박 사 학 위 논 문

이동형 애드 혹 네트워크를 위한 의사 거리 경로 탐색 방법 및 링크 안정성 예측 모델

이 민 구 (李 珉 九)
전자컴퓨터공학부(병렬 처리전공)
포항공과대학교 대학원
2007

이동형 애드 혹 네트워크를 위한 의사 거리 경로 탐색 방법 및 링크 안정성 예측 모델

A Pseudo-Distance Routing Algorithm and Link Stability Estimation Model for Mobile Ad Hoc Networks

A Pseudo-Distance Routing Algorithm and Link Stability Estimation Model for Mobile Ad Hoc Networks

by

Min-Gu Lee

Division of Electronic and Computer Engineering
(Parallel Computing Program)
Pohang University of Science and Technology

A thesis submitted to the faculty of Pohang University of Science and Technology in partial fulfillment of the requirements for the degree of Doctor of Philosophy in the Division of Electronic and Computer Engineering (Parallel Computing Program)

Pohang, Korea

11. 22. 2006

Approved by

Sunggu Lee
Major Advisor

이동형 애드 혹 네트워크를 위한 의사 거리 경로 탐색 방법 및 링크 안정성 예측 모델

이 민 구

위 논문은 포항공과대학교 대학원 박사 학위논문으로 학위논문 심사위원회를 통과하였음을 인정합니다.

2006년 11월 22일

학위논문심사 위원회 위원장 이 승 구 (인)

위 원 홍 성 제 (인)

위 원 김 종 (인)

위 원 정 홍 (인)

위 원 송 황 준 (인)

DECE 20023478 이 민 구 Min-Gu Lee, A Pseudo-Distance Routing Algorithm and Link Stability Estimation Model for Mobile Ad Hoc Networks. 이동형 애드 혹 네트워크를 위한 의사 거리 경로 탐색 방법 및 링크 안정성 예측 모델, Division of Electronic and Computer Engineering, 2007, 133P, Advisor : Sunggu Lee. Text in English.

ABSTRACT

This dissertation proposes a general routing protocol and link stability estimation model based on a pseudo-distance concept. Using the link stability estimation model, this dissertation also proposes a stable routing protocol for mobile ad-hoc networks with highly mobile nodes resulting in unstable links..

First, routing algorithm for mobile ad-hoc networks(MANETs) is studied. Previous routing algorithms for MANETs have focused on finding short-distance path(s) between communicating nodes. However, due to the dynamic and unreliable communication nature of MANETs, previously determined paths can easily become disconnected. Although dynamic routing can be used to circumvent this problem, determining a new route each time a packet needs to be sent involves a lot of overhead. An alternative form of dynamic routing involves maintaining valid routes in routing tables, which can be dynamically updated whenever network changes are detected. This dissertation proposes a new routing algorithm, referred to as pseudo-distance routing (PDR), that supports efficient routing table maintenance and dynamic routing based on such routing tables.

Second, the problem of supporting stable routing is studied. When using shortest-distance routing for mobile ad-hoc networks (MANETs), the physical

distances of links that constitute such paths tend to be very long since this leads to fewer hops between source and destination nodes. However, if the physical distance of a wireless link becomes so long that it approaches its transmission range, packet transmission error rates can increase drastically, resulting in an unstable link. Furthermore, packets are more likely to be lost due to external environment factors such as white noise and wireless interference if the signal strength is not strong enough. Therefore, it would be desirable for routing algorithms for MANETs to be able to select paths that are more likely to be stable. With this objective in mind, we propose an enhanced stability model (ESM) to estimate link stability based on signal strength. A routing algorithm based on this new model is also proposed. Simulations of the proposed ESM and previous link estimation models validate the superiority of the proposed approach. Simulations also show that the proposed routing algorithm performs particularly well when there are unreliable links.

To my parents, wife and children

Contents

List of Figures

List of Tables

1 Introduction

Wireless networking area has rapidly become a crucial component of computer networks and the demand of wireless networking has been growing exponentially in the past decade. Nowdays, we can browse the Internet or check e-mail using portable devices such as PDAs(Personal Digital Assistant) or Notebook PCs with wireless networking capability (using, e.g., IEEE 802.11 [3], Bluetooth [4, 5] or HiperLan [6, 7] devices) wherever we are within the range of pre-deployed base stations or wireless access points. The principal reasons for the rapid growth of wireless networks include need to support mobility of terminal nodes, reduced installation time and costs, long-term cost savings, etc.

1.1 Wireless Networks

Wireless network standards such as Bluetooth, IEEE 802.11 and HiperLan enable to create a wide range of new applications such as wireless broadband multimedia and data communications within its transmission range like home or office. Most of laptops and PDAs are sold with equipped wireless network devices such as IEEE 802.11 Network Interface Cards(NICs) or Bluetooth NICs. Nowadays, we can access network services virtually everywhere at any time in via wireless networking. In addition, commertial broadband wireless networking services are also available such as T-Mobile HotSpot [8] and KT Nespot [9] services that are based on the IEEE 802.11 networking technology.

1.1.1 Benefits of Wireless Networks

Mobility of portable devices with wireless networking capability enables users to move physically while using their networking applications without discontinues of services. For instance, retail stores build wireless networks in order to interconnect handheld bar-code scanners to the database server that stores current price of items and corresponding stock information. This enables that clerks can print out correct prices of items. Not only mobile applications, wireless networking technologies offer cost savings, especially if installations of wires are very expensive or difficult. Suppose that freeways, rivers or other obstacles separate buildings that need to be connected. Then wireless connection is more cost effective way to interconnect them rather than installing physical cables or leasing additional communication services. In addition, wireless networking also reduces the installation time of physical wire lines.

1.1.2 Characteristics of Wireless Networks

Even wireless devices have many benefits, they potentially have problems as listed below.

- **Multipath propagation**: Transmitted signals can combine with reflected ones that corrupt the signal at the receiver as shown in Figure 1.1. Direct signal is propagated signal directly to the receiver. Signals can be reflected if it hits an very large object such as buildings, walls and ground. If signals hits an impenetrable object, the signal bends at the edge of the object which is called diffraction as shown in Fig. 1.1. This diffracted signal enables that the signal can reach behinds the object which is not in line-of-sight. The amount of diffraction is depends on radio frequency. If signals pass through a medium that is composed of many small objects compared to wave length such as trees or street signs, scattering occurs. *Delay spread* is the amount of delay experienced by the last reflected signals compared to the first received signals. As the amount of the delay spread increased, the signal at the receiver is getting worse. This multipath propagation can be a significant problem that decreases quality of signal with indoor application.

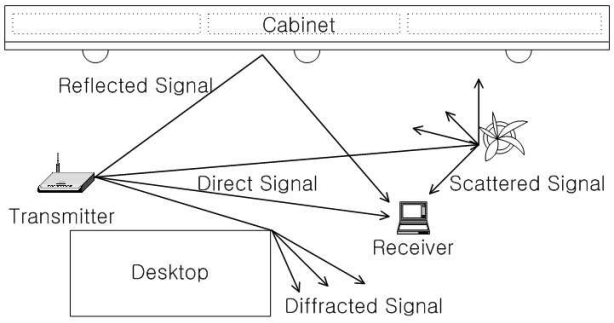


Figure 1.1: An example of multipath propagation.

- **Path loss**: Path loss can be expressed as the ratio of transmitted power to the power of the same signal received at the receiver. Path loss depends on various factors such as radio frequency or condition of wireless medium, and it grows exponentially as the distance increases between the transmitter and receiver. With typical indoor applications, the path loss increases approximately 20dB every 100 feet [10].

- **Interference**: Wireless devices may be suffered by interferences of the harmonics of transmission systems or other systems using similar radio frequencies in the local area. For instance, microwave ovens, which operate in the 2.4Ghz ISM(industrial, scientific and medical) radio band, can cause significant delay or bit errors to the IEEE 802.11 family devices that are most commonly used among wireless networking products. In addition, adjacent channel interference and co-channel interference also decrease the quality of signal at the receiver.

- **Limited computing and energy resources**: Most of mobile devices like laptops and PDAs tend to have limited computing power, memory due to limited battery capacity.

- **Low bandwidth**: Due to various factors described above, wireless networks support lower bandwidth than wired communication networks. Low bandwidth causes degraded quality of service, including higher jitter, delays, longer setup times, etc.

- **Highly variable network conditions**: Network conditions are very frequently changed in wireless networks. One of the distinguished characteristics of wireless networks is burst error of wireless channel. Supposet that a person crosses the direct line of sight from the transmitter to the receiver. Then quality of signal may be degraded since direct signal is no longer

availble, with bursty errors. However, after the person crossed the direct line of sight, communication between two nodes would be recovered.

- **Limited transmission resources**: Wireless communications shares the medium due to the limited availability of frequencies with restricted regulations. Users may experience excessive delays due to contentions of the shared medium.

1.1.3 Infrastructured Wireless Networks

Most of currently deployed wireless networks are operated in infrastructure-based configuration that base stations(BSs) or access points(APs) provide interface between wireless portable devices and other networks such as the Internet as routers or bridges with limited range of communications up to a few hundred meters. Figure 1.2 shows a typical configuration of infrastructured wireless networks. Mobile nodes within communication range of AP or BS can access the Internet through gateway functionality of AP or BS. However, a mobile node that is located outside of the communication range of the AP or the BS is not able to access the Internet even if it is within the communication range of other mobile nodes that is located within the communication range of the AP or the BS. Note that, mobile nodes that are connected to the AP or the BS can not communicate to other nodes directly in the environment of network configuration like Fig. 1.2.

Although infrastructure-based networks provide a great way to connect mobile devices to other network services, it still required pre-installed infrastructures like BSs or APs, and the costs associated with installing such infrastructures would be very high even if it does not expense cabling costs to the end-terminals i.e., mobile devices. Furthermore, in some application situations, setting up the infrastructures would be quite difficult, can not be installed or can not be installed

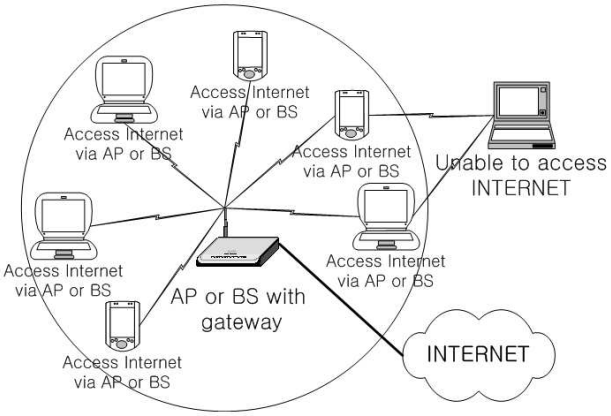


Figure 1.2: An example of a infrastructured wireless network.

in time. For all above reasons, alternative methods to support network connectivity without infrastructure have been gaining increased attention in recent years.

1.2 Mobile Ad Hoc Networks

Mobile ad hoc networks(MANETs) are composed of mobile nodes that communicate with each other using radio transmission without a fixed infrastructure or centralized administration dynamically. Such nodes can communicate with other nodes that are within their radio transmission ranges. Figure 1.3 shows an example of MANETs. As shown in Fig. 1.3, a MANET may consist of heterogeneous devices including laptops, PDAs, sensors, actuators, etc. Each node is able to communicate directly with other nodes that is located within its radio transmission range. If a node needs to communicate outside of its radio transmission range, intermediate nodes are used to relay packets from the source to the destination that forms multihop wireless MANETs.

MANETs are self-organizing and rapidly deployable. Mobile nodes can be set

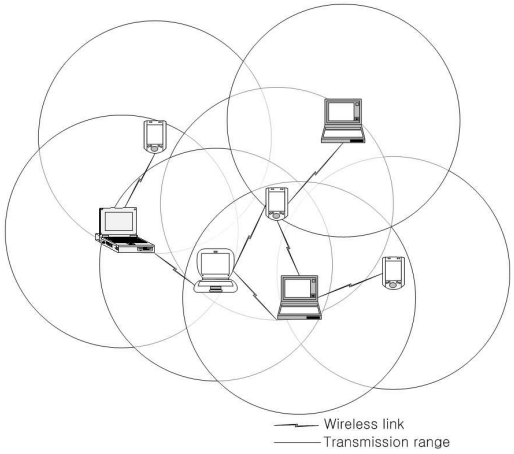


Figure 1.3: An example of mobile ad hoc networks(MANETs).

up anywhere at anytime without any network administration or infrastructures. Therefore, this new network architecture can achieve flexibility, mobility and ease of installation compared to infrastructured wireless networks. These properties of MANETs eliminate the constraints of infrastructured networks and enable mobile nodes to create and join networks on-the-fly for virtually any applications.

1.2.1 Application Areas of MANETs

Because of flexible, self-organizing and rapidly deployable properties, MANETs can be very useful to establish communications among a group of soldiers for tactical operations. It is very difficult to set up infrastructures in enemy territories or in inhospitable terrains in time. In such environment, a MANET can provide an efficient and effective communication mechanism because it is easy to set up and organize. In addition, MANETs can provide reliable communications compared to infrastructured networks because there is no single point of failures such as base stations, access points, central administration systems, etc. If infrastructured networks are used in battle field, enemies can easily break the

communication networks by destroying infrastructures.

Another application of MANETs is emergency operations such as search and rescue in disaster areas. The major factors that such applications prefer MANET are self-configurability with minimal overhead, independent operations to the infrastructure, mobility of nodes, and the unavailability of network infrastructure in nature terrain. Suppose that network infrastructures were destroyed by earthquake or a tidal wave. In such environments, MANETs would be a good solution for coordinating searching and rescuing activities by immediate deploying mobile devices with wireless networking technologies.

Wireless mesh network(WMN) [11, 1, 12, 13] that consists of mesh routers and mesh clients is another form of MANETs that enables to interconnect other networks like the Internet. Mesh routers have minimal mobility and form the backbone networks of wireless mesh networks in order to provide communication services toward other networks such as the Internet to mobile or fixed nodes/users. Figure 1.4 shows an example of wireless mesh networks that organizes backbone networks to access the Internet which is adapted from [1]. As shown in Fig. 1.4, wireless routers configures backbone networks without wires. It looks like cellular system but a WMN does not need to plan cellular networks. An user that connected to a WMN can browse the Internet using his handheld device via the backbone networks of wireless mesh routers. Note that mesh routers that are connected to the Internet may have wired connection to the Internet as in cellular system. The benefits of a WMN include no single point failure, quick and low cost of deployment of backbone networks, high scalability, extendability, high availability, etc. If service providers wish to increase the number of concurrent users that can access the cellular networks, additional infrastructural devices are need to be set up and carefully adjusted that are usually very expensive. Furthermore, previously deployed infrastructures have to be re-configured in order to set up additional infrastructural devices that also increases costs. However, maintenance

and extension of wireless mesh networks are very cost effective because wireless mesh networks support incremental deployments by self-configurable properties. Service provider can easily set up additional mesh routers to increase concurrent accessible users. WMNs can cover from home networks to metropolitan area networks as described in [1].

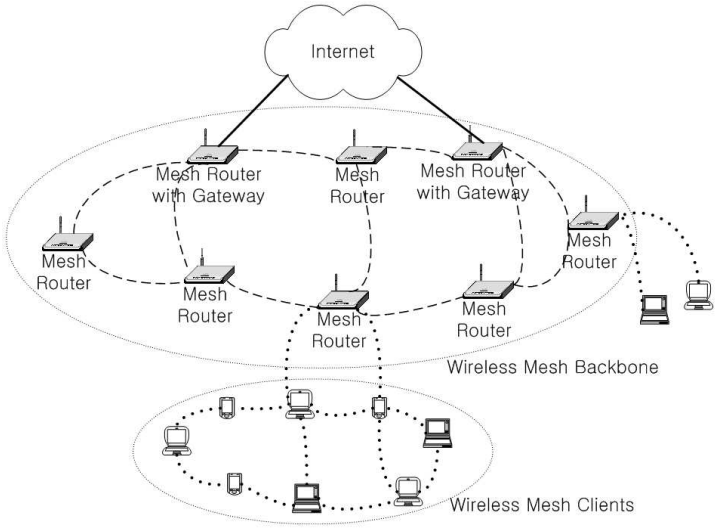


Figure 1.4: An example of wireless mesh networks(WMNs) - adapted from [1].

1.2.2 Characteristics of MANETs

As described earlier, MANETs have many advantages such as self-configuration, ease of deployment, etc. However, these benefits come at a price. MANETs inherit common characteristics that found in wireless networks such as multi-path propagation, path loss, interference, limited computing and energy resource, low bandwidth, highly variable network conditions and limited transmission resources. One of the properties that distinguishes MANETs from other wireless network architectures is that mobility of nodes changes the network topology fre-

quently. The links of a MANET is not fixed that status of links are changed over time and are dependent on the geographical location and mobility of nodes, characteristics of transmitter and receiver radio devices, and the signal propagation properties of the environment. Note that locations of nodes keep changing over time because of all nodes in the MANETs have potential of movement. Therefore, as movements of nodes are getting faster and movement patterns of nodes are independent, the topology of a MANET changes more dynamically. One of important features of MANETs is that there is no fixed routers due to lack of infrastructure. Therefore each mobile node itself must act as a router - storing and forwarding of packets, maintaining valid routes from source to destinations, etc. This property also introduces difficulties of routing that routers in MANETs are also mobile. Therefore, conventional routing algorithms are not directly applicable for MANETs. Note that the goal of routing algorithm for MANETs is not only to route optimally but also to be adaptable to highly dynamic changes in network topology. Not only in routing problem, infrastructureless arises other complexities such as difficulties of fault detection and management, packet losses due to topological changes, hidden terminal and exposed terminal problems, higher packet loss probability by confliction of channel access, variations in link and node capabilities that may cause asymmetric links, limited battery power of mobile routers, scalability management, guaranteeing quality of services, and lack of network securities. With these factors in mind, we will discuss design issues and constraints of MANETs in the next section.

1.2.3 Design Issues of MANETs

In this section, various issues and challenges that need to be considered when a MANET is designed are discussed. Major issues that affect the performance of MANETs are as follows.

- **Medium access control**: The primary operations of MAC(medium access control) in MANETs is the distributed arbitration for the shared channel. Because collision detection mechanism is not able to use in MANETs, another challenging issue is arisen to conventional CSMA/CD(carrier sensing medium access with collision detection)-based MAC protocols. Figure 1.5 shows an typical configuration that a hidden terminal problem arises. A Hidden terminal problem occurs when two or more nodes that are not able to detect each other due to being outside of each other transmission range but their transmission range is not disjoint. As shown in Fig. 1.5, collisions can occur at node MH1 when both MH2 and MH3 start transmitting there data packets toward MH1.

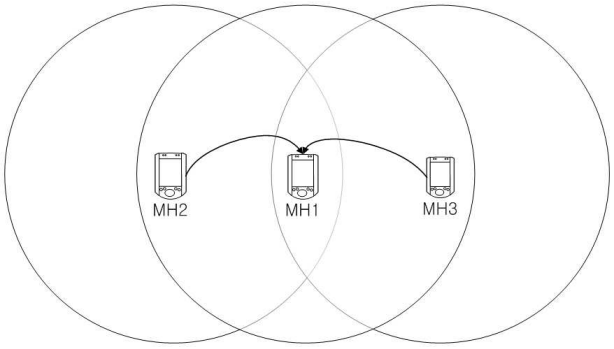


Figure 1.5: An example of a typical hidden terminal problem.

An exposed terminal problem occurs when a node has to wait until transmission is finished from a node within its transmission range to another node that is outside of the transmission range due to carrier sensing that is actually able to transmit to its own destination node. As shown in Figure 1.6 that shows a typical example of a exposed terminal problem, node MH3 can not send its own data packets to MH4 because it detects the carrier signal that node MH2 is sending to node MH1. Although packet

from MH3 to MH4 does not collide at MH1, MH3 can not transmit its data due to CSMA. An exposed terminal problem may thus result in loss of throughput of the network. MAC protocol for MANETs should be able

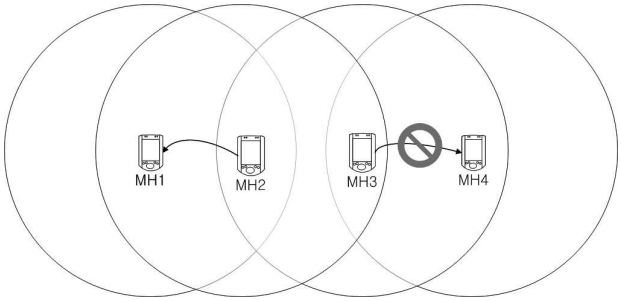


Figure 1.6: An example of a typical exposed terminal problem.

to support distributed operations due to lack of centralized coordination function, time synchronization, fairness to all competing nodes or flows.

- **Routing**: Due to dynamic nature of MANETs arising from factors such as the mobility of the nodes, low signal power, suspended states of intermediate nodes for energy conservation and interference in the wireless channel, network topology of a MANET is not fixed. These factors can cause frequent and unpredictable changes in network topology that increase difficulties and complexities to routing protocols. The primary objectives of routing protocols for MANETs are correct, reliable and delivery of packets and efficient route discovery and maintenance between a source and a destination. Conventional routing algorithms that are based on distance-vector and link-state-based routing protocols are not adequate in MANETs since they are not able to catch up frequent topological changes. Therefore, new routing protocols need to be designed to suit the specific needs of MANET environments. Detailed requirements of routing protocols for MANETs will be discussed in Chapter 2.

- **Multicasting and broadcasting**: Multicasting and broadcasting play an important role in typical applications of MANETs. In MANETs, conventional tree based multicast or broadcast structure is very unstable since topology of the network is frequently changed. Therefore they have to be frequently readjusted to catch up topological changes of MANETs.

- **Performance of transport layer protocols**: A connectionless transport layer protocol such as User datagram protocol(UDP) take into account neither flow control and congestion control nor reliable communication. Such protocols may degrade the network performance since they do not take into account the current network status such as congestions at the intermediate links, the rate of collision, or other factors.

  Another transport layer protocol in widely used is transmission control protocol(TCP). The performance of TCP is degraded due to broken paths, presence of stale routing information, high channel error rate, etc. As described in [14, 15, 16, 17, 18, 19], performance of TCP is severely impacted by mobility of nodes. Suppose that existing paths experienced frequent path breakage that required to configure routes due to the mobility of nodes and limited transmission range. If reconfiguration of route takes longer than the transmission timeout of the TCP at the sender, sender retransmits all packets in the contention window and executes the congest control algorithm that decrease the size of congestion window that results degrading of throughput. There are other cases that degrades network performance of TCP such as loss of ACK that invokes congest control process. Furthermore, channel capture problem [20, 21, 22, 23] can occur even in static configuration of nodes with multiple TCP sources. Therefore, transport layer protocols should address above issues to perform efficiently in MANETs.

- **Quality of service(QoS)**: One of the main issue of supporting QoS is defining service model of QoS that support users whether on a per-session basis or on a per-class basis. The other key issue of supporting QoS is finding feasible path that can satisfy users requirements. QoS management protocol should react promptly according to changes of network status.

- **Network security**: Because nodes in MANETs generally communicate with each other via open and shared broadcast medium(wireless channel), it is more vulnerable to security attack. In addition, the distributed and infrastructureless architecture prevent centralized security control.

- **Energy management**: In [11], energy management is defined as the process of managing the sources and consumers of energy in a node or in the network as a whole for enhancing the lifetime of the network. Energy management can be performed in part as following: (1) transmission power management, (2) battery energy management, (3) processor power management, (4) device power management.

- **Scalability**: Many future applications of MANETs would be composed of very large number of nodes. Even nowadays commercial deployments of of wireless mesh networks consist of a large number of wireless mesh routers and clients. Therefore, scalability is on of the key issue of designing MANETs.

1.3 Objective and Outline of Dissertation

This dissertation studies two major issues of supporting MANETs. The first is routing protocol that discover and maintain routes efficiently. The proposed routing protocol achieves high packet delivery ratio, short path length, fully distributed operations, etc. The second is link stabilty estimatio modelfor wireless

networks. Because of poor channel quality of wireless medium, carefull selection of path is required in order to increase packet delivery ratio. A new link stability estimation model and corresponding routing protocol are proposed in order to achieve high packet delivery ratio.

The remainder of this dissertation is organized as follows: Reviews of previous proposed routing algorithms that address highly dynamic environments will be reviewed in Chapter 2. The proposed routing algorithm, referred to *pseudo-distance routing (PDR)*, is presented in Chapter 3. In chapter 4, supporting stable routing is proposed to select paths that are more likely to be stable using the proposed link stability estimation model, referred to *enhanced stability model (ESM)*based on signal strength. Finally, this dissertation concludes in Chapter 5 with summary of main contributions and future works.

2

Routing in MANETs

There have been significant interests in routing algorithms for MANETs in the recent past. Routing algorithms that are developed for ad hoc networks can be divided into 3 categories based on routing information update mechanisms: *proactive, reactive* and *hybrid*. Proactive(or table-driven) routing algorithms attempt to maintain consistent and up-to-date routing information from each node to every other node in the network. In order to maintain consistent and up-to-date routing information, proactive algorithms must frequently exchange routing information. On the other hand, reactive(or on-demand) routing algorithms create routes only when desired by the source node. Whenever a node requires a route to a destination, it initiates a route discovery process within the network.

Hence, routing protocols in this category do not exchange routing information periodically. Hybrid routing protocols combine best features of the above two categories. Routings in the same zone which is distinguished by location of nodes are table driven. For routing beyond this zone, reactive algorithm is used.

2.1 Issues of Routing Protocols for MANETs

Due to dynamic nature of the network arising from factors such as the mobility of the nodes, low signal power, suspended states of intermediate nodes for energy conservation and interference in the wireless channel, network topology of a MANET is not fixed. These factors can cause frequent and unpredictable changes in network topology that increase difficulty and complexity to routing protocols. The primary objective of routing protocol for MANETs is correct, reliable and efficient route discovery and maintenance between a source and a destination. Conventional routing algorithms that are based on distance-vector and link-state-based routing protocols that were developed for fixed or infrastructured networks are not adequate for MANETs because they are not able to catch up frequent topological changes. In addition, routing protocol should consider constraints of resources including CPU clock cycles, amount of memory usage and battery related issues, poor channel quality, hidden and exposed terminal problems, etc.

2.1.1 Goals of Routing Protocols for MANETs

As discussed earlier, any conventional routing protocols can not be used in MANETs. Hence, specialized routing protocol that addresses the issues described above is required. Typical design goals of routing protocols for MANETs are following:

- **Fully distributed operation**: Centralized routing protocols are not scalable since they involve high control overheads in order to collect up-to-date network topology information and propagating route information to the network. In addition, centralized routing protocol may suffer from a single point of failure problem. On the other hand, distributed routing protocols do not need to collect up-to-date topological information of network to the central point and free to the risk of single point of failure.

- **Minimal control overhead**: It is needless to say that routing overhead should be minimized. Frequent topological changes of MANETs may increase the number of control messages to reflect the changes into routing information. Because control message consumes bandwidth and battery power, it should be minimized as possible.

- **Minimal processing overhead**: Computational intensive routing protocols require significant amount of processing power and memory usage. CPU cycles to process complex routing protocols consumes a lot of battery power which is one of the most strict constraints of applications in MANETs.

- **Loop free**: A loop of routing path delays message delivering to destination. In extreme case, messages are failed to deliver to destination if TTL of the message is expired by route loop. Because bandwidth is very scarce in wireless networks and packet processing/transmission consumes a lot batter power of mobile devices, extremely wasteful routing loop should be avoided at all time.

- **Multiple paths**: Due to dynamic nature of MANETs, a path that already discovered may be disconnection frequently. If routing protocols provide multiple paths, then one of other paths may still be valid. If applications

demand QoS routing, routing protocols should be able to provide multiple paths in order to find a feasible path that meets users constraints.

- **Avoiding packet loss** It is needless to say that routing protocol should deliver packets to destination correctly. However, due to mobility of nodes and other effects of wireless communication, packets can be lost during packet delivery. Therefore, routing algorithm for MANETs should support stable routing that delivers packets using stable links in order to avoid packet losses. In addition, due to mobilty, previously discovered routes are easily broken. In such case, packets that are already transmitted by source nodes are dropped at intermediate nodes in most routing algorithms that leads to low packet delivery ratio. In order to deliver packet correctly to destinations, routing protocols should reconstruct routes at tinermediate nodes in order to deliver already transmitted data packets from source nodes.

- **Quick convergence**: Convergence means that states of routing information are stable(= no changes) through the whole network after topological change event initiate routing maintenance procedure. It is needless say that routing should reflect current network status as quickly as possible to route data packet efficiently and reliably.

- **Localized maintenance of route**: Topological changes of the network should not affect the whole network in order to reduce control packets, processing overhead and convergence time. Propagation of routing information should be minimized that only nodes that are affected by topological changes update route information.

- **Minimal path**: Increased path length(= the number of hops of the route) results increased number of packet forwarding that consumes bandwidth

and energy, increased end-to-end delay, etc. In addition, if path length is increased, probability of path breakage is also increased that leads to higher packet loss probability. Therefore routing protocol should provide minimal paths as possible.

- **Scalability**: Typical application of MANETs are composed of a few tens of nodes, but some applications may be composed of a few hundreds of nodes or even more. In order to support such kind of applications, routing protocols should be scalable.

- **Supporting QoS**: Many applications require certain level of QoS. Routing protocols should be able to select routes that meets the constriants that user specified. In order to support QoS, routing protocol should support multiple paths and quick discovery of routes, etc.

2.1.2 Classifications of Routing Protocols

With these goals in mind, numerous routing protocols have been developed for MANETs. Such routing protocols can be classified into several types based on different criteria. The most widely accepted category of routing protocols for MANETs is how to update routing information for the dynamic topological changes. There are three major categories based on the routing update mechanisms.

- **Proactive or table-driven**: Table-driven routing protocols attempt to maintain consistent and up-to-date routing information from each node to every other node in the network. In order to maintain consistent and up-to-date routing information, table-routing protocols must exchange routing information frequently. Generally, routing information is flooded in the whole network except routing protocols based on hierarchical structure.

Whenever a node requires to have a path to destination, it executes an appropriate path finding algorithm using the topology information that it maintains. Protocols of this category are discussed in Section 2.3

- **Reactive or on-demand**: Reactive algorithms create routes only when they are desired by source nodes. Whenever a node seeks a route to a destination, it initiates a route discovery procedure throughout the network. Most of routing protocols that are classified in this category floods route request message into the whole network that results significant overhead of route discovery procedure. Protocols of this category are discussed in Section 2.4

- **Hybrid**: Routing protocols that are classified in this category combine the advantages of proactive and reactive protocols. Most of hybrid routing protocols employs table-driven routing protocols that nodes within a certain distance from the source node, or within a particular geographical region. However, if destination node is outside of its region, then it uses reactive routing protocols in order to reduce control overheads. Protocols of this category are discussed in Section 2.5

Figure 2.1 shows a classification of routing protocols for MANETs. Detailed of routing protocols in the Fig. 2.1 will be discussed in the remained part of this chapter.

2.2 Fundamental Routing Protocols

This section discusses two fundamental routing protocols that are basis of current routing protocols. The first one is distributed Bellman-Ford algorithm and the second one is link reversal algorithms.

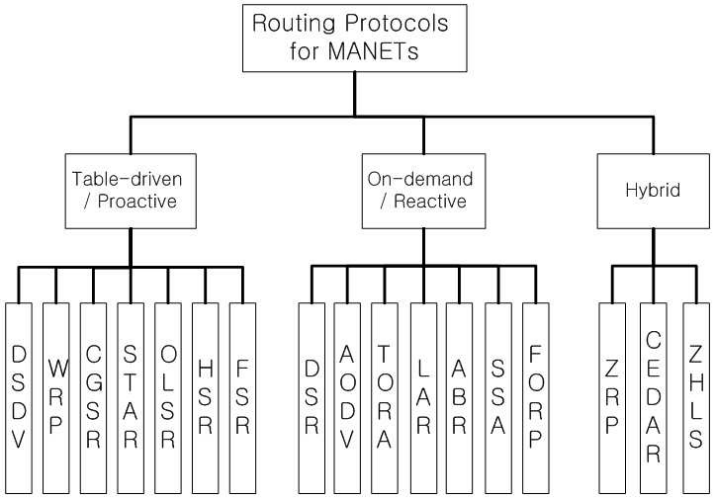


Figure 2.1: A classification of routing protocols for MANETs.

2.2.1 Distributed Bellman-Ford Algorithm

Distributed Bellman-Ford(DBF) [24, 25, 26] algorithm that was developed by Dimitri P. Bertsekas and Robert G. Gallager in 1987 is a table-driven protocol on basis of the Bellman-Ford Algorithm. Each router constantly maintains up-to-date routing table with information on how to reach all possible destinations in the network. Each node maintains a routing table of <destination, metric, successor> where metric can be hop-distance to destination, total delay, or cost to the destination. Each node sends path vector tuples <destination, distance> to all its immediate neighbors periodically to maintains up-to-date and consistent routing information.

However, DBF suffers *count-to-infinity* and route loop problem. Furthermore, DBF does not scale since increased route update overhead is rapidly increased with mobility.

2.2.2 Link Reversal Algorithms

Link reversal(LR) algorithms [2] that were developed by Eli M. Gafni and Dimitri P. Bertsekas in 1981 are on-demand routing protocols that try to maintain connectivity in frequently changing topology of packet radio network. The main objective of LR algoritms is maintaining one or more loop-free routes to a destination after arbitrary link or node failure in on-demand manner. Note that LR algoritms do not try to optimize routes.

The main idea of LR algorithms is completely ordering of nodes in a network for a given destination by assigning height to each node, and set a direction to each link according to the relative heights of adjacent nodes. By initially assigning the lowest height to the destination, LR algorithms can create a *destination oriented directed acycle graph(DAG)* that is rooted at the destination node. As following the definition of [2], a DAG is destination oriented if for every node there exists a directed path originating at this node and terminating at the destination. As long as a node has at least one outgoing link, it is guaranteed that it has loop-free routes to destination. When a node detects the event that it loses its last outgoing link toward the destination, a localized reaction - reverse directions of links - is initiated. Two link reversal algorithms were proposed: *full reversal* and *partial reversal*. In full reversal algorithm, each node except the destination reverses the directions of all of its incoming links at each iteration. On the contrary, in partial reversal algoritm, each node v_i except the destination maintains a list of its neighbor nodes v_j that have reversed the direction of the corresponding links. At each iteration, each node v_i reverses the direction of its links $e_{i,j}$ for all j that is not in the list and empties the list. If there is no such neighbors, then a node v_i reverses all of its incoming links and empties the list.

LR algorithms provide a lot of important and useful properties in MANETs such as multiple redundant paths to destination, localized maintenance of routes,

loop-free routes, etc. However, LR algorithms may cause significant detour of routes because it does not take route optimality into account. Furthermore, LR algorithms assume that there is no network partitioning. Because LR algorithms do not have any terminating condition of searching procedure for a new path when a network partition occurs or when a destination permanently leaves the network, LR algorithms may generate infinite route update messages until it constructs destination-oriented DAG. This behavior of LR algorithms can be viewed as another type of count-to-infinity problem of DBF algorithm.

2.3 Table-driven Routing Protocols

Table-driven routing protocols attempt to maintain consistent, up-to-date routing information from each node to every node in th network like conventional routing protocols used in wired networks. Each node maintains one or more tables to store routing information and propagates route updates throughout the whole network to maintain a consistent network view.

2.3.1 Destination Sequenced Distance Vector Routing(DSDV)

Destination sequenced distance vector(DSDV) routing protocol [27] that was proposed by C. E. Perkins and P. Phagwat in 1994 is a typical protocol in the table-driven category. DSDV is based on the classical DBF routing algorithm where each node maintains a table that contains the shortest distance and the first node on the shortest path to every other node in the network. DSDV uses increasing sequence number tags to prevent loops, in order to address the count-to-infinity problem and for faster convergence when it updates distance vector tables. Because DSDV is a table-driven routing protocol, each node always knows routes to all destinations at every node a priori at all time. Each node maintains a view of the network topology with distance vector table that has the num-

ber of hops to all possible destinations and exchanges the distance vector table between neighbors periodically to keep consistent and up-to-date view of the network topology. The tables may be forwarded to other nodes in the network if a node observes a significant changes in the local topology. There are two types of routing table update methods: incremental updates and full dump. Incremental updates are used when a node sees significant topological changes. Full dumps are used when a node sees significant topological changes or incremental updates are not able to use.

Routes in DSDV to all destinations is available at all time that implies almost no delay is involved when a source node seeks route to a destination.In addition, DSDV is an adaptation version of conventional routing protocol for wired networks, it is easy to migrate existing wired network protocols. However, since the updates should be propagated through the whole network in order to maintain up-to-date consistent view of the network topology at all node at all time, DSDV may generate heavy control traffics. Therefore, DSDV suffers from excessive control overhead that is proportional to the number of nodes in the network, and is not scalable for MANETs. Furthermore, it may take a long time to converge because changes in routing information need to be propagated through the whole network. Finally, DSDV provide only a single route to each destination.

2.3.2 Wireless Routing Protocol(WRP)

The wireless routing protocol(WRP) [28] that was proposed by S. Murthy and J. J. Garcia-Luna-Aceves in 1995 is also an table-driven routing protocol that inherits the properties of the DBF algorithm. In order to address count-to-infinity problem and fast convergence, WRP employs its own method of maintaining information regarding the shortest distance to every destination node in the network and the penultimate hop node on the path to every destination node. As other table-driven routing protocols, WRP also maintains up-to-date consistent view

of the network. The main differences between DSDV and WRP are table maintenance and update mechanisms. While DSDV maintains only one destination vector table, WRP uses a set of tables - distance table, routing table, link cost table, and a message retransmission list - to maintain more accurate view of network topology. To ensure accurate routing information, WRP periodically sends update message to neighbors. The update message contains a list of updates (the destination, the distance to destination, the predecessor of the destination), as well as a list of responses indicating which mobile should acknowledge the update. When a node receiving an update message, it not only updates the distance from transmitted neighbors but also checks the other neighbors' distance, hence convergence is much faster than DSDV since WRP incorporates predecessor information. A node can detect a link broken event by the number of missing update messages. When a node detects a link failure, it sends update messages to its neighbors, and neighbors that receives the update message will modify their distance table entries and check for new possible paths through other nodes.

Since WRP is also a table-driven protocol that is based on the DBF algorithm, it has the same advantages of DBF. In addition, it converges faster than DSDV and involves fewer table updates. However, computation complexity of maintenance of multiple tables demands a larger memory and more computing power for each node. Furthermore, the control overhead that is involved in updating table entries is almost the same as that of DSDV, hence WRP is not scalable and not suitable for highly dynamic environments. In addition, WRP provides only a single routes to each destination.

2.3.3 Clusterhead Gateway Switch Routing(CGSR)

Clusterhead gateway switch routing(CGSR) [29] that was proposed by C.-C. Chiang, H.-K. Wu, W.Liu and M. Gerla in 1997 is another table-driven routing protocol that tries to overcome shortcomings of DSDV. In order to over-

come shortcomings of other table-driven protocols that use flat network topology, CGSR incorporates a hierarchical network topology. CGSR organizes nodes into clusters that is defined as a group of nodes that can directly communicate with a cluster head that is elected within the cluster by employing a *least cluster change* [29] algorithm. There are three types of nodes in CGSR - *normal nodes*, *gateway nodes*, and *cluster head nodes*. Normal nodes belong to a cluster and can directly communicate with the cluster head of the cluster. Gateway nodes can communicate with two or more cluster heads that can relay packets from a cluster head to another cluster head. A cluster head controls a group of nodes.

Every member node maintains a routing table that contains the cluster-head for every node in the network. The routing table is broadcast periodically by each node using the DSDV protocol. Nodes that receive the table update message refresh its cluster member tables to ensure its validity. In addition to the cluster member table, each node maintains a routing table that keeps the list of next-hop nodes for each every destination. When a node receives a packet, a node can determine the next cluster head to the destination using its cluster member and routing tables. A packet sent by a node is routed to its cluster head, then the packet is forwarded to a gateway to another cluster head, and so on until the cluster of the destination node is reached. The cluster head of destination cluster transmits the packet to the destination.

Due to the hierarchical topology of CGSR, it can utilize better bandwidth and it is easy to implement priority scheduling scheme. However, path length of CGSR is increased compared to other table-driven protocols. Furthermore, routing is very instable if cluster head is frequently changed due to dynamicity of MANETs.The power consumption is also a significant problem in CGSR that cluster head and gateway nodes consumes much more battery power than normal nodes that leads to frequent changes of the cluster head.

2.3.4 Source-tree Adaptive Routing Protocol(STAR)

Source-tree adaptive routing protocol(STAR) that was proposed by J. J. Garcia-Luna-Aceves and M. Spohn is a table-driven routing protocol that does not require periodic routing updates, nor does it attempt to maintain optimum routes to destinations. In most routing protocols that try to provide optimum paths, the rate of routing updates is increases drastically if rate of topological changes of the network is increased. By contrast, STAR attempts to provide feasible paths that are not guaranteed to be optimal but involve much less control overhead. STAR attempts to minimize control overhead by (1) maintaining path information only for the destinations that the router needs to support, that is active routes, and (2) using the previously discovered paths as long as the paths are still valid, even if the paths are not optimum. In STAR protocol, each node maintains a *source-tree*. The set of links used by intermediate nodes is called a source tree of the router. By aggregating router's links to its neighbors and the source trees reported by its neighbors constitute a partial topology graph. Each node selects the next hop to each destination using its own source tree. Note that STAR differs from other link-state routing because it neither use nor send entire topology information of the network.

STAR generates low control overheads compared to other proactive protocols since it allows non-optimum routes. However, STAR also required to collect partial view of network topology that consumes a lot of bandwidth of the channel.

2.3.5 Optimized Link State Routing(OLSR)

The optimized link state routing protocol(OLSR) [30] that was proposed by T. H. Clausen, G. Hansen, L. Christensen and G. Behrmann in 2001 employs *multipoint relay* in order to forward link state packets efficiently. OLSR reduces the size of the control packets by declaring only a subset of the links in the link state

updates. OLSR also reduces the number of links that are used for forwarding the link state packets. This subset of links or neighbors are called multipoint relays that are designated for link state updates and are assigned the responsibility of packet forwarding. A multipoint relay node is a neighbor of corresponding node that has been designated to rebroadcast messages that received from the corresponding nodes. In order to implement efficient flooding algorithm of OLSR, each node must select a set of multipoint relay nodes that cover all nodes that are two hops away. The flooding algorithm is then modified that only multipoint relays rebroadcast. Note that, the routes that are constructed to reach each destination are restricted to the multipoint relay nodes. OLSR link state updates will be smaller than pure link state algorithms and each update message will be transmitted fewer times than pure link state algorithms. However, OLSR does not provide a scalable solution that can adapt well to very high rates of node mobility.

2.3.6 Hierarchical State Routing(HSR)

Hierarchical state routing(HSR) [31] that was proposed by A. Iwata, C. Chiang, G. Pei, M. Gerla and T. W. Chen in 1999 is a distributed multi-level hierarchical routing protocol that employs with membership management. The first level cluster is composed of nodes that are reachable in a single hop. The next level is clustered among the nodes that are elected as a leader of each of the first level clusters. Each node maintains information about its neighbors and corresponding links. These information is periodically broadcast within the cluster. Elected leader of the cluster exchanges these information among its peers in the neighborhood clusters using the next higher level clustering. Cluster leader exchanges topology information among cluster leaders, then it floods the information to the lower levels cluster leaders. HSR uses hierarchical addressing scheme that includes hierarchical ID and node ID where hierarchical ID is sequence of

IDs of cluster leaders of all levels starting from the highest level to the current node.

HSR can reduce routing table size due to its hierarchical structure. However, election of leader may increase overhead of routing protocols. In addition, HSR can not discover optimal path from source to destination.

2.3.7 Fisheye State Routing(FSR)

Fisheye state routing(FSR) [32] is a generalization of the global state routing(GSR) [33] protocol. FSR uses the fisheye technique to reduce routing overhead. Because a fish's eye only can see clearly near its eye's focal point. However, accuracy of fish's eye decreases if the distance of the object is far away from its focal point. This property is adapted to FSR that it collects accurate topology information of local area but not-so-accurate information about far-away nodes. The complete topology information of the network is maintained at each node. Each source node can compute the shortest path using the topology information that each node maintains. These topology information is exchanged periodically because instability of wireless links may cause excessive control overhead when event-driven is employed. The message size of a topology information packet is significantly reduced due to the removal of topology information regarding the far-away nodes.

FSR is suitable for large and highly mobile MANETs due to its multi-level scopes. However, performance of FSR is dependent on scope size and mobility value that is difficult to estimated apriori. Furthermore, due to not-so-accurate information of topology for far-away nodes, FSR can not discover optimal path to destination.

2.4 On-demand Routing Protocols

On-demand routing algorithms try to find valid routes only when a source node wishes to send packets. The major shortcomings of on-demand routing protocols are the possibility of significant delay during the route discovery procedure. Since route discovery is based on flooding of query messages in most reactive algorithms, it is also very costly. In real-time applications, excessive delays during route discovery and maintenance may lead to deadline misses.

2.4.1 Dynamic Source Routing(DSR)

Dynamic source routing [34] protocol that was proposed by D. B. Johnson and D. A. Maltz in 1996 is an on-demand routing protocol that implements source routing in MANETs. The major difference between this and the other on-demand routing protocols is that it is *beacon-less*.In order to find a path to its destination to establish a route, a source node floods a route request messages into the network. When the destination node receives the request message, it replies to the source with the route that the received request packet passed through. The route reply message traverses the path that the request message traversed. When source node receives a route reply message, it can send data packets along the route that the reply message specified. When a route is broken, a route error message is generated from the node adjacent to the broken link to inform the source node. Then source node reinitiates the route discovery procedure.

DSR can use the route cache at intermediate nodes that an intermediate node replies to the source node when it has a route to the corresponding destination. The cached entries at the intermediate nodes and the source node are removed when a route error message is received. Source node reinitiates the route discovery procedure again when it receives an route error message.

DSR does not need to broadcast route update messages periodically. In addi-

tion the intermediate nodes can utilize the route cache information efficiently to reduce the control overhead. However, DSR can not repair broken links locally. In addition stale route cache information could result in inconsistencies during the route reconstruction phase. Furthermore, the performance of DSR degrades rapidly with increasing mobility and it suffers from a scalability problem because each data packet sent by a source has to contain complete routing information and the size of a control message increases every time it visits an intermediate node. Finally, DSR provides only a single path to destination.

2.4.2 Ad-hoc On-demand Distance Vector(AODV)

Ad-hoc on-demand distance vector(AODV) [35] that was proposed by C. E. Perkins and E. M. Royers in 1999 is an on-demand approach that is widely accepted as routing protocol. AODV employs destination sequence number to identify the most recent path. Route discovery is the very similar to DSR that a route request message is flooded in the network by a source node, and destination node replies a route reply message to the source node. Since each intermediate node that receives a route reply message stores route information in its route cache, it can generate a route reply message when it has a valid route to the destination. AODV uses a destination sequence number in order to determine an up-to-date path information to the destination. A node updates its path information only when the destination sequence number of the current packet is greater than stored destination sequence number that received last. AODV checks validity of the route to the destination by comparing sequence number at the intermediate node with the destination sequence number of the route request message. While DSR uses source routing that a data packet carries the complete path to be traversed, in AODV, a source node and intermediate nodes store the next-hop information to a destination node. Therefore, AODV can overcome the shortcoming of DSR: source routing. If a node wishes to find a path to a

destination that has previously been determined by another node, it still needs to initiates route discovery by flooding a route request message. Note that flooding is a very costly operation that has to be undertaken even if intermediate nodes have cached route information. Link breakage is determined by observing the periodical beacons or through link-level acknowledgment. When a route becomes disconnected, then corresponding intermediate node notifies route breakage by setting hop count to ∞ of a route reply message.When the source node receives the route reply message with ∞ hop count, it reinitiates route discovery procedure again.

When an intermediate node detects a disconnected link, it can itself initiate a local route discovery procedure in several derivated AODV protocols [36, 37, 38] However, if it fails, additional time is required (when compared to source-initiate route discovery only) because the intermediate node reports the failure of local route discovery to the source node only after failure of local discovery is detected by the intermediate node. AODV also provides only a single path to each destination. However, ad hoc on-demand multipath distance vector(AOMDV) [39] routing, which is an extension of AODV, and other derived AODV protocols [40, 41] can provide multiple paths to each destination. Nevertheless, they still retain the other shortcomings of AODV.

2.4.3 Temporally Ordered Routing Algorithm(TORA)

Temporally ordered routing algorithm(TORA) [42] that was proposed by V. D. Park and M. S. Corson in 1997 is an on-demand routing algorithm that supports multiple paths to each destination. The basis of TORA is the LR algorithm. By assigning the lowest height to the destination, the algorithm creates a directed acyclic graph(DAG) rooted at the destination. To build a destination-oriented DAG, each node only needs to maintain routing information of adjacent nodes. Route discovery procedure is similar to other on-demand routing proto-

cols that flood route query message in the network. However, the route reply procedure builds a destination-oriented DAG that establishes multiple paths to the destination. The key idea of TORA is that there is a high probability that at least one path still exists to destination even if there were multiple topological changes since a destination oriented DAG has a loop-free route toward destination as long as each node has at least one outgoing link as proven in [2]. Therefore, TORA does not react to the topological changes until a node loses its last outgoing link. A localized reaction procedure is initiated that iteratively reverses the links of nodes whose paths are affected by this link failure until new routes are established when a node loses its last outgoing link. Therefore, TORA is very scalable and reactable routing algorithm in highly dynamic network environments. In addition, TORA can detect network partitioning that was severe problem of LR algorithms as stated in Section 2.2.2 using *reference level* and *sub-level* concepts. Route update messages are reflected back to the node that initiate the route maintenance phase. When the initiator node of route maintenance procedure detects network partitioning on receiving reflected route update message, it erases paths to the destination. As TORA inherits LR algorithms, it also has advantages of LR algorithms such as multiple paths, localized maintenance, loop-free routing, etc. However, TORA does not take the route optimality into account. Therefore, repeated reconstruction of destination-oriented DAGs may lead to long detour paths.

2.4.4 Location-Aided Routing Protocol(LAR)

Location-aided routing(LAR) [43] protocol that was proposed by Y. Ko and N. H. Vaidya in 1998 is an on-demand routing protocol that utilizes the location information to improve efficiency by reducing control overheads. However LAR utilizes location information in discovering and maintaining routes, external devices that provide location information such as global positioning system(GPS)

is mandatory in LAR. LAR forwards control packets to two geographical regions: *ExpectedZone* and *RequestZone*. An ExpectedZone is the region in which the destination node is expected to be located using the information regarding its location in the past and its mobility information. When past information is not available, the entire network area is considered to be the ExpectedZone. The RequestZoneis a geographical region that the path-finding control packets are permitted to be propagated. The route request messages are forwarded in the RequestZone only, and are discarded by nodes outside of the zone. When path discovery is failed within the specified RequestZone, LAR attempts to discover a path again with increased RequestZone in order to account for mobility and error of location estimation. Note that, LAR uses flooding but its region is restricted to a small geographical region.

LAR reduces the control packets by limited flooding scheme of route discovery messages using geographical information. However, GPS is not widely adopted in mobile devices yet, and it costs too much. In addition, GPS devices do not work in indoor environments, tunnels, or forests because GPS devices need to have direct line-of-sight access to the satellites.

2.4.5 Associativity Based Routing Protocol(ABR)

Associativity based routing(ABR) [44] that was proposed by C. K. Toh in 1997 is an on-demand routing protocol that selects routes based on the stability of the wireless links. A link is classified as stable or unstable based on its link stability estimation method that is determined by counting the periodic beacon messages that a node receives from its neighbors. Each node maintains the number of its continuous received messages. In order to discover route to destination, a source node floods a route request message throughout the network if a route is not available in its route cache. Each intermediate node forwards the route request message. Destination node may receive several route request messages from its

neighbors because it keeps receiving the route request message for a certain time period in order to select the path that has the maximum proportion of stable links. If more than two paths have the same stable links, then the shortest path is selected. If a route is broken, an intermediate node that is closer to the source node that detects the disconnection, initiates a localized route reconstruction procedure by broadcasting a route repair packet locally, termed the local query broadcast, with limited time-to-live(TTL). By this way, a broken link can be bypassed locally without flooding of a new route request message. However, if localized route reconstruction procedure is failed, then its uplink node reinitiates the route reconstruction procedure again. This localized route reconstruction procedure continues until it traverses half the length of the broken path or the route is repaired. If attempts of localized route reconstruction are failed, then the source node reinitiates route discovery procedure by flooding a new route request message in the network.

Because ABR prefers stable links, probability of path breakage is reduced. Therefore, ABR may reduce the overhead of flooding of route request messages. However, ABR suffers detour that it can not provide short paths to destination nodes. In addition, repetitive localized reconstruction procedure may result excessive delay during route reconstruction.

2.4.6 Signal Stability-based Adaptive Routing Protocol(SSA)

Signal stability-based adaptive routing protocol(SSA) [45] that was proposed by R. Dube, C. D. Rais, K.Y. Wang and S.K. Tripathi in 1997 is an on-demand routing protocol that uses signal stability as the major factor to find stable routes. Each node maintains a signal stability table that estimated using signal strengths of receiving beacon messages from its neighbors. This table is used by the nodes in the path to forward the incoming route request message over strong links to discover a stable route. If it failed to discover stable routes to the destination

over stable links only, SSA floods a new route request message throughout the network without the link stability consideration. When a link breaks, the end nodes of the broken link notify it to the corresponding end node of the path, i.e., the source and destination nodes. Then source node initiates route discovery via rebroadcasting a new route request message over stable links to find another stable route to destination.

The main advantage of SSA is that it can discover more stable routes when compared to the shortest path route selection protocols such as DSR and AODV. However,route discovery procedure of SSA may fail frequently since it disallows to forward route request message over unstable links. Therefore SSA may flood route request messages twice. In addition, SSA suffers from detour of routes because it tries to find stable links only.

2.4.7 Flow-Oriented Routing Protocol(FORP)

Flow-oriented routing protocol(FORP) [46] that was proposed by W. Su and M. Gerla in 1999 is an on-demand routing protocol that employs a prediction-based multi-hop-handoff mechanism to support time sensitive applications in ad hoc wireless networks. The main objective of FORP is supporting QoS in IPv6 based MANETs. If a source node or intermediate nodes initiate the route maintenance procedure after they detect a link breakage, it may results high packet loss and low QoS to users. In order to address such a problem, FORP estimates link expiration time(LET) using the node mobility and location information. The minimum of the LET values of all wireless links on a path is termed as the route expiry time(RET). It is assumed that each node can predict the LET of each links based on information of current position of nodes, directions, mobility, and transmission ranges, etc. Therefore FORP also need location information. When a source wishes to discover a route to destination, it broadcasts a route request message that carries a flow identification number and a sequence num-

ber that are unique for every session. An intermediate node that receives this message checks the sequence number in order to avoid route loop. Then, it updates local information such as sequence number, it forwards the route request message after it appends its node address and the LET of the link onto the route request message. When a destination receives the route request message that is expected to have better RET value than current path, then the destination generates a route reply message.

When the destination node expects that a route break is about to occur within a specfic time, it generates a hand-off message to the source node. Procedure of handling the hand-off message is similar to the procedure of route discovery. When the hand-off message is arrived at the source node, it selects the best path and sends new messages to the destination.

As noted above, FORP highly depends on the location services.Furthermore, estimation of LET is very difficult due to various reasons such as mobility, channel condition, etc. In addition, FORP does not provide shortest path to destination.

2.5 Hybrid Routing Protocols

In this section, we discuss the hybrid routing protocols that each node maintains the network topology information of several hops. This section includes zone routing protocol, core extraction distributed ad hoc routing protocol, and zone based hierarchical link state protocol.

2.5.1 Zone Routing Protocol(ZRP)

Zone routing protocol(ZRP) [47, 48] that was proposed by Z. J. Haas in 1997 is a hybrid routing protocol that combines the features of both proactive and reactive routing protocols. ZRP employs proactive routing scheme within a limited zone in the ρ-hop neighborhood of every node, and use a reactive scheme for

nodes beyond this zone, i.e., Intra-zone routing protocol is a table-driven routing protocol, and Inter-zone routing protocol is an on-demand routing protocol. In ZRP, each node maintains routing information to all nodes within its routing zone by periodic exchanges of routes update messages. If destination node is not in the zone of source node, then it bordercasts a route request message to its peripheral nodes. If any peripheral node of source has the destination node in its own zone, then it sends a route reply message to the source node. If not, each intermediate node rebordercasts the route request message.

When an intermediate node in an active path detects path broken, it initiates a localized reconstruction of routes in which the broken link is bypassed by means of short alternate path that results sub-optimal paths. In order to support optimal path, the sender should reinitiates the global path-finding process after a number of local configurations.

By combining advantages of table-driven and on-demand routing schemes, ZRP reduces the control overhead. However, in the absence of a query control, ZRP tends to generate higher control overhead than others. The query control must ensure that redundant or duplicate route request messages are not forwarded.

2.5.2 Core Extraction Distributed Ad Hoc Routing (CEDAR)

Core extraction distributed ad hoc routing(CEDAR) protocol that was proposed by P. Sinha, R. Sivakumar and V. bharghavan is a kind of QoS routing protocol. CEDAR is based on extracting core nodes that are approximate the minimum dominating set in the network. A dominating set of a graph is a set of nodes in the graph such that every node in the graph is either present in the dominating set or is a neighbor of nodes in the dominating set. CEDAR uses the core broadcast mechanism to deliver packets with mimimum number of involved nodes using dominant set of nodes. These nodes which is called core nodes take a

part in the core broadcast mechanism. Each core node maintains local topology information of its neighborhood. When a node wishes to discover a QoS path to a destination, it generates a route request message if the destination is not in the local topology table of its core node; otherwise, the path is immediately established using local topology information. Core nodes forwards the route request message to its neighbor core nodes if the destination is a member of the core node. A core node that has the destination as a core member replies to the source core. Once core path is discovered, QoS path is chosen.

CEDAR attempts to repair a broken path locally. If a route is broken, an intermediate node that is closer to the source that detects the disconnection sends a notification message to the source node and initiates route discovery procedure from itself to the destination node locally. Until the route recomputation, it drops every subsequent packet. By the way, source node that receives the route breakage notification immediately stop transmitting packets to the corresponding flow in order to prevent packet losses, and start discovering a new route to the destination.

CEDAR performs both routing and QoS path communication using core nodes. However, the movement of the core nodes severely affects the performance since route computation is carried out at the core nodes only. In addition, the core update information could cause a significant amount of control overhead.

2.5.3 Zone based Hierarchical Link state Routing (ZHLS)

Zone based hierarchical link state routing protocol(ZHLS) [49] that was proposed by M. Joa-Ng and I. T. Lu in 1999 is a hybrid hierarchical routing protocol that uses the geographical location information of the nodes. Similar to ZRP, ZHLS also employs a proactive approach inside the geographical zone and a reactive approach beyond the zone. In ZHLS, in order to assign zoneID to each node, GPS or similar infrastructure is required. The intra-zone routing table is main-

tained as shortest path on the node-level topology of the zone that is obtained by intra-zone clustering mechanism. The nodes that receive link responses from a node that is outside of the zone are called gateway nodes that are used for inter-zone routing. The zone level topology is constructed using zone link state packets. If a given gateway node moves away that cause a zone-level disconnection, routing can still take place with the help of the other gateway nodes due to hierarchical addressing.

Due to hierarchical approach, ZHLS can reduce the storage requirements and communication overhead. However, ZHLS generates additional overhead to create the zone-level topology. In addition, ZHLS supports only sub-optimal paths to the destination. Finally, ZHLS is strongly dependent on location information which is not available in all environments.

2.6 Summary

This chapter presented an overview of the major issues of designing of routing protocols and previously proposed routing protocols. At first, major challenges of routing protocol in MANETs are discussed such as mobility of nodes, limited bandwidth, hidden and exposed terminal problems, limited battery power, etc. Earlier version of routing protocols are reviewed such as distributed Bellman-Ford algorithm and link reversal algorithms. The former is very well-known and widely adopted routing protocol but it has a *count-to-infinity* problem and route loops. The latter is the first on-demand routing protocol for highly dynamic environments but it also had a similar problem of count-to-infinity in DBF algorithm that when network partition occurs.

Next, The proactive routing protocols are reviewed such as DSDV, WRP, CGSR, STAR, OLSR, HSR and HSR. Even protocols fall into this category address route loop and count-to-infinity problem, they still need large overhead due

to periodic updating of route information. Beside, the reactive routing protocols such as DSR, AODV, TORA, LAR, ABR, SSA and FORP are reviewed in this chapter. Even these protocols may require fewer overhead than protocols in proactive category, they still did not address all issues discussed in Section 2.1 such as providing multiple paths, shortest path as possible, long route discovery time, etc. The hybrid routing protocols are also reviewed: ZRP, CEDAR and ZHLS. These routing protocols are combines advantages of two categories but they also introduce additional overhead to generate hierarchy or zone topology.

In order to address as many issues as possible that described in previous section, hybrid type of routing protocol is required. The proposed routing algorithm that support multiple paths to destination that all nodes in the network maintains paths to the destination, localized maintenance for fast recovery of route failure, short route discovery when a node find a path to the destination, and so on is discussed in the next chapter.

3

Pseudo-Distance Routing

Because proactive routing protocols maintain routing tables that can be used immediately, they have almost zero route discovery time if we ignore route computation time, but they require a lot of control traffics in order to maintain up-to-date routing tables. On the other hand, reactive routing protocols have the possibility of significant delay during the route discovery and maintenance phases because they do not maintain up-to-date routing tables. Since route discovery and maintenance is based on flooding of query and update messages in most reactive algorithms, it is also very costly. In real-time applications, excessive delay during route discovery and maintenance may result in deadline misses. Therefore routing protocols for MANETs should be a hybrid of both proactive and

reactive routing properties. The proposed routing protocol, referred to *pseudo-distance routing*(PDR)[1] algorithm is proposed to support quick discovery of route by maintaining a route table in on-demand manner. In addition to the hybrid properties of both proactive and reactive protocols, PDR also has following properties: fully distributed operation, minimized control overheads, minimal processing overheads, processing using only neighbor's information, loop and stale free routing, multiple paths to destination, quick convergence by avoiding flooding of control traffics as much as possible, localized maintenance of routes, supporting minimal paths, and scalable operations.

3.1 Preliminaries

This section provides the background information that is required to understand the proposed routing algorithm. Since PDR is a kind of link reversal algorithm, LR algorithms are reviewd deeply again in this section.

3.1.1 Notation

A network is modeled as an undirected graph $G = (V, E)$, where V is a finite set of nodes and E is a set of bidirectional communication links at a given time instant. A (src, dst)-path is a finite sequence of nodes $P = (src = v_0, v_1, \cdots, v_i, \cdots, v_n = dst)$ such that for all $0 \leq i \leq n$, $e_{i,i+1} \in E$ and $v_i \neq v_j$ for all $v_i, v_j \in P$. $n = |P|$ is the length of P, which is the number of nodes in the path P excluding the source node. $D_{i,j}$ is the distance from v_i to v_j, which is the shortest length among all possible (v_i, v_j)-paths. Finally, a neighbor set of node v_i, written as N_i, consists of all nodes that have a bidirectional link to node $v_i \in V$.

[1] Preliminary version of this chapter was published in [50].

3.1.2 Link Reversal Algorithms Revisited

As discussed in the Section 2.2.2, link reversal(LR) algorithms achieved following very important contributions: (1) on-demand routing scheme to minimize reactions to the topological changes due to the node mobility, (2) localized maintenance enabled by fully distributed operations, (3) multiple paths and quick convergence by laying aside minimal path in consideration as described in [51]. In order to achieve above contributions, LR algorithms build and maintain *destination-oriented* directed acyclic graph(DAG) that is defined as a directed acyclic graph(DAG) is destination-oriented if, for every node, there exists a directed path originating at that node and terminating at the destination [2]. *Destination-disoriented* DAGs are DAGs that are not destination-oriented. Figure 3.1 shows an example of a destination oriented DAG. A Destination-oriented DAG provides redundant routes to the corresponding destination as shown in Fig. 3.1. For instance, v_3 has two routes to destination that $P^1_{3,Dest} = (v_3, v_5, v_6, v_{Dest})$ and $P^2_{3,Dest} = (v_3, v_4, v_6, v_{Dest})$. Note that there is no route loop as long as the destination-oriented DAG is maintained because destination oriented DAG is an acyclic graph.

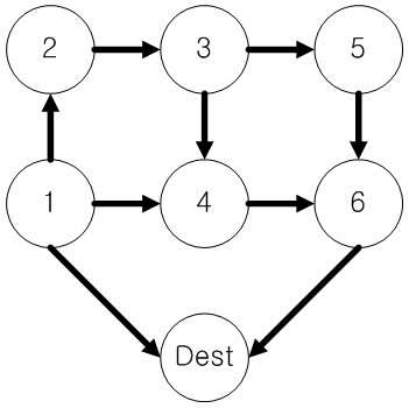


Figure 3.1: An example of a destination-oriented DAG (adopted from [2]).

There can be very many algorithms that map a given undirected network to the destination-oriented DAG. One of simple and effective algorithm is totally order the nodes of a network with respect to a given destination by assigning height to each node using distance to the corresponding destination, and set a direction to each link according to the relative heights of adjacent nodes as proposed in [2].

It is easy to see that a connected DAG is destination disoriented if and only if there exists a node that has no outgoing links except the destination. Note that there exists at least one routes to destination from all nodes in the network if the graph is destination-oriented. LR algorithms initiate route update phase only when a node detects that the DAG becomes destination-disoriented because it mainly concerns adaptability to the dynamic changes of the network. Now, consider the problem that transform a given connected destination disoriented DAG into a destination-oriented DAG. In LR algorithms, two methods are proposed to solve above problem: *full reversal* and *partial reversal*. In full reversal algorithm, each node that does not have any outgoing links except the destination reverses the directions of all its incoming links at each iteration. An example of the full reversal algorithm is provided in Figure 3.2. Suppose that a link $e_{6,Dest}$ is broken that results node v_6 loses its last outgoing link as shown in (a). Then, v_6 reverses all of its incoming links as shown in (b). Consequently, v_4 and v_5 lose their last outgoing links. Therefore v_4 and v_5 also reverse all of their links. v_6 loses its last outgoing links again and v_3 loses its last outgoing links as shown in (c) due to reversed links by v_4 and v_5 at earlier stage. Consequently, v_3 and v_6 also reverse their links, then v_2 loses its last outgoing links as shown in (d). Finally, v_2 reverses all of its links as shown in (e) that all nodes have at least one outgoing links toward destination that meets the definition of destination oriented DAG. Note that there is undue overhead for reversing already reversed links in full reversal algorithm such as v_6 loses its outgoing links again when v_4

and v_5 reverse their links in (c) of Fig. 3.2.

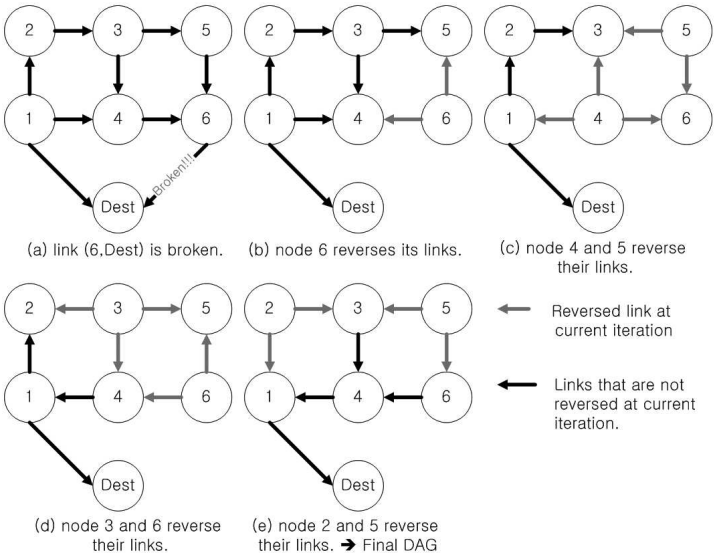


Figure 3.2: An example of the full reversal algorithm (adopted from [2]).

In partial reversal algorithm, each node v_i except the destination keeps a list of its neighbor nodes v_j that have reversed the directions of the corresponding links. At each iteration, each node v_i reverses the directions of its links $e_{i,j}$ for all nodes $v_j \in N_i$ that are not in the list and empties the list. If there is no such neighbor node v_j, then v_i reverses all of its incoming links and empties the list. Figure 3.3 shows an example of the partial reversal algorithm in detail. Suppose that a link $e_{6,Dest}$ is broken in (a) of Fig. 3.3 as same as in the full reversal example. Because the reversed link list of v_6 is empty, it reverses all of its links that results no outgoing links at the node v_4 and v_5 as shown in (b). Note that v_4 and v_5 store that the link to v_6 is reversed in the reversed link list. Then, as shown in (c), v_4 reverses its links $e_{4,3}$ and $e_{4,1}$ except $e_{4,6}$ which is stored in the reversed link list. v_5 also reverses the link $e_{5,3}$ except $e_{5,6}$ because $e_{5,6}$ are in the reversed link list. Note that v_6 does not need to reverse its links since $e_{4,6}$ and

$e_{5,6}$ are not reversed in partial reversal algorithm that was reversed in the full reversal algorithm. In (d), v_3 reverses its links except $e_{3,5}$ and $e_{3,4}$ due to the same reasons described above. Finally, as shown in (e), v_2 reverses its link $e_{2,1}$ only, then the graph becomes a destination-oriented DAG.

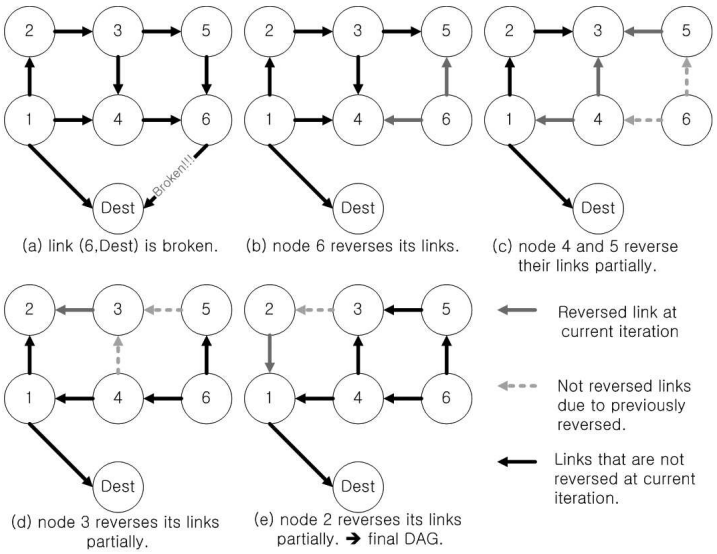


Figure 3.3: An example of the partial reversal algorithm (adopted from [2]).

As shown above example, LR algorithms provide multiple redundant paths to destination, localized maintenance of routes, loop-free routes, etc. Therefore, LR algorithms can quickly adapt to the changes of highly dynamic environments. However, LR algorithms have a heavy assumption that the network should be a connected graph during its life time. Because LR algorithms do not have any terminating conditions of the searching process for a new path to the destination, route search phase is processed unlimited iterations when a network partition occurs or when the destination leaves the network as described in Sect. 2.2.2.

TORA is also a kind of link reversal protocol that incorporates the partial-reversal algorithm which is a part of LR algorithm into MANETs that can detect

network partitioning using the *reference level* and *sub-level* concepts. Because TORA is a modified version of the partial reversal algorithm, it inherits advantages of partial reversal algorithm. However, TORA also does not take route optimality into account as the partial reversal algorithm, it has a severe shortcomings of long detour of routes by repeated reconstruction of destination-oriented DAGs as described in Sect. 2.4.3.

In order to overcome the main shortcoming of LR algorithms and TORA (the possibility of long detour routes), PDR employs a pseudo-distance concept instead of the "height" in LR algorithms and TORA. The key idea of PDR is selective reverses of links using pseudo-distance concept. While TORA LR reverse links without consideration of any topological information, PDR reverses selected links that are expected to have shorter distance and minimize the number of control messages. Furthermore, if a node initiates a route discovery phase for a destination, then all affected nodes in the network can maintain valid paths to that destination, thereby enabling fast route discovery for that destination during future searches that is only partially supported in TORA. Note that PDR also able to detect network partitioning as TORA, which is the major shortcoming of LR algorithm.

3.1.3 Assumptions

PDR assumes that all links in MANETs are bidirectional. Actual links in MANETs are not bidirectional. However, by building a routing algorithm on top of other protocols that support bidirectional communication, as in the Internet MANET encapsulation protocol (IMEP) [52].Furthermore, communication links of MANETs are very unreliable. Due to the unreliability of MANETs, some control messages may be lost during route discovery or maintenance phases. Because IMEP also provides reliable communication links using an ACK scheme, it is able to overcome the loss of control messages. Therefore we can achieve

reliable, bidirectional communication links using an IMEP layer. PDR assumes that detection of neighboring nodes is performed by an underlying layer such as MAC or other layers. Note that IMEP also provides connectivity information to the upper layer protocols using beacon messages, which is answered by each node hearing it with a hello message. PDR assumes a broadcasting network since most off-the-shelf wireless ad hoc devices are omni-directional (IEEE 802.11, bluetooth, etc.). PDR also assumes that each node has its own unique identifier(like a MAC address). Finally, PDR assumes that all nodes in the network is timely synchronized in order to detect network partition. Note that if PDR assume connected graph as LR algorithms, this assumption is not required.

3.2 Pseudo-Distance

As stated above, PDR adopts the destination-oriented DAG in [2]. In order to transform a given network graph $G = (V, E)$ into a destination-oriented DAG, each node needs to have its own height value. Unlike TORA and LR algorithms, a height in PDR is not a value representing temporal order but a *pseudo-distance* to the destination. The height of a node v_i relative to a node v_j is written as $H_{i,j} =< \lambda, -\alpha, -\beta >$. A pseudo-distance λ is a distance metric between a node v_i and v_j. α is the number of neighbors that have lower λ values than v_i and β is the number of neighbors that have the same λ value.

α represents the number of neighbors that are expected to be closer than the current node to the destination node and β represents the number of neighbors that are expected to be the same distance as the current node to the destination node. α and β are used to find a next-hop node, preferably closer to the destination, that has as many different paths to the destination as possible. Forwarding a packet to a neighboring node that has the largest α or β values should increase the number of possible redundant paths. Clearly, it is better to forward a packet

to neighbor that has more possible paths to the destination. Thus, PDR forwards packets to a neighbor with the largest α value, breaking ties with β values if possible.

PDR compares the heights of two adjacent nodes lexicographically. In the height metric, a minus sign is prepended to α and β. This is done to permit simple lexicographic comparisons. Then, in order to forward packets toward the destination, each node simply selects a neighbor with the minimum height as its next hop. Thus, all neighbors with the smallest λ value are considered first. Among all such neighbors, nodes with the smallest $-\alpha$ value are considered next. Then, among of these neighbors, nodes with the smallest $-\beta$ value are considered as next-hop candidates. If there is only one candidate remaining, it is chosen as the next-hop node. If there are still multiple candidates remaining, then the candidate with the lowest ID value is arbitrarily chosen as the next-hop node. To reduce the number of control messages required, temporarily incorrect α and β values are permitted since small deviations in the number of alternate subpaths are not catastrophic.

Two types of links are identified when using the pseudo-distance concept. Primary links are mainly used to route packets along paths that are as short as possible. Auxiliary links are used only when all primary links are broken. If the λ value of any two adjacent nodes are different, then the link is primary; otherwise, the link is auxiliary. When a node loses its last primary link, it need to check whether it has auxiliary outgoing links or not. If auxiliary routing is turned on, then the node does not update its height (in order to reduce the number of control messages being sent). However, if auxiliary routing is turned off, it tries to replace auxiliary links with primary links by increasing its pseudo-distance. Note that primary links are expected to reduce the distance toward the destination but auxiliary links are not. Therefore forwarding a packet along auxiliary links may introduce extra-hop detours in the path to the destination.

There are four cases to be considered when setting the directions of links. For any two neighbor nodes v_i and v_k with destination v_j,

- if $\lambda_{i,j} > \lambda_{k,j}$, then v_i sets its link $e_{i,k}$ as primary outgoing and v_k sets $e_{k,i}$ as primary incoming.
- if $\lambda_{i,j} = \lambda_{k,j}$ and $-\alpha_{i,j} > -\alpha_{k,j}$, then v_i sets its link $e_{i,k}$ as auxiliary outgoing and v_k sets $e_{k,i}$ as auxiliary incoming.
- if $\lambda_{i,j} = \lambda_{k,j}$, $-\alpha_{i,j} = -\alpha_{k,j}$ and $-\beta_{i,j} > -\beta_{k,j}$, then v_i sets its link $e_{i,k}$ as auxiliary outgoing and v_k sets $e_{k,i}$ as auxiliary incoming.
- if $\lambda_{i,j} = \lambda_{k,j}$, $-\alpha_{i,j} = -\alpha_{k,j}$, $-\beta_{i,j} = -\beta_{k,j}$ and $i > k$, then v_i sets its link $e_{i,k}$ as auxiliary outgoing and v_k sets $e_{k,i}$ as auxiliary incoming.

As described in [42], to build a destination-oriented DAG, PDR requires only local (neighbor) routing information (as in LR algorithms and TORA). Suppose that a node v_j is the destination. A node v_i should collect $H_{k,j}$ for all k where $v_k \in N_i$ in order to properly set all links $e_{i,k}$.

Figure 3.4 shows an example of a destination-oriented DAG, with destination node v_6, using the pseudo-distance concept. The numbers in the vertices represent unique identifiers for each node and the numbers beside the vertices are height values $H_{i,6}$ for each node. Solid arrows represent primary links and dotted arrows represent auxiliary links. Note that the pseudo-distance between two nodes is a multiple of δ, which is the default difference in λ between adjacent nodes.

To send packets, each node v_i simply selects a node $v_k \in N_i$ that has the smallest $H_{k,6}$. In Fig. 3.4, v_1 would select v_3 as its next hop because lexicographically $H_{3,6} < H_{2,6}$. Suppose that v_5 wishes to send a packet to destination v_6. In this case, pseudo-distances of both v_2 and v_7 are the same and both $e_{5,2}$ and $e_{5,7}$ are primary links. However, v_5 should be able to select v_2 because it

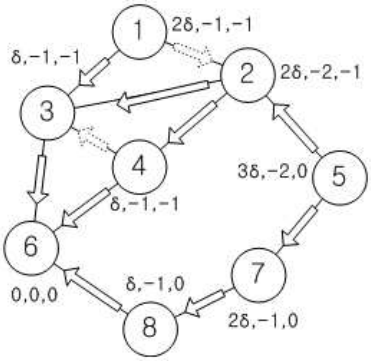


Figure 3.4: An example of a destination-oriented DAG using pseudo-distance concept for destination node v_6.

has more remaining paths to the destination node. v_5 selects v_2 as the next hop because lexicographically $H_{2,6} < H_{7,6}$. Suppose instead that v_5 selects v_7 as the next hop. In this case, if links $e_{7,8}$ or $e_{8,6}$ are broken, then the packet sent from v_5 will fail to be delivered to destination v_6. However, if v_5 selects v_2, then several failure(s) of links, such as $e_{2,4}$, $e_{4,6}$ and others, can be tolerated. Because the heights of both v_3 and v_4 are the same, i.e., $H_{3,6} = H_{4,6}$, and the identifier of v_4 is greater than v_3, v_4 sets $e_{3,4}$ as its auxiliary outgoing link.

3.2.1 Control Messages

Initially, every node except the destination sets its height to NULL. The destination node sets its height to zero. In order to assign and maintain height values, four types of control messages are defined.

- QRY is a route query message that is triggered when a node wishes to send a packet to a destination node. QRY is defined as $< DST, ORI, SEQ >$ where DST is the destination, ORI is the source node that initiates the route discovery phase and SEQ is a sequence number used to distinguish

QRY messages. QRY is forwarded toward the destination or an intermediate node that has a non-null height value.

- REP, which contains pseudo-distance information, is the reply message for a QRY. REP is defined as $< DST, H, SEQ, ORI >$ where H is the height of the node that sends the REP. REP is triggered by a QRY if a node has valid route information (actually, a node that has a non-null pseudo-distance) for the destination node. REP may also be generated by another REP to forward routing information.

- UPD is a route maintenance message used to reflect topological changes in a MANET. UPD is defined as $< DST, H, SEQ, ORI, r, \tau >$ where r is one bit flag that represents reflection of route updates, τ is the time that changes of network topology was detected at the ORI node. UPD is triggered when a local pseudo-distance λ value is changed. Note that for efficiency, in case only α or β values are changed, an UPD is not triggered because temporal inconsistencies in α and β values can be tolerated. The pseudo-distance λ is changed locally when a node loses all of its outgoing links.

- CLR is a route erasure message used to erase routes when network partition is detected. CLR is defined as $< DST, ORI, SEQ >$. CLR is triggered when a node that initiates UPD receives reflected UPD packet. When a node receives a CLR, then it erases its route information to the corresponding destination, then relays the CLR to its neighbors.

Note that PDR assumes that MANETs are broadcast networks. Thus, if v_i sends a control message, all nodes $v_k \in N_i$ can listen to this message and perform the required operations as described in later sections.

The algorithm for assigning pseudo-distance values consists of three phases. The first (route discovery) builds a destination-oriented DAG by assigning a

height to each node and the second (route maintenance) maintains the destination-oriented DAG whenever there are topological changes. Finally, when network partition is detected, then route erasure phase removes route information to the corresponding destination.

3.2.2 Route Discovery Phase

Each node should maintain a route requested flag, rr_j toward destination node v_j that has initial value as false. When a source node v_i wishes to send data packets to a destination node v_j, v_i first checks its height $H_{i,j}$. If $H_{i,j}$ is null, then v_i initiates route discovery phase by broadcasting QRY=$< v_j, v_i, qryseq_i >$ to its neighbors. After broadcasting QRY, v_i increases its own $qryseq_i$ value by 1 and set its rr_j to true. Figure 3.5 shows the pseudocode of procedure executed when a node v_k receives a QRY message from its neighbor node. An intermediate node v_k that receives a QRY from v_i rebroadcasts QRY=$< v_j, v_i, qryseq_i >$ if $H_{k,j}$ is also NULL and rr_j is false (lines 12–13 of Fig. 3.5). Then v_k should updates its rr_j to true as line 8 of Fig. 3.5 in order to avoid multiple forwarding of QRY messages. An intermediate node may receive multiple QRYs from its multiple neighbors. In that case, it broadcasts only the first QRY and drops other QRYs using the rr_j flag that each node maintains.

When the destination node v_j receives a QRY from its neighbor v_m, it broadcasts REP=$< v_j, H_{j,j}, repseq_j, v_i >$ where v_i is the source node that initiated the route discovery phase and $H_{j,j} =< 0, 0, 0 >$ (lines 2–3 of Fig. 3.5). Figure 3.6 shows the pseudocode for the procedure executed when v_k receives a REP from its neighbor with destination v_j. When v_k receives the REP, it updates its neighbor's table with information from the REP (line 2 of Fig. 3.6). Then v_k updates its pseudo-distance to $\lambda_{k,j} = \mathrm{p}.H.\lambda + \delta$, where $\mathrm{p}.H.\lambda = 0$, and the corresponding α and β values because the previous height $H_{k,j}$ was null (lines 5–8 of Fig. 3.6). After v_k updates $H_{k,j}$, it broadcasts REP=$< v_j, H_{k,j}, repseq_j, v_i >$ to its neigh-

```
1.  recvQRY(message p) {
2.      if(p.DST = v_k) {
3.          sendREP(H_{k,k}, repseq_k++);
4.      } else if (H_{k,j}! = NULL) {
5.          sendREP(H_{k,j}, repseq_k++);
6.      } else if(rr_j is false){
7.          sendQRY(p);
8.          rr_j = true;
9.      }
10.     return;
11. }
```

Figure 3.5: Pseudocode for recvQRY.

bors (line 9 of Fig. 3.6). Note that although v_j also receives this REP message, it simply drops the message because v_j is the destination (lines 3–4 of Fig. 3.6). An intermediate node v_k may receive multiple REPs from its neighbors. Suppose that an intermediate node v_k receives multiple REPs from v_l first and v_m last. When v_k receives a REP from v_l, it updates its pseudo-distance as $\lambda_{k,j} = \lambda_{l,j} + \delta$ (lines 5–9 of Fig. 3.6) because the previous value of $\lambda_{k,j}$ was null. Afterward, when v_k receives a REP from v_m, it compares $\lambda_{k,j}$ to $\lambda_{m,j}$ (line 10 of Fig. 3.6). If $\lambda_{k,j} - \lambda_{m,j} > \delta$, then v_k updates its pseudo-distance to $\lambda_{k,j} = \lambda_{m,j} + \delta$ and updates the corresponding α and β values (lines 11–12 of Fig. 3.6) because the path through v_m is expected to be shorter than the path through v_l. After updating its height, v_k broadcasts the REP to its neighbors as described line 13 of Fig. 3.6. Note that δ is the default one-hop difference between two adjacent nodes.

Figure 3.7 shows an example of the route discovery with destination node v_1.

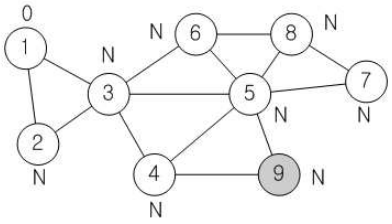
```
1.  recvREP(message p) {
2.      updateNeighbor(p);
3.      if(p.DST = v_k) {
4.          return;
5.      } else if (H_{k,j} = NULL) {
6.          λ_{k,j} = p.H.λ + δ;
7.          updateHeight();
8.          rr_j = false;
9.          sendREP(H_{k,j}, repseq_j);
10.     } else if (λ_{k,j} − p.H.λ > δ) {
11.         λ_{k,j} = p.H.λ + δ;
12.         updateHeight();
13.         sendREP(H_{k,j}, repseq_j);
14.     }
15.     return;
16. }
```

Figure 3.6: Pseudocode for recvREP.

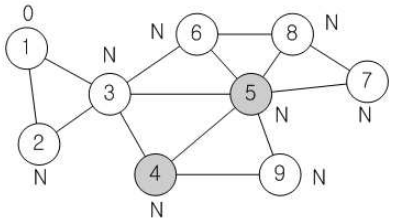
Shaded vertices represent nodes that broadcast QRY and slashed vertices represent nodes that broadcast REP. Solid arrows represent primary links, shaded arrows represent currently updated links and dotted arrows represent auxiliary links. Initially, in Fig. 3.7(a), v_9 triggers the route discovery phase by broadcasting QRY=$< v_1, v_9, 0 >$ to its neighbors. In (b), v_4 and v_5 forward the QRYs received (lines 6–9 of Fig. 3.5). In (c), v_2, v_6, v_7, v_8 also forward the QRYs received. In (d), v_1 broadcasts REP=$< v_1, H_{1,1}, 0, v_9 >$ (lines 2–3 of Fig. 3.5) because v_1 receives a QRY from v_2. In (e), v_2 and v_3 forward the REP received to their neighbors after updating their own routing tables with the new local pseudo-distance and corresponding α and β values (lines 5–9 of Fig. 3.6). Note that v_1 receives two REPs from v_2 and v_3, but simply drops the later messages (lines 3–4 of Fig. 3.6). In (f) and (g), other nodes keep forwarding REP to their neighbors and update their height values. Finally, (h) in Fig. 3.7 shows the resulting destination-oriented DAG directed toward v_1. Note that the pseudo-distances of the initial destination-oriented DAG are exactly proportional to actual distances.

3.2.3 Route Maintenance Phase

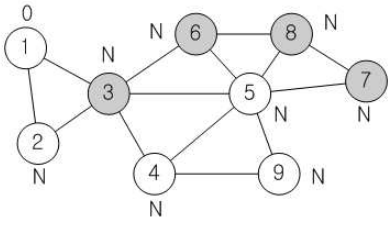
When a node v_k detects that a link is broken, it does not react if it still has corresponding outgoing links toward the destination in order to minimize control overheads. However, if v_k loses its last outgoing link toward destination node v_j, then v_k triggers a route maintenance phase. There are two options for the route maintenance phase as described earlier. In order to provide shorter routes, v_k can trigger a route maintenance phase when it loses its last primary outgoing link even if v_k still has auxiliary outgoing links. This routing protocol will be referred as PRI routing. On the other hand, in order to reduce the number of control messages required, v_k can trigger a route maintenance phase only when it loses all of its outgoing links, including auxiliary links. This routing protocol will be referred as AUX routing. Note that a node may lose its last outgoing link



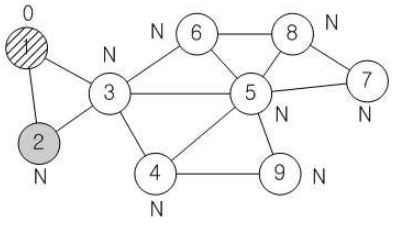
(a) v_9 generates a QRY=$< v_1, v_9, 0 >$.



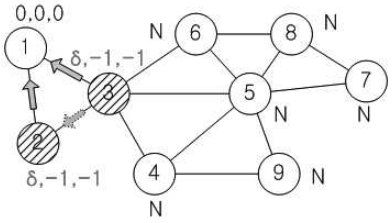
(b) v_4 and v_5 relay received QRYs.



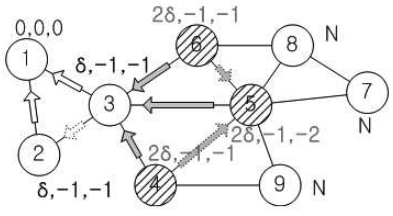
(c) v_3, v_6, v_7 and v_8 relay received QRYs.



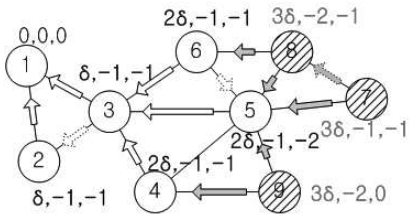
(d) v_3 relays received QRY while v_1 generates a REP=$< v_1, H_{1,1}, 0, v_9 >$.



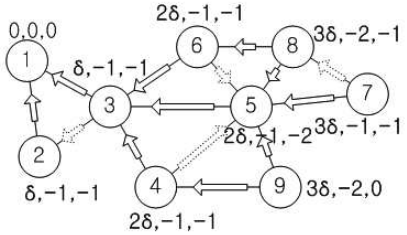
(e) v_2 and v_3 relay the REP generated by v_1 after updating their $H_k, 1$.



(f) v_4, v_5 and v_6 also relay the REP generated by v_1 after updating.



(g) v_7, v_8 and v_9 relay the REP generated by v_1 after updating.



(h) The final destination-oriented DAG of route discovery phase

Figure 3.7: An example of route discovery.

when all outgoing links become disconnected or when it changes link directions in response to an update message received from a neighboring node.

Figure 3.8 shows the pseudocode of the procedure executed when node v_k loses its last outgoing link toward destination v_j when PRI routing is in use. At the beginning of the procedure, it should check whether it requires to erasure the route or not using the procedure "checkErasureCondition()" that will be discussed in the next section. If it does not require to erase routes to the corresponding destination, it initiates link reversing procedure. If $\beta_{k,j} = 0$, then v_k does not have a neighbor such that $\lambda_{k,j} = \lambda_{l,j}$ for all $v_l \in N_k$. This case occurs when v_k does not have any auxiliary links. Therefore, v_k updates its pseudo-distance as $\lambda_{k,j} = min_{v_l \in N_k}(\lambda_{l,j}) + \delta$ in order to change some of its incoming links to primary outgoing links (lines 6–7 of Fig 3.8). On the contrary, if $\beta_{k,j} > 0$ and there exist a node $v_l \in N_k$ such that $\lambda_{k,j} < \lambda_{l,j}$, then v_k updates its pseudo-distance as $\lambda_{k,j} = \lfloor(\lambda_{k,j} + min_{v_l \in N_k}(\lambda_{l,j}))/2\rfloor$, where $\lambda_{k,j} < \lambda_{l,j}$, in order to change its auxiliary links to primary outgoing links (lines 8–9 of Fig. 3.8). Note that other incoming links remains unchanged in these cases in order to avoid route updates of other nodes. By this, PDR can reduce much of control messages. Finally, if $\beta_{k,j} > 0$ but there is no node that satisfies the condition $\lambda_{k,j} < \lambda_{l,j}$ for all $v_l \in N_k$, then v_k updates its pseudo-distance as $\lambda_{k,j} = min(\lambda_{l,j}) + \delta$ to change all of its incoming links to primary outgoing links (lines 10–11 of Fig. 3.8). After updating $H_{k,j}$ in either case, v_k broadcasts an UPD to its neighbors to notify them of the change in its pseudo-distance as lines 13–14 in Fig. 3.8.

Figure 3.9 shows the pseudocode of the procedure executed when node v_k loses its last outgoing link toward destination v_j when AUX routing is in use. The procedure is very similar to the procedure of PRI routing except it is initiated when a node loses its all of outgoing links including auxiliary outgoing links. At the beginning of the procedure, it should check whether it requires to erasure the route or not using the procedure "checkErasureCondition()" as in PRI routing.

```
1.  lostLastOutgoingLink(message p) {
2.      if(checkErasureCondition()) {
3.          erasureRoute();
4.          sendCLR(p);
5.          return;
6.      } else if(β_{k,j} = 0) {
7.          λ_{k,j} = min_{v_l ∈ N_k}(λ_{l,j}) + δ;
8.      } else if (there exist v_l such that λ_{k,j} < λ_{l,j} for all v_l ∈ N_k ){
9.          λ_{k,j} = ⌊(λ_{k,j} + min_{v_l ∈ N_k}(λ_{l,j}))/2⌋, where λ_{k,j} < λ_{l,j};
10.     } else {
11.         λ_{k,j} = min_{v_l ∈ N_k}(λ_{l,j}) + δ;
12.     }
13.     updateHeight();
14.     sendUPD(H_{k,j}, updseq_k++);
15.     return;
16. }
```

Figure 3.8: Pseudocode of procedure executed when last outgoing link is lost in PRI.

If it does not required to erase routes to the corresponding destination, it initiates link reversing procedure. If v_k does not have any auxiliary incoming link, then v_k updates its pseudo-distance as $\lambda_{k,j} = min_{v_l \in N_k} \lambda_{l,j} + \delta$ in order to convert its incoming links between v_k and the neighbors that have the minimum pseudo-distance to outgoing links (lines 6–7 of Fig. 3.9) as same as in PRI routing. On the contrary, if v_k has at least one auxiliary incoming link, and there exist a node $v_l \in N_k$ such that $\lambda_{k,j} < \lambda_{l,j}$, then v_k updates its pseudo-distance as $\lambda_{k,j} = \lfloor (\lambda_{k,j} + min_{v_l \in N_k}(\lambda_{l,j}))/2 \rfloor$, where $\lambda_{k,j} < \lambda_{l,j}$, in order to convert its auxiliary incoming links to outgoing links (lines 8–9 of Fig. 3.9). Note that other incoming links remains unchanged in order to avoid route updates of other nodes. Finally, if $\beta_{k,j} > 0$ but there is no node that satisfies the condition $\lambda_{k,j} < \lambda_{l,j}$ for all $v_l \in N_k$, then v_k updates its pseudo-distance as $\lambda_{k,j} = min(\lambda_{l,j}) + \delta$ to convert all of its incoming links to primary outgoing links (lines 10–11 of Fig. 3.9). After updating $H_{k,j}$ in either case, v_k broadcasts an UPD to its neighbors to notify them of the change in its pseudo-distance (lines 13–14 of Fig. 3.9).

The pseudocode of the procedure executed when v_k receives an UPD from its neighbor node, with destination v_j, is shown in Fig. 3.10. When v_k, which is not the destination, receives an UPD from its neighbor, it first updates its own routing table as line 2 of Fig. 3.10. Then it checks whether the time that the UPD was generated is newer than the last received UPD or not. If it has old τ value, then it updates its own $\tau_{k,j}$ and $ORI_{k,j}$ as the specified values in the received UPD message. It is mandatory to update $\tau_{k,j}$ and $ORI_{k,j}$ to detect network partitioning that will be discussed in the later of this section. If $\lambda_{k,j}$ was previously NULL and reflection flag of received UPD is false, v_l updates its $\lambda_{l,j}$ to $\lambda_{l,j} = \lambda_{m,j} + \delta$ and reset its reflection flag to false because v_k receives fresh UPD message that has routes to v_j (lines 9–13 of Fig. 3.10). Note that even $H_{k,j}$ is NULL, v_k does not update its pseudo-distance when it receives an UPD with reflection flag r = true because network partition is already detected and the

```
1.  lostLastOutgoingLink(message p) {
2.      if(checkErasureCondition()) {
3.          erasureRoute();
4.          sendCLR(p);
5.          return;
6.      } else if(number of auxiliary incoming link = 0) {
7.          λ_{k,j} = min_{v_l ∈ N_k}(λ_{l,j}) + δ;
8.      } else if(there exist v_l such that λ_{k,j} < λ_{l,j} for all v_l ∈ N_k ){
9.          λ_{k,j} = ⌊(λ_{k,j} + min_{v_l ∈ N_K}(λ_{l,j}))/2⌋ where λ_{k,j} < λ_{l,j};
10.     } else {
11.         λ_{k,j} = min_{v_l ∈ N_k}(λ_{l,j}) + δ;
12.     }
13.     updateHeight();
14.     sendUPD(H_{k,j}, updseq_k++);
15.     return;
16. }
```

Figure 3.9: Pseudocode of procedure executed when last outgoing link is lost in AUX.

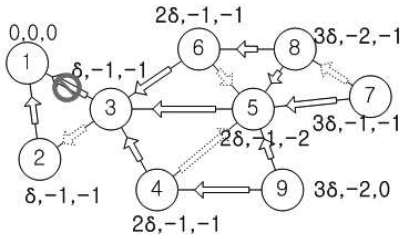
UPD message is a stale message. On the other hand, v_k may lose its outgoing link if $\lambda_{k,j} \leq$ p.$H.\lambda$ (line 14 of Fig. 3.10). In that case, v_k has to update its pseudo-distance by executing the procedure "lostLastOutgoingLink()" if it does not have any outgoing links as in lines 15-16 of Figure 3.10. Finally, if v_k receives an UPD message with $\lambda_{k,j}-$ p.$H.\lambda > \delta$, which indicates that a new shorter distance path to the destination has been found, v_k updates its pseudo-distance to $\lambda_{k,j}$ =p.$H.\lambda + \delta$ (lines 18–21 of Fig. 3.10).

Figure 3.11 shows a simple example of route maintenance that demonstrates the need for the use of δ in PDR. Vertices filled in with backslash lines represent nodes that are broadcasting an UPD message. Note that the example of Fig. 3.11 assumes that PRI routing is in use. AUX routing example will be discussed later in this section. In Fig. 3.11(a), node v_3 loses its last outgoing link toward destination node v_1 because link $e_{3,1}$ becomes disconnected. In Fig. 3.11(b), v_3 updates its pseudo-distance to $\lambda_{3,1} = \lfloor(\lambda_{3,1} + min_{n_i \in N_2}(\lambda_{i,1}))/2\rfloor$, where $\lambda_{i,1} > \lambda_{3,1}$ (line 9 of Fig. 3.8) because $\beta_{3,1} = 1$ and there are three neighbors that have greater pseudo-distance values than v_3 and meet the conditions of line 8 in Fig. 3.8. Note that all neighbors v_i except v_3 have $\lambda_{i,1} = 2\delta$. The node that satisfies $min_{n_i \in N_3}(\lambda_{i,1})$, where $\lambda_{i,1} > \lambda_{3,1}$, is v_5. Suppose that δ is 1. $\lambda_{3,1} = \lfloor(\lambda_{3,1} + \lambda_{5,1})/2\rfloor$ is 1, which is not acceptable in PDR because it can not reverse any links attached to v_3. Thus, PDR has to choose $\lambda_{3,1} = 2$, which results in link reversal topological changes for nodes v_4, v_5, v_6, v_7, v_8 and v_9. In order to deal with such cases, δ should be a sufficiently large integer value. Note that only a single step is sufficient for convergence in this example.

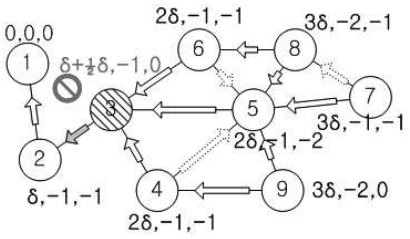
AUX routing can reduce the amount of control messages because it utilizes auxiliary outgoing links. Figure 3.12 shows the same example of Fig. 3.11 when AUX routing is in use. When the link $e_{3,1}$ is broken, no route update message is generated in AUX routing because v_3 still has one outgoing link even if there exists temporal inconsistency of height metric. The height metrics of v_3 that are

```
1.  recvUPD(message p) {
2.      updateNeighbor(p);
3.      if (p.DST ≠ v_k) {
4.          if(p.τ > τ_{k,j}) {
5.              τ_{k,j} = p.τ;
6.              ORI_{k,j} = p.ORI;
7.              r_{k,j} = p.r;
8.          }
9.          if (H_{k,j} = NULL and p.r is false) {
10.             λ_{k,j} = p.λ + δ;
11.             r_{k,j} = false;
12.             updateHeight();
13.             sendUPD(H_{k,j}, p.SEQ);
14.         } else if (λ_{k,j} ≤ p.H.λ) {
15.             if(isLostLastOutgoingLink()) {
16.                 lostLastOutgoingLink();
17.             }
18.         } else if (λ_{k,j}−p.H.λ > δ) {
19.             λ_{k,j} = p.λ + δ;
20.             updateHeight();
21.             sendUPD(H_{k,j}, p.SEQ);
22.         }
23.     }
24.     return;
25. }
```

Figure 3.10: Pseudocode for recvUPD.



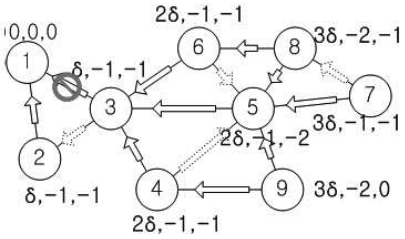
(a) A link $e_{3,1}$ is broken. Even v_3 still has an outgoing link but it needs to update its pseudo-distance because it does not have any PRI outgoing link.



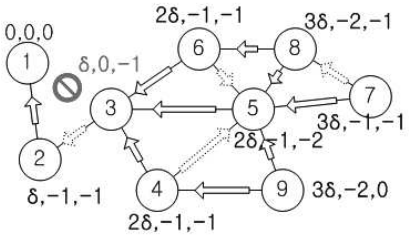
(b) v_3 updates its height as $H_{3,1} =< \delta + \frac{1}{2}\delta, -1, 0 >$ and generates an UPD message to convert its AUX outgoing link $e_{3,1}$ to PRI outgoing link.

Figure 3.11: An example of route maintenance of PRI routing.

stored in neighbor nodes of v_3 are still $< \delta, -1, -1 >$ but actual height metric of v_3 is $H_{3,1} =< \delta, 0, -1 >$. Note that PDR allows temporal inconsistency of α and β in order to reduce control messages as described above.



(a) A link $e_{3,1}$ is broken. However, v_3 still has an outgoing link $e_{3,2}$.



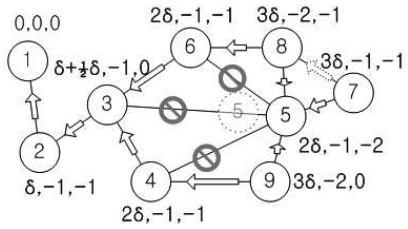
(b) v_2 updates its height as $H_{3,1} =< \delta, 0, -1 >$. However, no UPD message is generated in AUX routing.

Figure 3.12: An example of maintenance of AUX routing.

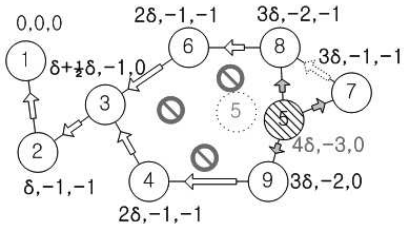
Figure 3.13 shows another example of route maintenance. In (a), v_5 moves toward the right direction. Therefore, links $e_{5,6}, e_{5,2}$ and $e_{5,4}$ become disconnected. For simplicity, suppose that all three links are broken at the same time. For both PRI and AUX routing, then v_5 updates its pseudo-distance to $\lambda_{5,1} =$

$min_{v_l \in N_5}(\lambda_{l,1}) + \delta = 4\delta$ (lines 6–7 in Fig. 3.8 or lines 6–7 in Fig. 3.9). Also, $\alpha_{5,1}$ becomes 3 since v_5 has three neighbors with smaller pseudo-distance and $\beta_{5,1}$ becomes zero since v_5 does not have any neighbors with the same pseudo-distance. The intermediate DAG is shown in Fig. 3.13(b). On receiving the UPD from v_5, v_7, v_8 and v_9 that are neighbors of v_5 update their neighbor's information as line 2 of Fig. 3.10. The result graph is shown in Fig. 3.13(c). Note that if AUX routing is in use, Fig. 3.13(c) is the final destination-oriented DAG. On the other hand, if PRI routing is in use, v_7 has to update its pseudo-distance in order to reverse some of its links because it loses its last primary outgoing link due to $\lambda_{5,1} = 4\delta$. Since v_7 loses its last outgoing link ($\lambda_{7,1} < \lambda_{5,1}$), it executes the "lostLastOutgoingLink()" procedure (lines 14–17 of Fig. 3.10). Then, v_7 updates its pseudo-distance to $\lambda_{7,1} = \lfloor(\lambda_{7,1} + \lambda_{5,1})/2\rfloor$ (lines 8–9 of Fig. 3.8) because v_7 has a neighbor v_5 that has $\lambda_{5,1} > \lambda_{7,1}$ and $\beta_{7,1}$ is 1. The final destination-oriented DAG of PRI routing is shown in Fig. 3.13(d).

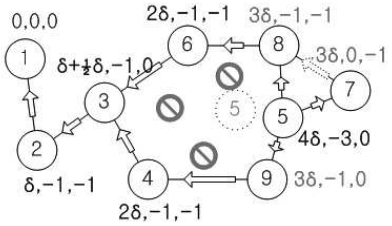
UPD messages are also generated when a new link is established between two nodes in order to notify their own height metrics to each other. Lets see an example of Figure 3.14 that a new node v_{10} newly joins the network. Then, a link $e_{9,10}$ is newly established as shown in Fig. 3.14(a). Suppose that v_9 detects link establishment earlier than v_{10}. Then v_9 generates an UPD message in order to notify its own height metric to its new neighbor node. On receiving this UPD message, v_{10} can set its height metric as shown in Fig. 3.14(b). On the other hand, when v_{10} detects the link establishment earlier than v_9, then it generates a QRY message to its corresponding neighbor to get valid routes. Then v_9 sends a REP message with its own height metric to v_{10}. On receiving a REP from v_9, v_{10} updates its own height metric as $H_{10,1} =< 4\delta, -1, 0 >$. Then, the new node v_{10} can have valid routes to destination.



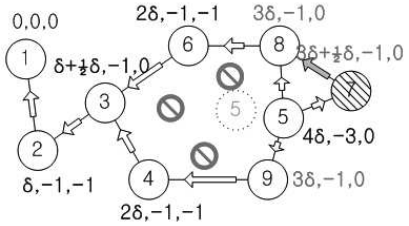
(a) Link $e_{5,2}, e_{5,4}$ and $e_{5,6}$ are broken due to mobility of v_5 concurrently.



(b) v_5 updates its height as $H_{5,1} =< 4\delta, -3, 0 >$ and generate an UPD message.

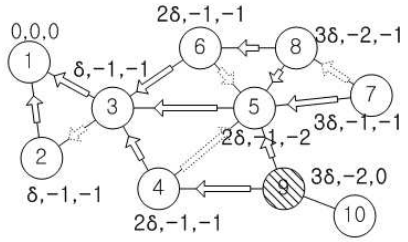


(c) Upon receiving the UPD from v_5, v_7 loses its last primary outgoing link but it still has an auxiliary outgoing link $e_{7,8}$. No more operations are required in AUX routing but not in PRI routing.

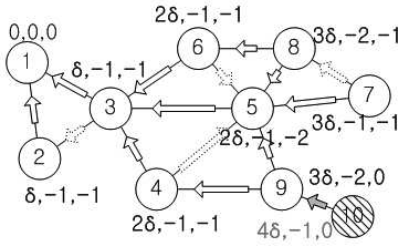


(d) In PRI routing, v_7 updates its height metric as $H_{7,1} =< 3\delta + \frac{1}{2}\delta, -1, 0 >$ and send an corresponding UPD message to convert its $e_{7,8}$ to primary outgoing link.

Figure 3.13: A second example of route maintenance.



(a) A new link $e_{10,9}$ is established. Therefore v_9 send an UPD message.



(b) v_{10} updates its height metric as $H_{10,1} =< 4\delta, -1, 0 >$.

Figure 3.14: An example of joining of a new node to the network.

3.2.4 Route Erasure Phase

TORA which is a successor of the partial reversal algorithm proposed a method of detecting network partitioning using *reference level* and *sub-level* concept. PDR can also detect network partition similar to TORA. In order to detect network partition, PDR uses three parameters that are a part of a UPD message as $< ORI, r, \tau >$ where ORI is the originator of the UPD message, r is a bit flag that represents reflection of the UPD message, and τ is the time that v_{ORI} detected the link broken event. When a node v_k detects loss of its last outgoing link(s), it generated an UPD with $< DST, H, SEQ, v_k$, false, Current Time>. Each intermediate node that receives the UPD message except destination updates its last receiving time and ID of the originator of the UPD message if the received UPD message is newer than previously received UPD messages as shown in lines 4–7 of Fig. 3.10. If a node loses its last outgoing link since it receives an UPD message, it executes "lostLastOutgoingLink()" procedure(lines 14–17 in Fig. 3.10. Figure 3.15 shows pseudocode of the "checkErasureCondition()" which is executed when an intermediate node v_k loses its last outgoing link due to an received UPD message from its neighbor node toward the destination node v_j. When v_k loses its last outgoing link by the received UPD message, then it executes "checkErasureCondition()" procedure. In the procedure of "checkErasureCondition()", v_k checks that it is the originator of the UPD message as line 2 of Fig. 3.15. If v_k is not the originator of the UPD message, then v_k checks whether there exists a neighbor node $v_l \in N_k$ that meets following conditions: (1) $ORI_{l,j} \neq \text{p}.ORI$, (2) $\tau_{l,j} \neq \text{p}.\tau$ (3) $r_{l,j} \neq false$ as line 7 of Fig. 3.15. If there is no such a neighbor, which represents that all of its neighbors are reversed their own links at least once due to the UPD message, v_k set its $r_{k,j}$ to true and return false. If there is such a neighbor, which represents some of its neighbors are not reversed their links by the UPD message, "checkErasureCondition()" procedure

returns false. Then v_k sends an UPD message as follows the "recvUPD()" procedure of Fig. 3.10 with its reflection value $r_{k,j}$. When an intermediate node v_l that received this reflected UPD message, it performs normal operations except it updates its own $r_{l,j}$ as true(lines 4–8 of Fig. 3.10). When v_m which is the originator of the UPD message loses its last outgoing links due to the received UPD message, it checks whether there exists a neighbor node $v_n \in N_m$ that meets following conditions: (1) $ORI_{n,j} \neq \text{p}.ORI$, (2) $\tau_{n,j} \neq \text{p}.\tau$, (3) $r_{n,j} \neq true$ as lines 2–5 in Fig. 3.15. If there exists such a neighbor, then "checkErasureCondition()" procedure returns false at line 11 of Fig. 3.15 that leads to normal link reversal procedure. However, if there is no such a neighbor, it returns true as line 4 of Fig. 3.15. Then v_m generates a CLR message as lines 3–5 in Fig. 3.8 or in Fig. 3.9 because all neighbors of v_m are reversed their own links by the UPD message that was reflected by v_l and generated by v_m.

```
1.  checkErasureCondition(message p) {
2.      if(p.ORI = v_k) {
3.          if(there is no v_l ∈ N_k s.t ORI_{l,j} ≠ p.ORI ∥ τ_{l,j} ≠ p.τ ∥ r_{l,j} ≠ true) {
4.              return true;
5.          }
6.      } else {
7.          if(there is no v_l ∈ N_k s.t ORI_{l,j} ≠ p.ORI ∥ τ_{l,j} ≠ p.τ ∥ r_{l,j} ≠ false) {
8.              r_{k,j} =true;
9.          }
10.      }
11.     return false;
12. }
```

Figure 3.15: Pseudocode for checkErasureCondition procedure.

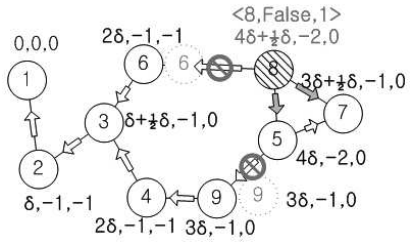
Figure 3.16 shows the pseudocode executed when a node v_k receives a CLR message from its neighbor toward destination v_j. If v_k is the destination node, then it generates a fresh UPD message with r = false. Otherwise, it checks whether $H_{k,j}$ is NULL or not(line 4 of Fig. 3.16). If $H_{k,j}$ not NULL, then it checks its $r_{k,j}$. If $r_{k,j}$ is true which represents that it already tried to find routes to destination using link reversal procedure already, it erases its own $H_{k,j}$, then send the CLR message to its neighbors(lines 4–9 in Fig. 3.16. Otherwise, no operation is required because it still has valid routes to the destination.

```
1.  recvCLR(message p) {
2.      if(p.DST = v_k) {
3.          sendUPD(H_{k,k}, updseq_k++);
4.      } else if(H_{k,j} ≠ NULL) {
5.          if(r_{k,j}) {
6.              eraseHeight();
7.              sendCLR(p);
8.          }
9.      }
10.     return;
11. }
```

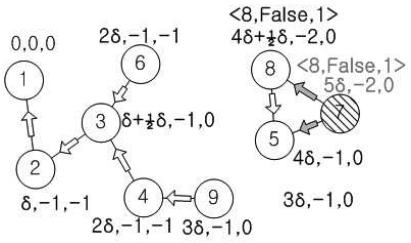
Figure 3.16: Pseudocode for recvCLR procedure.

Figure 3.17 shows an example of network partition and corresponding route erasure operations. Initially, as in Fig. 3.17(a), link $e_{8,6}$ and $e_{5,9}$ were broken at time 1 simultaneously. Because v_5 still has an outgoing link $e_{5,8}$, v_5 does not require to reverse its links. However, v_8 should reverse its incoming links because it does not have any outgoing links. Therefore v_8 updates its pseudo-distance as $\lambda_{8,1} = 4\delta + \frac{1}{2}\delta$ and corresponding α and β as $\alpha = 2$ and $\beta = 0$. Then v_8

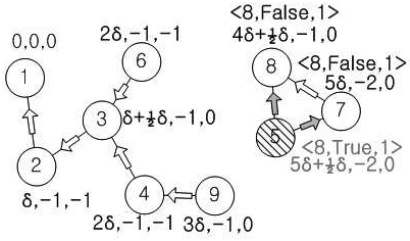
generates an UPD message as $< v_1, H_{8,1}, updseq_{8,1}, v_8, false, 1 >$. On receiving the UPD message that v_8 generated, v_7 lose its last outgoing link. Therefore it updates its pseudo-distance as $\lambda_{7,1} = 5\delta$ and corresponding α and β as in (b) of Fig. 3.17 since it does not have any auxiliary links. Note that v_7 updates its $< ORI, r, \tau >$ to $< v_8, false, 1 >$ as the UPD message specified because there is a neighbor v_5 that meets the conditions of line 7 of Fig. 3.15. Then v_7 broadcasts an UPD message as $< v_1, H_{7,1}, updseq_{8,1}, v_8, false, 1 >$ as in Fig. 3.17(b). By then, v_5 loses its last outgoing link toward destination v_1 due to the UPD message broadcast by v_7. Therefore v_5 updates its pseudo-distance as $\lambda_{5,1} = 5\delta + \frac{1}{2}\delta$ and corresponding α and β. Note that because there is no neighbor node $v_l \in N_5$ that meets conditions of line 7 of Fig. 3.15, it sets its reflection flag $r_{5,1}$ to true. Then, v_5 broadcasts an UPD message as $< v_1, H_{5,1}, updseq_{8,1}, v_8, true, 1 >$ as in Fig. 3.17(c). In Fig. 3.17(d), v_8 loses its last outgoing link again, but it has a neighbor node v_7 that has false reflection flag, it simply updates its height metric and broadcasts an UPD message. Note that v_7 receives the UPD message from its neighbor node v_5 where reflection flag r is true, but it does not update its height metric because still has an outgoing link to destination v_1. On receiving the UPD message broadcast by v_8, v_7 updates its height metric since it loses its last outgoing links, and broadcasts an UPD message with reflection flag $r =$ true as in Fig. 3.17(e), because there is no neighbor that meets the conditions of line 7 of Fig. 3.15. In Fig. 3.17(f), v_5 updates its height metric due to lost of last outgoing link and broadcasts an UPD message with $r =$ true. v_8 which is the originator of the UPD message receives an UPD message in Fig. 3.17(g). Because there is no neighbor nodes that meets the conditions of line 3 of Fig. 3.15, it generates a CLR message as $< v_1, v_8, seqclr_{8,1} >$ after erasing its height metric. Finally, v_5 and v_7 also erase their routes because they are not the destination and their reflection flags are true as shown in Fig. 3.16.



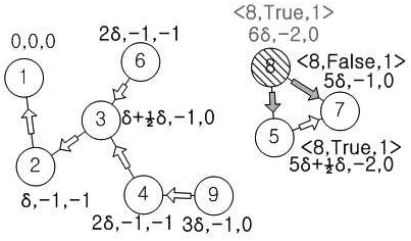
(a) Due to loss of links $e_{5,9}$ and $e_{8,6}$, v_8 generates an UPD while v_5 does not.



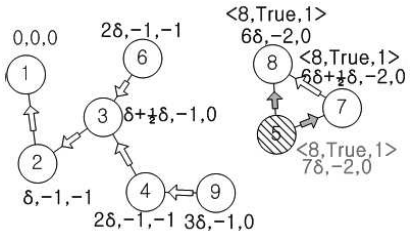
(b) v_7 reverses its links with $< ORI = v_8, r =$false, $\tau = 1 >$.



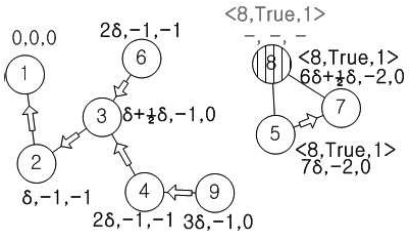
(c) v_5 reverses its links with r=true since all nodes $v_k \in N_5$ have the same $< ORI, r, \tau >$.



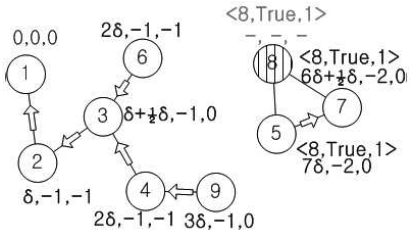
(d) v_8 reverses its links with r=true because $r_{7,1}$ is false.



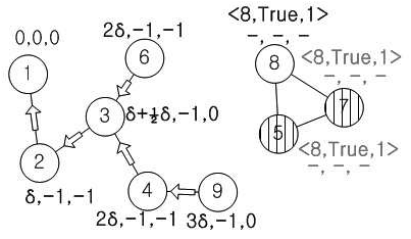
(e) v_7 reverses its links with r=true.



(f) v_5 reverses its links again.



(g) v_8 generates a CLR since $\forall v_k \in N_8, ORI_{k,1} = ORI_{8,1}, r_{k,1} =$true, $\tau_{k,1} = \tau_{8,1}$.



(h) v_5 and v_7 also erase their routes and propagate CLRs.

Figure 3.17: An example of route erasure when network partition occurs.

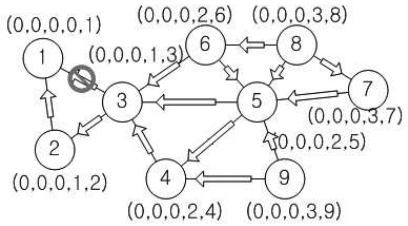
3.2.5 Route Loop due to Temporal Inconsistency of AUX Routing

In order to reduce the amount of control messages, PDR allows temporal inconsistency of α and β. PDR generates UPD messages only when pseudo-distance λ of local node is changed i.e. no UPD message is generated when only α or β are changed. Therefore in some cases, route loop may be occurred due to temporal inconsistency of α and β when AUX routing is in use. In order to detect route loop caused by temporal inconsistency of height metric, AUX routing checks previous hop of the received data packet. If the previous hop of the received data packet is local node, then it should generate an UPD message to notify its current height metric to its neighbor nodes. Suppose that a node v_k receives a data packet from v_l where previous hop of the packet is v_k. Then, v_k updates its pseudo-distance $\lambda_{k,j} = \lambda_{k,j} + \frac{1}{2}\delta$ and broadcasts an UPD message and enqueue the data packet into its own local queue. Upon receiving the UPD message that v_k broadcast, its neighbors also update their own pseudo-distance and corresponding height metric and broadcasts UPD messages. Therefore, route loop can be resolved. When a node receives an UPD or REP message from its neighbor, then it forwards stored data packets in its own local queue.

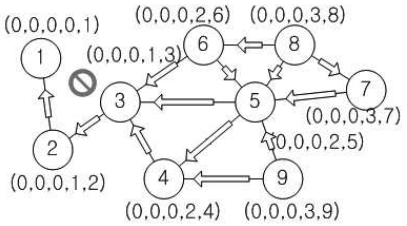
3.2.6 Performance Comparison of PDR to TORA by Examples

As described in earlier sections, control messages are generated when destination oriented DAG is broken in both PDR and TORA. Suppose that TORA is in use to the same example of Fig. 3.11. PRI routing generates one UPD message and AUX routing generates no UPD message as discussed in Section 3.2.3 while TORA generates no UPD message since v_3 still has one outgoing link $e_{3,2}$ as shown in Fig. 3.18.

In order to highlight differences of the number of control messages between



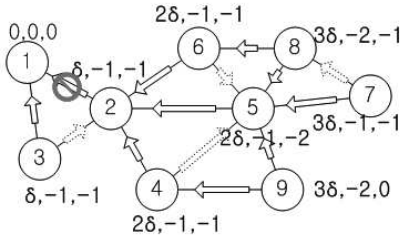
(a) Link $e_{1,2}$ is broken.



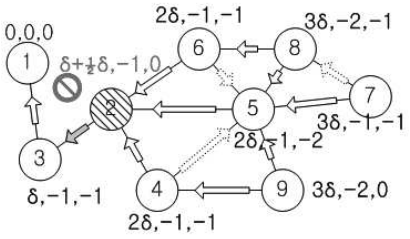
(b) No operation is required since v_3 still has one outgoing link.

Figure 3.18: A TORA example with the MANET of Fig.3.11.

PDR and TORA, lets see a slightly modified example that IDs of nodes v_2 and v_3 are exchanged from the above maintenance example. Figure 3.19 shows a maintenance operations of PDR. Note that IDs of v_2 and v_3 are exchanged from the above example. Suppose that v_2 loses its last outgoing link $e_{2,1}$ as shown in Fig. 3.19(a). Then both PRI and AUX routing generate only one UPD packet to convert its auxiliary incoming link $e_{2,3}$ to outgoing link as shown in Fig. 3.19(b). However, TORA requires 7 steps for convergence as shown in Figure 3.20. The details of each step of TORA maintenance are described in Fig. 3.20.



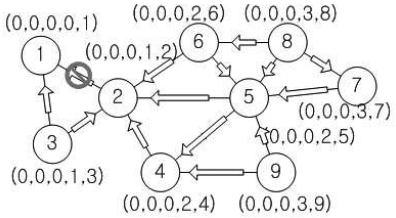
(a) Link $e_{1,2}$ is broken.



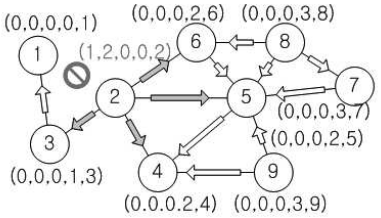
(b) v_2 reverses its links and send an UPD.

Figure 3.19: An additional example of route maintenance of PDR.

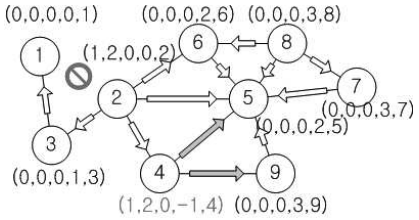
Figure 3.21 compares PDR with TORA in path length. Fig.3.21(a) shows the initial routes of PDR when $\delta = 4$ and Fig. 3.21(b) shows the initial routes of



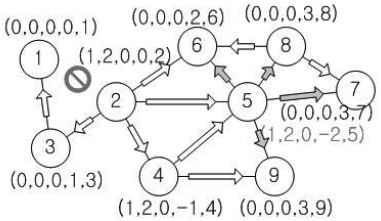
(a) Link $e_{1,2}$ is broken.



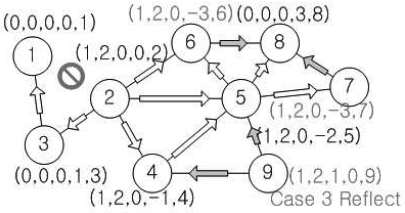
(b) v_2 reverses its links and send an UPD (generate).



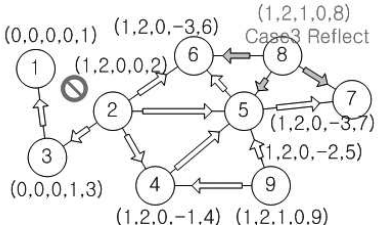
(c) v_4 propagates the UPD because it loses all downstream links (propagate).



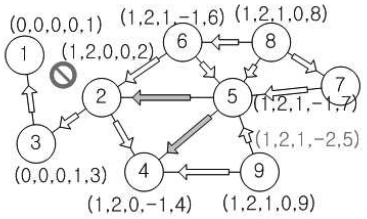
(d) v_5 propagates the UPD because it also loses all downstream links (propagate).



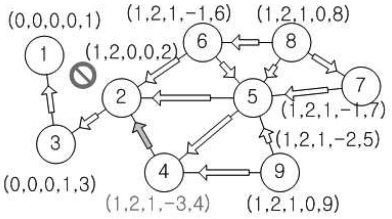
(e) v_6 and v_7 propagate the UPDs. v_9 loses all downstream links and all of its neighbors have reference level (reflect).



(f) v_8 also reflects the UPD.



(g) v_5 propagates the UPD message.



(h) Finally, v_4 propagates the UPD.

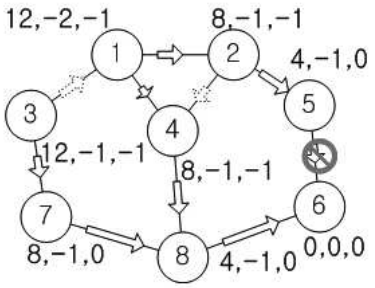
Figure 3.20: A TORA example with the MANET of Fig.3.11.

TORA for the same MANET. Suppose that the link $e_{5,6}$ becomes temporarily disconnected. Then v_5 updates its height in both algorithms; $H_{5,6} =< 12, -1, 0 >$ in PDR as shown in Fig. 3.21(c) and $H_{5,6} =< 1, 5, 0, 0, 5 >$ in TORA as shown in Fig. 3.21(d). If the link $e_{5,6}$ is re-established later, then v_5 updates its pseudo-distance to find shorter paths in PDR as shown in Fig. 3.21(e). However, TORA does not update any height values because both v_5 and v_6 have their own height values as shown in Fig. 3.21(f). The destination-oriented DAG produced with TORA is shown in Fig. 3.21(f). The (2,6)-path selected by PDR is $P_{2,6}^{PDR} = (2, 5, 6)$ and the length of this path is $|P_{2,6}^{PDR}| = 2$. However, the (2,6)-path selected by TORA is $P_{2,6}^{TORA} = (2, 4, 8, 6)$ with length $|P_{2,6}^{TORA}| = 3$. Note that the distance of a (2,6)-path is $D_{2,6} = 2$ since that is the length of the shortest path from node 2 to node 6. As stated in Sect.2.4, TORA may produce paths with increased path lengths after a few topological changes.

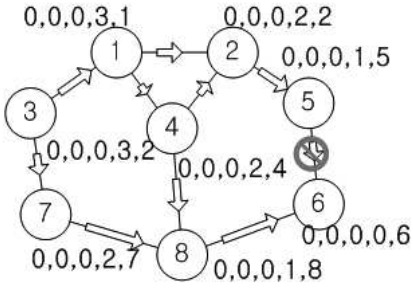
3.3 Evaluation

Simulations were conducted to evaluate the benefits and costs of the PDR routing algorithm using ns-2[53], which is a discrete event simulator tool commonly used in networking research. We compared the performance of PDR with TORA and AODV because TORA is a previously proposed link reversal algorithm and AODV is most widely accepted routing protocol for MANETs. The other algorithms discussed in Chapter 2 are not compared because they all have major drawbacks, such as high overhead and requirement of external location services, and thus are not directly comparable to our approach.

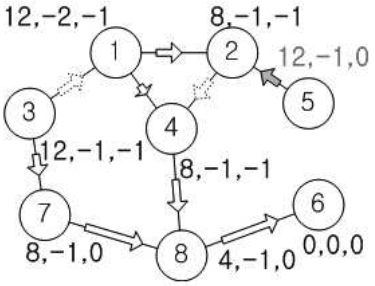
There are five major parameters that affect the performance of PDR routing: (1) node density, (2) mobility of nodes, (3) number of source nodes per a destination, (4) beacon period of IMEP that affect detection time of topological changes in PDR and TORA, (5) δ as a default difference in pseudo-distance between



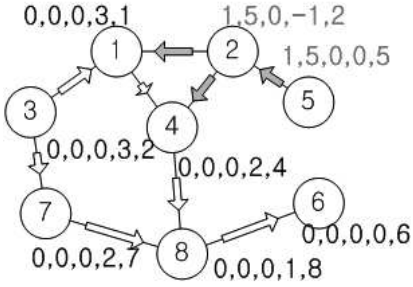
(a) Link $e_{5,6}$ is temporally broken in PDR.



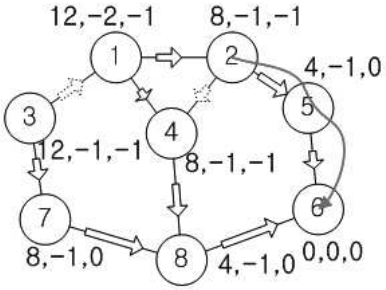
(b) Link $e_{5,6}$ is temporally broken in TORA.



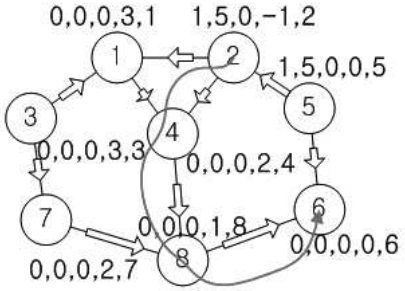
(c) Reconstructed destination-oriented DAG in PDR.



(d) Reconstructed destination-oriented DAG in TORA.



(e) Reconstructed destination-oriented DAG after reestablishment of the link $e_{5,6}$ in PDR.



(f) Reconstructed destination-oriented DAG after reestablishment of the link $e_{5,6}$ in TORA.

Figure 3.21: Comparison of PDR with TORA.

adjacent nodes in PDR. Performance of the most of routing protocols are dependent on density of nodes. Performance of AODV, TORA and PDR versus node density are compared in Section 3.3.2. Performance of routing protocols also depends on mobility of nodes. Therefore, performances of PDR is versus mobility of nodes are compared to other routing protocols in Section 3.3.3. Communication patterns also affect the peroformance of routing protocols.Therefore PDR, TORA and AODV performances versus number of source nodes per a destination are compared in Section 3.3.4. Detection of topological changes can also affect performances of PDR and TORA. Therefore, in Section 3.3.5 performance of PDR and TORA versus beacon period of IMEP is compared since IMEP takes reponsibility of detection of topological changes. Finally, δ as a default difference in pseudo-distance between adjacent nodes in PDR also can affect the performance of PDR since use of δ can reduce the number of routing messages as shown in Section 3.2.3. Therefore performance of PDR versus δ are presented in Section 3.3.6.

3.3.1 Simulation Environment

The distributed coordination function(DCF) of IEEE 802.11[3] for wireless LANs is used for the MAC and PHY layers. The data rate is set to 11Mbps that is a rate widely supported by 802.11b devices. The simulation space is a 1500m x 500m area, and the communication range of each node is set to 250m. As stated in Section 3.1.3, PDR is implemented on top of IMEP. Table 3.1 shows detailed parameters of IMEP. The mobility of the nodes are controlled by a mobility generator function in ns-2 that uses a random destination model[54]. Each node starts its journey from a random location to a random destination with a randomly chosen speed that is uniformly distributed between 0 and a maximum speed. When a node arrives at its destination, it stays there for specified time (pause time) and then restarts its journey to another random destination with

Table 3.1: IMEP parameters

Beacon Period	1.0 s
Max beacon waiting time	3 times of Beacon Periods
Beacon jitter	0.01 s

a randomly chosen speed. Finally, the simulation time is set to 130 seconds. A source sends 256 bytes of UDP packet to its destination every 1 second from 10 seconds after the simulation starts to 125 seconds. The maximum speed of each node and the number of nodes are varied for each simulation scenario. Data are collected for ten different simulation scenarios. Other simulation parameters are shown in the table 3.2. Constants of AODV are shown in Table 3.3. Note that there is no modification of AODV constants.

3.3.2 Simulation Results Versus Number of Nodes

As the number of nodes increased, more control messages are required in PDR and TORA because they have to exchanges route information with more nodes. Note that AODV may require less control messages because it only maintains single path to destination and it does not require to detect network partition. PDR exchanges height metrics between nodes when a new link is established in order to find shorter path than they have. On the other hand, TORA does not exchange height metrics between nodes when a new link is established in order to reduce control messages. Therefore, it is easy to expect that PDR may require more control messages than TORA. However, PDR can reduces control messages using δ as described in the example of Fig. 3.11 that results the number of control messages of PDR can be comparable to that of TORA. Note that control messages

Table 3.2: Constants used in simulation.

Radio model	Two-ray ground
RTS/CTS	Enabled
Preamble length of IEEE 802.11	Short preamble (72 bits)
Carrier Sensing Threshold	1.559e-11
Receiving Threshold	3.652e-10
Carrier Frequency	914e+6
Transmitted Signal Power	0.28183815
System loss	1.0
Antenna gain at transmitter	1.0
Antenna gain at receiver	1.0
Antenna height	1.5

Table 3.3: Constants of AODV used in simulation.

Number of times a Route Request is retried	3
Time before a Route Request is retried	6 s
Time before broken link is deleted from routing table	3 s
Time for keeping Route Request node	3 s
Time for keepting reverse route information for Route Reply	3 s

of both PDR and TORA are increased linearly as the number of nodes increased.

AODV does not update its path until the previously discovered path is broken even if shorter path is established.Therefore AODV suffers from detour since it can not find shorter path even if new shorter paths are established. TORA results detour because it does not take route optimality into account while PDR does. Therefore, PDR may shows best performance in path length. In PDR, PRI routing may provide shorter path than AUX routing because auxiliary outgoing links have higher probability to be a detour than primary outgoing links.

Packet delivery ratio is increased as the density of nodes is increased because there is more route redundancies. However, packet delivery ratio of AODV would be worse than others since it does not able to reroute the data packets that are already transmitted at the source node if path is broken at intermediate nodes along the path. On the other hand, packet delivery ratios of PDR and TORA are expected to better than AODV since they can reroute the data packets that are already transmitted at the source node. In addition, PDR may have higher packet delivery ratio than TORA because PDR attempts to choose a paths with more alternative paths to the destination.

Characteristics of Simulation Scenarios

Table 3.4 shows the average number of link connnectivity changes, route changes, and the number of destination unreachable of simulation scenarios that are used in simulation versus number of nodes. Link changes represents how many links are connected and disconnected during simulation time i.e. it happens whenver a node get into or out of directed communication rnage to other node. Route changes represents the total number of route changes between any two mobile nodes during simulation. The number of destination unreachable represents the total number of cases where two mobile nodes are not reachable with respect to each other during simulation time. It is easy to expect that the num-

Table 3.4: Average number of link connectivity changes, route changes and destination unreachables of scenarios that are used in simulation versus number of nodes.

# Nodes	Link Changes	Route Changes	Unreachable Dest.
20 nodes	272.6	2530.2	446.8
30 nodes	593.2	6215.3	456.5
40 nodes	1134	9354.3	222.8
50 nodes	1729.4	13828.5	132.4

ber of link changes and the number of route changes are increased as the number of nodes are increased. On the contrary, the number of unreachable destinations is decreased as the number of nodes is increased since density of nodes are increased. Therefore, we can expect that as the number of nodes are increased, the number of control messages and packet delivery ratio are increased but path lengths are decreased as discussed above section.

Path Length Versus Number of Nodes

Figure 3.22 shows the average path length of PRI, AUX, TORA and AODV versus the number of nodes when nodes are moving up to 10m/s. It should be note that the average path lengths considers only successfully received packets. As expected, for all routing protocols, path lengths tend to be decreased as the number of nodes is increased. PRI shows the best performance in terms of path length and AUX shows relatively similar performance to PRI while TORA and AODV shows worse performance. This is because PDR tries to keep pseudo-distances as small as possible even if it introduces more control messages. In the average for all cases, path length of PRI is 2.88 hops, path length of AUX is

3.04 hops, path length of TORA is 3.23 hops and path length of AODV is 3.25 hops. It should be noted that performance of path length of routing protocols are different from expectation when the number of nodes is 20 because packet delivery ratios of TORA and AODV are very low as shown next. Packets are easily dropped if the path length gets long since the path can be broken more easily thatn shorter paths. Therefore, path lengths tend to decrease if packet delivery ratio is decreased since dropped packets are not counted in path length.

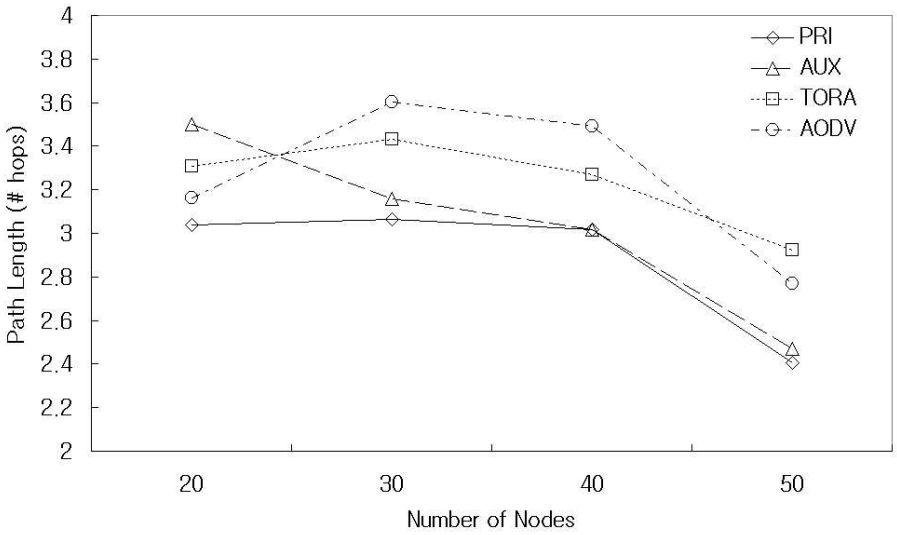


Figure 3.22: Path length vs. number of nodes where v_{max} = 10m/s, beacon period = 1s and $\delta = 4096$.

Packet Delivery Ratio Versus Number of Nodes

Figure 3.23 shows average packet delivery ratios versus the number of nodes. The packet delivery ratios are calculated as the number of successfully received packets at the destination divided by the total number of messages sent by the

sender. Packet delivery ration tends to increased as the number of nodes is increased as expected. PRI and AUX routing show better packet delivery ratio than others as expected since PDR attempts to choose a paths with more alternative paths using α and β in height metric. Note that, unlike TORA, PDR always attempts to choose paths that are as short as possible that are more unstable links as described in the next chapter. Packet delivery ratio of AODV is worse than others since it does not maintain multiple redundant paths. Once a data packet is transmitted at the source node, then it follows the previously discovered path.A path can be broken while the data packet passes through the path. In such case, intermediate nodes that detects link breakage drops the data packets since it is not able to find a new path. However, source nodes keep transmitting data packets through the path until they are notified the link breakage events by the intermediate node. In the average for all cases, PRI delivers 97.09% of its packets, AUX delivers 97.15%, TORA delivers 94.09% but AODV delivers only 87.41% of its packets. Therefore packet delivery ratio of AODV is worse than others.

Number of Control Messages Versus the Number of Nodes

Figure 3.24 shows the average number of control messages versus the number of nodes. As expected, the control messages of PDR and TORA are increased almost linearly as the number of nodes is increased. It should be noted that the control messages of AODV is much less than others since it does not maintain multiple paths and it does not require to detect network partition as stated above. There is a trade-off relation between the number of control messages generated and packet delivery ratio as discussed earlier. PDR requires more control messages than TORA in cases of the number of nodes are less than 30 but PDR requires less control messages if the number of nodes are greater than 40 since the effect of δ that reduce amount of control messages are highlighted as the number

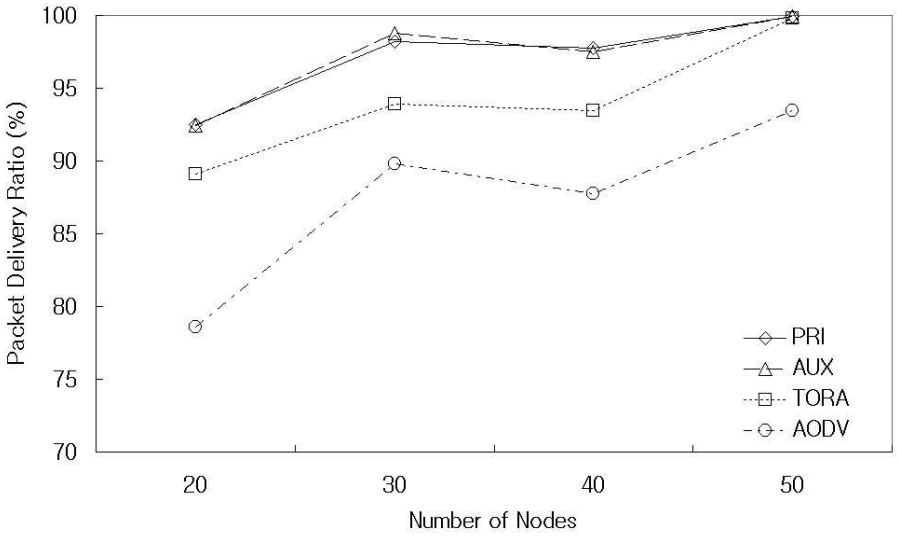


Figure 3.23: Packet delivery ratio vs. number of nodes where v_{max} = 10m/s, beacon period=1s and $\delta = 4096$.

of nodes are increased. PRI and AUX show comparable performance in control messages even if each node exchanges its height information with its neighbor node whenever a new link is established. It should be also noted that there is a trade-off relation between the number of control messages generated and the detour ratio.

3.3.3 Simulation Results Versus Mobility of Nodes

Mobility of nodes can be modeled as maximum speed of nodes. As the number of topological changes increased, more control messages are required for all routing protocols. The number of topological changes are mainly dependent on the mobility of nodes. If nodes move fast, frequent topological changes occur, and vice versa. As stated in Section 3.3.2, AODV generates less control messages than others while TORA and PDR competes.

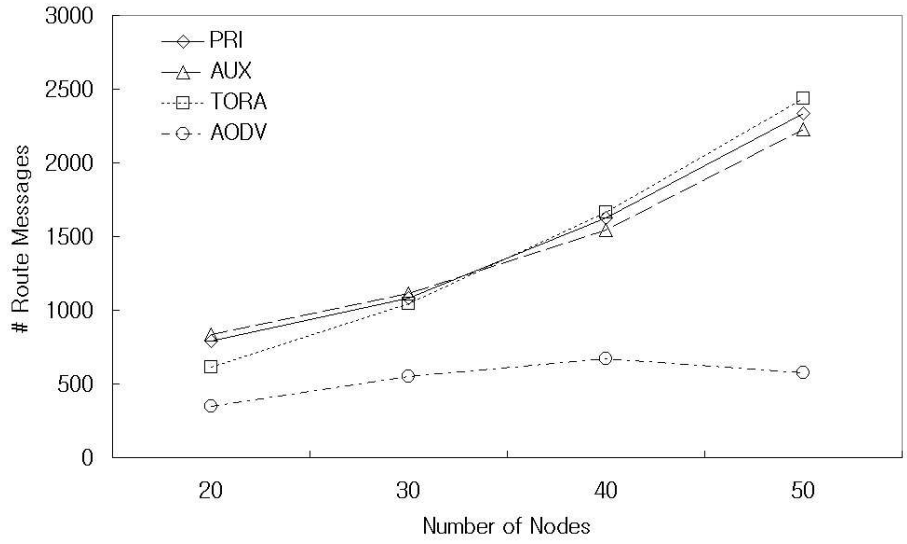


Figure 3.24: Number of control messages vs. number of nodes where v_{max} = 10m/s, beacon period=1s and $\delta = 4096$.

Path length may not rely on the speed of nodes since path length should be almost same if the nodes are distributed uniformly even if nodes move fast. However, PDR shows less path length than others due to the same reasons discussed in Section 3.3.2.

Packet delivery ratios should be decreased as the node mobility increased due to frequent topological changes increase chances to drop data packets. PDR should show better packet delivery ratio and AODV performs worse than others as same as versus number of nodes cases.

Characteristics of Simulation Scenarios

The average number of link connectivity changes, route changes, and the number of destination unreachables of simulation scenarios that are used in simulation versus maximum speed of nodes are shown in Table 3.5 As expected, as

Table 3.5: Average number of link connectivity changes, route changes and destination unreachables of scenarios that are used in simulation versus maximum speed of nodes.

Max. Speed	Link Changes	Route Changes	Unreachable Dest.
5 m/s	531.8	4,296.3	65.8
10 m/s	994.5	7,950.5	162
15 m/s	1,411.6	11,597.5	121.2
20 m/s	1,732.3	14,800.4	19.4

the maximum speed of nodes increased, dynamicity of network also increased. Therefore, the number of link changes and the number of route changes are increased as the maximum speed of nodes is incresed. However, the number of unreachable destination is not affectedby the maximum speed of nodes.

Path Length Versus Maximum Speed of Nodes

Figure 3.25 shows the effect of mobility on path lengths when the number of nodes is fixed at 50. As expected, there is almost no relation between node mobility and path length in Fig. 3.25. Both PRI and AUX show better performance than TORA and AODV in terms of path lengths as in Fig.3.22, Note that performance of AODV in path length is worse than any other routing protocols. In the average for all cases, path length of PRI is 2.79 hops, path length of AUX is 2.65 hops, path length of TORA is 3.11 hops and path length of AODV is 3.42 hops. Note that PDR is a kind of greedy algorithm. Therefore, it is not guaranteed that selection of path at the intermediate node is always optimum. Consequently, the performance of PRI in terms of path length can be worse than AUX in some simulation scenarios.

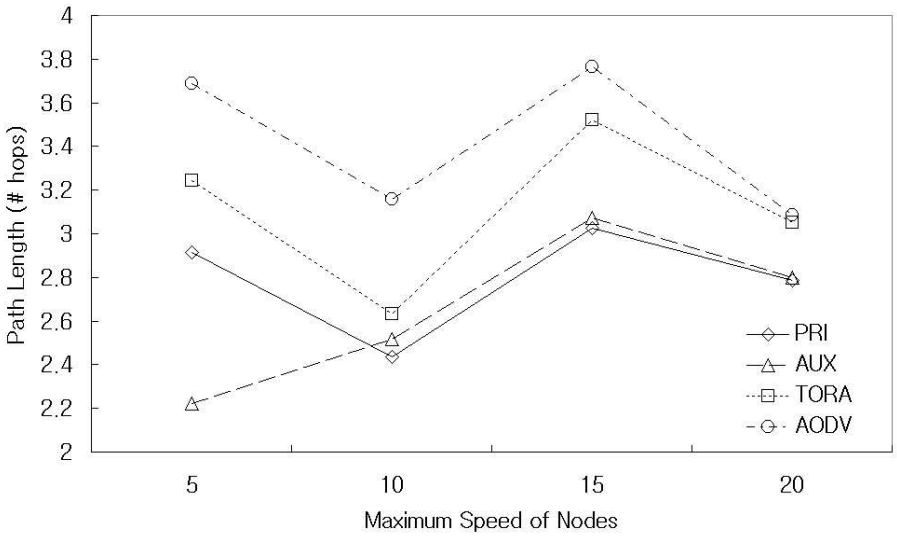


Figure 3.25: Path length vs. node mobility where # of nodes is 50, beacon period=1s and $\delta = 4096$.

Number of Control messages Versus Maximum Speed of Nodes

Figure 3.26 shows average packet delivery ratios versus node mobility. Packet delivery ratios are decreased as mobility of node is increased as expected above. Note that PRI and AUX deliver almost all packets(99.91% and 99.87%) while TORA delivers 98.28% and AODV delivers only 94.09% of packets in the average for all cases.

Number of Control Messages Versus Maximum Speed of Nodes

Figure 3.27 shows the average number of control messages generated versus node mobility. As the maximum speed of nodes is increased, the number of topological changes also increases as shown in Table 3.5. Therefore, the number of control messages for all routing protocols also tend to increase. As node mobility

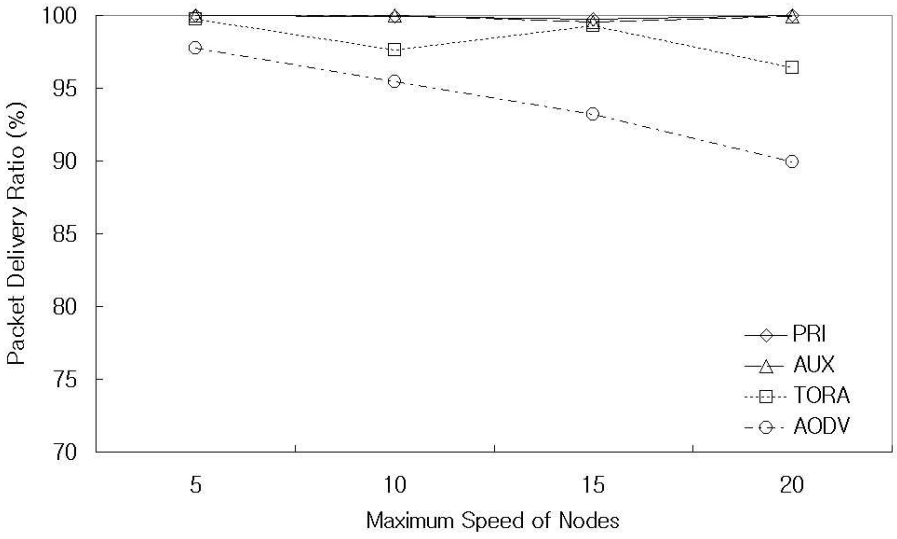


Figure 3.26: Packet delivery ratio vs. node mobility where # nodes is 50, beacon period=1s and $\delta = 4096$.

is increased, number of link breakage and newly establishment are also increase that results more control messages in PDR since PDR exchanges height metrics when a new link is established. Even the mobility of nodes is increased, the actual number of control messages required for PDR are comparable to TORA while AODV generates much less control messages than others as discussed above.

3.3.4 Simulation Results Versus Number of Source Nodes per a Destination

As long as nodes are distributed uniformly and source/destination pairs are randomly selected, it is expected that there is no direct relation between path length and the number of source nodes. Packet delivery ratio can be affected by the number of sources due to contention of the path. In addition to contention, since AODV drops all packets that follow the path which includes the broken

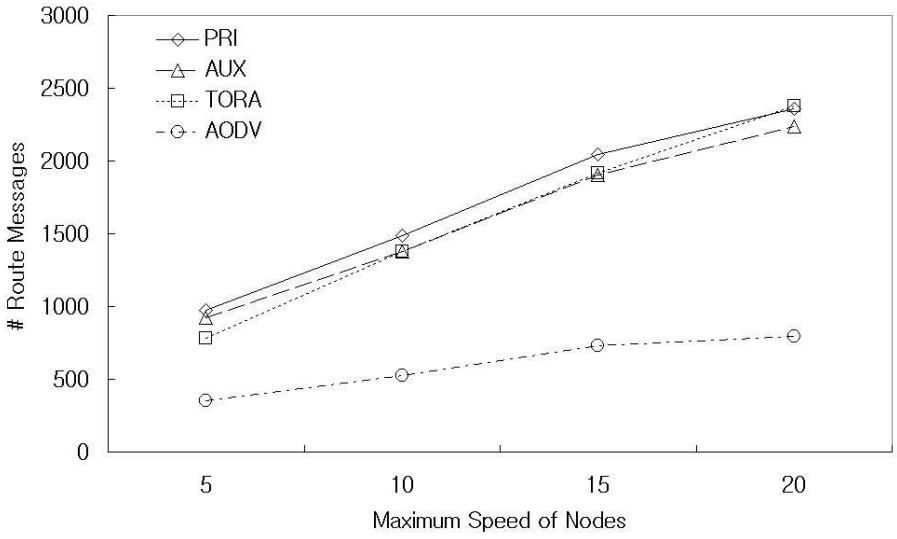


Figure 3.27: Number of control messages vs. node mobility where # nodes is 50, beacon period=1s and $\delta = 4096$.

link to destination, packet delivery ratio of AODV becomes worse as the number of source nodes per a destination is increased. Suppose that a direct link to destination which is shared among several paths is broken. Until new paths are found from all source nodes, packets are kept dropping while PDR and TORA are not affected since they provide multiple redundant paths. Note that in AODV, link breakage events should be notified to the source node in order to discover new path to the destination, which takes more time as number of sources are increased. Therefore, packet delivery ratio of AODV becomes worse as the number of source nodes per a destination is increased. Furthermore, AODV requires more control messages as the number of source nodes per a destination increased because AODV only support pure peer-to-peer communication patterns. While PDR and TORA provide multiple paths from multiple nodes in the network by incorporating destination-oriented DAG, AODV requires to maintain routes for

each source to the destination pair. Therefore, control messages of AODV should be rapidly increased as the number of source nodes per a destination is increased while PDR and TORA are not.

Path Length Versus Number of Source Nodes per a Destination

Figure 3.28 shows path lengths of routing protocols versus number of nodes per a destination. As expected, the number of source nodes does not affect path length as expected. Path lengths are slightly greater than others when the number of source nodes are 1 since distances between randomly selected sources to a randomly selected destination are very long as 6.33 hops in PRI, 6.46 hops in AUX, 7.41 hops in TORA and 6.55 hops in AODV in one of the simulation scenario out of ten. As other simulation cases, PRI shows the best performance in path length, and AUX follows similarly while TORA and AODV suffer long detour of routes.

Packet Delivery Ratio Versus Number of Source Nodes per a Destination

Figure 3.29 shows packet delivery ratio of routing protocols versus number of nodes per a destination. Packet delivery ratio of PDR and TORA are slightly decreased as the number of source nodes per a destination is increased. On the other hand, packet delivery ratio is rapidly decreased as the number of source nodes per a destination is increased as expected. In average of all cases, PRI delivers 99.37% of packets, AUX delivers 99.15% of packets, TORA delivers 98.85% of packets while AODV delivers only 83.38% of packets.

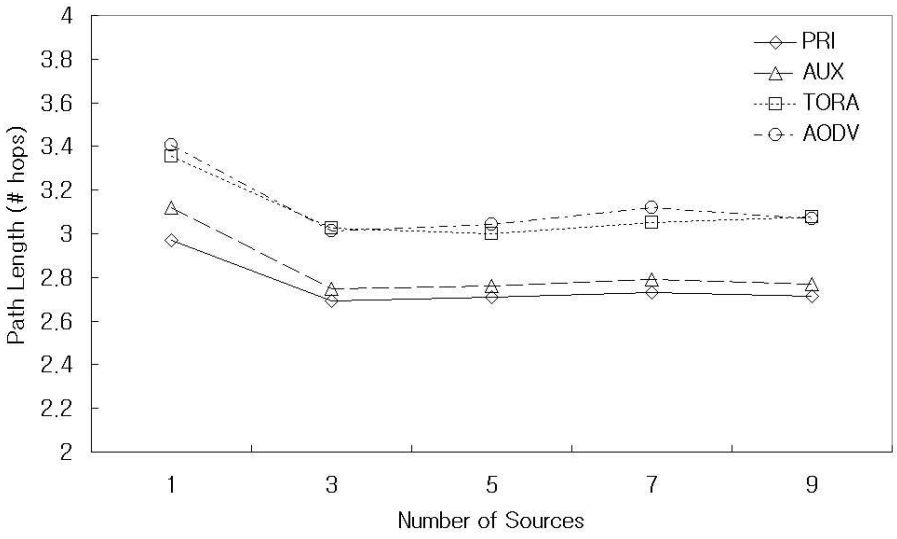


Figure 3.28: Path length vs. number of source nodes per a destination where # nodes is 50 nodes, $v_{max} = 10$m/s, beacon period=1s and $\delta = 4096$.

Number of Control Messages Versus Number of Source Nodes per a Destination

Figure 3.30 shows the number of control messages generated versus number of source nodes per a destination. Note that the scale of Y axis of Fig. 3.30 is different to others. In AODV, the amount of control messages are rapidly increased as the number of source nodes per a destination as expected. However, the amount of control messages are increased very slowly in PDR and TORA since PDR and TORA provide multiple redundant paths for multiple nodes. In detail, when the number of source node per a destination is one which is same as previous simulation setups, PRI generates 2311.1 control messages, AUX generates 2171 control messages, TORA generates 2408.9 control messages, and AODV generates 913.7 control messages. On the other hand, PRI generates 2532.3 control messages,

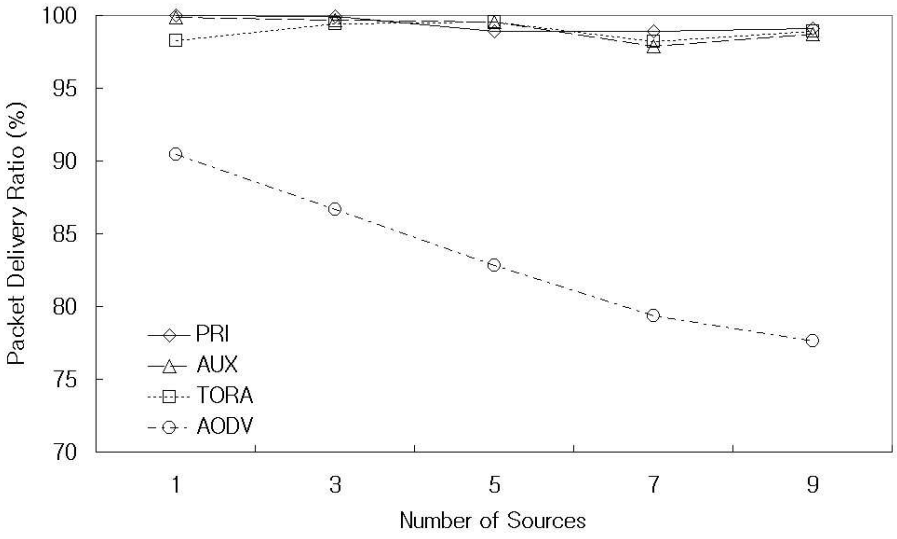


Figure 3.29: Packet delivery ratio vs. number of source nodes per a destination where # nodes is 50, $v_{max} = 10$m/s, beacon period=1s and $\delta = 4096$.

AUX generates 2454.2 control messages, TORA generates 2612.1 control messages while AODV generates 8523 control messages when the number of source nodes per a destination is 9.

3.3.5 Simulation Results Versus the Beacon Period of IMEP

Since both PDR and TORA are implemented on top of IMEP layer, topological changes of networks are detected by IMEP layer which is relied on beacon exchanging scheme of IMEP layer. Note that, as the beacon period is increased, there can be greater chances to detect multiple event as a single event at once. Therefore it is expected that less control messages are required if beacon period is increased, i.e. number of control messages are increased as the beacon period decreased since more topological changes are detected. Besides, path lengths are not affected to the beacon period because path length depends on the distribu-

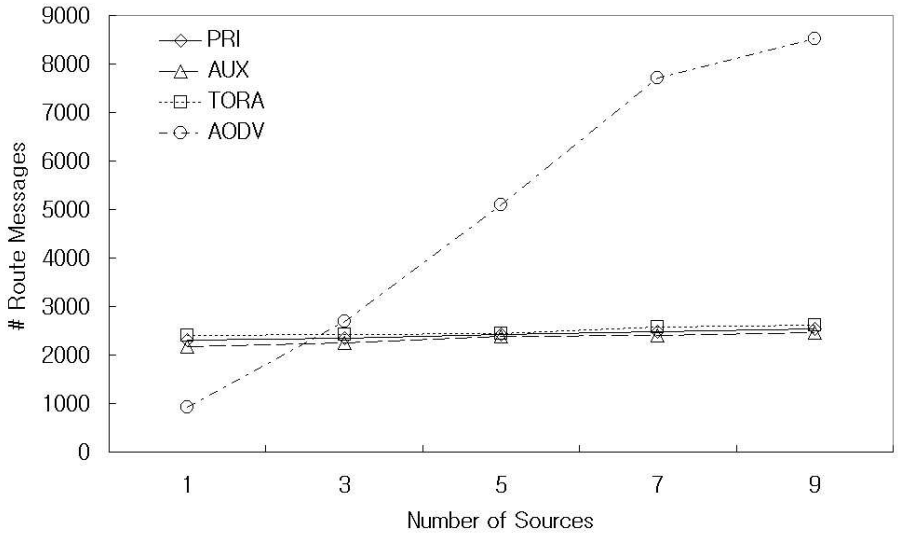


Figure 3.30: Number of control messages vs. number of source nodes per a destination where # nodes is 50, $v_{max} = 10$m/s, beacon period=1s and $\delta = 4096$.

tions of nodes in the network. Packet delivery ratios are also not affected to the beacon period because even if detection of topological changes are delayed, packets can be delivered due to multiple redundant paths to destination. Note that performances of PDR are compared to TORA only in this section since AODV does not rely on IMEP.

Path Length Versus Beacon Period of IMEP

Figure 3.31 shows path length versus beacon period of IMEP. TORA shows worse performance than PRI and AUX. As expected, there is no relation between path length and beacon period.

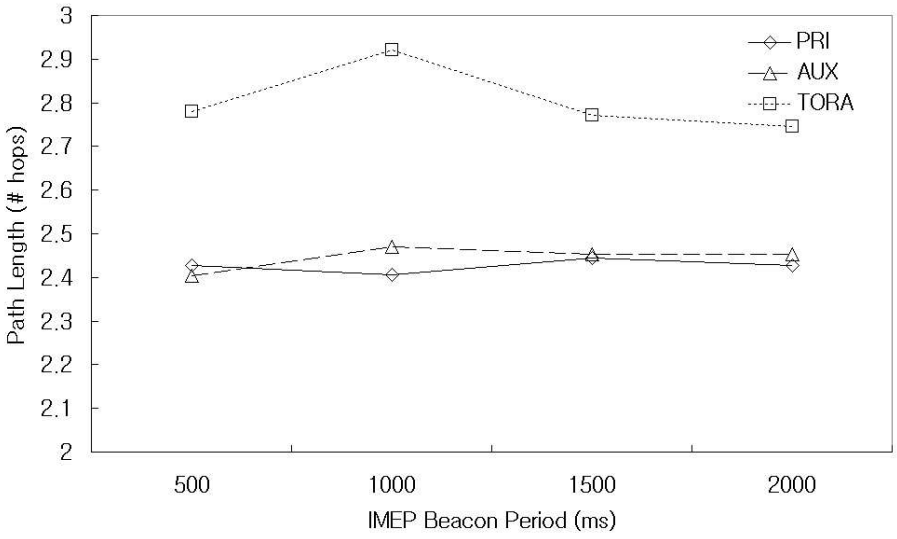


Figure 3.31: Number of control messages vs. beacon period of IMEP where # nodes is 50, $v_{max} = 10$m/s and $\delta = 4096$.

Packet Delivery Ratio Versus Beacon Period of IMEP

Figure 3.32 shows packet delivery ratios versus beacon period of IMEP. As expected, all three protocols shows similar performance in packet delivery ratio. Note that the scale of Y-axis of Fig. 3.32 is differente from previous graphs that begins from 99%. In detail, PRI delivers 99.98% of packets, AUX delivers 99.74% of packets and TORA delivers 99.63% of packets in average. Therefore, there is no meaningful differences in packet delivery ratio versus beacon period of IMEP.

Number of Control Messages Versus Beacon Period of IMEP

Figure 3.33 shows the number of control messages generated versus beacon period of IMEP. Amount of control messages are slightly decreased when beacon

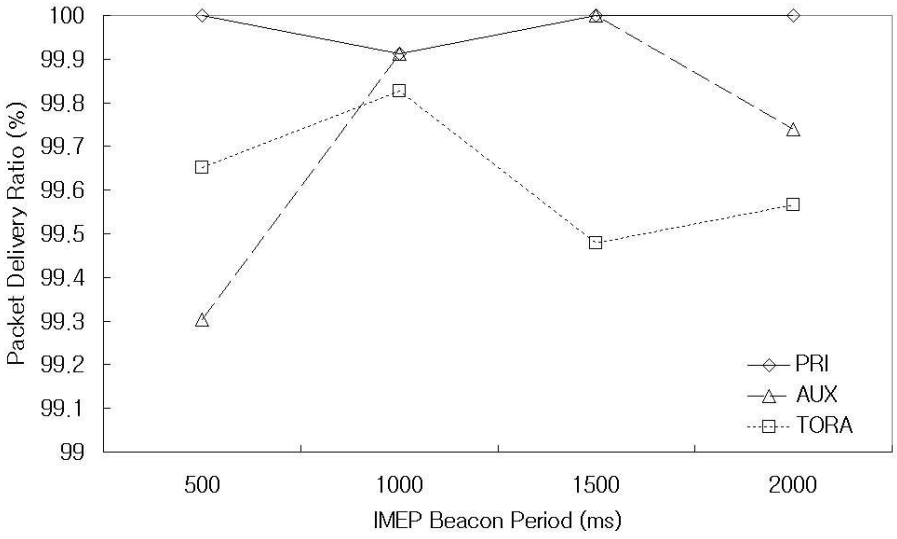


Figure 3.32: Packet delivery ratios vs. beacon period of IMEP where # nodes is 50, $v_{max} = 10$m/s and $\delta = 4096$.

period is 1000ms than beacon period is 500ms as discussed earlier. However, there is no meaningful decrements of control messages when beacon period is longer than 1500ms. As other simulation results, AUX shows less control messages than PRI while TORA generates more control messages than PDR.

3.3.6 Simulation Results Versus δ

In order to decrease amount of control messages, PDR introduces the δ. If δ becomes smaller, the effect of δ gets reduced. In addition, if δ is too small, then it can not reduce control messages after repeated reconstruction of routes. Therefore δ should be sufficiently large. However, large δ requires more storage and control message overhead since it requires more bits to maintain δ. Note that only PRI and AUX utilize the δ.

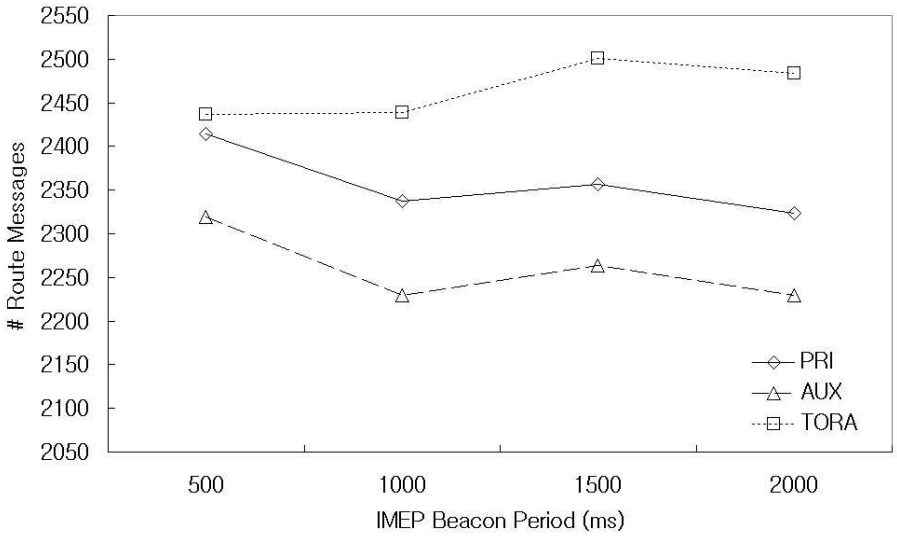


Figure 3.33: Number of control messages vs. beacon period of IMEP where # nodes is 50 nodes, $v_{max} = 10$m/s and $\delta = 4096$.

Path Length Versus δ

Figure 3.34 shows path lengths versus δ. Path lengths of PRI is shortest when $\delta = 1$ while not in AUX. However there is no meaningful differences in path lengths. As other simulation results, PRI shows shorter path lengths than AUX.

Packet Delivery Ratio Versus δ

Figure 3.35 shows packet delivery ratios versus δ. Both PRI and AUX shows almost perfect packet delivery ratio. There is no meaningful relation between packet delivery ratio versus δ.

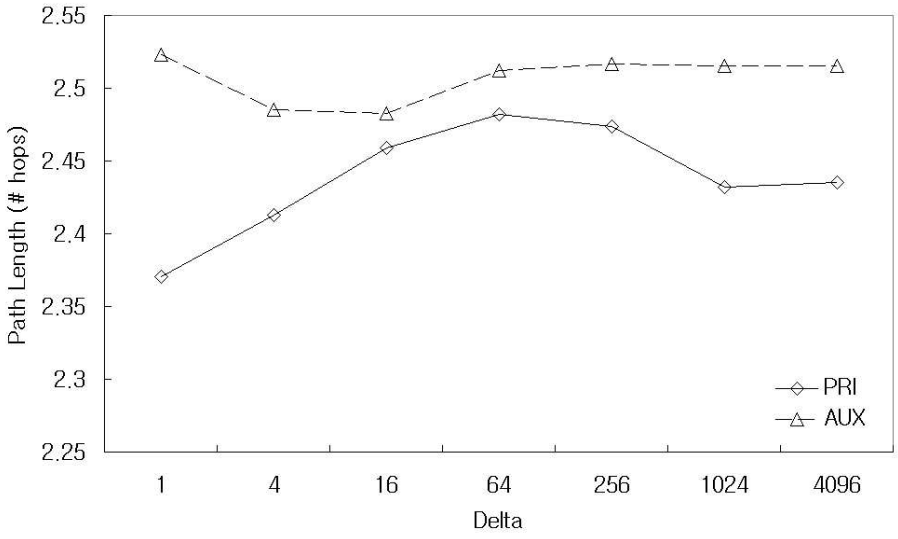


Figure 3.34: Number of control messages vs. δ where # nodes is 50 with v_{max} = 10m/s and beacon period=1s.

Number of Control Messages Versus δ

Figure 3.36 shows amount of control messages versus δ. When δ is 1, more control messages are required but number of control messages are not decreased meaningfully if δ is greater than 4. In detail, PRI generates 1627 control messages when $\delta = 1$ but it generates 1484 control messages when $\delta = 64$ while AUX generates 1430.6 when $\delta = 1$ and 1374.5 when $\delta = 64$.

3.4 Conclusion

In this chapter, a new routing algorithm, termed *pseudo-distance routing (PDR)*, that discovers and maintains short-distance, multiple paths from all nodes in the network to each destination node in the network for MANET is proposed. Analysis of example cases and simulation results show that PDR that includes

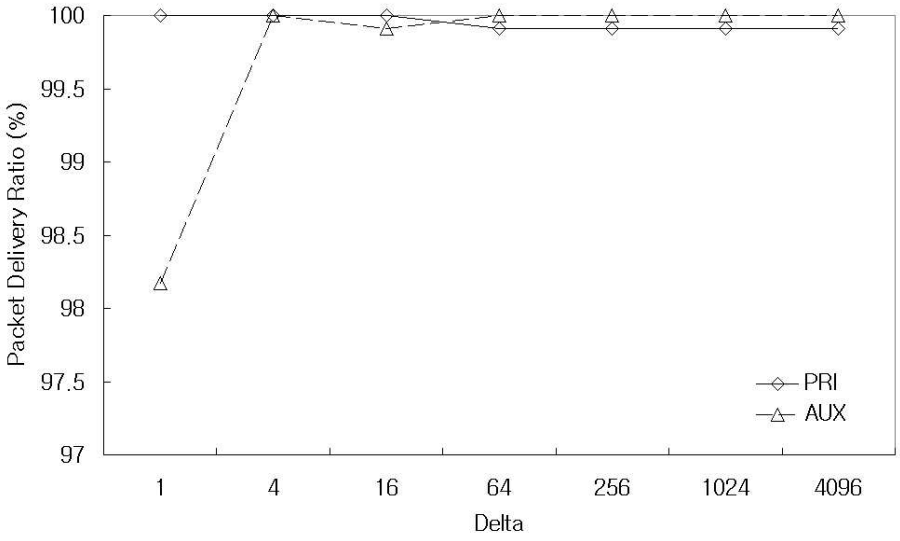


Figure 3.35: Packet delivery ratios vs. δ where # nodes is 50, v_{max} = 10m/s and beacon period=1s.

PRI and AUX provides shorter paths than TORA (a previously proposed routing algorithm with features similar to PDR), while PDR generates comparable number of control messages to TORA. However, PDR generates more control messages than AODV because AODV does not provides multiple paths from multiple nodes in the network. PDR outperforms in packet delivery ratio than AODV and TORA. Since AODV does not support multiple redundant paths to destination, packet delivery ratio of AODV is worse than PDR and TORA. PDR results in a higher packet delivery ratio than TORA because PDR chooses paths with more alternative sub-paths to each destination. Performance of PDR is highlighted when the number of source node per a destination node is increased as described in Section 3.3.4. Due to these features, it is claimed that PDR is a practical routing algorithm for MANET environments. However, PDR assumes global time as TORA in order to detect network partitions. Therefore, as a fu-

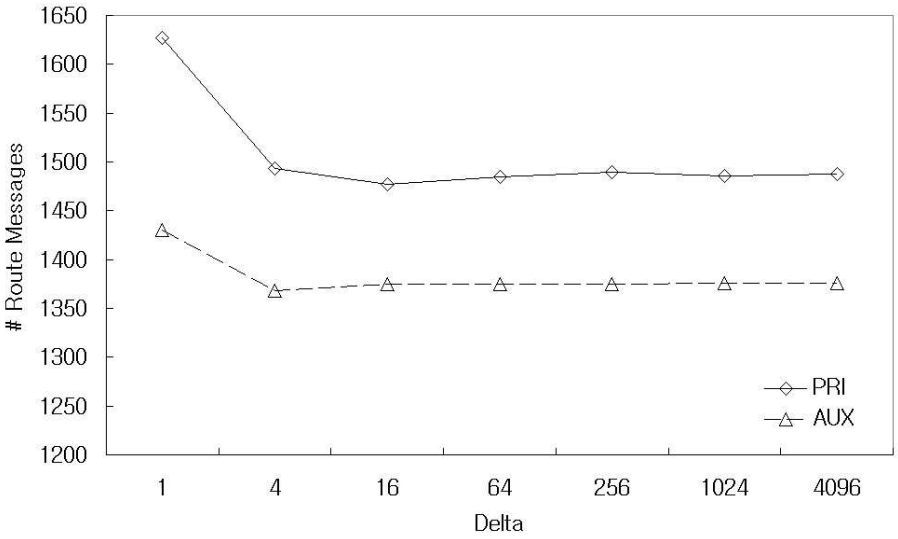


Figure 3.36: Number of control messages vs. δ where # nodes is 50, v_{max} = 10m/s and beacon period=1s.

ture work, the assumption of globally synchronized time should be removed.

4

Link Stability and Stable Routing

A routing algorithm for mobile ad hoc networks (MANETs) should not only route using short-distance paths, but should also be adaptable to highly dynamic changes in network topology since the network topology can change frequently and wireless communication channels are inherently unreliable. Given a routing algorithm targeted toward finding optimal (in terms of distance) paths, the physical distance between two neighboring nodes within a path tends to be very long since this results in fewer hops. Such distances may even be close to the effective transmission range between nodes as shown in [55]. In this case, a small movement of any of the nodes involved may cause packet loss due to link disconnection. Furthermore, packets can be lost due to noise or interference in the

wireless channel if the signal strength is very weak. Therefore, a MANET routing algorithm should not only seek to find short-distance paths, it should also strive to find stable paths that take into account the mobility of nodes, low signal power and interference in the wireless channel.

In this chapter, a new link stability estimation model[1] and a routing algorithm based on this new model are proposed. Section 4.1 reviews previously proposed routing methods that take into account link stability. Section 4.2 discusses various link stability estimation models and the proposed link stability model. Section 4.3 discusses routing algorithms that are able to support stable routing, and Section 4.5 shows simulation results for various link stability models on top of the target routing algorithm. Finally, conclusions are presented in Section 4.6.

4.1 Previous Routing Protocols that Consider Link Stability

Signal stability-based adaptive routing(SSA)[45] estimates link stability based on signal strength. Each node measures signal strengths from other nodes. If a node receives a strong signal from a neighbor, which typically results if two nodes are close to each other, the link is considered as stable. If possible, SSA tries to find a path using only stable links. If it fails to find a stable path, then it tries to find a path using all possible links, resulting in an ordinary path. When a failed link is detected, an intermediate node sends an error message to the source node to notify it that the path is broken. Then the source reinitiates another path search process in order to find a new path – this causes undue overhead and is thus undesirable.

Associativity-based routing(ABR)[44] tries to find long-lived paths to destina-

[1] Preliminary version of this chapter was published in [56].

tions using estimations of link stability based on beacon messages. ABR searches all possible paths to find a path with strong links. Therefore, a path is selected for each destination based on link stability. However, the link stability model that ABR uses is not accurate for some mobility patterns.

Link life based routing protocol(LBR)[57] is another stability-based routing protocol. LBR converts signal strength into distance using a free space propagation model assumption. Based on estimated distance and maximum speed of nodes, LBR estimates link lifetime. When the source node initiates a route request, each intermediate node attaches its estimated link lifetime to the route request message. When the destination receives a route request message, it can calculate the path lifetime for that path based on the estimated link lifetimes in the path. Therefore, the destination can select a path that is expected to have the longest lifetime. In order to react to path breakage, proactive and reactive maintenance is proposed in LBR. In reactive maintenance, the source node needs to reinitiate a route request to the destination, which results in increased delay and control overhead. In proactive maintenance, a backup path is found prior to path breakage. However, the estimated path lifetime is not valid when a path is broken. Therefore, the backup path may be unstable.

The approaches discussed above require the delivery of an error message to the source node followed by reinitiation of route discovery when path breakage is detected. However, reinitiating route discovery is a very costly operation that may not be acceptable for time critical applications such as those requiring QoS routing. Furthermore, the stable routing algorithms discussed above attempt mainly to reduce routing overhead. Even if a stable path is selected when the path is initially discovered, the probability of successful packet delivery (packet delivery ratio) can decrease because the signal strength of links in the path can weaken. The purpose of stable routing should be not only reducing routing overhead but also increasing packet delivery ratio. Therefore, we propose a new stable routing

algorithm that is aimed at increasing packet delivery ratio.

4.2 Link Stability Models

In order to support stable routing, proper estimation of link stability is required. In [58], link stability is modeled in a statistical manner based on node movement models. However, statistical approaches are not adequate for general applications because the mobility patterns of nodes cannot be known a priori. In [44], a link stability estimation model is proposed using periodic beacon signals. In order to estimate link stability, every node sends beacon messages periodically. If the number of continuous received beacon messages are beyond a certain threshold from its neighbor, then the link is considered as stable since ABR is based on the idea that nodes that have been stationary for a threshold period are less likely to move. However, this idea is not so accurate because not all nodes follow the mobility patterns that ABR assumes. The other approaches are based on signal strength. The basic idea is that signal strength weakens if the distance between two nodes grows farther apart. A path composed of weak links can easily become broken. Therefore, a signal strength-based estimation model marks a link as stable if the signal strength of the link is greater than a certain threshold.

Let us use the following notation in discussing link stability models. v_i represents a node with a unique identifier i and $e_{i,j}$ is the link between node v_i and v_j. SS_j is the signal strength of a packet received from node v_j and $SScum_j$ is the cumulative signal strength of packets received from v_j. DSS_j is the differentiated signal strength (i.e., the change in signal strength from the value measured during the previous measurement period) of neighbor v_j. ρ is an weight factor of $SScum_j$ that defines how much previous signal strength affects current $SScum_j$. Finally, Thr is the signal strength threshold above which a signal is considered to be stable.

4.2.1 Signal Strength-Based Link Stability Estimation Model(SBM)

SBM, proposed in [45], estimates link stability using signal strengths. Each node monitors signals from its neighboring nodes. If the signal strength of a received packet is higher than a certain threshold, the link to that neighbor is considered stable. Figure 4.1 shows the pseudocode for the procedure followed by SBM when v_i receives a packet from v_j.

$SScum_j = \rho SScum_j + (1 - \rho)SS_j$

if($SScum_j > Thr$) {

    $e_{i,j}$ is stable.

} else {

    $e_{i,j}$ is unstable.

}

Figure 4.1: Pseudocodes of SBM.

Figure 4.2 shows estimation results for link stability when SBM is used. The circle with 45 degree slash marks (the stable zone) is the area where the signal strength is greater than Thr. Only nodes in the slashed area can be considered as nodes connected by stable links. The vertically slashed circle area (outer circle) is the maximum communication range of v_1. When mobile nodes are inside the small ovals, the link between those nodes and v_1 can be considered as stable. Link $e_{1,2}$ when v_2 is on path segments (1), (2), (5), (6) and (7) is considered as unstable because the signal strength received from v_1 is less than Thr. However the link $e_{1,2}$ is considered as a stable link when v_2 is on path segments (3) and (4) because the signal strength received is greater than Thr. Link $e_{1,3}$ is always considered as unstable because the signal strength received by v_3 is less than the threshold Thr throughout its journey.

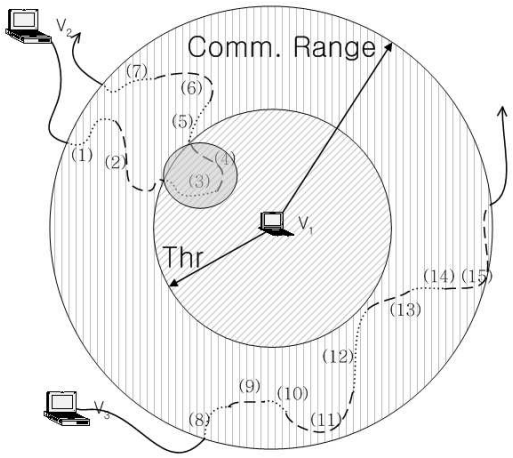


Figure 4.2: Pseudocodes and estimation results of SBM.

4.2.2 Advanced Signal Strength-Based Link Stability Estimation Model(ASBM)

ASBM, proposed in [55], takes differentiated signal strength (DSS) values into account when estimating the direction of node movement. DSS indicates whether the signal strength is getting stronger or weaker. If the signal strength is getting stronger, this means that the two nodes are getting closer together and the link is getting stronger. Therefore links with increasing signal strengths are considered as stable. If the signal strength is getting weaker, this means that the two nodes are getting farther apart and the link may become disconnected. In addition, a very weak initial signal strength between two nodes also indicates a weak link. Thus, a link in which the signal strength is getting progressively weaker or is less than a threshold is considered as unstable. Since ASBM takes DSS into account, it can detect movements of nodes that can weaken link stability. Therefore, the threshold for ASBM can be set lower than the threshold for SBM, which means that the stable area is larger than with SBM. Figure 4.3

shows the pseudocode for ASBM.

```
SScum_j = ρSScum_j + (1 − ρ)SS_j
DSS_j = SScum_j − prevSScum_j
if(SScum_j > Thr) {
  if(DSS_j > 0) {
    e_{i,j} is stable.
  } else {
    e_{i,j} is unstable.
  }
} else {
  e_{i,j} is unstable.
}
prevSScum_j = SScum_j
```

Figure 4.3: Pseudocodes of ASBM.

Figure 4.4 shows estimated results for link stability when ASBM is used. Note that the area of the stable zone for ASBM is larger than that for SBM because Thr of ASBM is less than Thr of SBM. When v_2 is on path segments (2), (3), (4), (5) and (6), v_2 is inside the stable zone. DSS_2 is positive when v_2 is on path segments (2) and (3) because v_2 and v_1 are getting closer. Therefore, the link $e_{1,2}$ is considered as stable when v_2 is on path segments (2) and (3). However, DSS_2 is negative when v_2 is on path segments (4), (5) and (6) because v_2 and v_1 are getting farther apart. Therefore, link $e_{1,2}$ is considered as unstable when v_2 is on path segments (4), (5) and (6). Note that when v_2 is on a path segment (4), the distance between v_1 and v_2 is very close. Even if v_2 starts to move out immediately, it can be considered as stable because it may need a lot of time to move out of the communication range of v_1. Therefore, ASBM may result in

fewer stable links than SBM. Based on similar reasoning, link $e_{1,3}$ is considered as stable when v_3 is on path segments (9) and (12).

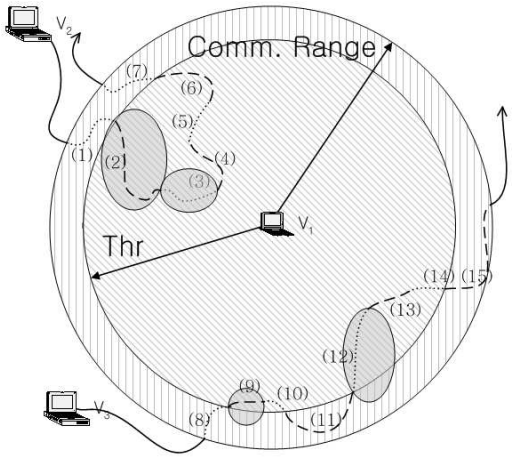


Figure 4.4: Estimation results of ASBM.

4.2.3 Enhanced Stability Model (ESM)

A major shortcoming of ASBM is that it considers the link $e_{1,2}$ as unstable when v_2 is on a path segment (4) in Fig. 4.4. In order to overcome this shortcoming of ASBM, we propose a new link stability estimation model, termed the Enhanced Stability Model (ESM), that uses two thresholds. In ESM, a link is considered as stable when two nodes are located very close to each other. ESM uses two threshold Thr_1 and Thr_2 with the property $Thr_1 > Thr_2$. If the signal strength is greater than Thr_1, then the link is always considered as stable because the distance between the two nodes is very small. However, if the signal strength is less than Thr_1 but greater than Thr_2, then DSS is used to estimate link stability as in ASBM. In addition, due to external environment factors like obstacles, interference and white noise, signal strength can decrease even when the locations of both nodes are fixed.

$SScum_j = \rho SScum_j + (1-\rho)SS_j$

$DSS_j = SScum_j - prevSScum_j$

if($SScum_j > Thr_1$) {

  $e_{i,j}$ is stable.

} else if($SScum_j > Thr_2$) {

  if($DSS_j > \mu$) {

    $e_{i,j}$ is stable.

  } else {

    $e_{i,j}$ is unstable.

  }

} else {

  $e_{i,j}$ is unstable.

}

$prevSScum_j = SScum_j$

Figure 4.5: Pseudocodes of ESM.

Suppose that signal strength is slightly decreased by external environment factors. In this case, ASBM considers the link as unstable because DSS becomes negative even though the actual link may still be stable. Therefore, we add a parameter μ where $\mu < 0$ to address this problem. A link is considered as unstable in ESM only when $DSS < \mu$. Figure 4.5 shows the pseudocode for ESM.

Figure 4.6 shows the estimated results for ESM. Path segments (2) and (3) are considered as stable because the signal strength is greater than Thr_2 and $DSS_2 > 0$. However, a path segment (4), which was considered as an unstable link in ASBM, is considered as a stable link in ESM because the signal strength for v_2 is greater than Thr_1 even if two nodes are getting farther apart. In addition, a path segment (5) is also considered as stable in ESM because $DSS_2 > \mu$ even though v_2 is moving toward the outside of the communication range of v_1. However, a path segment (6) is considered as unstable because $DSS_2 < \mu$. Path segments (9) and (10) are considered as stable because $SScum_3 > Thr_2$ and $DSS_3 > \mu$ even though v_3 is moving toward the outside of the transmission range of v_1 when v_3 is on a path segment (10). Furthermore, path segments (12) and (13) are also considered as stable for the same reason.

4.3 Stable Pseudo-Distance Routing (S-PDR) Algorithm

Since link stability continually changes in a MANET, a routing algorithm for such a network should be able to dynamically adapt to link stability changes in selecting a path to each destination. However, most previous algorithms provide only a single path to each destination. Thus, once a path has been selected, new link stability information cannot be used to change the path to the destination. Unlike such rigid algorithms, TORA[42] and PDR provide multiple paths, and a

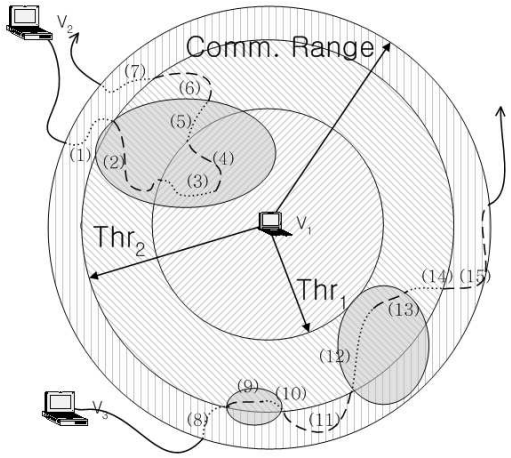


Figure 4.6: Estimation results of ESM.

path to each destination can be selected on a hop-by-hop basis. The most recent link stability information for each link can be used in making hop-by-hop routing decisions. Of these two algorithms, PDR is chosen as our base routing algorithm because PDR shows better performance than TORA as shown in previous chapter.

Note that there are two types of links in PDR. Primary links are mainly used to route packets along shortest-distance paths. Auxiliary links are used when all primary links are broken. User can select whether auxiliary outgoing link can be used to forward packets or not because auxiliary outgoing links tend to be detour to the destination. PRI is abbreviation of primary only routing that auxiliary links are excluded in routing, and AUX is abbreviation of auxiliary routing that auxiliary outgoing links are included in routing. Note that PRI shows shorter path than AUX but route overhead in terms of control messages are increased than AUX.

PDR provides multiple paths to destination, but it does not take link stability into account. Therefore, we need to modify the PDR algorithm to select

stable links. This modified algorithm is referred to as the stable pseudo-distance routing (S-PDR) algorithm. Since S-PDR already requires each node to store information for each of its neighbors, we can simply add a variable that represents stability into this neighbor information table. The "estimated" stability of a link $e_{i,j}$ is updated whenever a node receives packets from its neighbors using one of the estimation models discussed in Section 4.2. When a node selects its next hop, S-PDR selects a neighbor with the minimum height from among the nodes connected by stable links. If there are no stable outgoing links, S-PDR simply selects a minimum-height neighbor in order to reduce the path length as in PDR. Note that S-PDR can be divided into PRI and AUX parts as same as PDR.

4.4 Selecting Threshold Values for S-PDR

Performance of ESM depends on threshold values: Thr_1 and Thr_2. This section provides a guideline how to select threshold values. There are several propagation models such as free space model [59], two-ray ground reflection model [60, 61], shadowing model [60], etc. Shadowing model is generalized and widely accepted model among them. However, shadowing model is too complex to analyze since it is a statistic model. On the other hand, receiving signal power in free space model and two-ray ground model is a function of physical distance d between two radios. Therefore it is easy to analyze receiving signal power in hand. Note that as discussed in Section 1.1.2, radio signal is propagated following multiple paths that decreases the quality of signal at receiver. Since two-ray ground model considers both a direct path and a reflected path to the ground while free space model only considers direct path, two-ray ground model is used to analyze parameters of ESM.

4.4.1 Assumptions

Maximum speed of a node is assumed as v_{max} and transmission range of radio signal is assumed that d_r. Minimum signal strength that a radio can successfully receive is assumed $RXThresh\_$. Beacon messages are periodically transmitted at every t_b seconds to notify its signal strength without any delay or jitter. Note that PDR periodically exchanges beacon messages in order to detect topological changes. Each node receiving the beacon message estimates link stability using the received beacon message from its neighbors. Therefore, link stability is re-estimated once at every t_b seconds. Received signal power in two-ray ground model when distance between two radio devices is d is estimated by

$$P_r(d) = \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{d^4 L} \tag{4.1}$$

where P_t is transmitted signal power, G_t is antenna gain at the transmitter, G_r is antenna gain at the receiver, h_t is height of antenna at the transmitter, h_r is height of antenna at the transmitter, h_r is the height of antenna at the receiver, and L is the system loss. Finally, there is no packet loss due to other external environments for simplicity in this section.

4.4.2 Selecting Threshold Values

Thr_1 should be a safety condition that guarantees that both nodes are within their communication range even they are moving toward opposite directions until next estimation time of link stability. Since estimation of link stability is triggered when a node receives a beacon message from its corresponding neighbor, next estimation time of link stability is t_b. The maximum physical distance that a node can move within a beacon period is $v_{max}t_b$, that introduces the maximum difference in physical distance between two nodes during t_b to $d_{max} = 2v_{max}t_b$. Suppose that there are two nodes v_1 and v_2. Then the link $e_{1,2}$ can be safely

estimated as stable at v_1 if received signal strength of current beacon message is greater than the signal strength of a node that physical distance is $d_r - d_{max}$. Therefore a link is estimated as stable if received signal strength $RxPr$ meets following inequality.

$$RxPr_{stable} \geq P_r(d_r - d_{max}) = \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{(d_r - d_{max})^4} \tag{4.2}$$

Therefore, we can set Thr_1 as $P_r(d_r - d_{max})$. It would be very useful that express Thr_1 as a function of $RXThresh\_$ instead of absolute signal strength since absolute value is difficult to interpret. Since $P_r(d_r) = RXThresh\_$ and we can express the minimum signal strength as $RxPr_{stable} = \sigma RXThresh\_$, we can rewrite the Inequality (4.2) as shown below:

$$\begin{aligned} \sigma RXThresh\_ &\geq P_r(d_r - d_{max}) \\ \sigma \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{{d_r}^4} &\geq \frac{P_t G_t G_r {h_t}^2 {h_r}^2}{(d_r - d_{max})^4} \\ \sigma &\geq \frac{{d_r}^4}{(d_r - d_{max})^4} \end{aligned} \tag{4.3}$$

Strict estimation of link stability may decrease packet delivery ratio because it reduces the number of stable links. Since S-PDR simply select a link with minimum height metric that are more likely to lose data packets if there is no stable links, it would be better to have more stable links even though it can not guarantee packet delivery during the beacon interval. It is also useful to express Thr_2 as a function of $RXThresh\_$ as Thr_1. Therefore Thr_2 can be expressed as $Thr_2 = \kappa RXThresh\_$. κ can be any value within $(1.0, \sigma)$. Note that a node has been getting closer together when stability is estimated if links are estimated as stable. Therefore we can expect that physical distance of two nodes are not increased due to inertia. If S-PDR tries to guarantee delivery of data packets

during only half of beacon period, then we can obtain κ as

$$\kappa \geq \frac{{d_r}^4}{(d_r - \frac{d_{max}}{2})^4} \tag{4.4}$$

4.5 Simulation Results

Simulations were conducted to evaluate the performance of the various stability models considered and to evaluate the benefits of routing using stable links. The simulation tool used was ns-2[53], which is a discrete event simulator commonly used in networking research. In order to model wireless connections accurately, the distributed coordination function (DCF) of the IEEE 802.11 standard for wireless LANs was used for the MAC and PHY layers. The data rate was set to 11 Mbps as this is a rate supported by the most common IEEE 802.11b devices.

The simulation scenarios used were based on the following setup. The simulation space was a 1500m $\times$ 500m area, and the communication range of each node was set to 250m. The mobility of the nodes was controlled by a mobility generator function in ns-2 that uses a random destination model with 20m/s maximum speed. Finally, the simulation time was set to 130 seconds. A source sends 256 bytes of UDP packet data to its randomly chosen destination at every 0.2 second from 10 seconds after the simulation starts to 125 seconds. Data were collected for 20 different simulation scenarios. Other simulation parameters are shown in Table 4.1 that are the same as in Chap. 3.

4.5.1 Error Model used in Simulation

The error model used is a modification of the basic ns-2 error model. Basically, all packets in ns-2 are successfully received if the signal power is greater than the receiving threshold. In ns-2, each node that receives a packet calculates

Table 4.1: Constants used in simulation.

Radio model	Two-ray ground
RTS/CTS	Enabled
Preamble length of IEEE 802.11	Short preamble (72 bits)
Carrier Sensing Threshold	1.559e-11
Receiving Threshold	3.652e-10
Carrier Frequency	914e+6
Transmitted Signal Power	0.28183815
System loss	1.0
Antenna gain at transmitter	1.0
Antenna gain at receiver	1.0
Antenna height	1.5

its receiving signal strength $RxPr$ using a propagation model based on a free-space, two-ray ground reflection or shadowing model. If the calculated $RxPr$ is greater than $RXThresh\_$, the threshold of the receiving packet, then the packet is successfully received. However, a packet received with a weak signal strength can easily be corrupted or lost due to various external environment factors such as white noise, wireless interference and other circumstances in actual wireless networks. Therefore, the ns-2 error model was modified to simulate a more reasonable error model. In our implementation, we update the receiving signal strength as $RxPr = RxPr - [0, RXThresh\_ \times MASS]$, where $MASS$ is a floating-point value in [0, 1]that represents the maximum attenuation of the signal strength. If the signal power $RxPr$ is greater than $(1 + MASS) \times RXThresh\_$, the packet is always successfully received. Otherwise, the packet can be lost with a random probability factor. Note that as $MASS$ is increased, the probability of packet loss also increases.

4.5.2 Selecting Parameters

Following parameters above, we can calculate appropriate parameters. At first, Thr_1 of ESM can be set as $Thr_1 = \rho RXThresh\_$ where $\rho = \frac{(d_r)^4}{(d_r - d_{max})^4} \approx 2.00$ and Thr_2 can be set as $Thr_2 = \kappa RXThresh\_$ where $\kappa \approx 1.40$ as the Inequality (4.4). In ESM, μ is used to overcome attenuation of signal strength by external environments. We modified the error model as Section 4.5.1, maximum attenuation of signal strength is limited to $RXThresh\_ \times MASS$. Therefore μ should be selected a value within $[0, MASS]$. We simply set $\mu = MASS/2$ as the average of $[0, MASS]$.

For the simulation, Thr of SBM is set as $2.0 \times RXThresh\_$ and Thr of ASBM is set as $1.4 \times RXThresh\_$. In ESM, ρ is set as 2.0, κ is set as 1.4, and μ is set as $-0.1 \times RXThresh\_$. Therefore, Thr_1 is $2.0 \times RXThresh\_$ and Thr_2 is $1.4 \times RXThresh\_$ for the simulation.

4.5.3 Performance of Primary Routing (PRI)

Figure 4.7 shows packet delivery ratio versus $MASS$. The plot for PRI-NONE shows the results for PRI without a stability estimation model, and the PRI-SBM plot shows the results for PRI with the stability estimation model used in SBM. The PRI-ASBM plot shows the results for PRI with the estimation model of ASBM, and the PRI-ESM plot shows the performance of PRI with the estimation model of ESM. As expected from Section 4.5.1, the packet delivery ratio is decreased if $MASS$ is increased and vice versa. Because PRI-NONE does not take link stability into account, it shows the worst performance in terms of packet delivery ratio. However, since PRI-NONE selects next-hop nodes from among minimum-height neighbors, path lengths produced by PRI-NONE should be the shortest. PRI-ASBM shows the worst performance, in terms of packet delivery ratio, among the stable routing algorithms. Note that the number of

stable links are fewer than with other methods because, even if two nodes are very close, ASBM excludes links from the stable link list when $DSS < 0$. The result is that ASBM usually selects its next hop node from among minimum-distance-path neighbors as in PRI-NONE, thereby producing poor performance in terms of packet delivery ratio. However, the performance of ASBM in terms of path length is good. PRI-ESM shows the best performance as $MASS$ is increased because the link stability estimation method used by ESM is very accurate.

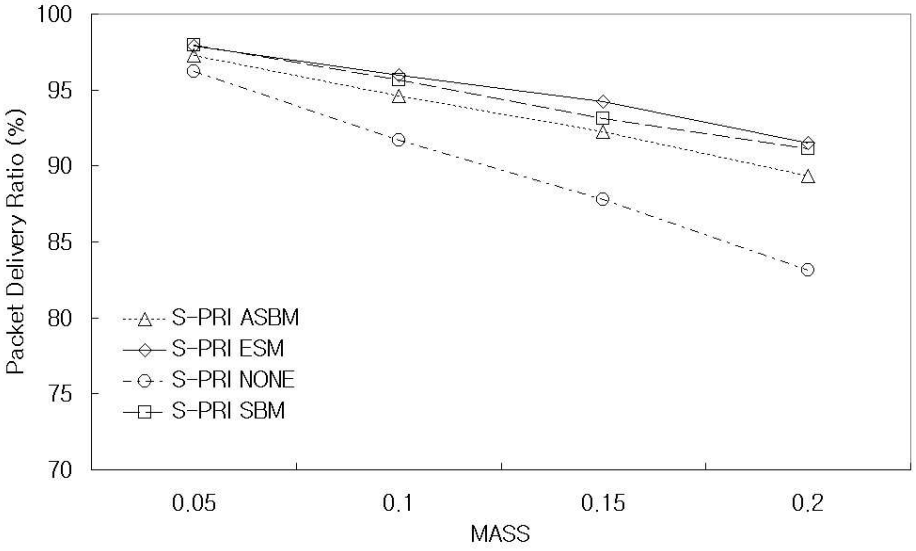


Figure 4.7: Packet delivery ratio of PRI routing using various link stability models.

Figure 4.8 shows path length versus $MASS$. PRI-NONE shows better performance than all other algorithms because PRI-NONE only selects minimum-distance-path neighbors. As expected, PRI-ASBM shows the best performance among the S-PDR variants in terms of path length because PRI-ASBM tends to select its next hop using unstable links (selecting a minimum-distance-path using those links) because the number of stable links are fewer than with the

other methods. Path lengths for PRI-ASBM and PRI-ESM are greater than PRI-NONE because these former algorithms select stable paths even if detours are necessary. PRI-SBM shows the worst performance in terms of path length because the next stable-hop-node is located relatively closer than with other methods. Note that the stable areas for PRI-ASBM and PRI-ESM are larger than for PRI-SBM.

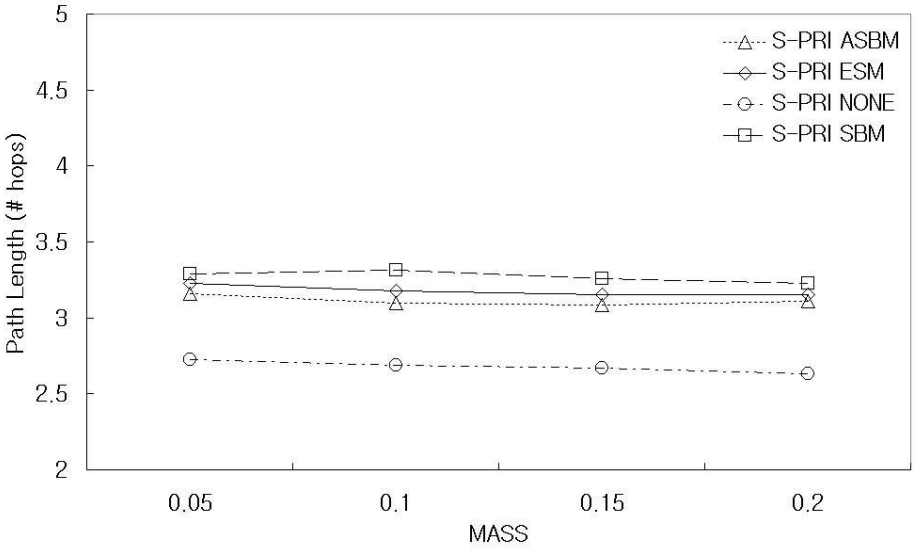


Figure 4.8: Path lengths of PRI routing using various link stability models.

4.5.4 Performance of Auxiliary Routing (AUX)

Figure 4.9 shows packet delivery ratio versus $MASS$. AUX-NONE performs worst in terms of packet delivery ratio because AUX-NONE does not take link stability into account (like PRI-NONE). AUX-ASBM also performs worst in terms of packet delivery ratio among the stable routing algorithms because the number of stable links is fewer than other methods (like PRI-ASBM). As shown in the figure, AUX-ESM performs best in terms of packet delivery ratio as expected.

As $MASS$ is increased, the performance gap between ESM and other methods also increases. ESM outperforms all other link stability models considered when the wireless communication channels used become very unreliable. Note that the packet delivery ratio for AUX is greater than that for PRI because AUX routing utilizes more outgoing links — it considers both primary outgoing links *and* auxiliary outgoing links when searching for stable links.

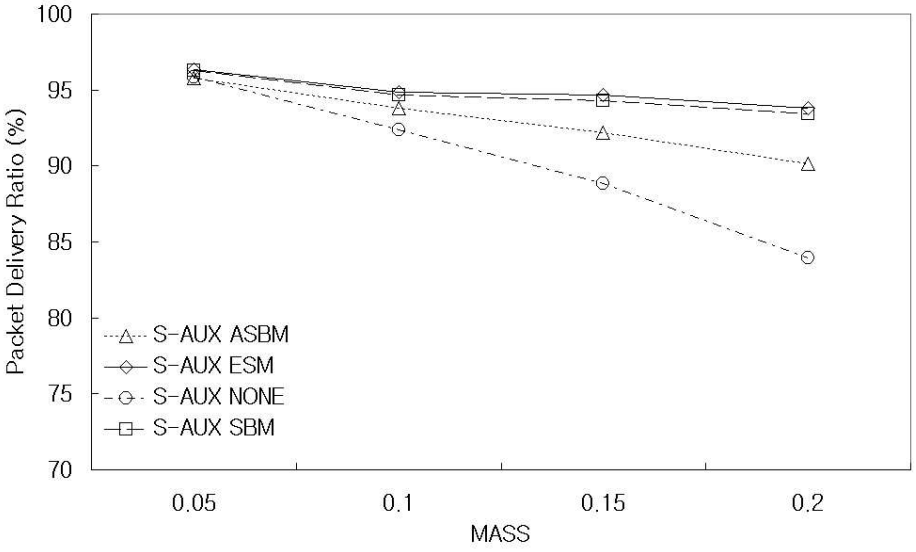


Figure 4.9: Packet delivery ratio of PRI routing using various link stability models.

Figure 4.10 shows path length versus $MASS$ for AUX routing. As expected, the path length of AUX-NONE is the shortest, and the path length of AUX-ASBM is the second-shortest as in the PRI case. AUX-SBM performs the worst, in terms of path length, for the same reason as in the case of PRI routing. AUX-ESM performs worse than AUX-ASBM in terms of path length. Nevertheless, the difference in packet delivery ratio, which is our main concern in this chapter, favors the AUX-ESM method.

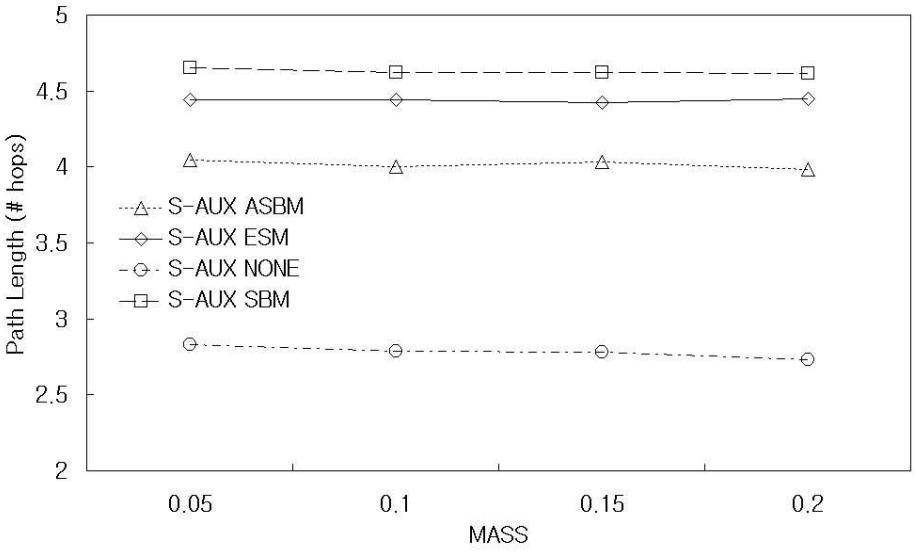


Figure 4.10: Path lengths of PRI routing using various link stability models.

4.6 Conclusion

In this chapter, a new link stability estimation model, termed enhanced stability model (ESM), that can be used to estimate the stability of communication links in MANETs is proposed. Analysis of example cases and simulation results show that ESM-based routing tends to perform better than routing using previous link stability estimation models in terms of the ratio of packets successfully delivered to their destinations (packet delivery ratio). Furthermore, as the reliability of the channel gets worse, the relative benefit of ESM-based routing becomes more pronounced.

5

Concluding Remarks and Discussion

Wireless networking area has rapidly became a crucial component of computer netowrks and the demand of wireless networking has beencgrown exponentially in the past decade. Many researchers and users are considers mobile ad hoc network which is a wireless networking without infrastructures for their application due to its property of infrastructureless. However, efficiently supporting of routing of MANETs is essential to their applications due to infrastructureless property of ad hoc network. Therefore, there has been significant interest in routing algorithms for mobile ad hoc networks in the recent past.

There are three major categories for routing protocols: *proactive, reactive* and *hybrid*. Proactive routing protocols maintains up-to-date consistent routing

information among nodes in the network which introduces significant control traffics. On the other hand, reactive routing protocols do not maintain up-to-date routing tables but they discover routes when initiated by source nodes. However, reactive protocols also have several shortcomings such as costly route discovery, low packet delivery ratio, etc. Therefore hybrid routing protocols that combine best features of proactive and reactive routing protocols were proposed.

The goals of routing protocols for MANETs are presented in Section 2.1.1 as (1) fully distributed operation, (2) minimal control overheads, (3) minimal processing overheads, (4) loop-free, (5) high packet delivery ratio, (6) multiple paths, (7) quick convergence, (8) localized maintenance of route, (9) minimal path, (10) scalability. Even a lot of routing protocols are proposed, no routing protocols satisfies all goals of routing protocols. In order to provide all properties of listed above, a new routing protocol termed *pseudo-distance routing(PDR)* was proposed in this dissertation.

PDR is based on link reversal algorithm that was firstly proposed in [2]. Even if partial and full reversal algorithms provide distributed operations, minimized control overheads, minimal processing overheads, loop-free, high packet delivery ratio, multiple paths, quick convergence, localized maintenance of routes and scalability, they are not able to detect network partition which causes unlimited interation of link reversals. In order to overcome of unlimted iteration of link reversals, TORA [42] was proposed. However, TORA does not consider route optimality in terms of path length. Therefore path length in TORA gets longer as repeated iterations. On the other hand, PDR, which is also a link reversal algorithm, provides short paths and able to detect network partitions.

Analysis of example cases and simulation results show that PDR which include PRI and AUX provides shorter paths than TORA and AODV. In addition, PDR provides better packet delivery ratio than TORA and AODV while comparable amount of control messages to TORA. However, PDR requires more control

messages than AODV because PDR provides multiple paths for all nodes in the network. Even if PDR requires more control messages than AODV, the amount of control messages of AODV is more rapidly increased than PDR if the number of source nodes per a destination node is increased since control messages of AODV is directly related to the number of source to destination pairs and network size. Note that control messages of PDR is propotional to the number of destination and network size. Therefore, if the number of source nodes per a destination nodes is over 3, then PDR is more efficient in terms of control messages than AODV as shown in Section 3.3.4. However, PDR assumes global time as TORA in order to detect network partition. As a future work, assumption of global time should be removed.

In addition to routing protocol, link stability estimation model is also proposed in Chap. 4. As stated above, routing protocol should deliver data packets to destination correctly. However, due to dynamicity of MANETs and other external environments, packets are likely to be lost. Therefore, routing algorithm should provide stable paths to destination. In order to select stable paths, link stabilty should be estimated. *Enhanced stability model(ESM)* was proposed to estimate link stablilty accurately. By incorporating two threshold and differentiation of received signal strength, ESM estimates link stability more accurately. Furthermore, S-PDR is also proposed that supports stable routing based on link stability information. S-PDR selects its next hop as the minimum height neighbor among the links that are estimated as stable in first round. If first round attempt is failed, that represents no stable outgoing links, then it select the minimum height neighbor as normal PDR does. Therefore, if the number of links that are estimated as stable is not enough, then S-PDR provides shorter path than others but packet delivery ratio gets worse.

Analysis of example cases and simulation results show that ESM can estimate link stability more accurate than others such as SBM [45] and ASBM [55]. ESM

outperforms than others in terms of Packet delivery ratio. However, path length is longer than ASBM since ASBM tends to choose unstable but shorter links to destination due to its inaccurate link estimation.

Since S-PDR selects a minimum height neighbor as a next hop if there is no stable outgoing links, there is more chances to lose data packets. Therefore, it would be helpful to select the most stable links among outgoing links. Towards that ends, link stability estimation model should provide normalized value based on its link stability. Then S-PDR should be able to select next hop that are expected to be more stable and shorter paths.

Bibliography

[1] I. F. Akyildiz, X. Wang, and W. Wang, "Wireless mesh networks: a survey," *Computer Networks*, vol. 47, pp. 445–487, Mar. 2005.

[2] E. M. Gafni and D. P. Bertsekas, "Distributed algorithms for generating loop-free routes in networks with frequently changing topology," *IEEE Transactions on Communications*, vol. 29, pp. 11–18, Jan. 1981.

[3] IEEE Standards Department, "IEEE Standard 802.11-1997: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications," 1997.

[4] Bluetooth SIG, "Specification of the Bluetooth System 1.2," 2004.

[5] J. Kardach, "Bluetooth architecture overview," *Intel Technology Journal*, vol. Q2, May 2000.

[6] ETSI - Broadband Radio Access Networks, "HIgh PErformance Radio Local Area Network (HIPERLAN) Type 1; Functional specification," 1998.

[7] T. Wilkinson, T. G. C. Phipps, and S. K. Barton, "A report on HIPERLAN standardization," *International Journal of Wireless Information Networks*, vol. 2, pp. 99–120, Apr. 1995.

[8] T-Mobile, "Get more from your mobile life." http://hotspot.t-mobile.com/services_about.htm.

[9] KT NESPOT, "KT NESPOT Service." http://www.nespot.com/web/eng/service/as.html.

[10] G. Geier, *Wireless LANs: Implementing High Performance IEEE 802.11 Networks*. Indianapolis, Indiana, USA: SAMS, 2001.

[11] C. S. R. Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*. Upper Saddle River, New Jersey, USA: Prentice Hall, 2004.

[12] T. Krag and S. Buettrich, "Wireless mesh networking." http://www.oreillynet.com/pub/a/wireless/2004/01/22/wirelessmesh.html, Jan. 2004.

[13] N. R. Group, "Self-organizing neighborhood wireless mesh networks." http://research.microsoft.com/mesh/.

[14] G. Holland and N. Vaidya, "Analysis of TCP performance over mobile ad hoc networks," in *Proceedings of the 5th annual ACM/IEEE International Conference on Mobile Computing and Networking*, pp. 219–230, Aug. 2002.

[15] K. Chandran, S. Raghunathan, S. Venkatesan, and R. Prakash, "A feedback based scheme for improving TCP performance in ad-hoc wireless networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 34–39, 2001.

[16] J. Liu and S. Singh, "ATCP: TCP for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 19, pp. 1300–1315, Jul. 2001.

[17] Z. Fu, X. Meng, and S. Lu, "How bad TCP can perform in mobile ad hoc networks," in *Proceedings of the IEEE International Symposium on Computers and Communications*, pp. 298–303, Jul. 2002.

[18] A. Ahuja, S. Agarwal, J. P. Singh, and R. Shorey, "Performance of TCP over different routing protocols in mobile ad-hoc networks," in *Proceedings of IEEE Vehicular Technology Conference(VTC 2000)*, pp. 565–569, May 2000.

[19] T. D. Dyer and R. V. Boppana, "A comparison of TCP performance over three routing protocols for mobile ad hoc networks," in *Proceedings of the*

2nd ACM International Symposium on Mobile Ad Hoc Networking & Computing, pp. 56–66, Oct. 2001.

[20] K. Tang and M. Gerla, "Fair sharing of MAC under TCP in wireless ad hoc networks," in *Proceedings of IEEE MMT'99*, Oct. 1999.

[21] S. Xu and T. Saadawi, "Does the IEEE 802.11 MAC protocol work well in multihop wireless ad hoc networks?," *Communications Magazine, IEEE*, vol. 39, no. 6, pp. 130–137, 2001.

[22] S. Xu and T. Saadawi, "Revealing the problems with 802.11 medium access control protocol in multi-hop wireless ad hoc networks," *Computer Networks*, vol. 38, no. 4, pp. 531–548, 2002.

[23] K. Xu, S. Bae, S. Lee, and M. Gerla, "TCP behavior across multihop wireless networks and the wired internet," in *Proceedings of the 5th ACM International Workshop on Wireless Mobile Multimedia*, pp. 41–48, 2002.

[24] D. P. Bertsekas and R. G. Gallager, "Distributed asynchronous bellman-ford algorithm," in *Data Networks*, ch. 5.2.4, pp. 325–333, Prentice Hall, Englewood Cliffs, 1987.

[25] R. Bellman, "On a routing problem," *Quarterly of Applied Mathematics*, vol. 16, no. 1, pp. 87–90, 1958.

[26] L. F. Jr., "Network flow theory," Paper P-923, The RAND Corperation, Santa Moncia, California, August 1956.

[27] C. E. Perkins and P. Phagwat, "Highly dynamic destination-sequenced distance vector routing (DSDV) for mobile computers," in *Proceedings of ACM SIGCOMM*, pp. 234–244, Oct. 1994.

[28] S. Murthy and J. J. Garcia-Luna-Aceves, "A routing protocol for packet radio networks," in *Proceedings of ACM/IEEE MOBICOM*, pp. 86–95, Nov. 1995.

[29] C.-C. Chiang, H.-K. Wu, W.Liu, and M. Gerla, "Routing in clustered multihop, mobile wireless networks," in *Proceedings of IEEE Singapre International Conference on Networks*, pp. 197– 211, Apr. 1997.

[30] T. H. Clausen, G. Hansen, L. Christensen, and G. Behrmann, "The optimized link state routing protocol, evaluation through experiments and simulation," in *Proceedings of IEEE Symposium on Wireless Pseronal Mobile Communications*, Sept. 2001.

[31] A. Iwata, C. Chiang, G. Pei, M. Gerla, and T. Chen, "Scalable routing strategies for ad-hoc wireless networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1369–1379, 1999.

[32] G. Pei, M. Gerla, and T.-W. Chen, "Fisheye state routing in mobile ad hoc networks," in *ICDCS Workshop on Wireless Networks and Mobile Computing*, pp. D71–D78, 2000.

[33] T.-W. Chen and M. Gerla, "Global state routing: A new routing scheme for ad-hoc wireless networks," in *IEEE International Conference on Communications 98*, vol. 1, pp. 171–175, Jun. 1998.

[34] D. B. Johnson and D. A. Maltz, *Dynamic Source Routing in Ad Hoc Wireless Networks*. Kluwer Academic Publishers, 1996.

[35] C. E. Perkins and E. M. Royers, "Ad hoc on-demand distance vector routing," in *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, pp. 90–100, Feb. 1999.

[36] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing." draft-ietf-manet-aodv-09.txt, Nov. 2001.

[37] S.-J. Lee, E. M. belding Royer, and C. E. Perkins, "Scalability study of the ad hoc on-demand distance vector routing protocol," *International Journal of Network Management*, vol. 13, pp. 97–114, Mar. 2003.

[38] S.-Y. C. Michael Pan and S.-D. Wang, "Local repair mechanisms for on-demand routing in mobile ad hoc networks," in *Proceedings of 11th Pacific Rim International Symposium on Dependable Computing*, Dec. 2005.

[39] M. K. Marina and S. R. Das, "On-demand multipath distance vector routing in ad hoc networks," in *Proceedings of IEEE International Conference on Network Protocols*, pp. 14–23, Nov. 2001.

[40] M.-S. Kim, K. J. Kwon, M. Y. Chung, T.-J. Lee, and J. Park, "A modified AODV protocol with multi-paths considering classes of services," in *Computational Science and Its Applications 2004 - LNCS 3043*, pp. 1159–1168, 2004.

[41] S. J. Lee and M. Gerla, "AODV-BR: backup routing in ad hoc networks," in *Proceedings of IEEE Wireless Communications and Networking Conference(WCNC2000)*, vol. 3, pp. 1311–1316, Sept. 2000.

[42] V. D. Park and M. S. Corson, "A highly adaptive distributed routing algorithm for mobile wireless networks," in *Proceedings of IEEE INFOCOM*, pp. 1405–1413, Apr. 1997.

[43] Y. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proceedings of ACM MOBICOM*, pp. 66–75, Oct. 1998.

[44] C. K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Communications*, vol. 4, pp. 1–36, Mar. 1997.

[45] R. Dube, C. D. Rais, K. Wang, and S. Tripathi, "Signal stability-based adaptive routing for ad hoc mobile networks," *IEEE Personal Communications Magazine*, pp. 36–45, Feb. 1997.

[46] W. Su and M. Gerla, "IPv6 flow handoff in ad hoc wireless networks using mobility prediction," in *Proceedings of IEEE GLOBECOM*, pp. 271–275, Dec. 1999.

[47] Z. J. Haas, "The routing algorithm for the reconfigurable wireless networks," in *Proceedings of ICUPC*, vol. 2, pp. 562–566, Oct. 1997.

[48] Z. J. Haas and M. Perlman, "The performance of query control schemes for the zone routing protocol," in *Proceedings of ACM SIGCOMM98*, Oct. 1998.

[49] M. Joa-Ng and I. T. Lu, "A peer-to-peer zone-based two-level link state routing for mobile ad hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1415–1425, Aug. 1999.

[50] M.-G. Lee and S. Lee, "A pseudo-distance routing(PDR) algorithm for mobile ad-hoc networks," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, vol. E89-A, pp. 1647–1656, Jun. 2006.

[51] A. B. BcDonald, "A mobility-based framework for adaptive dynamic cluster-based hybrid routing in wireless ad-hoc networks." Ph.D. Dissertation proposal, University of Pittsburgh, 1999.

[52] M. S. Corson and V. Park, "An internet MANET encapsulation protocol (IMEP)." http://tools.ietf.org/html/draft-ietf-manet-imep-spec-00.txt, 1998.

[53] VINT Group. http://www.isi.edu/nsnam/ns.

[54] I. Rubin and Y.-C. Liu, "Link stability models for QoS ad hoc routing algorithms," in *Proceedings of IEEE VTC2003, Fall*, pp. 3084–3088, Oct. 2003.

[55] G. Lim, K. Shin, S. Lee, H. Yoon, and J. Ma, "Link stability and route lifetime in ad-hoc wireless networks," in *Proceedings of IEEE International Conference on Parallel Processing Workshops*, pp. 116–123, Aug. 2002.

[56] M.-G. Lee and S. Lee, "A link stability model and stable routing for mobile ad-hoc networks," *IFIP Int'l Conf. on Embedded and Ubiquitous Computing(EUC2006), Lecture Notes in Computer Science(LNCS) 4096*, vol. 4096, pp. 904–913, Aug. 2006.

[57] B. S. Manoj, R. Ananthapadmanabha, and C. S. R. Murthy, "Link life based routing protocol for ad hoc wireless networks," in *Proceedings of IEEE Conference on Computer Communications*, pp. 573–576, Oct. 2001.

[58] I. Rubin and Y.-C. Liu, "Link stability models for QoS ad hoc routing algorithms," in *Proceedings of IEEE VTC2003 Fall*, pp. 3084–3088, Oct. 2003.

[59] H. T. Friis, "A note on a simple transmission formula," in *Proceedings of IRE*, vol. 34, 1946.

[60] T. S. Rappaport, *Wireless communications, principles and practice*. Prentice Hall, 1996.

[61] K. Fall and K. Varadhan, "The ns Manual(formerly ns Notes and Documentation)." http://www.isi.edu/nsnam/ns/ns-documentation.html, 2005.

요 약 문

이 논문은 이동형 애드-혹 네트워크에서 경로를 탐색하는 방법과 링크의 안정성을 예측 하는 모델을 제안한다. 뿐만 아니라 제안된 링크 안정성 예측 모델을 이용하여 안정 경로를 탐색할 수 있는 방법을 제안한다.

첫 주제는, 이동형 애드-혹 네트워크(MANETs)에서의 경로를 탐색하는 방법에 대하여 논한다. 기존의 경로 탐색 방법들은 최단경로를 찾는 것을 목표로 하였다. 그러나MANETs에서는 비 신뢰적이고 동적으로 변화하는 네트워크의 특성상 기존에 탐색되어진 경로의 연결성이 끊어지기 쉽다. 이런 문제를 극복하기 위해서 동적인 경로 탐색 방법들이 연구되어 왔지만 경로가 단절될 때마다 새로운 경로를 다시 찾는 것은 그 부하가 매우 크다. 다른 방법으로는 경로의 변화가 발견될때마다 새로 경로를 수정하는 형태의 경로 탐색 방법이다. 이 논문에서는 의사-거리 경로 탐색 방법이라는 효율적으로 경로 탐색표를 관리하고 동적으로 네트워크의 변화에 적응한 넝로 탐색 방법을 제시한다.

두번째 주제는안정적으로 패킷을 전달할 수 있는 경로 선택 방법에 대하여 논한다. 최단 경로 탐색 방법들에 의해서 찾아진 경로는 그 길이는 짧을 수 있으나 각 노드와 노드 사이의 거리는 매우 길어지게 되고, 이는 각각의 노드의 통신 범위에 다다르게 된다. 그러나 두 노드 사이의 물리적 거리가 멀어지게되면 무선 네트워크의 특성상 패킷 분실확률이 급격히 증가하게 되고, 이는 경로를 불안정하게 만든다. 더우기 수신 신호의 강도가 충분히 높지 않다면, 백색 잡읍이나 상호 간섭등의 외적 요인에 의해서 수신된 패킷을 잃어버릴 수도 있다. 그러므로 바람직한 경로탐색 방법은 안정적인 경로를 탐색해서 패킷의 분실 확률을 낮출 수 있어야 한다. 이러한 목적을 달성하기 위해서 본 논문에서는 수신 신호의 강도를 이용하여 안정성을 판단하는 발전된 안정성 예측 모델을 제안하고 제안된 방법을 통해서 예측된 링크 안정성 정보를 이용하여 보다 안정된 경로를 찾을 수 있는 안정 경로 탐색 방법을 제안한다. 제안된 방법은 다른 방법들에 비해서 우수한 성능나타냄을 전산모사를 통해 보여주며, 특히 링크의 안정성이 떨어지는 경우 매우 효과적임을 보인다.

Curriculum Vitae

Personal Information

Name	:	Min-Gu Lee
Date of Birth	:	June 13th, 1977
Address	:	863-2, Sangdaewon 3-dong, Sungnam, Kyungki 462-123, Republic of Korea

Education

Feb. 2007 : Ph.D in Division of Electronic and Computer Engineering, POSTECH, Pohang, S. Korea
Thesis Title:
이동형 무선 애드 혹 네트워크를 위한 의사 거리 경로 탐색 방법 및 링크 안정성 예측 모델
(A Pseudo-Distance Routing Algorithm and Link Stability Estimation Model for Mobile Ad Hoc Networks)
Advisor: Prof. Sunggu Lee(이승구 교수님)

Feb. 2002 : M.S. in Division of Electronic and Computer Engineering, POSTECH, Pohang, S. Korea
Thesis Title:
TMO를 기반으로한 분산환경에서의 실시간 항공기 착륙 전산 모사 시스템의 구현
(Implementation of a TMO Based Real-Time Airplane Landing Simulator on a Distributed Computing Environment)
Advisor: Prof. Sunggu Lee(이승구 교수님)

Feb. 2000 : B.S. in Division of Electronics, Communication and Radio Engineering, Hanyang University, Seoul, S. Korea

Experience

Mar. 2004 - Aug. 2004	:	Visiting researcher, Univerisity of California, Irvine, Research on supporting real-time systems in mobile ad-hoc networks.
Mar. 2000 - Feb. 2007	:	Teaching and research assistant, POSTECH. in Dept. Electronic and Electrical Engineering
Jun. 1998 - Jun. 1999	:	President of WEBTeam, Hanyang University, Developed geographic information system for Hanyang University and other websites

Attended Projects

1. "Development of ASICs for LCD Image Quality Improvement Algorithms," Samsung Electronics, May 2005 - Feb. 2007
2. "Research on Providing Support for Distributed Real-Time Applications in Mobile Ad-Hoc Networks," Korea Research Foundation, Dec. 2004 - Nov. 2005
3. "Development of a High-Speed Wireless Network Interface Supporting USB 2.0," POSDATA Inc., Jan. 2002 - Dec. 2002
4. "Development of an EML Interface Channel," KAIST (funded by Ministry of Science and Technology (MOST)), Jun.2001 - Jun. 2002
5. "Research on an Object-Oriented Real-Time Distributed Simulation System," Korea Research Foundation, Sep. 2000 - Aug. 2002
6. "Development of a Scalable Web Server" NeoMain Inc., May 2000 - Nov. 2000

Publications

Journals

1. Min-Gu Lee and Sunggu Lee, "A Pseudo-Distance Routing(PDR) Algorithm for Mobile Ad-hoc Networks," *IEICE TRANSACTIONS on Fundamentals of Electronics, Communications and Computer Sciences*, Vol.E89-A, No.6, pp.1647–1656, Jun. 2006.

2. Min-Gu Lee, Sunggu Lee and K. H. Kim, "Implementation of a TMO-structured real-time airplane landing simulator on a distributed computing environment," *Software: Practice and Experience*, Wiley InterScience, Vol.34, No.15, pp.1441–1462, Nov. 2004.

Lecture Notes in Computer Science Series

1. Min-Gu Lee and Sunggu Lee, "A Link Stability Model and Stable Routing for Mobile Ad-hoc Networks," *IFIP Int'l Conf. on Embedded and Ubiquitous Computing(EUC2006), Lecture Notes in Computer Science(LNCS)* Vol.4096, Springer, pp.904–913, Aug. 2006.

Conference Proceedings

1. Min-Gu Lee and Sunggu Lee, "QoS Support for Mobile Ad-Hoc Networks Based on a Reservation Pool," *The 9th IEEE International Symposium on Object and Component-Oriented Real-time Distributed Computing*, April 24-26, Gyeongju, Korea, 2006

2. Min-Gu Lee and Sunggu Lee "A Pseudo-Distance Routing(PDR) Algorithm for Mobile Ad-hoc Networks," *The 20th International Technical Conference*

On Circuits/Systems, Computers and Communications, July 4-7, Jeju, Korea, 2005

3. Min-Gu Lee and Sunggu Lee "Delay Analysis for Statistical Real-Time Channels in Mobile Ad-Hoc Networks," *The 9th IEEE Workshop on Object-oriented Real-time Dependable Systems*, Feb. 2-4, Sedona, AZ, USA, 2005

4. Min-Gu Lee and Sunggu Lee "Implementation of a TMO-Based Real-Time Airplane Landing Simulator on a Distributed Computing Environment," *The 7th IEEE International Workshop on Object-oriented Real-time Dependable Systems*, Jan. 7-9, San Diego, CA, USA, 2002

Domestic Journals

1. Ju-Ho Hyun, Sunggu Lee, Sang Cheol Kim and Min-Gu Lee "An Efficient Scheduling Method Taking into Account Resource Usage Pattern on Desktop Grids," in *Journal of KISS: Computer Systems and Theory*, Vol. 33, No. 7, Jul. 2006.

Domestic Conference Proceedings

1. Ju-Ho Hyun, Sunggu Lee, Sang Cheol Kim and Min-Gu Lee "An Efficient Scheduling Method Taking into Account Resource Usage Pattern on Desktop Grids," in *Proceedings of KISS Special Interest Group on High Perofrmance Computing*, Feb. 2006.

Patents

1. "Method for routing of the mobile ad-hoc network(PENDING)", No.: 10-2006-0078075, KR, Aug. 18, 2006.