# INTEGRATED SECURE MULTIPATH MOBILE AD HOC NETWORK

**Supachote Lertvorratham**

**A Dissertation Submitted in Partial**

**Fulfillment of the Requirements for the Degree of**

**Doctor of Philosophy (Computer Science)**

**School of Applied Statistics**

**National Institute of Development Administration**

**2010**

**INTEGRATED SECURE MULTIPATH MOBILE AD HOC NETWORK**

**Supachote Lertvorratham**

**School of Applied Statistics**

---

Associate Professor ...P. Hiranvanichakorn..........Advisor

(Pipat Hiranvanichakorn, D.E.)

The Examining Committee Approved This Dissertation Submitted in Partial Fulfillment of the Requirements for the Degree of Doctor of Philosophy (Computer Science).

Assistant Professor.........................................Committee Chairperson

(Ohm Sornil, Ph.D.)

Associate Professor ....P. Hiranvanichakorn.............. Committee

(Pipat Hiranvanichakorn, D.E.)

Assistant Professor ....Pramote........................................Committee

(Pramote Kuacharoen, Ph.D.)

Assistant Professor..................................................Committee

(Chonawat Srisa-An, Ph.D.)

....L. Bosuwan.........................................Dean

(Lersan Bosuwan, Ph.D.)

October 2010

# ABSTRACT

| | |
|---|---|
| **Title of Dissertation** | Integrated Secure Multipath Mobile Ad Hoc Network |
| **Author** | Supachote Lertvorratham |
| **Degree** | Doctor of Philosophy (Computer Science) |
| **Year** | 2010 |

The dramatic growth of wireless communication technology today makes wireless equipment affordable and widely used. In a situation where a temporary wireless network needs to be established but fixed-base station is not available, a mobile ad hoc network can be implemented. For implementing a mobile ad hoc network, I realize that there are some challenges. Firstly, most of mobile ad hoc network routings rely on establishment of minimum hops route. I argue in this case that there can be some other metrics, which can be used to establish a better performance and more robust route. Secondly, in many circumstances, wireless networks require security features. My challenge is how to obtain necessary security features with ad hoc network routing.

This dissertation is contributed for designing and evaluating mobile ad hoc network routing protocols that are embedded with innovative path measurement techniques and security functions. I proposed two routing protocols in this paper. The first protocol is an on-demand multipath ad hoc network routing protocol called “Predicted Multipath Routing Protocol (PMP)”. The protocol concerns with discovery of a set of routes that have high efficiency and are robust. To measure robustness and efficiency of a route, I proposed two measurements. The first measurement is the Degree of Availability (DA) that anticipates the future survival of a path. It basically predicts the future signal strength of each pair of nodes using regression analysis. The second measurement that measures path efficiency is called “Estimated Path Throughput Value (ETV)”. The ETV relies on the packet loss ratio of link between each pair of nodes.

The second protocol is an improvement of the PMP called “Secured Predictive Multipath Routing Protocol (SPMP)”. The design of this protocol is focused on solid security processes to discover secure routes. The protocol is incorporated with three main processes. The first process is node authentication process. The protocol allows a node to authenticate itself either as a group member or as an individual trustful node. The second process is secure route discovery process. This process maintains integrity of routing information during route establishment from a source to a destination. The third process is secure data forwarding process that keeps secrecy of end-to-end data exchange and deliverable path. The SPMP also provides a predefine target receiver technique that enable a node to identify only trusted receivers of route request packets and to limit the maximum number of route request packets not to be exceed the number of node in a network.

# ACKNOWLEDGEMENTS

I would like to take this opportunity to express my gratitude to all the people at The School of Applied Statistics for a great environment during the entire time I have worked on this thesis. Most notably, heartfelt thanks to my advisor, Dr. Pipat Hiranvanichakorn, for his ideas, comments and great support. Also, thanks to all of my teachers for their support and encouragement.

Supachote Lertvorratham
October 2010

# TABLE OF CONTENTS

# LIST OF TABLES

**Tables** **Page**

# LIST OF FIGURES

**Figures** **Page**

# CHAPTER 1

# INTRODUCTION

## 1.1 Background

Wireless technology is not a new technology and there has been a dramatic growth of wireless communications since the last decade. Furthermore, wireless technologies today are inexpensive and become a rapid way for internet connection in locations where there are poor infrastructures, wireless networks have been widely implemented with not only limited local areas but also metropolitan in developed countries. Even though, most of implementation of wireless networks requires base stations, the fixed-location wireless networks are not applicable in some environments such as communications among vehicles in a battle field, ship fleet in the sea or equipments in a rescue area. As a result, mobile ad hoc networks consequently emerge to recover a situation that a central communication device is not capable to handle communications among nodes. A mobile ad hoc network (MANET) consists of a number of terminals or nodes that communicate among each others without any centralized device. In a MANET, every node in the network performs some additional functions such as finding routes and acts as a middle man to forward data to the destination.

Most of current wireless communications depend on radio waves with various frequencies. In the Earth's atmosphere, propagations of radio signal are limited in distance. In a wireless network with a base station, the base station occupies a limited service area and acts like a hub to manage communications among mobile stations in the service area. Each base station has communication links with the others via either wired network or satellite communication. In the case, each base station handles routing and a mobile station can extend the connectivity service in a location that in different from the home location where the service was registered by roaming. Unlike fixed-location wireless network, there is no base station in an ac hoc network. In a mobile ad hoc network, each communication terminal performs additional routing functions and acts as a router. Therefore, the connectivity area depends on location of mobile stations, mobility of nodes, capability of communication equipments, and routing protocol.

In fact, routing method is a major difference between fixed-location networks and ad hoc networks. Since there is mobility of routers in a mobile ad hoc network, a route between two mobile stations is always unstable. This make routing method of a mobile ad hoc network more complicate than that of a fixed-location network. As a result, there were many studies of routing protocols for ad hoc network since the protocols that are working with wired networks do not applicable with the ad hoc network unless these protocols are not modified. The studies of ad hoc routing protocol mainly concern with developing a standard that how nodes decide which way to route data between mobile devices in a network (Lin, Midkiff and Park, 2003:1162). By nature of a mobile ad hoc network, the topology of network is dynamic and nodes in the network have to discover the topology of their network. The basic idea of how the topology is detected is that a node may announce its presence and should listen for announcements broadcasted by its neighbours.

There are many types of ad hoc network routing protocols. Among these, proactive and reactive routing protocols are widely studied because they do not need any special equipment. For proactive routing protocols, every node maintains fresh lists of destinations and routes by periodically distributing routing tables throughout the network. The main disadvantages of proactive routing are some problems about route maintenance. Typically, route maintenance in proactive routing protocol is costly and consume high amount of data. In many protocols, the proactive routing has slow reaction on route failures and restructuring. Examples of proactive routing protocols are Clusterhead Gateway Switch Routing (CGSR) (Ching-Chuan, Hsiao-Kuang, Winston and Mario, 1997:197), Destination-Sequenced Distance Vector (DSDV) (Perkins and Bhagwat, 1994:234) and Topology Dissemination Based on Reverse-Path Forwarding (TBRPF) (Bhargav, Richard and Fred, 2004:1). For reactive or on-demand routing protocol, a node finds a route whenever it needs to communicate with a destination by flooding the network with route request packet. The concepts of reactive routing bring about low route maintenance cost and are appropriate for dynamic topology network. However, reactive routing consumes high latency time in route finding and there may be excessive flooding of route discovery data that leads to network congestion. The instances of reactive routing protocol are Dynamic Source Routing (DSR) (Johnson and Maltz, 1996:153) and Ad-hoc On-demand Distance Vector (AODV) (Perkins, 2003:1).

As I concerned about on-demand communication among nodes in network, I focused my study on reactive routing protocols. During the early period of reactive routing protocol studies, many researchers emphasized on improvement the efficiency of single path routing which has been facing problems in maintaining high performance paths and assuring the availability of communicating route. Consequently, introduction of multipath routing protocols emerged to recover limitations of single path routing. The basic idea of multipath routing is that a mobile node discovers more than one route to the destination and keeps some routes as spare routes or sends data via discovered routes simultaneously to increase network performance.

Besides the improvement of network performance, securities on mobile ad hoc networks were focused by a group of researchers. In fact, securities are essential for ad hoc network because of several reasons. One reason is the widespread of using wireless communication equipments. The advancement of today technologies causes wireless communication equipments to be smaller and more efficient. When the more wireless communications are utilized, the more protection of personal data is required. Another reason is the needs of ad hoc network users. In many circumstance, ad hoc networks are used to exchange secret data among specific users. For example, when troops of soldier are communicated in a hostile area, the communicators must be ensured that the secret data are not intercepted by enemies. The prior studies regarding wireless network securities consisted of data encryption, node authentication and secure route finding. Regardless some studies of security of ad hoc network that relied on existing wired network security, many researchers agree that design of securities associated with ad hoc network is more difficult than that of wired network. For instance, in an ad hoc network, every node acts as a host and a route simultaneously, the network is obviously more vulnerable from men-in-the-middle attacks than a wired network which has dedicated routers to perform routing functions. There have been ad hoc network security researches in various schemes but my consideration for the securities on mobile ad hoc networks is that they are extremely needed in some particular environments, for example, communications in warfare that require secure and robust route to save the valuable information from unauthorized access or enemies' attacks.

## 1.2 Motivation

There are complications in mobile ad hoc networks because the topology of a network is frequently changed, the distance in wireless communication is limited and the network is vulnerable or easy to be attacked. Therefore, previous researches regarding mobile ad hoc networks were focused on two main problems. First is how a mobile ad hoc network efficiently discovers its topology and establishes a high performance route during a certain period. Second is how to make communication in the network safe when data securities are required.

For topology discovering and high performance route establishment, the design of an efficient ad hoc network routing protocol becomes a challenge. DSR and AODV are instances of famous routing protocols during the early studying period of mobile ad hoc network. Though they are proven in term of route discovery efficiency (Hu and Johnson, 2001:1), these traditional single path routing protocols still have some problems in maintaining route availability during data transmission that causes deficiency of the network. Thus, multipath routing protocols (Nasipuri and Das, 1999:64; Lee and Gerla, 2000:1311; Lee and Gerla, 2001:3201) were introduced to improve efficiencies of single path routing protocols. For multipath routing, a source mobile node attempts to find a number of routes to the destination. Then, selected routes are utilized into two ways. First, the source node may choose the best route to be the primary route and determines the others as alternative routes, which are eligible when the active route becomes invalid. Second, the node can split data into pieces and sends them over all valid routes simultaneously to increase communication efficiency (Mao, Wang, Lin and Panwar, 2002:1; Mao, Wang, Lin and Panwar, 2003:1721; Wei and Zakhor, 2004:496).

Typically, routes establishment of multipath routing protocols cope with finding a set of disjoint paths with minimum hop. Unfortunately, minimum hop does not necessarily produce the best performance route (De Couto, Aguayo, Bicket and Morris, 2003:134). For wireless communications, the radio signal becomes weaker when the distance between two nodes is lengthened. In many cases, a route incorporated with long distance nodes may faces more signal loss, congestion and higher risk of path failure. These lead to poor performance network. As a result, a number of studies of quality of services for ad hoc network were proposed in order to guarantee network performance (Nikaeing and Bonnet, 2002:1; Akkaya and Younis, 2003:710; Sun and Hughes, 2003:408; Chen and Heinzelman, 2004:1715; Tuduce and Gross, 2004:259; Chen and Heinzelman, 2005:561; Jain, Sharma and Banerjee, 2006:288).

For network security, there were researches in security on mobile ad hoc network that focus on different problem domains and relied on different approaches. In fact, in a mobile ad hoc network, security can be implemented in every layer. On the physical and MAC layer, a secure data communication can be executed by using signal scramble. On the network layer, data encryption and secure routing protocol are employed to ensure that no unauthorized node be a part of communicating route. On application layer, integrated securities like data encryption, key exchange and digital signature are applied, for instances PGP or SSL. In my view point, for mobile ad hoc network, implementing security on network layer is more appropriate than other layers with three reasons. First, implementation of network layer security is free from sticking with hardware manufacturer that is very limited and inflexible. Second, security on network layer is not too complicate and consumes less amount of overhead. Third, since the main function of network layer is to find communication routes, only security on network layer is able to obtain secure routes.

Another interesting problem when researchers consider security on mobile ad hoc networks is that how a mobile station authenticates itself with others. According to our researches, the certification from a central trusted entity seems to be most acceptable. An interesting issue is that how the central trusted certification is utilized for a mobile ad hoc network routing protocol.

According to above reasons, I realized that there are challenges in development of a secure mobile ad hoc network routing protocol. The first challenge is how to improve quality of routing in ad hoc network in term of performance and robustness. The next challenge is how to obtain necessary security features with ad hoc network routing process without the requirement of central certificate authority. Therefore, my research objectives aimed to improve a multipath ad hoc network routing that was embedded with necessary security characteristics. The research was focused on two areas. First is to improve multipath ad hoc network performance using performance measurements and predictive models. Second is to securitize the multipath ad hoc network routing with current data security mechanisms and self-authentication strategies.

## 1.3 Research Overview

This dissertation contributes for innovative path measurement techniques and security functions with a mobile ad hoc network routing protocol. The objectives of this research are as follows:

1. To introduce techniques of path measurements that use statistical model to allow mobile ad hoc networks predict future performance of discovered path.
2. To develop a multipath ad hoc network routing protocol that incorporates with control data collection processes and proposed path measurement techniques.
3. To propose security models and an authentication method that do not require central trusted certificate agent.
4. To integrate the proposed security models and the authentication method with a multipath ad hoc network routing protocol and develop a secure multipath ad hoc network routing protocol that focuses on solid security in discovering secure routes.

My dissertation was separated into two parts. The first part is contributed for designing and evaluation an effective multipath routing protocol that covers the explanation in chapter 2 and 3. In the chapter 2, I describe the details of related researches of reactive routing protocols that I concentrate on two famous single path routing protocols and a multipath routing protocols. Consequently, I present a new multipath routing protocol called the Predictive Multipath Routing Protocol (PMP) in chapter 3. In this chapter, I introduced a path measurement technique that reports the availability and efficiency of a path. The measurement technique involves with forecasting the future availability of a path and end-to-end path performance using regression analysis. There are two path measurements in my model. The first measurement is the Degree of Availability (DA). The DA bases on potential signal strength and is used to scale availability of a path. The second measurement is a measurement of end-to-end path efficiency called the Estimated Path Throughput Value (ETV) that relies on the packet loss ratio between each pair of communicating node. Next to the measurements, I present techniques that the protocol uses for choosing the best ETV path and limits the maximum number of route request packet to the number of nodes in a network by allowing a node forwards a route request packet only once. The major part of chapter 3 dedicated for details of route establishment and route maintenance functions of the protocol. In this section, I depict design of control packets and data packet as well as route discovery and route maintenance process. The rest of chapter 3 is simulation where I show my protocol test results and analysis of the results.

The second part of my dissertation contains details of designing and evaluating a secure multipath routing protocol. This part consists of chapter 4, 5 and 6. Chapter 4 describes the fundamental of encryption algorithms that I use for my secured multipath routing protocol. There are two classes for the encryption algorithms. The first class is called Symmetric-Key Encryption Algorithms which associates with method of encryption and decryption using the same key. The second class is named Public-Key Encryption Algorithms. The algorithms concern with techniques to generate a dual keys that data can be encrypted with one key and decrypted with another key. Next in chapter 4, I include message digests or hash functions that are employed to support authentications. Another section of this chapter is digital signature. In this section, I conclude message signing and verification method. The last section of chapter 4 presents my related studies of secure multipath routing protocol. I concentrate on three protocols. Those are ARAN (Sanzgiri, Dahilly, Leviney, Shieldsz and Belding-Royer, 2002:78), SRP (Papadimitratos and Hass, 2002:27) and SEAD (Hu, Johnson and Perrig, 2002:3). In chapter 5 and 6, I narrate information about my secured multipath routing protocol called The Secure Predictive Multipath Routing Protocol (SPMP). There are several particular functions that are used for SPMP. The description of group authentication and related functions will appear in the early section of chapter 5. In the middle of chapter 5, I explain designs of routing processes of SPMP and security analysis. In the chapter 6, I propose the individual authentication of SPMP which is the enhancement of SPMP with group authentication. I also show simulation results in the last sections of the chapter 5 and 6. Finally I will conclude my contribution in the chapter 7.

# CHAPTER 2

# RELATED RESEARCHES OF

# REACTIVE ROUTING PROTOCOLS

## 2.1 Introduction to Mobile Ad Hoc Network Routing Protocol

A mobile ad hoc network (MANET) is a network that incorporates with a collection of arbitrary wireless mobile nodes. Unlike a typical wireless network, a MANET allows a number of mobile nodes to freely form a network without specific pre-configuration and each mobile node acts a host and a router simultaneously. Since a MANET does not require any base station or access point, the network is extremely useful in a circumstance that there is no available infrastructure. Therefore, MANET is beneficial for circumstances that require rapid-forming and temporary networks, for example, military missions in a battlefield, emergency rescue operations in a disaster area, engineering tasks in a large construction site, law enforcement activities in particular locations and so on. In the case, establishment of a MANET is a challenging issue because of following reasons:

1. Radio signal can be propagated with limited distance.
2. Locomotion of mobile nodes causes their positions frequently change.
3. The appearance of a new mobile entry can happen at all times.
4. The breakage for communication equipments of a mobile node makes the node becomes invalid in the network.

These limitations induce the change in topology of a network as well as communication path between two mobile nodes. As a matter of fact, a MANET requires an efficient routing protocol that has quick response and obtains good routes. As a result, development of mobile ad hoc network routing protocol becomes one of a challenge among researchers and brings about a number of proposals of MANET routing protocols during last decade.

MANET routing protocols can be classified into three categories: proactive routing protocol, reactive routing protocol and hybrid routing protocol. The proactive routing protocol relies on a basic idea that nodes must continuously search for routing information within a network. An advantage is that a node immediately knows a route when the node needs to communicate with other nodes. However, there are some drawbacks exist. First, when network topology is changed, the proactive routing protocol takes a long time for restructuring the network. Second, the protocol needs to exchange a large amount of control packet for route maintenance. Instances of proactive routing protocols are CGSR (Ching-Chuan, Hsiao-Kuang, Winston and Mario, 1997:197), DSDV (Perkins and Bhagwat, 1994:234), and TBRPF (Bhargav, Richard and Fred, 2004:1). The reactive routing protocol, on the other hand, depends on on-demand driven concept. According to the protocol, a node only executes route finding activities whenever it desires to communicate with a destination. The benefits of this concept are that the network consumes low route maintenance cost and the dynamic topology network is easy to manage. Contrary to the proactive routing protocol, the reactive routing protocol consumes high latency time in route discovery. For each route discovery activity, network must employ a log of control packet that leads to network clogging. There are many proposed reactive routing protocols include Ad-hoc On-demand Distance Vector (AODV) (Perkins, 2003:1) and Dynamic Source Routing (DSR) (Johnson and Maltz, 1996:153). The hybrid routing protocol incorporates advantages of both proactive and reactive routing protocols. Conceptually, the hybrid routing protocol divides network into groups. Each group realizes its member nodes as prospects in which routes among the prospects are proactively establishes. When communication across groups is required, the protocol carries out on-demand route finding strategies to establish routes among groups. Therefore, the performance of network that use the hybrid routing protocol is depended on amount of nodes proactively activated and type of traffic volume. An example of the hybrid routing protocol is Zone Routing Protocol (ZRP) (Zygmunt and Marc, 1997) and Broadcast Resolution Protocol BRP (Zygmunt, Marc and Prince, 2002).

There are some studies concerned with architectures of QoS model for supporting MANET (Nikaeing and Bonnet, 2002:1; Chen and Heinzelman, 2004:1715). The main purpose of these studies is to indicate which QoS parameters should be used in each network layer. These studies do not explain the application of using QoS parameters with ad hoc network routing protocol. The other studies associated with design and implement routing scheme for wireless ad hoc networks to support QoS, for example, Akkaya and Younis proposed an energy-aware QoS routing protocol for wireless sensor networks (Akkaya and Younis, 2003:710). The protocol concerns with finding an optimal path based on measuring energy consumption, error rate and end-to-end delay so that bandwidth of a path can be guaranteed as well as providing the most-efficient-consumption path in wireless sensor networks. In a wireless sensor network, nodes are grouped in clusters. Each cluster has a gateway node which acts as a router for communicating with the other cluster. For energy-aware QoS routing, the topology of network is realized as a tree where the energy resource metrics are reported by one-hop neighbor. In my opinion, the protocol that requires dedicated gateway nodes may not work well with high mobility network. The mobility of nodes makes a possibility that a gateway node may be out of communication range of the other gateway nodes and the entire cluster will not be able to communicate with the others in the network. Sun and Hughes proposed Adaptive Multi-path Routing (AMPR) (Sun and Hughes, 2003: 408) to work with a number of QoS parameters including end-to-end delay, packet loss, bandwidth and signal-to-noise. This routing scheme relies on aggregated values of QoS parameters as well as decision making model that enable nodes pickup the good routes. The main concept AMPR is based on source routing scheme that is similar to DSR. The difference is that AMPR allows route information attached with metric values to be sent along the finding path and the source node is responsible for choosing the best path. Though AMPR establishes high quality multiple paths, I argue the concept of path discovery of AMPR that the established paths of AMPR are not disjointed so that the protocol may not gain advantage from multi-path discovery in high mobility networks. Chen and Heinzelman introduced a protocol to discover a quality route using estimated residual bandwidth. According to their proposal, each node in a network has to calculate bandwidth consumption and disseminate the bandwidth information to its neighbors. Consequently, the residual bandwidth is estimated during route discovery period. The model proposed by Chen and Heinzelman is applicable for hop-to-hop single path routing like AODV. Nevertheless, usage of this protocol may face a difficulty that the protocol requires knowledge of bandwidth requirement from working application. Thus, the application may be equipped with a feature of estimation of bandwidth consumption.

## 2.2 Dynamic Source Routing (DSR)

The Dynamic Source Routing protocol (DSR) (Johnson and Maltz, 1996:153) is a single path routing protocol for mobile ad hoc networks which have up to about two hundred nodes. The protocol does not need any network infrastructures but nodes in a network must cooperate in discovering topology of the network. The cooperation of network nodes involves with forwarding packets among each other to enable communications over multiple hops when nodes are not in the wireless transmission range of one another.

The highlight of DSR is that the protocol automatically determines and maintains all routing when a node join or leave the network with low overhead and reacts very quickly to changes in the network. The principle of DSR consists of following rules;

1. Every node in a network has a route cache to maintain routes from it to destinations. For example, if communication between node *A* and node *D* requires intermediate hops *B* and *C* as illustrated in the figure below;

*A* -> *B* -> *C* -> *D*

Then, node *A* maintains route information to *D* by *A* -> *B* -> *C* -> *D* and *B* maintains route information to *D* by *B* -> *C* -> *D*

2. When a source needs to submit data to a destination, if source found a route to the destination in its route table, the source always attaches source route information with data packets. This source route information is used as topology map from source to destination. Intermediate nodes use the source route information to decide if it should participate in forwarding the packet. Source route information is truncated by the intermediate node before forwarding the packet. For example of a

route from *A* -> *B* -> *C* -> *D*, source route information launched by *A* is *B* -> *C* -> *D*. When *B* receives a packet from *A*, the intermediate node *B* forward source route information to *C* by *C* -> *D*.

3. The DSR protocol is composed of two main mechanisms that work together to allow the discovery and maintenance of source routes. They are;

3.1 Route Discovery - the mechanism which is only activated by a source node to obtain a route to a destination when the source attempts to send a packet to the destination and does not already know a route to the destination.

3.2 Route Maintenance - the mechanism by which a source node is able to detect, while using a source route to the destination, if the network topology has changed such that the route to destination becomes invalid because a link along the route no longer works. When Route Maintenance indicates a source route is broken, the source can invoke Route Discovery again to find a new route for subsequent packets to the destination. Route Maintenance for this route is used only when the source is actually sending packets to the destination.

The main concepts of DSR are dedicated to Route Discovery and Route Maintenance. For Route Discovery, when a source originates a new packet addressed to a destination node, the source broadcasts a control packet call Route Request (RREQ). The RREQ is handled and forwarded by the sequence of hops that the packet is to follow on its way to the destination. To handle an incoming RREQ, an intermediate node finds a route in its route cache, if a route is found, the node returns the route information back to the source with a Route Reply packet (RREP). If no route is found in the cache, the intermediate node attaches the RREQ with its own information and forwards the RREQ to its neighbors.

For example, suppose a node *A* is attempting to discover a route to node *D*. The RREQ initiated by *A* would proceed as follows:

*A* -> (RREQ-A) *B* -> (RREQ-A,B) *C* ->(RREQ-A,B,C) *D*

In the case of Route Maintenance, during data transmission using a source route, each node transmitting the packet is responsible for confirming that data can flow over the link from that node to the next hop. For example, in the situation shown below, node *A* has originated a packet for node *D* using a source route through intermediate nodes *B* and *C*:

*A* -> *B* -> *C* -> *D*

In this case, node *A* is responsible for the link from *A* to *B*, node *B* is responsible for the link from *B* to *C* and node *C* is responsible for the link from *C* to *D*. A node can find a broken link by checking acknowledgements of data packet sent. For instance, when node *A* sends a packet to *B*, *A* expects to receive an acknowledgment from *B*. If there is no acknowledgement within a certain period, *A* determines that link from *A* to *B* is invalid. Theoretically, acknowledgements are often provided at no cost because they are often in a part of MAC sub-layer standard. However, it is necessary that the DSR must be tied up with MAC sub-layer protocol in order that the network protocol can receive broken-link-report from MAC sub-layer.

When a node is reported a broken link, it removes this link from its route cache, initiates a Route Error packet (RERR) and submits the RERR to associated nodes. Related nodes, when receive the RERR, do the same way as their reporter did and forward the RERR to the next hops in order that all participating nodes in the route eliminate the broken link in their route cache.

Though DSR has fast response to network change and consumes low overhead, the protocol is still facing some limitation. Firstly, the number of nodes is limited by the protocol because data packet size becomes bigger when network grow and the efficiency of the network will dramatically dropped. Secondly, the protocol does not concern about performance of established route. Frequently, a route discovered by DSR has poor performance and is not reliable. Thirdly, the protocol establishes only one route for each route discovery activity. As a result, the high mobility network may be suffered from high packet drop rate caused by route invalidation.

## 2.3 Ad Hoc On-Demand Distance Vector Routing (AODV)

The Ad Hoc On-Demand Distance Vector Routing (AODV) (Perkins, 2003:1) is an on-demand routing protocol like DSR. AODV attempts to recover DSR disadvantage by allowing each node indicates route for data traveling instead of attaching route information with data packet from the source. Therefore, AODV is able to handle larger networks. However, AODV still retains the desirable feature of DSR that routes are maintained only between nodes which need to communicate.

Similar to DSR, AODV deals with route table management and requires route discovery and route maintenance mechanisms. Nodes in a network have to maintaining route table information even for short-lived routes. A route table entry consists of following fields:

- Destination IP address
- Destination sequence number
- Number of hops needed to reach destination
- Next hop
- List of precursors

The main idea of route table management of AODV is that, for each destination, a node keeps only one-hop neighbor's information.

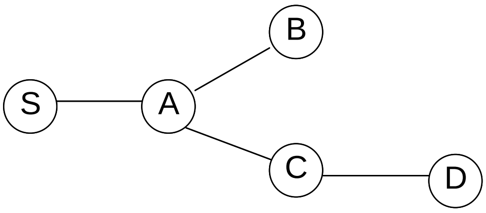

**Figure 2.1** Example of Network Topology from S to D

I explain how AODV work by starting with an instance of network, as shown in figure 2.1, where node S and D are the pair of communicators. To establish a route from S to D, S broadcasts a Route Request packet (RREQ) to its neighbor. The RREQ contains following information:

- *Source IP address* : the address of S

- *Request ID* : the counter maintained by S
- *Destination IP address* : the address of D
- *Source sequence number* : a sequence counter used in the route entry pointing towards the source of the RREQ
- *Destination sequence number* : a sequence number assigned and incremented by D
- *Hop count* : number of hop to the packet has made

The *Request ID* is the counter maintained by S that is incremented each time S's RREQ is broadcasted. Therefore, the combinations of *Source IP address* and *Request ID* can identify the uniqueness of the RREQ that allow any nodes verify and discard the duplicated RREQ.

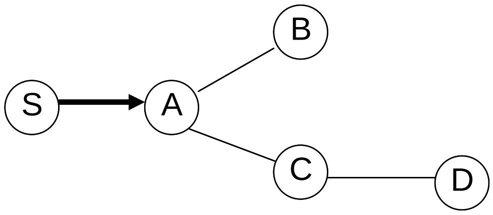

**Figure 2.2** The Direction of RREQ Broadcasting from S to A

Besides the *Request ID*, the RREQ of AODV uses another two control numbers, which is *Source sequence number* and *Destination sequence number*, to control unnecessary RREQ packets floating in the network and avoid the Bellman-Ford "counting to infinity" problem. These numbers are described shortly.

According to the figure 2.2, when the RREQ reach A, A makes a reverse route entry for S and find a route to D in its route table. If A does not have routes to D, then A rebroadcasts the RREQ.

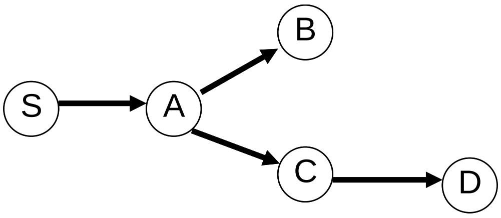

**Figure 2.3** The Direction of RREQ Forwarding from S to D

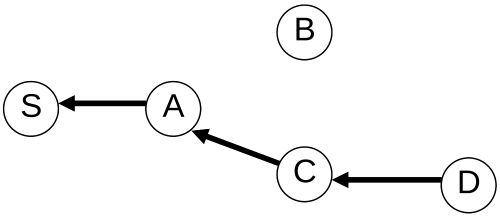

**Figure 2.4** The Direction of RREP Forwarding from D to S

As shown the figure 2.3, B and C do the same process as A did until the RREQ reach D. D, which is the destination, sends a Route Reply packet (RREP) to S by via C and A respectively. The RREP contains following information:

- *Source IP address* : the address of S
- *Destination IP address* : the address of D
- *Destination sequence number* : a sequence number assigned and incremented by D

As I described the elements of the RREQ, if S has never known D, then S assigns the *Destination sequence number* values zero. The *Destination sequence number* will be incremented by D and put into the RREP. The figure 2.4 illustrated the direction of RREP that is forward from D to S. In the mean time, all participant nodes, which receive the RREP, keep the route information to D in their route tables.

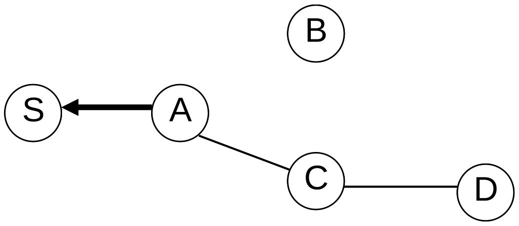

**Figure 2.5** The Direction of RREP Created and Sent by A When A Has a Route to D

Notice that an intermediated node (A or C in this case) can also create and send a RREP to S if the node finds a more recent path than the one previously known. The more recent path is determined by the *Destination sequence number*. As illustrated in the figure 2.5, if A has a route to D and the *Destination sequence number* in A's route table is greater than or equal to the *Destination sequence number* appeared in the RREQ, then A determines that the path A has is a fresh enough path. Contrary, the less *Destination sequence number* in A's route table means that the route to D in the route table of A is out-of-dated.

For the AODV, each intermediate node must maintain reverse route entry in its route table when the node receives the RREQ, for example, A creates a route entry to destination S when A receives the RREQ from S. In the case, the *Source sequence number* in the RREQ is used to determine the fresh-enough paths.

In the case of route maintenance, associated nodes are responsible for monitoring the link status of next hops in active routes. To check the connectivity, nodes that are parts of an active route broadcast a Hello message for every specific time. If the neighbors do not receive the message with in a certain period, the neighbor nodes may assume that the link is currently lost. Besides sensing Hello messages, a node may be notified link failure by other ways. Firstly, a node may consider broken link when data packet cannot be sent within active-route-timeout interval. Secondly, a node may be informed Route Error messages (RERR) from other nodes. When a node found a lost link, it performs following steps:

1. Invalidating existing routes in its route table
2. Listing affected destinations
3. Determining which neighbors may be affected

4. Increment the prior destination sequence number and include the new destination sequence number *N* in the Route Error packet (RERR)

5. Delivering an appropriate Route Error packet (RERR) to such neighbors

Associated nodes that receive the RERR can invalidate existing routes by marking its route to the destination as invalid. When the source S receives the RERR, it initiates a new route discovery for D using destination sequence number at least as large as *N*. When D receives the new route request with destination sequence number *N*, D will set its sequence number to *N* unless the destination sequence number that is held by D is already larger than *N*.

AODV results advantages that the latest routes to destination can be established on demand and low connection setup delay. Nevertheless, AODV lacks support for high throughput routing metrics. AODV is designed to support the minimum hop count which has a tendency to have longer distance between each hop in a route and encounter a higher risk of broken link. As a result, AODV is possible to establish inconsistent and low performance routes.

## 2.4 Split Multipath Routing Protocol (SMR)

Split Multipath Routing Protocol (SMR) (Lee and Gerla, 2001:3201) is an on-demand routing protocol that aims to establish a set of disjoint paths from a source to a destination. The protocol based on dynamic source routing scheme and algorithm to discover maximum number of disjoint paths. Similar to DSR, SMR requires route discovery and route maintenance mechanisms that use source routing approach so that route request, route reply and data packet are transmitted along with source route information. However, SMR has two main differences from DSR. Firstly, for SMR, the intermediate nodes are not allowed sending route reply messages (RREP). This rule relies on the idea that destination is responsible to establish a set of disjoint paths by learning all available routes from source. Secondly, duplicated route request (RREQ) messages are dropped only the messages generate multiple paths that are mostly overlapped. For DSR, the protocol reduces number of RREQ floating in networks by allowing nodes drop every duplicated RREQ which nodes have ever forwarded. This strategy is traded with the limitation of ability to establish only one route. Instead, SMR allows intermediate nodes not only forward the first received RREQ but also rebroadcast all duplicate packets that traversed through other different incoming links whose hop count is not larger than that of the first received RREQ.

There are two main routing processes for SMR; route discovery and route maintenance processes. When the source needs to discovery a set of disjoint route, it launches a RREQ that contains its address and a sequence number. Intermediates, who receive the RREQ, can either attach its address to the RREQ and forward the packet or drop the packet. Intermediates are allowed to forward the RREQ only if the packet traverses through different incoming link from the previously received RREQ and hop count is not large than those of received RREQ. This method causes all available routes information can be forwarded to the destination so that it can select the routes.

When the destination node receives the first RREQ, it sends RREP that contains all associated node addresses for the entire path via this route. Later, the destination must wait for a certain period to collect more RREQ from different links and pick up a set of routes. To select the routes following criteria are applied:

1. The selected route must be maximally disjoint to the route that is already replied.
2. If there is more than one maximally disjoint route, then the shortest hop distance route must be chosen.
3. If there is more than one maximally disjoint route and their hop distance are equal, then the quickest among them are selected.

Finally, all selected route information is replied to the source with RREP through discovered routes.

Route maintenance of SMR is similar to that of DSR. When a node fails to communicate to its next hop, it sends a route error (RERR) back to its original node. The RERR is consequently forwarded to the source and the source removes the broken route from its route table. Finally, the source node can perform one of these two alternatives:

1. Immediately reinitiate the route discovery in order to maintain the number of active routes as equal as a threshold.

2. Use the rest of active routes until there are no more routes available. When there is no active route, the source initiates the route discovery again.

The communicating nodes can exchange data through multiple routes using per-packet allocation scheme that is a data delivery technique by splitting data into pieces and sending pieces of data through multiple routes simultaneously.

There are some advantages for SMR. A benefit is that SMR is capable to discover maximum number of disjoint route. Moreover, nodes use less memory for route cache because SMR allows only source maintain route information to destination and the intermediates do not reply source route information from its cache. However, there are some arguments for this protocol. First, when a node is allowed to forward a RREQ more than one time, it is possible to have large number of RREQ packets traveling in the network. Second, like DSR and AODV, SMR does not have an efficient throughput routing metrics. Instead, it realizes the minimum hop route to be the best route.

## 2.5 QoS-aware Routing Based on Bandwidth Estimation

One of challenges of development of MANET routing protocol is the design of routing protocol to support quality of service (QoS) that guarantee the performance of established routes in the network. There are many considerations for QoS, including network congestion, signal strength, packet delivery, bandwidth and delay jitter. In 2005, Chen and Heinzelman proposed some techniques of bandwidth estimation as well as a MANET protocol that is based on AODV (Chen and Heinzelman, 2005: 561).

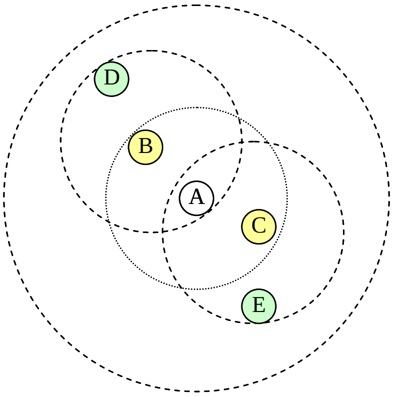

**Figure 2.6** The Interference Range and Two-hop Neighborhood Range of Node A
**Source:** Chen and Heinzelman, 2005: 561.

According to Chen and Heinzelman’s proposal, approximate residual bandwidth is estimated during routing establishment period. The key concept of bandwidth estimation is that each node calculates the residual bandwidth within two-hop neighborhood range. Chen and Heinzelman referred the reason of using two-hop neighborhood range with IEEE 802.11 MAC that, for a wireless channel, the interference range is normally twice the transmission range. As shown in the figure 2.6, the big circle represents the interference range of A whereas B and C are in the transmission range of A. D and E, which is in the transmission range of B and C respectively, is realized as the second-hop neighbor of A. Therefore, A estimates the residual bandwidth by relying on B, C, D and E’s bandwidth usage.

**Table 2.1** The Structure of Bandwidth Information Broadcasted by Node A.

| ID | Consumed Bandwidth | Timestamp |
|---|---|---|
| Neighbor ID 1 | Consumed Bandwidth | Timestamp |
| . | . | . |
| . | . | . |
| . | . | . |
| Neighbor ID n | Consumed Bandwidth | Timestamp |

To enable a node calculates the residual bandwidth, each node periodically calculates its current bandwidth usage, attached the current bandwidth as well as bandwidth consumption of its one-hop neighbors with a hello message and broadcast the hello message. The structure of bandwidth information is represented by the table 2.1. The first row is the node's own information and the following rows are neighbors' information. When a node receives a hello message for its neighbor, the node will know the second-hop neighbors from the message. Consequently, nodes in network keep this bandwidth information in the buffers. Finally, the residual bandwidth can be simply calculated by:

*weight factor = (RTS + CTS + (Data + MACHdr + IPHdr) + ACK) / Data*

*residual bandwidth = ( raw channel bandwidth – overall consumed bandwidth ) / weight factor*

Where *RTS*, *CTS* and *ACK* are the size of control packet of MAC layer, *Data* is the size of data, *MACHdr* is the size of MAC header and *IPHdr* is the size of IP header. For example, if data has 1500 bytes length, then weigh factor should be:

$$(44 + 38 + (1500 + 52 + 20) + 38) / 1500 = 1.128$$

As I mentioned previously that a modified AODV was proposed to implement with residual bandwidth quality of service. To incorporate the residual bandwidth in the AODV, route request packet (RREQ) and route reply packet (RREP) of AODV are modified. In addition, implementing of the routing protocol also requires knowledge of bandwidth a node minimally needs for its running application. In fact, this QoS-aware routing protocol has two models; admission and adaptive. The admission model supports applications that require the network to guarantee minimum bandwidth. The adaptive model is more relax. The model works with applications that can adjust coding rate according to available bandwidth reported by the network.

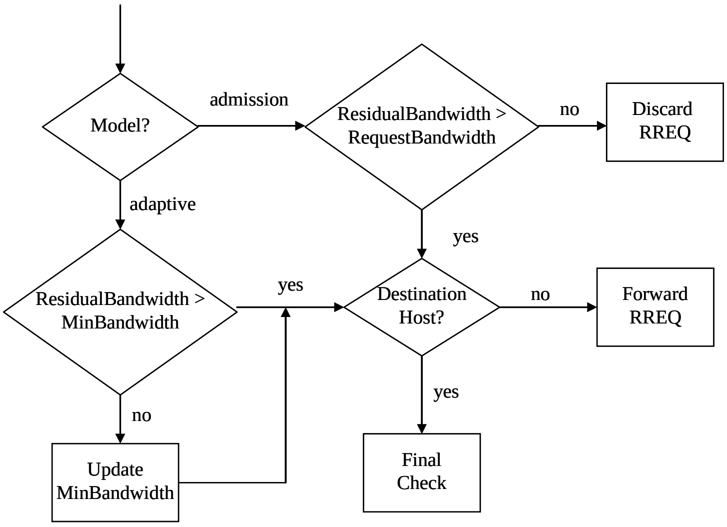

**Figure 2.7** Node’s Decision after Receiving a RREQ

**Source :** Chen and Heinzelman, 2005: 561.

When a source initiates the route discovery process, the source broadcast a route request packet (RREQ) that contains model-flag (admission or adaptive), bandwidth request, minimum bandwidth and AODV REQ header. After receiving the RREQ from the source, an intermediate calculates it residual bandwidth and makes decision according to the figure 2.7. In the case of the model-flag is the admission model, if the residual bandwidth is greater than the request bandwidth, the intermediate forward the RREQ, otherwise they discard the packet. If model-flag is the adaptive model, the intermediate compares the residual bandwidth with the minimum bandwidth. The intermediate will update the minimum bandwidth if the residual bandwidth dropped underneath the minimum bandwidth. Finally, the node forwards the RREQ.

When the destination receives the RREQ, it performs the same procedures as what the intermediates did and executes the final check procedure. The final check procedure associates with recalculation of minimum bandwidth (*MinBandwidth*) that is depended on the number of hops (*HopNumber*). The new minimum bandwidth value can be computed by;

$$MinBandwidth = MinBandwidth / HopNumber$$

Finally, the destination sends the RREP that contains *MinBandwidth* and AODV RREP header to the source. When intermediates receive the RREP, they update the *MinBandwidth* in their route tables and forward the RREP to the source.

Similar to AODV, QoS-aware routing uses hello message to detect route availability. However, QoS-aware routing protocol requires some additional processes regarding disseminate bandwidth information among nodes. These processes are adopted in order to solve problem when a node finds a broken link and reports the route error (RERR) to the other participants. When the source receives a RERR, the node generally initiates a route discovery. During route discovery interval period, it is possible that other intermediates have not yet updated bandwidth information of their neighbors. If intermediates cannot update their buffers in a timely fashion, the established route may not be good enough to support upper layer applications.

To solve mentioned problem, QoS-aware routing utilizes another type of Hello message called “Immediate Hello Message” besides RERR. The structure of Immediate Hello Message is exactly the same as that of hello message except for the message header that allows a node extinguishes an Immediate Hello Message from a regular hello message. When an error is found, the node which report route error will broadcast an Immediate Hello Message immediately after sending RERR to participants. Consequence of receiving the Immediate Hello Message, a neighbor will suddenly start over broadcasting its regular hello message. Finally, an intermediate node is expected to receive both RERR and bandwidth information almost the same time and the node can eliminate the broken route from its route table as well as adjust bandwidth information from its buffers.

## 2.6 Linear Regression Model

The linear regression model (Erwin, 1999) is a statistical model that is based on equation;

$$y = \alpha + \beta x$$

The model is beneficial for investigating the dependence between two variables; *x* and *y*. The variable *x* is called the "independent variable" that is controlled by an observer. The variable *y* is the variable the observer is interested in which the value of *y* is depended on *x*. An example related to this paper are the radio signal strength *y* dBm on the distance between two nodes *x*. In the linear regression model, the value of *y* varies by the value of *x* is in a straight line manner. Therefore, the direction of the change of *y* is defined by two constants; α and β where α is *y*-intercept and β is the slope of the line.

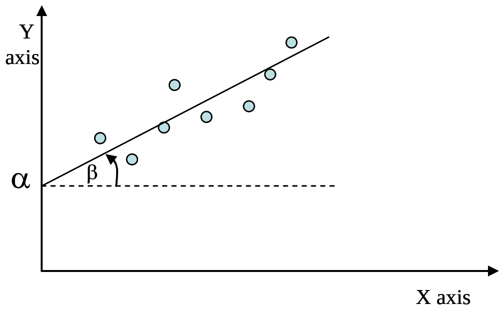

**Figure 2.8** The Relationship between Values of *x* and *y* in a Linear Manner

To build a linear regress model, a number of historical data of *x* related to *y* are collected. Consequently, α and β are calculated by statistical methods. As illustrated in the figure 2.8, dots in the chart are values of *x* and *y*. When the model is built by identifying α and β, The model will come up with a regression line that shows the dependence of *y* on *x*.

Initially, the coefficient of *x* and *y* is calculated. The coefficient of *x* and *y* can be realized as the population covariance of *x* and *y*. When dot $(x_1, y_1), (x_2, y_2), \dots , (x_n, y_n)$ are plotted into a two-axis chart, the model needs to know the degree how much scatter of the dots are. Thus the coefficient of *x* and *y* can be computed by:

$$cov(x,y) = \frac{\sum(x_i-x_{avq})(y_i-y_{avg})}{n - 1} \qquad (1)$$

where *cov(x,y)* = coefficient of *x* and *y*,

$x_i$ = value of *x*,

$y_i$ = value of *y*,

$y_{avg}$ = average $y$,

$x_{\text{avg}}$ = average $x$,

$n$ = number of $x$ or $y$.

Next, the variance of $x$ is computed. The variance of $x$ is the measurement of deviation of population $x_1, x_2, \ldots, x_n$. The representation of this measurement can be referred as square of standard deviation (or variance). The computation of variance of $x$ is as following formula:

$$S^2_x = \frac{\sum(x_i - x_{avg})^2}{n\ \text{-}1} \qquad (2)$$

where

$S^2_x$ = variance of $x$,

$x_{\text{i}}$ = value of $x$,

$x_{\text{avg}}$ = average $x$,

$n$ = number of $x$.

To find the value of α and β using statistic model, the following formulas are used:

$$\beta = \frac{cov(x, y)}{\text{S}^2_x} \qquad (3)$$

$$\alpha = y_{avg} - \beta x_{avg} \qquad (4)$$

where

ß = the slope coefficient derived from linear regression,

*cov(x,y)* = coefficient of $x$ and $y$ referred in the equation (1),

$S^2_x$ = variance of $x$ referred in the equation (2),

α = the intercept of y axis,

$x_{\text{avg}}$ = average $x$,

$y_{avg}$ = average $y$.

As a result, ß will always lie between 1 and -1. The linear regression model can be used for predicting the future value of $y$ by representing $x$.

# CHAPTER 3

# THE PREDICTIVE MULTIPATH ROUTING PROTOCOL (PMP)

## 3.1 Overview of PMP

The Predicted Multipath Routing Protocol (PMP) is an on-demand routing protocol that bases on source routing scheme. According to PMP, nodes in network maintain their own route table that is incorporated with a number of routes and measurement parameters. When nodes in network use PMP to obtain sets of route to destinations, routes in each route set are prioritized by network efficiency measurements. Consequently, each node chooses the most efficient route to be the primary route and the others to be the backup routes. Data can be exchanged along with the active route. When the active route is invalid, the node picks up the best among the backup routes to be the successor.

PMP is incorporated with two types of network measurement. The first measurement is called “Degree of Path Availability (DA)” which targets on predicting future availability of a route. The second measurement associates with measuring efficiency of a route and is named “Estimated Path Throughput Value (ETV)”.

There are three main processes for PMP: node probing, route discovery and route maintenance. Node probing is a process that enabling nodes in network collect data to measure a path. Route discovery associates with establishment of a set of high quality paths using path efficiency prediction models and path shortening function. Route maintenance concerns with how associated nodes of a path perform when a node finds that the path become invalid.

## 3.2 Path Robustness Measurement

In MANET, change in network topology is a major factor for path availability. Since it is difficult to discover location of nodes without using special equipments

(Doval and O' Mahony, 2002), I adopt investigation of signal strength as a measurement of robustness of a path. The measurement of a path robustness aims to predict the future survival of a path. To allow a node knows signal strength of its neighbours, all nodes in network must execute node probing process that each node broadcasts dedicated pulse packets for every certain period. When a node receives a pulse packet from its neighbours, it keeps signal strength information in a window for a particular period. Consequently, the node is able to calculate potential signal strength and the packet loss ratio as well as anticipate the future values of these metrics.

My approach bases on an assumption that all nodes in the network use the same equipments and are in the same environment. Therefore, the measurement of a link can be scaled by using simplex scheme. For instance, if there is a link between nodes X and Y, I assume that the signal strength of X received by Y and of Y received by X are equal. As a result, this link is simply measured by investigating X's measurement.

When a node moves, depending on the direction and speed, it is possible to be out of communication range of its neighbours and causes link failure. If a node moves further, the signal strength will be decreased. The signal strength can be measured using the Receive Signal Strength Indicator (RSSI) (IEEE, 1999). The RSSI is a parameter reported by a network interface card that has value between 0 to $\text{RSSI}_{Max}$ (IEEE, 1999). Since there is no standard for $\text{RSSI}_{Max}$ value which is differently defined by different hardware vendors (Pavon and Choi, 2003:1108), I normalize signal strength metric by

$$\text{RSSI}_{Norm} = \text{RSSI} / (\text{RSSI}_{Max} + 1)$$

where $\text{RSSI}_{Norm}$ is the value of normalized signal strength and $0 \leq \text{RSSI}_{Norm} \leq 1$. For example, when $\text{RSSI}_{Max}$ equals to 31, the RSSI of 16 makes the value of $\text{RSSI}_{Norm}$ to 0.5.

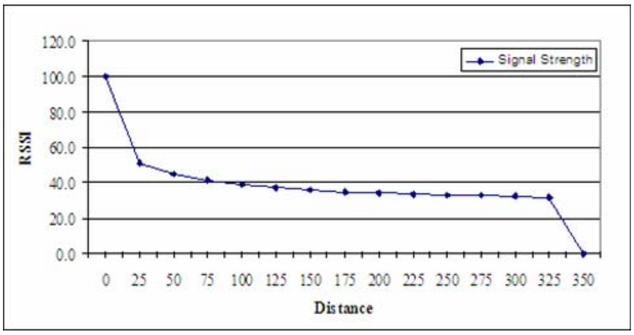

**Figure 3.1** Relationship between RSSI and Distance

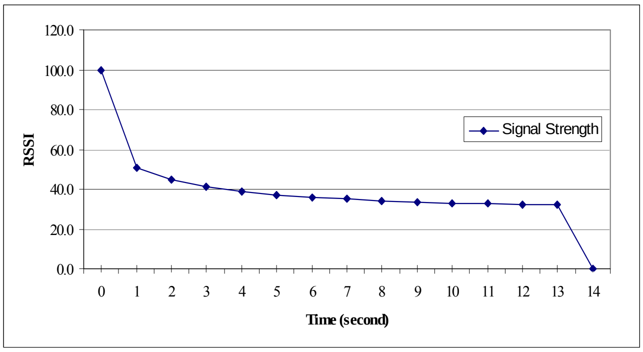

**Figure 3.2** Relationship between RSSI and Time When Nodes Move at 25m/s

By nature of radio signal propagation in free space, radio signal power is proportional to the inverse of the square of distance from the source. Typically, the radio signal power is valued by mW or dBm, that mW reports the rate of radio wave energy, and dBm is a product of logarithmic function of mW (IEEE, 1999). Even though there is no standard of relationship between RSSI and radio signal power metric, most of hardware vendors define value of RSSI based on dBm. Therefore, for a range of distance, the relationship between RSSI and distance will be logarithmic. I experiment in my simulation for two nodes that move apart with constant speed as shown in figure 3.1 and 3.2. In the figure 3.1, the results have shown that the RSSI correlated with a range of distance in logarithmic. As illustrated in the figure 3.2, when the nodes move apart with aggregate constant speed at 25 m/s, the relationship between RSSI and time will be as same as that is shown in the figure 3.1. Hence, I employ a linear regression model for forecasting the potential RSSI$_{Norm}$. The regression model is a probabilistic model that based on statistical information. The model can be described by

$$y = \alpha + \beta x$$

where $y$ is a dependent variable, $x$ is an independent variable, α is y-intercept and β is the slope of the line.

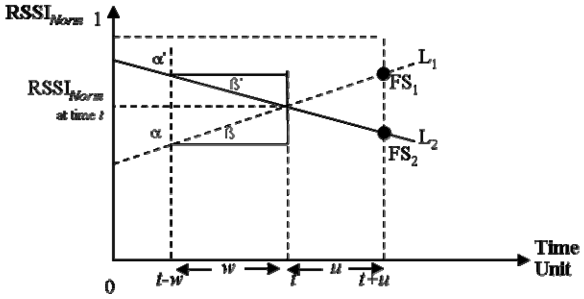

**Figure 3.3** A Chart of Two Regression Lines L1 and L2

To compute potential RSSI$_{Norm}$, every node has a window of size $m$ to keep the RSSI$_{Norm}$ value for every probing time $i$. In the figure 3.3, the potential RSSI$_{Norm}$ depends on historical data during last $w$ time. If the regression line L$_1$ is the increasing line then the anticipated signal strength for the next time $u$ will be FS$_1$. On the other hand, if the regression is decreasing as shown with L$_2$ then the predicted signal strength decreases to FS$_2$. In the case, α is the signal strength of time $t$-$w$ and I can compute ß which is the slope coefficient derived from linear regression based on historical signal strength data from time $t$-$w$ to time $t$ by

$$\text{ß} = \frac{\mathrm{cov(TU, SS)}}{S^2_{\mathrm{TU}}} \tag{1}$$

$$S^2_{\mathrm{TU}} = \frac{\Sigma(\mathrm{TU}_i - \mathrm{TU}_{avg})^2}{m - 1} \tag{2}$$

$$\text{cov}(TU, SS) = \frac{\sum(TU_i - TU_{avg})(RSSI_{Norm-i} - RSSI_{Norm-avg})}{m - 1} \qquad (3)$$

where

ß = the slope coefficient derived from linear regression,

cov(TU,SS) = coefficient of time unit and signal strength,

$S^2_{TU}$ = variance of time from time *t-w* to time *t*,

$TU_i$ = time unit of time *i* from time *t-w* to time *t,*

$RSSI_{Norm\text{-}i}$ = $RSSI_{Norm}$ of time *i* from time *t-w* to time *t*,

$RSSI_{Norm\text{-}avg}$ = average $RSSI_{Norm}$ from time *t-w* to time *t*,

$TU_{avg}$ = average time unit from time *t-w* to time *t*,

*m* = number of window size to keep the signal strength value for every probing time.

From the current time *t*, if *u* is a certain time next from *t* that a node must sustain the path availability, the node forecasts the potential signal strength from its neighbour of time *t+u* by:

$$RSSI_{Norm\ (t+u)} = \min(1, ßu + RSSI_{Norm\ -t})$$

where $RSSI_{Norm\ (t+u)}$ is the normalized potential signal strength for the neighbour at elapsed time *u* and $0 \leq RSSI_{Norm\ (t+u)} \leq 1$, $RSSI_{Norm\ -t}$ is the normalized signal strength of current time *t*, and ß is the slope coefficient.

The signal loss from a single link causes the whole path fail. I therefore consider the minimum normalized signal strength of links in the path to justify future survival of the path with *k* links by:

$$DA = \min(RSSI_{Norm\text{-}1}, \ldots, RSSI_{Norm\text{-}j}, \ldots, RSSI_{Norm\text{-}k})$$

where DA represents the Degree of Path Availability and $RSSI_{Norm\text{-}j}$ is the potential normalized signal strength of link *j* in the path.

## 3.3 Path Efficiency Measurement

The path efficiency measurements focus on estimated total number of data packets launched for one data packet successfully delivered. The estimation of number of packets is basically derived from packet loss ratio. The packet loss ratio is the proportion of a packet that is not successfully received to total packet transmitted within a certain period. During a period when a node receives pulse packets from its neighbour, it can compute packet loss ratio of its neighbour. Finally, the estimation of end-to-end total packets can be achieved by using a statistical approach. In the case, I introduce a metric called the Estimated Path Throughput (ETV).

### 3.3.1 Packet Loss Ratio

The packet loss ratio is the proportion of number of packets that was losing during data transmission to number of total packets a node submitted. The packet loss ratio requires collection of link pulse data. According to pulse a link, a node broadcasts a dedicated packet for every fixed period *t*. The neighbour keeps the pulse data in the window during the last *w* seconds. For every time *t*, a node investigates the number of pulse in window *M* and calculates the average packet loss ratio by:

$$R = \frac{E - P}{E}$$

where R is the average packet loss ratio. P is the number of pulse received in the window of size *m*. E is the number of pulse that should have been received. The loss ratio is consequently used for path throughput computation.

### 3.3.2 The Estimated Path Throughput

The Estimated Path Throughput is a measurement based on the number of potential packets that are completely sent by the source and received by the destination within a certain period. The path throughput is impacted by packet loss and number of hops in the path. For packet loss of a link, a node must resend a new packet if the node found that transmitted packet did not reach the target. If a node has to transmit every two packets in order to achieve one packet communication, the throughput of a link is likely to be reduced to 50%. In the sense of number of hops, when the source transmits a packet, intermediate nodes must repeat the data to its neighbours until they reach the destination. If the transmission among nodes is perfect, the throughput of a path with two hops is considered to be a half of that of a path without intermediate node. It is not necessary that the minimum hop-count path has the best throughput. In fact, the minimized hop-count route is likely to have longer distance among nodes, making nodes communicate with poorer signal and facing more packet loss.

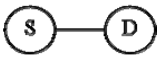

**Figure 3.4a** A Path without Intermediate Node

**Figure 3.4b** A Path with One Intermediate Node

The Estimated Path Throughput Value (ETV) is computed using the estimated number of packets flooding in the path within a specific period. The value of ETV is between 0 and 1. The figure 3.4a shows the communication between two nodes. If there is a perfect communication between node S and D and both of them transmit a data packet with the rate 1 packet per 1 unit of time, the ETV is computed by:

$$\text{ETV} = \frac{1}{\text{Total Number of Packet}}$$

This situation makes the maximum ETV of 1 since there is only one packet transmitted in the path. When there is another hop existing in the path as shown in the figure 3.4b, there must be at least two packets flooding along the path which spend 2 units of time to deliver a packet to the destination. If the communications of all nodes are still perfect, the ETV becomes 0.5. In the situation where there is 50% packet loss with two hops communications, the path is likely to have 4 packets flooding in the network or it consumes 4 units of time to completely deliver a packet. Therefore, the ETV will be:

$$\text{ETV} = \frac{1}{4}$$

Since the average packet loss ratio is used for estimating the number packets flooding along the path and if I define the maximum throughput to be 1, then the ETV will be:

$$\text{ETV} = \frac{1}{\sum_{i=1}^{n} \frac{1}{(1 - R_i)}}$$

Where $R_i$ is the average packet loss ratio of link i and $n$ is the number of links in the path.

## 3.4 Routing Protocol

Basically, the PMP is related to a number of routing functions including node probing, path shortening, route discovery, data transferring and route maintenance. Nodes which are executing PMP have to maintain a route table for keeping route information. The details of routing functions and route table are described in this section.

### 3.4.1 Node Probing

Node probing is a technique that allows a node reports link status to its neighbors and collect linkage quality information from its neighbors. A node probes links by broadcasting a dedicated probe packet for every fixed period. When receiving a probe packet, a receiver is also able to get signal strength information reported by the network interface card. In the case, the receiving node maintains its neighbor's link information in a window. For example, node *X* broadcasts a probe packet for every period *t*. The recipient *Y* keeps probe results in the window of *q* time slots during the last *w* time units.

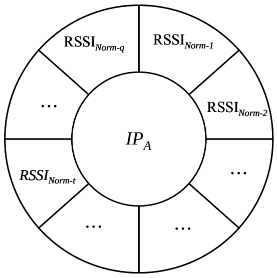
$\text{RSSI}_{Norm\text{-}q}$
$\text{RSSI}_{Norm\text{-}1}$
…
$\text{RSSI}_{Norm\text{-}2}$
$IP_A$
$RSSI_{Norm\text{-}t}$
…
…
…

**Figure 3.5** A Window with *q* Circular Time-slots

I depict the characteristic of the window in figure 3.5. When node *Y* finds a new neighbor node *X*, *Y* creates a window which has *q* circular time-slots and keeps the IP address of node *X* as window identity. For every certain period, *Y* stores received $\text{RSSI}_{Norm}$ from *X* into each time-slot. If *Y* does not receive a prospected pulse packet, the next time-slot is filled with blank. As a result, *Y* can calculate potential signal strength, packet loss ratio and ETV when needed.

To decide whether a neighbor node should be added to or eliminated from a neighbor node list, a node uses regression forecasting scheme that is explained in 3.2. If the potential $\text{RSSI}_{Norm}$ is less than a threshold, then the node removes neighbor information from the neighbor node list. Otherwise, the node still maintains the neighbor node in the list.

### 3.4.2 Path Shortening

Path shortening is a function that allows a node recreates a new better ETV path from an existing path. Path shortening relies on a concept that number of hop can be an influence of ETV. In another word, path with less number of intermediate nodes has an opportunity to have better ETV. The path shortening function comprises of two activities. The first activity concerns with rebuild the minimum hop path from an existing path and the second associates with computing ETV of the new path to compare that of the existing path. For instance, when a node receives route information from its neighbor's RREQ, the node checks whether the incoming path can be shortened. As illustrated in figure 3.6, a path from *S* to *C* can be either *S*-*A*-*C* or *S*-*A*-*B*-*C*. If the path *S*-*A*-*C* has ETV equal to 0.7 while the path *S*-*A*-*B*-*C* has ETV at 0.6. During a period, if C knows that both *A* and *B* are the neighbors of *C* and *C* is informed route information *S*-*A*-*B* from *B* but does not receive route information *S*-*A* from *A*, *C* can rebuild a new path *S*-*A*-*C*. Consequently, *C* recalculates the ETV of *S*-*A*-*C* to compare with the ETV of *S*-*A*-*B*-*C*. If *C* finds that the path *S*-*A*-*C* has better ETV then *C* reports *S*-*A*-*C* as the path to *D*.

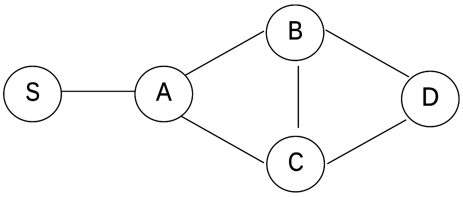

**Figure 3.6** Connectivity Graph of Nodes in a Network

I define the path shortening function by

Shorten(Info$_{Route\text{-}i}$, $IP_i$)

Where Info$_{Route\text{-}i}$ is sequences of IP address from source to node *i* and $IP_i$ is the IP address of node *i* which is the current hop. The instance of representation according to figure 3.5 is

Info$_{Route\text{-}C}$ = { $IP_S$, $IP_A$, $IP_B$ , $IP_C$ }
Shorten(Info$_{Route\text{-}C}$, $IP_C$)

The result of Shorten() function is { $IP_S$, $IP_A$, $IP_C$ }

### 3.4.3 Route Discovery

The route discovery process associates with finding a set of robust and high-efficient disjoint paths. Route from the source to the destination can be viewed as a graph which nodes in the network represent vertices and links between each pairs of vertices refers to edges. The disjoint-paths are a set of paths that there is no common edge for any paths from the source to the destination.

The route discovery employs route request/reply strategies to bring up a number of candidate routes. When a node needs to communicate with other nodes, it initially finds whether there is at least a route to the destination in its route table. If a route or a set of prioritized routes is found, the node utilizes the existing route information for data exchange. On the other hand, if there is no route information, the node starts up route discovery process by launching a Route Request (RREQ) packet over the network.

A RREQ packet is an IP packet that Options section of IP header is used to contain route request information that have following structure.

**Table 3.1** The Structure of Route Request

| 8 Bits | 16 Bits | 32 Bits | 64 * X Bits | 1 Bit | 32 * Y Bits |
|---|---|---|---|---|---|
| Type | Req ID | Destination | Intermediate Info | Terminator | Recipients |

**Table 3.2** The Detail Structure of Intermediate Info

Intermediate Info

| 32 Bits | 16 Bits | 16 Bits |
|---|---|---|
| Intermediate Address | Packet Loss Ratio | $\text{RSSI}_{Norm}$ |

The first field of route request information is *Type*. The *Type* is used to distinguish between different categories of packet used for PMP. The *Req ID* is identity of a RREQ packet. This field allows a node that received the RREQ to determine whether the packet is a new packet or an expired packet. The *Destination* represents IP address of the destination node. The *Intermediate Info* refers to an array of 64 bits data chunk of information of each hop. Each chunk consists of three elements. The first element is *Intermediate Address* that identifies IP address of next hop. The second and third element is *Packet Loss Ratio* and $RSSI_{Norm}$ that indicates the value of packet loss ratio and normalized signal strength of next intermediate node. The *Intermediate Info* is ended by the *Terminator*. In fact, the Terminator is a special character which separates the *Intermediate Info* and *Recipients* information. The *Recipients* represents an array of IP address of nodes that are intended to be recipients of the RREQ. A node identifies the recipients by investigating its neighbor node list that is explained in 3.4.1.

After broadcasting the first RREQ, the source node will wait for a certain period in order to receive a number of Route Reply (RREP) packets from the destination. If the source does not find any replies from the destination, it reattempts to find routes by launching another RREQ. The source will cease route discovery process when a number of RREQs are launched without receiving any RREP.

The route discovery allows only the destination node to send the RREP back to the source in order that a set of qualified routes can be generated. Every intermediate node maintains a RREQ cache to temporarily keep incoming active RREQs. When an intermediate node receives a RREQ, it decides whether the packet should be dropped by performing the following instructions:

Step 1. If the *Req ID* of the RREQ is a duplicated number then drop the packet, otherwise go to the next step.

Step 2. Check the source route information in the RREQ. If the node's neighbors are listed in route information, then perform path shortening function to recreate new source route information.

Step 3. Review the source route information. Compute the Degree of Path Availability (DA) of the path from source to itself. If the DA is less than the DA threshold, then the node drops the packet. Otherwise, go to the next step.

Step 4. Keep the RREQ information into the cache, update the RREQ with the new source route information (*Intermediate Info*) and assign node's neighbors to be the *Recipients* of RREQ.

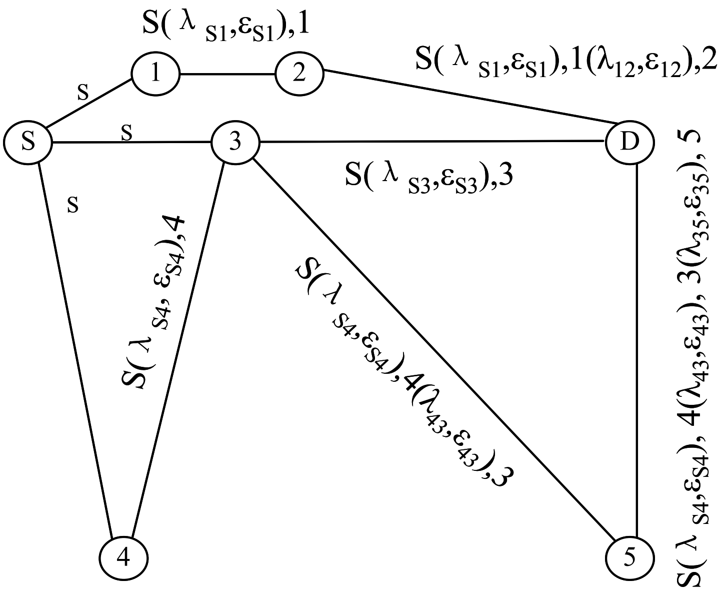

**Figure 3.7** Route Information That is Submitted Each Hop.

To build the new source route information, an intermediate node attaches Packet Loss Ratio and RSSI$_{Norm}$ of its preceding link with the source route information and forwards the RREQ to its neighbour. The instance of how source route information and path measurements values are attached with a RREQ is illustrated in the figure 3.7. The figure 3.7 shows route information that is submitted each hop where λ refer to RSSI$_{norm}$ and ε refer to ETV. When node 1 receives a RREQ from S, it attaches the RSSI$_{norm}$ of S ($\lambda_{S1}$) and ETV of S ($\varepsilon_{S1}$) with the route request and then broadcast the RREQ. When node 2 receives a RREQ from 1, it also attaches the RSSI$_{norm}$ of node 1 ($\lambda_{12}$) and ETV of node 1 ($\varepsilon_{12}$) with the route request and forward the attached RREQ. The node D, when receives the RREQ from node 2, knows the whole path with RSSI$_{norm}$ and ETV of each hop ($\lambda_{S1},\varepsilon_{S1} > \lambda_{12},\varepsilon_{12} > \lambda_{2D},\varepsilon_{2D}$). Then the node D waits for a while to collect more paths. In this case, the node D can collect three paths: S>1>2>D, S>3>D and S>4>3>5>D. The node D finally ranks these paths, build RREPs and send the RREPs back to the source.

After receiving the first RREQ, the destination waits for a certain period to collect a number of qualified paths. To discover a set of acceptable disjoint paths, the destination performs following tasks:

1. Compute the DA and ETV of each path. Eliminate paths which have the DA lower than a threshold.
2. Rank the remaining paths by ETV.
3. Since the problem of finding disjoint path is NP (Bavejay and Srinivasanz, 2000:255; Roy and Garcia-Luna-Aceves, 2002:561), the destination node realizes the maximum number of accepted paths and ceases choosing the paths when the number of accepted paths reaches the threshold.

All accepted paths and their ranks are contained in RREPs and reported back to the source. There may be more than one RREP depending on the number of discovered paths. In this case, the destination considers the rank number to be the sequence number of RREP. In addition, delivery of RREP packets is in unicast manner. Like RREQ, a RREP packet uses Options section of IP header to store qualify path information. The structure of Options section of RREP packet is as follows:

**Table 3.3** The Structure of Route Reply

| 8 Bits | 16 Bits | 8 Bit | 32 * X Bits |
|---|---|---|---|
| Type | Req ID | Rank | Intermediates |

The *Type* field is an identifier of RREP packet. The *Req ID* is the number that refers to *Req ID* of RREQ. The *Rank* is the sequence number derived from path ranking process. The *Intermediates* represents the array of IP address of intermediate nodes of the path.

When receiving the RREP, an intermediate node put the route information into the route table in order that the intermediate node can fix a broken route or report the broken route to associated nodes when link failure is found. Then the node merely forward the RREP back to the next hop until the RREP reaches the source. Finally, the source node keeps all received routes information into its route table.

When the source receives the set of RREP after route discovery, it uses the highest priority route as the primary route to deliver data packets and the others as the backup routes. If the primary route fails, the second priority backup route is then activated. The source node will execute route discovery processes again when all routes are invalidated.

### 3.4.4 Route Table

According to PMP, nodes in a network have to maintain a route table. The route table has following structure:

**Table 3.4** The Structure of Route Table

| 4 Bytes | 4 Bytes | 1 Byte | 4 * X Bytes |
|---|---|---|---|
| Destination | Precursor | Rank | Intermediates |

*Destination* = IP address of the destination.

*Precursor* = IP address of precursor. If the node is a source node, this field will be empty.

*Rank* = Number of priority of the path from 0 1 2 ….

*Intermediates* = Array of IP addresses of intermediates. If there is no intermediate, this field will be empty.

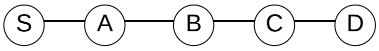

**Figure 3.8** A Path from Source S to Destination D

Information stored in a route table can be explained by using the figure 3.8. The node S, for example, maintains a route as a variable-length record. A route record bases on single direction that is pointed to *Destination* D. Each route record also stores precursor information that is used for route maintenance. In this case, the node is a source then *Precursor* is empty. Each route may incorporate with a number of intermediate nodes that are maintained in the *Intermediates* which is a variable-length field, for example, the *Intermediates* consist of A B and C. Since I use IP address as a node identity and each IP has 4 bytes length, the length in bytes of *Intermediates* can be either empty or product of four and the number of intermediate nodes. Therefore, the size of the *Intermediates* field of a route from S to D is equal to 4 x 3 = 12 bytes.

Routes that have the same Destination can be view as a route set. Routes in each route set are ranked by ETV that rank number start from 0.

### 3.4.5 Data Transfer

When the source node discovers a set of routes to the destination, it attached the first-rank route information into the data packet and transmits the packet to next hop. For example, if the route from source S to destination D comprises of node A, B and C. The source S contains the route information into the data packet by:

A > B > C

The next participant node A, after receiving the data packet from S, eliminate its address from route information and forward the packet to B. The route information sent from A is as follows:

B > C

All participants of this route will forward the data packet until the packet reaches the destination. If the current route is found broken, the source node will pick up the next-rank route as an active route for data traveling.

### 3.4.6 Route Maintenance

The route maintenance process is activated when a broken link or path error is found by a node in the network. The path error typically happens when a node, which is a member of the path, moves out of communication range of its neighbour or is not able to communicate with its neighbour.

For the PMP, a node detects path errors from two events. The first event occurs when a node receives a failure from data link layer during data transmission to the next hop. Another event happens when a node does not receive a probe packet from its neighbour for a certain period.

When a node found broken link to the next hop, the node performs following instructions:

1. Finds a spare path to the destination in the route table. If a backup path is found, then the node uses this path as a new active path by put the new source route information into the sending packet and go to next step. Otherwise, go to the next step.

2. Generates a route error packet (RERR) that contains broken link information and report back to the source.

The way to deliver RERR is unicast via IP packet. For the RERR packet, the Options section of IP packet has following format:

**Table 3.5** The Structure of Route Error

| 8 Bits | 16 Bits | 32 Bits | 32 Bits |
|---|---|---|---|
| Type | Sequence Number | Original Address | Broken Address |

The *Type* is a distinguisher of PMP control packet. *Sequence Number* is the number that is generated by the originator of RERR. *Original Address* is the IP address of the node which found broken link and create RERR. *Broken Address* is the IP address of the node that originator found inactive.

When an associated node receives the RERR, it eliminates all inactive paths from its route table.

## 3.5 Simulation and Result Analysis

The simulation shows the experiment of result performances of my routing protocol. I benchmarked the PMP with two protocols; DSR and SMR. In my dissertation, I did not consider the performance of QoS-based routing protocols that I studied. The reason is that these protocols have some limitations that I could not explicitly make a comparison with the PMP. Concerning with the modified AODV to support residual bandwidth estimation that proposed by Chen and Heinzelman (Chen and Heinzelman, 2005:561), I focused on two limitations. First, the protocol needed cooperation of application, which was using the network, to report the required bandwidth. This is not practical in my simulation. Second, the protocol aimed to find a route that had enough bandwidth while the PMP emphasized on discovering routes that had minimum packet loss. Though the shortage of residual bandwidth might cause congestion and increasing packet loss, my simulation based on a condition that bandwidth was large enough to support data transmission among nodes.

The other QoS aware routing protocols that I had studied were energy-aware QoS routing (Akkaya and Younis, 2003:710) and Adaptive Multi-path Routing (AMPR) (Sun and Hughes, 2003:408). For the energy-aware QoS routing, nodes must be equipped with sensors to report energy consumption. In the case of AMPR, the protocol depended on QoS parameters reported by physical layer such as end-to-end delay, bandwidth and signal-to-noise. These were also not practical in my simulation.

To benchmark the PMP performance, I used Jist-SWANS (Bar, 2004a; Bar, 2004b) to evaluate and compare the performance of PMP, DSR and SMR (Lee and Gerla, 2001:3201).

### 3.5.1 Environments

I modeled a network of 50 mobile nodes located randomly within a 1000 by 1000 meter space. Each node propagated the radio signal at the approximate range of 250 meters with 2 Mbps bandwidth. The radio propagation in the network has random signal-to-noise ratio up to 10 percent.

The mobility and direction of each node was set to random waypoint speed zero to 10 m/s. I used IEEE 801.11b Distribution Coordination Function (DCF) to implement MAC protocol. There were ten pairs of clients and servers were assigned to transmit constant bit rate (CBR) data for 10 packets per second. The size of data payload was 512 bytes. The running simulation time was executed for 300 seconds which I recorded result measurement values for every 30 seconds. For the PMP, every node broadcasts a probe packet for every one second. The nodes started sending data after a 30 second initial timeframe to allow every node submitted a number of probe packets until the network was stable. Since the computational times regarding forecasting model of the PMP vary by the capability of the processors and other resources of nodes, I randomly defined delay time up to 10 milliseconds before transmitting data.

### 3.5.2 Performance Measurement

There were three metrics that I used to measure the performance of the protocols (Corson and Macker, 1999) as follows:

1) Packet delivery ratio: The ratio of the number of data packets completely received by the destinations to the number of data packets generated by the sources.

2) Data yield: The ratio of number of data packets completely received by the destinations to the number of total data packets floating in the network.

3) Number of packet dropped: The number of data packets those are dropped during data transmission process.

### 3.5.3 Simulation Results of PMP Comparing with DSR

The figure 3.9 shows the packet deliver ratio of PMP and DSR. I found that the networks begin to be stable after passing the 90 second-running-period. When network was stable, the packet delivery ratios of PMP were in between 90%-95%. I observed that the packet delivery ratios of PMP were better than those of the DSR. This event can be explained that PMP is multi-path routing protocols that have reserved routes while the DSR does not have, so that PMP can immediately utilize the reserved routes when a broken link was found. In the case of DSR, the protocol must reinitiate the route discovery if the active route becomes invalid. Another reason was that PMP concerned with picking up more durable routes than those selected by DSR. I notice that, route discovery for on-demand routing protocols generally consumes time and always causes a number of data transmission failure that impact the packet delivery ratio

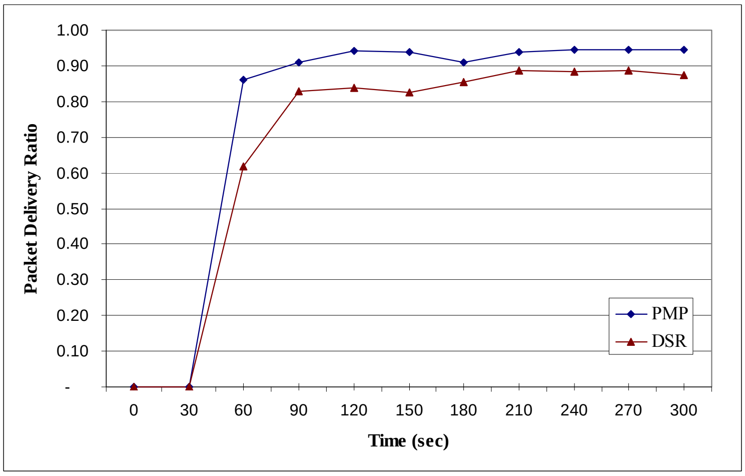

**Figure 3.9** Packet Delivery Ratios

In Figure 3.10, I measured the data yields that scale the efficiency of these three protocols for every 30 seconds. Basically, there are two factors that impact the data yields. The first factor is number of hops from source to the destination and the second factor is communication efficiency between each pair of hops. Since PMP attempted to find the best end-to-end throughput path while DSR focused on choosing paths which have the minimum number of hops, PMP had higher data yields than the DSR. According to the simulation results, the data yields of the PMP were approximate 4% higher than those of the DSR. This simulation proved that the minimum hop number does not necessarily obtain maximum performance.

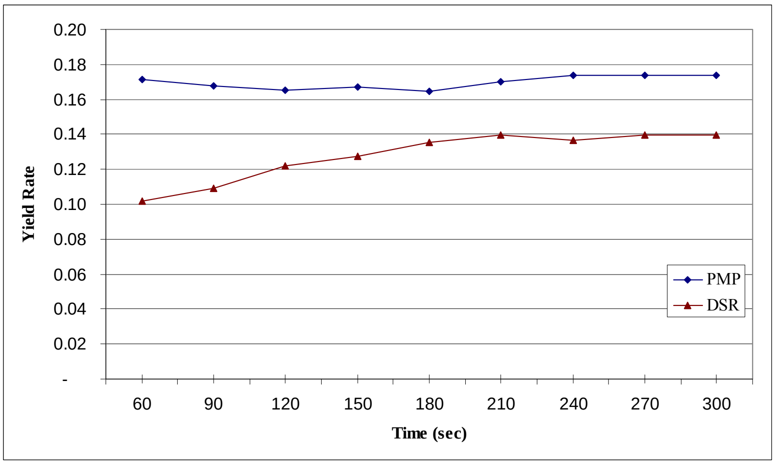

**Figure 3.10** Data Yields

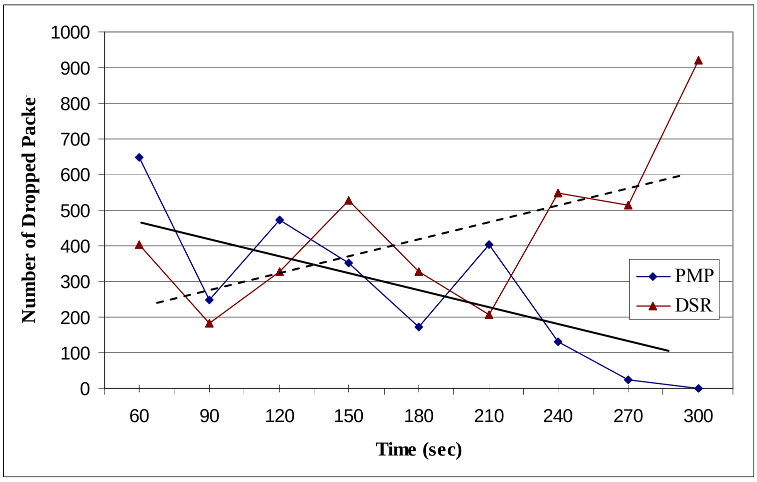

**Figure 3.11** Number of Dropped Packets

Fig. 3.11 depicted the number of packets that are dropped during each 30 seconds period. For the first 30 seconds, I found that the network had not been stable yet. This means that the nodes in network took time in route discovery before nodes started sending data. The dropped packets of DSR appeared to be less than those of PMP for the first 30 seconds because the DSR consumed less time in route discovery. When the network was stable, the dropped packets were in a narrow range between 200 to 600 packets for each 30 seconds or 7 to 20 packets per one second. This is normal because my simulation made total 100 data packets were generated and transmitted for each second. If there were approximate 1 to 4 hops from a source to a destination, the total number of data packets that were floating in the network for a second was approximated 100 to 400 packets. In my simulation, I defined the random signal-to-noise ratio at around 10%. Therefore, the theoretical dropped packets for each second would be 10 to 40 packets.

However, I investigated that the dropped packets of DSR during the last 30 seconds were dramatically increased while those of PMP dropped to zero. This phenomenon was resulted from my simulation that nodes totally stopped sending data when the running period reached 270 seconds. The reason I continued to observe network results for 30 seconds after ceasing data transmission was that I wanted to investigate how much data were stored in the buffers and how much data were dropped. The explanation of how data packets ware buffered before transmission is illustrated in the figure 3.12.

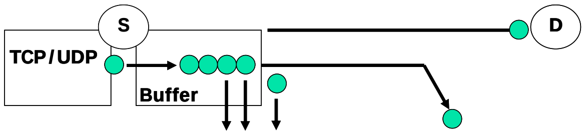

**Figure 3.12** Buffering of Data Packet before Transmission

According to the figure 3.12, when a data packet is prepared to be submitted from the transport layer, the data packet is not immediately transmitted but the packet is pushed into a buffer queue. Then data packets in the queue are consequently transmitted in the first-in-first-out manner. A packet will be dropped when the transmission is failed, the packet is expired or the buffer is full and there is a new packet is pushed into the buffer.

For the case of DSR, the invalidation of an active route happened frequently. When there was no route, data packets were buffered until a route was established. During the last 30 seconds of my simulation, there were a large amount of data packets stuck in the buffer and all packets could not be released from the buffer within 30 seconds. Finally, the large amount of data packets that remained in the buffer was dropped and reported.

To compare overall dropped packets of both protocols, two trend lines were drawn. The thick line represents the trend of dropped packets of PMP and the dashed line refers to the trend of dropped packets of DSR. These lines show that the tendency of dropped packets of DSR is an increasing trend. The direction of this trend is opposite to the trend of PMP which declines when network is stable. I explain from this issue that the PMP provides multi-path routing while the DSR generates only one path for each route establishment. The different routing strategies make the PMP maintains more robust routing than the DSR.

### 3.5.4 Simulation Results of PMP Comparing with SMR

Fig. 3.13a and 3.13b shows the packet deliver ratio of PMP and SMR. I observed from the result that the packet deliver ratios of the PMP are higher than of those of the SMR by approximately three to five percent. The reason was that the PMP has tendency to find better paths than SMR and PMP has more efficient route maintenance process than SMR. However, both protocols are satisfactory for a mobility network. The ratios increased dramatically for the first 30 seconds and climbed up to the maximum value at 300 second simulation time. There is no significant difference between packet deliver ratios under circumstance of with radio signal fading and without radio signal fading. However, I observed the maximum of packet deliver ratio under circumstance of with radio signal fading is higher than those of under circumstance of without radio signal fading approximately one to two percent. This phenomenon can be explained that under radio signal fading circumstance, a node has more chance to receive a packet even though the radio signal is weaker when two nodes move further apart. Under the circumstance of without radio signal fading, the signal strength value is realized to be constant within a range of distance. If a node moves beyond a certain distance from its neighbour, the communication between two nodes will be turned down.

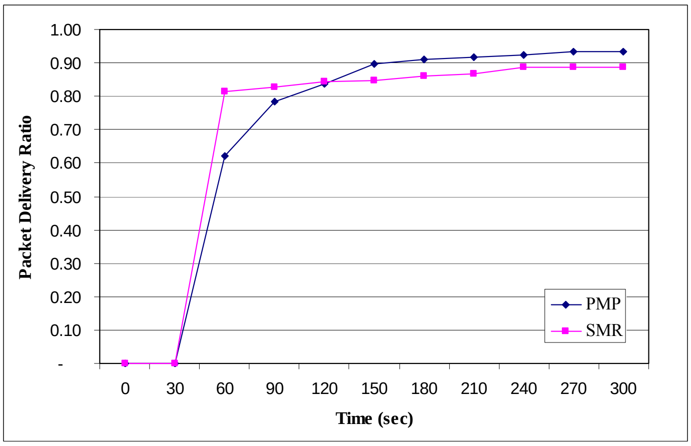

**Figure 3.13a** Packet Delivery Ratios without Radio Signal Fading

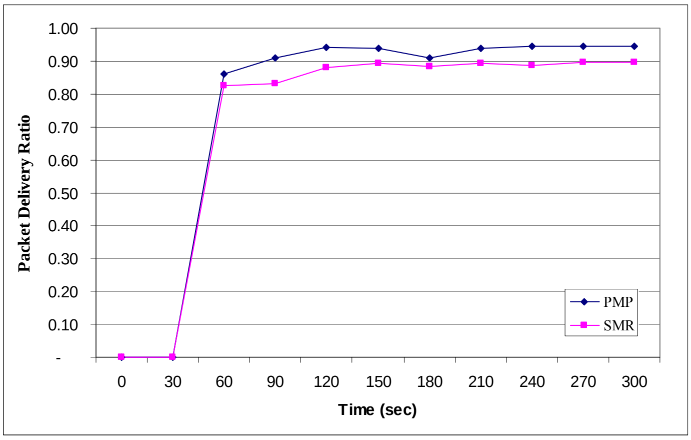

**Figure 3.13b** Packet Delivery Ratios with Radio Signal Fading

The figure 3.14a and 3.14b presented the data yields that reflect the efficiency of both protocols for every 30 seconds. Since the PMP attempts to find the best end-to-end throughput path while SMR focuses on choosing paths which have the minimum number of hops. According to the simulation results, the data yields of the PMP are 15 to 20 percent higher than those of the SMR. This simulation proved that the minimum hop number does not necessarily obtain maximum performance.

When comparing data yields with different circumstances, I found that the data yields under with radio signal fading circumstance were less than the data yields under without radio signal fading circumstance about 40%. This is obvious because communication efficiency between two nodes becomes worse if radio signal strength gets weaker.

The figure 3.15a and 3.15b depict the number of packets that are dropped during each 30 seconds period. The results showed that SMR dropped more data packets than PMP. I explain this occurrence on two reasons. First, PMP finds paths that have better communication efficiency than SMR. The PMP refuses to accept a path if it found a weak potential signal strength between a pair of hops in the path. As a result, the paths found by PMP are more durable than SMR. Second, PMP has the better route maintenance process. For the PMP, if a broken link is found during data transmission period, the intermediate node tries to solve such problem by finding a new path to the destination rather than dropping data packet and let the source identifies a new path.

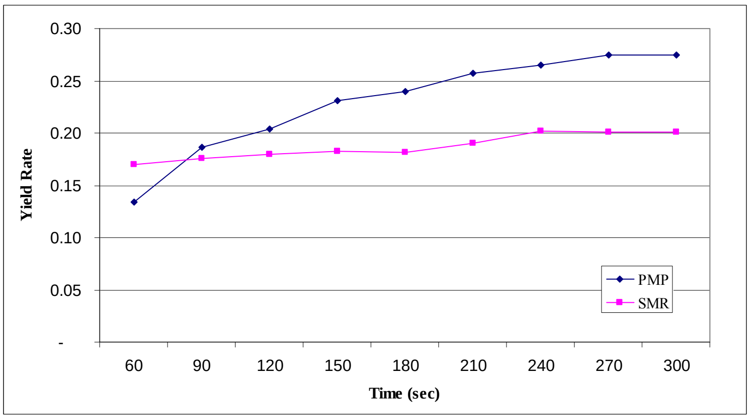

**Figure 3.14a** Data Yield without Radio Signal Fading

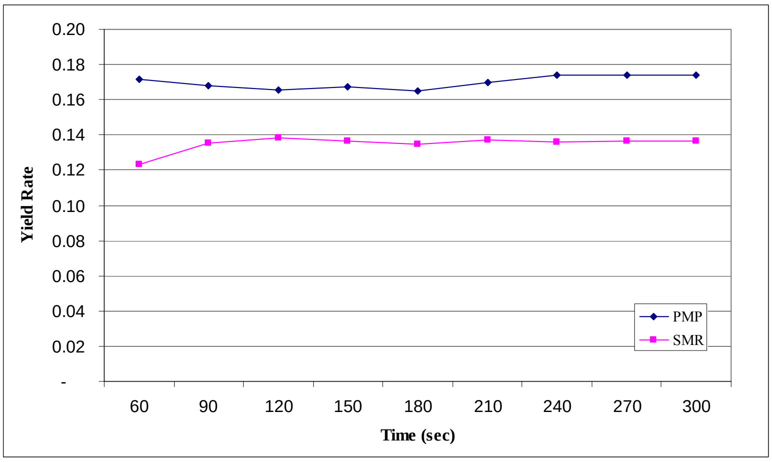

**Figure 3.14b** Data Yield with Radio Signal Fading

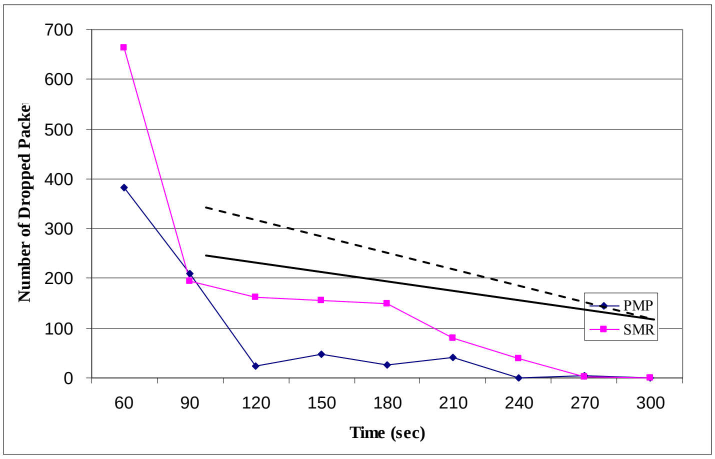

**Figure 3.15a** Number of Dropped Packets without Radio Signal Fading

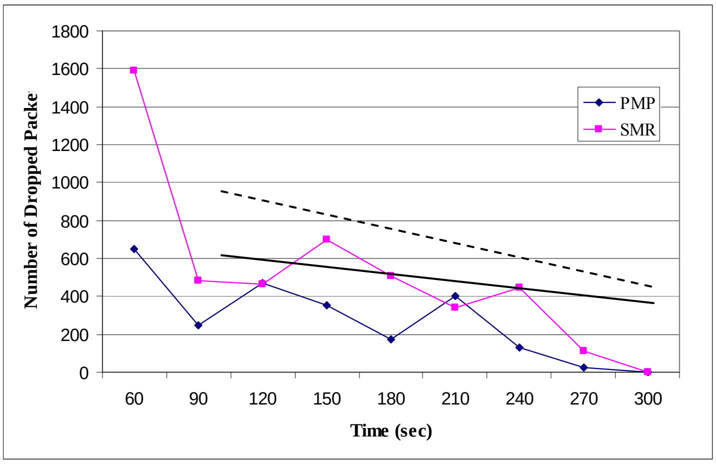

**Figure 3.15b** Number of Dropped Packets with Radio Signal Fading

I also observed that the number of dropped packets under with radio signal fading circumstance is much greater than those under without radio signal fading circumstance. This can be explained that the radio signal fading causes a great impact to efficiency of communication between two nodes. When comparing the trend lines of PMP (thick lines) and SMR (dashed lines), the slopes of the trends of PMP are lower than those of SMR. This means that PMP generates more durable routes than SMR.

# CHAPTER 4

# PRINCIPLES OF NETWORK SECURITY

# AND RELATED RESEARCHES

## 4.1 Introduction to Network Security

Network security is not a new issue for computer network researches. There is plenty of security topic discussed in both educational and commercial area of information and communication technology. Though security is a broad topic and concerns with various problems, the network security problems can be normally grouped into five categories: confidentiality, authentication, nonrepudiation, integrity control and availability. Confidentiality concerns with how to prevent confidential information fall into the hand of unauthorized person or make the information unreadable for trappers. Authentication associates with methods of a person to present himself and his presentation is trusted by others before exchanging information in the network. Nonrepudiation deals with techniques to ensure that messages issued from a person cannot be changed or forged whereas any changes in the messages can be proved. Integrity control related to establishment of processes and mechanisms to guarantee a receiver that the message is not maliciously modified in transit. Finally, availability regards to the defense against any attacks that aim to demolish communications in a network.

Not a single technique can solve entire security problems. In fact, a solution generally requires combination of various processes and mechanisms. However, most of security solutions adopt some primitive techniques include data encryption, message digest and signature.

Data encryption is mainly for data secrecy or confidentiality. For data encryption, messages are coded using a key in order that the unreadable messages can be safely submitted through a network. An authorized recipient is allowed to read these messages by decrypting the coded messages using a correct key. In the case, messages encryption and decryption require particular algorithms. Currently, there are two types of key encryption algorithms: symmetric-key encryption algorithms and asymmetric-key encryption algorithms.

Message digest is employed when a network needs to authenticate a new entry or control data integrity but not for data secrecy. The fundamental of message digest depends on one-way hashing that a hashed message is generated from an original message using an algorithm. The outcome of message digest must have two characteristics. Firstly, there is no way to convert the hashed message back to the original. Secondly, if there is a change in the original message, even one bit, the message digest procedure will generate a different output.

Signature is a technique to authenticate an owner of a message. The signature requires key encryption algorithms to allow a sender signs his/her messages to build a trust to a receiver that the message is really initiated by him/her. When messages are signed with sender’s signature, the sender then cannot later disclaim the contents of the messages and the receiver cannot forge messages and claim that the forged messages are derived from the sender.

## 4.2 Symmetric-Key Encryption Algorithms

The basic idea of data encryption with symmetric-key is that data is encrypted and decrypted using the same key. As illustrated in the figure 4.1, the sender encrypted the message that he/she wants to send using an encryption function with a key. According to this scheme, the receiver must know the key and encryption algorithm the sender uses. When the receiver received encrypted message, he is allowed to read the message by executing decryption function with the same key.

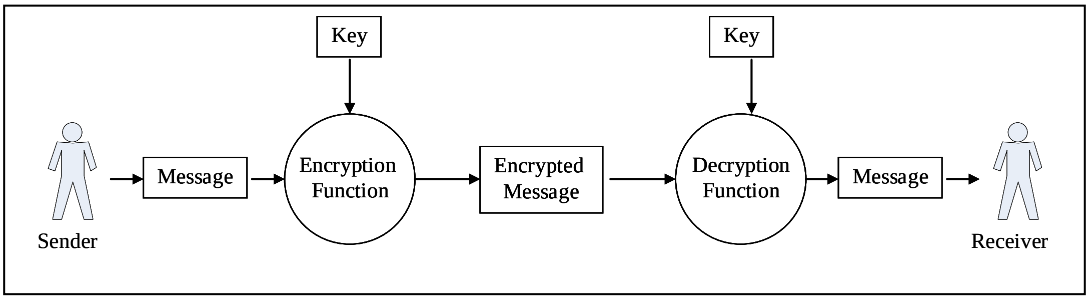

**Figure 4.1** Encryption and Decryption Using Symmetric-key

There is a number of symmetric-key encryption algorithms exist today. Among these algorithms, I focus on two popular encryption standards: The Data Encryption Standard (DES) (National Institute of Standards and Technology (NIST), 1999) and The Advanced Encryption Standard (AES) (National Institute of Standards and Technology (NIST), 2001).

### 4.2.1 The Data Encryption Standard (DES)

The Data Encryption Standard (DES) is a message cipher algorithm that has been widely accepted by network industry for three decades. Fundamentally, DES requires 56 bits key to work with 19 stages of encryption process. Initially, the original message is divided into 64-bit-blocks. Consequently, each block is transposed and then parameterized by different functions of the key. Finally, the ciphered block is inversely transposed.

In 1979, there were some concerns about vulnerability of DES because DES key length was too short and the advancement of computing technology made DES encryption easier to be broken. Therefore, an introduction of Triple DES (3DES) emerged. According to 3DES, DES features were expanded to two keys and three encryption stages while the encryption and decryption algorithms were still depending on DES. For instance, if $K_1$ and $K_2$ are keys of 3DES, then the encryption methods are as following stages:

Stage 1, the original message is encrypted with $K_1$ using encryption function of DES. Suppose the result of the first encryption stages is $Message_{C1}$.

Stage 2, $Message_{C1}$ is decrypted with $K_2$ using decryption function of DES to make the result $Message_{C2.}$

Stage 3, *Message*$_{C2}$ is encrypted with $K_1$ using encryption function of DES again. Consequently, the output of this stage appears to be the secured message that is ready to be sent.

When compared with DES, the result of 3DES encryption is harder to be broken but 3DES consume more computing resources.

### 4.2.2 The Advanced Encryption Standard (AES)

In 2000, many researchers realized that DES was no longer useful for strong cryptographic requirement. Then, the Advanced Encryption Standard (AES) was proposed. Similar to DES, AES divides the original message into blocks before encrypting theses blocks with a key and the encryption methods rely on substitution and permutations in multiple rounds. The differences are that AES supports key lengths and block sizes from 128 bits to 256 bits. Though the key lengths and block sizes can be chosen independently, AES specifies that, the key lengths are limited to 128, 192 and 256 bits if the block size is assigned to be 128 bits. Great advantages of AES are that AES not only produces high security, but also needs less computing resources. These advantages make AES is widely acceptable in network security industry.

## 4.3 Asymmetric-key Encryption Algorithms

The early proposal of asymmetric-key encryption (or public-key encryption) emerged from an attempt to solve key distribution problem. When symmetric-key encryption faces a difficulty that how to safely distribute a shared key between two communicants, the asymmetric-key encryption overcomes this problem by allowing each communicant generates and distributes a public-key without worry about keeping secrecy of key distribution. For one difference with symmetric-key encryption, asymmetric-key encryption algorithms are based on number theory rather than on substitution and permutation.

The principle of public-key encryption algorithms involve with generating a pair of keys, private-key and public key. One can encrypt a messaging with a key and decrypt the encrypted message with another key. For this paper, I state two popular algorithms: RSA and Elliptic curve.

### 4.3.1 The RSA Algorithm

The RSA (Rivest, Shamir and Adleman, 1978:120) was developed in 1977. The method of RSA relies on prime number and Fermat's and Euler's theories (William, 2003), and the scheme of RSA is a block cipher in which the length of block can be freely assigned. However, the typical size of a block is 1024 bits. A disadvantage of RSA is that, the key size must be at lease 1024 bits in order to establish good security. The summary of RSA can be explained as follows:

Key Generation

1. Select two prime numbers, $p$ and $q$.
2. Calculate $n = pq$.
3. Calculate $z = (p-1)(q-1)$.
4. Select an integer $e$ that $e$ is relatively prime to $z$.
5. Calculate $d = e^{-1} \bmod z$.
6. Then public key ($KU$) = [$e$, $n$] and private key ($KR$) = [$d$, $n$].

Encryption

Message $M < n$

Ciphered Message $C = M^e (\text{mod } n)$

Decryption

Ciphered Message $C$

Message $M = C^d (\text{mod } n)$

An important issue of RSA is the complexity of the computation requirements. Since encryption and decryption of RSA involve exponentiation over the integers, the use of RSA needs much more computing resources and time when compared with symmetric-key algorithms.

### 4.3.2 The Elliptic Curve Cryptography (ECC)

The elliptic curve cryptography is a public-key algorithm that currently becomes a standard and challenge of RSA. The interesting point is, when compared with RSA, elliptic curve cryptography offers equal security with less overhead consumption.

The principal concept of ECC (William, 2003:304) bases on cubic equations that can be classified into two types. The first type is called Prime Curve or Elliptic Curves over $Z_p$. The Prime Curve associates with cubic equation in which the variables and coefficients all take on value in the set of integers from 0 thru p-1, for some prime number p. The equation of Prime Curve is as following form:

$$y^2 \bmod p = (x^3 + ax + b) \bmod p \qquad \text{where}$$
$$(4a^3 + 27b^2) \bmod p <> 0 \bmod p$$

When constants *a* and *b* and prime number *p* are set, there will be a set of coordination *x* and *y* that satisfy these equation. The set of values *x* and *y* that satisfy these equation under definition of *a*, *b* and *p* is represented by $E_p(a, b)$. For example, if *a* and *b* are set to 1 and *p* is 23, then $E_{23}(1, 1)$ comprises of following elements;

{ (0,1), (0,22), (1,7), (1,16), (3,10), (3,13), (4,0), (5,4), (5,19), (6,4), (6,19), (7,11), (7,12), (9,7), (9,16), (11,3), (11,20), (12,4), (12,19), (13,7), (13,16), (17,3), (17,20), (18,3), (18,30), (19,5), (19,18) }

The $E_p(a, b)$ is utilized in the sense that one of the members of $E_p(a, b)$ is selected and used for elliptic cryptography.

Another type of ECC equations is Binary Curve which is named Elliptic Curves over GF($2^n$). The Binary Curve relies on using set of integer between $2^n – 1$ for variables and coefficients in a cubic equation. Unlike the Prime Curve, the Binary Curve takes the form

$$y^2 + xy = x^3 + ax^2 + b$$

where x, y and coefficients a and b are elements of GF($2^n$), which represents a set of finite field of order $2^n$ that can be defined over polynomials.

The ECC processes of keys generation and data encryption can be explained as follows:

Key Generation

1. Define global public elements:

   $E_p(a,b)$ is elliptic curve with parameters $a$, $b$ and $p$, where $p$ is a prime number or an integer of the form $2^n$

   $G$ is a point on elliptic curve whose order is large value $n$

2. Establish private key $n_A$ where $n_A < n$
3. Calculate public key $P_A = n_A * G$

Encryption

Message $M$

Encode $M$ to be send as an $x$-$y$ point $P_m$

Ciphered Message $C_m = \{ kG, P_m + kP_B\}$ where $k$ is a random positive integer

Decryption

Ciphered Message $C_\mathrm{m}$

Calculate $P_m = P_m + kP_B - n_B(kG) = P_m + k(n_BG) - n_B(kG)$

Decode point $P_m$ to Message $M$

As an example, if $p$ = 751, $a$ = -1, $b$ = 188 and G = (0,376). When $A$ wishes to send a message to $B$, $A$ must initially transform the message to x, y coordination (e.g. $P_m$). Suppose $A$ sends message to $B$ in point $P_m$= (562,201) and A chooses random number k = 386. If B's public key is $\mathrm{P_b}$ = (201,5), then

386(0,376) = (676,588) and

(562,201)+386(201,5) = (385,328)

Thus cipher text is { (676,588), (385, 328) }

Though ECC concerns with complicate equations and requires more processes for encryption when comparing with the RSA, the ECC needs shorter key and lower computation efforts in order to establish equal securities.

## 4.4 Message Digests

Typically, message digests are employed for authentication and maintaining data integrity. The basic idea of message digests is based one-way hash functions. According to the message digests, a long message is processed by a hash function to establish a fixed-length string. This function has following aspects:

1. It consumes low computing overhead.
2. It is impossible to find the original message from the hashed string.
3. It is difficult to find some other messages that produce as same hashed string as the original message.

For this paper, I mention two famous message digest functions: MD5 (Rivest, 1992) and SHA-1 (NIST, 1993).

### 4.4.1 MD5

The functions of MD5 concern with mixing a message in multiple rounds to generate 128 bits hashed message. Initially, the original message is divided into 512-bit block. The last block is padded to 448 bits and appended by 64-bit integer of the length of the original message. Then a 128-bit output buffer is reserved. Finally, multiple rounds of computation are executed in which each block of the message is mixed with the buffer in each four round. Nowadays, MD5 is still in use even though it has been facing plenty of attacks for almost two decades.

### 4.4.2 SHA-1

SHA-1 stands for Secure Hash Algorithm 1. The concept of SHA-1 is similar to MD5 that there is message mixing for 512-bit blocks. The difference is SHA-1 generates 160-bit message digest. SHA-1 starts with making message length to be multiple of 512 bits by padding the original message with a 1 bit and follows with many 0 bits. Consequently, length of the original message in the form of 64-bit number is ORed into the low-order 64 bits. Then particular methods of message mixing are executed in loop. Currently, there are considerations to enhance securities of SHA-1 by extending the length of message digest of SHA-1 to 256 bits and more.

## 4.5 Signature

Signature can be realized as an implementation of key encryption algorithms and message digests in order to maintain electronic document integrity. The use of signature with an electronic message can satisfy following requirements:

- The sender can trustworthily identify himself/herself to the receiver via a message.
- The message is protected from fraud or change.
- The sender cannot disclaim the message he/she sent.
- The receiver cannot concoct a counterfeit message, which is claimed that the message is issued by the sender.

The scheme of signature is depended on assumptions. A scheme that is used in the real world today relies on the condition that communicants who need signature have to trust a third-party agent. The trusted agent, who is called a certificate authority (CA), must issue a specific data to certify identity of each communicant in order that the communicant is allow to identify himself by submitting this data with his messages. This data is called “certificate”. Technically, the implementation of signature under the condition of the need of CA requires both public-key encryption algorithms and message digests. I explain this implementation in term of technical methods as follows:

1. Assume that both sender ($S$) and receiver ($R$) trust a CA
2. The CA generates a pair keys, private key ($PriKey_{CA}$) as well as public key ($PubKey_{CA}$) and then publicizes $PubKey_{CA}$ to $S$ and $R$.
3. $S$ generates their own pair keys, private key $PriKey_S$ and public key $PubKey_S$. Then $S$ gives $PubKey_S$ to the CA.
4. The CA issues a certificate ($Cert_S$) to $S$. The information contained in the $Cert_S$ are in the form of

$PubKey_S$ + $\text{Sign}_{\text{CA}}(PubKey_S)$ where

$\text{Sign}_{\text{CA}}(PubKey_S) = \text{E}_{\text{PriKey-CA}}[\text{MD}(PubKey_S)]$,

MD() = Message digest function, eg. SHA-1 or MD5,

$\text{E}_{\text{PriKey-CA}}$ = Encryption using private key of the CA.

5. *S* authenticates himself by sending $Cert_S$ to the Receiver (*R*). For this event, *R* verifies the signature of the CA (represented by $\text{Sign}_{\text{CA}}(PubKey_S)$ in step 4) by computing

Result = true if $\text{MD}(PubKey_S) = \text{D}_{\text{PubKey-CA}}(\text{Sign}_{\text{CA}}(PubKey_S))$,

false otherwise.

where

MD() = Message digest function, eg. SHA-1 or MD5,

$\text{D}_{\text{PubKey-CA}}$ = Decryption using public key of the CA.

If the result is true then the *R* trusts $Cert_S$ and $PubKey_S$.

6. The *S* sign a message before sending to the *R* by

$\text{Sign}_{\text{S}}(\text{message}) = \text{E}_{\text{PriKey-S}}[\text{MD}(\text{message})]$,

where

MD() = Message digest function, eg. SHA-1 or MD5.

$\text{E}_{\text{PriKey-S}}$ = Encryption using private key of S .

7. The *S* submit the message together with the signature to the *R* in the form of

message + $\text{Sign}_{\text{S}}(\text{message})$

8. The *R* can verify the message by computing

Result = true if $\text{MD}(\text{message}) = \text{D}_{\text{PubKey-S}}(\text{Sign}_{\text{S}}(\text{message}))$,

false otherwise.

where

MD() = Message digest function, eg. SHA-1 or MD5,

$\text{D}_{\text{PubKey-S}}$ = Decryption using public key of the *S*

If the result is true then the *R* trusts the message.

For this dissertation, I describe only the scheme of signature under an existence of a CA because this signature scheme is the most practical in the real world.

## 4.6 Related Studies of Secure Routing Protocol

There were many researchers have proposed secured ad hoc network routing protocol by relying on different assumption and approach for decades. However, I am interested on the studies of the establishment of secure routes using authentication, digital signature and key agreement mechanisms. For my studies, I refer to three secured ad hoc network routing protocols. There are the ARAN (Sanzgiri , Dahilly , Leviney , Shieldsz and Belding-Royer, 2002:78), SRP (Papadimitratos and Hass, 2002:27) and SEAD (Hu, Johnson and Perrig, 2002:3).

### 4.6.1 The Authenticated Routing for Ad hoc Networks (ARAN)

The Authenticated Routing for Ad hoc Networks (ARAN) is a secure source routing protocol that prevents undesired nodes to participate in network routing. The protocol is basically depended on cryptographic certificate mechanism. According to the assumption of ARAN, there must be a trusted certificate server which is responsible for issuing keys and certificate to all valid nodes.

According to ARAN, a node must ask for a certificate from the trusted server before communicating with the others. The certificate contains a set of trusted information to identify the communicants. In the case, a certificate of node *A* can be represented by

$$cert_A = [IP_A, K_{A+}, t, e]K_{T-} \quad \text{where}$$

$cert_A$ is the certificate the trusted server issues to *A*, $IP_A$ is the IP address of *A*, $K_{A+}$ is the public key of *A*, *t* is a timestamp of when the certificate was created, and *e* is the expired time of the certificate. This information is signed by $K_{T-}$ which is the private key of the trusted server.

The routing scheme of ARAN consists of route discovery and route maintenance. For the ARAN, end-to-end authentication is executed during route discovery. For example, when node *A* initiates route discovery, it broadcasts a route discovery packet (*RDP*) to its neighbors. The route discovery data are encrypted with private key of *A* as following form:

$$RDP_A = [type, IP_D, cert_A, N_A, t]K_{A-} \quad \text{where}$$

$RDP_A$ is the signed route discovery packet issued by $A$, *type* is type of the packet which is a constant "RDP", $IP_D$ is the IP address of the destination node, $cert_A$ is the certificate of source node $A$, $N_A$ is a nonce, and $t$ is the current time. The $RDP$ data is signed by $A$ using private key of $A$ ($K_{A-}$). The ARAN uses the nonce to detect duplicate packets. Each time $A$ performs route discovery, it increase the value of the nonce. The receiver checks this value to see whether the receiving $RDP$ is duplicated with the previous one. If the receiver found a duplicate packet, it simply discards the packet.

When received a $RDP$, an intermediate node signs the incoming $RDP$ with its private key, attaches its certificate with the packet and forwards the packet to its neighbors. For example, let $B$ be a node that receives $RDP$ from $A$. $B$ signs and forwards a packet by

$$RDP_B = [RDP_A]\ K_{B-},\ cert_B \quad \text{where}$$

$RDP_B$ is the signed route discovery packet launched by $B$, $RDP_A$ is the signed route discovery packet issued by $A$, $K_{B-}$ is the private key of $B$ that is used to sign $RDP_A$, and $cert_B$ is the certificate of $B$ issued by the trusted server.

Let $C$ be a neighbor of $B$. $C$ validates the signature of $B$ with the receiving certificate. Then $C$ records node $B$ as its predecessor, removes $B$'s signature, signs the original packet with its private key, attaches the certificate with the packet, and finally rebroadcasts the packet as following form:

$$RDP_C = [RDP_A]\ K_{C-},\ cert_C \quad \text{where}$$

$RDP_C$ is the signed route discovery packet launched by $C$, $RDP_A$ is there signed route discovery packet issued by $A$, $K_{C-}$ is the private key of $C$ that is used to sign $RDP_A$, and $cert_C$ is the certificate of $C$ issued by the trusted server.

A main concept of ARAN for handling a route discovery packet is that each route discovery packet floating in the network is signed by each hop. As a result, it is not possible for malicious nodes to redirect the routes by modifying route information contained in the route discovery packet.

When the *RDP* reaches the destination, the node authenticates the source node, issue a reply message, and unicasts the packet to its predecessor. Let *D* be the destination node, the reply message is:

$$REP_D = [type, IP_{A,}\ cert_{D,}\ N_{A,}\ t]\ K_{D-} \quad \text{where}$$

$REP_D$ is the reply message that is signed by private key of *D* ($K_{D-}$), *type* is the packet type which value is “REP”, $IP_A$ is the IP address of source node *A*, $cert_D$ is the certificate of the destination node *D*, $N_A$ is the nonce and request packet, *t* is the timestamp.

When the intermediate node receives the reply packet, it does the same way as it did with the request packet. Let *C* is the node received the $REP_D$. The packet that *C* submits back to its predecessor is as follows:

$$REP_C = [REP_D\,]\ K_{C-},\ cert_C \quad \text{where}$$

$REP_C$ is the signed route reply message using $K_{C-}$ which is the private key of *C*, $REP_D$ is the signed REP issued by *D* and $cert_C$ is the certificate of *C*. Since each node checks the nonce and signature of its original, the network is capable to avoid both impersonating and replay attacks from malicious nodes.

The route maintenance process of ARAN is simple. When a node detect an error, it generates an error message (ERR), signs the message, and reports to the active nodes in the route. Let *B* is a node that issues an error message to *C*. The error message is represented as following form:

$ERR_B = [\ type, IP_A, IP_D, cert_B, N_B, t\ ] K_{B-}$ where

$ERR_B$ is the route error packet signed by $K_{B-}$, *type* is the type of message which is a constant value of “ERR”, $IP_A$ is the IP address of the source $A$, $IP_D$ is the IP address of the destination $D$, $cert_B$ is the certificate of $B$, $N_B$ is the nonce, and $t$ is the timestamp that are used for duplicate packet detection. In this case, the network can protect denial-of-service attacks because the error message is signed and allowed the receiving node validate the source of the message.

In summary, the concept of ARAN is target for protections of various kinds of active attacks. However the proposal of ARAN does not mention how data confidentiality is implemented after a secure route is established.

### 4.6.2 The Secure Routing Protocol (SRP)

When ARAN is depended on the requirement of a certificate authority, the Secure Routing Protocol (SRP) is under different assumption. The SRP relies on a condition that trusted infrastructure is not applicable in a network and the network is not secure. The fundamental of SRP focuses on security association between each pair of nodes. In another word, a pair of communicants must have a trust relationship between each other and the relationship must be pre-instantiated, for example, by the knowledge of the public key of their communicants. Moreover, both parties must be able to negotiate a shared secret key, for example, via the Elliptic Curve Diffie-Hellman algorithm.

SRP supports routing protocols that use route request ($Q$) /route reply ($P$) scheme. Conceptually, SRP secures route request and route reply messages by hashing the messages with a shared key. For instance, if $S$ is the source and $D$ is the destination. $D$ should ideally receive secure route request information from $S$ via following message:

$Q_{S,D}, H\{Q_{S,D}, K_{S,D}\}$ where

$Q_{S,D}$ is the route request packet from $S$ to $D$, $H\{\ \}$ is a message digest function (MD5 or SHA-1) and $K_{S,D}$ is a shared key between $S$ and $D$.

In the same scenario, the route reply message response by *D* can be viewed in the form of

$$\{R_{S,D}, route\}, H\{R_{S,D}, route, K_{S,D}\} \qquad \text{where}$$

$R_{S,D}$ is the route reply packet from *D* to *S*, *route* is the path information, $H\{\ \}$ is a message digest function and $K_{S,D}$ is a shared key between *S* and *D*.

The key strategy of SRP to implement and control secured route request and route reply is the use of control numbers, which is generated by a source. The SRP has two types of numbers. The first number is Query Sequence Number ($Q_{seq}$) that is unique for each destination and increased monotonically for each query sequence. The $Q_{seq}$ is utilized by the destination to detect outdated route requests. The second control number is Query Identifier ($Q_{ID}$). $Q_{ID}$ is a random number that is used by intermediate nodes to identify the request. Besides the control numbers, SRP maintains a hashed message called *"The Message Authentication Code (MAC)"*. The *MAC* is a 96-bit long field generated by a message digest algorithm using entire IP header, the basis protocol route control packet (route request or route reply) and the shared key as inputs.

To discover a route to a destination *D*, a source *S* generates a route request message ($Q_{S,D}$) that contains IP header, basis routing protocol packet and additive SRP header. The SRP header consists of packet type, $Q_{seq}$, $Q_{ID}$ and *MAC*. An intermediate node handles the incoming $Q_{S,D}$ by maintain $Q_{ID}$ in the query table. The node is allowed to forward only the packet which $Q_{ID}$ does not match with existing $Q_{ID}$ as well as the pair IPs of *S* and *D*.

When *D* receives the $Q_{S,D}$, *D* first verifies the packet that it has originated from *S*. Then *D* investigates $Q_{seq}$ compared with the last sequence number to assure that the packet is not outdated. Finally, *D* generates a number of route replies ($R_{S,D}$) sending back to *S*. Like the route request, the $R_{S,D}$ has an attachment of SRP header, which contains $Q_{seq}$, $Q_{ID}$ and hashed message of route reply information. As a result, *S* can verify the freshness and originator of the reply.

The route maintenance, in fact, is not directly related to SRP since SRP covers only end-to-end message signing. However, SRP has proved that mislead route errors, generated by a node which is not associated in the path, can be detected and ignored by relevant nodes.

### 4.6.3 Secure Efficient Ad Hoc Distance Vector Routing Protocol (SEAD)

SEAD is an improvement of the Destination-Sequenced Distance-Vector Routing Protocol or DSDV (Perkins and Bhagwat, 1994:234). that is proactive routing protocol based on distance vector routing. For DSDV, every node maintains routing information for all known destinations and periodically sent the information to all neighbors to maintain the topology of network. The routing information is stored in a table in which each table entry consists of following data:

- destination address
- the next node to reach to destination
- metric which is the number of hops to reach the destination
- a sequence number originated from destination
- install time when the entry was made
- a pointer to a table holding information on how stable a route is

The DSDV also utilizes the sequence number to identify the status of linkage to each destination by using even number as alive link and odd number as dead link. For a certain period, a node advertises all routing information to its neighbor. On each advertisement, the own destination sequence number, which always be even number, is increased. Recipients will update their routing table according to received advertisement. If a node detects that one of its neighbor is no longer reachable, the node increase sequence number of this node by 1 (odd sequence number) and set metric to infinity. Consequently, the node immediately advertises this broken link to its existing neighbors.

Since the DSDV is unsecured, Hu and et al. proposed SEAD that is based on the design of DSDV but embedded with some security features. The highlight of SEAD is the authentication methods that use one-way hash chains and tree-authenticated values. To create a one-way hash chain, a node initializes a random value $x$ and computes a list of hash values $h_0, h_1, h_2, \ldots, h_n$ by

$$H_0 = hash(x)$$
$$H_i = hash(H_{i-1})$$

where $hash()$ is a hash function and $0 < i \leq n$. For example, if $i = 2$ then the hash chain value is

$$hash\ (hash\ (hash(x)))$$

SEAD start implementing tree-authenticated values by assigning a secret authenticate value to each node and placing values $v_0, v_1, v_2, \ldots, v_n$ at the leaf node. Then $v_i$ *is* hashed by

$$v'_i = hash(v_i)$$

To build tree-authenticated values, a binary tree is forms and hash chaining is executed up to root. As an example as shown in figure 4.2, if there are eight nodes in which each node is assigned a authenticate value $v_0, v_1, v_2, \ldots, v_7$ respectively. The value of $v_i$ is hashed to make the result $v'_i$. Then the next layer of hash values is committed, for example

$$m_{11} = hash(v'_0 \,||\, v'_1)\ ,$$
$$m_{12} = hash(v'_2 \,||\, v'_3)\ ,$$
$$m_{21} = hash(v'_4 \,||\, v'_5)\ ,$$
$$m_{22} = hash(v'_6 \,||\, v'_7)$$
$$m_{01} = hash(m_{11} \,||\, m_{12})$$
$$m_{02} = hash(m_{21} \,||\, m_{21})$$
$$m_0 = hash(m_{01} \,||\, m_{02})$$

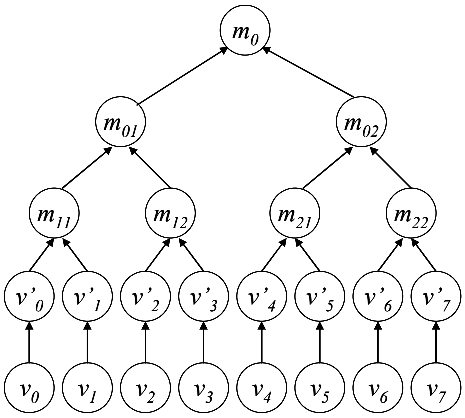

**Figure 4.2** Tree Authenticated Values with 8 Leaf Nodes

**Source :** Perkins and Bhagwat, 1994:234.

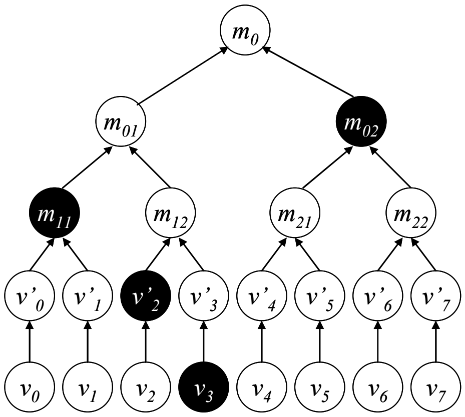

**Figure 4.3** Submitted Values of $v_3$ Authentication

As the figure 4.3, suppose that a node wants to authenticate a value of $v_3$, the node send $v_3$ together with $i$ = 3 and all sibling nodes of the nodes on the path from $v_3$ to root that includes $v'_{2,}$ $m_{11}$ and $m_{02}$ to the recipients. The recipients who have an authentic route value $m_0$ then can verify this value by

$$hash\ (m_{02} \,||\, hash\ (m_{11}\ \ ||\, hash\ (v'_2 \,||\, hash(v_3))))$$

Finally SEAD allows a node authenticate itself to its neighbors during routing information exchange process. As a result, SEAD needs a trusted entity to assign authenticate value to each node, establishes tree-authenticated values and disperses the values to all nodes in network.

SEAD is proved that the protocol is robust for an attack of injecting of incorrect routing state in other node by modifying the sequence number or the routing metric. However, SEAD is still vulnerable from an attack by tempering next hop or destination field in routing information (Rajendra and Mohit, 2010:1793).

# CHAPTER 5

# THE SECURED PREDICTIVE MULTIPATH ROUTING PROTOCOL (SPMP) WITH GROUP AUTHENTICATION

## 5.1 Overview

The Secured Predictive Multipath Routing Protocol (SPMP) is an on-demand multipath routing protocol based on the Predictive Multipath Routing Protocol (PMP). The protocol is incorporated with the basis of multipath routing protocol as well as solid security processes to discover secure routes that only trusted nodes can be participants of the paths. The SPMP allows only selected nodes to be participating in route discovery process. In the case, a node must identify target receivers of route discovery message and contain the target receiver information into route discovery packet. As a result, a node can quarantine a suspect node and prevent the suspect node to participate in routing. In addition, SPMP determines signal strengths of each pair of link before selecting a set of qualified routes and lets the source chooses the best-performance-path to be the primary path and the others to be the backup paths.

The SPMP basically consists of three phases. Firstly, nodes in a network execute authentication processes to make them become trusted nodes. The authentication can be done in either group or individual. The group authentication assures that all trusted nodes are in the same group. In the case, a shared series of passphrases is used for the group authentication. For the individual authentication, a node makes trust with the others as both a group member and a certified node. I explain the group authentication in this chapter and both group and individual authentication in the next chapter. Secondly, a source node discovers a set of secured routes to the destination in an on-demand basis. During the route discovery, the exchanged route information is secured by using data encryption and signing functions. Finally, communicating nodes perform end-to-end key exchange to build data secrecy as well as maintain linkage of current valid paths.

The design and implement of SPMP with group authentication relies on following assumptions:

1. Nodes in a network must occupy shared secret information, which is called "passphrases", and a node is trusted only if the node is able to correspond with the others using the same set of passphrases. The public key of the trusted node is also accepted by other nodes in the group. In another word, there is no malicious node among the group members.

2. The information of cancellation or appendage of shared passphrases is exchanged among nodes using location-limited channels (Balfanz, Smetters, Stewart and Wong, 2002) or off-line process which can be physical contacts, communication with human operation or secure communication channels. In the case, there is at least one information distribution center which is responsible to verify participants and dispense the shared set of passphrases to the trusted persons.

3. Each node is allowed to generate it own private/public keys. However, every node in the network has to use the same cryptographic algorithm.

All messages for the protocol are encapsulated with IP header.

## 5.2 Notations

This section presents the notation used throughout this chapter.

*Pass*: a passphrase used for building trust among nodes in a network.

$S_{Pass}$ : a shared set of passphrases.

$Pub_i$ : a public key of node *i*.

$Pri_i$ : a private key of node *i*.

$IP_i$ : an IP address of node *i*.

$TN_i$ : an abbreviation of trusted node *i*.

$L_{NN\text{-}i}$ : a list of neighbor nodes of node *i*.

$L_{TN\text{-}i}$ : a list of trusted nodes of node *i*.

*x*: total number of intermediate nodes in a path

*Seq*: a sequence number.

RSSI$_{Norm}$: normalized receive signal strength indicator.

PSSI: potential signal strength indicator.

Info$_{Route\text{-}i}$ : route information of node *i*, a sequence of IP address from source to node *i*.

Info$_{Target\text{-}i}$ : a set of target receiver nodes from node *i*.

Hash(message): a hash function for a message.

Shorten(Info$_{Route\text{-}i}$, $IP_i$): path shortening function executed by node *i*.

[message] $Pri_A$: node *A* signs a message with private key of node *A*.

$RDN_A$: random number that node *A* generates and uses for message signing.

<<message>>$Pri_A$: encryption of a message using private key of node *A*.

<message>*Key*: encryption of a message using symmetric key.

## 5.3 Associated Functions

The functions that are related to security handling for SPMP consist of the Hash Chain, the Message Signing and the Node Authentication.

### 5.3.1 Hash Chain

Hash chain can be described as a sequence of hash function working with a series of messages. I use hash chain for building trust between two communicating nodes that have shared secret messages. If *A* and *B* are communicating nodes and $Message_1$ and $Message_2$ are sequence of shared messages used by *A* and *B* to build trust between them. Since the shared secret messages should not be revealed to external entities, *A* and *B* exchange hashed secret messages instead of submitting secret messages. In addition, *A* and *B* can use their public keys as identities of themselves. When Hash() is the hash function that generates a hashed message, then *A* can build a trust to *B* by sending a hashed message of $Message_1$ combined with a random number and public key of *A* to *B*. I represent the hashed message submission by

$RDN_A$ = a random message generated by *A*

$$A \rightarrow B : \text{Hash}(Message_1, RDN_A, Pub_A), RDN_A, Pub_A$$

The reason of using $RDN_A$ as a part of message hashing is that the hashed message will always be dissimilar when the same message is hashed. This is useful for protecting a malicious node to abuse the previous hashed message. Though a malicious node can advertise the previous hashed message without the knowledge of the receiver, the malicious node does not have the private key which is used for route finding. As a result, the malicious node is still not allowed to be a participant node.

When *B* receives the hashed message from *A*, *B* trusts *A* because only *A* and *B* know $Message_1$. Consequently, *B* must response a message back to *A* to build a trust to *A* by replying *A* with hashed message of received message combined with $Message_2$ and public key of B to inform *A* that *A* is trusted by *B*. I represent this response by

$$B \rightarrow A : \text{Hash}(\text{Hash}(Message_1, RDN_A, Pub_A), Message_2, Pub_B), Pub_B$$

Finally, when *A* receives the response message from *B, A* verifies *B*'s response by

1. getting $Message_2$ which *A* know,
2. performing hash function for previous *A*'s hashed $Message_1$ combined with $Message_2$ and $Pub_B$ that *A* get from *B*,
3. comparing the result with incoming *B*'s hashed message.

The result of this process can be summarized that *A* trusts *B* in the same way that *B* trusts *A*.

### 5.3.2 Message Signing

Message signing is a method to ensure that a message has not been changed by any forwarders. Message signing utilizes asymmetric key mechanism and message digest function to build a signature that is attached with the submitted message.

Let *Message* is the message that node *A* wants to sign, then *A* can sign *Message* with the private key $Pri_A$ by

$$\text{Signature}_{\mathit{Message}} = \ll\text{Hash}(\mathit{Message})\gg \mathit{Pri}_A$$

$$[\mathit{Message}]\mathit{Pri}_A = \mathit{Message}, \text{Signature}_{\mathit{Message}}$$

A receiver who has $Pub_A$ can verify the message by decrypting the encrypted Hash(*Message*) with $Pub_A$. Consequently, the receiver activates hash function with *Message* and compares the Hash(*Message*). I represent the message verification by

$$\text{Result} = \begin{cases} \text{TRUE} & \text{if Hash}(\mathit{Message}) = \ll\text{Signature}_{\mathit{Message}}\gg\text{-}\mathit{Pub}_A \\ \text{FALSE} & \text{otherwise} \end{cases}$$

where "<< $\text{Signature}_{\mathit{Message}}$ >>-$Pub_A$" is the decrypted of signature using public key of node *A*.

### 5.3.3 Group Authentication

An assumption of SPMP with group authentication is that nodes must maintain a shared set of passphrases. I represent a set of *n* number of passphrases by

$$S_{Pass} = \{ Pass_1, Pass_2, \dots , Pass_i, \dots , Pass_n \}$$

For SPMP, every node in the network must authentication itself to be a group member before entering into the network. Initially, each node has two buffers; a list of trusted nodes and a list of neighbor nodes;

1. A list of trusted node information. I represent *m* number of trusted node information that is maintained by node *A* by

$$TN_i = (IP_i , Pub_i)$$

$$L_{TN\text{-}A} = \{ TN_1, TN_2, \dots , TN_i, \dots , TN_m \}$$

2. A list of neighbor nodes. I represent *k* number of neighbor nodes that is maintained by node *A* by

$$L_{NN\text{-}A} = \{ IP_1, IP_2, \dots , IP_i, \dots , IP_k \}$$

The authentication process is initiated by probing function. Every node in the network broadcasts dedicated probe packets for every certain period. For instance, node *A* launches a probe packet

$$A \rightarrow \text{broadcast} : [\ Seq\ ]\ Pri_A$$

When receiving the probe packet, the receiver *B* checks whether the incoming probe packet is from the node which *B* trusted. If *B* has never trusted *A* then *B* starts authentication process otherwise *B* updates the $L_{NN\text{-}B}$ by.

$$L_{NN\text{-}B} = L_{NN\text{-}B} \cup \{\ IP_A\ \}$$

The group authentication process of each pair of nodes is achieved by using passphrase request-response scheme. When *A* authenticates with *B*, the negotiating nodes perform following steps:

Steps of Group Authentication:

Step 1: When *n* is the number of passphrase, *A* picks up a number *i* where $1 \le i \le n$. Then *A* sends a request to *B* by

$$RDN_A = \text{a random message generated by } A$$
$$A \rightarrow B : \ \text{Hash}(Pass_i, RDN_A, Pub_A), i, RDN_A, Pub_A$$

There are two reasons that *A* uses $RDN_A$ and $Pub_A$ to be a part of message for hash function. First the $RDN_A$ make decoding of the passphrase become harder because the hashed message is always different every time *A* authenticate itself. Second, $Pub_A$ can be protected from forge since change in $Pub_A$ will be checked by rehashing the message.

Step 2: *B* verifies the incoming request by getting $Pass_i$ which *B* know, performing hash function for $Pass_i$ combined with $RDN_A$ and $Pub_A$ and comparing the result with incoming hashed message. If the hashed message is correct then *B* trusts *A* and $B$ realizes that $Pub_A$ is belonging to $A$. Consequently, $B$ calculates $j = i + 1$ if $j \leq n$ otherwise $j = 1$ and responses a message back to $A$ by

$$B \rightarrow A: \text{ Hash(Hash(}Pass_i, RDN_A, Pub_A,), Pass_j, Pub_B), Pub_B$$

Notice that $B$ executes hashing chain function explained in 5.3.1 in order that $A$ can verify $B$'s reply. If $A$ trusts $B$ and $A$ accepts $Pub_B$ as $B$'s public key then $A$ and $B$ find that the negotiated nodes are trust nodes. Both $A$ and $B$ append the new trusted node information, which consist of the IP address got from IP packet header and the public key, to $L_{TN\text{-}A}$, $L_{NN\text{-}A}$, $L_{TN\text{-}B}$, and $L_{NN\text{-}B}$ by

$$TN_B = (IP_B\ ,\ Pub_B)$$
$$L_{TN\text{-}A} = L_{TN\text{-}A} \cup \{\ TN_B\ \}$$
$$L_{NN\text{-}A} = L_{NN\text{-}A} \cup \{\ IP_B\ \}$$

$$TN_A = (IP_A\ ,\ Pub_A)$$
$$L_{TN\text{-}B} = L_{TN\text{-}B} \cup \{\ TN_A\ \}$$
$$L_{NN\text{-}B} = L_{NN\text{-}B} \cup \{\ IP_A\ \}$$

## 5.4 Routing Protocol

The route discovery process, which is activated in on-demand basis, aims to find a set of secured disjoint-paths that have maximum ETV. The route discovery of the SPMP comprises of two main functions: the route requisition and route establishment.

### 5.4.1 The Route Requisition

When a source $S$ cannot find a route to the destination $D$, $S$ starts up route discovery process by launching a signed Route Requisition (RREQ) message that contains public key, source route information and target neighbors' IP address such that

$$\text{Info}_{Route\text{-}S} = \{ IP_S \}$$

$$\text{Info}_{Target\text{-}S} = L_{NN\text{-}S}$$

$$S \rightarrow \text{broadcast} : [\ Seq, Pub_S, IP_D, \text{Info}_{Route\text{-}S}, \text{Info}_{Target\text{-}S}\ ]Pri_S$$

A neighbor node *A*, when receiving a RREQ, must make a decision either to drop or forward the RREQ. Since the network performance will be dramatically depreciated from overwhelming unnecessary RREQ floating in the network, I introduce a rule set that a node is allowed to drop the incoming RREQ when:

1. The incoming message does not pass the verification process that *A* uses the public key (for example, $Pub_S$) that is obtained during authentication process to check the signed message of *S*. *A* will drop the message if *S* is an untrusted node or the change in the message is found.

2. *A* found that the *Seq* of received RREQ is duplicated with the one that is received in the past. I explain from this rule that, the protocol guarantees minimum RREQ forwarding by allowing a node forwards only one RREQ route request.

3. *A* is not a target node (in the case, *A* is not an element of $\text{Info}_{Target\text{-}S}$). This means only nodes that exist in the $\text{Info}_{Target\text{-}S}$ are allowed to forward the RREQ.

Finally, *A* reforms and forwards the RREQ by

$$\text{Info}_{Route\text{-}A} = \text{Info}_{Route\text{-}A} \cup \{ IP_A \}$$

$$\text{Info}_{Route\text{-}A} = \text{Shorten}(\text{Info}_{Route\text{-}A}, IP_A)$$

$$\text{Info}_{Target\text{-}A} = \{\forall\ IP_i \in L_{NN\text{-}A}, \text{ if } IP_i \notin \text{Info}_{Route\text{-}A} \}$$

$$A \rightarrow \text{broadcast} : [\ Seq, Pub_S, IP_D, \text{Info}_{Route\text{-}A}, \text{Info}_{Target\text{-}A}\ ]Pri_A$$

### 5.4.2 Route Establishment

After broadcasting the first RREQ, the source node waits for a certain period in order to receive a number of Route Reply (RREP) messages from the destination. If the source does not find any replies, it reattempts to find routes by launching another RREQ. The node terminates the route discovery process when launching a number of RREQs without receiving any RREP.

The route establishment process allows only a destination to build the RREP in order that a set of qualified routes can be generated. The destination has a certain period after arrival of the first RREQ to collect a number of qualified routes and return the RREPs back to the source. After collecting a set of RREQs, the destination performs following steps:

1. Rejects the incoming RREQ that do not pass the verification process, which complies with the rules of dropping RREQ explained in the last section.
2. Executes path shortening function that I described in the Chapter 3.
3. Eliminates a route that has an overlapped path with the previous ones.
4. Ranks the routes. The greatest value of ETV is determined to be the first rank. If there are paths that have equal ETV, the path that has earlier arrival time is judged to have higher priority.

The accepted path is reported back to the source by Route Reply (RREP) packets using the rank number as sequence number. To limit the number of paths maintained by associated nodes, the destination may set the maximum number of RREP and ignore the incoming RREQ when the sequence number reaches the threshold. In addition, the destination has a certain period after the arrival of the first RREQ to return RREPs back to the source. The destination will drop all following RREQs if the waiting period exceeds a threshold. Since routes reporting requires data availability and secrecy, the destination and intermediate nodes are allow only unicast transmission in sending the RREP among intermediate nodes.

Let *D* is the destination node, *D* build non-repudiation of route information by signing the $\text{Info}_{Route}$ and the sequence number. The secrecy of route information is maintained by encrypted the whole reply message with public key of target receiver. *D* sends a RREP to the adjacent node by

$$\text{Info}_{Route\text{-}D} = \{ IP_S, \ldots, IP_{inter\text{-}i}, \ldots, IP_{inter\text{-}x}, IP_D \}$$
$$D \rightarrow inter\text{-}x : \ll [\text{ Seq}, Pub_D, \text{Info}_{Route\text{-}D} ]\, Pri_D \gg Pub_{inter\text{-}x}$$

The intermediate node *i* keeps route information into the route table and forwards the RREP to *i-1* by

$$inter\text{-}i \rightarrow inter\text{-}i\text{-}1 : \ll [\text{ Seq}, Pub_D, \text{Info}_{Route\text{-}D} ]\, Pri_D \gg Pub_{inter\text{-}i\text{-}1}$$

When the source *S* receives a set of RREPs, *S* ranks the paths by ETV and uses the highest priority route as the primary route to deliver data packets and the others as the backup routes. If the primary route fails, the second priority backup route is then activated. The source node will execute route discovery processes again when all routes become invalid.

### 5.4.3 Data Transfer and Forward

Since encryption with asymmetric key algorithms consumes high computing cost, I proposed that end-to-end data exchange should be encrypt with a share key that is compromised between source and destination. In the case, *S* can exchange a share key with *D* during the event of exchanging the first chunk of data by

$\text{Info}_{Route\text{-}S} = \{\, IP_S, \ldots, IP_{inter\text{-}i}, \ldots, , IP_{inter\text{-}x}, IP_D \,\}$

$\text{Key}_{encryted} = \ll[\, Key \,]\, Pri_S \gg Pub_D$

$S \rightarrow inter\text{-}1 : \ll[\, \text{Info}_{Route\text{-}S}, \text{Key}_{encryted}]\, Pri_S \gg Pub_{inter\text{-}1}, \text{<Data>}Key$

$inter\text{-}i \rightarrow inter\text{-}i{+}1 : \ll[\, \text{Info}_{Route\text{-}S}, \text{Key}_{encryted}]\, Pri_{inter\text{-}i} \gg Pub_{inter\text{-}i+1},$
$\text{<Data>}Key$

$inter\text{-}x \rightarrow D : \ll[\, \text{Info}_{Route\text{-}S}, \text{Key}_{encryted}]\, Pri_{inter\text{-}x} \gg Pub_{inter\text{-}D},$
$\text{<Data>}Key$

Notice that the first chunk of data may be encrypted and transmitted together with encrypted symmetric key. For sending the next chunk of data after symmetric key exchange, *S* encrypted and transmitted the data as follows:

$\text{Info}_{Route\text{-}S} = \{\, IP_S, \ldots, IP_{inter\text{-}i}, \ldots, , IP_{inter\text{-}x}, IP_D \,\}$

$S \rightarrow inter\text{-}1 : \ll[\, \text{Info}_{Route\text{-}S} \,]\, Pri_S \gg Pub_{inter\text{-}1}, \text{<Data>}Key$

$inter\text{-}i \rightarrow inter\text{-}i{+}1 : \ll[\, \text{Info}_{Route\text{-}S}]\, Pri_{inter\text{-}i} \gg Pub_{inter\text{-}i+1}, \text{<Data>}Key$

$inter\text{-}x \rightarrow D : \ll[\, \text{Info}_{Route\text{-}S}]\, Pri_{inter\text{-}x} \gg Pub_{inter\text{-}D}, \text{<Data>}Key$

An important consideration of SPMP is that all compromised node must maintain secrecy of source route information. Therefore, source route information in a data packet must be signed by the sender and encrypted by public key of the next hop.

### 5.4.4 Route Maintenance

Route maintenance process associates with how participating nodes report errors when they find broken links between themselves and their neighbors. The broken link may be caused by movement of nodes in network or malfunction of communication equipment of a node. In the case, the node generates a route error packet (RERR) that contains broken link information and reports to the participating nodes.

According to the protocol, a path error is detected from two events. The first event happens when a node does not receive a probe packet from its neighbor for a certain period. The second event emerges when a node receives failure information from data link layer during data transmission to the next hop. When node *A* receives failure information for the link to *X*, *A* performs following procedures:

1. Finds a spare path to the destination in the route table. If a backup path is found, then the node uses this path as a new active path by put the new source route information into the sending packet and go to next step. Otherwise, goes to next step.

2. Generates a route error packet (RERR) that contains broken link information and report back to the source. If node *B* and *S* are associated nodes of the error path, then the RERR is sent from *A* to *B* and from *B* to *S* by

$$A \rightarrow B : [\ \mathrm{Seq}, IP_X\ ]\ Pri_A$$

$$B \rightarrow S : [\ \mathrm{Seq}, IP_X\ ]\ Pri_B$$

Finally all associated nodes eliminate all inactive paths from their route table.

## 5.5 Security Analysis

Since SPMP purposes to maintain security for a wireless network, the protocol must be endurable from various kinds of attacks. Generally, each type of attack focuses on penetrating specific area of weakness of network security. Attacks of wireless network can be viewed as passive attacks, active attacks and attacks from compromised nodes.

### 5.5.1 Protection from Passive Attacks

In passive attack, an intruder traps valuable information by listening to traffic without disrupting the network. Passive attacks can be categorized into two types: data stealing and topology disclosure.

For data stealing, an intruder aims to intercept valuable data during data exchange in a network. As a result, data secrecy plays a major role to defense against this kind of attack. The SPMP protects this kind of attack by adopting end-to-end data encryption to prohibit external entities read data without a valid key.

For topology disclosure, an intruder intends to reveal the topology of the network. The SPMP protects the information of structure of the network by providing source route information encryption for RREP and data packet that only associated trusted nodes are allowed to read this information.

### 5.5.2 Protection from Active Attacks

Active attack associates with injection of arbitrary packets into a network to disrupt the operation of the network, limit network availability, gain authentication or attract packets destined to other nodes. For security analysis, a number of major active attacks including black hole, man-in-the- middle, replay attack and denial of service are investigated.

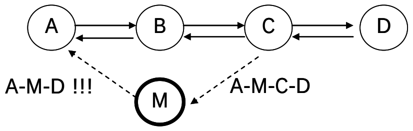

**Figure 5.1** The Black Hole Attack

Black hole attack is an attack that aims to make a malicious node to be a compromised node. In the case, an attacker advertises a zero metric for all destinations causing all nodes around it to route packets towards it. This attack can be performed by counterfeiting RREQ or RREP. As illustrated in figure 5.1, when malicious node M is actually compromising in the path A-M-C-D but M wants to be a compromised node by reporting false information to A to deceive A that path A-M-D is a qualified path. However, this issue cannot happen because the SPMP has authentication process for each node. Any route requisition packets from untrusted nodes will be ignored. In addition, changing in route information in a RREP is prohibited because the information is signed by the destination.

In protecting man-in-the-middle attack, SPMP allows a node to be a compromised node only if the node is found a trusted node. All packets launched by a node that do not pass authentication process will be ignored. Moreover, the SPMP requires source node to encrypt and sign messages. Hence, the malicious node is not allowed to change any data in a packet.

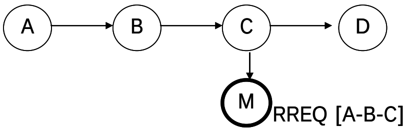

**Figure 5.2a** Replay Attack During Route Request Period 1, Node M Keeps RREQ Information that Received from C.

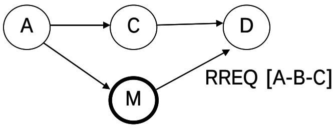

**Figure 5.2b** Replay Attack During Route Request Period 2, Node M Pretends to be Node C and Submits the Old RREQ to Node D.

Replay attack is an active attack when an intruder sends old advertisements to a node causing it to update its routing table with stale routes. As shown in figure 5.2a and 5.2b, the malicious node M keeps a RREQ packet from node C and pretends to be node C by submitting the old RREQ to D. For the SPMP, every packet has incremental sequence number and the whole data is signed. The replay attack is protected because every node is instructed to reject all packets, which sequence number is less than the maximum sequence number of the packet the node received in the past. In addition, for data packet, D can detect the counterfeit data issued by M since M is not the participant node of the path.

For denial of service (DoS), an intruder injects a large number of packets with expectation that the network will collapse. Typically, the DoS attack can be executed in the data link layer and the DoS attack in the data link layer is hard to protect. However, for the DoS attack in the network layer, this kind of attack will never be achieved since a malicious node fails to authenticate itself during self identification process and all packets launched by a malicious node will be rejected.

### 5.5.3 Protection from Compromised Nodes

Attacks that are the hardest to protect are attacks from a compromised node. Typically, attacks from compromised nodes derive from two purposes. First, a trusted node needs to be a compromised node of a path or wants to repartition the network. To achieve this purpose, the trusted node might perform routing table poisoning by advertising incorrect routing information to other nodes. The second purpose is that a node wants to destroy existing communication by reporting counterfeit route errors to associated nodes.

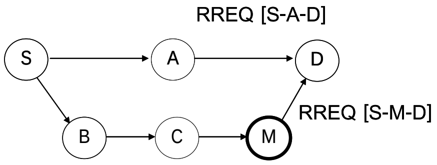

**Figure 5.3a** An Attack That a Malicious Node Counterfeits a RREQ with Only One Hop-count from a Source

For SPMP, a malicious trusted node attempts to be a compromised node with two actions. First, the node may counterfeit a RREQ with only one hop count from a source to deceive the neighbors with an expectation that the node is chosen to be a compromised node. As illustrated in the figure 5.3a, M is a malicious node which receives a RREQ from C. Then, M counterfeits the RREQ by shorten path to D. This event can be achieved only if M is a trusted node. However, the achievement cannot be guaranteed because an intermediate node forwards only the first received RREQ and ignores the later coming RREQ and the destination has a limited time to collect a number of RREQ before replies the chosen routes.

Second, a malicious trusted node need collaboration with other trusted nodes to do tunneling attack.

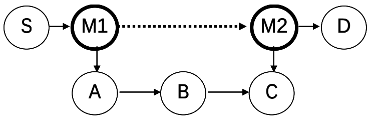

**Figure 5.3b** Tunneling Attack

Tunneling attacks is an attack that a compromised node collaborates with one or more nodes to misrepresent path lengths from source and destination and make them to be parts of available path. As a matter of fact, tunneling attacks is a threat to all multipath routing protocols which rely on maximally disjoint paths. The figure 5.3b illustrates tunneling attack during route discovery where M1 joins with M2 to report wrong routing information to D. When receiving a RREQ from S, M1 encapsulated the RREQ as if it is data of M1 and sends it to M2 via A B and C. When M2 receives the encapsulated data, M2 decapsulates the RREQ, attaches its routing information as if the RREQ was passed from M1 and forwards to D. After path correction period, D misunderstands that the path S-M1-M2-D is the best path and send the RREP back to M2. M2 then tunnels the RREP back to M1 in order that M1 can forward the RREP to S. Fortunately, SPMP can relieve this threat because the protocol allows a node forwards only the first coming RREQ and allows path collection only a limited of time. As a result, a long-delay packet will be discarded by an intermediate node.

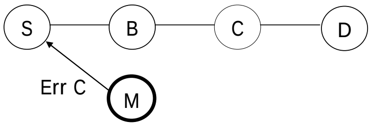

**Figure 5.4** Counterfeit Route Error Information Launched by Compromised Node M

In the case of launching RERR to destroy existing route, as depicted in figure 5.4, a compromised node M might generate a counterfeit route error and report to participant node S. In the sense of group member trust, this misleading will be caught by the receiver because every RERR must be encrypted and signed by private key of sender. Before accepting the error information, the node S can easily check whether node M is a member node of the path from S to D. The node S must reject the error message. However, the SPMP cannot protect an attack when the compromised node, which is an associated node of the path, advertises a wrong RERR to other member nodes of the path.

## 5.6 Simulation and Result Analysis

An objective of simulation was to investigate impact of network performance caused from lengthened data packet and additional security processes. Since, the performance of PMP was evaluated in the chapter 3, I worked on performance comparison only between SPMP and PMP.

In my simulation, each node occupied 1024 bits RSA private and public key. All cryptographic functions with asymmetric key relied on RSA algorithm. Data were encrypted and decrypted with shared key using triple DES.

### 5.6.1 Environments

The performance of SPMP and PMP were simulated using Jist-SWANS. I simulated 1000 square meter field where 50 mobile nodes were moving with various speed. The radio signal propagation was set to approximate 250 meters ranges with

IEEE 801.11b Distribution Coordination Function (DCF). The radio propagation in the network has random signal-to-noise ratio up to 10 percent.

Among nodes in the field, I randomly selected 10 pair of clients and servers to transmit constant bit rate (CBR) 512 bytes of packet for every 200 millisecond. The simulation was executed for 300 seconds which measurements were recorded every 30 seconds. Every node broadcasted a probe packet for every one second. The clients started sending data after 30 seconds initial period to allow every node launched a number of probe packets until the network was stable. Since the computational times regarding forecasting model, data encryption and decryption of the SPMP vary by the capability of the processors and other resources of nodes, I randomly defined delay time up to 10 milliseconds before transmitting data.

### 5.6.2 Performance Measurements

There were four metrics used for the simulation consists of:

1. Packet delivery ratio: The ratio of the number of data packets completely received by the destination to the number of data packets generated by the source.

2. Average goodput: The average speed of real data transmission from source to destination. A goodput can be explained as the number of useful bit per unit of time forwarded by the network from a source to a destination, excluding protocol overhead, and excluding retransmitted data packets. The calculation of a goodput can be done by dividing total number of data bits of data completely received by the destination with total delay time of data delivery from source to destination.

3. Percent of dropped packet: The number of data packets those are dropped during data transmission process to the number of total packets.

4. The number of cumulative route request: The number of RREQ packets established and launched by source nodes. The number of route request is used to measure the robustness of a path and informs how often source nodes initialized route request process.

### 5.6.3 Simulation Results

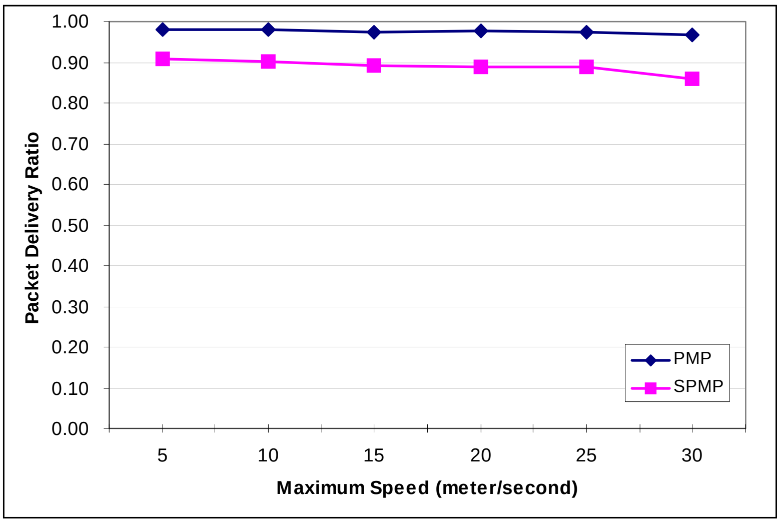

**Figure 5.5a** Packet Delivery Ratio of SPMP and PMP at Maximum Random Speed from 5 to 30 m/s

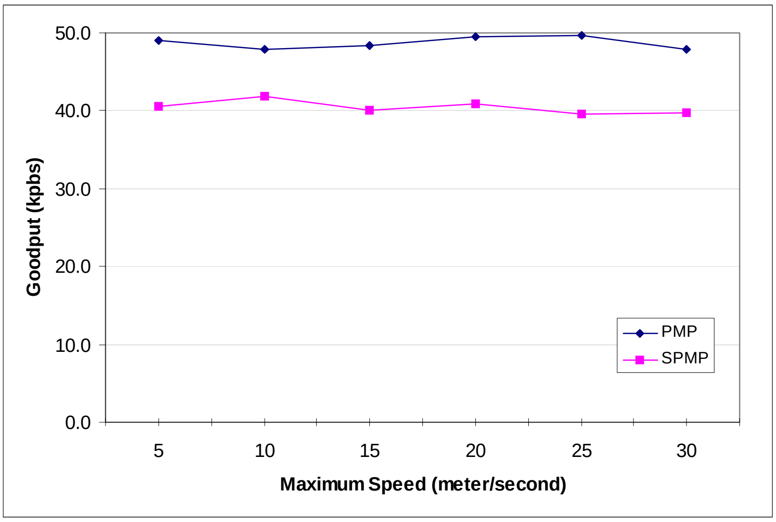

**Figure 5.5b** Average Goodput of SPMP and PMP at Maximum Random Speed from 5 to 30 m/s

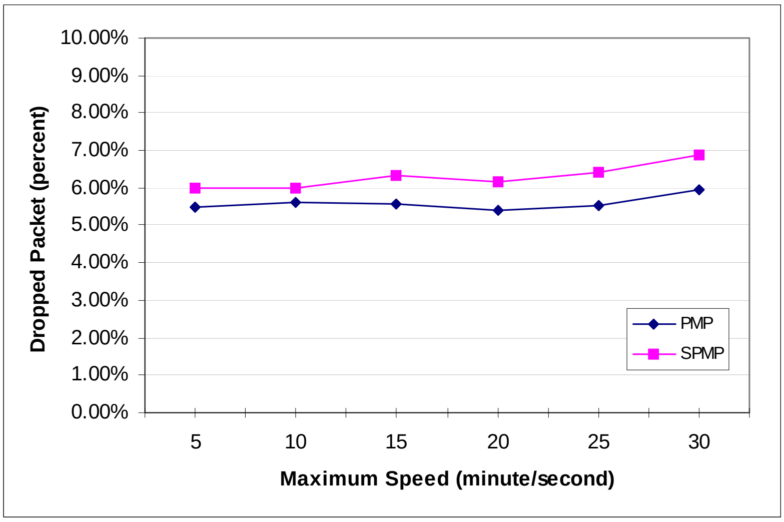

**Figure 5.5c** Percent Dropped Packet of SPMP and PMP at Maximum Random Speed from 5 to 30 m/s

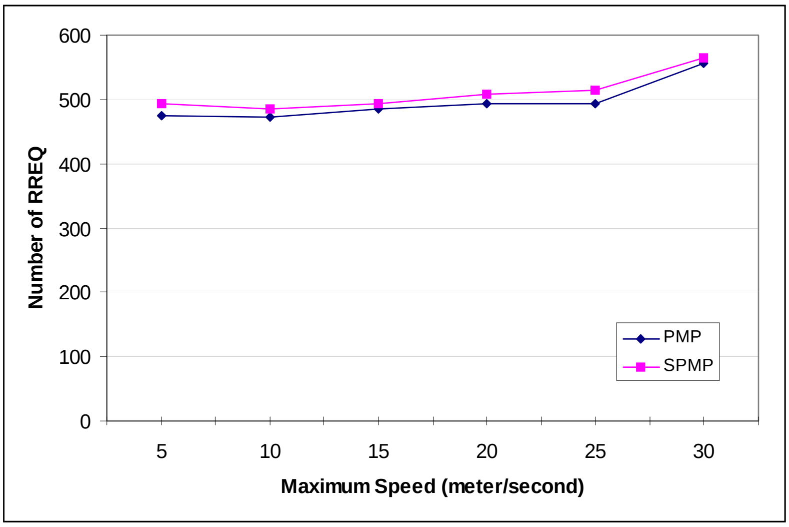

**Figure 5.5d** Number of Route Request Packet That are Initially Generated by Source for SPMP and PMP at Maximum Random Speed from 5 to 30 m/s

Figure 5.5a – 5.5 d show performance comparisons between SPMP and PMP for various maximum random speeds. In figure 5.5a, the packet delivery ratios of both PMP and SPMP are slightly dropped when maximum random speed increases. The result shows that the output of PMP is still higher than those of SPMP approximately 4% to 9%. The main reason is that data encryption lengthens the packet to 20% and causes deficiency in packet delivery.

The longer packet also affect higher dropped packet ratio. As shown in figure 5.5c, percent dropped packet of SPMP tends to be greater than those of PMP. The increasing in maximum random speed causes a little higher dropped packet ratio. I found from figure 5.5a and 5.5c that both protocols still work well in a moderate speed mobile network.

As shown in figure 5.5b, the average goodputs of SPMP are in between 35 to 45 kbps while PMP has better rates in between 45 to 50 kbps. This phenomenon can be explained that the SPMP has higher protocol overhead especially longer size of packet caused from data encryption. However, the efficiency of SPMP dropped only 7% when compared with that of PMP. This event shows that overheads from security of SPMP do not significantly harm the performance of the network.

In figure 5.5d, events of finding new routes of PMP and SPMP tend to increasingly occur when the maximum random speed gets higher. Though both SPMP and PMP use the same route discovery algorithm, a path from PMP is slightly more robust than a path acquired by SPMP. I found that the security overhead of SPMP causes a longer delay period and increasing in data transmission time raises possibility of path invalidation.

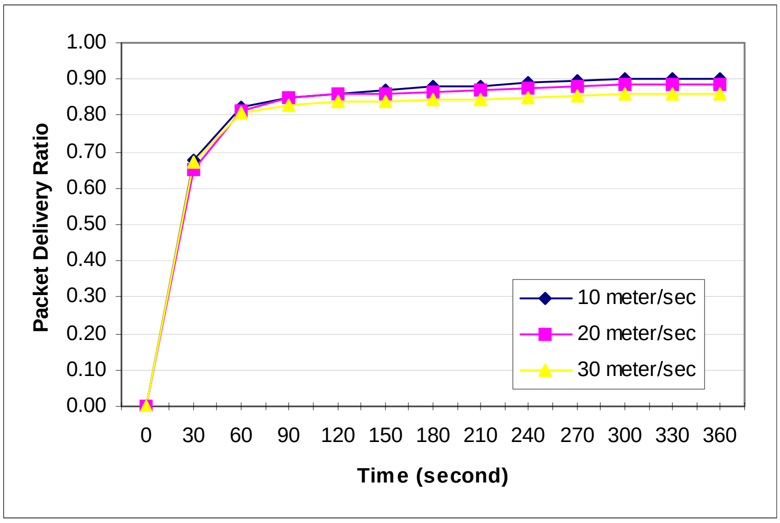

**Figure 5.6a** Packet Delivery Ratio of SPMP for Various Speed for Every 30 Seconds

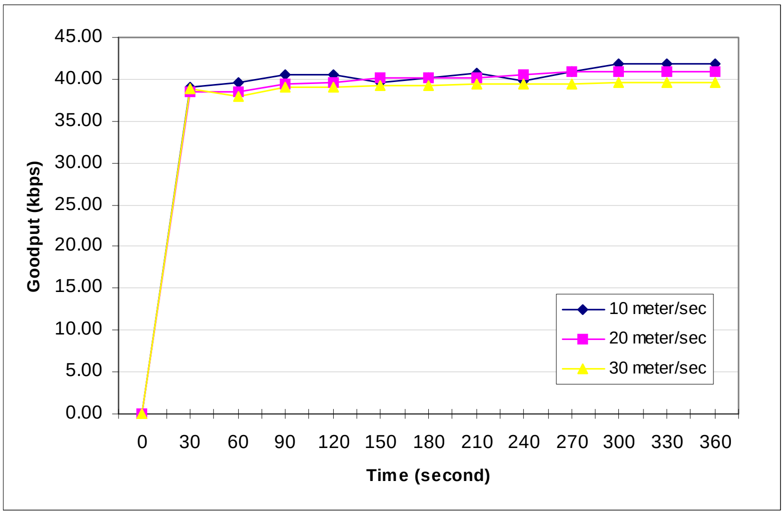

**Figure 5.6(b)** Average Goodput of SPMP for Various Speed for Every 30 Seconds

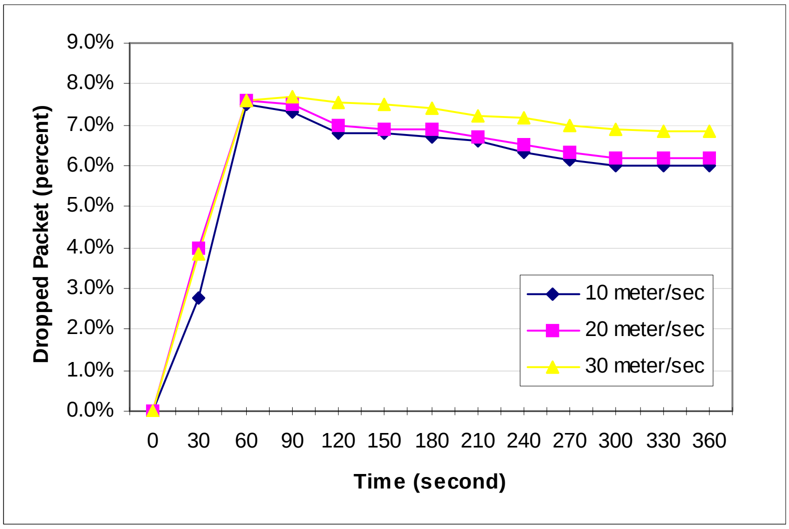

**Figure 5.6c** Percent Dropped Packet of SPMP for Various Speed for Every 30 Seconds

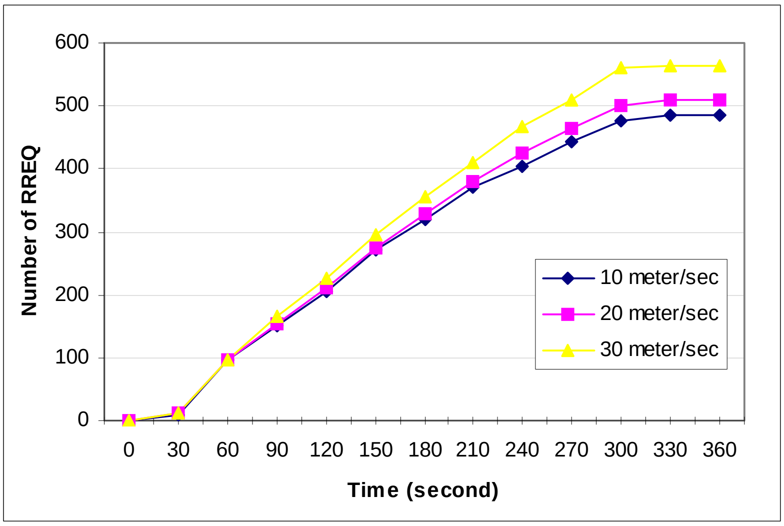

**Figure 5.6d** Number of Cumulative Route Request Packet That are Initially Generated by Source of SPMP for Various Speed for Every 30 Seconds

Figure 5.6a – 5.6d depict the performance of SPMP with different maximum random speed for each 30-second pause time. In figure 5.6a, the packet delivery ratio drop within a narrow range when maximum random speed increased. The dropping rates are approximately 0.5% to 1% that is not significant. As shown in figure 5.6b, the range of average goodputs is between 38 and 42 kbps. According to the results, the increasing in mobile speed does not make any sense with average goodputs because of some reasons. First, the performance of a route (ETV) is controlled by the protocol when a route is established. Second, in the simulation environment, direction of locomotion of a node was random and there was equal possibility for nodes in network to be apart or closer. Finally, a goodput counts only data packets that are completely delivery from sources to destinations and ignores all data packets that are launched by sources but do not reach destinations. Therefore, probability of broken route does not have any impact with the goodput. In the figure 5.6c, the percent of dropped packets has a tendency to increase when mobility speed of a node increase. This is normal because the link between two nodes is easier to be broken under higher mobility speed circumstance. The increasing in speed of mobility nodes also shortens the life of a route. As shown in the figure 5.6d, the number of route request packet that is initially generated by source tends to be increase when the speed is raised.

According to my observation, I conclude that the SPMP establishes a set of routes that are robust and have efficiency enough to handle moderate speed mobile network. The simulation results show a little decreasing in route performances that are acceptable.

# CHAPTER 6

# THE SECURED PREDICTIVE MULTIPATH ROUTING PROTOCOL (SPMP) WITH INDIVIDUAL AUTHENTICATION

## 6.1 Overview

A disadvantage of SPMP with group authentication explained in the chapter 5 is a limitation that a node trusts the others only as a group member but not individually. Hence, I introduce an extension of SPMP with group authentication, called SPMP with individual authentication, in order that a node can make trusts with the others not only in group-based but also individually. Basically, the SPMP with individual authentication needs certification from a central trusted server. For the SPMP with individual authentication, each node is allowed to perform individual authentication as well as group authentication to make trust to the other nodes in both group member and individual view. Like the group-based authentication that was mentioned in the last chapter, the individual authentication is made by each pair of nodes. In this chapter, I introduce a method of individual authentication of SPMP and simulation results.

The design and implement of SPMP with individual authentication is based on following assumptions:

1. Nodes in a network must share secret information, which is called “passphrases”, and a node is trusted only if the node is able to correspond with the others using the same set of passphrases.

2. The protocol requires a central trusted server to distribute the set of passphrases and issue certificates to requester. Passphrases distribution and certification issuing can be executed by an off-line process.

3. Nodes in the network are capable to generate there own private/public keys but all nodes has to use the same cryptographic algorithm.

4. All nodes must have the public key of the trusted server or the public key of the trusted server is generally known.

All messages for the protocol are encapsulated with IP header.

## 6.2 Notations

The following notations will be used throughout this chapter.

$Cert_i$ : a certificate of node *i*.

$Pub_{ts}$ : a public key of the trusted server.

$Pri_{ts}$ : a private key of the trusted server.

*Pass*: a passphrase used for building trust among nodes in a network.

$\mathrm{S}_{Pass}$ : a shared set of passphrases.

$Pub_i$ : a public key of node *i*.

$RDN_A$ : a random number generated by *A*.

$TN_i$ : an abbreviation of trusted node *i*.

$L_{NN\text{-}i}$ : a list of neighbor nodes of node *i*.

<<message>>$Pri_{ts}$ : encryption of a message using private key of the trusted server.

<message>*Key*: encryption of a message using symmetric key.

## 6.3 Individual Authentication

The individual authentication of SPMP requires that nodes in the network must request for a certificate from a trusted server. Initially, a node generates its own public and private key. The public key is sent to the trusted server to make a certificate. Certification issuing by the trusted server *TS* to node *A* is represented by

$$Cert_A = \ll Pub_A \gg Pri_{ts}$$

$$TS \rightarrow A : Cert_A$$

where $Cert_A$ is the certificate of $A$, $Id_A$ is the identity of $A$ e.g. IP address of $A$, $Key$ is a symmetric key, $Pri_{ts}$ is the private key of the trusted server and $Pub_A$ is the public key of $A$.

Besides maintaining a certificate, a node must keep a shared set of passphrases issued by the trusted server. The set of $n$ number of passphrases is represented by

$$\mathrm{S}_{Pass} = \{ Pass_1, Pass_2, \ldots , Pass_i, \ldots , Pass_n \}$$

A node can make trust to its neighbor by performing individual authentication. For example, node $A$ and $B$ can make trust to each other by exchange their certificates and hashed passphrases. When $A$ authenticates with $B$, the negotiating nodes perform following steps:

Step 1: If $n$ is the number of passphrase, $A$ picks up a random number $i$ where $1 \leq i \leq n$. Then $A$ sends a request to $B$ by

$$RDN_A = \text{a random message generated by } A$$
$$Cert_A = \ll Pub_A \gg Pri_{ts}$$
$$A \rightarrow B: \ \mathrm{Hash}(Pass_i, RDN_A), i, RDN_A, Cert_A$$

$A$ attaches $RDN_A$ with the passphrase before hashing the whole message to dissimilate the hashed message. Therefore, messages hashed by this process will always be different in order that the passphrase cannot be easily decoded.

Step 2: $B$ verifies the incoming request by getting $Pass_i$ which $B$ know, performing hash function for $Pass_i$ combined with $RDN_A$ and comparing the result with incoming hashed message. If the hashed message is correct then $B$ trusts $A$ as $B$'s group member.

To verify $Cert_A$, $B$ decrypts $\ll Pub_A \gg$ by the public key of the trusted server ($Pub_{ts}$). Then $B$ can read $Pub_A$. $B$ trusts that the $Cert_A$ is issued by the trusted server $TS$ because only $TS$ can encrypt this message. Consequently, $B$ calculates $j = i + 1$ if $j \leq n$ otherwise $j = 1$ and responses a message back to $A$ by

$$Cert_B = << Pub_B >> Pri_{ts}$$

$$B \rightarrow A: \text{ Hash(Hash(}Pass_i, RDN_A\text{)}, Pass_j\text{)}, Cert_B$$

Notice that *B* executes hashing chain function in order that *A* can verify *B*'s reply to ensure that *B* is A's group member. In addition, *A* can verify $Cert_B$ by using the same methods that *B* did. Finally, *A* and *B* keep their neighbor's public key in the trusted node list used for routing protocol.

For SPMP, every node in the network maintains a list of trusted nodes. The list of trusted nodes comprises of address and public key of each trusted node. I represent *m* number of trusted node information that is maintained by node *A* by

$$TN_i = (IP_i , Pub_i)$$

$$L_{TN\text{-}A} = \{ TN_1, TN_2, \dots , TN_i, \dots , TN_m \}$$

where $TN_i$ is the information of trusted node *i*, $IP_i$ is the address of node *i*, $Pub_i$ is the public key of node *i* and $L_{TN\text{-}A}$ is a set of trusted nodes.

## 6.4 Routing Protocol and Security Analysis

The only difference between SPMP with group-based and individual authentication is the authentication methods. Both of them use the same routing protocol that I described in the last chapter. Therefore, I refer the detail descriptions of routing protocol to the section 5.4 of the chapter 5.

As the matter of fact, the authentication methods are not involving with establishment of secured routing and data exchange. The aspects of routing security are explicitly obtained by the functions of routing protocol. Therefore, I refer the security analysis to the section 5.5 of the chapter 5.

## 6.5 Simulation and Result Analysis

A main objective of simulation was to learn the difference of network performance between two authentication methods of the SPMP. Thus, I run a simulation to investigate network performances of the SPMP with individual authentication compared with those of the SPMP with group authentication.

In my simulation, asymmetric key encryption/decryption relied on RSA algorithm with 1024 bits private and public key. For the symmetric key encryption, I use triple DES.

### 6.5.1 Environments

The performance values of SPMP are simulated using Jist-SWANS. I simulated 1000 square meter field where 50 mobile nodes were moving with various speed. The radio signal propagation was set to approximate 250 meters ranges with IEEE 801.11b Distribution Coordination Function (DCF). The radio propagation in the network has random signal-to-noise ratio up to 10 percent.

Among nodes in the field, I randomly selected 10 pair of clients and servers to transmit constant bit rate (CBR) 512 bytes of packet for every 200 millisecond. The simulation was executed for 300 seconds which measurements were recorded every 30 seconds. Every node broadcasted a probe packet for every one second. The clients started sending data after 30 seconds initial period to allow every node launched a number of probe packets until the network was stable. Since the computational times regarding forecasting model, data encryption and decryption of the SPMP vary by the capability of the processors and other resources of nodes, I randomly defined delay time up to 10 milliseconds before transmitting data.

### 6.5.2 Performance Measurements

There were four metrics used for the simulation consists of:

1) Packet delivery ratio: The ratio of the number of data packets completely received by the destination to the number of data packets generated by the source.

2) Average goodput: The average speed of real data transmission from source to destination. A goodput can be explained as the number of useful bit per unit of time forwarded by the network from a source to a destination, excluding protocol overhead, and excluding retransmitted data packets. The calculation of a goodput can

be done by dividing total number of data bits with number of data transmission packet.

3) Percent of dropped packet: The number of total data packets of every node those are dropped during data transmission process to the number of total data packets floating in the network.

4) The number of cumulative route request: The number of RREQ packets established and launched by source nodes. The number of route request is used to measure the robustness of a path and informs how often source nodes initialized route request process.

### 6.5.3 Simulation Results

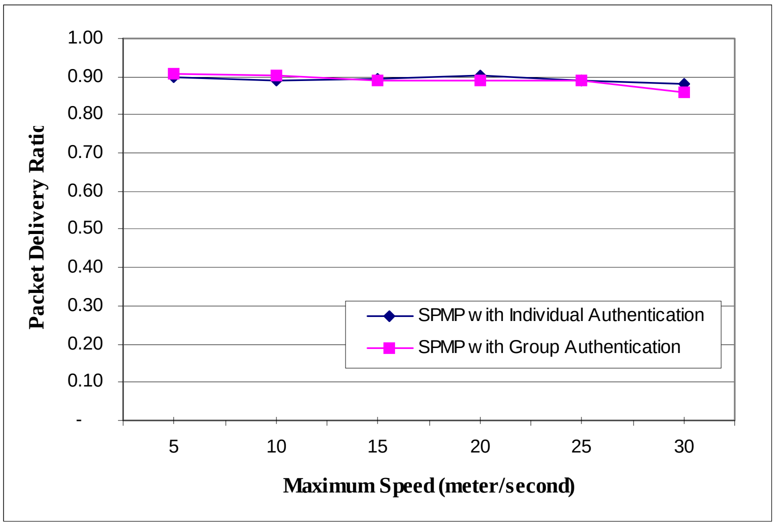

**Figure 6.1** Packet Delivery Ratios of SPMP with Individual Authentication and SPMP with Group Authentication at Maximum Random Speed from 5 to 30 m/s

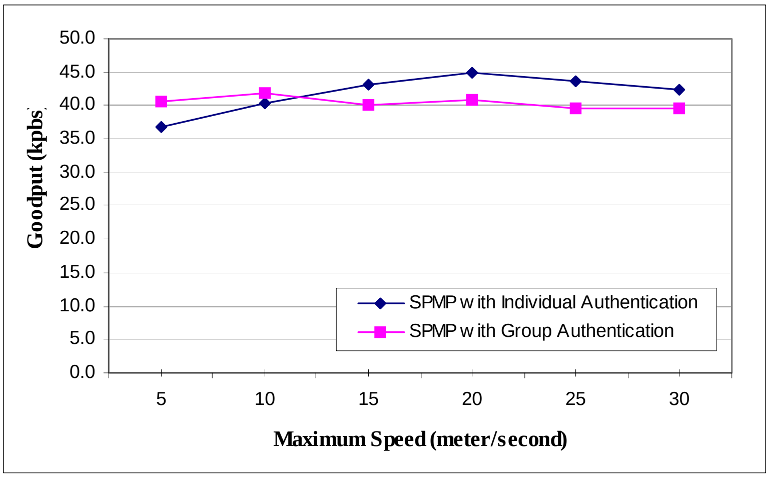

**Figure 6.2** Average Goodput of SPMP with Individual Authentication and SPMP with Group Authentication at Maximum Random Speed from 5 to 30 m/s

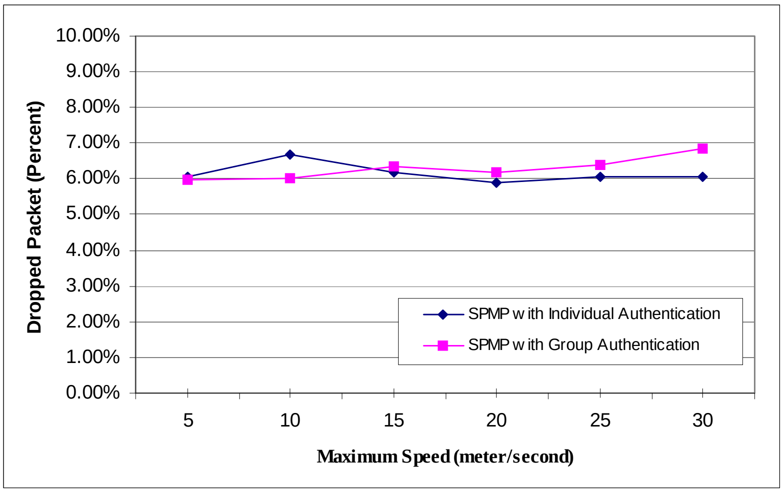

**Figure 6.3** Percent Dropped Packet of SPMP with Individual Authentication and SPMP with Group Authentication at Maximum Random Speed from 5 to 30 m/s

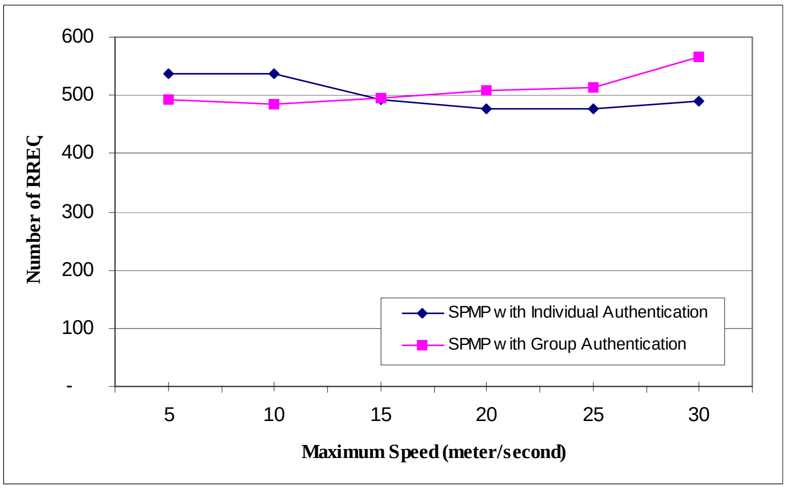

**Figure 6.4** Number of Route Request Packet That are Initially Generated by Source for SPMP with Individual Authentication and SPMP with Group Authentication at Maximum Random Speed from 5 to 30 m/s

Figure 6.1 – 6.4 show performance comparison between SPMP with individual authentication and SPMP with group authentication for various maximum random speeds. In the figure 6.1, the packet delivery ratios of both protocols are in between 86% to 91%. There are no significant variances of the packet delivery ratios between these two protocols. As shown in the figure 6.2, the average goodput of the simulated protocols were in the range of 36 – 45 kpbs. Typically, the goodput was implicitly depended on the quality of discovered path. I justified these results that the variances of goodput were not significant and the goodput of these two protocols were in the same range. When I investigated the percent dropped packet, I found that the dropped packets illustrated in the figure 6.3 varied in between 6% – 7%. The dropped packets are generally resulted from signal-to-noise. In my simulation, I assumed that the network had signal-to-noise varied from 0% to 10%. The dropped packets that were in between 6% to 7% should be normal. When comparing the results of dropped packets between SPMP with individual authentication and SPMP with group authentication, I found no materiality in differences. The figure 6.4 showed the number of route request packets generated by sources. I observed that the average route request between simulated protocols was approximate 500 packets and the variances were in between 5% to 8%. The results of route request seem to be in

the same direction as the other results that there were no significant variances between the results of the simulated protocols.

In conclusion, the impacts from the individual authentication comparing with the group authentication for SPMP routing were equivalent because of two reasons. First, the routing processes of these two protocols are exactly the same. Only difference is the authentication method that may consume different computational time but not for network performance. Second, the sizes of authentication packets of SPMP with individual authentication are as equal as those of SPMP with group authentication. As a result, the performances of network are not affected by the lengthening of control packets floating in the network.

# CHAPTER 7

# CONCLUSION

The interested problems in my research are related to development of an efficient and robust mobile ad hoc routing protocol as well as implementing security with the protocol. In the case, regression analysis is employed a statistical tool to anticipate the future performance and availability of network. Initially, I introduce a multipath routing protocol for ad hoc network called “Predicted Multipath Routing Protocol (PMP)”. The protocol forms a set of disjointed paths that have well-enough availability and high performance. The strategy of qualify path selecting is depended on two measurements: The Degree of Availability (DA) and The Estimated Path Throughput Value (ETV). The DA relies on potential signal strength, which is affected by mobility of nodes in the network. The ETV engages with the packet loss ratio that can scale efficiency of a path.

Consequently, the PMP is improved by embedding security characteristic with the protocol. As the matter of fact, the implementation of security toward the ad hoc routing protocol is aimed to discover a set of secured routes which maintain five security aspects for a wireless network that including availability, secrecy, integrity, authentication and non-repudiation. In addition, I realize that node authentication in wireless network is always a problem. Thus, I introduce a self-authentication based secure routing protocol, called “Secured Predictive Multipath Routing Protocol (SPMP)”. The SPMP consists of three main phases. The first phase associates with authentication process that enabling a pair of nodes compromise their trust. The second phase concerns with on-demand route discovery. The last phase is incorporated with key exchange infrastructure in order to maintain end-to-end data secrecy.

The result of network performance evaluation of both PMP and SPMP appeared to be satisfactory. My security analysis also shows that the SPMP provides a good security that is well enough to protect threats from various major network attacks.

My contribution consists of techniques that are beneficial for improving the performance and security of mobile ad hoc network. This dissertation gives us another view of communication technology. Some proposed techniques are possibly applied for improving current wireless network technology. Also, the security view toward this dissertation is possible to provide an idea to develop security standard of wireless technology in the future.

# BIBLIOGRAPHY

Akkaya, K. and Younis, M. 2003. An Energy-Aware QoS Routing Protocol for Wireless Sensor Networks. In **The 23rd International Conference on Distributed Computing Systems Workshops.** Providence, Rhode Island: IEEE Computer Society. Pp. 710-715.

Andrew, S.T. 2003. **Computer Networks.** 4th ed. New Jersey: Prentice Hall.

Balfanz, D.; Smetters, D.K.; Stewart, P. and Wong, H.C. 2002. Talking To Strangers: Authentication in Ad-Hoc Wireless Networks. Retrieved April 30, 2007 from http://citeseer.ist.psu.edu/balfanz02talking.html.

Bar, R. 2004a. JiST – Java in Simulation Time User Guide. Retrieved April 30, 2007 from http://jist.ece.cornell.edu/docs/040319-jist-user.pdf.

Bar, R. 2004b. SWANS – Scalable Wireless Ad Hoc Network Simulator User Guide. Retrieved April 30, 2007 from http://jist.ece.cornell.edu/ docs/040319-swans-user.pdf.

Bavejay, A. and Srinivasanz, A. 2000. Approximation Algorithms for Disjoint Paths and Related Routing and Packing Problems**.** **Mathematics of Operations Research.** 25(2): 255-280.

Bhargav, B.; Richard, O. and Fred, T. 2004. Topology Dissemination Based on Reverse-Path Forwarding (TBRPF). Retrieved May 30, 2007 from http://tools.ietf.org/html/rfc3684.

Bhaskar, R.; Augot, D.; Adjih, C.; Muhlethaler, P. and Boudjit, S. 2007. **New Technologies, Mobility and Security.** Paris, France: Springer. Pp. 633.

Chen, L. and Heinzelman, W. 2004. Network Architecture to Support QoS in Mobile Ad Hoc Networks. In **IEEE International Conference on Multimedia and Expo (ICME'04).** Naples, Italy: IEEE. Pp. 1715-1718.

Chen, L. and Heinzelman, W. 2005. QoS-aware Routing Based on Bandwidth Estimation for Mobile Ad Hoc Networks. **IEEE Journal on Selected Areas in Communications.** 23(3): 561-572.

Ching-Chuan, C.; Hsiao-Kuang, W.; Winston, L. and Mario, G. 1997. Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel. In **IEEE Singapore International Conference on Networks, SICON'97.** Singapore: IEEE. Pp. 197-211.

Corson, S. and Macker, J. 1999. Mobile Ad Hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations. Retrieved April 30, 2007 from http://www.rfc-archive.org/getrfc.php?rfc=2501.

De Couto, D. J.; Aguayo, D.; Bicket, J. and Morris, R. 2003. A High Throughput Path Metric for MultiHop Wireless Routing. In **The International Conference on Mobile Computing and Networking**. San Diego, CA: ACM MobiCom. Pp. 134-146.

Doval, D. and O’ Mahony, D. 2002. Nom: Resource Location and Discovery for Ad Hoc Mobile Networks. Retrieved April 30, 2007 from http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.20.3990.

Erwin, K. 1999. **Advanced Engineering Mathematics.** 8th ed. Singapore:Wiley.

Gouda, M.G. and Jung, E. 2004. Certificate Dispersal in Ad-Hoc Networks. In **The 24th International Conference on Distributed Computing Systems.** Tokyo, Japan: IEEE Computer Society. Pp. 616-623.

Hu, Y.C. and Johnson, D.B. 2001. Implicit Source Routes for On-demand Ad Hoc Network Routing. In **The 2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing.** Long Beach, CA: ACM. Pp. 1-10.

Hu, Y.C.; Johnson, D.B. and Perrig, A. 2002. SEAD: Secure Efficient Distance Vector Routing for Mobile Wireless Ad Hoc Networks**.** In **The Fourth IEEE Workshop on Mobile Computing Systems and Applications.** Callicoon, NY: IEEE. Pp. 3-13.

IEEE. 1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications. ANSI/IEEE Std 802.11-- Part 11. Retrieved January 30, 2010 from http://standards.ieee.org/findstds/errata/802.11a-errata.pdf.

Jain, M.; Sharma, P. and Banerjee, S. 2006. QoS-guaranteed Path Selection Algorithm for Service Composition. In **The 14th IEEE International Workshop on Quality of Service.** New Haven, CT.: IEEE. Pp. 288-289.

Johnson, D. and Maltz, D. 1996. Dynamic Source Routing in Ad Hoc Wireless Networks. **Mobile Computing.** 353(1996): 153-181.

Jung, E.; Elmallah, E.S. and Gouda, M.G. 2007. Optimal Dispersal of Certificate Chains**. IEEE Transactions on Parallel and Distributed Systems.** 18(4): 474-484.

Lee, S.J. and Gerla, M. 2000. AODV-BR: Backup Routing in Ad Hoc Networks. In **WCNC.2000 IEEE Wireless Communications and Networking Conference**. Chicago, IL: IEEE. Pp. 1311-1316.

Lee, S.J. and Gerla, M. 2001. Split Multipath Routing with Maximally Disjoint Paths in Ad Hoc Networks. In **ICC 2001. IEEE International Conference.** Helsinki, Finland: IEEE. Pp. 3201-3205.

Lehane, B. and Doyle, L. 2003. Shared RSA Key Generation in a Mobile Ad Hoc Network. In **Military Communications Conference.** Boston, MA: MILCOM. Pp. 814-819.

Lin, T.; Midkiff, S. and Park, J. 2003. A Framework for Wireless Ad Hoc Routing Protocols. **Wireless Communications and Networking.** 2: 1162-1167.

Mao, S.; Wang, Y.; Lin, S. and Panwar, S. 2002. Video Transport over Ad-hoc Networks with Path Diversity. **Mobile Computing and Communications Review.** 1(2): 1-2.

Mao, S.; Wang, Y.; Lin, S. and Panwar, S. 2003. Video Transport over Ad Hoc Networks: Multistream Coding with Multipath Transport. **IEEE Journal on Selected Areas in Communications.** 21(10): 1721-1737.

Montenegro, G. and Castelluccia, C. 2002. Statistically Unique and Cryptographically Verifiable (SUCV) Identifiers and Addresses**.** In **The 9th Annual Network and Distributed System Security Symposium.** San Diego, CA: The Internet Society. Pp. 1-13.

Nasipuri, A. and Das, S.R. 1999. On-demand Multipath Routing for Mobile Ad Hoc Networks. In **The 8th International Conference on Computer Communications and Networks.** S. Dixit, A. Somani and E.K. Park, eds. Boston, MA: IEEE. Pp. 64-70.

National Institute of Standards and Technology (NIST). 1993. Secure Hash Standard. In **Federal Information Processing Standards (FIPS) 180-1.** Retrieved January 30, 2010 from http://www.itl.nist.gov/fipspubs/fip180-1.htm.

National Institute of Standards and Technology (NIST). 1999. Data Encryption Standard (DES). Retrieved January 30, 2010 from http://www.itl.nist.gov/fipspubs/fip46-2.htm.

National Institute of Standards and Technology (NIST). 2001. Advanced Encryption Standard (AES). Retrieved January 30, 2010 from http://csrc.nist.gov/publications/fips/fips197/fips-197.pdf.

Nikaeing, N. and Bonnet, C. 2002. A Glance at Quality of Service Models for Mobile Ad Hoc Networks. In **Le 16eme Congrès DNAC (De Nouvelles Architectures pour les Communications).** Paris, France: EURECOM Consortium. Pp. 1-8.

P. Hiranvanichchakorn and S. Lertvorratham 2010. Using Regression Analysis for Improving Multipath Ad Hoc Network Performance. **The International Journal of Computer and Applications.** 32(2): 1-9.

Papadimitratos, P. and Hass, Z. J. 2002. The Secure Routing for Ad Hoc Networks. In **The SCS Communication Networks and Distributed Systems Modeling and Simulation Conference (CNDS 2002).** San Antonio, Texas: The Society for Modeling and Simulation International (SCS). Pp. 27-31.

Patwardhan, A.; Parker, J.; Joshi, A.; Iorga, M. and Karygiannis, T. 2005. Secure Routing and Intrusion Detection in Ad Hoc Network**.** In **The third IEEE International Conference on Pervasive Computing and Communications (PERCOM '05).** Kauai, HI: IEEE Computer Society. Pp. 191-199.

Pavon, P. and Choi, S. 2003. Link Adaptation Strategy for IEEE 802.11 WLAN via Received Signal Strength Measurement. In **The ICC'03 IEEE International Conference on Communications.** Anchorage, AK: IEEE. Pp. 1108-1113.

Perkins, C. 2003. Ad Hoc On-demand Distance Vector (AODV) Routing. In **The Internet Engineering Task Force.** Retrieved April 30, 2007 from Mobile Ad-hoc Networks Working Group, http://www.ietf.org/rfc/rfc3561.txt

Perkins, C. and Bhagwat, P. 1994. Highly Dynamic Destination-Sequenced Distance Vector (DSDV) for Mobile Computers. In **The SIGCOMM 1994 Conference on Communications Architectures, Protocols and Applications.** London, UK: ACM SIGCOMM. Pp. 234-244.

Pervaiz, M.O.; Cardei, M. and Wu, J. 2002. Routing Security in Ad Hoc Wireless Network. **IEEE Communications Magazine**. 40(10): 70-75.

Rajendra, P. M. and Mohit, K. 2010. Taxonomy of Routing Security for Ad-Hoc Network. **International Journal of Computer Theory and Engineering.** 2(2): 1793-8201.

Rivest, R.L. 1992. The MD5 Message-digest Algorithm. RFC 1321. Retrieved January 30, 2010 from http://tools.ietf.org/html/rfc1321.

Rivest, R.L.; Shamir, A. and Adleman, L. 1978. A Method for Obtaining Digital Signatures and Public-Key Cryptosystems. **Communications of the ACM.** 21(2): 120-126.

Roy, S. and Garcia-Luna-Aceves, J.J. 2002. An Efficient Path Selection Algorithm for On-demand Link-state Hop-by-hop Routing. In **The 11th International Conference on Computer Communications and Networks.** Miami, Florida: IEEE Communications Society. Pp. 561-564.

S. Lertvorratham and P. Hiranvanichchakorn. 2007. Path Selection Strategies for Multipath Ad Hoc Network. In **The IASTED International Conference on Communication Systems and Networks**. Phuket, Thailand: IASTED. Pp. 164-170.

Sanzgiri, K.; Dahilly, B.; Leviney, B. N.; Shieldsz, C. and Belding-Royer, E. M. 2002. A Secure Routing Protocol for Ad Hoc Networks. In **The 10 th IEEE International Conference on Network Protocols (ICNP'02).** Paris, France: IEEE. Pp. 78-87.

Sun, H. and Hughes, H. 2003. Adaptive Multi-path Routing Scheme for QoS Support in Mobile Ad-hoc Networks. In **The 2003 International Symposium on Performance Evaluation of Computer and Telecommunications** Systems **(SPECTS'03).** Montreal, Canada: The Society for Modeling and Simulation International. Pp. 408-416.

Tuduce, C. and Gross, T. 2004. Resource Monitoring Issues in Ad Hoc Networks. In **International Workshop on Wireless Ad-Hoc Networks.** Oulu, Finland: Centre for Wireless Communications, University of Oulu. Pp. 259-264.

Vetriselvi, V. and Parthasarathi, R. 2003. Secure Communication for Multi-Path Ad-Hoc Network. In **The Conference on Convergent Technologies for Asia-Pacific Region (TENCON 2003).** Taipei, Taiwan: IEEE. Pp. 1086 - 1090.

Wei, W. and Zakhor, A. 2004. Multipath Unicast and Multicast Video Communication over Wireless Ad Hoc Networks. In **The 1$^{st}$International Conference on Broadband Networks (Broadnets).** San Jose, CA: IEEE Computer Society. Pp. 496-505.

William, S. 2003. **Cryptography and Network Security Principles and Practice.** 3$^{rd}$ ed. New Jersey: Prentice Hall.

Wu, K. and Harms, J. 2001. QoS Support in Mobile Ad Hoc Networks. **Interdisciplinary Journal of Crossing Boundaries.** 1(1): 92-106.

Zapata, M.G. 2002. Secure Ad hoc On-Demand Distance Vector (SAODV) Routing. **Mobile Computing and Communications Review**. 6(3): 106-107.

Zheng, H.; Omura, S.; Uchida, J. and Wada, K. 2004. An Optimal Certificate Dispersal Algorithm for Mobile Ad Hoc Networks. In **The third International Symposium on Parallel and Distributed Computing/Third International Workshop on Algorithms, Models and Tools for Parallel Computing on Heterogeneous Networks (ISPDC/HeteroPar'04)**. Ireland: University College Cork. Pp 42-48.

Zhou, L. and Hass, Z. J. 1999. Securing Ad Hoc Networks. **IEEE Network.** 13(6): 24-30.

Zygmunt, J. H. and Marc, R. P. 1997. The Zone Routing Protocol (ZRP) for Ad Hoc Networks. Retrieved April 30, 2007 from http://tools.ietf.org/html/draft-ietf-manet-zone-zrp-00.

Zygmunt, J. H.; Marc, R. P. and Prince, S. 2002. The Broadcast Resolution Protocol (BRP) for Ad Hoc Networks. Retrieved April 30, 2007 from http://tools.ietf.org/html/draft-ietf-manet-zone-brp-02.

# BIOGRAPHY

| | |
|---|---|
| **NAME** | Supachote Lertvorratham |
| **ACADEMIC BACKGROUND** | Bachelor of Accounting,<br>Chulalongkorn University, Thailand,<br>1985<br>Master of Business Administration<br>(Management Information System),<br>The University of Dallas, Texas, U.S.A.,<br>1991 |