate our reputation scheme with energy usage is also worthwhile to investigate.

5.2.2 Quality-of-Services Support

MANETs need adequate resources to support more demanding applications and provide QoS guarantees. However, they have limited bandwidth and their dynamic topology poses challenges in finding feasible paths. Yagan and Tham (2005) propose a reinforcement learning method for minimizing QoS violations with

respect to bandwidth, queueing delay and buffer loss in MANETs. An interesting extension would be to incorporate our reputation scheme for avoiding malicious nodes as well as to support QoS traffic.

5.2.3 Mobility

MANETs can be applied in search and rescue or military operations. In such cases, some mobile nodes need to adjust their physical position in order to maintain network connectivity. However, mobile nodes may not optimally form a connection or even connect at all. Reinforcement learning can be used to find an optimal policy for node mobility decision and connection in MANETs. For example, Chang, Ho and Kaelbling (2005) propose a reinforcement learning method to control packet routing decisions and node mobility in MANETs. Our reputation scheme can be extended to stimulate cooperation and connectivity among behaving nodes.

REFERENCES

Basagni, S., Conti, M., Giordano, S. and Stojmenovic, I. (2004). **Mobile Ad Hoc Networking**. IEEE Press.

Bianchi, G. (2000). Performance Analysis of the IEEE 802.11 Distributed Coordination Function. **Journal on Selected areas in communications** 18(3):535-547

Buchegger, S. (2005). Self-Policing Mobile Ad Hoc Networks by Reputation System. **Communications Magazine**. July 2005, pp:101-107.

Buchegger, S., Tissieres, C. and Le Boudec, J-Y. (2004).A Test-Bed for Misbehaviour Detection in Mobile Ad-hoc Networks- How Much Can Watchdogs Really Do?,.**IEEE of the 6th Workshop on Mobile Computing Systems and Applications.** December 2004, pp:102- 111.

Buchegger S. and Le Boudec, J-Y(2003). The effect of rumor spreading in reputation systems for mobile ad hoc networks.**In Proc.WiOpt'03 (Modeling and Optimization in Mobile, Ad hoc and Wireless Networks),**2003.

Cheng, Y-H., Ho, T. and Kaelbling, L.P. (2004). Mobilized ad hoc networks. **International Conference on Autonomic Computing**.

Daigle, J.N. (1992). **Queueing Theory for Telecommunications**. Oxford: IRL Press.

Dewan, P. and Dasgupta, P. (2003). Trusting routers and relays in ad hoc networks.**International Conference on Parallel Processing Workshops**, Proceedings. October 2003, pp:351- 358.

Dewan, P., Dasgupta, P. and Bhattacharya A. (2004). On using reputations in ad hoc networks to counter malicious nodes. **Parallel and Didtributed System**, Proceedings. Tenth International Conference on 7-9 July 2004, pp. 665-672.

Leon-Garcia, A. (1994). **Probability and Random Processes for Electrical Engineering** (2nd ed.). Addison-Wesley.

Kao, E.P.C. (1997). **An Introduction to Stochastic Process**. Duxbury Press.

Kleinrock, L. (1975). **Queueing Systems Volume I: Theory**. John Wiley and sons.

Luo, H., Cheng, J. and Lu, S. (2004). Self-Coordinating Localized Fair Queueing in Wireless Ad Hoc Networks. **Transactions on Mobile computing** 3(1): 86-98.

Maneenil, K. and Usaha, W. (2005). Preventing malicious nodes in ad hoc networks Using reinforcement learning. **International Symposium on Wireless Communication Systems**, September 2005, pp: 289-292.

Marti, S. and Garcia-Molina, H.(2003). Identity crisis: anonymity vs reputation in P2P systems.**Tthe 3rd International Conference on Peer-to-Peer Computing**, September 2003, pp:134- 141.

Rebahi, Y., Mujica-V, V.E., and Sisalem, D. (2005). A Reputation-Based Trust Mechanism for Ad hoc Networks. IEEE Symposium on Computers and Communications.

Sutton, R.S., and Barto, A.G.(1998). **Reinforcement Learning: An introduction.** Massachusetts: The MIT Press.

Tijims H.C. (2003). **A First Course in Stochastic Models**. Wiley.

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, pp:220 – 224.

Usaha, W. and Maneenil, K. (2006). Identifying Malicious Nodes in Mobile Ad Hoc Networks using a Reputation Scheme based on Reinforcement Learning. **TENCON** 2006.

Vassilaras, S., Vogiatzis, D. and Yovanof G.S. (2005). Misbehavior Detection in Clustered Ad-hoc Networks with Central Control. **International Conference on Information Technology: Coding and Computing**.

Wang, B., Soltani, S. and Shapiro, J. (2005). Local Detection of Selfish Routing Behavior in Ad Hoc Networks. **International Symposium on Parallel Architectures, Algorithms and Networks**, Proceedings.

Yagan, D. and Tham, C-K. (2005). Adaptive Qos Provisioning in Wireless Ad Hoc Networks: A Semi-MDP Approach. **WCNC 2005**.

Yan, H. and Lowenthal, D. (2005). Toward Cooperation Fairness in Mobile Ad hoc Networks. **Wireless Communications and Networking Conference**.

Lui, Y. and Yang, Y.R. (2003). Reputation Propagation and Agreement in Mobile A Hoc Networks [On-line]. Available: http://ieeexplore.ieee.org

APPENDIX

LIST OF PUBLICATIONS

LIST OF PUBLICATIONS

Maneenil, K. and Usaha, W. (2005). Preventing malicious nodes in ad hoc networks Using reinforcement learning. **International Symposium on Wireless Communication Systems**. pp: 289-292.

Maneenil, K. and Usaha, W. (2005). A Reinforcement Learning Method for Avoiding Malicious Mobile Nodes in Wireless Ad Hoc Networks. **Electrical Engineering Conference**. pp: 71-74.

Usaha, W., Maneenil, K. (2006). Identifying Malicious Nodes in Mobile Ad Hoc Networks using a Reputation Scheme based on Reinforcement Learning. **TENCON 2006**.

BIOGRAPHY

Miss Karnkamon Maneenil was born on September 19, 1981 in Muang District, Srisaket Province. In 2000, she began studying for her Bachelors degree at School of Telecommunication Engineering, Institute of Engineering at Suranaree University of Technology, Nakhon Ratchasima Province. After graduating, she continued to study for a Masters degree at the School of Telecommunication Engineering, Institute of Engineering, Suranaree University of Technology.