ENHANCING THE PERFORMANCE OF ACTIVE CONNECTIONS IN MANETS THROUGH DYNAMIC ROUTE AND POWER OPTIMIZATION

by

ZEKI BILGIN

A dissertation submitted to the Graduate Faculty in Computer Science
in partial fulfillment of the requirements for the degree of
Doctor of Philosophy, The City University of New York

2010

This manuscript has been read and accepted by the Graduate Faculty in Computer Science in satisfaction of the dissertation requirement for the degree of Doctor of Philosophy.

	(Bilal Khan)
Date	Chair of Examining Committee
	(Theodore Brown)
Date	Executive Officer

Bilal Khan

Nancy Griffeth

Shamik Sengupta

Ala Al-Fuqaha

Supervisory Committee

THE CITY UNIVERSITY OF NEW YORK

Abstract

ENHANCING THE PERFORMANCE
OF ACTIVE CONNECTIONS IN MANETS
THROUGH DYNAMIC ROUTE AND POWER OPTIMIZATION

by

ZEKI BILGIN

Advisor: Bilal Khan

In this thesis, we consider two significant problems that occur within active connections in mobile ad hoc networks (MANETs). These are: (A) degradation of path optimality in terms of hop count, and (B) failures on the constituents links of a path. Both phenomena occur over time because of node movement. Our investigation considers what can be done to minimize their occurrence of both, *after the problem of initial route selection has been resolved by standard MANET routing protocols.*

In developing solutions to the aforementioned problems, we identified two broad and complementary approaches:

(i) **Variable topology, fixed power:** These approaches assume that the transmission power of the nodes is kept fixed, but the topology of the connections is modifiable during their lifetimes.

(ii) **Variable power, fixed topology:** These approaches assume that the topological structure of the connection must be kept fixed, but the transmission power levels used by constituent nodes is adjustable.

Within approach (i), we developed (A) two new route optimization schemes that seek to shorten path lengths by eliminating inessential hops "on-the-fly", without relying on promiscuous mode of wireless cards, and (B) two new route maintenance schemes that circumvent impending link failures and heal broken links in an efficient way. We implemented our schemes in the ns2 packet level network simulator, as extension to the Ad hoc On Demand Distance Vector (AODV) routing protocol. Through extensive simulations, we show that our schemes are able to optimize path lengths, increase connection lifetime, reduce overall control traffic overhead, decrease end-to-end delay, and provide energy savings in packet transmissions.

Within approach (ii), we developed (B) several new dynamic power budget distribution schemes. These were evaluated using a new model in which each connection is assigned a fixed power budget, and seeks to distribute this budget among its constituent nodes so as to increase the connection's lifetime. We implemented our schemes as a discrete event simulation. Through extensive simulation experiments, we showed that our schemes are able to consistently improve connection lifetimes without excessive additional control traffic overhead.

The conclusions of both studies are seen to hold scalably as one varies situational parameters such as network size, number of connections, and node mobility levels.

ACKNOWLEDGMENTS

Many people helped me to accomplish this work. They deserve my appreciation and deep thanks. First of all, I would like to express my gratitude to my supervisor, Prof. Bilal Khan who was abundantly helpful and offered invaluable assistance, support and guidance. Deepest gratitude are also due to the members of the supervisory committee, Prof. Nancy Griffeth, Prof. Shamik Sengupta and Prof. Ala Al-Fuqaha without whose knowledge and assistance this study would not have been successful. I would also like to thank the Executive Director Prof. Ted Brown for his support and assistance.

Special thanks also to all my graduate friends, especially members of Turkish National Police (TNP); Dr. Omer Demir, Dr. Ayhan Akbulut, Hakan Aydogan, Koray Kaleli, and Seyfullah Bucak for their company and invaluable assistance.

I would like to convey thanks to Turkish National Police for providing the financial support and opportunity of having a graduate education abroad. I would also like to thank The Graduate Center, and John Jay College of Criminal Justice for providing facilities during my education.

I would like to express my love and gratitude to my beloved family; for their understanding and endless love, through the duration of my studies.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

CHAPTER 1

INTRODUCTION

1.1 Mobile Ad Hoc Networks

Ad hoc wireless networks have attracted increasing interest in recent decades, as they have found many military and commercial applications. A mobile ad hoc network (MANET) is a special type of wireless ad hoc network consisting of a group of *mobile* devices–hereafter referred to simply as "nodes". These nodes are collectively capable of forming a wireless network, without the need for any wired or wireless infrastructure. Thus, a MANET can be regarded as an autonomous system of mobile nodes, which may (or may not) have gateways to the Internet, and/or to other fixed networks [17, 29, 37]. Since MANETs do not require a communication infrastructure to form a network among wireless devices, they can provide a rapidly deployable communication environment for rescue teams, military operations, space exploration, and other settings in which no existing telecommunication infrastructure is available. A most commonplace and simple example of a MANET arises when laptops which include wireless access cards are configured to operate in "Ad hoc mode".

MANETs are a natural historical descendant of earlier base-station based wireless networks. In this earlier form of network, wireless communications took place between a mobile node and a base station, while base-station interconnectivity was achieved

using a fixed wired/wireless infrastructure. Modern cellular telephone networks continue to operate within this framework. Dealing with the mobility in these types of wireless networks is easy (by comparison), since it requires efficiently maintaining a distributed directory of mobile user locations[1].

In contrast, in a wireless ad hoc network, connections between nodes are constructed in multihop fashion, with each node acting as a router. Nodes participating in such data forwarding between a source and destination node are said to define a "path" or "route". The main challenge of "routing" in these settings is to ensure that nodes are able to select suitable routes to forward data traffic to the intended destinations. In a MANET environment, because each node is able to move independently in any physically feasible direction, the topology[2] of the network is subject to continuous change. In the face of dynamic topological changes due to node movement, routing becomes difficult, and so has been the subject of significant research efforts within the wireless networking community. We review some of the prior work regarding this topic, from which the present work draws its research questions.

1.2 Routing in MANETs

The main function of routing is to find an initial path between a source and destination node, and to subsequently maintain data forwarding between the two nodes throughout the entire duration[3] of the connection. Routing has been the subject of

1. See, e.g. the 1995 results on sparse partitions by Awerbuch and Peleg [9].

2. The set of communication links between node pairs that *could* be used to transmit data [44].

3. The word *lifetime* can have two possible interpretations: (i) until one of the endpoints determines that it does not with to be party to the connection any longer; (ii) until one of the links in the connection breaks because of node mobility. We will use the word **duration**

extensive research, in both static settings (where nodes are fixed) and dynamic settings (where they are mobile). The existing literature presents a plethora of routing protocols that have been developed by researchers, since the area has been particularly active during the past two decades (1990-2010). Each of routing protocols developed is suited to particular settings and has its own merits and advantages. The routing protocols are commonly classified as either proactive, reactive, or hybrid (a combination of the two types), depending on how and when routing information is obtained and maintained by nodes.

Proactive routing protocols include protocols such as Optimized Link State Routing Protocol (OLSR) and Destination-Sequenced Distance-Vector Routing (DSDV) follow in the footsteps of wired infrastructure routing protocols like Open Shortest Path First (OSPF). In such protocols, nodes maintain next-hop routing information, regardless of whether the routes are required by connection requests. This route information is updated whenever a change in the underlying network topology occurs. In settings where nodes move frequently and connections are long-lived, proactive routing protocols may be inefficient due to high levels of routing-related control traffic generated in response to node movements that induce topological changes in the network structure.

Reactive routing protocols include protocols such as Ad hoc On Demand Distance Vector (AODV) and Dynamic Source Routing (DSR). These protocols discover a route only when it is required[4]. It is a common knowledge that reactive routing protocols are more appropriate for mobile ad hoc networks, especially if node mobility levels

when we intend first, less technical definition, and the word **lifetime** only in the sense of definition (ii).

4. Discovered routes may be cached for a brief interval of time even after they cease to be used for data forwarding, as a performance optimization.

are high, connections interarrival times are large, or connection durations are long. In these settings reactive routing protocols delay incurring systemic costs in response to topology changes until routes are actually requested. As a consequence, in settings of highly mobile nodes with interconnections of long duration, reactive routing protocols incur lower control traffic overhead than their proactive counterparts.

Regardless of whether the approach is reactive or proactive, the following is a partial list of some of the desired characteristics of a routing protocol [37]:

- *Minimum route acquisition delay:* The time required to establish a route from a source node to a particular destination should be short. This metric tends to be better for proactive routing protocols than for reactive routing protocols, since protocols belonging to former group keep routing information available regardless of whether there is any connection request between nodes, while reactive protocols establish routes upon request.

- *Quick route reconfiguration:* Dynamic changes in network geometry and topology due to unpredictable node movements require the routing protocol to be responsive in reconfiguring routes to circumvent link breaks, packet loss, and route suboptimality which may emerge over time along active paths.

- *Loop-free routing:* This is essential to provide timely data delivery, and to avoid unnecessary waste of network bandwidth and buffer resources. Alternately, a routing protocol may choose to take the approach of detecting and addressing any routing loops which may arise.

- *Distributed routing approach:* This is required because ad hoc networks are decentralized in nature.

- *Minimum control traffic overhead:* The number of control packets exchanged in the process of establishing and maintaining routes should be as small as possible. This is desired because control packets: (i) consume valuable network resources such as bandwidth, power, and buffer space; (ii) increase channel contention at link layer due to the shared transmission medium; (iii) can collide with data packets, thereby reducing network throughput.

- *Scalability:* We would like our protocol to perform efficiently in families of networks whose parametric descriptions reflects anticipated trends. For example, we anticipate having networks with greater number of nodes, higher node density, and higher levels of mobility. The extend to which the performance of a routing algorithm responds to such trends is a predictor of its ability to avoid obsolescence in the face of expected trends.

- *Provisioning of Quality of Service:* A routing protocol may be required to provide a level of quality of service (QoS) for connections. QoS might be specified in terms of parameters such as throughput, delay, jitter, packet delivery ratio, etc.

- *Support for time-sensitive traffic:* Some applications such as tactical communications require support for time-sensitive traffic. For this type of application, the routing protocol should be able to support real-time guarantees.

- *Security and privacy:* Depending on application, a certain level security could be required from routing protocol. The requirements could include resilience to denial-of-service, impersonation, resource consumption and other attacks.

1.3 Reactive routing and AODV

Although most reactive routing protocols have similar methods for triggering route discovery process, they exhibit minor differences in the specifics of how they maintain routing information. For example, AODV keeps only the next hop for each requested destination at each node, and does not embed any extra information into packet headers. In contrast, DSR uses a source routing strategy, inserting the entire transit list into its packet headers. In this research, for concreteness, we will focus on the AODV. However, the solutions we develop will be broadly adaptable to other reactive protocols.

In AODV, when one node (referred to as the source) requires a connection to another node (referred to as the destination), the former initiates a global route discovery operation. This is done by "flooding" a ROUTE_REQUEST (RREQ) message in the network[5]. When (at least) one of these messages is received by the intended destination, or by a node which has a "fresh enough" route to the destination, a ROUTE_REPLY (RREP) message is sent back to the originator of the ROUTE_REQUEST, reversing the route taken by the RREQ. As the ROUTE_REPLY packet travels back towards to the originator of the request, each node along the return route inserts next hop information into its routing table for the destination requested. Once the source node receives the ROUTE_REPLY message, it can start forwarding data towards the destination along the newly route established route. Further details of AODV's operation are described in the RFC [40].

5. Flooding is a way to distribute routing information updates quickly to every node in a large network. It is achieved essentially by performing a distributed breadth-first traversal of the network.

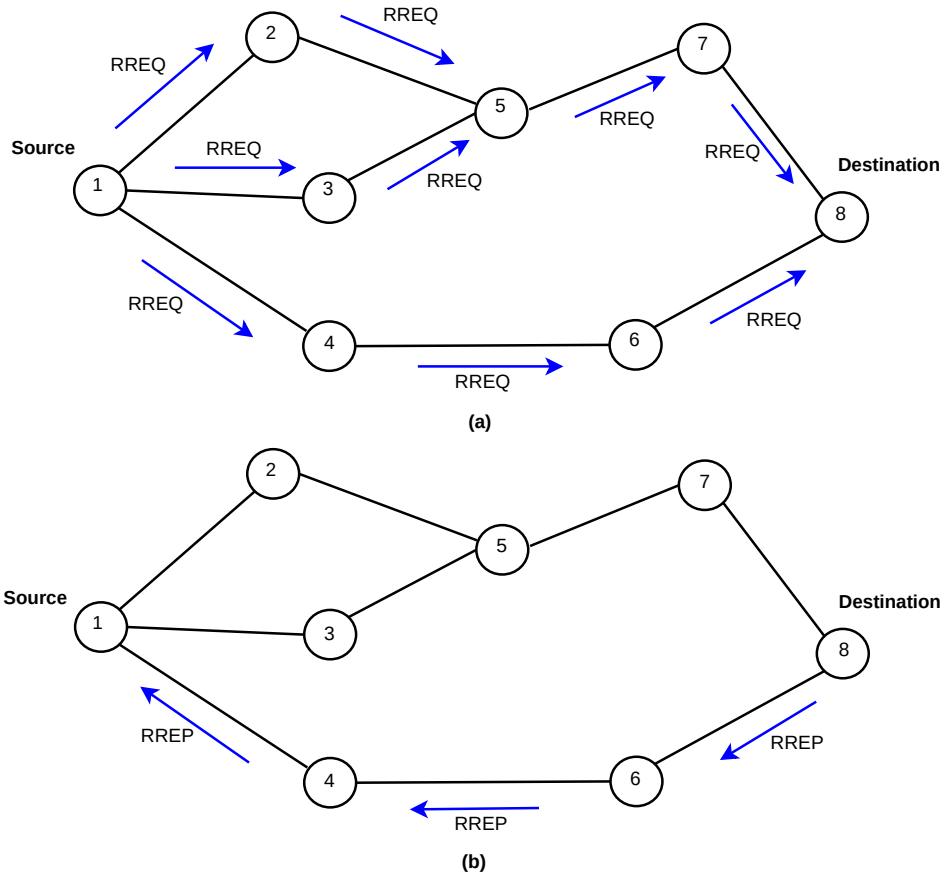


Figure 1.1: AODV route discovery: (a) path of RREQ, and (b) path of RREP.

In AODV, nodes mainting routing table entries as long as the route is active. A route is considered **active** as long as data was recently seen traveling along it, from the source to the destination. Once the source stops sending data packets, the routing table entries for each of the connection's constituent links will time out, get marked inactive, and be deleted. The routing table entries responsible for data forwarding could also be disrupted if the links themselves break due to node mobility. This can happen even while connection is active, i.e. while the source node is sending data packets. A link **breaks** or becomes **disconnected** when one endpoint of the link moves out of transmission range of the other endpoint. If one of the links along an active route breaks, a route repair is attempted provided the broken link is closer to

the source than to the destination. If the break is closer to the destination than the source, or if the attempted route repair fails, the upstream endpoint of the broken link propagates a ROUTE_ERROR (RERR) packet towards the source node, to inform it of the now-unreachable destination. After receiving the ROUTE_ERROR, if the source node still desires the route, it can reinitiate a global route discovery.

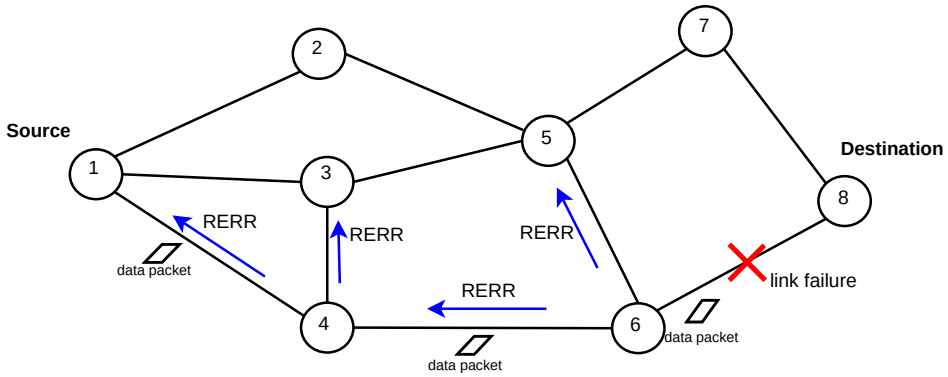


Figure 1.2: Link failure and RERR in AODV.

AODV uses destination sequence numbers to avoid the formation of routing loops, and to compare the freshness of routing table entries. Each node maintains a destination sequence number (DSN) for each entry (i.e. destination) in its routing table, and keeps this number updated as specified. A node updates the DSN whenever it gets new information about it via RREQ, RREP, or RERR packets related to the destination. A destination node is authoritative concerning its own sequence number, which is incremented in the following two circumstances [40]:

- Immediately before a node originates a route discovery, it increments its own sequence number. This prevents conflicts with previously established reverse routes towards the originator of a RREQ.
- Immediately before a destination node originates a RREP in response to a RREQ, it must update its own sequence number to be equal to larger of its

current sequence number and the destination sequence number specified in the RREQ packet.

1.3.1 AODV Routing Table Structure

According to AODV [40], each node u maintains a routing table $u.T$ which consist of a set of entries indexed by destination IP address. The entry corresponds to destination v is referred to as $u.T[v]$. The entry $u.T[v]$ consists of a set of fields:

- Destination Sequence Number, i.e. $u.T[v].dsn$,
- Valid Destination Sequence Number flag, i.e. $u.T[v].flag$,
- Other state and routing flags (e.g., valid, invalid, repairable, being repaired)
- Network Interface, i.e. $u.T[v].iface$,
- Hop Count (number of hops needed to reach destination), i.e. $u.T[v].hops$,
- Next Hop, i.e. $u.T[v].next$,
- List of Predecessors, i.e. $u.T[v].pred$
- Lifetime (expiration or deletion time of the route), i.e. $u.T[v].expiry$. Note that this parameter is updated continuously as long as the route is active, i.e. there is data flow going on the connection.

The above notation will be used in all formal analyses of our extensions to standard AODV, in Sections 2.10 and 3.9.

1.4 Research Questions

While designing a routing algorithm for MANETs, one of the greatest challenges is tuning the response of the algorithm to the movement of nodes, since mobility can yield rapid changes in network topology. In static networks, the problem of routing is relatively easy, because the problem needs to be solved infrequently. Once a route is established for a pair of source and destination nodes, it is likely to remain valid a long period of time because link failures and new links are rare events in static networks. This is not the case in MANETs, however, where nodes are assumed to move independently and unpredictably.

In MANETs, routing is much more difficult, since in the presence of mobility, node movements induce frequent network topology changes. These changes can occur because of:

(X1) The *formation* of new links, and

(X2) The *disconnection* of existing links

We are interested in two major problems introduced by mobility in mobile ad hoc networks which use reactive routing protocols. These are:

(Y1) Degradation of route optimality, and

(Y2) Link breakages.

As we shall see, X1 is the root cause of Y1, and X2 is the root cause of Y2. We will explain problems Y1 and Y2 in detail in Chapters 2 and 3, respectively. Here we

describe them only briefly, to give an overview of the main concerns.

Degradation of route optimality: Mobility may make connections suboptimal because new links may arise, making alternate shorter routes feasible. This is illustrated in Figure 1.3. Such situations commonly occur when reactive routing protocols are used, since in these protocols the topological structure of a route remains fixed as long as the connection is active and its constituent links are intact. This brings us to

QUESTION A:

In the face of node mobility, how can we dynamically maintain route optimality throughout the lifetime of the connection?

Link breakages: A link ceases to be operational when the autonomous movement of one of the link's endpoints causes it to fall out of transmission range of the other endpoint. This is illustrated in Figure 1.4. Disconnection of links generally entails extra work for routing protocols, since the routes of active connections must be repaired. More precisely, when a link within an active connection is broken, the routing protocol generally reinitiates route discovery operation, which usually requires flooding–an expensive operation in terms of control traffic incurred. This brings us to

QUESTION B:

How can we prevent or fix possible link breakages effectively?

In developing solutions to address the above two research questions, we consider two broad and complementary classes of approaches. These are

APPROACH (i):

Variable topology, Fixed power. These approaches assume that the transmission power of the nodes is kept fixed, while the topology of the connection is modifiable.How can we prevent or fix possible link breakages effectively?

and

APPROACH (ii):

Variable power, Fixed topology: These approaches assume that the topological structure of the connection is kept fixed while the transmission power of the constituent nodes is adjustable.

1.5 Organization of the Thesis

in the previous section, we introduced the two research questions that are of interest to us, and the two classes of solution approaches we will consider. The three concrete investigations we conduct are:

1. **Dynamic Route Optimization:** Dynamic optimization of connection performance metrics, particularly reducing path stretch without excessive control traffic. This is Question A via approach (i). It is the subject of Chapter 2.

2. **Dynamic Route Maintenance:** Circumvention of connection failures caused by link breakages. This is Question B under approach (i). It is the subject of Chapter 3.

3. **Dynamic Power Distribution:** Maximization of connection lifetime through dynamic distribution of transmission power. This is Question B under approach (ii). It is subject of Chapter 4.

Next, in Chapters 2, 3 and 4, we describe the above three investigations, respectively. Within each chapter we shall:

- Give a precise statement of the problem, and its significance;
- Review related prior works, their merits and shortcomings;
- Describe desirable characteristics of a new solution;
- Explain our proposed solutions;
- Give a methodology for evaluating the proposed solutions; and
- Present results of experiments, and discuss the implications.

In addition, in Chapters 2 and 3, we provide some theoretical results demonstrating that the proposed methods are both sound and robust. Finally, in Chapter 5, we provide conclusions synthesizing our results for each problem and corresponding proposed solution methods. We also describe our ongoing and future research work.

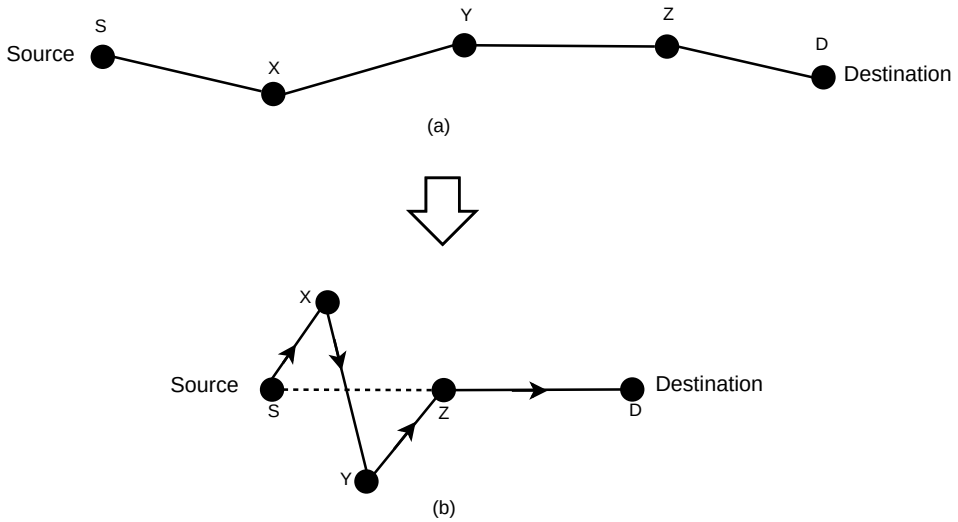


Figure 1.3: (a) Initial route, and (b) how suboptimality arises.

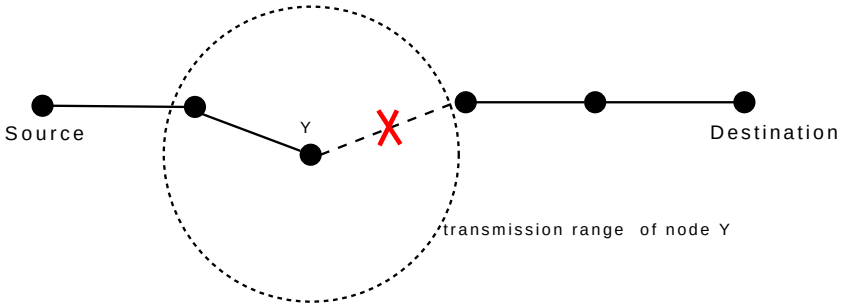


Figure 1.4: An example of link breakage on a connection.

CHAPTER 2

DYNAMIC ROUTE OPTIMIZATION

In this chapter, we consider the problem of *dynamic route optimization*, which is to say, the minimization of connection path lengths (as measured in hops) over the duration[1] of a connection. Our investigation takes place in the context of mobile ad hoc networks (MANETs). Within these settings, we consider how and why the routes of active connections can diverge from optimal, and we seek to develop schemes which can detect the suboptimality of a route, and rectify it. More precisely, the schemes will shorten routes "on-the-fly" by surgically eliminating inessential hops from the path. We will seek to do all this, of course, in a decentralized and scalable manner, without incurring unacceptable additional control traffic, and without sacrificing existing routing guarantees.

To achieve the aforementioned objectives, in this chapter, we present two new route optimization schemes, as extensions to established reactive routing protocols. We name these schemes (i) **1-hop Shrink**, and (ii) **Multihop Shrinking**. The two schemes are designed as an extension to the Ad hoc On Demand Distance Vector (AODV) routing protocol [40]. The schemes are implemented in ns2 [3], a discrete event and packet level network simulator frequently used to conduct large scale simulation studies. Through extensive simulations, we show that the proposed schemes

1. See footnote 3 of Chapter 1 (pp. 2) for the distinction between the terms lifetime and duration.

provide notable improvements in route optimality, as well as the following enhancements in network performance:

i. Increased connection lifetime;

ii. Improved packet delivery ratio, i.e. higher values for the ratio of packets received at destinations to packets sent by the sources;

iii. Reduced average end-to-end delay for delivered packets;

iv. Decreased average energy required per delivered packet; and

v. Reduced average control traffic overhead per delivered packet.

We also show, by formal analysis, that the proposed extensions are consistent with the underlying AODV protocol, in the sense that the extensions do not invalidate AODV's loop-free routing guarantee. Although we have specifically used AODV as a starting point in this research, the proposed schemes can readily be adapted to other reactive routing protocol such as Dynamic Source Routing (DSR) [25], Signal Stability based Adaptive Routing (SSA) [20], etc., since we have made no strong AODV-specific assumptions in their design.

The rest of the chapter is organized as follows. In Section 2.1, we give an informal description of the problem, and describe the core challenges. Then, in Section 2.2, we describe the significance of the problem, and the potential benefits obtained by its resolution. In Section 2.3, we provide a formal definition of the problem. Then, in Section 2.4, we provide an overview of related prior work. its merits, and its limitations. Our constraints and objectives in developing the schemes are described in Section 2.5; system assumptions are outlined in Section 2.7. We present our new

schemes in detail in Section 2.8. The performance analysis of our new schemes, together with the description of simulation environment and experimental setup are given in Section 2.9. Finally, in Section 2.10, we formally analyze the extent to which our schemes operate in robust, loop-free manner.

2.1 Informal Statement of the Problem

In reactive routing protocols, connection routes between source-destination pairs are constructed on-demand, at the outset, upon receipt of a request from the source node. Once the reactive routing protocol has constructed a route from the source to the destination *the route remains in use as long as the connection is active.* Although node mobility can greatly change path geometry[2], data packet transmissions along the route is interpreted as an implicit confirmation of the route's existing topology[3]. Restructuring a route's topology has historically been considered infeasible while data is in transit along the connection. This adherence to a static topological structure over long periods of time, even in the face of radical changes in route geometry, introduces serious inefficiencies. When a connection is first established, the number of hops that the route takes tends to be close to optimal–the least number of hops, as calculated by, say, Dijkstra's shortest path algorithm. As time passes, however, nodes move, and the connection, retaining its combinatorial structure, tends to utilize many more hops than the instantaneous optimal (min-hop) path between source and destination. Since end-to-end latency is strongly correlated with node queuing/processing, this has undesirable consequences on connection performance. An example of this type of scenario is illustrated in Figure 2.1.

2. The positions of the sequence of nodes along a connection *in space.*
3. The combinatorial sequence of nodes along the route.

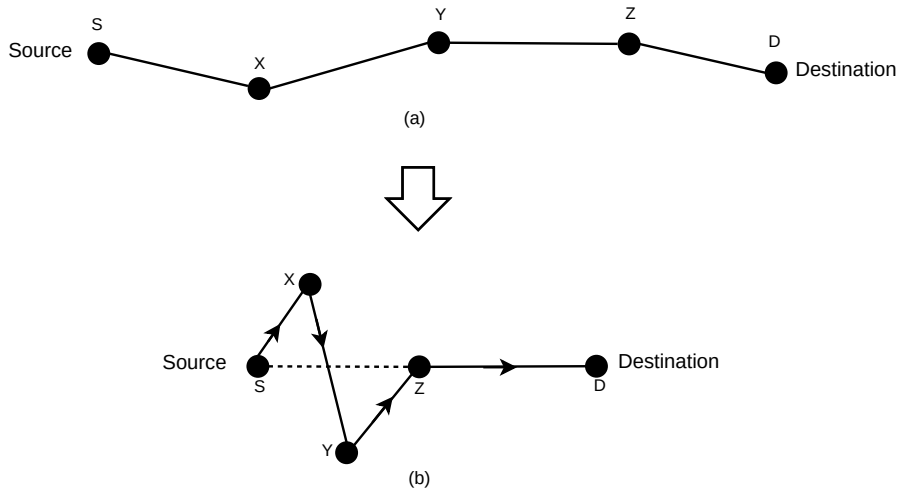
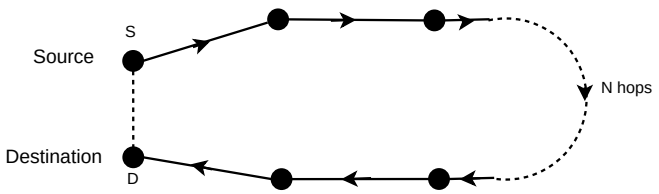


Figure 2.1: (a) Initially optimal route; (b) how mobility causes suboptimality.

In the top of Figure 2.1, we see that initially node S establishes an optimal 4-hop connection to node D. Over time, although node geometry changes, route topology remains fixed (as long as the connection is active). Thus, as seen in bottom of Figure 2.1, even when a 2-hop alternative route becomes feasible, connecting nodes S and D through Z, the data packets continue to follow the initially constructed (now suboptimal) 4-hop path. Depending on the movement pattern of the nodes, even more severe suboptimalities may arise. A worst-case example scenario is illustrated below, where we see a 1-hop alternative (i.e. direct connection) between *Source* and *Destination* nodes, even though the routing algorithm is still using the initially constructed N-hop route. Such events are likely to occur in dense mobile ad hoc networks due to the dynamic nature of the network topology. The aforementioned problem of path degradation is prominent in all reactive MANET routing protocols.



2.2 Significance of the Problem

What are the advantages of optimizing route lengths in a wireless network? What are the disadvantages of using unnecessarily long routes? In this section, we examine the impact of route optimization on both connection and network performance metrics. Because many performance metrics are related, an improvement in one frequently leads to a cascade of side effects on other metrics. Below, we consider six metrics which would be ameliorated through dynamic route optimization. The first three are connection-level improvements; the last three are aggregate network-level improvements.

1. **Decreased delay in packet delivery:** Although there are many causes of end-to-end delay in packet delivery, end-to-end packet delays are directly proportional to path length [56]. Thus, the delay expected for shorter routes is lower than it is for longer routes[4]. This claim is confirmed experimentally by Wu et al. [56].

2. **Reduced energy consumption:** The energy consumption of wireless communications is closely related to the number of packet transmissions. To deliver a data packet from source node to destination node, it is necessary to retransmit the packet at each hop along the route. The more hops on the route, the more transmissions are required to deliver a packet. Therefore, route optimization can reduce the amount of energy consumed to deliver a packet to its destination.

3. **Increased connection lifetime:** Link failures on active routes may cause packet losses and extra control traffic overhead during recovery. The relationship

4. This effect is glaringly obvious when there is a shorter path which uses a *subset* of the longer path's constituent nodes.

between the connection lifetime and route optimality has been the subject of considerable study [16, 31, 34, 53, 57], and findings indicate that shorter routes are more robust in the face of node mobility. The expected lifetime of shorter routes is higher than it is for longer routes, or put another way, longer routes are more sensitive to node movements. This is to be expected since the probability of experiencing link degradation (and in the worst-case, failure) somewhere along a path, is directly proportional to the number of links (i.e. hops) on the path. It follows that the more hops there are on a path, the more likely connection degradation (or disconnection) is to occur.

4. **Reduced contention at link layer:** Contention is a collision between between the transmission of two nodes sharing a common channel for data communication. Nodes in a MANET generally use the same link layer channel to transmit packets. They must, therefore, compete with each other to acquire the channel. The greater the number of nodes trying acquire use of the common channel, the greater the level of contention. When the number of transmissions being carried out in the network is reduced (as is achieved by dynamic route optimization), it can be expected to yield less contention at the link layer.

5. **Increased packet delivery ratio:** Packet Delivery Fraction (PDF) is the percentage of packets sent by sources that are successfully delivered to their destinations. The PDF of a connection depends on several factors, such as contention at the link layer, link quality, etc. Thus, when the contention is lower, packet delivery ratio is improved, and dynamic route optimization can help to improve PDF in a network.

6. **Reducing routing control traffic:** Dynamic route optimization can help to reduce control traffic overhead incurred by the routing protocol. Since con-

nection lifetimes are increased, fewer route discovery attempts are made. The latter is an expensive component of routing related control traffic.

2.3 Formal Statement of the Problem

In this section, we introduce some formal notations that will be used to refer to aspects of the MANET's state over time. Let V be a set of MANET nodes. According to AODV [40], each node $u \in V$ maintains a routing table. The routing table of node u at time t will be denoted as $u(t).T$. This table consists of a set of entries indexed by destination IP address. The entry which corresponds to destination v is referred to as $u(t).T[v]$. The entry $u(t).T[v]$ consists of a set of fields:

- Destination Sequence Number, denoted $u(t).T[v].dsn$,
- Valid Destination Sequence Number flag, denoted $u(t).T[v].flag$,
- Other state and routing flags (e.g., valid, invalid, repairable, being repaired)
- Network Interface, denoted $u(t).T[v].iface$,
- Hop Count (number of hops needed to reach destination), denoted $u(t).T[v].hops$,
- Next Hop, denoted $u(t).T[v].next$,
- List of Predecessors, denoted $u(t).T[v].pred$
- Lifetime (expiration or deletion time of the route), i.e. $u(t).T[v].expiry$. This lifetime parameter is updated continuously as long as the route is active, denoted there is data flow going on the connection.

Let $\ell : V \times \Re \rightarrow \Re^2$ be a function which describes the 2-dimensional position of nodes V over time. Thus $\ell(u,t) \in \Re^2$ is the position of node u at time t.

Each MANET node u is assumed to have a constant transmission radius r_{tx}, which takes into account fixed transmission power, node receiver sensitivity, and background noise. This defines a dynamic edge set

$$E : \Re \rightarrow 2^{V \times V},$$

where

$$(u,v) \in E(t) \Leftrightarrow \| \ \ell(u,t) - \ell(v,t) \ \|_2 \leq r_{tx}.$$

The graph $G(t) = (V, E(t))$ is taken to be the link layer network at time t.

2.3.1 Performance measures

Definition 1. *Let $S(s,d)$ represent the set of data packets originated by node s and destined to node d. Let $R_A(s,d) \subseteq S(s,d)$ be the subset which actually arrive using routing algorithm A. For each $p \in R_A(s,d)$, we have a time $t_A(p)$, at which the packet p was delivered to d.*

Definition 2. *For each $p \in R_A(s,d)$, let $d_{OPT.A}(p)$ be the length in hops of the shortest path from node s to node d in $G(t_A(p))$.*

Definition 3. *For each $p \in R_A(s,d)$, the value $d_A(p)$ represents the number of hops the packet p travels to reach node d, starting from node s, in a network operating according to routing protocol A.*

Definition 4. *Given a network routing protocol A, we define*

$$stretch(A) = \frac{1}{\sum_{s,d \in V} |R_A(s,d)|} \sum_{s,d \in V} \sum_{p \in R_A(s,d)} \frac{d_A(p)}{d_{OPT.A}(p)}.$$

and

$$PDF(A) = \frac{|\cup_{s,d \in V} R_A(s,d)|}{|\cup_{s,d \in V} S(s,d)|}.$$

This is the quantity $stretch(A)$ is what we seek to evaluate and minimize. Of course, the operation of any given scheme A, will in general introduce its own operational costs, e.g. control traffic as measured in packets sent that were not part of $\bigcup_{s,d \in V} R(s,d)$. We will seek to quantify and minimize these secondary costs. In addition, we will consider the extent to which the scheme is affected by the choice of movement patterns ℓ and node density, reflected in the parameter r_{tx}.

2.4 Related Work

Although a few researchers have considered the problem of dynamic route optimization in the context of reactive routing protocols in MANETs, the problem has not received widespread attention (a recent 2009 paper by Jia et al. [24] corroborates our impressions on this matter). In this section, we review related prior research on this problem. These approaches can be classified into two groups: (i) Those based on **Promiscuous mode**, and (ii) **Special-packet based approaches**.

2.4.1 Promiscuous mode based approaches

The methods in this group leverage the ability of wireless ad hoc network nodes to be put into *promiscuous mode*. Once placed in promiscuous mode, a node receives and processes *all* packets transmitted by other nodes within range, even when it is not the intended recipient. This is possible, because wireless transmission is broadcast, or shared medium. In promiscuous mode, all packets a node hears are accepted by the MAC layer and passed to higher layers. The major drawback of using promiscuous mode high power consumption, since the node's central processing unit is busy processing every packet received (regardless of whether the packet is really intended for the node or not). As a consequence, operation in promiscuous mode tends to degrade the (power-related) lifetime of the MANET's constituent devices.

Seminal work in this field is that of Wu et al. [56], in 2000, which considers several dynamic route optimization schemes for the routing protocols DSR, SSA, AODV, and ZRP. In their work, the authors exploit the promiscuous receive mode of wireless devices to collect fresh routing information. We will summarize their route optimization scheme for AODV as follows: When a node B transmits a data packet to its next hop C on the route towards the destination D, all other nodes within 1-hop neighborhood, being in promiscuous mode, also receive this packet. Nodes receiving the data packet retrieve required information from the packet header, including hop count to destination, sequence number, source and destination addresses. If a node receiving this data packet has a shorter fresh route for the same destination D, then it informs node B about the shorter route by sending a ROUTE_REPLY packet including the related information. Upon receiving such a ROUTE_REPLY packet, node B modifies its routing table accordingly, thereby resulting in a modification of the path structure.

Building on the work of Wu et al [56], other studies [23, 22, 58] exploit the promiscuous mode of the wireless cards to search for possible shortcuts along a route. One exception to this is the research of Gui and Mohapatra [23], in 2003, who suggest an alternative way to get fresh routing information from the packets, by exploiting their MAC layer headers. More precisely, they propose to insert information required for path optimization into the MAC layer header of the packets, and enable the nodes to retrieve information from these headers during MAC layer interactions, *without utilizing promiscuous mode*. However, such a solution requires complex cross-layer design, as well as a modification of packet header formats.

Gui and Mohapatra [23] are also the first to introduce the term "(n,k) shortcut" (where $k < n$), to describe situations when an alternative k-hop path exists in place of an n-hop path between two nodes of interest. Given this terminology, the authors propose an algorithm, SHORT, which is able to detect $(n, 1)$ and $(n, 2)$ shortcuts between any pair of the nodes on the path. There are some special cases in which the SHORT algorithm is unable to detect possible shortcuts, including when the source or destination nodes are directly involved in the shortcut. The SHORT algorithm also requires extra memory to save collected information in order to make comparisons.

Giruka et al. [22], in 2004, present another promiscuous mode based path compression algorithm, named PCA. The authors stress that their algorithm resolves the problematic issues in the SHORT, including the inability to detect shortcuts which involve the source or destination nodes. The recent work of Yen et al. [58], 2010, contains a similar route optimization mechanism to the SHORT [23]. Unfortunately, the authors do not provide any experimental results regarding the effectiveness of the proposed path optimization mechanism.

2.4.2 Special-packet based approaches

In this second group of approaches, the nodes of the wireless network inform each other about the alternative routes through the transmission of special packets, thereby obviating the need rely on promiscuous mode operation. Although this is advantageous, the careful design of protocols becomes more important because of extra control traffic potentially incurred.

Saito et al. [47], in 2004, propose a proximity-based dynamic path shortening algorithm. By proximity, the authors mean a discrete notion of "nearness". They define the proximity area of a node as the area in which the received SNR (signal-to-noise ratio) value is more than a defined threshold. This proximity area is used to trigger optimization process as follows: Whenever two adjacent nodes enter each other's proximity area, it is taken as a sign that they are too close to each other, and a shortening process is initiated involving triplets of nodes. These triplets consist of the two proximate nodes, together with one additional node that is precisely one hop upstream or downstream from the pair. The objective of the algorithm is to shorten the two-hop path to 1-hop path by eliminating the inessential middle node, if possible. To do this, the two end nodes of the triplet send special packets to each other in order to check if there is direct link between them. If so, the node that is farthest upstream within the triplet modifies its routing table to cut out the middle node. The authors presented two versions of the protocol, one for DSR and one for AODV. In our own work, one of the schemes we developed (described later in this chapter) has some similarities with the AODV version of the approach taken by Saito et al. For example, we too consider shortening triplets (i.e. two-hop subpaths) into direct links (i.e. one-hop) by eliminating the middle node. However, our own work diverges

from Saito et al.'s in several aspects: (i) We amortize tests against data traffic, rather than using a proximity triggered approach, and (ii) we have resolved a deficiency in their implementation that prevents the destination node from being able to activate the optimization process. Lastly, (iii) our protocols operate end-to-end, rather that on triplets of nodes of interest.

Similarly, Zapata [59], in 2005, presents a shortcut detection mechanism for active routes that does not rely on the promiscuous mode of the devices. In his proposal, a special packet (Shortcut Request–SREQ) is broadcasted with TTL of 0 at each node along the route starting from the source node to the destination node. The SREQ packet includes the IP addresses of the source, the destination, and the next hop on the route, in addition to the distance in hops to both endpoints of the path. All 1-hop neighbors can receive a broadcasted SREQ packet; upon receiving such a packet, they check if there is a shortcut between them and the sender of SREQ packet by comparing the hop count metric for the same route (if defined) in their routing table. If a shortcut exits, the corresponding node sends back a SREP packet to inform the sender of the SREQ about the shortcut. Upon receiving a SREP packet, a node modifies its routing table so that the next hop for the corresponding destination is specified as the sender of SREP packet[5].

5. Regrettably, Zapata's scheme has not been tested by extensive experiments, as is evidenced by the following oversight in the protocol's design: A node may have a valid routing table entry for the destination and yet might no longer lie on the connection itself. If such a node responds to an SREQ by sending an SREP, it will cause the formation of an invalid route. One of the route optimization schemes we present in this chapter uses similar ideas, but our protocol does not fall prey to the aforementioned pitfall. Moreover, our own schemes are validated by experimental studies conducted using extensive packet-level simulations with ns2.

2.5 Design Objectives

In designing our solutions for dynamic route optimization, we keep the following objectives in mind:

(I) **The solution should optimize path length.** The primary objective of the optimization process should be elimination of inessential hops on the connection. Of course, there may be other benefits obtained implicitly such as decrease in end-to-end delay, power savings, reduction in the contention, increase in the lifetime of the connection, etc.

(II) **The solution should not rely on promiscuous receive mode.** In promiscuous receive mode, wireless cards are able to receive and process any packets they sense, regardless of packet destination. However, in this mode, equipment may consume excessive power because of having to continuously process received signals. It also may not have sufficient residual time to process and send their own packets. As a counterpoint to this objective, other researchers, including Wu et al. [56] propose several dynamic route optimization algorithms for routing protocols (DSR, SSA, AODV, and ZRP) in which they assume the use of promiscuous receive mode within wireless nodes.

(III) **The solution should not be based on GPS.** A solution based on location information provided by GPS requires a satellite level infrastructure. Even if the location information is obtained by triangulation techniques, it still likely requires extra infrastructure, like directional antennae. As a counterpoint to this objective, Park and Voorst [39] present an algorithm called "Anticipated route maintenance" which predicts whether a link between two nodes will be

broken within a predefined time interval, based on node locations and velocities as determined using GPS[6]. The implementation and performance evaluation of their scheme was published subsequently in [7].

(IV) **The proposed solution should not incur too much control traffic.** If the solution incurs too much control traffic, then other network parameters such as network throughput, contention level at the link layer, or delay in packet delivery may become degraded.

(V) **The solution should provide loop free routing guarantees.** Most routing protocols satisfy the condition of being loop-free. This is an important property because having routing loops consumes network capacity gratuitously, and it requires extra work to detect the loops once formed.

(VI) **The proposed solution should be responsive.** If the time required for the proposed solution to complete its path-shortening task is too long with respect to node velocity, then the expected benefit from the solution may not be achieved because by the time shortening is completed, it is obsolete.

(VII) **The solution should not be based on a specific routing protocol.** In general, the solution should be readily adaptable as an extension of a traditional routing protocol, like AODV, or DSR. To this end, we will make minimal assumptions in developing the solution. For example, we will assume that only the next hop information is available in routing tables since this is the case for almost all routing protocols to date.

---

6. Park and Voorst's algorithm consist of two phases: *Expand* and *Shrink*. The *Expand* routine prevents the route from being broken by inserting bridge nodes into a weak link; the *Shrink* routine eliminates unnecessary hops and shortens the path, thereby preventing it from becoming unnecessarily long. Unfortunately, in order for the shrinking phase to work, the nodes must exchange routing table information.

2.6 Evaluation Considerations

In evaluating our solutions, we keep the following consideration in mind:

(I.) **The solution should be evaluated at different node mobility levels.** One of the most common mobility models used in MANET simulations is Random Waypoint (RWP) mobility model. In this model, nodes randomly select their destinations within the deployment area, and start to move towards their destination at the randomly selected speeds. When they reach their destinations, they may pause (stay motionless) for some time before moving again towards new randomly chosen destination. In many studies which use this model in their measurements, the level of the mobility is adjusted either by changing the pause-time between consecutive movements of the nodes [41, 12]. The longer pause time, the lower the mobility. However, such adjustments lead to the production of a totally different sequence network topologies for each setting of mobility level. We believe that such an experimental setup may not be a good choice for discovering the relationship between mobility level and protocol performance. In this work, a different node mobility model is introduced, to address the shortcomings of the pause-time based RWP mobility model.

(II.) **The solution should be evaluated under different assumptions of network size and connection density.** This is essential to evaluate the scalability of our proposed schemes.

(III.) **The solution should be validated by extensive simulations.** The time and space complexities of the experiments performed in ns2 are high. Because of this, most prior studies perform only one trial in their works [41, 12]. However,

in order to show that conclusions are not particular to the specific scenarios of the experiments, many trials must be performed.

2.7 Assumptions

To make the proposed schemes readily adaptable to other reactive routing protocols, we are careful to not make strong AODV-specific assumptions. The following assumptions were made in the course of this work:

A1. *Uniform transmission power:* All nodes have the same transmission power; this allows us to assume bidirectionality of the links in the network.

A2. *Next hop information is available in routing tables:* Since this is required for packet forwarding, it is not a strong assumption.

A3. *Hops to destination information is present in routing tables:* At each node along a connection, the routing table maintains information about the number of hops to the destination. This is supported by AODV and many other traditional reactive routing protocols (or can be easily added to them).

A4. *Received signal strength measurement:* Nodes are able to estimate link quality based on the signal strength of received packets. Based on this, we classify links into two groups: (i) If the received signal strength is lower than a predefined threshold level[7], then the link the data packet last traversed is labeled a **weak**

7. We took this to be the strength of a received signal at 90% of the common transmission radius. For example, if the transmission radius is 250m, then threshold level corresponds 225m, when we consider the two-ray ground reflection model [45] as implemented in ns2.

link, and regarded as likely to break soon; (ii) If a link is not weak, then it is regarded as a **strong link**.

A5*. (Optional) *Distance (hop count) to the source:* Although it is not required for the operation of our schemes, some reduction in control traffic overhead can be obtained if we assume that the hop count from the source is also available at each node along the route. This information can be obtained from the IP headers of data packets travelling along the connection, if we assume that all data packets are originated with a specific TTL (e.g. 255).

2.8 Proposed Solutions

We propose two different route optimization schemes: (i) **1-hop Shrinking** and (ii) **Multihop Shrinking**. In the first scheme, we shorten two-hop subpaths of active connections into one-hop alternatives, by eliminating the middle nodes when possible. The Multihop scheme extends the 1-hop method in several aspects. Most significantly, it is capable of shortening subpaths of length $n \geq 2$ within active connections (into one-hop equivalents) when possible, while using low control traffic overhead. In the following subsections, we begin by describing the two proposed schemes in detail.

2.8.1 1-hop Shrinking Scheme

In the rest of this subsection, we refer the proposed 1-hop Shrinkin dynamic route optimization mechanism simply as **Shrink**. The Shrink mechanism becomes active after a route between a source-destination pair has been constructed by the underlying

routing protocol. The objective of the mechanism is to shorten unnecessarily long paths by eliminating inessential hops. The potential advantages of such an operation are described in Section 2.2.

The **Shrink** mechanism is initiated periodically by the source node of each connection, as long as the connection is active (i.e. has data being sent along it). Rather than considering the period to be a fixed time interval, we view it as stochastically determined by the data rate. Concretely, the source node initiates the Shrink mechanism with some fixed probability **p** every time a data packet is to be sent along the connection. The natural question that arises is what value of p should be used? This is one of the questions we will consider in the subsequent subsections, where we will consider $p = 1/4,\ 1/8,\ 1/16,\ 1/32$. In general, we will denote the Shrinking mechanism with $p = 1/\alpha$ as **Shrink-**α.

At the heart of the **Shrink** mechanism is the reduction of a 2-hop subconnection to 1-hop subconnection by eliminating an unnecessary relay node. The protocol that achieves this will be described with the aid of Figure 2.2.

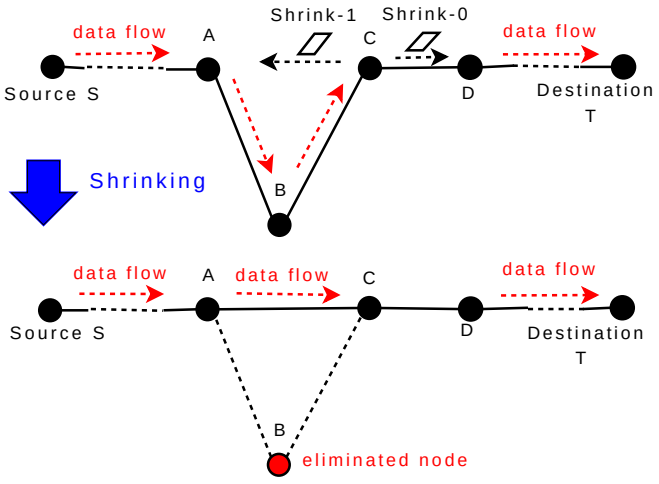


Figure 2.2: The Shrink mechanism.

From the vantage point of the source node: Suppose that source node S has established a connection to destination node T, and the route between them is constructed by the routing protocol (e.g. AODV). Assume furthermore that node S is sending data packets at constant bit rate of F packets/second. In our proposal, the shrinking mechanism is initiated by the source node S, as follows: Whenever node S has a data packet to send, with probability p it generates (an additional) special packet of type Shrink-0[8]. A Shrink-0 packet contains the following fields:

- The IP address of the previous node in the connection. Since S is the first node in the connection, there is no previous hop, and so in this case, the Shrink-0 packet generated by S has a special sentinel value in this field.
- The IP address of the sender (i.e. node S in this case),
- The IP address of the final destination (i.e. node T).
- The time to live (TTL) value (which is set to 1 by S).

Node S makes the Shrink-0 packet and unicasts it to the next hop, along on the same route as the data packet.

From the vantage point of intermediate nodes: When an intermediate node (in the example, node C) receives a Shrink-0 packet from the previous hop (i.e. node B), it produces two Shrink packets, one with flag 0 and one with flag 1. The packets include the IP addresses of the previous node, the current node, and the final destination. The Shrink-0 message is sent to the next hop (i.e. node D) along the path towards the destination. The Shrink-1, on the other hand, is sent to the upstream

8. Here the 0 represents the status of a flag in the packet header.

node two hops away (i.e. node A) with TTL of 1. Notice that the IP address of the upstream node two hops away (i.e. node A) is retrieved from the header of received Shrink-0 packet.

Elimination of the unnecessary hops: In this stage, two subsequent links on the route are replaced by just a single one, provided all necessary conditions are satisfied: When a node receives Shrink-1 packet from another node, then it can be concluded that there may be a shortcut available between the receiver and the sender of Shrink-1 message. However, the quality of this link may not be good enough to warrant changing the routing table. Therefore, before any update to the routing table, the quality of the new link is checked by regarding the received signal strength at the receiver (i.e. at node A in Figure 2.2). If the received signal strength is greater than a predefined threshold value, the node updates its routing table so that the next hop for the final destination (specified in the Shrink-1 packet) is replaced with the address of the originator of the Shrink-1 packet. The process is illustrated in Figure 2.2.

In the previous paragraphs, we have explicated the proposed Shrink mechanism in terms of data flowing along a single connection, in order to make the exposition simpler. In practice, however, there can be many data flows (i.e. many connections) between different source and destination pairs, simultaneously. Such a situation does not alter the operation of the proposed mechanism, since the shrinking of each connection occurs independently of all others.

The key *new* ideas of the proposed protocol are:

1. To send a Shrink-1 packet two hops away in the opposite direction from data flow, and using it to check if there is a shortcut between upstream and down-

stream nodes of each triplet along the route.

2. To amortize route optimization traffic against actual data traffic along the connection, by performing shrinking stochastically relative to data packets.

2.8.2 Multihop Shrinking

The approach of this subsection, termed Multihop Shrinking, extends the previously described method of 1-hop shrinking, in two crucial ways:

1. The new mechanism does not confine itself to successive *triplets* of nodes, thereby making it possible to make dramatic global reductions in path length. The Multihop Shrinking scheme detects direct shortcuts, not only between the nodes in a contiguous triplet, but rather between *any pair* of nodes within the connection.

2. The control traffic overhead incurred by the proposed Multihop Shrinking is approximately half of what was incurred by its predecessor, 1-hop Shrinking. At the same time, the new scheme yields significant improvements over classical AODV in terms of retaining path optimality over time.

Like 1-hop Shrinking, the objective of the Multihop Shrinking is to shorten unnecessarily long connections by eliminating inessential hops as illustrated in Figure 2.3. In order to do this, the mechanism periodically checks to see if there is any direct shortcut between non-adjacent pairs of the nodes within the connection. More precisely, the mechanism repeatedly checks whether there is any upstream node within the direct transmission range of a node, performing this check for each node along

the connection. If there is such a shortcut between two distant nodes on the same route, and if the channel quality of the shortcut is "good enough", then the Multihop Shrink mechanism modifies the connection topology so that two end-point nodes of the shortcut connect to each other directly, thereby eliminating the inessential intermediary *node(s)* between them (see Figure 2.3). The advantages of such an operation are detailed in Section 2.2.

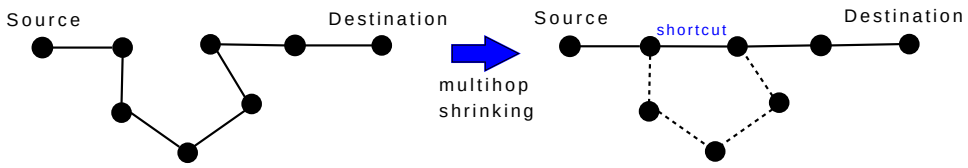


Figure 2.3: Main idea of *Multihop Shrink*.

The Multihop Shrinking mechanism is only activated *after* the routing protocol has established a route between source and destination. After the connection has been established, and for as long as the connection is active, the source node periodically sends a special Shrink packet downstream towards the target. This packet triggers the shrinking operation at each node it traverses.

The periodic sending of the Shrink packet is stochastically determined by the flow of data on the connection. More precisely, the source node of the connection initiates the shrinking process when a certain fixed amount of data traffic has been sent on the connection. In our experiments, we consider connections that carry constant bit rate (CBR) traffic that is conveyed in fixed size packets and thus, the shrinking process is initiated every time a specified number ($\boldsymbol{p}$) of data packets have been sent. We can avoid keeping a data packet counter at each node for each destination by implementing the counters probabilistically: each source node initiates a Shrink packet

with probability $1/p$ whenever it sends a data packet[9]. In what follows, however, for simplicity of exposition, we will not consider this space optimization. A natural question that arises concerns the effect of the choice of p on the performance of the optimization scheme. This is just one of the questions we will investigate in the subsequent sections, where we will consider $p = 4$, 8, 16, 32. In general we will denote the Shrinking mechanism when $p = \alpha$ as **MShrink-α**.

As the Shrink packet travels downstream, nodes that see the special packet attempt to discover a shortcut to *upstream* nodes on the connection[10]. In summary, the ***Multihop Shrink*** mechanism achieves its goal of optimizing connection topology by replacing inefficient multihop subconnections with a direct 1-hop connection thereby eliminating unnecessary relay node(s). The protocol that achieves this will be described with the aid Figure 2.4.

Initiation of the process: Suppose that source node S has established a connection to destination node T, and the route between them has already been constructed by a reactive routing protocol such as AODV. Assume furthermore that node S is sending data packets at constant rate of F packets/second. In our proposal, the shrinking mechanism is initiated by the *third node B*, which can recognize itself as the third node by looking at the corresponding field in the IP header of data packets received, as per assumption A4*. Periodically, upon receiving a certain number of data packets (4, 8, 16, and 32, respectively), the *third node B* initiates the procedure by making a

9. Admittedly, in this space optimization, the operation of the random number generator consumes time.

10. A further optimization is evident now, since such a shortcut can never be found by the source (the first node), or the second node. Thus, in scenarios where the assumption A4* can be made (see Section 2.7), the shrinking operation is initiated by the *third node* within the connection, rather than at the source. In settings where assumption A4* cannot be made, identifying the third node may not be feasible: the Multihop Shrink scheme still operates correctly, but is slightly more wasteful in terms of control traffic overhead.

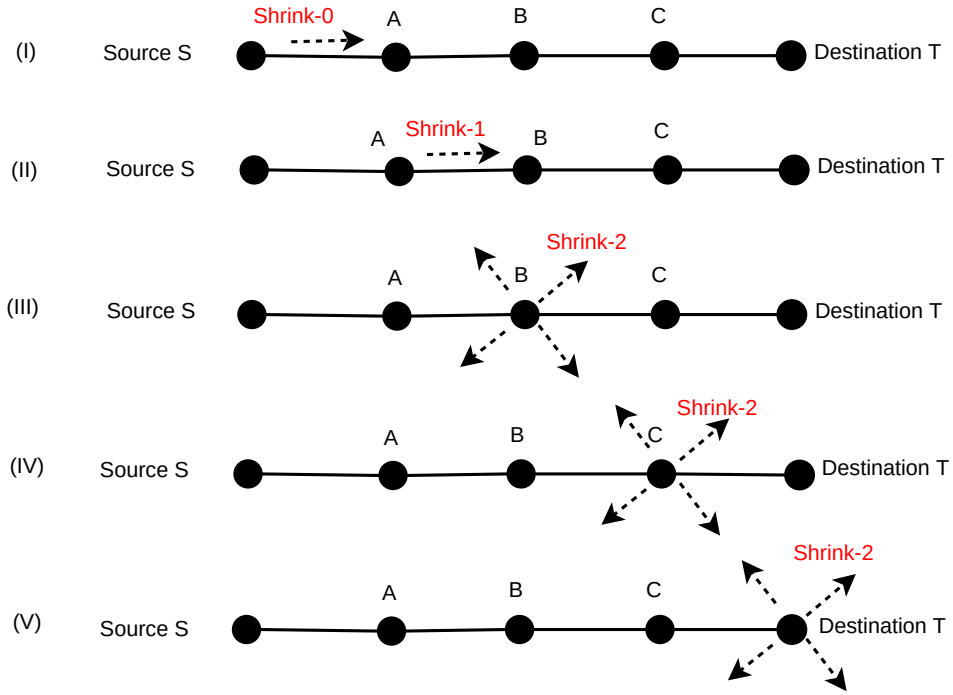


Figure 2.4: How Multihop Shrinking works.

special Shrink packet, and *broadcasting* it with TTL=1 (which means the packet can go at most 1-hop away from its originator). A Shrink packet contains the following fields:

- **sender**: The IP address of the sender of the Shrink packet (i.e. node B in the example),
- **next-hop**: The IP address of the next hop (i.e. node C),
- **final-destination**: The IP address of the final destination (i.e. node T).
- **hops-to-destination**: The number of hops to the final destination (available in the routing table)
- **TTL**: The time to live (TTL) value (which is set to 1).

From the vantage point of other nodes: A node which receives a Shrink packet performs actions which depend on its relative placement within the connection with respect to sender of the Shrink packet. Each node, upon receiving the Shrink packet, determines its relationship to the sender of the Shrink packet, from the following set of five mutually exclusive classes:

1. *The next hop:* A node identifies itself to be the *next hop* by noting that its own IP address is the one specified in the next-hop field of Shrink packet it received. In that case, it modifies the received Shrink packet by updating the related fields using the information in its routing table, and then broadcasts the updated Shrink packet.

2. *Further downstream hops:* A node recognizes itself to be in this class if its routing table indicates that the hop count to the destination is smaller than the value in the hops-to-destination field of the received Shrink packet. Nodes in this category discard the received Shrink packet.

3. *The previous hop:* A node identifies itself to be the *previous hop* when it determines that the sender of the Shrink packet is listed in its routing table as the next hop to the final-destination. This node discards the received Shrink packet.

4. *Further upstream nodes:* A node recognizes itself to be in this class if its routing table indicates that the hop count to the destination is greater than the value in the hops-to-destination field of the received Shrink packet. If this is the case, then it can be concluded that there may be a shortcut available between the node and the sender of Shrink message. However, the quality of the new shortcut must be be good enough to warrant changing the routing table. To

ensure this, before any update to the routing table, the quality of the new link is checked by determining if the signal strength at which the Shrink packet was received is greater than a predefined threshold level. If so, the node updates its routing table to make the next hop for the final-destination (as specified in the Shrink packet) be the address of the sender the Shrink packet.

5. *Irrelevant nodes:* When a node receives Shrink packet, but there is no next hop in its routing table for the final-destination specified in the Shrink packet, the node classifies itself as irrelevant to the Shrink packet, and simply ignores it.

Every node receiving a Shrink packet falls into precisely one of the above five classes, and behaves as specified. In practice, there can be many data flows (i.e. many connections) between different source/destination pairs, all active simultaneously. Such a situation does not alter the operation of proposed mechanism, since each node responds to different Shrink messages arising from different transient connections independently. No further state needs to be maintained at each node than what is mandated in AODVs routing table format and a 4 bit counter for each destination address in the routing table[11].

11. As noted earlier, this minor space requirement can be removed by considering a scheme in which Shrink packets are generated probabilistically with probability $1/p$ every time a data packet is sent from the source node. However, this in increases the time required to process each data packet, since random number generation takes time.

2.9 Performance Analysis via Simulation

2.9.1 Simulation Environment

The proposed route optimization algorithms have been implemented in the network simulator ns-2.33 [3]. This software supports a packet-level, discrete-event network simulations, and is commonly employed by researchers in the area of wired and wireless networking. It provides implementations of routing and multicast protocols over wired and wireless (local and satellite) networks. It is written in C++ and an object oriented version of *Tcl* named *OTcl* [3, 4].

MAC layer representation: *ns*2 offers implementations of several MAC layer protocols for the simulation of wireless networks [5]. In our experiments, we used the implementation of IEEE 802.11 distributed coordination function (DCF) as the MAC layer protocol [2]. This provides us with a very good representation of contention in channel utilization.

Radio propagation model: *ns*2 contains the implementations of three alternative radio propagation models: the free space model, the two-ray ground reflection model, and the shadowing model. We have used a hybrid of the first two models in our own experiments. The hybrid is described in what follows.

In the free space model, it is assumed that the signals emerging from the transmitter follow a line-of-sight path towards receiver while degrading along the way. The received signal power is calculated as a function of the distance from the transmitter

according to the Friis transmission equation [21].

$$P_r(d) = \frac{P_t G_t G_r \lambda^2}{4(\pi)^2 d^2 L},$$

where P_t is the transmitted signal power, G_t and G_r are the antenna gains of the transmitter and the receiver respectively, $L \geq 1$ is the system loss, and λ is the wavelength. It is quite common to set $G_t = G_r = 1$ and $\lambda = 1$ in ns simulations.

On the other hand, the two-ray ground reflection model takes into account both the direct path and a ground reflection path of the traveling signal. It is known [45] that this model yields more accurate prediction at long range compared to the free space model. The received power at distance d is estimated by

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

where h_t and h_r are the heights of the transmit and receive antennas respectively.

Comparing the two equations above, it can be easily seen that the latter equation exhibits more power loss, as distance increases. However, the two-ray model does not give good results at short distances due to the oscillation caused by the constructive and destructive combination of the two rays. Therefore, ns2 suggests to use a combination of both models in such a way that the free space model is valid when d is small, and the two-ray model is used when d is big. The cross-over distance, d_c, for this hybrid model, is the point at which both equations give the same result. In our experimental setup, this occurs around 100 meters. This can be obtained by solving $d_c = (4\pi h_t h_r)/\lambda$ with the transmission range of nodes set to 250 meters.

Post-processing of trace files: At the end of each experiment, ns2 produces a trace file including all packet interactions at the MAC, network, and agent layers. In order to retrieve meaningful results from these trace files, we need to process them. During the post-processing stage of these trace files, we had to deal with two issues:

- *Space limitations*: Trace files from experiments often require huge amounts of storage space. For example, a single trace file produced from an experiment including 100 nodes simulated for 600 seconds, may require up 1GB space in hard drive. Since we have many schemes to evaluate under several networks regimes, mobility levels, and traffic loads, we had to address space limitations by running the computations across a cluster of workstations with independent disks.

- *Time limitation*: Processing trace files often takes a lot of time since related information must be extracted from each packet at various layers. For example, it could routinely take up to 24 hours to calculate average end-to-end delay from a trace file of a 1200 second simulation of 50 nodes with 10 data traffic loads. To make this process faster, we loaded required information from trace files into a MySQL database system. Then, we wrote SQL scripts to compute the required information from the database, and shell scripts to prepare data files for plotting graphs.

Plotting the graphs: The graphical representations of the experimental results were made with aid of the "gnuplot" software [1].

2.9.2 Experimental Setup

The performance of our proposed extensions was compared to the original AODV , under different assumptions of network size, mobility level, and data traffic load.

Networks: Several network sizes were investigated, comprised of between 50 and 150 nodes. Node density was kept fixed at 50 nodes per 700m x 700m. Using these spatial parameters also allowed us to make comparisons with other researchers, who considered similar areas and node densities [15, 41]. Initial placement of nodes was uniform random placement.

Traffic Load and Patterns: The traffic connections were initiated between randomly chosen source and destination pairs chosen uniformly from the nodes. The number of connections varied between 1 and 50, depending on the specific experiment. Traffic sources generated constant bit rate (CBR) traffic consisting of packets of size 512 bytes, at a rate of 4 packets per second.

2.9.3 Mobility Model

We would like to investigate the performance of the proposed mechanisms under different mobility levels such as high mobility, low mobility, or no-mobility cases. To do this, we have modified the random waypoint mobility model, which is provided by $ns2$, as follows.

$ns2$ allows us to produce random movement scenarios based on the random way point mobility (RWP) model with pause times [14]. In the standard RWP model,

the nodes are randomly situated within the deployment area at the beginning of the simulation. Each node then starts to move towards a randomly chosen location in the deployment area with a speed chosen at uniformly random between 0 and v_{max} m/s. When a node reaches its destination, there is a **pause time** before it starts a new movement. The pause time is used to adjust level of mobility. When pause time is set to 0, the nodes move continuously without stopping, which provides a maximally mobile setting. On the other hand, when pause time is set to the duration of the entire simulation, the nodes remain fixed; this provides us with a stationary network. Intermediate values of pause time correspond to intermediate levels of mobility. We found that this standard RWP model did not elucidate the effects of mobility on the proposed optimization schemes, because varying the pause time yields totally different sequences of link level changes in the network topology. Because the experiments performed at different mobility levels are making reference to totally different sequences of network topologies, we get incomparable performance measurements.

To investigate the effect of different mobility levels on the proposed schemes, the same sequence of network topologies must be generated, regardless of mobility level. At high mobility levels, the sequence should be played more rapidly than at low mobility levels. In order to achieve this effect, we modified the random way point mobility model as follows.

First we set the pause time to zero (nodes move without stopping between subsequent movements). Then we generated a movement plan with maximum speed of 5 m/s. To obtain higher mobility scenarios, we multiplied the velocity of each node by $\beta > 1$. Since the velocity of each node is multiplied by the same factor β, and there is no

pause time between subsequent movements of nodes, the resulting sequence of network topologies is the same regardless of mobility-level. The advantage of this modified RWP is that as β is increased, the same sequence of (link level) networks arise, albeit more quickly. We considered $\beta = 1, 2, 3, 4, 5$, with larger values of β signifying higher mobility scenarios. Note that under this mobility model, the simulation duration changes inversely with β. For example, when maximum speed is 5 m/s the simulation duration is 1200 seconds, while it is 600 seconds when the maximum speed is 10 m/s. Our analysis of performance metrics (with respect to mobility) is sensitive to the fact that simulation durations are different as β varies.

2.9.4 Performance Metrics

To measure the performance of the proposed route optimization schemes, we used the following metrics:

- **Normalized Path Length** (NPL) is measured by noting for each data packet delivered to its destination, both the number of hops that packet traveled, and the length of the optimal source-destination path at the time of packet delivery. Then discrepancy between these numbers is then computed, both as ratio and as a difference. Note that the length of the optimal path for each delivered packet is provided by $ns2$ based on Dijkstra's shortest path algorithm. The NPL metric may be the most significant one in our investigation, because the primary objective of the proposed schemes is to optimize path lengths.

- **Packet delivery fraction** (PDF) is calculated as the percentage of successfully delivered data packets to destination nodes, out of the data packets originated

at source nodes. This metric permits us to determine if there is an increase or reduction in packet delivery ratio because of route optimization process.

- **Normalized Routing load** (NRL) is calculated as the number of routing-related control packets transmitted *per* data packet delivered to its destination. This metric allows us to quantify how much extra control traffic is incurred by the proposed algorithms, and therefore, it can be regarded as a measure of the cost of the proposed schemes.

- **Average connection lifetime**(ACL), where the time period between two consecutive link breakages on a connection is taken as *a lifetime*. We compare the average value of these periods when dynamic route optimization is being user, to their duration under pure AODV. This metric can be used to confirm the relationship between connection lifetime and route length.

- **Average End-to-End Delay**(E2E)– For each packet delivered to destination, we calculate the time interval between the time the packet left source node, and the time it reached to destination. The average of all such values gives us the average delay in packet delivery. As described in section 2.2, dynamic route optimization is expected to yield reduction in delay of packet delivery. This metric is the basis of verifying the extent to which the expectation of latency improvements are borne out in practice.

- **Normalized Energy consumption per packet**(NEN)– We count the number of transmissions which occurred, and the number of data packets that were successfully delivered, both weighted by packet sizes. The ratio of these two numbers gives us the energy consumption per data packet delivered. A reduction in the number of hops that the data packets travel could be expected to

provide energy saving per packet transmission. We can check this hypothesis by using this metric.

Trials:

To increase our confidence in conclusions drawn from the analysis of simulation data, we repeated 10 trial experiments for each setting of the independent variable (with mobility plans and traffic patterns generated as above). We then computed the mean and standard deviations of the performance metrics (dependent variables), and examined how these varied as the independent variable (system parameter) was changed.

2.9.5 Results and Analysis

2.9.5.1 1-Hop Shrinking

In this subsection, we present and analyze experimental data from 1-Hop Shrinking simulations, considering the performance metrics described in the previous section. In the rest of the subsection, we call the proposed route optimization mechanism as simply **Shrink**. We evaluated four different versions of the proposed Shrink scheme, namely SHRINK4, SHRINK8, SHRINK16, and SHRINK32. SHRINKα (where α is 4, 8, 16, and 32) means that an end-to-end shrink operation is initiated by source node after every α data packets are sent by the source node. Therefore, SHRINK4 and SHRINK32 represent the most and least frequent shrink operations, respectively, while SHRINK8 and SHRINK16 are intermediary between them.

Normalized Path Length (NPL):

In this set of experiments, we measured the optimality of the paths for pure AODV and AODV+Shrink(α) under different mobility levels and network regimes. This metric, NPL, may be the most important one as it captures the extent to which the proposed scheme meets the primary objective of optimizing path length. The results are averaged based on 10 experimental trials, each of which consists of about 50,000 data packets processed in a network of (i) 50 nodes with 10 traffic loads, and (ii) 100 nodes with 10 traffic loads. The outcomes are normalized with respect to the optimal path lengths as calculated by Dijkstra's shortest path algorithm. We vary mobility settings from a maximum velocity of 5 m/s ($\beta = 1$) to a maximum velocity of 25 m/s ($\beta = 5$).

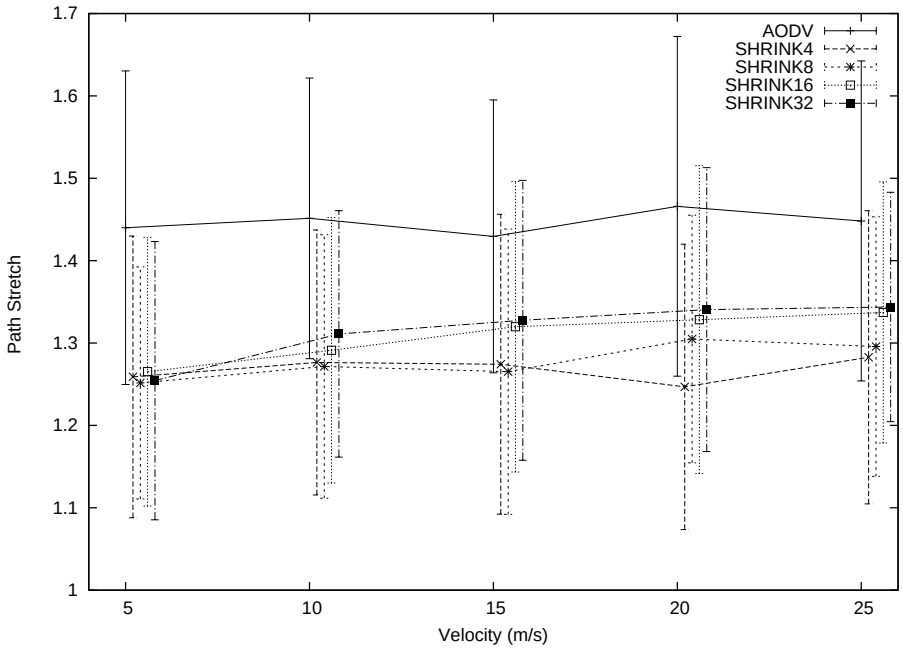


Figure 2.5: NPL vs. velocity (n=50, c=10)

Figure 2.5 depicts the extent to which Shrink scheme maintains path optimality at different mobility levels for 50-node network, as quantified by mean normalized aver-

age path length. According to the figure, the paths produced by standard AODV are about 45% longer than optimal, regardless of the mobility level. When the Shrink is in effect, however, path lengths are between 25% and 34% longer than optimal, depending on both mobility level and α version of the scheme. At this point, we should notice that many error bars in the figure overlap each other, which means it may not be justifiable to compare the tested mechanisms based on mean values of the experimental results. On the surface, if error bars overlap each other, it is possible that a scheme (e.g. SHRINK16) can occasionally result in better performance than another one (e.g. SHRINK8), even though average value of the former (e.g. SHRINK16) is worse than average value of the latter (e.g. SHRINK8). In order to determine whether the tested mechanisms are comparable based on average values of the experimental trials, we used the following method:

For each pair of schemes and each mobility level, we calculated the correlation coefficient between the performance measures of the corresponding experiment trials. This can yield a coefficient between -1.0 and 1.0. The more a coefficient is closer to 1.0, the more perfect the correlation between corresponding pair of curves. In a perfect correlation between curves, we can conclude that in spite of overlapping error bars, performance of the schemes moves up/down in unison on the y-scale in different trials, and for this reason the schemes can be related in the manner suggested by their mean values. In this section, due to space considerations, we will provide correlation coefficient tables only for the mobility level of 15 m/s. Other velocities produce correlation tables with similar value range.

Table 2.1 lists correlation coefficients for the curves in Figure 2.5 when maximum velocity is 15 m/s. Since all coefficients are greater than 0.9, there are considerable

correlation between the experiments underlying the curves.

Table 2.1: Correlation coefficients for curves in Figure 2.5.

vel=15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.981	0.957	0.956	0.968
SHRINK4	-	1	0.991	0.953	0.971
SHRINK8	-	-	1	0.947	0.958
SHRINK16	-	-	-	1	0.908
SHRINK32	-	-	-	-	1

Given this, we can interpret results based on average values of experiments. According to the figure, the paths produced by standard AODV are about 45% longer than optimal one, regardless of the mobility level. When Shrink is in effect, however, path lengths become at most 34% of optimal. From this, it can be readily concluded that *the proposed mechanism works.* Another observation that can be deduced from the figure is that shrinking operations running more frequently (i.e. SHRINK4 as opposed to SHRINK32) yield better results, especially at high mobility levels. This is because it is less likely for us to miss a possible shortcut when shrinking operations are done more frequently. It can be also seen from the figure that, for schemes SHRINK16 and SHRINK32, normalized path lengths increase slightly as mobility level increases. This is because the decisions made by the shrinking operation yield benefits for ever shorter time intervals as mobility increases. For schemes SHRINK4 and SHRINK16, normalized path lengths draw almost a flat pattern as mobility level increases. This is because their operational shrink frequency is so high that mobility level does not (yet) affect their performance in path optimality. Pure AODV also has a flat curve with respect to mobility level, but the reason for this is that (i) AODV reconstructs routes more frequently as mobility level increases due to more frequent link breakages, and (ii) paths tend to be near-optimal at the beginning of their construction. An interesting pattern exhibited by the curves in Figure 2.5 is

that versions of the proposed scheme converge to path stretch 1.25 at 5 m/s, but get gradually separated from each other as mobility level increases. This means that excessive usage of shrinking operation in low mobility settings (e.g. SHRINK4 at 5 m/s) does not provide extra improvement in path optimality. This indicates that there is non-trivial lower bound > 1 on path stretch achievable by the proposed *Shrink-*α scheme, regardless of the value of α.

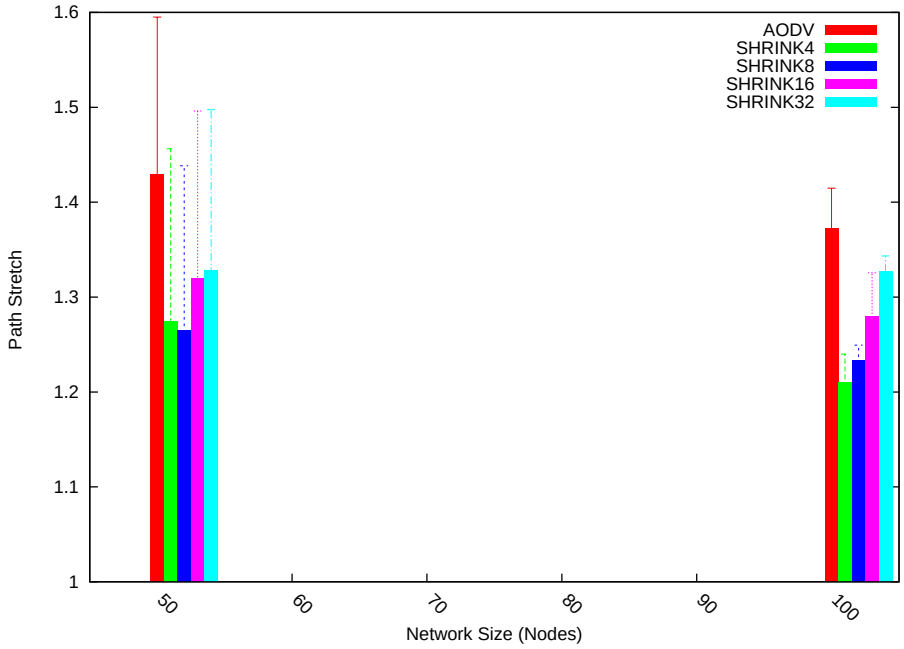


Figure 2.6: NPL vs. network size (v=15m/s, c=10)

From another point of view, Figure 2.6 depicts relationship between normalized path length and network size, while maximum velocity is fixed at 15 m/s. According to the figure, the proposed Shrink scheme continues to perform well in large networks, which is indicative of its scalability. The 100-node analogue of Figure 2.5 is Figure 2.7, which shows normalized path optimality at different mobility levels in a network with 100 nodes and 10 traffic connections. The performance of the evaluated mechanisms in a 100-node network is comparable to their performance in a 50-node network, with the only difference being a magnification of the divergence between the proposed schemes.

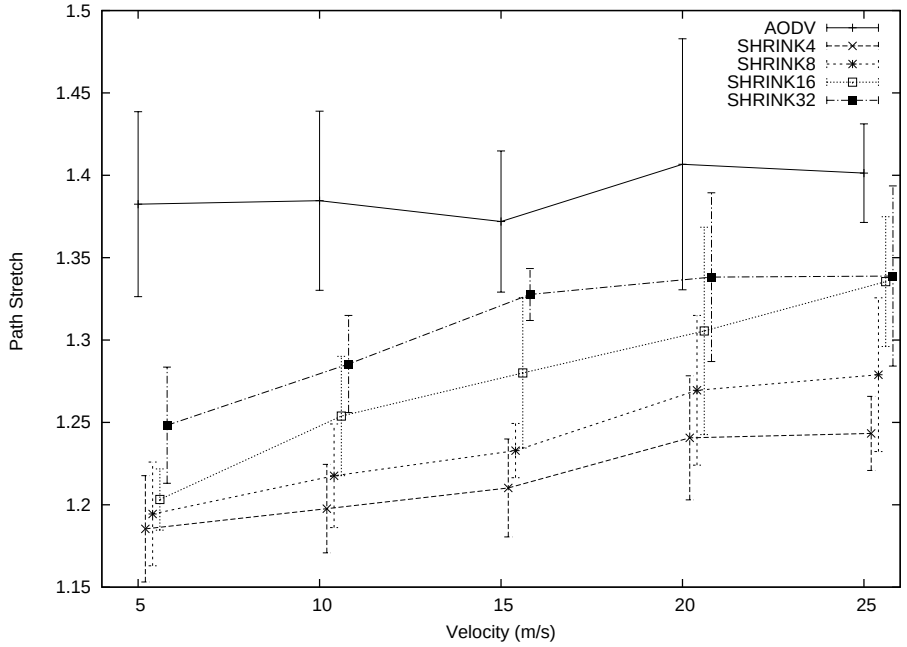


Figure 2.7: NPL vs. velocity (n=100, c=10)

Average energy consumption per packet (NEN):

In this next set of experiments, we measured the average energy required to deliver data packets to corresponding destinations. We call this metric *normalized energy* (NEN). Figure 2.8 shows NEN at different mobility levels for a network with 50 nodes and 10 traffic connections. According to the figure, pure AODV consumes, on average, between 1425 units and 1590 units of energy per delivered data packet. The amount rises as mobility level increases. On the other hand, when the proposed Shrink scheme is in effect, average energy consumption per data packet varies between 1305 units and 1528 units, depending on both the version of the proposed scheme, and the mobility level.

Table 2.2 indicates correlation coefficients for the curves in Figure 2.8 at maximum velocity of 15 m/s. As seen in the table, correlation coefficients are bigger than

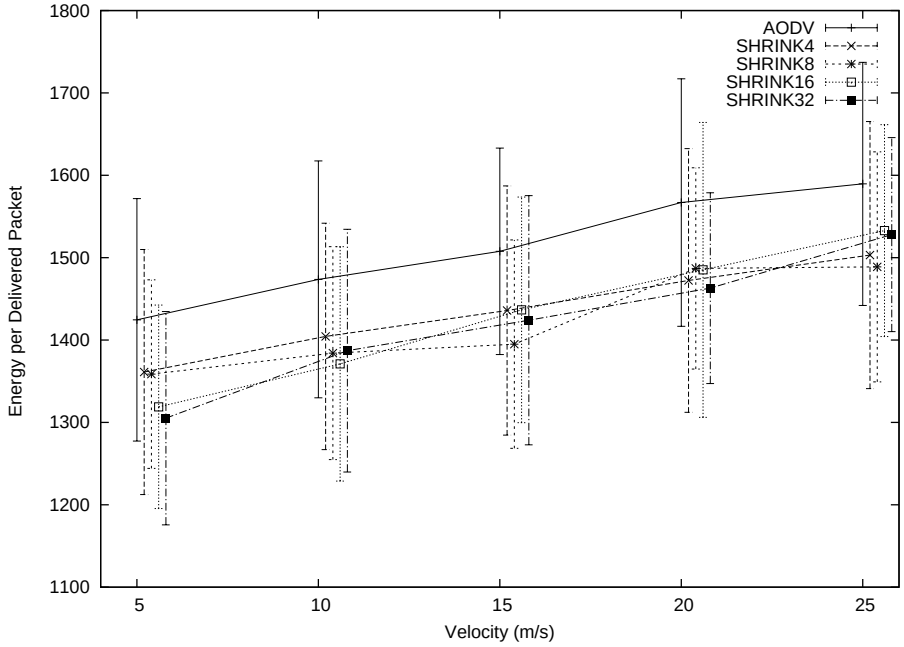


Figure 2.8: NEN vs. velocity (n=15, c=10)

Table 2.2: Correlation coefficients for curves in Figure 2.8.

vel = 15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.977	0.927	0.939	0.930
SHRINK4	-	1	0.958	0.943	0.952
SHRINK8	-	-	1	0.919	0.933
SHRINK16	-	-	-	1	0.923
SHRINK32	-	-	-	-	1

0.919, which implies there are considerable correlation between the experimental data underlying curve pairs, and their means may be taken as directly comparable even in the face of overlapping error bars. It is obvious from the curves then that the proposed schemes reduce the average energy required to deliver a data packet to its destination. This is because data packets follow shorter routes when the proposed dynamic route optimization scheme is used, and thus, fewer data transmissions take place. It can be also be seen from Figure 2.8 that SHRINK32 has a lower NEN value than SHRINK4 at 5 m/s, while SHRINK4 has lower NEN value than SHRINK32 at 25 m/s. Notice that the curves cross each other at around 22 m/s. Given that

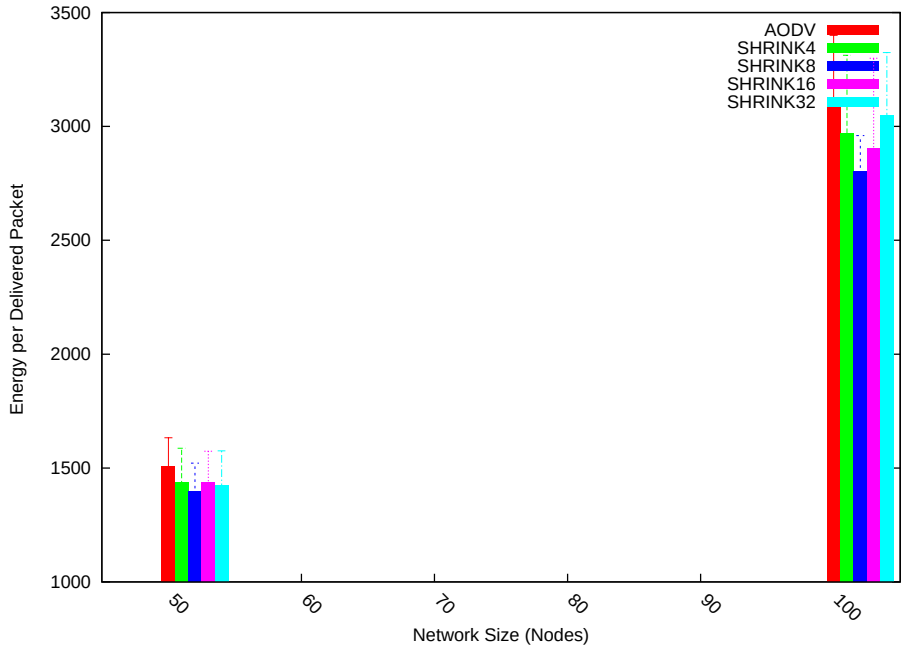


Figure 2.9: NEN vs. network size (v=15m/s, c=10)

data traffic is assumed to be CBR, and shrinking is amortized against data traffic, it is understandable that in low mobility settings one can apply shrinking operations less frequently, and obtain energy savings. In contrast, it is preferable to apply the shrinking operation more frequently in higher mobility settings, since the nodes have moved further in between shrinking phases. Another observation deduced from the figure is that NEN rises as mobility level increases. This can be explained by the fact that there is an increase in control traffic overhead in high mobility settings, due to more frequent link failures.

From another perspective, Figure 2.9 shows the relationship between average energy consumption per data packet delivered, and network size. According to the figure, energy consumption increases as network size gets larger. This is because (i) expected path lengths is longer in large networks than it is in small networks, and therefore, data packets follow longer routes which require more transmissions, and (ii) control

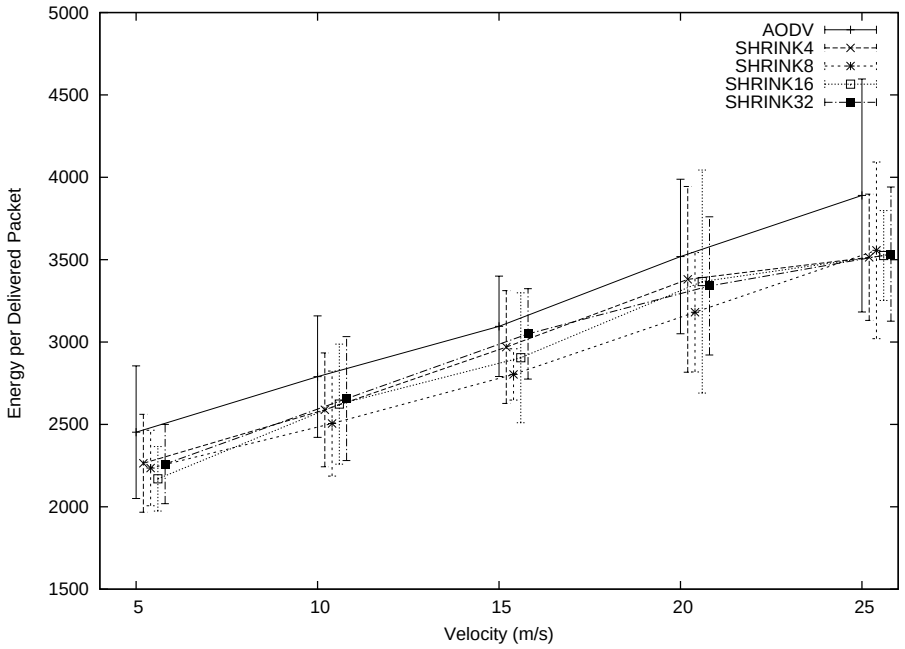


Figure 2.10: NEN vs. velocity (n=100, c=10)

traffic overhead becomes high in large networks, since flooding becomes expensive, and this in turn, consumes more energy. In addition, we can infer from Figure 2.9 that there is an optimum frequency for shrinking operations (with respect to NEN minimization), and this optimal depends on network size. For example, when maximum velocity is 15 m/s, SHRINK8 performs better than all other versions, including pure AODV, in both 50-node and 100-node networks. The results of the 100-node version of this experiment are depicted in Figure 2.10, which shows average anergy consumption per data packet delivered, at different mobility settings, in a network with 100 nodes and 10 traffic connections. It can be seen from the figure that the proposed scheme reduces NEN values even in larger networks, regardless of mobility level.

Normalized Routing Load (NRL):

In this set of experiments, we examined the traffic load incurred by the proposed Shrink mechanism. Figure 2.11 indicates overall routing load normalized by the number of data packets delivered to destinations, for both pure AODV and AODV+Shrink(α) mechanisms at different mobility levels. According to the figure, pure AODV has NRL values of 0.43, 0.78, 1,13, 1,47 and 1.75 at maximum velocity settings of 5 m/s, 10 m/s, 15 m/s, 20 m/s, and 25 m/s respectively. When the proposed Shrink scheme is applied, these values rise a certain amount, depending on version of the scheme. For instance, when maximum velocity is 15 m/s, normalized routing loads of AODV, SHRINK32, SHRINK16, SHRINK8, and SHRINK4 are about 1.13, 1.19, 1.29, 1.43, and 1.88 respectively.

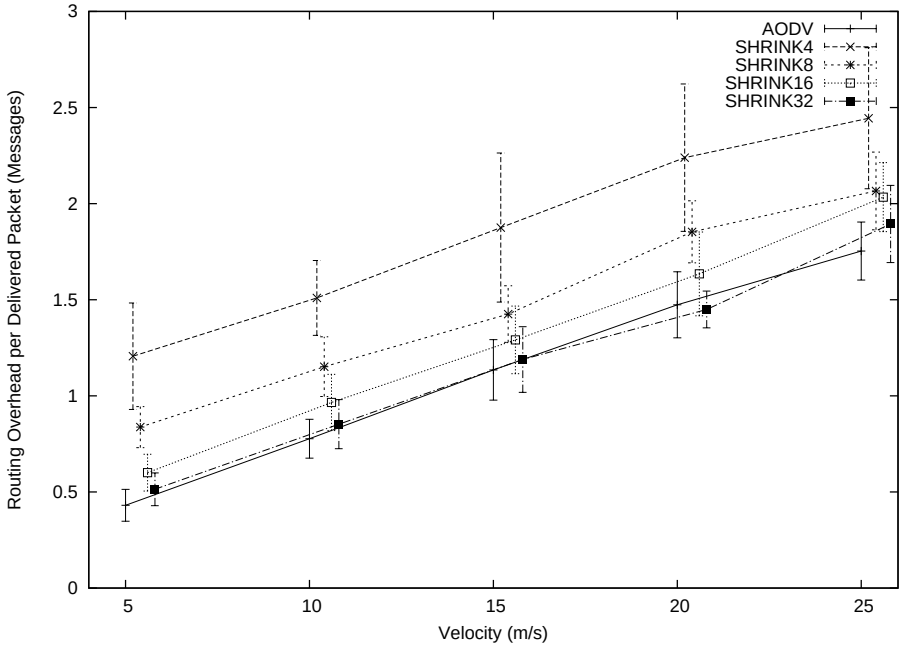


Figure 2.11: NRL vs velocity (n=50 nodes, c=10)

Table 2.3 contains correlation coefficients for the curves in Figure 2.11 at maximum velocity of 15 m/s. As seen in the table, correlation coefficients are in the range 0.70-0.95, indicating significant correlations between curve pairs. Since we see correlation between trials, we interpret the curves in Figure 2.11, in spite of some overlap in

Table 2.3: Correlation coefficients for curves in Figure 2.11.

vel = 15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.823	0.881	0.705	0.830
SHRINK4	-	1	0.950	0.791	0.820
SHRINK8	-	-	1	0.826	0.880
SHRINK16	-	-	-	1	0.801
SHRINK32	-	-	-	-	1

the error bars. It can be seen from the figure, that pure AODV frequently has the lowest NRL value. This is because the proposed *Shrink* scheme incurs extra control traffic at the expense of improving route optimality. In addition, it can be deduced from the figure that the Shrink schemes that apply the shrinking operation more frequently (e.g. SHRINK4) introduce much more control traffic than other versions of the scheme which apply shrinking operations less frequently (e.g. SHRINK32).

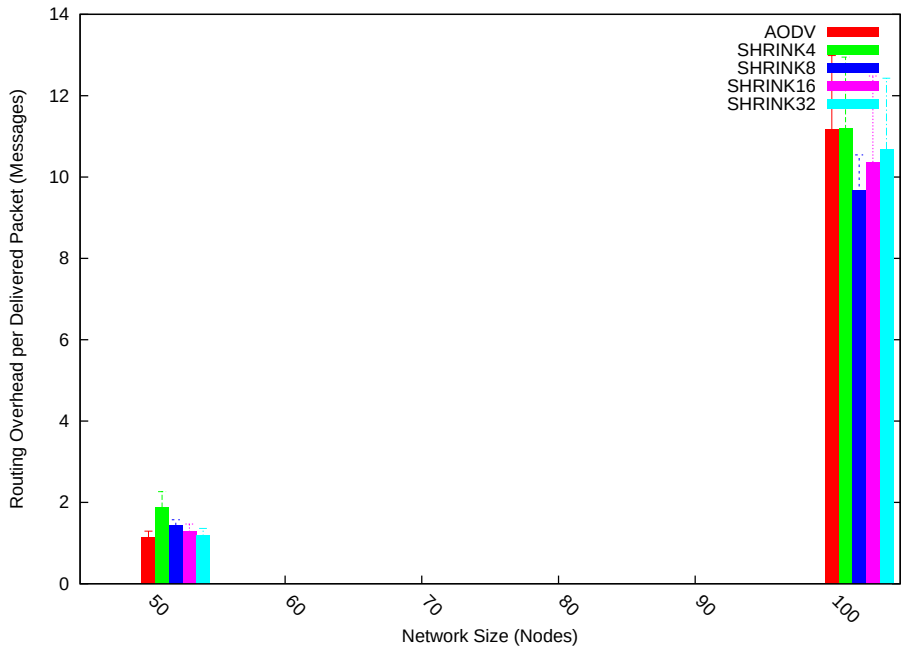


Figure 2.12: NRL vs network size (v=15, c=10)

From another perspective, Figure 2.12 indicates relationship between normalized routing load (NRL) and network size. An interesting observation deduced from the figure

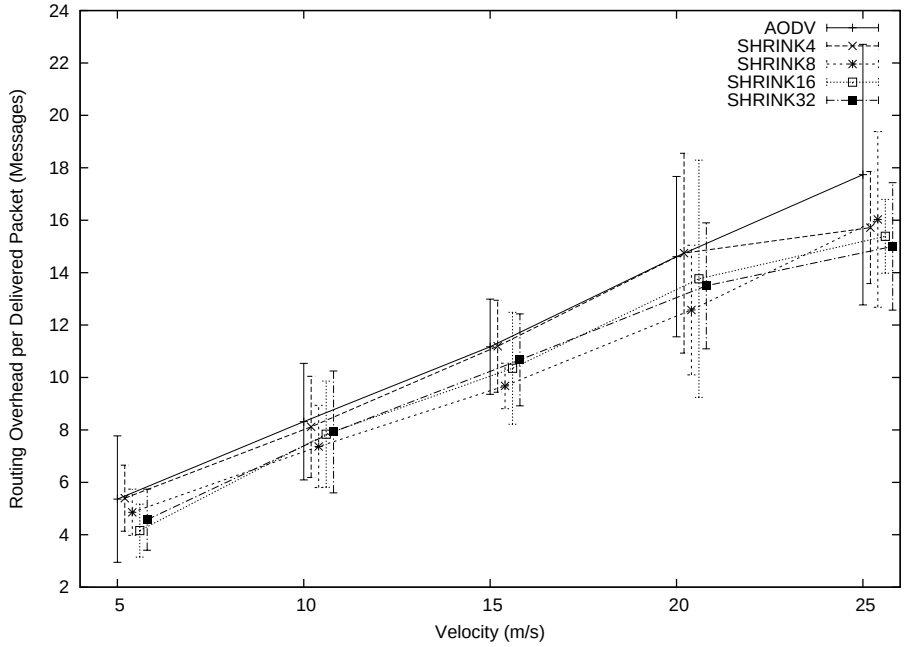


Figure 2.13: NRL vs velocity (n=100, c=10)

is that applying the proposed Shrink scheme reduces NRL in large networks (e.g. in 100-node network), in spite of the fact that it incurs extra control traffic for route optimization. For example, in 100-node network, applying any version of the Shrink scheme yields lower NRL values compared to pure AODV. This is because route optimization reduces the expected number of global route discovery attempts, which is an expensive operation (in terms of control traffic) in large networks. We will confirm this theory in the subsequent experiments, where we investigate average connection lifetime. Figure 2.13 is 100-node version of Figure 2.11. It describes the relationship of NRL at different mobility levels, for a network of 100 nodes and 10 traffic connections. It can be seen from this figure that the proposed Shrink scheme reduces NRL in large networks, regardless of the mobility level. The amount of the reduction, however, depends on which version of the scheme is used, and the mobility level of the nodes.

Packet Delivery Fraction (PDF):

In this set of experiments, we investigated possible impact of the proposed Shrink scheme on packet delivery fraction (PDF), at different mobility levels and in several network regimes. Figure 2.14 shows PDF at different mobility levels for AODV and AODV+Shrink(α), in a network including 50 nodes and 10 traffic connections. According to the figure, pure AODV has a PDF value between 99.3% and 96.8% depending on mobility level, linearly decreasing with velocity of nodes. On the other hand, the figure shows that AODV+Shrink(α) has PDF values varying between 96.7% and 92.4% depending on both mobility level and the version of the proposed scheme. Because the error bars overlap significantly, we provide Table 2.4, which lists the correlation coefficients for the experiment trials across schemes. As the table shows, correlation coefficients are quite low in some cases, implying insignificant correlation between the curves. Mean values of SHRINK(α) curves cannot be compared directly, though we can conclude that pure AODV often has better PDF than AODV+Shrink, since the curves lie outside of each other's error bars. We also checked correlation coefficients for the trials which comprised the 100 node version of this experiment, seen in Figure 2.16. Somewhat surprisingly, the correlation coefficients became higher when the number of nodes was increased. This is evident from Table 2.5. So, for the larger 100 node network, we can compare performance of the proposed Shrink(α) schemes using the mean values of the experimental trials, as reflected in the curves of Figure 2.16.

From Figure 2.16 we see that pure AODV yields better PDF than AODV+Shrink(α), in all mobility settings. Achieving lower PDF with the proposed *Shrink(α)* schemes is understandable, since they increase contention levels at the link layer due to their

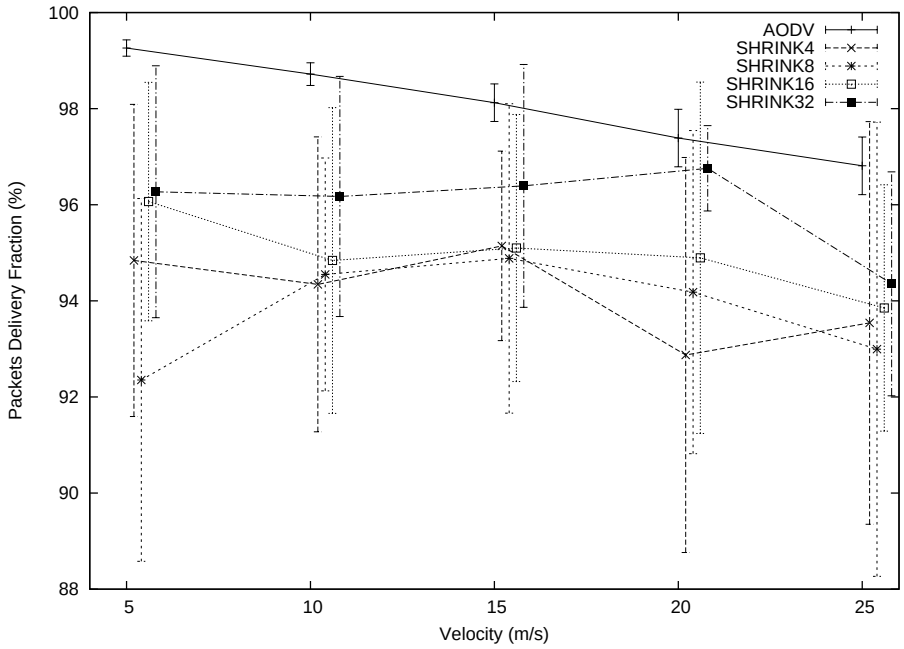


Figure 2.14: PDF vs. velocity (n=50, c=10)

Table 2.4: Correlation coefficients for curves in Figure 2.14.

v	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.193	-0.312	-0.538	0.705
SHRINK4	-	1	-0.661	0.147	0.258
SHRINK8	-	-	1	0.176	-0.266
SHRINK16	-	-	-	1	0.008
SHRINK32	-	-	-	-	1

additional control traffic, and this results in more packet collisions and drops. For the same reason, the Shrink schemes which produce relatively fewer shrink operations (e.g. SHRINK32) have a higher PDF than the Shrink schemes which apply more frequent shrink operations (e.g. SHRINK4). It can be also deduced from Figure 2.16 that there is decreasing trend in PDF as mobility level increases. This is because high mobility increases the number of dropped packets, since link failures become more frequent.

Figure 2.15 shows relationship between PDF and network size, indicating a decrease

Table 2.5: Correlation coefficients for curves in Figure 2.16.

vel = 15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.870	0.909	0.965	0.772
SHRINK4	-	1	0.749	0.839	0.871
SHRINK8	-	-	1	0.982	0.778
SHRINK16	-	-	-	1	0.792
SHRINK32	-	-	-	-	1

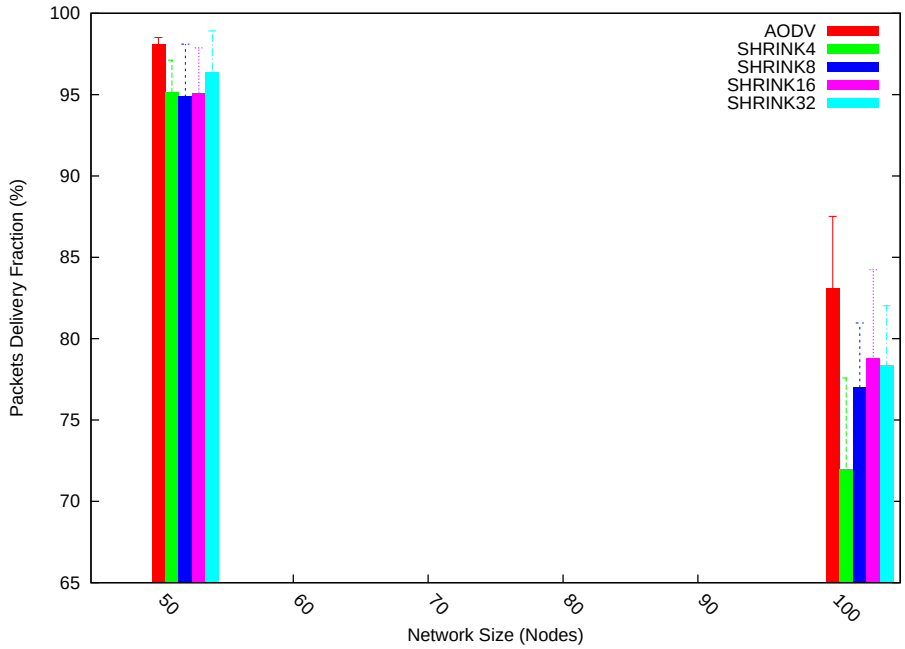


Figure 2.15: PDF vs. network size (v=15, c=10)

in PDF for all proposed schemes, as network size increases. This is explicable by the fact that normalized routing load grows as network size increases. Additionally, there is more contention in large networks, which makes packet losses more likely.

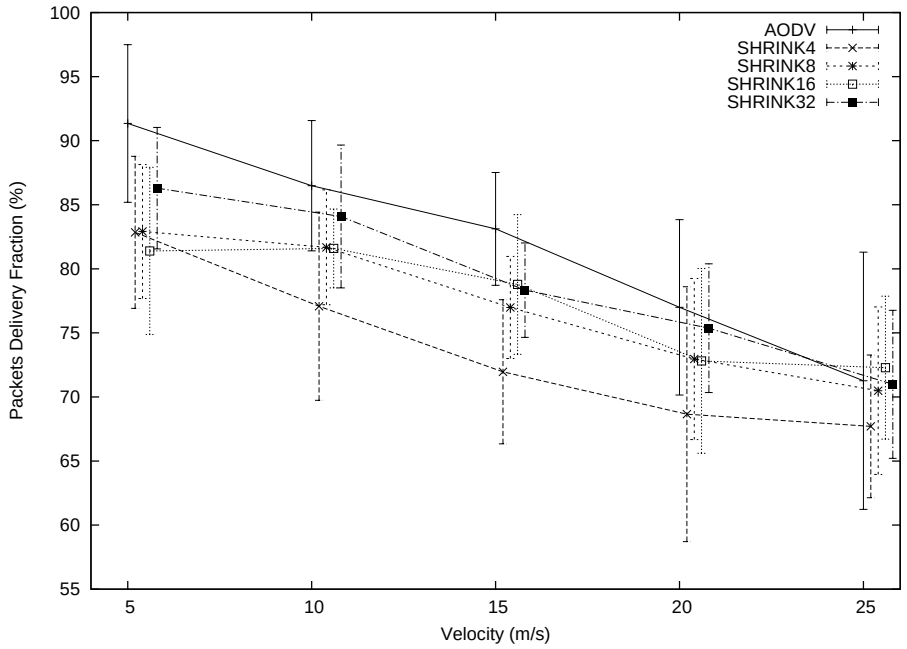


Figure 2.16: PDF vs. velocity (n=100, c=10)

Average end-to-end delay (E2E):

In this set of experiments, we measured average end-to-end delay for delivery of data packets to their destinations. Figure 2.17 depicts average end-to-end delay based on 10 trials, in which each trial consist of approximately 50,000 data packets delivered to 10 different destinations in a 50-node network. According to the figure, pure AODV has end-to-end delay between 0.033 sec and 0.082 sec, linearly increasing with mobility level. The AODV+Shrink mechanism shows different responses depending on version α of the Shrink scheme. For example, SHRINK4 yields the longest delay ranging from 0.049 to 0.089, linearly increasing with mobility level. The other three versions of the Shrink scheme (i.e. SHRINK-8, 16, 32) have relatively lower average delay, varying between 0.027 and 0.068, depending on both mobility level and version of the scheme.

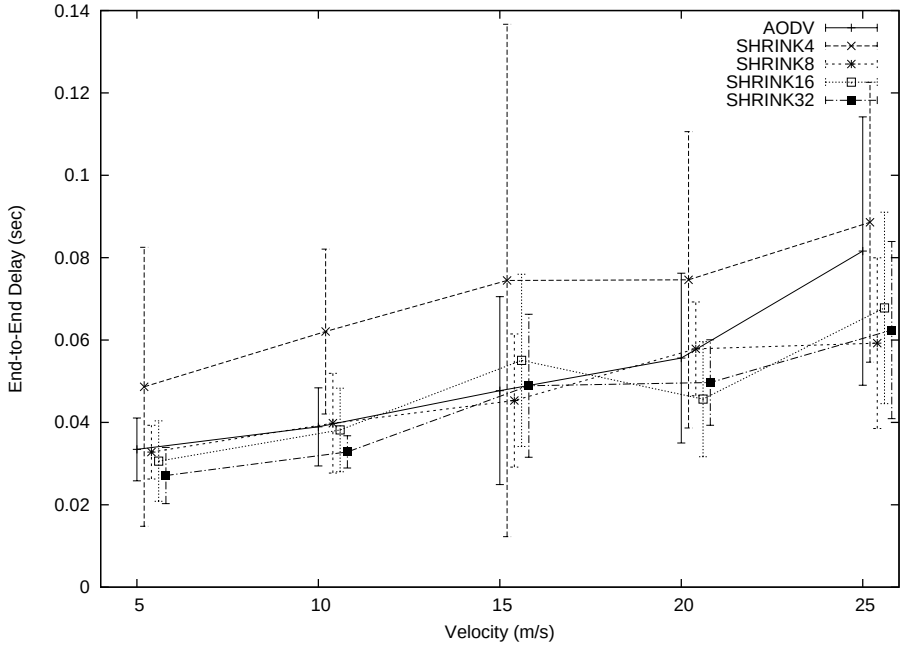


Figure 2.17: E2E vs. velocity (n=150, c=10)

The correlation coefficients for the trials underlying Figure 2.17, when maximum velocity is set to 15 m/s, are shown in Table 2.6. Notice that these coefficients are not close to 1, indicating a weak correlation between the performance of trials underlying the different curves. As a result, the performance comparison of our schemes cannot be conducted by comparing the curves themselves. When we conducted the 100-node version of this experiment, we obtained the graph in Figure 2.19. Unfortunately, even in these larger experiments, the correlation coefficients continued to be small (see table 2.6). Comparisons of end-to-end delay are inconclusive.

Table 2.6: Correlation coefficients for curves in Figure 2.17.

vel = 15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.377	0.419	0.501	0.866
SHRINK4	-	1	0.423	0.432	0.135
SHRINK8	-	-	1	0.020	0.267
SHRINK16	-	-	-	1	0.227
SHRINK32	-	-	-	-	1

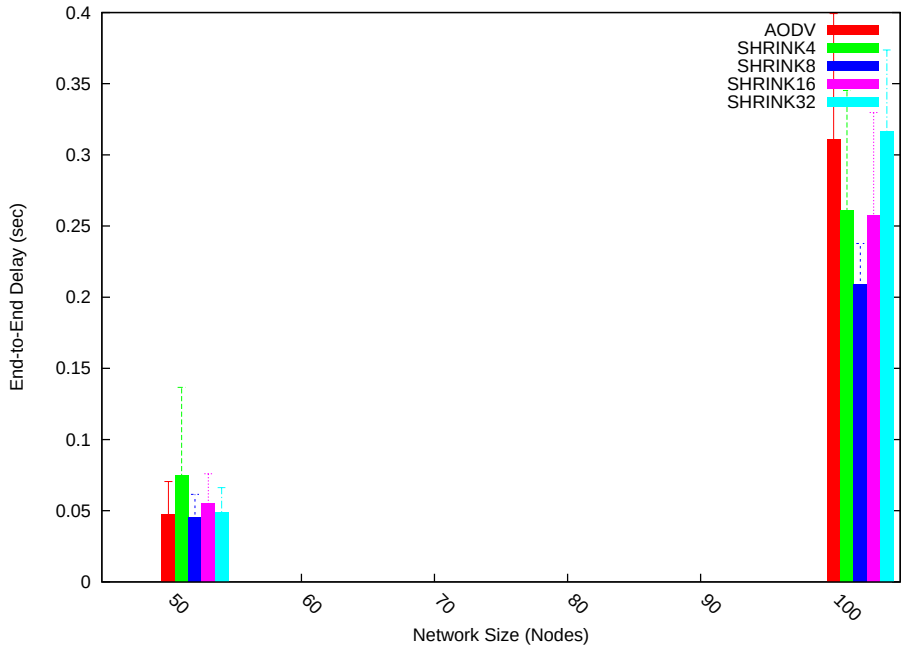


Figure 2.18: E2E vs. network size (v=15, c=10)

Figure 2.18 indicates relationship between end-to-end delay and network size, when maximum velocity is 15 m/s. It can be seen from the figure, that average end-to-end delay rises as network size gets larger. This is explicable since the expected path length in larger networks is greater than it is in small networks, and so it requires more hops to be traveled for each packet delivery.

Average Connection Lifetime (ACL):

In this set of experiments, we measured average lifetime of active routes. Figure 2.20 depicts average connection lifetime for pure AODV and AODV+Shrink schemes at different mobility levels, for networks of 50 nodes with 10 traffic connections. According to the figure, average connection lifetime is inversely proportional to mobility level. For example, average connection lifetime for pure AODV is about 30.13 seconds, 16.61 seconds, 11.48 seconds, 8.82 seconds, and 7.40 seconds when maximum

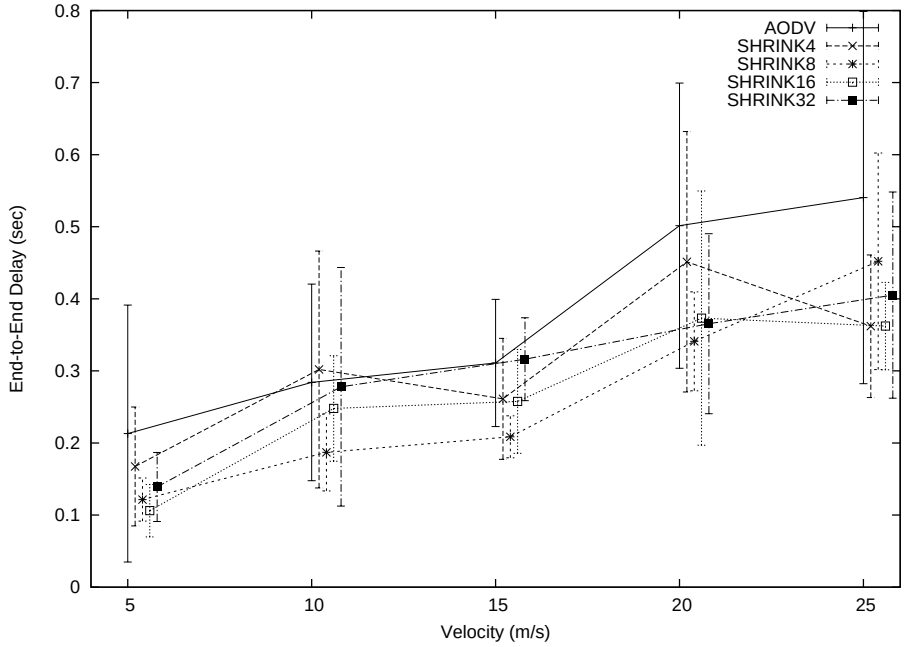


Figure 2.19: E2E vs. velocity (n=100, c=10)

velocity is 5 m/s, 10 m/s, 15 m/s, 20 m/s, and 25 m/s respectively. The same decreasing patterns can also be seen for the curves in the figure, which correspond to our proposed schemes.

Since the error bars in Figure 2.20 overlap each other, we calculated correlation coefficients for experiments underlying each pair of schemes. The subset of correlations for when maximum velocity is 15 m/s are listed in Table 2.7. From the table, we see that coefficients are between 0.75 and 0.88. We took this as a significant correlation between the performance of underlying trials, and compared the schemes on the basis of the mean values of their experiment trials.

It is obvious from Figure 2.20 that average connection lifetime increases, when the Shrink scheme is used (compared to AODV), regardless of version α of the proposed scheme. This is because a link failure is less likely on shorter routes.

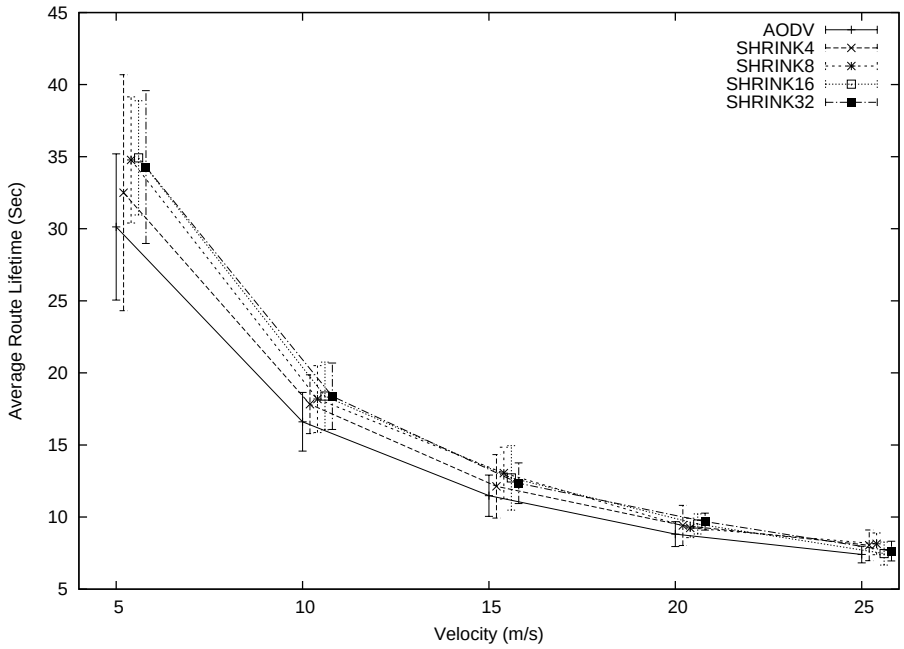


Figure 2.20: ACL vs. velocity (n=50, c=10)

Table 2.7: Correlation coefficients for curves in Figure 2.20.

vel = 15 m/s	AODV	SHRINK4	SHRINK8	SHRINK16	SHRINK32
AODV	1	0.884	0.698	0.784	0.881
SHRINK4	-	1	0.761	0.743	0.836
SHRINK8	-	-	1	0.760	0.753
SHRINK16	-	-	-	1	0.797
SHRINK32	-	-	-	-	1

Figure 2.21 shows relationship between average connection lifetime and network size, when maximum velocity is 15 m/s. It can be inferred from the figure that average connection lifetime decreases as network size increases. This is explicable since average route length grows as network size increases, and this makes routes more sensitive to mobility.

Figure 2.22 describes the average lifetime of routes at different mobility levels, for a network of 100 nodes with 10 data traffic connections. According to the figure, average connection lifetime decreases as mobility level increases. Another observation

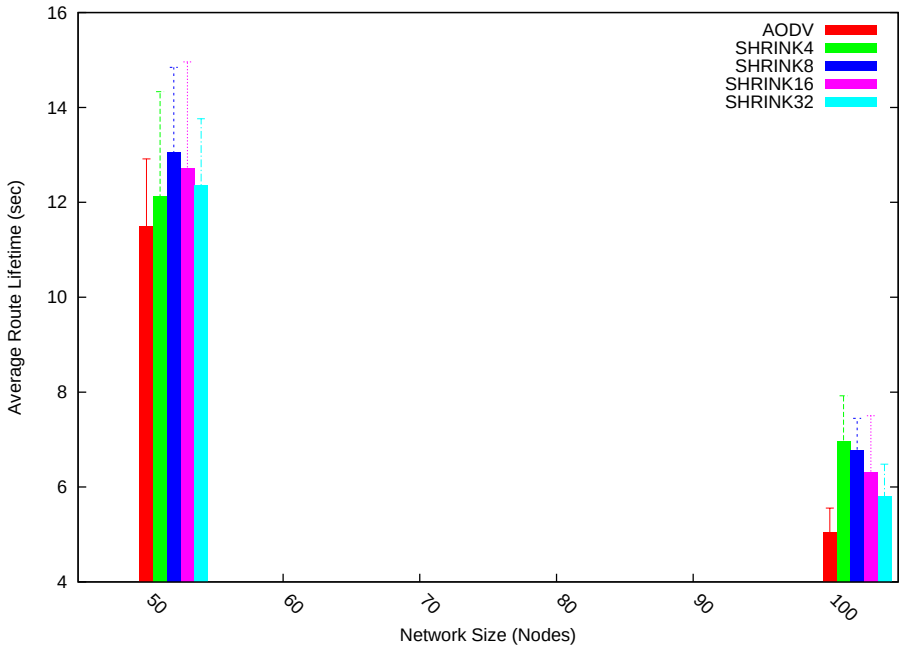


Figure 2.21: ACL vs. network size (v=15, c=10)

is more frequent shrinking operations (e.g. SHRINK4) yields better performance (with respect to average connection lifetime).

2.9.5.2 Multihop Shrinking

In the next subsection, we present and analyze the experimental results of our simulations of the *Multihop Shrinking* scheme. These are considered from the vantage point of the performance metrics mentioned in section 2.9.4. We evaluated four different versions of the proposed *Multihop Shrinking* scheme, i.e. MSHRINK4, MSHRINK8, MSHRINK16, and MSHRINK32, under different mobility levels and in varying network regimes. MSHRINK-α (where α is 4, 8, 16, and 32) is the scheme in which the end-to-end shrink operation is initiated on a given connection by source node after every α data packets have been sent (on that connection). Therefore, MSHRINK4 and

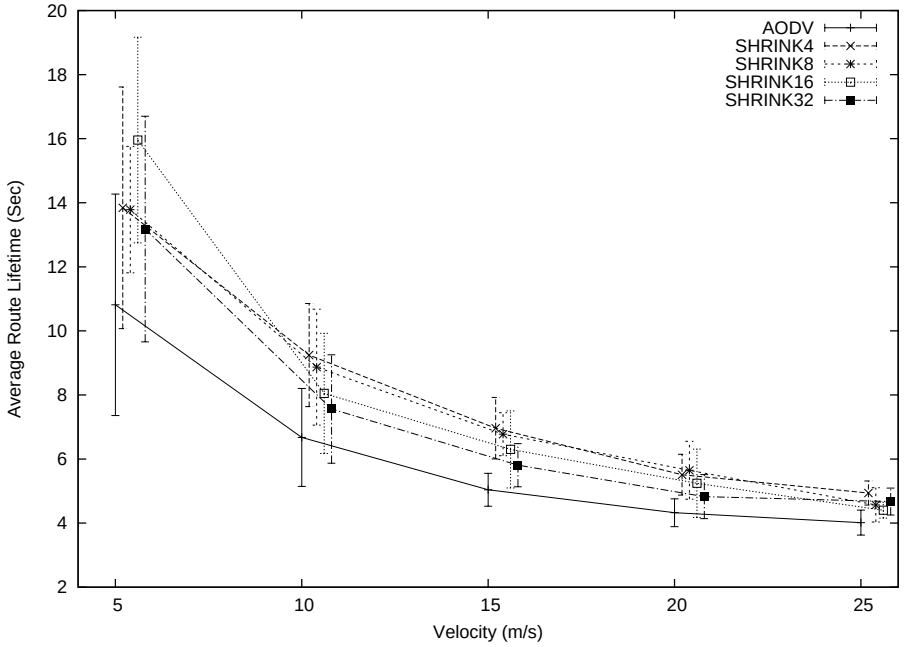


Figure 2.22: ACL vs. velocity (n=100, c=10)

MSHRINK32 represent the most and least frequent shrink operations respectively, while MSHRINK8 and MSHRINK16 are intermediate between them.

Normalized Path Length (NPL):

In this set of experiments, we measured the optimality of the paths for pure AODV and $AODV + MultihopShrinking(\alpha)$ under different mobility levels and in varying network regimes. The NPL metric may be the most important one, as it tests the extent to which the proposed scheme meets the primary objective of optimizing path lengths. The results are the average of 10 trials, each of which consists of about 50,000 data packets processed in a network of 50 nodes to 100 nodes, with 10 data connections. The outcomes are normalized with respect to the optimal path lengths as determined by Dijkstra's shortest path algorithm. We vary the mobility level from maximum velocity of 5 m/s ($\beta = 1$) to 25 m/s ($\beta = 5$).

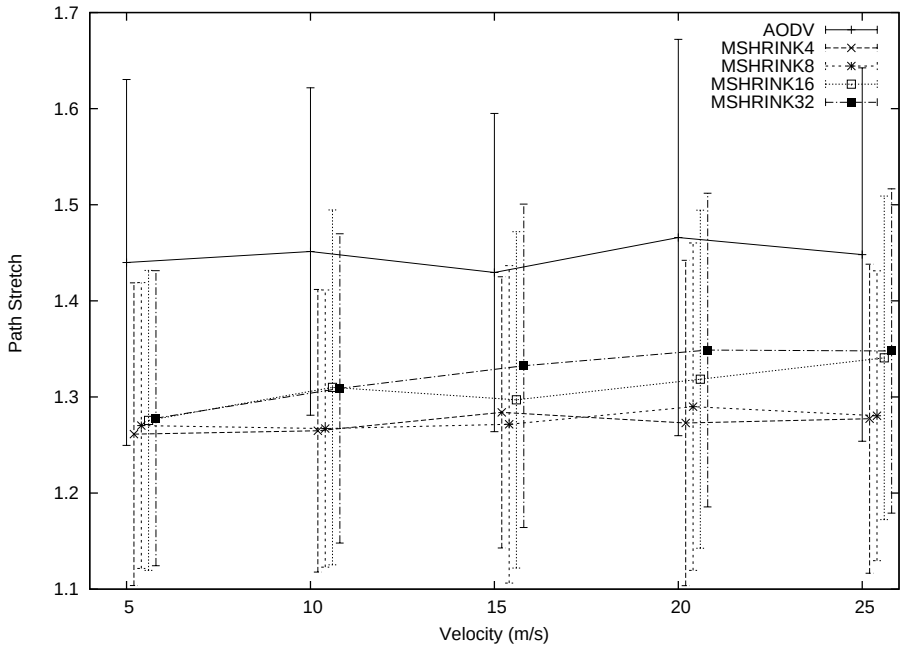


Figure 2.23: NPL vs. velocity (n=50, c=10)

Figure 2.23 depicts the extent to which the *Multihop Shrinking* scheme maintains path optimality at different mobility levels for 50-node network, as quantified by mean normalized average path length. According to the figure, the paths produced by standard AODV are about 45% longer than optimal, regardless of the mobility level. When *Multihop Shrinking* is in effect, however, path lengths are within 34% of optimal.

Since error bars in Figure 2.23 overlap each other, we calculated correlation coefficients between each pair of the curves in order to check whether we can compare schemes based on mean values. Table 2.8 indicates the coefficients when maximum velocity is 15 m/s. In the table, coefficients are seen to be greater than 0.9, indicating significant correlations between experimental trials underlying each curve, and suggesting that the tested mechanisms can be compared based on their average values even though their error bars overlap.

Table 2.8: Correlation coefficients for curves in Figure 2.23.

vel = 15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.924	0.960	0.943	0.881
MSHRINK4	-	1	0.966	0.938	0.906
MSHRINK8	-	-	1	0.960	0.930
MSHRINK16	-	-	-	1	0.916
MSHRINK32	-	-	-	-	1

The most significant result which can be inferred from Figure 2.23 is that *the proposed mechanism can reduce path lengths.* Another observation from the graph and table is that shrinking operations running more frequently (e.g. MSHRINK4) yield better results in terms of path optimality, especially at high mobility levels. This is explicable, since it is less likely that we miss a shortcut if shrinking operations are done more frequently. It can be also seen from the figure that, for schemes MSHRINK16 and MSHRINK32, normalized path lengths increase slightly as mobility level increases. This is explained by noting that decisions made by the shrinking operation yield benefits for ever shorter time intervals when mobility increases. For schemes MSHRINK4 and MSHRINK16, normalized path lengths are almost flat as a function of mobility level. This is because their shrink frequency is so high (with respect to velocity range examined) that mobility levels do not impact their path optimality performance. Pure AODV also has a flat curve with respect to mobility level, because (i) AODV reconstructs routes more frequently as mobility levels increase due to more frequent link breakages and subsequent recovery attempts, and (ii) paths tend to be near-optimal at the time of their construction. Another significant pattern that curves in Figure 2.23 show is that versions of the proposed scheme converge to a path stretch of 1.26 at 5 m/s, and get gradually more separated from each other as mobility level increases. This implies that excessive shrinking operations in low mobility settings (e.g. MSHRINK4 in 5 m/s settings) does not provide additional improvement in path

optimality. Thus, there appears to be a lower bound (> 1) on path stretch that can be achieved by the proposed *Multihop Shrinking* scheme.

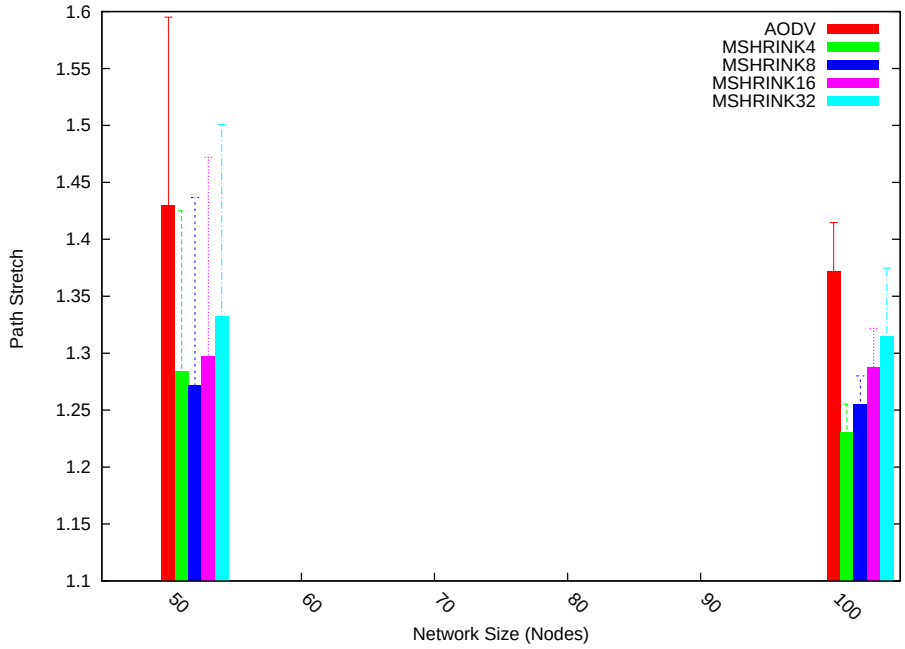


Figure 2.24: NPL vs. network size (v=15, c=10)

From another perspective, Figure 2.24 shows the relationship between normalized path length and network size, when maximum velocity is 15 m/s and traffic load is 10 connections. According to the figure, the proposed *Multihop Shrinking* scheme continues to perform well in larger networks. Moreover, it is seen from the figure that the differences between versions of the proposed scheme become more significant in the larger 100-node network. Figure 2.25 indicates path optimality at different mobility levels, for a network of 100 nodes with 10 data connections.

Normalized Energy Consumption (NEN):

In this set of experiments, we measured the average energy required to deliver data packets to corresponding destinations, referred to hereafter as normalized energy

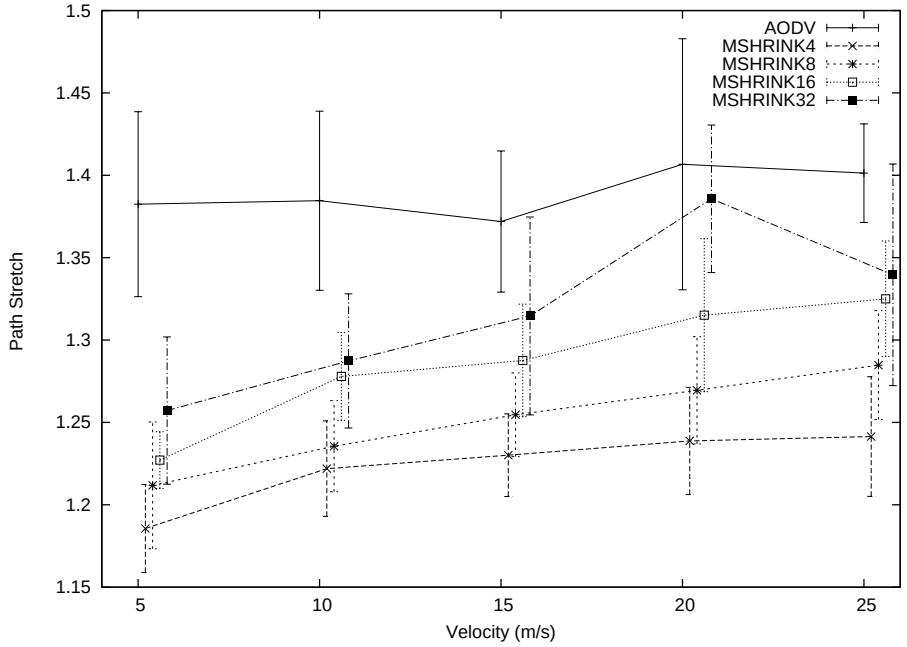


Figure 2.25: NPL vs. velocity (n=100, c=10)

(NEN). Figure 2.26 shows average energy consumption per data packet at different mobility levels, for a network of 50 nodes with 10 data connections. According to the figure, pure AODV requires, on average, between 1425 units and 1590 units of energy for each data packet delivered to its destination, with requirements rising linearly as mobility level increases. On the other hand, when the proposed *Multihop Shrinking* scheme is in effect, the average energy required for this task varies between 1280 units and 1500 units, depending on both version of the proposed scheme and the mobility level.

The error bars in Figure 2.26 overlap each other, so we check correlations to determine whether we can use average values and compare the curves. The correlation coefficients for experiments underlying each pair of curves in the figure as shown in Table 2.2, for the case when maximum velocity was 15 m/s. As seen from the table, correlation coefficients are all bigger than 0.841, indicating significant amount of correlation

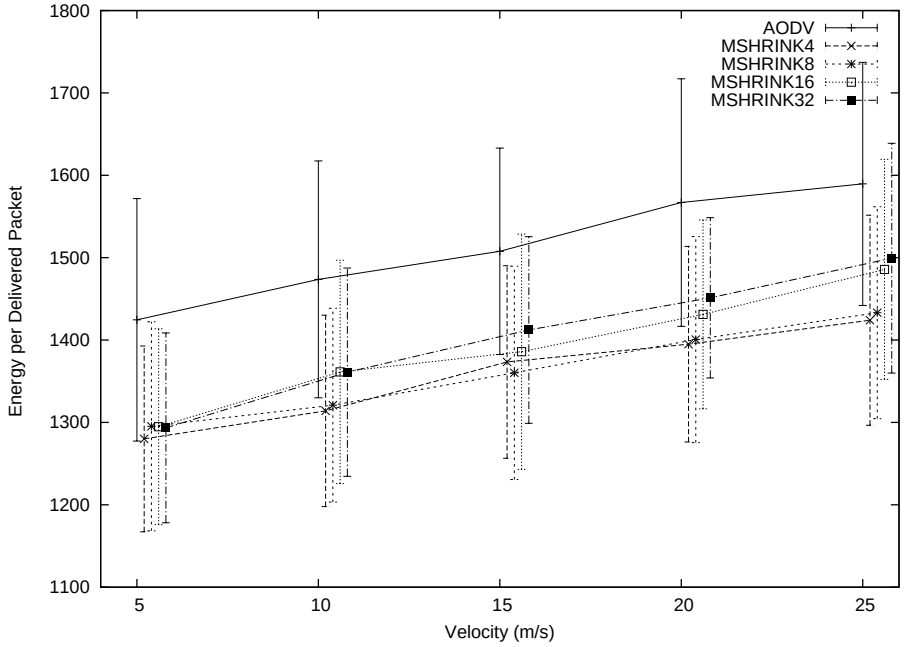


Figure 2.26: NEN vs. velocity (n=50,c=10)

Table 2.9: Correlation coefficients for curves in Figure 2.26.

vel =15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.841	0.931	0.919	0.860
MSHRINK4	-	1	0.996	0.863	0.845
MSHRINK8	-	-	1	0.912	0.894
MSHRINK16	-	-	-	1	0.882
MSHRINK32	-	-	-	-	1

between schemes, and suggesting that the experiments can be compared based on average values at each parameter setting, as reflected by the curves themselves.

It is clear from Figure 2.26 that the proposed schemes reduce the energy required to deliver a data packet to its destination. This is because data packets follow shorter routes when the proposed dynamic route optimization scheme is in effect, and so fewer data transmissions are required in order to deliver the data packets to their destinations. Another interesting observation from the figure is that shrinking schemes which

run more frequently (e.g. MSHRINK4) yield better results in terms of energy consumption regardless of mobility level, in spite of the fact that they spend much more energy executing shrink operations. This is because the versions of the proposed *Multihop Shrinking* scheme which run more frequently (e.g. MSHRINK4) increase path optimality much more than the versions which run less frequently, and this doubly reduces energy consumption for both data and shrink operations. On the other hand, we see from the figure that there is a rise in energy consumption as mobility level increases. This is due to increases in control traffic overhead at higher speed settings, where link failures are more common.

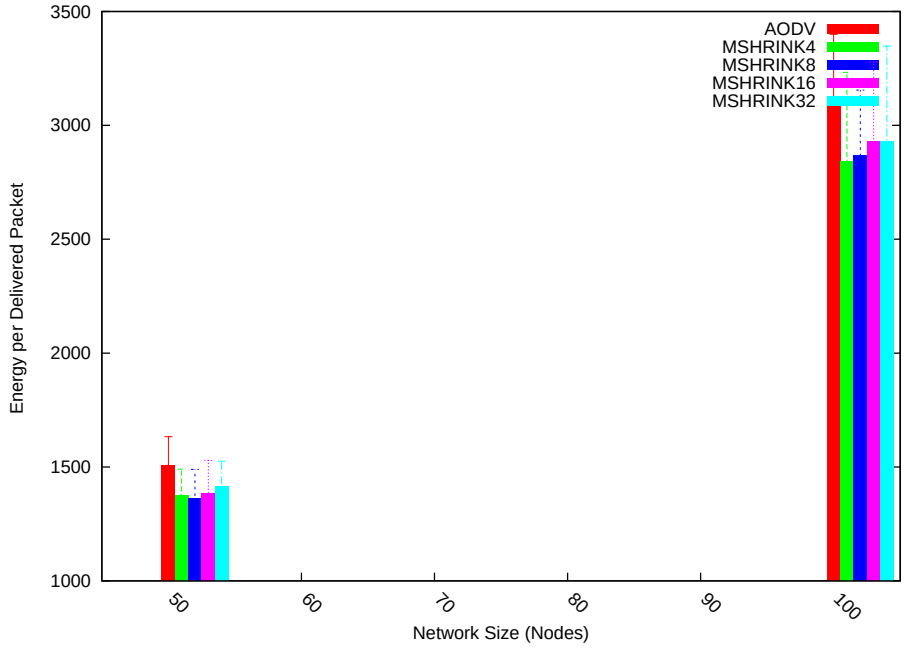


Figure 2.27: NEN vs. network size (v=15, c=10)

Figure 2.27 illustrates that energy consumption per packet increases when network size increases. This is because (i) expected path length in large networks is longer than it is in small networks, so data packets follow longer routes which require more transmissions and hence energy consumption, and (ii) control traffic overhead becomes higher in large networks, since route discovery is more expensive. Figure 2.28 is the

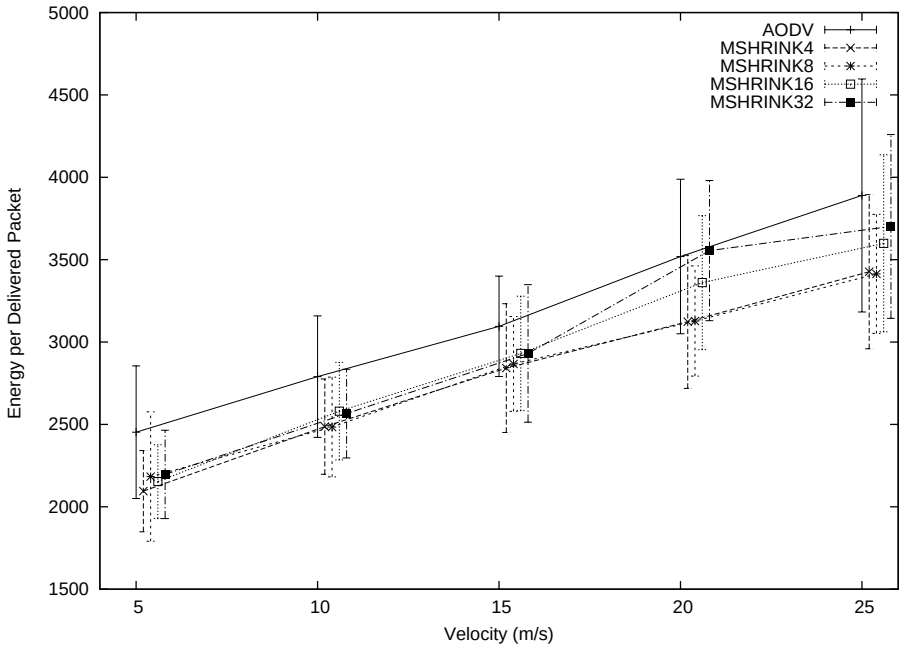


Figure 2.28: NEN vs. velocity (n=100, c=10)

100-node version of Figure 2.26, and exhibits similar trends.

Normalized Routing Load (NRL):

In this set of experiments, we examined the control traffic load incurred by the proposed *Multihop Shrinking* scheme, normalized by the number of packets delivered. Figure 2.29 indicates normalized routing load of both pure AODV and $AODV + MultihopShrinking$ mechanisms at different mobility levels. According to the figure, NRL increases proportionally with mobility level for all mechanisms evaluated. For example, AODV has NRL of 0.43, 0.78, 1.13, 1.47, and 1.75 at maximum velocity of 5 m/s, 10 m/s, 15 m/s, 20 m/s, and 25 m/s respectively. Our schemes exhibit very similar trends. According to Figure 2.29, MSHRINK16 and MSHRINK32 generally have lower NRL, MSHRINK4 has higher NRL values than AODV. It is interesting that the proposed *Multihop Shrinking* scheme (for versions of MSHRINK16 and MSHRINK32)

have control traffic overhead that is comparable to AODV, in spite of the fact that they introduce extra control traffic for route optimization! The reason lower NRL is witnessed for these schemes is that link failures are rarer on shorter (optimized) routes than on longer ones. More specifically, since the proposed schemes reduce the path length, they yield longer connection lifetimes and fewer route discovery attempts–the latter events being extremely expensive in terms of control traffic incurred. On the other hand, because SHRINK4 produces much more control traffic, it does not reduce overall NRL.

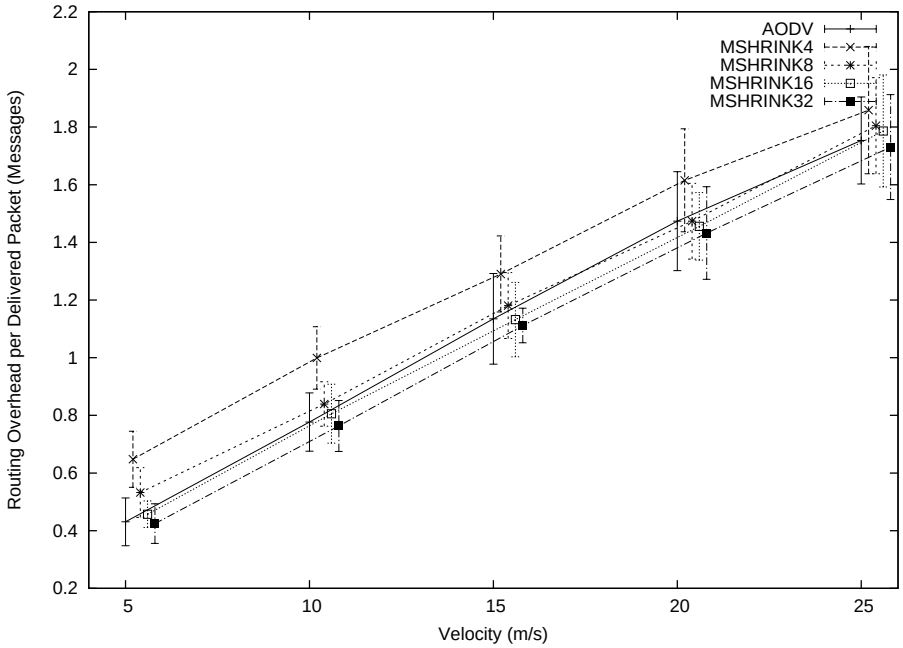


Figure 2.29: NRL vs. velocity (n=50, c=10)

Table 2.10 includes the correlation coefficients corresponding each pair of curves in Figure 2.29, when maximum velocity is 15 m/s. According to the table, the coefficients are quite varied between 0.494 and 0.908 (ignoring auto correlations, whose values are of course 1). These weak correlations indicate that most schemes are incomparable against each other. On the other hand, the next Table 2.11 indicates the correlation coefficients for each pair of curves in Figure 2.31, when maximum velocity is 15

Table 2.10: Correlation coefficients for curves in Figure 2.29.

vel = 15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.666	0.908	0.783	0.740
MSHRINK4	-	1	0.792	0.679	0.494
MSHRINK8	-	-	1	0.743	0.586
MSHRINK16	-	-	-	1	0.742
MSHRINK32	-	-	-	-	1

Table 2.11: Correlation coefficients for curves in Figure 2.31.

vel = 15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.739	0.941	0.838	0.911
MSHRINK4	-	1	0.895	0.904	0.884
MSHRINK8	-	-	1	0.945	0.975
MSHRINK16	-	-	-	1	0.988
MSHRINK32	-	-	-	-	1

m/s. Since the coefficients in this second table are close to 1, the experiments in the corresponding graph can be compared based on average values at each parameter setting.

Figure 2.30 depicts relationship between NRL and network size. Interestingly, it is seen in the figure that all versions of the proposed *Multihop Shrinking* scheme show better performance than pure AODV in terms of NRL in larger networks. For example, in 100-node network, AODV has the highest NRL value, as the *Multihop Shrinking* scheme has relatively lower NRL values. This can be explained by the fact that route optimization reduces the expected number of route discovery attempts, and these are expensive in large networks. Figure 2.31 presents detailed information regarding NRL in a 100-node network at different mobility levels.

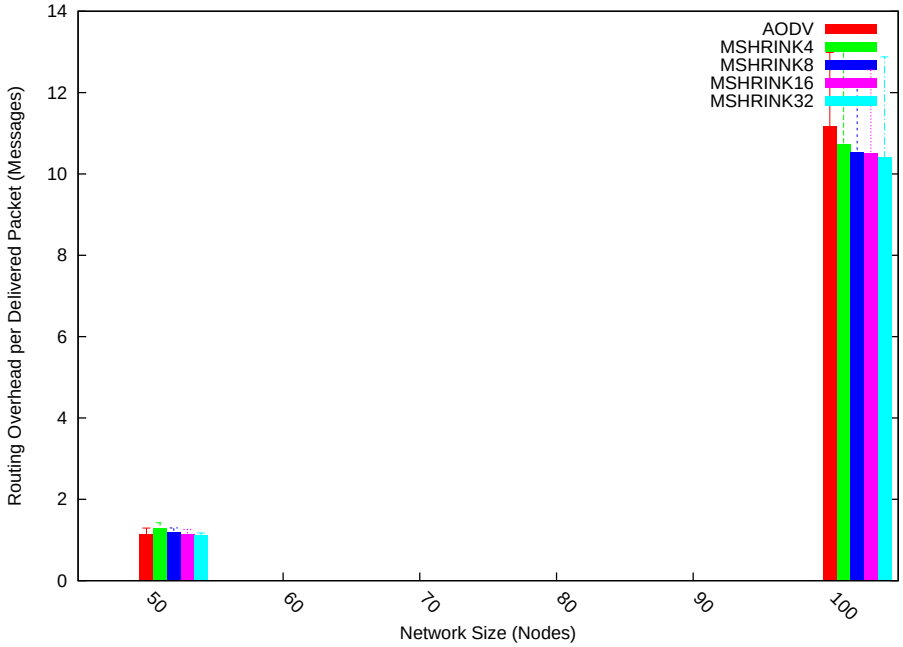


Figure 2.30: NRL vs. network size (v=15,c=10)

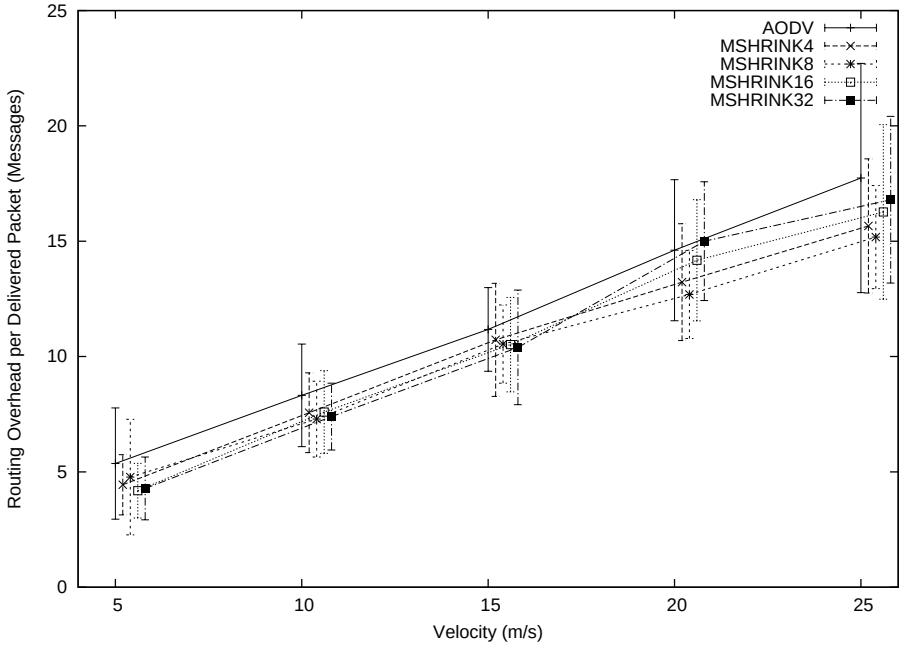


Figure 2.31: NRL vs. velocity (n=100, c=10)

Packet Delivery Fraction (PDF):

In this set of experiments, we measured packet delivery fraction (PDF) for pure AODV and $AODV + MultihopShrinking(\alpha)$ under different mobility levels and network regimes. Figure 2.32 shows PDF for both pure AODV and different versions of the proposed shrinking mechanism, in a network of 50 nodes with 10 data connections.

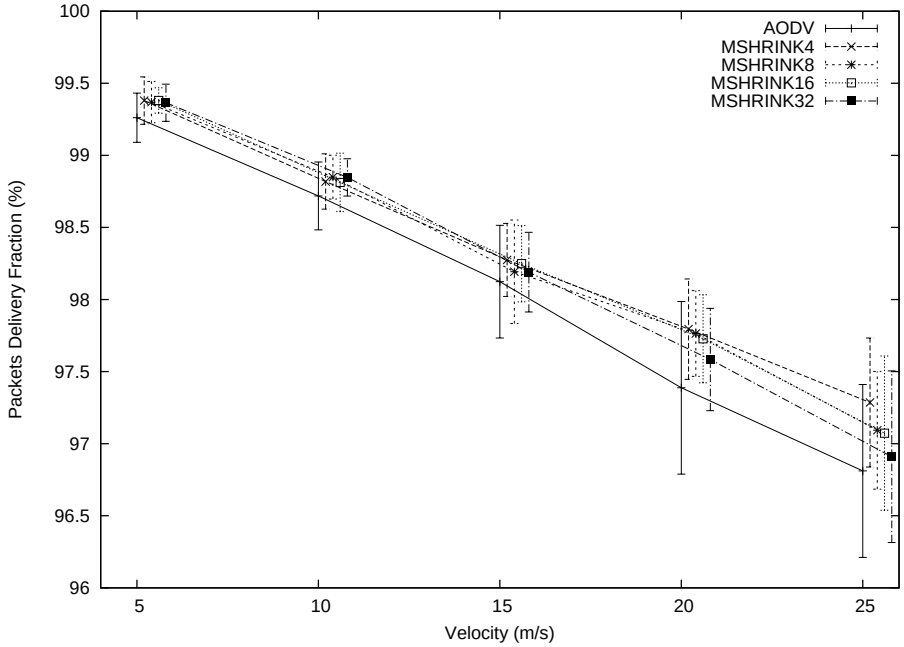


Figure 2.32: PDF vs. velocity (n=50, c=10)

According to the figure, pure AODV has a PDF value between 99.3% and 96.8% , linearly decreasing as mobility level increases. On the other hand, the figure indicates that, when the proposed scheme is in effect, PDF varies between 99.4% and 97.0% depending on both mobility level and version α.

Table 2.12 includes correlation coefficients for the curves in Figure 2.32 when maximum velocity is 15 m/s. Notice that the coefficients in the table are not close to 1. Given these weak correlations, we cannot compare schemes based on mean values

Table 2.12: Correlation coefficients for curves in Figure 2.32.

v = 15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.625	0.745	0.805	0.601
MSHRINK4	-	1	0.512	0.569	0.231
MSHRINK8	-	-	1	0.525	0.160
MSHRINK16	-	-	-	1	0.450
MSHRINK32	-	-	-	-	1

Table 2.13: Correlation coefficients for curves in Figure 2.34.

vel =15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.883	0.930	0.923	0.959
MSHRINK4	-	1	0.911	0.916	0.920
MSHRINK8	-	-	1	0.991	0.994
MSHRINK16	-	-	-	1	0.993
MSHRINK32	-	-	-	-	1

reflected in the curves in Figure 2.32. For this reason, we conducted a larger scale experiment using 100 nodes. Figure 2.34 depicts the results, and Table 2.13 lists the correlation coefficients for the trials underlying the curves. Since the coefficients are very close to 1 for larger networks, the trials underlying the curves are correlated, and we can legitimately compare the schemes based on their mean values, as reflected in the curves within the figure.

It is clear from Figure 2.34 that the proposed schemes increase PDF. For example, when maximum velocity is 25 m/s, pure AODV has a PDF value of 71.3%, but increases to 74.0%, 75.3%, 76.5%, and 76.6% when MSHRINK32, MSHRINK16, MSHRINK8, and MSHRINK4 are applied (respectively). More frequent shrinking (e.g. MSHRINK4) yields better PDF. The increase in PDF is because of the decrease in contention due to shorter routes and lower control traffic overhead (reported in the previous experimental results). An increase in PDF is can be interpreted as a reduction in packet losses. Although several factors may cause packet losses, two of

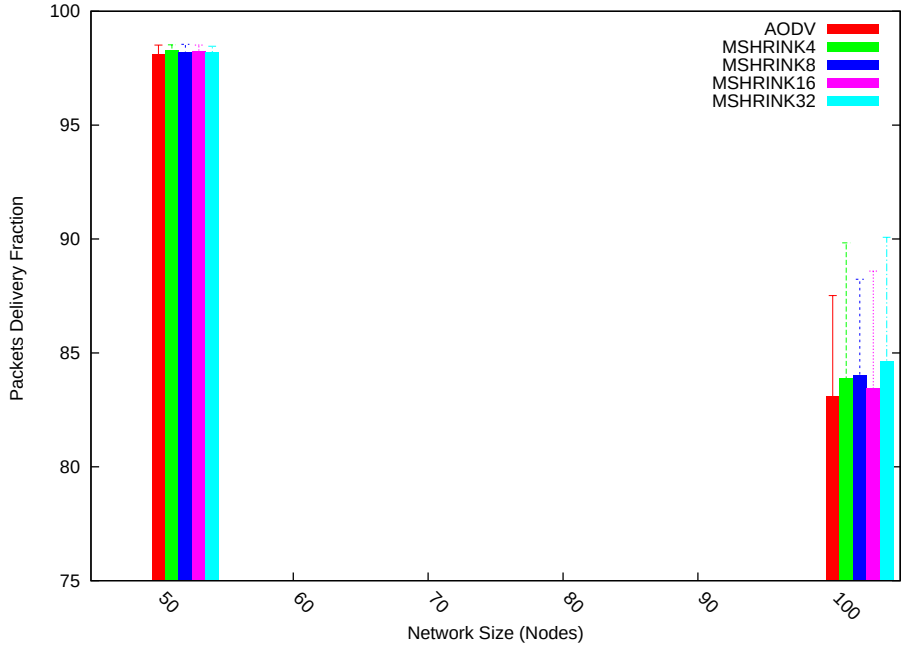


Figure 2.33: PDF vs. network size (v=15, c=10)

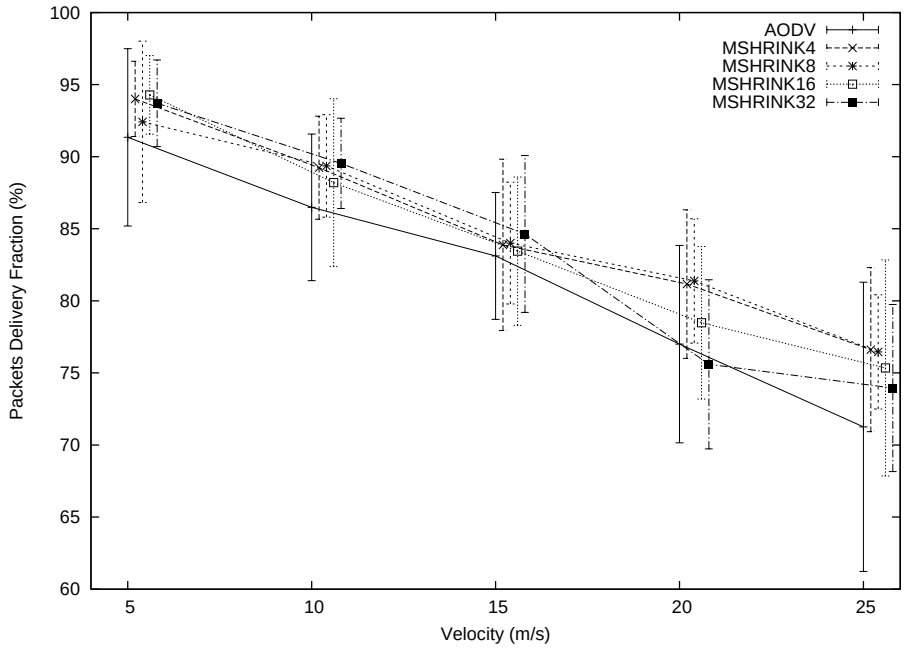


Figure 2.34: PDF vs. velocity (n=100, c=10)

them are directly related to our scheme: (i) a packet may be dropped when there is a collision at the link layer, and (ii) a packet may be dropped if the link it is about to be sent on fails. Regarding (i), shorter connections have fewer hops, and thus produce less contention. Concerning factor (ii), the proposed mechanism decreases the frequency of link breakages in active connections (as we will show in the following subsections). This implies that the number of packets dropped due to link failures is lower. Lastly, as mobility increases PDF decreases (uniformly in all cases), since link breakages are more common at higher mobility levels.

Average End-to-end Delay (E2E):

In this set of experiments, we measured average end-to-end delay of data packets delivered to destinations. Figure 2.35 shows average end-to-end delay based on 10 trials, in which each trial consist of approximately 50,000 data packets delivered to 10 different destinations in a 50-node network. According to the figure, AODV has end-to-end delay between 0.033 sec and 0.082 sec, linearly increasing as mobility level increases. The Multihop Shrinking scheme, in contrast, has end-to-end delay varying between 0.026 and 0.071 depending on both the mobility level and version of the scheme. Since error bars of curves overlap, we compute experiment correlations. Given the low correlation coefficients shown in Table 2.14, we cannot justify comparing the schemes based on their mean performance values.

Figure 2.36 shows relationship between end-to-end delay and network size, when maximum velocity is 15 m/s. The figure shows average end-to-end delay increases with network size. This is because expected path length in larger networks is greater than in small networks; since more hops are traversed, longer delays experienced.

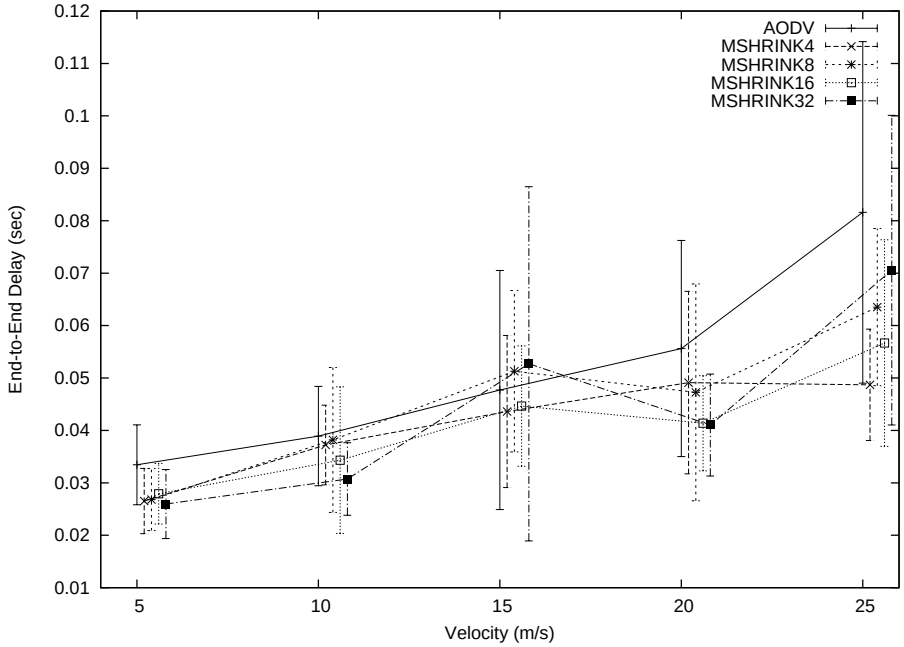


Figure 2.35: E2E vs. velocity (n=50, c=10)

Table 2.14: Correlation coefficients for curves in Figure 2.35.

vel =15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.427	0.680	0.919	0.280
MSHRINK4	-	1	0.450	0.233	-0.071
MSHRINK8	-	-	1	0.670	-0.057
MSHRINK16	-	-	-	1	0.329
MSHRINK32	-	-	-	-	1

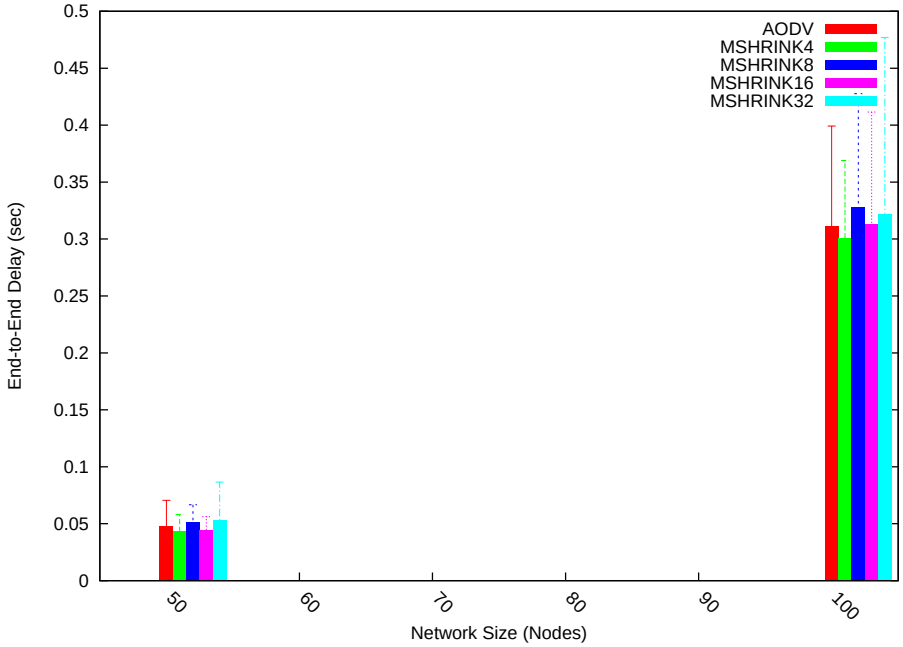


Figure 2.36: E2E vs. network size (v=15, c=10)

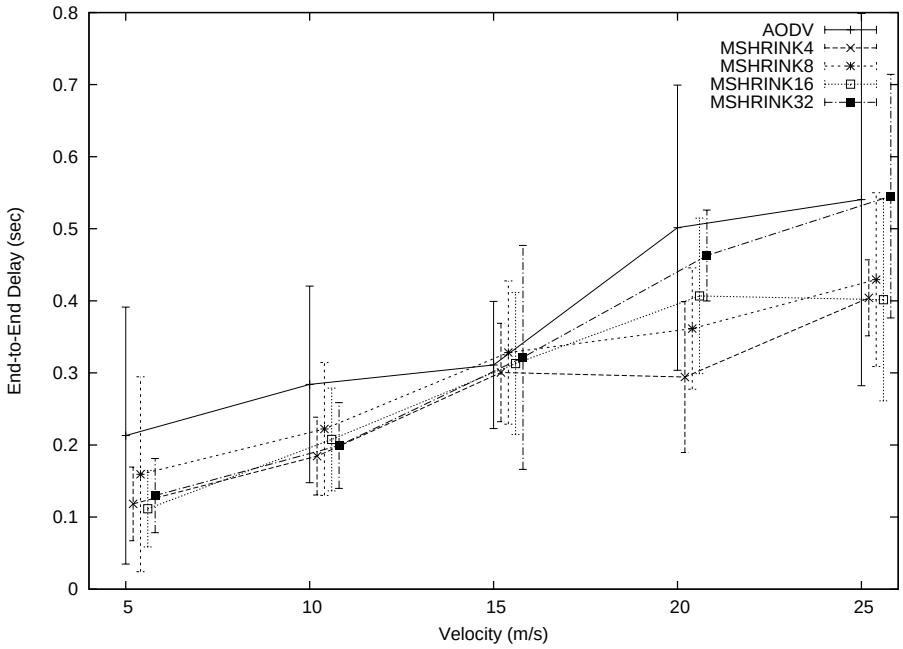


Figure 2.37: E2E vs. velocity (n=100, c=10)

Average Connection Lifetime (ACL):

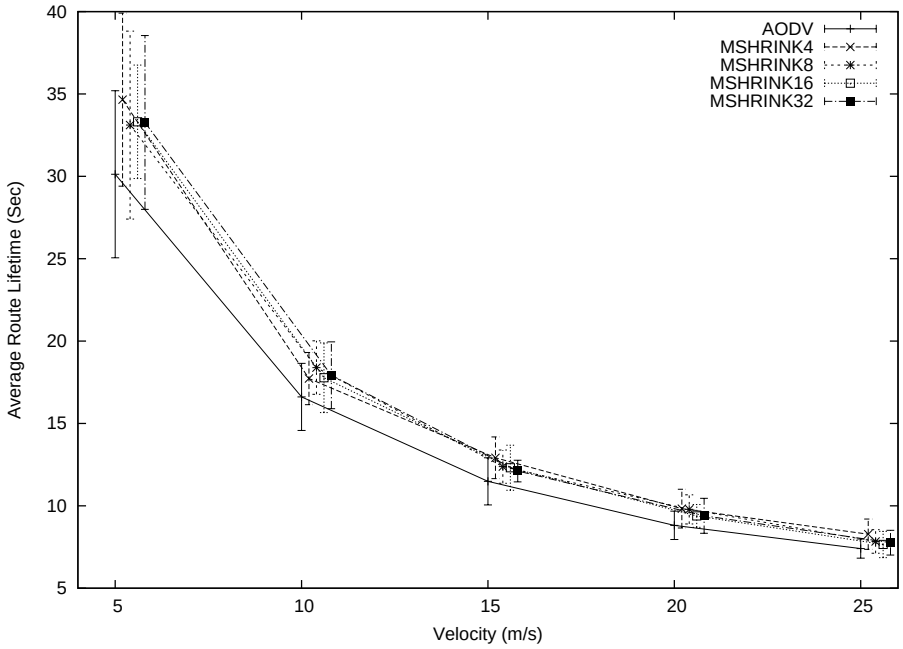


Figure 2.38: ACL versus velocity (n=50, c=10)

In this set of experiments, we measured average lifetime of active connections. Figure 2.38 shows average connection lifetime at different mobility levels, for a network of 50 nodes with 10 data connections. According to the figure, average connection lifetime is inversely proportional to mobility level. For example, average connection lifetime for pure AODV is about 30.13 seconds, 16.61 seconds, 11.48 seconds, 8.82 seconds, and 7.40 seconds at 5 m/s, 10 m/s, 15 m/s, 20 m/s, and 25 m/s respectively. The same trend is also exhibited by our proposed schemes.

Table 2.15 contains correlation coefficients for the curves in Figure 2.38. The minimum value among the coefficients is 0.55, and all others are typically around 0.75 or higher. With some caution, we can compare schemes based on mean values. Figure 2.38 shows that compared to pure AODV, the average lifetime of routes is increased when the proposed *Multihop Shrinking* schemes are used. This is because shrinking

Table 2.15: Correlation coefficients for curves in Figure 2.38.

vel = 15 m/s	AODV	MSHRINK4	MSHRINK8	MSHRINK16	MSHRINK32
AODV	1	0.645	0.880	0.750	0.732
MSHRINK4	-	1	0.698	0.701	0.550
MSHRINK8	-	-	1	0.610	0.587
MSHRINK16	-	-	-	1	0.779
MSHRINK32	-	-	-	-	1

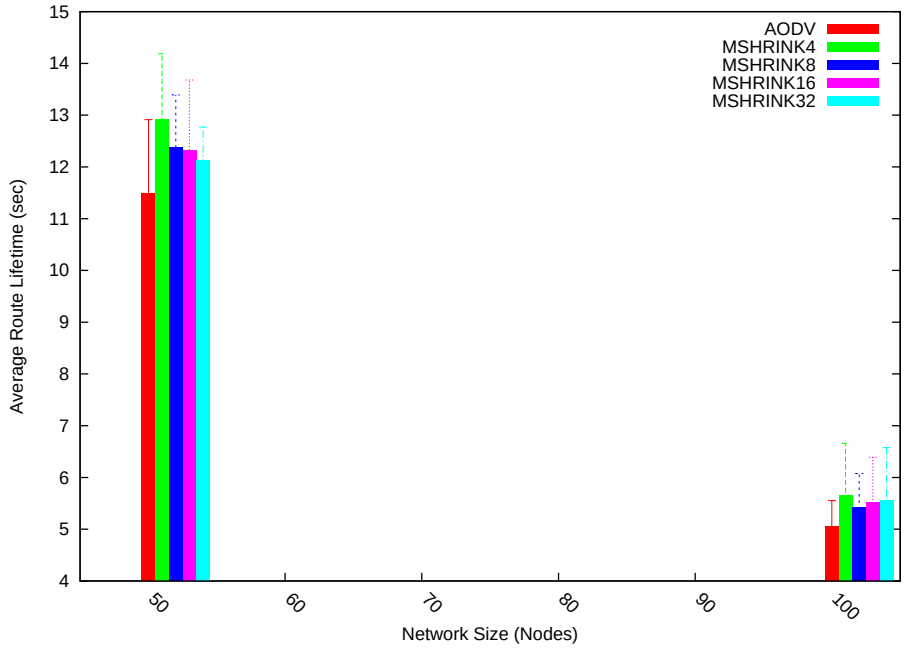


Figure 2.39: ACL vs. network size (v=15,c=10)

realigns the topological structure of the connection with the geometric positions of the nodes. It can also be explained by the fact that it is less likely to experience a link failure on shorter paths than it is on longer paths.

Figure 2.39 shows relationship between average connection lifetime and network size at maximum velocity of 15 m/s. From the figure, we can infer that average connection lifetime decreases as network size increases. This occurs because average connection length gets longer as network size gets larger, which implies that routes, being longer,

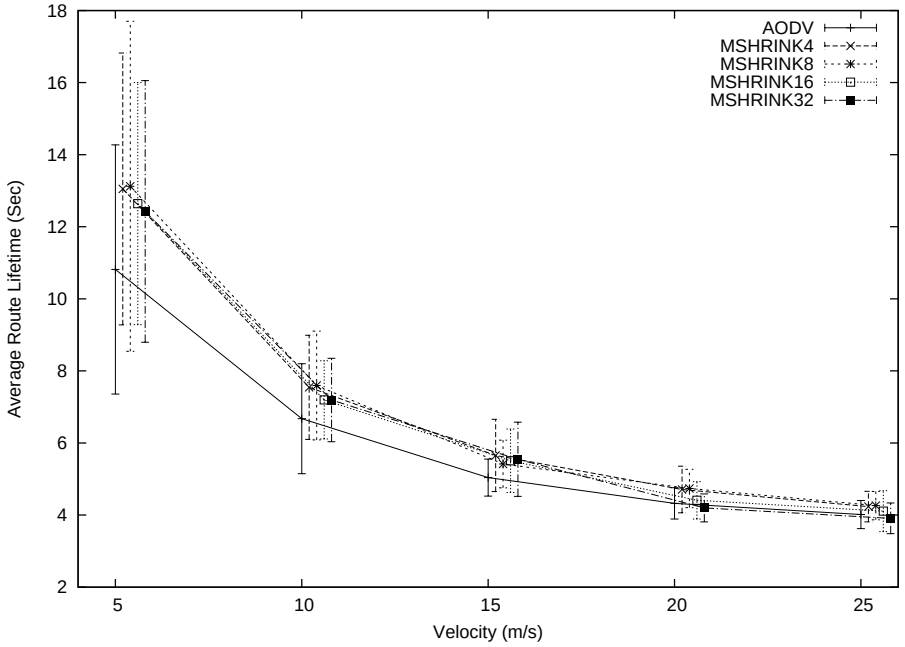


Figure 2.40: ACL versus velocity (n=10, c=10)

are more fragile in the face of mobility within large networks. Figure 2.40 shows average lifetime of routes versus mobility level, for 100-node networks including 10 traffic connections. According to the figure, average connection lifetime decreases with velocity.

2.9.5.3 1-hop Shrink versus Multihop Shrink

In the preceding two subsections, we measured performance of different versions of the proposed (i) *1-hop Shrinking*, and (ii) *Multihop Shrinking* schemes, separately, and compared them to pure AODV under different mobility level settings, and in network regimes, from the point of view of several metrics. In this subsection, we compare performance of 1-hop Shrink and Multihop Shrink against each other (with pure AODV as a reference). Rather than considering all 4 variants of each of the

1-hop Shrink and Multihop Shrink, we report here on only representative version of each, namely SHRINK16 and MSHRINK16.

Normalized Path Length (NPL)

In this set of experiments, we measured the optimality of paths for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16, under different mobility level settings and network regimes. The results are the average of 10 trials, each of which consists of about 50,000 data packets processed in networks including either 50 nodes or 100 nodes, with 10 traffic connections. The outcomes are normalized with respect to the optimal path lengths calculated by Dijkstra's shortest path algorithm. We vary mobility settings from maximum velocity of 5 m/s to 25 m/s.

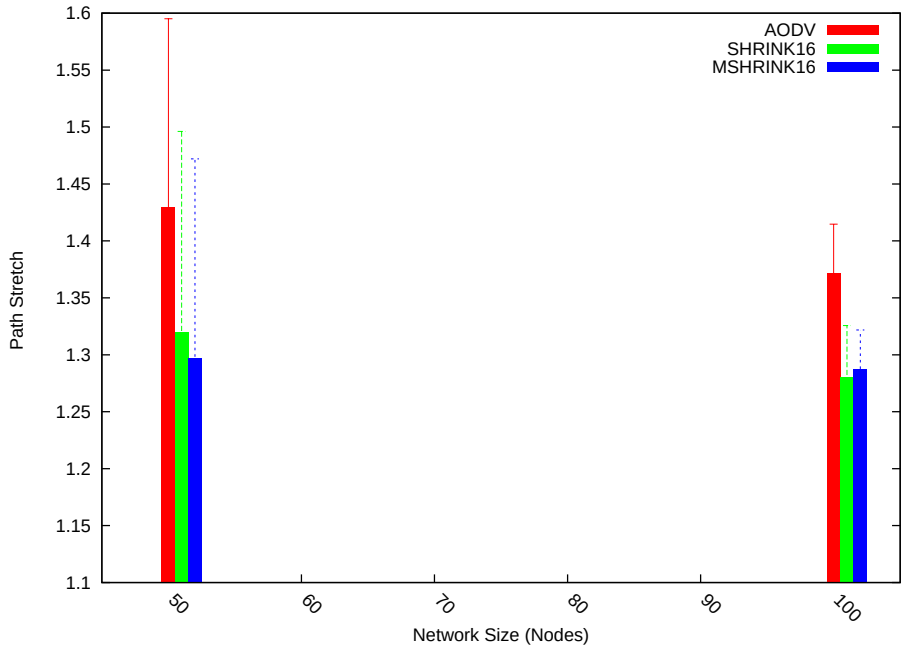


Figure 2.41: v=15, c=10

Figure 2.41 depicts path optimality as network size changes from 50 nodes to 100 nodes while maximum velocity is 15 m/s. Table 2.16 includes correlation coefficients

of trials underlying Figure 2.41. The coefficient relating SHRINK16 and MSHRINK16 in 50-node network is 0.931, while in 100-node network it is 0.806. This indicates significant correlation between experimental results of each trial for these schemes, suggesting that they may be compared based on average values of the experimental trials.

Table 2.16: Correlation coefficients for curves in Figure 2.41.

	50 nodes			100 nodes		
vel =15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.956	0.943	1	0.363	0.826
SHRINK16	-	1	0.931	-	1	0.806
MSHRINK16	-	-	1	-	-	1

According to Figure 2.41, both SHRINK16 and MSHRINK16 schemes improve path optimality with respect to pure AODV. It seems that MSHRINK16 outperforms SHRINK16 marginally in a 50-node network, but it is the other way round in 100-node network. The difference between the performance of SHRINK16 and MSHRINK16 is very small, and it is difficult to ascertain whether this effect is significant enough to require explanation.

Figure 2.42 shows the normalized path length at different mobility levels, for a network of 100 nodes with 10 traffic connections. According to the figure, both the proposed SHRINK16 and MSHRINK16 schemes have lower path stretch than pure AODV at all mobility levels. Nevertheless, the schemes approach the performance of pure AODV as mobility level increases. In conclusion, asymptotically (for high enough mobility) there is not much difference between performance of SHRINK16 and MSHRINK16 schemes from the point of path optimality. This is expected since at extremely high levels of mobility all path shortening operations are obsolete by the time they are completed.

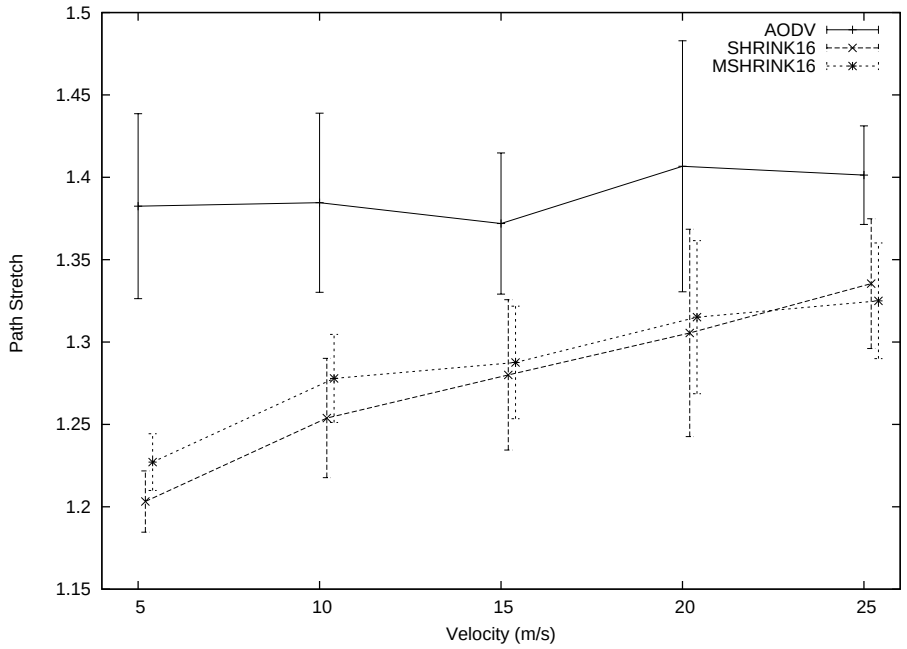


Figure 2.42: n=100, c=10

Normalized Energy Consumption (NEN):

In this set of experiments, we compare performance of SHRINK16 and MSHRINK16 schemes from the point of view of the average energy required to deliver each data packet to its destination. Figure 2.43 depicts the performance of the schemes in networks having different sizes. According to the figure, pure AODV requires 1508 units energy, on average, to deliver each data packet to its destination in 50-node network. On the other hand, when SHRINK16 (resp. MSHRINK16) are applied, the required energy for the same task, on average, becomes 1437 units (resp. 1386 units). Looking at a larger 100-node network, the average energy consumption per data packet delivery for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 is 3095 units, 2901 units, and 2930 units respectively.

Before interpreting these results, we need to determine whether we can use mean val-

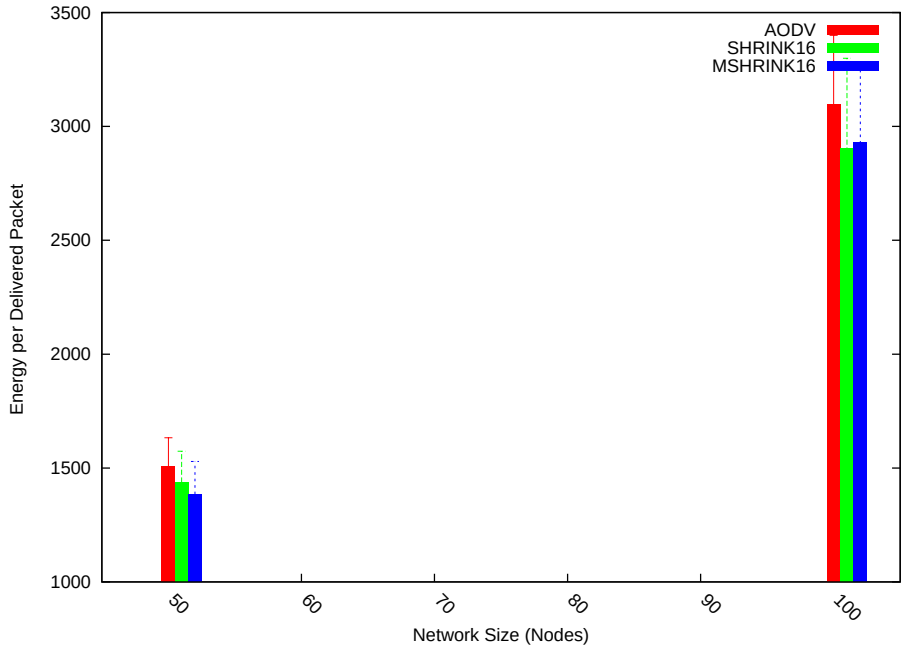


Figure 2.43: v=15, c=10

ues of experimental trials by calculating corresponding correlation coefficients for the trials underlying the curves in the figure. Table 2.17 indicates correlation coefficients for each pair of measured mechanism when maximum velocity is 15 m/s. According to the table, correlation coefficients between SHRINK16 and MSHRINK16 for 50-node and 100-node networks are 0.946 and 0.985 respectively, which suggests these schemes can be compared based on average values of the experimental trials.

Table 2.17: Correlation coefficients for curves in Figure 2.43.

	50 nodes			100 nodes		
vel = 15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.939	0.919	1	0.777	0.867
SHRINK16	-	1	0.946	-	1	0.985
MSHRINK16	-	-	1	-	-	1

Since correlation is significant, it can be deduced from Figure 2.43 that both SHRINK16 and MSHRINK16 schemes reduce energy consumption in comparison to pure AODV. Besides this, MSHRINK16 and SHRINK16 are incomparable in 50-node and 100-node

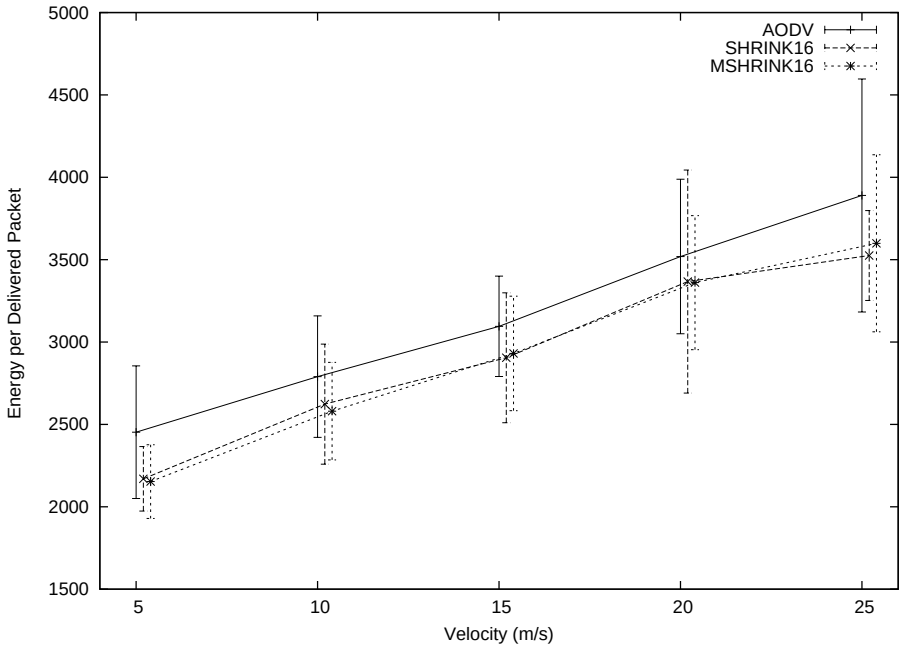


Figure 2.44: n=100, c=10

networks. Figure 2.44 depicts average energy consumption per delivered packet at different mobility levels in a 100-node network. According to the figure, both proposed schemes exhibit very similar performance, but both clearly outperform pure AODV.

Normalized Routing Load (NRL):

In this set of experiments, we compared the traffic load incurred by the proposed SHRINK16 and MSHRINK16 schemes. Figure 2.45 indicates overall routing load, normalized by the number of data packets delivered to destinations, for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 mechanisms in different network sizes.

Table 2.18 contains the correlation coefficients for trials underlying the curves in Figure 2.45. It can be deduced from the tables that the performance of the schemes may be compared using the data from the larger 100-node network experiments, since

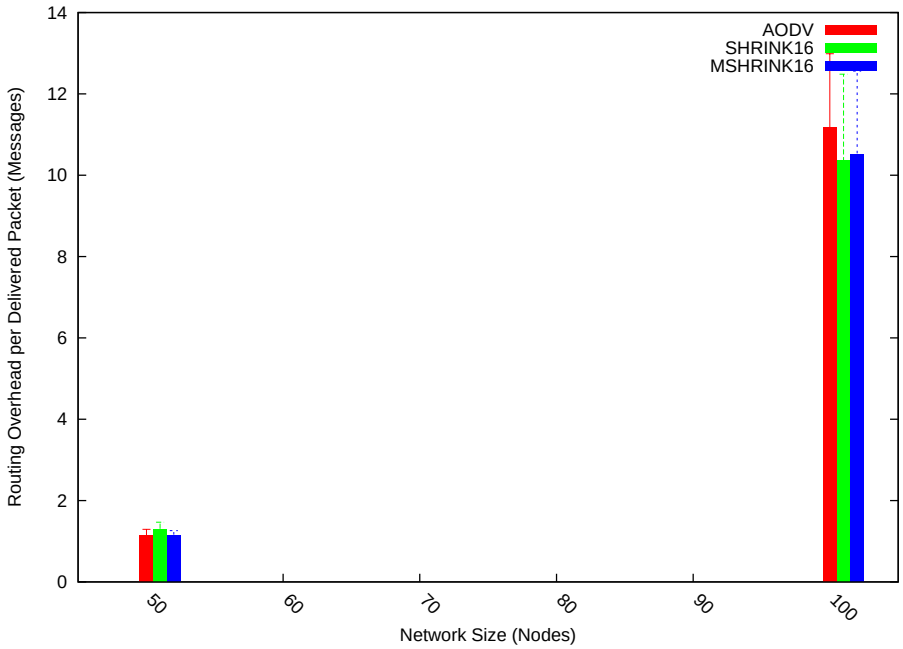


Figure 2.45: v=15, c=10

correlation coefficients between SHRINK16 and MSHRINK16 are 0.967 for 100-node network experiments (but are only 0.760 for 50-node network). Figure 2.46 shows NRL for different mobility levels in 100-node network for the proposed mechanisms. According to the figure, it is very difficult to distinguish which of our schemes outperforms the other, but it is clear that they both surpass pure AODV in terms of NRL, regardless of mobility level.

Table 2.18: Correlation coefficients for curves in Figure 2.45.

	50 nodes			100 nodes		
vel =15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.705	0.783	1	0.765	0.838
SHRINK16	-	1	0.760	-	1	0.967
MSHRINK16	-	-	1	-	-	1

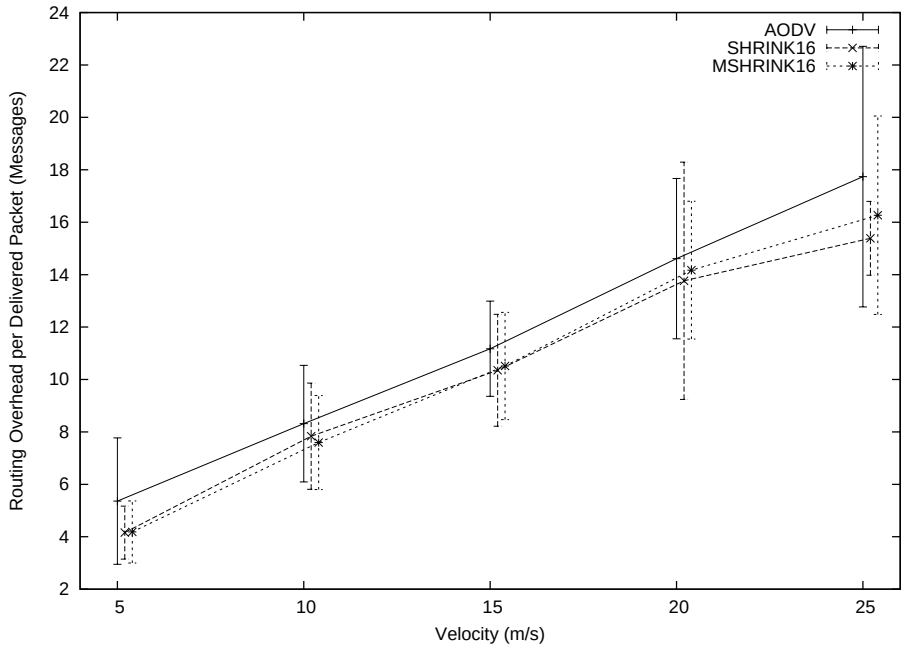


Figure 2.46: n=100, c=10

Packet Delivery Fraction (PDF):

For each of the preceding performance metrics (i.e. NPL, NEN, NRL), both SHRINK16 and MSHRINK16 exhibited similar performance on the metric of interest. In the next set of experiments, we measured packet delivery ratio. Here we saw that the two proposed schemes are distinct in terms of PDF performance. Figure 2.47 indicates PDF for different network sizes when maximum velocity is set at 15 m/s.

Table 2.19: Correlation coefficients for curves in Figure 2.47.

	50 nodes			100 nodes		
vel =15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.805	-0.538	1	0.922	0.965
SHRINK16	-	1	-0.470	-	1	0.948
MSHRINK16	-	-	1	-	-	1

Table 2.19 contains correlation coefficients for trials corresponding to Figure 2.47. The table shows it is not justifiable to compare average performance of the schemes

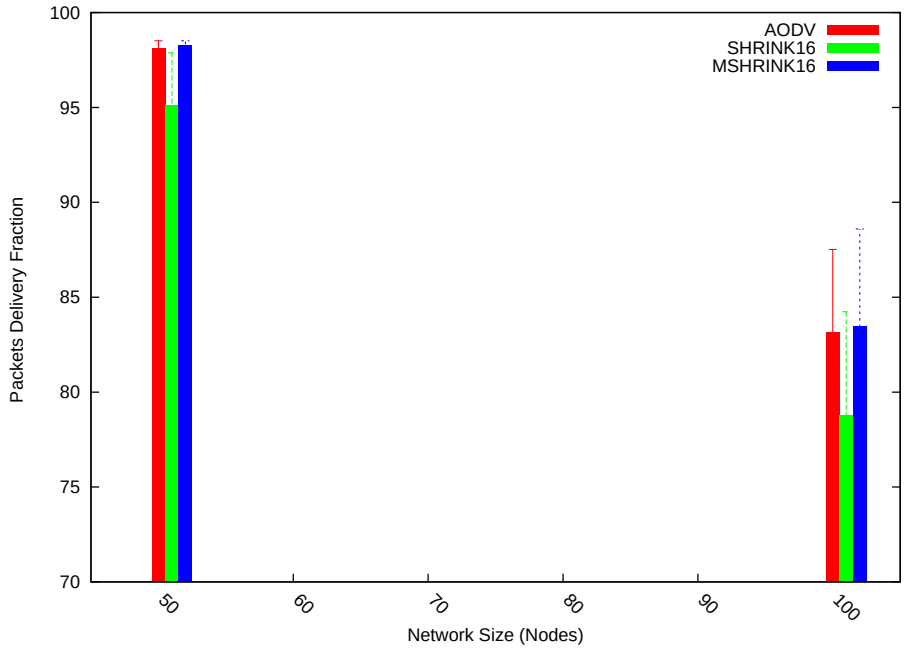


Figure 2.47: v=15, c=10

in 50-node networks. In contrast, the coefficients are close to 1 for large network experiments, e.g. the 100-node network. From Figure 2.48 and Table 2.19, it is clear that MSHRINK16 yields higher PDF than SHRINK16 by 13% at 5 m/s setting, 3% at 25 m/s setting, and intermediate values at other mobility settings.

Average end-to-end delay (E2E):

In this set of experiments, we measured average end-to-end delay in packet delivery to destinations. Figure 2.49 depicts average end-to-end delay for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 on different sized networks, based on 10 trials, in which each trial consisted of approximately 50,000 data packets being delivered on 10 different connections.

Table 2.20 includes correlation coefficients for Figure 2.49. According to the table, the correlation coefficient between SHRINK16 and MSHRINK16 is 0.463 in 50-node net-

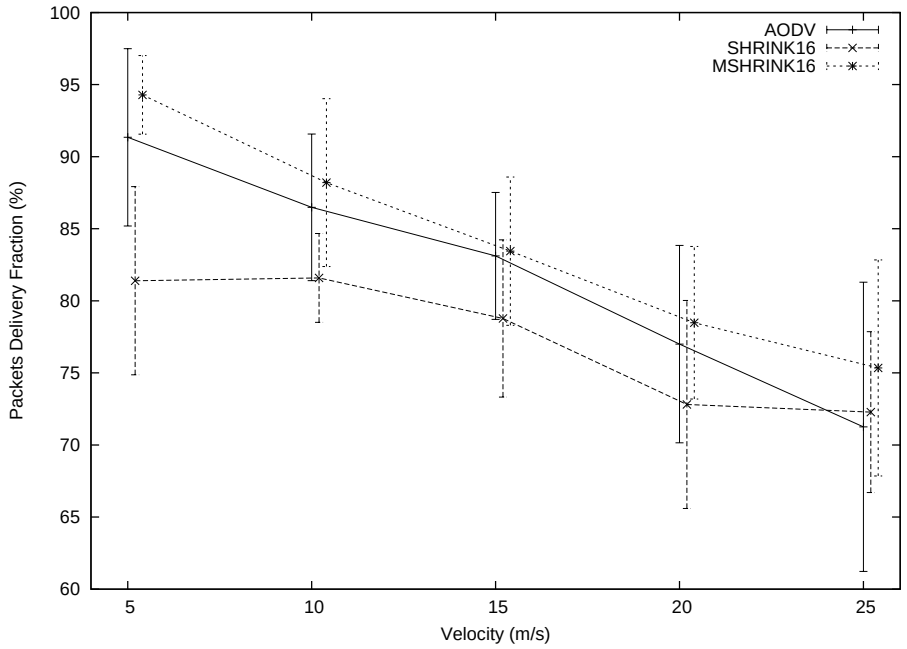


Figure 2.48: n=100, c=10

work, and is 0.709 in 100-node network, indicating it is difficult to justify comparing these schemes based on mean performance values.

Table 2.20: Correlation coefficients for curves in Figure 2.49.

	50 nodes			100 nodes		
vel =15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.919	0.501	1	0.827	0.343
SHRINK16	-	1	0.463	-	1	0.709
MSHRINK16	-	-	1	-	-	1

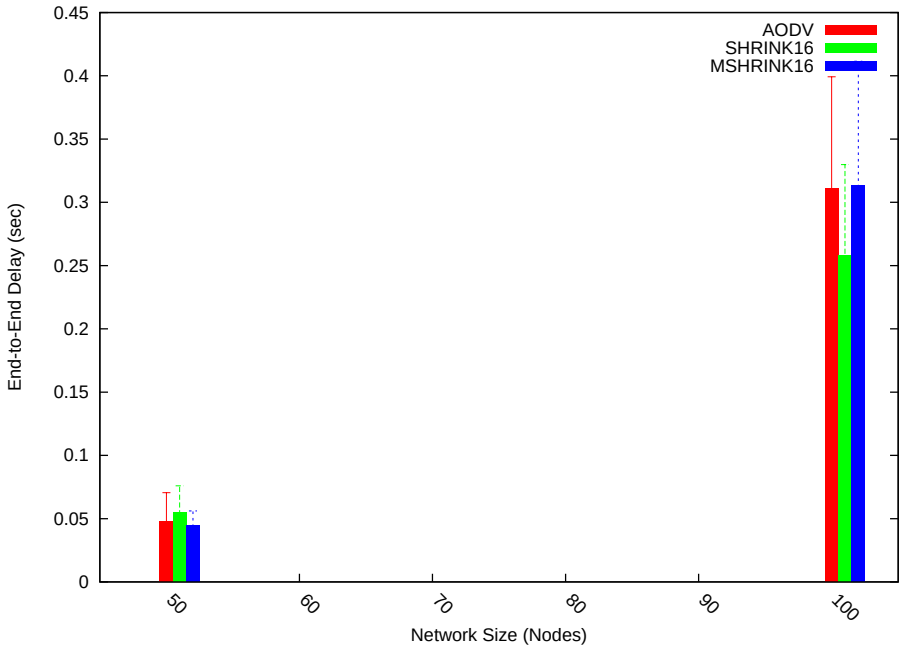


Figure 2.49: v=15, c=10

Average Connection Lifetime (ACL):

In this set of experiments, we measured the average lifetime of active connections. Figure 2.51 shows average connection lifetime for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 in different network sizes. Table 2.21 shows corresponding correlation coefficients for trials underlying the curves in Figure 2.51. Since the correlation coefficient between SHRINK16 and MSHRINK16 is 0.777 in 50-node network, and is 0.967 in 100-node network, performance of the evaluated mechanisms can be conducted based on the curves in the 100 node experiment.

According to Figure 2.51, applying SHRINK16 yields longer-lived routes in comparison with pure AODV and AODV+MSHRINK16. This can also be seen from Figure 2.52, which shows the average connection lifetime at different mobility levels for the larger 100-node network experiments.

5

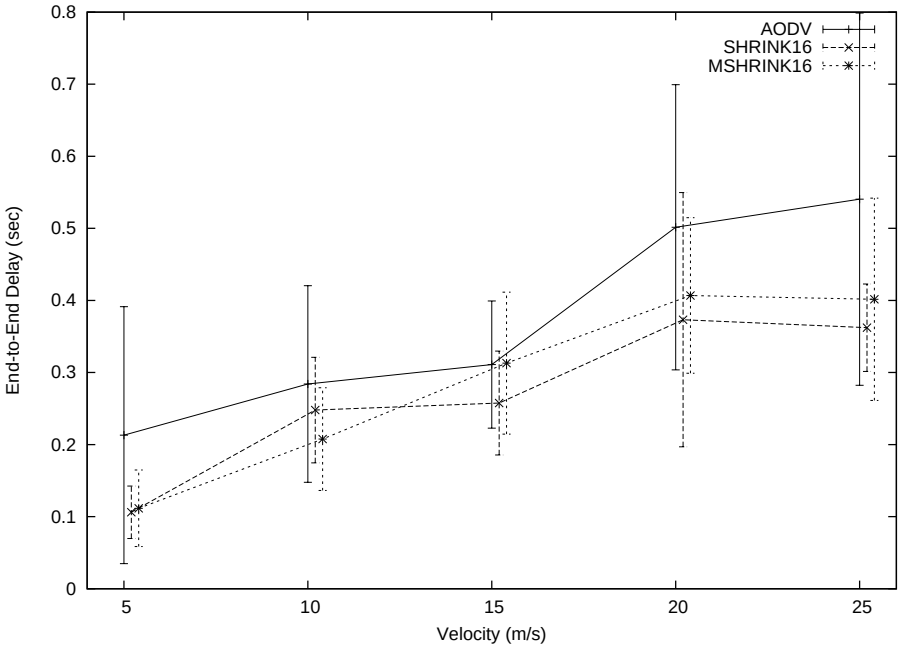


Figure 2.50: n=100, c=10

Table 2.21: Correlation coefficients for curves in Figure 2.51.

	50 nodes			100 nodes		
vel =15 m/s	AODV	SHRINK16	MSHRINK16	AODV	SHRINK16	MSHRINK16
AODV	1	0.784	0.750	1	0.515	0.673
SHRINK16	-	1	0.777	-	1	0.967
MSHRINK16	-	-	1	-	-	1

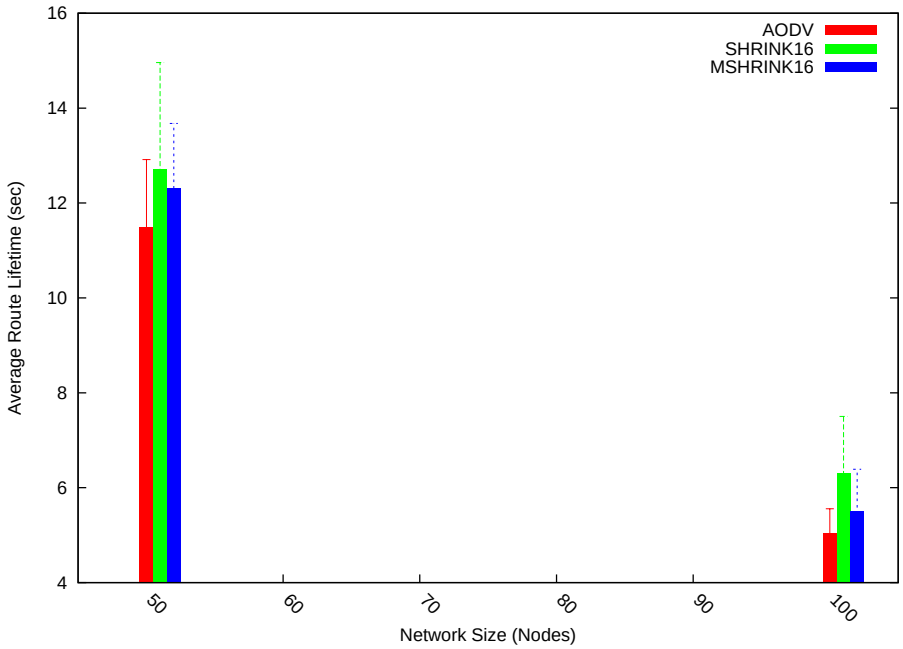


Figure 2.51: v=15, c=10

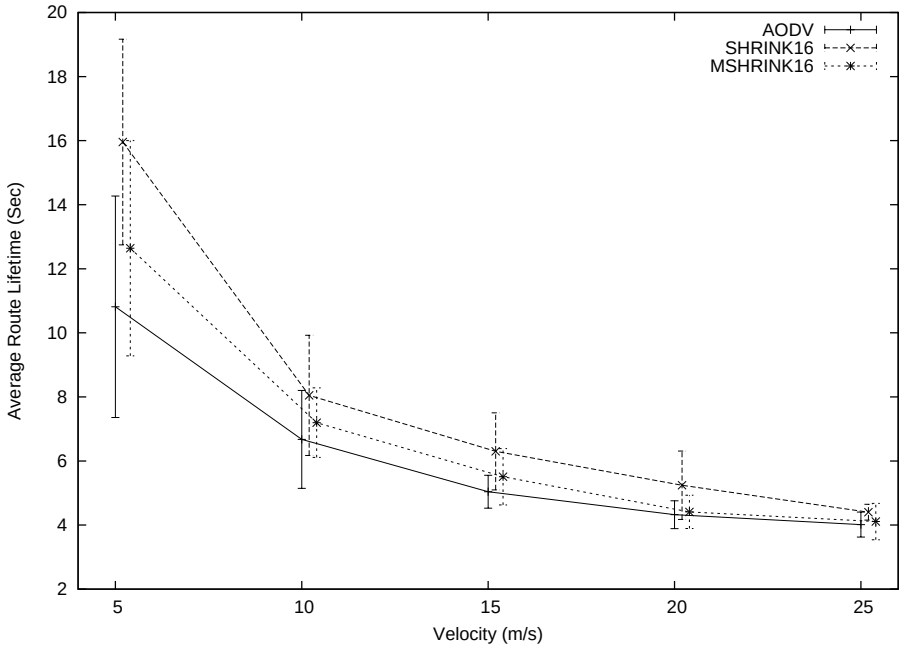


Figure 2.52: n=100, c=10

2.10 Formal Analysis of Scheme Invariants

In this section we seek to prove that the 1-hop Shrinking and Multihop Shrinking schemes are both loop-free. To express the argument formally, we shall need to introduce some preliminary definitions, which will build on the notation of Section 1.3.1 (pp. 9).

Definition 5. *Given two nodes $u, d \in V$, we define the sequence*

$$u \overset{t,d}{\rightsquigarrow} = \{w_0 = u, w_1, \ldots, w_{k-1}, w_k, \ldots\}$$

where

$$w_i(t).T[d].next = w_{i+1}.$$

In arguments where there is no ambiguity about the destination, we will simplify the notation to $u \overset{t}{\rightsquigarrow}$. We will frequently view $u \overset{t,d}{\rightsquigarrow}$ as a finite ordered set by truncating it at the point where it begins to repeat.

Definition 6. *Given two distinct nodes $u, v \in V$, we define a relation of u being upstream of v relative to d at time t:*

$$v \overset{t,d}{\rightarrow} u \Leftrightarrow v \in u \overset{t,d}{\rightsquigarrow}.$$

Note that this implies that there exists a sequence $w_0 = u, w_1, \ldots, w_{k-1}, w_k = v$ such that

$$w_i(t).T[d].next = w_{i+1},$$

for $i = 1, \ldots k-1$. If $v \notin u \overset{t,d}{\rightsquigarrow}$, we will write $v \overset{t,d}{\not\rightarrow} u$. In arguments where there is no ambiguity about the destination, we will simplify the notation by dropping the variable

d. Whenever $u \xrightarrow{t} v$*, we will frequently abuse the notation and use the relation symbol to also denote the ordered set*

$$u \xrightarrow{t} v = \{w_0 = u, w_1, \ldots w_k = v\}.$$

To prove that 1-hop Shrinking and Multihop Shrinking schemes are loop-free, we shall assume that AODV, when running in pure form, is loop-free. This fact has been established [42, 40], and so, we take it as an axiom in subsequent arguments.

Axiom 1. *(AODV is loop-free) In a network where all nodes* $v \in V$ *are running pure AODV, if* $u, v, d \in V$ *are distinct nodes and* $u \xrightarrow{t,d} v$*, then* $v \not\xrightarrow{t,d} u$*.*

2.10.1 1-hop Shrinking is Loop-Free

Lemma 1. *In a network running AODV, destination sequence numbers cannot decrease over time. Formally, given two nodes* $u, d \in V$*, and two times* $t' > t$*,*

$$u(t').T[d].dsn \geqslant u(t).T[d].dsn$$

Proof. Immediate, from the AODV specification. □

Lemma 2. *At any instant in time, the vertices along the forwarding path to destination d have non-decreasing dsns. formally, suppose* $u, d \in V$ *are two nodes in a network running AODV, and* $u \overset{t,d}{\rightsquigarrow}$*. Then at any time t, the successive nodes* $w_i \in u \overset{t,d}{\rightsquigarrow} (i = 0, \ldots, k-1)$ *:*

$$w_i(t).T[d].dsn \leqslant w_{i+1}(t).T[d].dsn$$

Proof. Whenever a *dsn* is changed at u for a valid routing table entry, it is to reflect what is specified in a received RREP message. Suppose the RREP message was received from v. Then according to the AODV specification, the operation of (i) changing the *dsn* and (ii) changing the forwarding pointer (to point to v) must be done *atomically*. However, the v does not send the RREP unless it has completed these two operations itself. Combining with Lemma 1, the assertion is proved. □

Lemma 3. *If the forwarding entry in a routing table changes, then the dsn associated with that entry increases. Formally, given two nodes $u, d \in V$, and two times $t' > t$,*

$$u(t').T[d].next \neq u(t).T[d].next \Rightarrow u(t').dsn > u(t).dsn$$

Proof. Forwarding pointers for d at node u are changed on receipt of a RREP. RREPs are received in response to RREQs. RREQs are either (i) generated as part of local recovery, or (ii) forwarded on behalf of other nodes because there is not a fresh enough route available. In case (i) the node u is required to use a higher *dsn* than the one associated with the previous valid routing table entry. In case (ii) the received request, by virtue of being forwarded, had a higher *dsn* than the present routing table entry (if one was present at u). In both cases, the RREP that returns will specify a *dsn* that is *higher* than the *dsn* associated with the last valid routing table entry for d at node u. □

Proposition 1. *If the relative position of two nodes on a flow is swapped, the dsn at the new downstream node must increase. Formally, suppose $u, v \in V$ are nodes in a network running AODV, for which $u \overset{t,d}{\rightarrow} v$. If at a later time $t' > t$, we have that $v \overset{t',d}{\rightarrow} u$, then it must be that*

$$u(t).T[d].dsn < u(t').T[d].dsn$$

Proof. By Axiom 1, there cannot be a time t_0 for which both $v \overset{t_0}{\rightarrow} u$ and $u \overset{t_0}{\rightarrow} v$. Since $v \overset{t',d}{\rightarrow} u$, there must be a time $t_1 \in [t, t')$ after which u ceases to be upstream of v, i.e. such that for all $t_2 \in [t_1, t']$,

$$u \overset{t_2}{\not\rightarrow} v.$$

Since $v \overset{t',d}{\rightarrow} u$, there must be a local recovery initiated (or global discovery which passed through) a node b at time t_3. The node b must have been downstream of v at a time t_3^- i.e.

$$b \in v \overset{t_3^-,d}{\rightsquigarrow}$$

and the RREQ generated by b must have been responded to by some node a that was *ultimately* upstream of u, i.e.

$$a \overset{t',d}{\rightarrow} u.$$

Suppose that this local recovery which started at t_3, was responded to by node a at time $t_3 + \epsilon_1$, and the response returned to b by time $t_3 + \epsilon_1 + \epsilon_2$, where

$$[t_3, t_3 + \epsilon_1 + \epsilon_2] \subseteq [t, t'),$$

For AODV to have avoided loops, we know that

$$t_3 + \epsilon_1 + \epsilon_2 > t_1$$

According to the AODV specification, if the local recovery that was initiated (or the global discovery that was forwarded) by b, was responded to by a, then the previous valid *dsn* associated with destination d at node b was strictly smaller than

the corresponding dsn at a.

$$b(t_3^-).T[d].dsn < a(t_3 + \epsilon_1).T[d].dsn$$

Again, according to the AODV specification, when the local recovery (or the forwarded global discovery) completes, the dsn at the b increased to match the value at a. Specifically:

$$a(t_3 + \epsilon_1).T[d].dsn = b(t_3 + \epsilon_1 + \epsilon_2).T[d].dsn$$

Since $t_3 + \epsilon_1 + \epsilon_2 < t'$, by Lemma 1,

$$b(t_3 + \epsilon_1 + \epsilon_2).T[d].dsn \leqslant b(t').T[d].dsn$$

After the RREQ is completed, at time t', $v \overset{t',d}{\rightarrow} u$ through this arc made from b to u. In particular, $u \in b \overset{t',d}{\rightsquigarrow}$, so by Lemma 2,

$$b(t').T[d].dsn \leqslant u(t').T[d].dsn$$

On the other hand, since $b \in v \overset{t_3^-,d}{\rightsquigarrow}$, by Lemma 2,

$$v(t_3^-).T[d].dsn \leqslant b(t_3^-).T[d].dsn$$

and since $t \leqslant t_3^-$, by Lemma 1

$$v(t).T[d].dsn \leqslant v(t_3^-).T[d].dsn$$

and since $u \overset{t,d}{\rightarrow} v$, by Lemma 2,

$$u(t).T[d].dsn \leqslant v(t).T[d].dsn$$

Combining the previous inequalities, we get that

$$u(t).T[d].dsn < u(t').T[d].dsn$$

The Proposition is proved. □

Remark 1. *To prove loop-freeness of the* **Shrink** *scheme, we augment it as follows. Suppose we have a triplet of consecutive nodes u, w, v along a route to d. When a node u sends a SHRINK0 message to v at time t_0, it also places the current dsn associated with the destination into the message, i.e. $u(t_0).T[d].dsn$. Node w receives the message at time t_1, and sends it to v at time t_2 (including $v(t2).T[d].dsn$). When v receives this message at time t_3, it sends a SHRINK1 message to u at time t_4. This SHRINK1 message contains the original $u(t_0).T[d].dsn$, as carried by the SHRINK1 message from u to w and w to v. Node u, upon receiving the SHRINK1 message at time t_5, acts on it only if*

$$u(t_0).T[d].dsn = u(t_5).T[d].dsn$$

Theorem 2. *If AODV is loop-free, the AODV+1-hop Shrinking is loop-free.*

Proof. Following the nomenclature of Remark 1, suppose, towards contradiction, that the first loop caused by AODV+1-hop Shrinking occurs because of a routing table change executed upon receipt of a SHRINK1 message by node u at time t_5, sent to it

by downstream node v. For a loop to arise when u acts on the SHRINK1 (making v the next hop to d) it would have to be the case that at time t_5, node v was actually upstream of u. But we know that earlier, at time t_0, when the SHRINK0 underlying the SHRINK1 was sent by node u, node u was upstream of node v. From Proposition 1, we know that if the relative position of two nodes on a flow is swapped, the dsn at the new downstream node must increase. This implies that the dsn at u increased in the interval $[t_0, t_5]$. But this contradicts the fact that for the SHRINK1 to have been acted upon it was necessarily the case that

$$u(t_0).T[d].dsn = u(t_5).T[d].dsn$$

The contradiction proves the theorem. □

2.10.2 Multihop Shrinking is Loop-Free

Remark 2. *We take a similar approach in proving the loop-freeness of our* **Multihop Shrinking** *scheme. To carry out the proof, we will need to make some modifications to the manner in which it operates, in order to obtain analogous assurances necessary for the proof: Suppose node u is initially upstream of node v. Node u broadcasts a SHRINK message at time t_0, which is propagated far downstream, hop-by-hop, ultimately reaching node v. Suppose now the SHRINK message is broadcast by node v at a time t_4, and it is received by node u at time t_5, which seeks to act on the SHRINK message by making a direct shortcut to v. We would like to be absolutely certain that node u cannot be downstream of v by time t_5, even though it was upstream of v at some previous time t_0.*

We augment the **Multihop Shrinking** *protocol as follows. The SHRINK message M will carry with it the following additional fields:*

- *source: the ultimate source of this SHRINK message, i.e. the source of the connection.*
- *destination: the ultimate destination of this SHRINK message, i.e. the destination of the connection.*
- *dsn: the destination sequence number associated in the routing table entry at the sender of this SHRINK message.*
- *ssn: the shrink sequence number, set by the connection source and incremented for each successive SHRINK message generated.*

Nodes augment their state by having map Aux, keyed by $(source, destination)$ where the value associated with this key is a list of (ssn, dsn) pairs, sorted by increasing ssn. Nodes enforce an upper bound of K on the length of these lists, by remove entries in a FIFO manner. More specifically, when a node is the next-hop recipient of a SHRINK message M, it looks up $Aux[(M.source, M.destination)]$, which is a list of (ssn, dsn) pairs. If this list is already of length K, it removes the first (oldest) pair. It then adds the pair $(M.ssn, M.dsn)$ to the end of the list.

When a node u receives a SHRINK message M from downstream (i.e. it is not the next-hop recipient of the message) at time t, it does the following:

First, it looks up $Aux[(M.source, M.destination)]$, which is a list of (ssn, dsn) pairs. It checks within this list for a pair (x, y) in which the first entry $x = M.ssn$. If no

such entry is found, the message M is not acted upon. If an entry is found, then y is compared against node u's current dsn value for the destination. If

$$u(t).T[M.destination].dsn \neq y,$$

then the message M is disregarded. Otherwise, it is processed as a potential shortcut.

Theorem 3. *If AODV is loop-free, the AODV+Multihop Shrinking is loop-free.*

Proof. Following the nomenclature of Remark 2, suppose, towards contradiction, that the first loop caused by AODV+Multihop Shrinking occurs because of a routing table change executed upon receipt of a SHRINK message by node u at time t_5, sent to it by downstream node v. For a loop to arise when u acts on the SHRINK (making v the next hop to d) it would have to be the case that at time t_5, node v was actually upstream of u. But we know that earlier, at time t_0, when the original SHRINK was sent by node u, node u was upstream of node v. From Proposition 1, we know that if the relative position of two nodes on a flow is swapped, the dsn at the new downstream node must increase. This implies that the dsn at u increased in the interval $[t_0, t_5]$. But this contradicts the fact that for the SHRINK to have been acted upon by u it was necessarily the case that

$$u(t_0).T[d].dsn = u(t_5).T[d].dsn$$

The contradiction proves the theorem. □

Remark 3. *We note that the Aux table has size $O(c)$ where c is the number of connections. Hence, if each node is on average the source/destination of a constant number of connections, the amortized size of the Aux tables is $O(|V|)$.*

CHAPTER 3

DYNAMIC ROUTE MAINTENANCE

Maintaining active connections in mobile ad hoc networks (MANETs) involves preventing and addressing link failures. These are challenging issues in MANETs, owing to the dynamic character of network topologies therein. Nodes in these types of networks may move both independently and unpredictably, causing breakages of existing links (as well as the formation of new links) between node pairs. Exploiting newly formed links for route optimization was the subject of Chapter 2. In this chapter, we focus on the dual problem, namely the issue of link breakages due to node mobility.

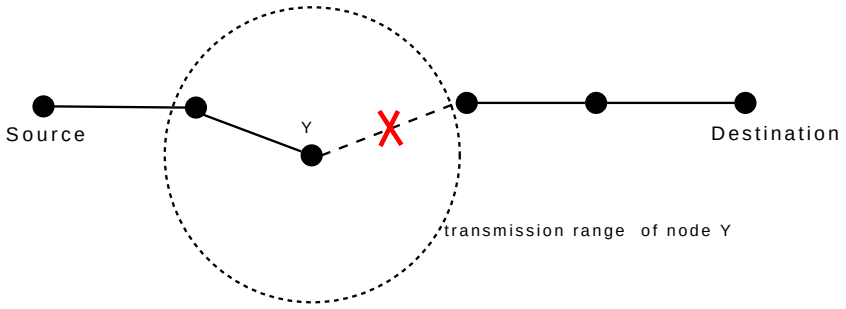


Figure 3.1: Link breakage on a connection.

A link between two nodes fails when they move out of each other's transmission range. This event is illustrated in Figure 3.1. When a link failure occurs, it is generally the responsibility of the underlying routing protocol to reinitiate route discovery for all active connections that were previously routed over the failed link. This process is quite expensive in terms of incurred control traffic. To make matters worse, since

connections in MANETs are constructed in a multihop manner, they become disconnected when even a single one of their constituent links fails. In high mobility scenarios, link failures can occur very frequently, and this in turn degrades overall network performance. In such settings, it is highly desirable to prevent (or postpone) mobility-related link failures. Toward this end, we consider two different classes of route maintenance schemes:

i. Prefailure link **expansion** methods,

ii. Postfailure **local recovery** methods.

The main difference between these two classes of methods is related to *when* they act, relative to time of link failure. The former class of methods aim to prevent impending link failures *before* they happen, whereas methods in the latter class try to repair link failures *after* they occur (in an efficient, effective way). In this research, we developed one scheme in each of the two classes. These schemes were implemented in the ns2 network simulator [3], which is a commonly employed discrete event and packet level network simulation software package. Through extensive simulation experiments, we show that our novel schemes provide notable improvements in the following routing performance metrics:

i. Decrease in overall routing load,

ii. Increase in packet delivery ratio,

iii. Improvement in path optimality, and

iv. Reduction in the number of global flooding events.

We have also provided a theoretical analysis showing that our schemes do not affect the loop-free guarantees provided by the underlying routing algorithm, which in this case, is the Ad hoc On Demand Distance Vector (AODV) protocol [40]. Although we have designed extensions to AODV in this chapter, our schemes can readily be adapted as extensions to almost any reactive routing protocol, since we intentionally made no strong assumptions specific to AODV during the design process.

The rest of the chapter is organized as follows. In Section 3.1, we give an informal description of the problem, and describe underlying sources of difficulties. Then, in Section 3.2, we describe the significance of the problem and the potential benefits obtained through its resolution. In section 3.3, we provide a comprehensive review of the related prior work, and the shortcomings of previous solutions. Our constraints in designing these schemes are described in Section 3.4, and system assumptions are outlined in Section 3.6. Ultimately, we present the novel schemes, in detail, in Section 3.7. The performance analysis of the proposed schemes, together with a description of the simulation environment and experimental setup, are given in Section 3.8. Finally, we prove that the proposed schemes provide a loop-free routing in Section 3.9.

3.1 Statement of the Problem

We are interested in preventing link breakages due to mobility, and in addressing link breakages once they occur. A link between two nodes fails when they move out of each other's transmission range, as illustrated previously in Figure 3.1. Since connections in MANETs are constructed in a multihop manner, they become disconnected when even a single one of their constituent links fails. In high mobility scenarios, such link

failures may occur frequently, degrading overall network performance. It is desirable therefore to prevent or postpone mobility-related link failures. In this chapter, we shall move towards goal this by seeking answers to the following concrete questions:

1. How can we detect a possible link breakage *before* it happens?
2. How can we prevent an impending link breakage *before* it happens?
3. How can we efficiently repair a link *after* it breaks?
4. While preventing or repairing link failures, can we also maintain (or even improve) route optimality?

3.2 Significance of the Problem

Link breakages generally entail extra work for routing protocols, since connections must be repaired. In particular, link disconnections generally have the following undesirable consequences:

- *Service interruptions.* In some critical applications, users are very sensitive to service interruptions on their connections. A link disconnection on an active connection is catastrophic for this type of application.
- *Increase in routing overhead.* In order to re-establish a connection between a pair of source and destination nodes, it is often required to initiate a network-wide route discovery process, which is very expensive in terms of control traffic incurred.

- *Increase in packet loss.* The greater the frequency of link disconnections, the more packet loss occurs, since packets arriving at a broken link are queued until the finite queue at the node's egress port becomes full, at which point packets are dropped.

- *Increase in average delay:* The average delay in packet delivery can become high in the presence of link disconnections, since it takes time to construct a new route that replaces or repairs the broken one.

3.3 Related Work

3.3.1 Expansion

One of the earliest research in this field is the work of Park and Voorst [39], in 2001, who presented an algorithm called "Anticipated route maintenance" that predicts whether a link between two nodes will break within a predefined time interval. The authors used the node locations and velocities, determined via GPS, to derive their likelihood estimates. Park and Voorst's algorithm consists of two phases: Expand and Shrink. The Expand routine prevents the route from being broken by inserting bridge nodes into a weak link. The Shrink routine eliminates unnecessary hops and shortens the path, thereby preventing it from becoming unnecessarily long. The implementation of the scheme, and its performance evaluation was published by Al-Shurman et al. in 2004 [7]. Unfortunately, their scheme uses GPS, and requires nodes along a connection to exchange routing table information.

In contrast to Park and Voorst's GPS-based approach, Qin et al. [43], in 2002, pro-

posed a link breakage prediction algorithm based on the change in the signal strength of consecutive received packets. When a node estimates that there is an incident link that is likely to break soon, it initiates a "Broken route message" and requests the source node to find an alternate path. The authors' proposed mechanism tries to find an alternate path *before* the path become disconnected. They do not attempt to eliminate the frequency of global route discovery, but rather seek to minimize the expected number of dropped packets during the fail-over sequence by anticipating the link's imminent failure. Although their algorithm is developed specifically for Dynamic Source Routing (DSR), it is claimed to be easily adaptable into other routing protocols.

Following Park and Voorst's GPS-based solution, a recent 2008 paper by Sjaugi et al. [50] appeals to location information for nodes (provided by GPS), in order to detect unsafe links–defined as links whose geometric length exceeds a certain threshold distance. The location information is updated by piggybacking it onto the headers of packets. When a link is found unsafe, local (1-hop) broadcasting is performed, in order to find a bridge node which can serve as an intermediate relay node between endpoints of the unsafe link. The path expanding routine is similar to what was proposed in Park and Voorst. However, Sjaugi et al. do not comment about shrinking operations, and so, their proposed mechanism may cause paths to become arbitrarily (and unnecessarily) long.

3.3.2 Local Recovery

While many researchers [36, 30, 38, 35, 8, 49] have proposed *local recovery* protocols, none of them presents a viable solution to the degradation in path optimality which

necessarily occurs over time as successive local recovery operations take place along a route. We discuss the three most recent approaches in some detail, as they are representative of the types of ideas that have been developed.

Liu et al. propose, in 2003, a local recovery protocol called PATCH [35], that is specifically designed for DSR. In their scheme, when a link on an active route becomes disconnected, the upper endpoint of the broken link initiates local recovery by broadcasting a ROUTE_REQUEST packet with low TTL (e.g. TTL=2). Upon receiving such a ROUTE_REQUEST packet, the nodes located on the downstream side of the broken link send back a ROUTE_REPLY packet to the initiator node, along the opposite path taken by the ROUTE_REQUEST. Once the ROUTE_REPLY packet arrives at the initiator of the local recovery operation, the patch is made. The main shortcoming of this approach is that path optimality degrades over time, since path lengths increase.

In a more recent 2008 study, Alshanyour and Baroudi [8] present another local recovery protocol called Bypass-AODV. In this scheme, whenever a link breaks at the MAC layer, the upstream endpoint of the broken link initiates the proposed local recovery protocol, by broadcasting a bypass-RREQ message with TTL of 2. This bypass-RREQ is a modified route request message, that is sent *regardless of the position where the break occurs along the route* (this is in contrast to AODV's local recovery logic). In a response to a received bypass-RREQ, only the downstream endpoint of the broken link is allowed to send back a bypass-RREP (which is a modified route reply message). When the initiating node of the local recovery operation receives the bypass-RREP message, it updates its routing table, and starts transmitting buffered packets on the updated route. Like PATCH, the major drawback of Bypass-AODV

is that it produces unnecessarily long routes, especially after several local recovery attempts have occurred.

Another recent 2008 paper by Shi and Deng [49] also proposes an improvement to AODV's default local repair protocol. In this work, the authors use "Hello" messages between neighbors (along an active connection) as a means to detect possible link failure. If a node does not receive a Hello message from its neighbors for some time, it concludes that the corresponding link between them has failed, and a local recovery protocol is initiated. The protocol is, however, based on the assumption that each node on a connection has knowledge of its upstream neighbor. Regrettably, this is not the case in AODV. Nonetheless, making this assumption, the authors propose that upon detecting link disconnection, the downstream endpoint of the broken link broadcasts an unsolicited Route Reply message (without waiting for any Route Request message), with the objective of reaching the upstream endpoint of the broken link through broadcasting via intermediate nodes. This approach is problematic because the overhead of continuous periodic "Hello" traffic is onerous in lower mobility settings (unless nodes can sense their position, though this introduces additional capability requirements for the nodes).

3.4 Design Objectives

In designing our solutions, we keep the following objectives in mind:

(I) **The solution should not be based on location information such as GPS.** A solution based on location information provided by GPS requires satellite level infrastructure. Even if the location information is obtained by another

technique like triangulation it still quite likely requires extra equipment such as directional antennae. Park and Voorst [39] developed an algorithm which predicts whether a link between two nodes will break within a defined time interval, using the nodes' locations and velocities, determined using GPS. Their algorithm consist of two phases: *Expand* and *Shrink*.

(II) **The solution should not incur too much control traffic.** One of our objectives is to reduce the amount of control traffic required to fix a broken link in case of link failure. Routing protocols generally solve this problem by re-initiating the route discovery process from the source node. An effective solution should solve this problem by incurring less control traffic. Regarding this issue, Qin et al. [43] propose a link breakage prediction algorithm based on the change in the signal strength of consecutive received packets. When a node estimates that there is a adjacent link which is likely to be broken soon, it initiates a "Broken route message" and requests the source node to find an alternative path. While, Qin's mechanism tries to find an alternative path before the path become disconnected, it does not try to eliminate the need for a global route discovery.

(III) **The solution should not degrade route optimality.** While fixing a broken link or preventing an impending one, the solution might cause the relevant route to become unnecessarily long. For example, a recent 2008 paper by Sjaugi et al. [50] appeals to location information for nodes (as provided by GPS) in order to detect unsafe links–that is, links whose geometric length exceeds a certain threshold distance. When an unsafe is found, local (1-hop) broadcasting is performed to find a bridge node between endpoints of the unsafe link. This is a path expanding routine, however, the proposed mechanism may cause paths

to become arbitrarily (and unnecessarily) long.

(IV) **The solution should provide loop free routing guarantees.** Most routing protocols provide a guarantee of loop-free routes. This is an important property for network routing since loops consume network capacity gratuitously, and it requires extra work to detect routing loops. Any proposed solution should maintain the loop free property of the system intact.

(V) **The solution should not be applicable to only one specific protocol.** In general, the solution should work in harmony with any of the traditional reactive routing protocols, such as AODV and DSR, or at the very least should be readily adaptable to any of them. Toward this end, we will make very minimal assumptions while designing our solution schemes.

(VI) **The solution should be event-triggered.** In order for the solution to have low control traffic, it should be initiated only when it is required, rather than operating on the basis of periodic events.

(VII) **The solution should be responsive.** If the time required to complete preventative actions in anticipation of link breakage takes too long (with respect to node velocity), then the expected benefit from the solution may not materialize because the link will break before the preventative action can complete. If the time required to complete post-breakage recovery takes too long (with respect to node velocity), then packet losses will mount in the interim interval. These concerns indicate that any proposed solution schemes need to operate responsively.

3.5 Evaluation Considerations

In evaluating our solutions, we keep the following considerations in mind:

(I) **The solution should be evaluated under different mobility assumptions.** One of the most common mobility models used in MANET simulations is Random Waypoint (RWP) mobility model. In this work, a different node mobility model is introduced that addresses the shortcomings of the classical RWP model. It was described in detail earlier, in Section 2.9.3 (pp. 45).

(II) **The solution should be evaluated under different assumptions of network size and connection density.** This is essential to evaluate the scalability of our proposed schemes.

(III) **The solution should be validated by extensive simulations.** The time and space complexities of the experiments performed in ns2 are high. For this reason, most relevant studies perform only one trial in their experiments (e.g. [41, 12]). In order to show that conclusions are not particular to specific scenarios, many trials must be performed.

3.6 Assumptions

To make the proposed schemes adaptable to other reactive routing protocols, we are careful to not make strong AODV-specific assumptions. The following assumptions were made in developing the our solutions:

A1. *Uniform transmission power:* All nodes send packets with the same transmission

power; this also allows us to assume bidirectionality of the links in the network.

A2. *"Next hop" information is present in routing tables:* This is required for packet forwarding, and therefore, is not a strong assumption.

A3. *"Hops to destination" information is present in routing tables:* At each node on a connection, the routing table maintains information about number of hops to the destination. This is supported by AODV and other traditional reactive routing protocols (or can be easily added to them).

A4. *Received signal strength measurement:* Nodes are able to estimate link quality based on the signal strength of received packets. Based on this, a node can classify adjacent links into two categories: (i) If the received signal strength is lower than a predefined threshold level[1], then the link the data packet arrived on is called a *weak link*. Weak links are regarded as likely to break soon; (ii) If a link is not weak, then it is regarded as *strong*.

A5*. *Distance (hop count) to the source:* (Optional) Although it is not required for the operation of our scheme, some reduction in control traffic overhead can be obtained if we assume that the hop count from the source is also available at each node along the route. This information can be obtained from the IP headers of data packets traveling along the connection, if we assume that all data packets are originated with a specific TTL (e.g. 255) (requiring no extra field in routing tables).

1. We took this to be the strength of a received signal at 90% of the common transmission radius. For example, if the transmission radius is 250m, then threshold level corresponds 225m, when we consider the two-ray ground reflection model [45] as implemented in ns2.

3.7 Proposed Solutions

We propose two different schemes called (i) **Expansion** and (ii) **LocOpt** The foremost difference between them is the time they are triggered. The Expansion scheme activates just before a link failure, whereas the LocOpt is initiated after a link breakage. In the following subsections, we describe each scheme in detail.

3.7.1 Expansion

The Expansion scheme detects weak links by measuring the received signal strength of incoming data packets, and performs preemptive actions in order to eliminate these weak links on-the-fly before failures occur, thereby, circumventing expensive global route recovery. These actions entail the following two fundamental operations:

- **Expand operation:** A *weak* link is replaced with two *strong* links by inserting an extra relay node between two endpoints of the *weak* link as illustrated in Figure 3.2.
- **Shrink operation:** A subpath which includes the *weak* link is eliminated by either (i) directly connecting the downstream endpoint of the weak link to an upstream node, or (ii) directly connecting the upstream endpoint of the weak link to the final destination, as illustrated in Figure 3.3. Such shortcuts may come into existence because of topological changes in network due to node movements.

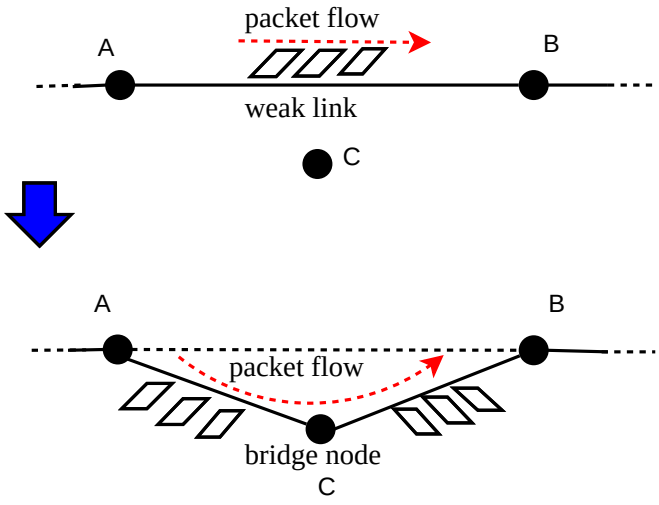


Figure 3.2: Expanding operation.

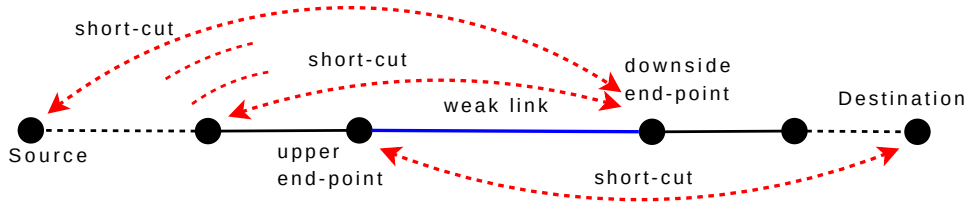


Figure 3.3: Shrink operation.

Both of the above operations can result in eliminating weak links without causing *any* interruption on the connection[2]. When a weak link is detected, both operations are attempted simultaneously. The Shrink (resp. Expand) operation may not always succeed because of unavailability of appropriate shortcuts (resp. bridge nodes). If both succeed, the side-effects of the Shrink operation supersede the side effects of the Expand operation by rendering the expanded segment irrelevant to the connection flow. Notice that the expand operation performs its task by lengthening the path, while the Shrink operation fulfills its mission by shortening the path. Over time, the two complement each other.

2. Brief out of order delivery may occur as packets queued along the old route interleave with packets forwarded on the new route, but we assume that higher-layer protocols like TCP will address this transient issue.

Implementation

The implementation is explained with reference to Figure 3.4.

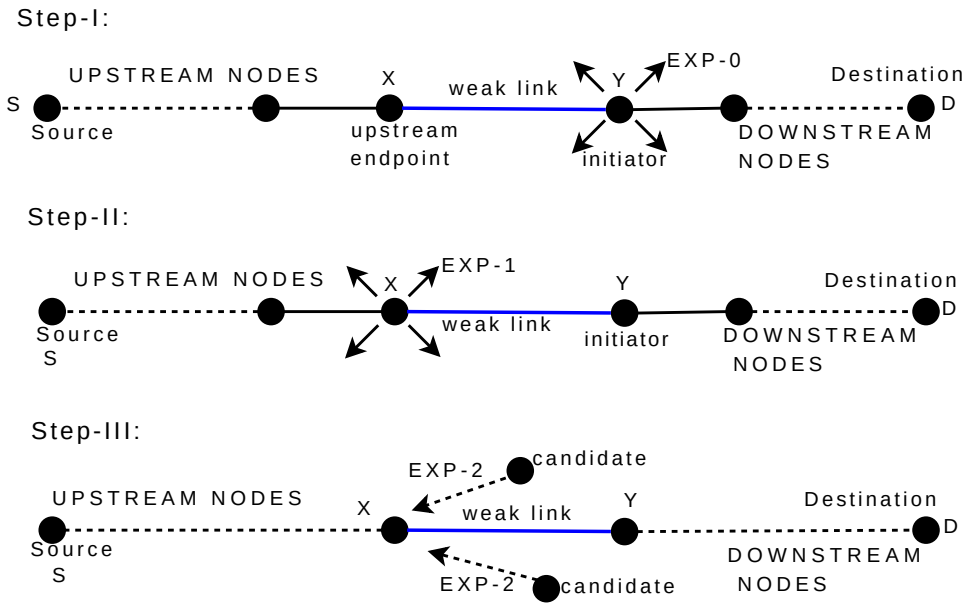


Figure 3.4: Steps I, II and III of the Expansion scheme.

The protocol is built using EXP messages which contain the following fields:

- **source-node** : The IP address of the source node of the connection.
- **final-destination**: The IP address of the final destination of the connection.
- **initiator**: The IP address of the node initiating Expansion.
- **previous-hop**: The IP address of the node before the initiator (on the route from source-node to final destination) (can be obtained from ARP). .
- **hops-to-destination**: The number of hops to the final destination (available in the routing table).

- **flag**: 0, 1, 2, or 3, depending on whether the message is an EXP-0, EXP-1, EXP-2, or EXP-3.

Step Ia. Initiation of the process: When a node (i.e. Y) detects, on the basis of the signal strength of a received data packet, that it is incident to a weak link (e.g. X–Y), the node is declared an ***initiator***. An initiator node starts the Expansion process by preparing a special EXP-0 packet (i.e. initiator=Y, previous-hop=X, final destination=D, source node=S, flag=0) and broadcasts it with TTL of 1.

Step Ib. Applying shortcut from the upstream nodes: After EXP-0 is broadcast by the initiator node, any upstream node on the connection that receives it "over a strong link"[3], the upstream node modifies its routing table so the initiator becomes the next hop for the destination declared in the received EXP-0 packet. The weak link is thus eliminated by shortening the path. If there are shortcuts between many upstream nodes and the initiator node, the shortcut established by the upstream node closest to the source node determines the flow of packets. Note that if both steps Ib and step II are successful, then the shortcut established in step Ib simply nullifies the side effects of step II in terms of the route of packet flow.

Step II. Applying shortcut to the destination: After the EXP-0 packet is broadcast by the initiator node, the upper endpoint of the weak link (i.e. node X) receives it and prepares another packet, identical to the received EXP-0 except that the previous-hop field is cleared, and flag=1. This EXP-1 is broadcast by X with a TTL of 1, as illustrated in Step-II of Figure 3.4. If the final destination (i.e. node D) receives the EXP-1 packet over a strong link, it sends a special unicast EXP-3 packet[4]

3. i.e. with a signal strength exceeding the weak link threshold, as defined in footnote 1
4. Only the flag field is relevant in EXP-3 packets.

to the originator of EXP-1 (e.g. node X) in order to inform it about the possible shortcut between them. Upon receiving the EXP-3 packet, the upper endpoint of the weak link (e.g. node X) updates its routing table so that the next hop is directly the destination node D. In this manner, the weak link is eliminated, and the path is be shrunk. One might wonder why only the destination node is taken as a potential candidate, rather than the more general strategy of finding downstream shortcuts. This is because there may be nodes with a valid route for to the destination (in their routing table) which are no longer located on the route itself (because of shortcut operations executed in earlier occurrences of steps Ib/II). Such nodes may mislead the proposed scheme if they are taken as candidate downstream shortcuts. The analogous problem does not manifest in searching for upstream shortcuts, because the initiator is guaranteed to lie on the connection path.

Step III. Bridge node insertion:

Regardless of whether the previous steps find a shortcut to eliminate the weak link, the proposed scheme performs a path expansion process to circumvent immanent link breakage. In this process, an extra relay node is inserted into the route. It is obvious that such a bridge node should be selected among the nodes located in the vicinity of the two endpoints of the weak link. However, the problem is that there is no information available regarding location of the nodes. Therefore, we developed a path expansion process by benefiting from EXP-0 and EXP-1 packets, which are broadcasted in earlier steps by the downstream and upstream endpoints of the weak link respectively. The *expand* process works as follows:

If a node receives EXP-0 and EXP-1 packets (with matching initiator, source and destination fields), both with high signal strength, it concludes that it is a candidate

bridge node which lies in the intersection of transmission radii of weak link's endpoints. Candidate bridge nodes send a special unicast packet, EXP-2, to upstream endpoint of the weak link (i.e. node X) in order to advertise itself as a candidate (see Step-III in Figure 3.4).

Bridge quality: Because there may be many candidate bridge nodes, the upper endpoint of the weak link may receive several EXP-2 packets. To facilitate the choosing from among the candidates, the EXP-2 message is augmented to carry an assessment of the bridge node's candidacy. In our implementation, each candidate bridge node records the received signal strength of the EXP-0 and EXP-1 messages, and puts the lower of these two values in the EXP-2 message. Upon receiving the first EXP-2 packet, the upstream endpoint of the weak link executes the following procedure: It inserts the bridge node by modifying its own routing table entry for D to be the sender of the EXP-2 packet. Upon receiving later EXP-2 packets, it compares the channel quality specified for the current bridge node, against the one advertised in the later EXP-2 packet. The later EXP-2 is is acted upon only if the bridge it advertises is of higher quality than the previous one. Bridge node selection implicitly converges once all candidates have send their EXP-2 packets.

3.7.1.1 Updating Distance Information in Routing Tables

In AODV, there is a field in node routing tables which maintains hop-count to destination. The values kept in this field must be strictly increasing along a route from destination to source in order to prevent loop formation during the route discovery processes [40]. However, when a bridge node is inserted into an active route as in the proposed expand process, this monotonicity may be disturbed because the field's

values in upstream nodes from the bridge node are no longer correct; they need to be increased by 1. This invariant violation may also cause malfunctioning of the proposed *Shrink* scheme, since nodes use the hop-count to destination to determine their topological relationship to the initiator. To fix this problem, all upstream nodes from bridge node to source node need to be informed about the new correct hop-count to destination. The challenge is that nodes on the route are generally not aware of their upstream nodes, rather they only know the next (downstream) hop towards a specific destination. To address this issue, we developed the following mechanism:

When the upstream endpoint of a weak link receives an executable EXP-2 packet (meaning the advertised bridge node will be inserted) it prepares a special packet, called EXP-4, and broadcasts it with TTL of 1 (initiator=X, final destination=D, source node=S, flag=4). When a node receives EXP-4, it infers that distance updating is to take place. To do this, the node checks its routing table to determine if the sender of the EXP-4 packet is the next hop for the specified destination in the packet. If it is, then it updates the hop-count to destination field in its routing table, and then broadcasts a similar EXP-4 packet in which its address is in the initiator field. Thus, the update process progresses towards source node using EXP-4 messages, and stops when it reaches the source node.

3.7.2 LocOpt

3.7.2.1 Local Recovery

In this section, we describe a new alternative local recovery protocol for AODV. The objective of the proposed local recovery protocol is to patch a broken link on a route by splicing a *single* extra relay node into the route between the two endpoints of a broken link. The process is described with reference to the illustration in Figure 3.5.

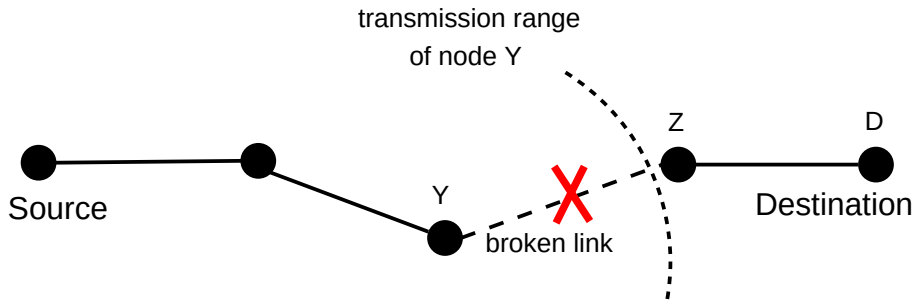


Figure 3.5: A connection failure.

The local recovery process is initiated by the upstream endpoint of a broken link (e.g. in the figure, by node Y) whenever the MAC layer reports the failure of a link that is known to have been actively carrying outbound traffic towards some destination. Such a link failure report occurs after several unsuccessful MAC layer attempts to reach the next hop during a packet transmission. When this occurs, the initiator node (i.e. node Y) begins buffering data packets for the corresponding destination, and prepares a RREQUEST packet which includes the IP addresses of both the previously declared next-hop (i.e. node Z) and the final destination (node D). Notice that including the previously declared next-hop in a RREQUEST packet *is a modification* of standard AODV packet formats, introduced by us to support the

proposed local recovery protocol. Then, node Y broadcasts the prepared RREQUEST packet with a TTL of 2, and waits for corresponding RREPLY packets while buffering incoming data packets for the connection under repair. Because we seek to reduce control traffic overhead (relative to what is incurred by standard AODV), in our protocol only two nodes are allowed to send back a RREPLY packet. These are: the previously declared next-hop (i.e. node Z), and the final destination (node D). Because of this, the initiating node of the local recovery protocol will receive at most two RREPLY messages. If it receives both of them, the RREPLY coming from the final destination node is given priority, since acting on it would result in a shorter connection route. Upon receiving a RREPLY packet, the initiator node updates its routing table and starts to send buffered data packets, in order. In this manner, the connection is repaired by splicing in an intermediate node.

The above local recovery operation is attempted at most twice. If the second attempt also does not succeed, then the initiator node drops all data packets that it has buffered, and sends an RERROR packet to the source node of the connection (i.e. node S). Upon receiving an RERROR packet along to route towards the source node, all nodes invalidate the corresponding entries in their routing table. The source, upon receiving the RERROR, initiates a global route discovery process.

The reader might object such that successive local recovery operations can produce unnecessarily long paths, over time. Indeed, although the proposed local recovery protocol can resolve route disconnections in a more efficient, faster and cheaper way, it would certainly result in paths that were far from topologically optimal, as a side effect of continuously inserting new nodes into a route. To eliminate such a side effect, we will simultaneously introduce a *route optimization* protocol. However, before we

can describe this protocol, we need to fix some of the inadvertent side effects of local recovery.

Distance Update Mechanism

In AODV, there is a dedicated field in the routing tables of the nodes which maintains the hop-count to the destination. The values kept in this field must be strictly increasing along any route from destination to source, in order to prevent routing [40]. However, when an extra relay node is inserted into an active route (as is the case in the proposed LocOpt protocol above), this monotonicity invariant may be violated because the field values at nodes that are upstream of the broken link are no longer correct, and need to be increased by 1. To fix this problem, all nodes upstream of the broken link need to be informed about the new correct hop-count to the destination. This situation is similar to what was encountered in the Expansion algorithm earlier. The challenge is that nodes on the route are not aware of their upstream nodes, and only know the next (downstream) hop towards the destination. To address this difficulty, we developed the following mechanism:

When a local recovery operation is successfully completed, the initiator node of the local recovery operation prepares a special packet, called DISTANCE, which includes: the IP addresses of the source node (i.e. node S), the destination of the path (i.e. node D), and itself (i.e. node Y). The initiator (i.e. node Y) *broadcasts* this DISTANCE packet with TTL of 1. When a node receives DISTANCE packet, it infers that distance updating is to take place. To do this, the node checks its routing table to determine if the sender of the DISTANCE packet is the next hop for the specified destination in the packet. If it is, then it updates the hop-count to destination field in its routing table, and then broadcasts a similar DISTANCE packet in which its

address is written into the sender field. Thus, the update process progresses towards source node using DISTANCE packets; the distance update process stops when it reaches the source node.

3.7.2.2 Route Optimization in Local Recovery

As mentioned earlier, local recovery operations inherently degrade route optimality because they lack a global perspective during recovery process. It is difficult to provide both local recovery and route optimization at the same time, as we noted in Section 3.3. In contrast to prior local recovery approaches presented in the literature, we would like to maintain (or even improve) route optimality while performing local recovery operations. To do this, we will combine the proposed local recovery mechanism with one of the route optimization algorithms proposed in Chapter 2 (i.e. the 1-hop Shrinking and Multihop Shrinking schemes), making the two operate collaboratively. Since the performance of Multihop Shrinking is superior to that of 1-hop Shrinking (see Chapter 2), we prefer to use the Multihop Shrinking scheme. To perform the integration, we must modify the Multihop Shrinking scheme in two ways:

- *Initiation of the mechanism:* The Multihop Shrinking mechanism was originally designed to run in a periodic manner, amortized against data traffic. We modify it here to be event-triggered, so that the whole mechanism, LocOpt, is initiated only when necessary.

- *Two new fields in routing tables:* We have showed in the previous chapter that the original Multihop Shrinking does not harm loop-free property of the

routing protocol. In order to satisfy this condition in LocOpt mechanism, we need to take extra precautions in route optimization process. towards this end, we introduce a new parameter named *shrink sequence number*, (SSN), which is incremented by the source node every time a new route optimization is initiated. To keep this number as well as source node's IP address, nodes open two new fields in their routing tables.

Initiation of the process: In contrast to the original data-driven operation (see Chapter 2), the Multihop Shrinking operations here are activated only *after* completion of "Local Recovery" and "Distance Update" (see Subsection 3.7.2.1). More precisely, Multihop Shrinking is initiated when the source node (e.g. node S) of a path receives a DISTANCE packet, which happens because of the distance update mechanism that is initiated post local recovery. Once the source node receives a DISTANCE packet, it increments the shrink sequence number in its routing table for the destination node (i.e. node T), and then initiates the shrinking mechanism by unicasting a special SHRINK-0 packet to the next hop (i.e. node A) on the route.

A Shrink-X packet (where X = 0, 1, 2)always contains the following fields:

- **sender**: The IP address of the sender of the Shrink packet,
- **next-hop**: The IP address of the next hop,
- **final-destination**: The IP address of the final destination,
- **hops-to-destination**: The number of hops to the final destination,
- **flag**: Can be 0, 1, or 2.

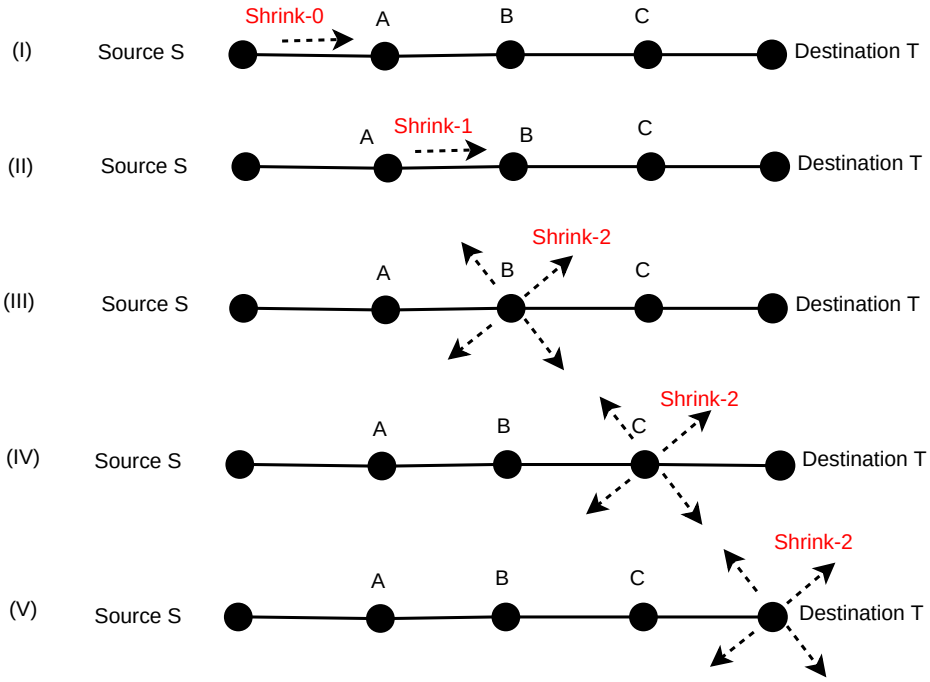


Figure 3.6: How Multihop Shrinking works.

- **shrink sequence no**: The sequence number of the operation which is incremented by the initiator of the process (i.e. the source node) at each new operation.

- **source**: The IP address of the source node.

So, for example, in the SHRINK-0 packet set by S: sender = S, next-hop = A, final-destination = T, hops-to-destination = 4, flag = 0, and source = A. Once the second hop (e.g. node A) receives the SHRINK-0 packet, it retrieves the information from the received packet, and updates corresponding fields in its routing table (i.e. shrink sequence number and source node address for the specified destination entry). Then, it prepares a similar packet, SHRINK-1, and unicasts it to the next hop (i.e. the third node) along the route. When the third node (i.e. node B) receives a SHRINK-1 packet, again it updates related fields in its routing table for this destination entry,

and then prepares a similar packet, SHRINK-2. Nonetheless, this time the node *broadcasts* the SHRINK-2 packet with TTL of 1 (Notice that SHRINK-0 and SHRINK-1 were unicast packets). Likewise, the process progresses similarly towards the destination node. Thus, each node which sends SHRINK packet along the route updates shrink sequence numbers and corresponding source node IP in its routing table for this destination. Notice that the downstream nodes, which has not yet broadcasted any SHRINK packet, would maintain stale shrink sequence numbers until their turn to come for SHRINK packet processing and broadcasting. Hereby, it is provided that only the upstream nodes relative to originator of a SHRINK packet becomes effective in route optimization process. As we will notice later, this provides loop-free operations in LocOpt scheme.

From the vantage point of other nodes: A node which receives a Shrink-2 packet performs actions depending on its relative placement on the connection with respect to sender of the Shrink packet. A node, upon receiving the Shrink-2 packet, determines its relationship to the Shrink-2 packet, selected from the following set of five mutually exclusive classes:

- *The next hop:* A node identifies itself to be the *next hop* by noting that its own IP is the one specified in the next-hop field of Shrink-2 packet it received. In that case, it modifies the received Shrink-2 packet by updating the related fields using the information in its routing table, and then broadcasts the updated Shrink-2 packet with TTL of 1.

- *Further downstream hops:* A node recognizes itself to be in this class if its routing table indicates that the shrink sequence number it has for corresponding source destination pair is less than the shrink sequence number specified in

received Shrink-2 packet for the same source destination pair. Nodes in this category discard the received Shrink-2 packet.

- *The previous hop:* A node identifies itself to be the *previous hop* when it realizes that in its routing table, the sender of the Shrink-2 packet is listed as the next hop to the final-destination. This node just discards the received Shrink-2 packet.

- *Further upstream nodes:* A node recognizes itself to be in this class if its routing table indicates that the following two conditions are satisfied.

  - The tuple of shrink sequence number and source node address is the same with the one specified in received Shrink-2 packet.
  - The hop count to the destination is greater than the value in the hops-to-destination field of the received Shrink-2 packet

  If this is the case, then it can be concluded that there may be a shortcut available between this node and the sender of Shrink-2 message. However, the quality of this new hypothesized link may not be good enough to warrant changing the routing table. Therefore, before doing any update to the routing table, the quality of the prospective new link is checked by looking at the signal strength at which the Shrink-2 packet was received. If the received signal strength is greater than a predefined threshold level, then the node updates its routing table in such a way that the next hop for the final-destination (as specified in the Shrink-2 packet) is replaced with the address of the sender the Shrink-2 packet.

- *Irrelevant nodes:* When a node receives Shrink-2 packet, it but there is no next hop in its routing table for the final-destination specified in the Shrink-2

packet, then the node classifies itself as irrelevant to the Shrink-2 packet, which is simply ignored.

Every node receiving a Shrink-2 packet falls into precisely one of the above five classes, and behaves as specified. In practice, there can be many data flows (i.e. many connections) between different source/destination pairs, all active simultaneously. Such a situation does not alter the operation of proposed mechanism, since each node responds to different Shrink messages arising from different transient connections independently. No further state needs to be maintained at each node than what is mandated in AODVs routing table format.

Usage of shrink sequence number also provides to distinguish each route optimization process belong to different source-destination pairs. In other words, if there are many source nodes sending data packets to the same destination, route optimization packets (i.e. SHRINK packets) do not affect disjoint parts of these connections, that is to say, route optimization process only affects the corresponding connection.

3.8 Performance Analysis via Simulation

3.8.1 Simulation Environment

The simulation environment was described in Section 2.9.1 (pp. 42).

3.8.2 Experimental Setup

The performance of the original AODV and the proposed schemes are compared for the following network size, mobility models, and traffic/connection pattern as follows:

Networks: Several network sizes are investigated, comprised of between 50 and 100 nodes. Node density is kept fixed at 50 nodes per 700 m x 700 m. Initial placement of nodes is uniformly random.

Traffic Patterns: The traffic connections are initiated between randomly chosen source and destination pairs chosen uniformly from the nodes. The number of connections varied between 1 and 50, depending on the type of experiment. Traffic sources generate constant bit rate (CBR) traffic consisting of packets of size 512 bytes, at a rate of 4 packets per second.

Mobility Levels: Maximum velocity of nodes is changed from 5 m/s to 25 m/s, representing the lowest mobility level to the highest mobility level respectively. The mobility model adopted is described in the following subsection.

3.8.3 Mobility Models

The mobility model To investigate the performance of the proposed mechanism under different mobility levels, we modified random waypoint (RWP) mobility model . It is described in detail in Section 2.9.3 (pp. 45).

3.8.4 Performance Metrics

The following four metrics are evaluated:

- *Normalized Routing load* (NRL)– The number of routing-related control packets transmitted *per data packet delivered to its destination.*
- *Packet Delivery Fraction* (PDF)– The percentage of data packets submitted at a source, that were delivered to their intended destination.
- *Path Stretch* (PS)– For each data packet delivered to its destination, we note both the number of hops that packet traveled and the length of the optimal source-destination path at the time of packet delivery. Then we calculate the discrepancy, both as ratio and as a difference.
- *Number of Flooding Events*– (NFE) The number of network-wide route discovery processes (i.e. floodings) during simulation period.

Trials: To increase our confidence in conclusions drawn from the analysis of simulations, we repeated up to 10 trial experiments for each setting of the independent variable (with mobility plans and traffic patterns generated as above). We then computed the mean and standard deviations of the performance metrics (dependent variables), and examined how these varied as the independent variable was changed.

3.8.5 Results and Analysis

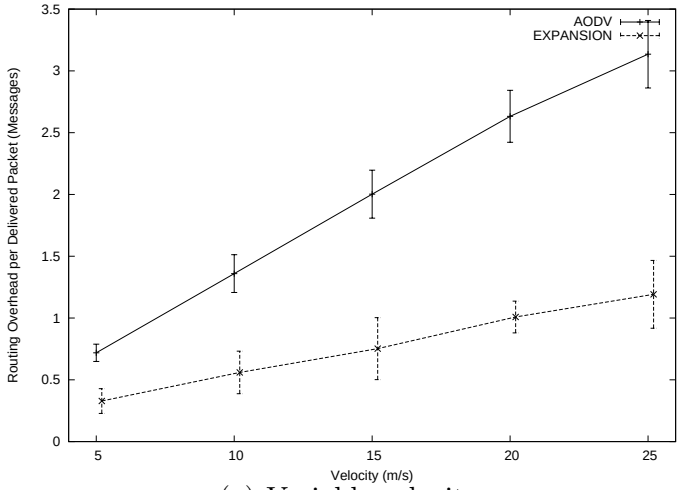
In this section, we present and analyze experimental simulation results of our proposed schemes from perspective of the four performance metrics described above.

3.8.5.1 Expansion

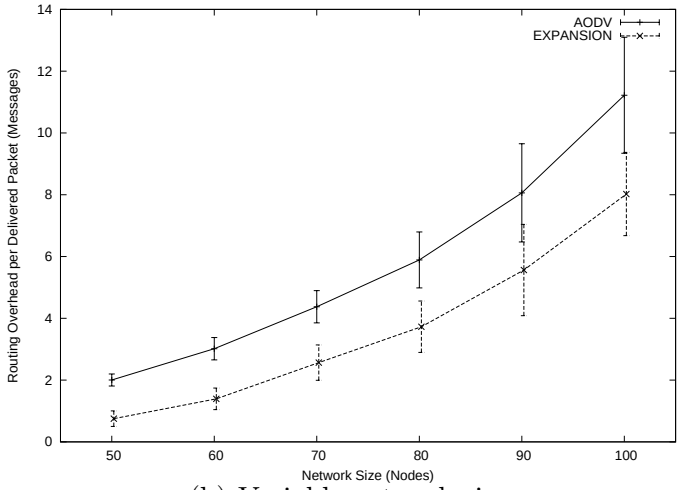
Normalized Routing Load (NRL):

In this set of experiments, we investigated NRL of both pure AODV and the extended version of AODV with the Expansion scheme under different mobility levels, network sizes, and traffic loads. Figure 3.7(a) indicates the relationship between NRL and different mobility levels for 50-node networks with 10 data connections. According to the figure, NRL is linearly growing for both mechanisms as mobility level increases. For example, NRL of pure AODV ranges from 0.72 to 3.13, depending on mobility level, while NRL of AODV+Expansion varies between 0.33 and 1.19. From this values and Figure 3.7(a), it is clear that the proposed scheme reduces NRL between 54% and 62%, depending on mobility level settings, and that the gains are more pronounced at higher mobility levels.

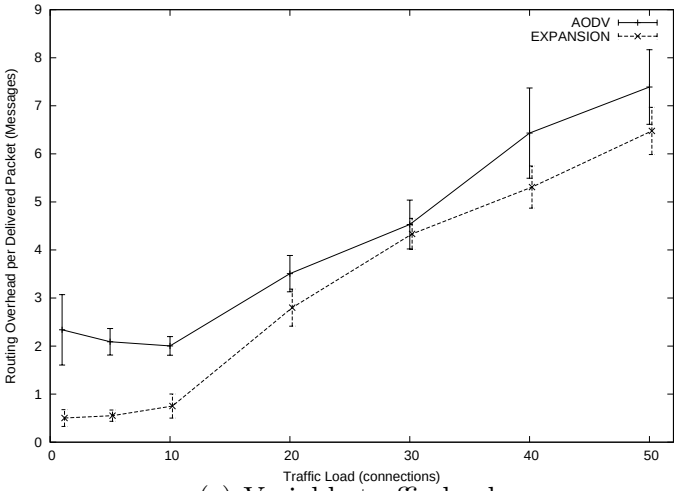
In the second set of NRL experiments, we consider the effect of network size, while mobility level setting is 15 m/s and data load is 10 connections. As seen in Figure 3.7(b), NRL of both compared mechanisms increases as network size gets larger, obviously because of the fact that route discovery operation in larger networks is much more expensive than it is in small networks. According to the figure, NRL of pure AODV in networks including 50, 60, 70, 80, 90, and 100 nodes is 2.0, 3.0, 4.37, 5.89, 8.06, and 11.22, on average, respectively. When the Expansion scheme is in effect, the corresponding NRLs are 0.75, 1.39, 2.57, 3.73, 5.56, and 8.02 respectively. These numbers indicate that the Expansion scheme reduces NRL between 62.5% and 29% depending on network size. The relative advantage of Expansion decreases for larger networks because updating hop-count information, which is a part of the



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.7: Normalized Routing Load (NRL)

Expansion scheme, becomes more expensive where connection length increases.

In the third set of NRL experiments, we consider the effect of data traffic load– reflected in the number of active connections, for 50-node networks at mobility level settings of 15 m/s. As seen in Figure 3.7(c), NRL is linearly increasing as data traffic load becomes higher than 10 connections. For example, NRL of pure AODV is around 2.0 when traffic load is equal or less than 10 connections, while it grows to 7.39 when traffic load is increased up to 50 connections. The same pattern is also seen when the Expansion scheme is in effect. The reason for the increase in NRL as data load gets higher is the reduction in data packet delivery ratio[5]. This interpretation will be confirmed in the results of a subsequent set of experiments concerning packet delivery fraction. It is obvious in the figure that the Expansion scheme reduces NRL when data load is equal or less than 20 connections. For higher data loads, we cannot reliably say this due to overlapping error bars.

Packet Delivery Fraction (PDF):

In this set of experiments, we investigated possible impact of the proposed *Expansion* scheme on packet delivery fraction (PDF) under different mobility levels, network sizes, and traffic loads. Figure 3.8(a) shows PDF for pure AODV and AODV+Expansion at different mobility levels, when the network consists of 50 nodes and 10 data connections. According to the figure, pure AODV has packet delivery ratio, on average, between 98.8% and 94.6%, depending on mobility level, and is seen to be linearly decreasing as mobility level increases. When the *Expansion* scheme is applied, PDF takes values, on average, between 99% and 97.6%. It is clear from

5. Recall that NRL is the normalized control traffic overhead with respect to packet deliver ratio.

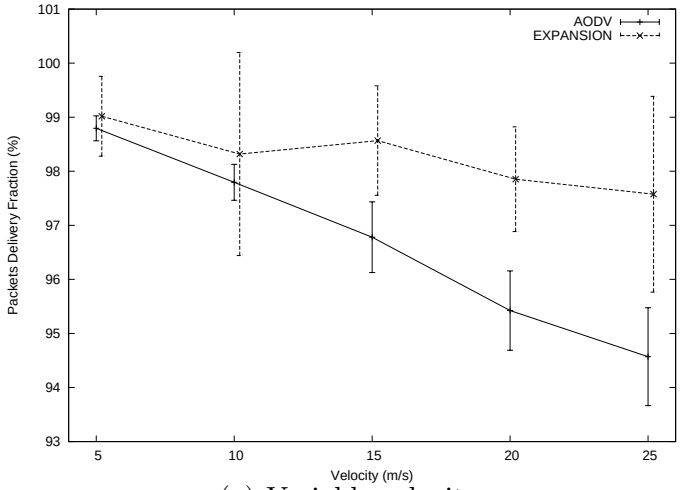
the figure that the proposed *Expansion* mechanism improves PDF at high mobility settings such as 15 m/s or higher. Due to overlapping error bars at low mobility settings (i.e. 5 m/s and 10 m/s), it is difficult to make reliably comparisons based on given mean values for this mobility settings. The proposed scheme achieves this improvement by preventing anticipated link breakages which would cause packet losses.

In the second set of PDF experiments, we consider the effect of network size. As seen in Figure 3.8(b), PDF decreases for both pure AODV and AODV+Expansion as network size gets larger. This is because (i) the larger the network, the more contention (assuming a fixed node density) and (ii) the larger the network, the longer the path between random source-destination pairs, and hence, the greater the packet losses on any given connection. Since the mechanisms have similar mean values as well as overlapping error bars, they are not comparable in this set of experiments.

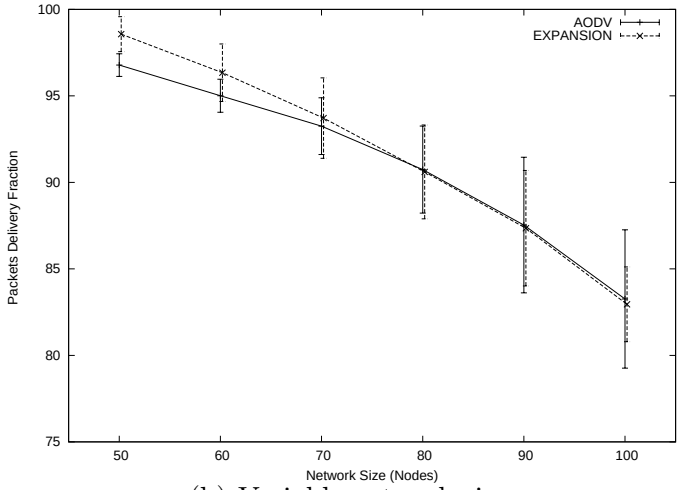
In the third set of PDF experiments, we investigate the effect of increased data load. According to Figure 3.8(c), PDF decreases as traffic load increases for both schemes. The cause for this is increasing contention levels at the MAC layer. It is difficult to compare performance of the mechanisms when data load is between 10 and 40 connections since mean values of the curves are too close to each other, as well as error bars overlap. For the other data loads, it can be deduced from the figure that the *Expansion* scheme improves PDF by up to 17%, and that it yields almost 100% PDF at low data loads.

Path Stretch (PS):

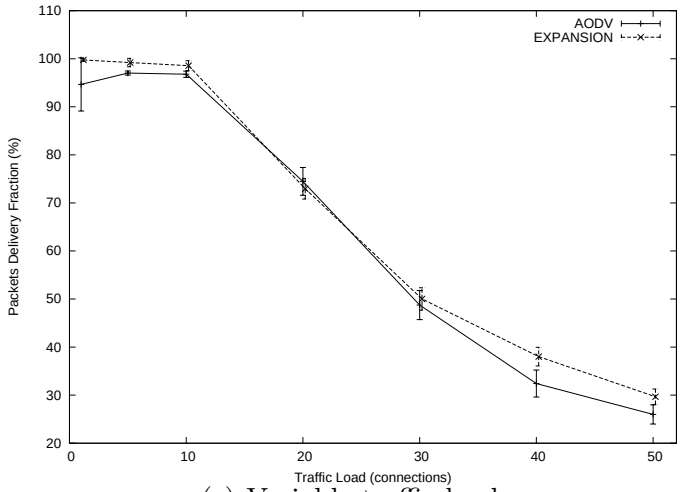
We see from the preceding analysis that the *Expansion* scheme is capable of signifi-



(a) Variable velocity



(b) Variable network size



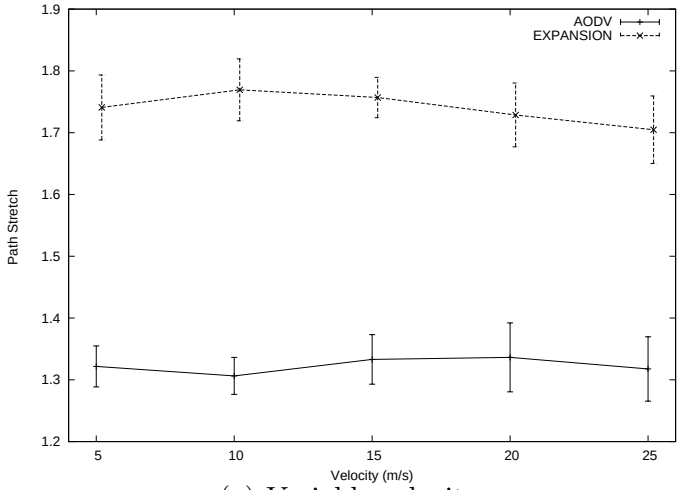
(c) Variable traffic load

Figure 3.8: Packet Delivery Fraction (PDF)

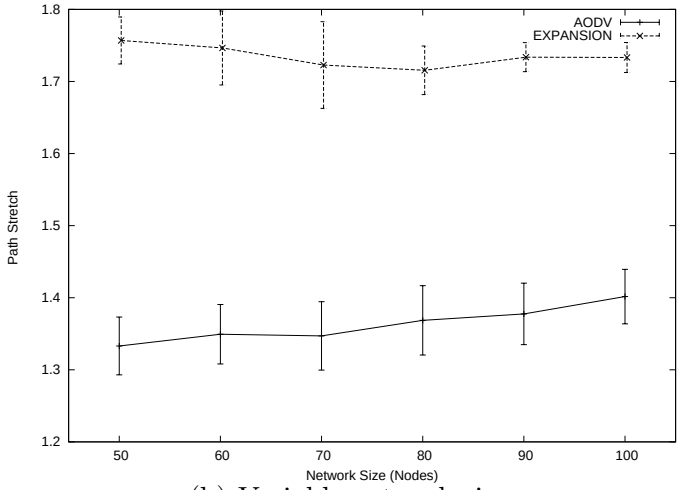
cantly reducing normalized traffic overhead, without compromising on packet delivery fraction. The question is, at what price is this achieved? Clearly, since the scheme relies on path expansion, the scheme will be worse than pure AODV when it comes to path optimality. To shed light on this issue, in the next set of experiments, we investigate path optimality (i.e. path stretch) in terms of hop count, for both pure AODV and AODV+Expansion under different mobility levels, network sizes, and data traffic loads. Notice that path lengths measured for each data packet delivered are *normalized with respect to optimal path length*, which would be calculated according to Dijkstra's shortest path algorithm, at the instantaneous time of packet delivery.

Figure 3.9(a) shows path stretch at different mobility levels, when network consists of 50 nodes and 10 data connections. It is seen from the figure that pure AODV has path stretch around 1.3, while AODV+Expansion has path stretch around 1.7, meaning that the Expansion scheme increases path lengths about 30%. The results are to be expected, since the *Expansion* scheme causes packets to travel on paths even further from optimal than those selected by AODV–this is because the *Expand* operation (recall that it is responsible for inserting bridge node into endpoints of a weak link) favors forming longer paths over incurring global route discovery when connection links fail. The *Shrink* operation (recall that it shortens paths if possible) balances the *Expand* operation to some extent. It can also be deduced from the figure that mobility level has no significant impact on path stretch.

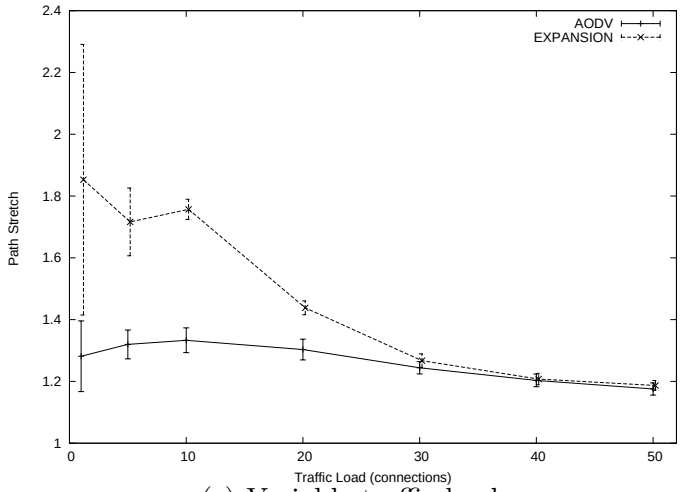
In the second set of path optimality experiments, we consider impact of network sizes. Figure 3.9(b) depicts path stretch for both pure AODV and AODV+Expansion under different network sizes. It is seen from the figure that path stretch of pure AODV increases slightly from 1.33 to 1.40 as network size gets larger, whereas path stretch



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.9: Path Stretch (PS)

of AODV+Expansion decreases a little bit 1.76 to 1.73. This may be because of the fact that it is more likely to perform a successful *Shrink* operation in large networks.

In the third set of path optimality experiments, we investigate the impact of changes in data traffic load. Figure 3.9(c) indicates path stretch for both pure AODV and AODV+Expansion under different traffic loads. According to the figure, path stretch of pure *AODV* increases slightly as data load changes from 1 connections to 10 connections, then it gradually decreases and stabilize at around path stretch of 1.18. On the other hand, path stretch of AODV+Expansion is about 1.85 when the traffic load is 1 connection, and then it significantly decreases as traffic load increases, finally converging to a path stretch of about 1.19. A possible explanation of this behavior is that the number of successfully completed "expand" operations decreases greatly as data load of the network increases because of too much contention. It is difficult to compare performance of the mechanism when data load is higher than 30 connections since their mean values are too close to each other, and also corresponding error bars overlap. However, when data load is less than 30 connections, it can be deduced from the figure that pure AODV has better path stretch than AODV+Expansion.

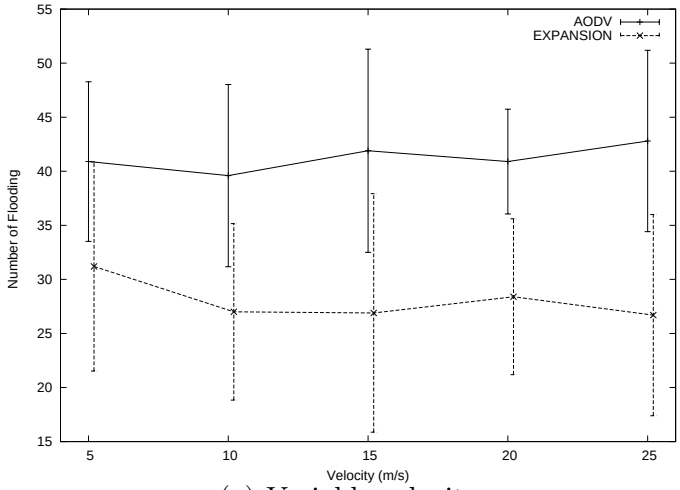
Number of Flooding Events (NFE):

In this set of experiments, we investigate possible impact of the *Expansion* scheme on the expected number of global route discovery attempts (i.e. flooding) in a network. Figure 3.10(a) indicates the number of flooding events for both pure AODV and AODV+Expansion under different mobility levels. According to the figure, there are about 41 flooding, on average, realized in case of pure AODV, while this number is around 27 in case of AODV+Expansion, regardless of mobility level. Nevertheless, It may not be reliable to compare the mechanisms based on these mean values since the

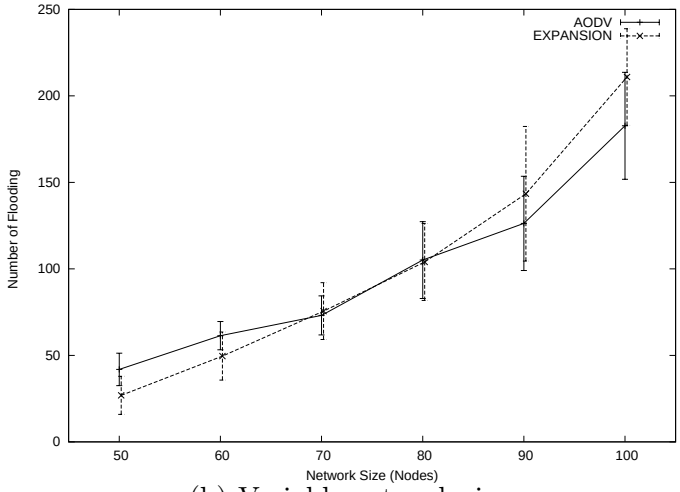
error bars in the graph are partly overlap to each other. Notice that the curves in the graph are almost flat with respect to mobility level, which may incorrectly imply that mobility level has no impact on the number of flooding attempt. In fact, mobility level greatly affects the number of flooding. The reason of having flat curves in this graph is that simulation duration in the experiments is adjusted inversely proportional with the maximum velocity level, as mentioned in Section 3.8.3.

In the second set of "the number of flooding" experiments, we consider impact of network size. Figure 3.10(b) depicts the number of flooding attempts for both compared mechanisms under different network sizes. It can be seen in the figure that the number of flooding increases as network size gets larger for both the mechanisms. This is because of that expected connection lifetime becomes shorter in large networks due to longer paths. Longer paths are more fragile with respect to node mobility, which results in more frequent route discovery processes. Since the error bars in the graph overlap each other, we can not make reliable comparative conclusions based on mean performance values of experiment trials.

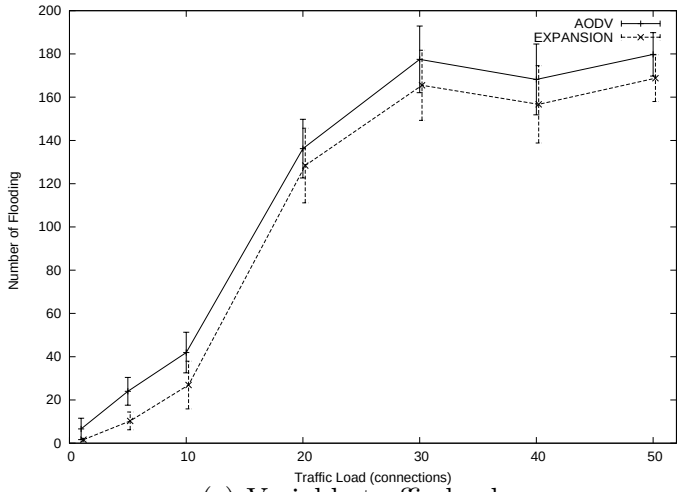
In the third set of "the number of flooding" experiments, we investigate impact of data traffic load. Figure 3.10(c) indicates relationship between the number of flooding events and data traffic load for both proposed mechanisms in a network including 50 nodes, when maximum velocity of nodes is set to 15 m/s. As can be seen from the figure, the number of flooding events increases for both the mechanisms, until traffic load reaches 30 connection. After this point, the number of flooding events appears to remain fairly constant, even as data load is increased further. Again, since the error bars in the graph overlap each other, unfortunately, we can not make reliable comparative conclusions based on mean performance values of experiment trials.



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

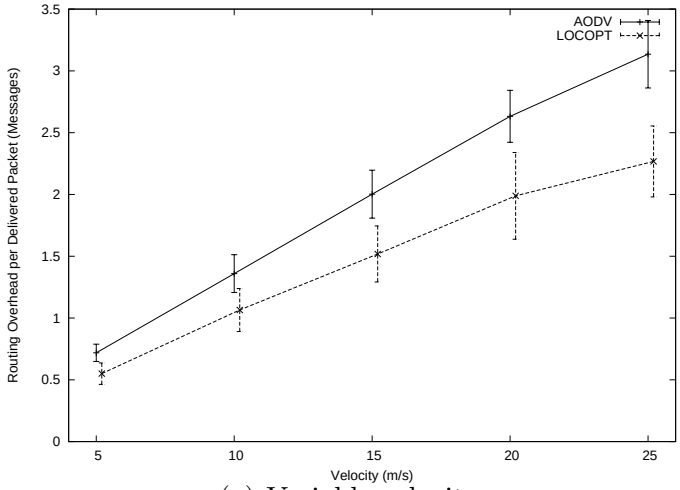
Figure 3.10: Number of Flooding Events (NFE)

3.8.5.2 Local Recovery

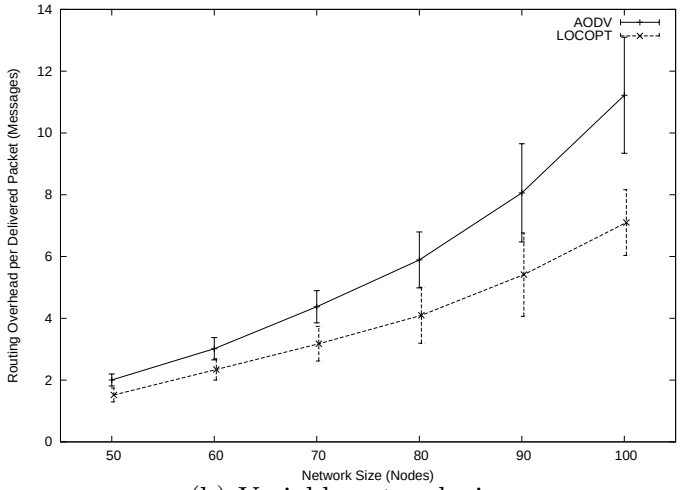
Normalized Routing Load (NRL):

We begin by considering normalized routing load (NRL), that is, the number of routing-related control packets transmitted *per data packet delivered to its destination.* We investigated NRL of pure AODV and AODV+LocOpt under different assumptions of node mobility level, network size, and traffic load. Figure 3.11(a) depicts the relationship between NRL and mobility levels. According to the figure, for example, when nodes are moving at 5 m/s and 25 m/s, AODV utilizes 0.72 and 3.13 control packets per data packet delivered respectively, while AODV+LocOpt uses only 0.55 and 2.27, which means 24% and 28% reduction in NRL. This is because the proposed mechanism decreases the number of global route discovery attempts performed in the network, as will be shown in the subsequent set of experimental results. It is clear from the figure that the proposed scheme reduces NRL of the network regardless of node mobility level. More importantly, as node mobility increases, the advantage of the proposed scheme over AODV *increases*, implying that the proposed solution scales as networks become increasingly more mobile.

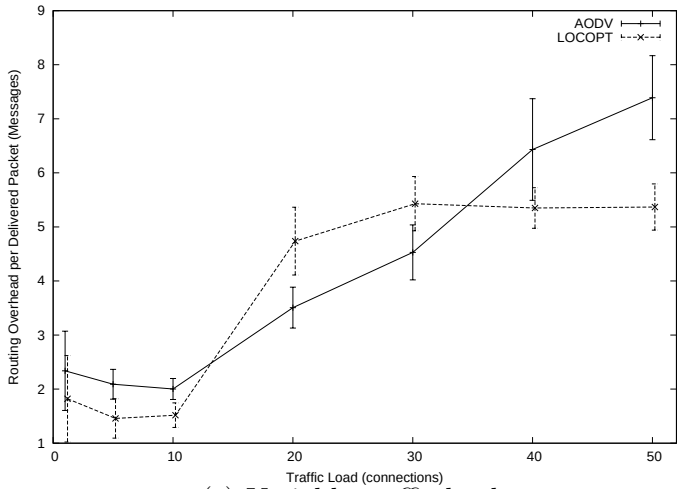
In the second set of NRL experiments, we consider the effect of network size. As seen in Figure 3.11(b), NRL of both the mechanisms rises as mobility level increases. It can be deduced from the figure that the relative advantage of the LocOpt scheme is significant and *increases* as the network size gets larger. We note that because we keep node density fixed in our experiments, increasing network size implicitly increases the geometric extent of the network. When the network has 50 nodes, AODV uses 1.32 times as many control packets (per delivered data packet) as LocOpt does; but when



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

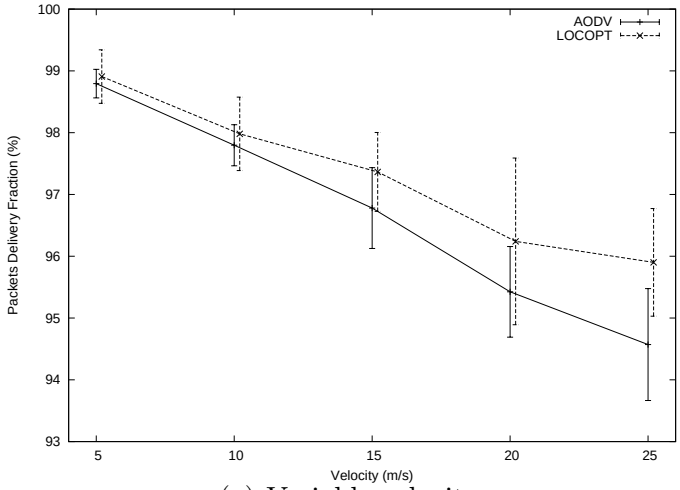
Figure 3.11: Normalized Routing Load (NRL)

the network has 100 nodes, AODV uses more than 1.58 times as many control packets (per delivered data packet) as LocOpt does. The relative advantage gained by the LocOpt scheme increases for larger networks because the importance of local recovery is more critical in large networks where the cost of global route discovery are much more expensive.

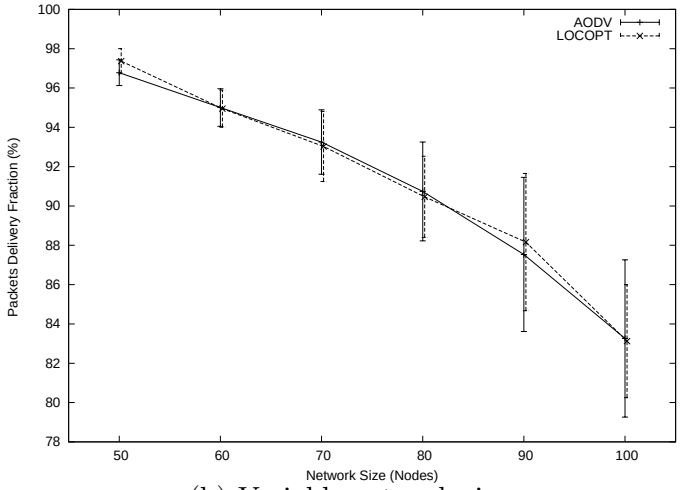
In the third set of NRL experiments, we consider the effect of data traffic load. As seen in Figure 3.11(c), there are overlapping error bars in the figure, and therefore it is difficult to make reliable comparisons based on mean performance values of experiment trials. Still it can be deduced from the figure that NRL remains almost fixed until data loads of 10 connections for both mechanisms, and then increases at higher data loads, which is a sign that network is saturated at data load of 10 connections.

Packet Delivery Fraction (PDF):

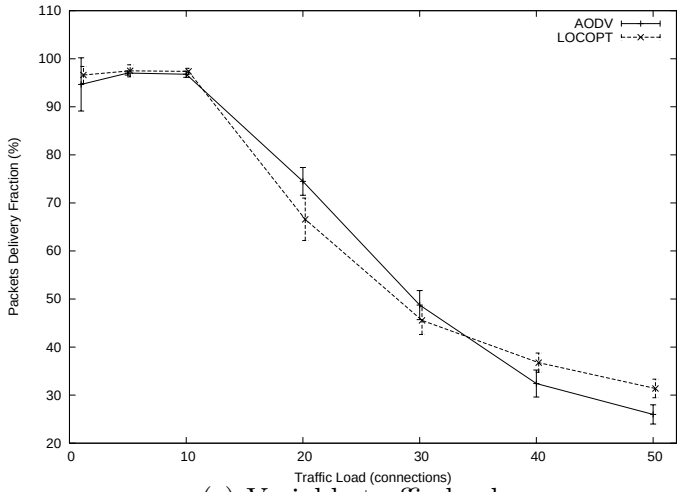
In this set of experiments, we investigated possible impact of the proposed *LocOpt* scheme on packet delivery fraction (PDF) under different mobility levels, network sizes, and traffic loads. Figure 3.12(a) shows PDF for pure AODV and AODV+LocOpt at different mobility levels, when the network consists of 50 nodes and 10 data connections. According to the figure, pure AODV has packet delivery ratio, on average, between 94.6% and 98.8%, depending on mobility level, linearly decreasing as maximum velocity level is increasing. When the *LocOpt* scheme is applied, PDF takes values, on average, ranging from 95.9% to 98.9%. Yet these mean values are not enough to make reliable comparative conclusions due to overlapping error bars as seen in the figure.



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.12: Packet Delivery Fraction (PDF)

In the second set of PDF experiments, we consider the impact of network sizes. As seen in Figure 3.12(b), PDF decreases for both pure AODV and AODV+LocOpt as network size gets larger. This is due to increasing contention and packet losses. Again, we can not provide any comparative conclusion due to overlapping error bars.

In the third set of PDF experiments, we investigate the effect of different data loads. According to Figure 3.12(c), PDF remains flat until data load of 10 connections, and then decreases for both the mechanisms, obviously because of increasing contention. Data load of 10 connections is emerged as saturation limit of the networks including 50 nodes.

Path Stretch (PS):

We see from the preceding analysis that the *LocOpt* scheme is capable of significantly reducing normalized traffic overhead without compromising on packet delivery fraction. We can have our cake apparently, but can we eat it too?

Another important performance metric for a local recovery protocol is how it affects the path optimality. Most if not all previous attempts at crafting a local recovery mechanism have failed to address the issue of route degradation. This is a well-known side effect of local recovery [8, 35]. To investigate this issue, in this set of experiments, we calculated, for each data packet delivered to its destination, the number of hops that the packet traveled. We compared this with the number of hops on the optimal source-destination path (at the time of packet delivery). The ratio of the first quantity to the second was taken as the normalized path length for the packet. We computed the normalized path length for all delivered packets, and showed the mean and standard deviations of these values. Figure 3.13(a) shows that

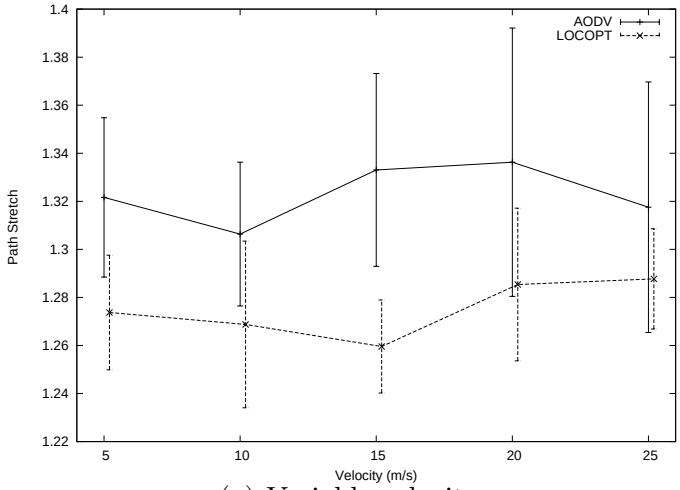
pure AODV has higher path stretch, on average, than AODV+LocOpt. For example, when maximum velocity is 5 m/s, path stretch of AODV is about 1.32, as it is about 1.27 in case of AODV+LocOpt. Nevertheless, this are not enough to conclude that the proposed LocOpt scheme reduces path stretch at all mobility settings, because the error bars partly overlap.

In the second set of path optimality experiments, we consider the impact of network size. As seen in Figure 3.13(b), AODV+LocOpt outperforms pure AODV in terms of path stretch even in large networks. For example, when a network consists of 100 nodes, path stretch of pure AODV is 1.40, as it is 1.28 in the existence of *LocOpt*.

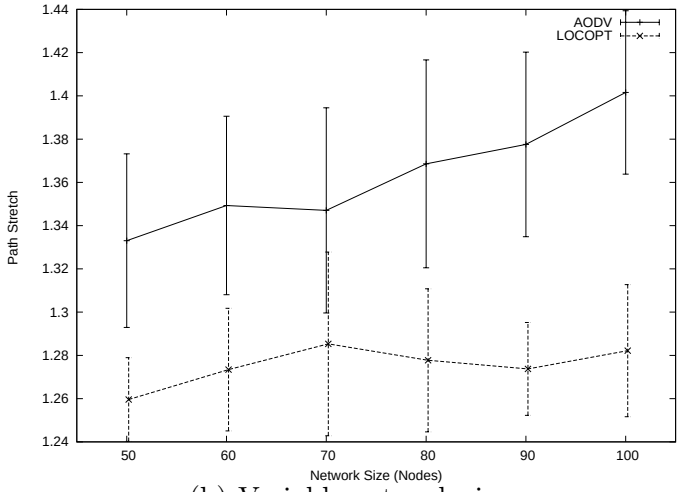
In the third set of path optimality experiments, we investigate the effect of data traffic load. As can be seen from Figure 3.13(c), AODV+LocOpt outperforms pure AODV especially when there are 10 or more connections as data loads, and achieves better advantage at higher data loads, indicating scalability.

Number of Flooding Events (NFE):

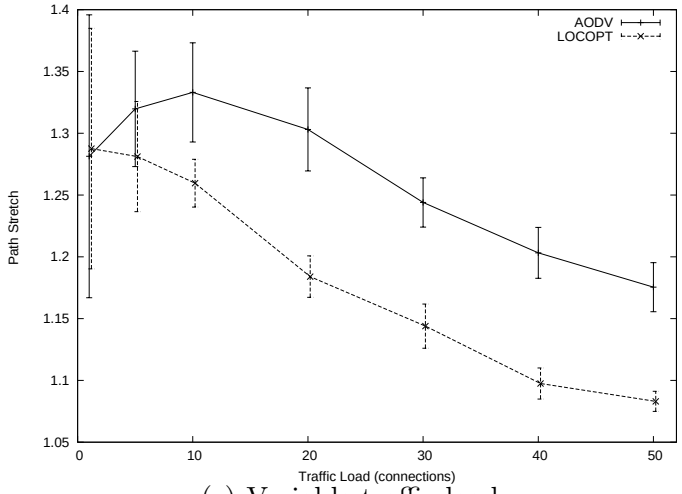
In this set of experiments, we investigate possible impact of the *LocOpt* scheme on the expected number of global route discovery attempts (i.e. flooding events) in a network. Figure 3.14(a) indicates the number of flooding events for both pure AODV and AODV+LocOpt under different mobility levels. According to the figure, there are about 41 flooding, on average, realized in case of pure AODV, while this number becomes only 18 in case of AODV+LocOpt. This is true regardless of mobility levels. It can be deduced from these results that the *LocOpt* scheme reduces the number of flooding approximately 56%. Notice that the curves in the graph are almost flat with respect to mobility level, which may be incorrectly regarded as implying that mobility



(a) Variable velocity



(b) Variable network size



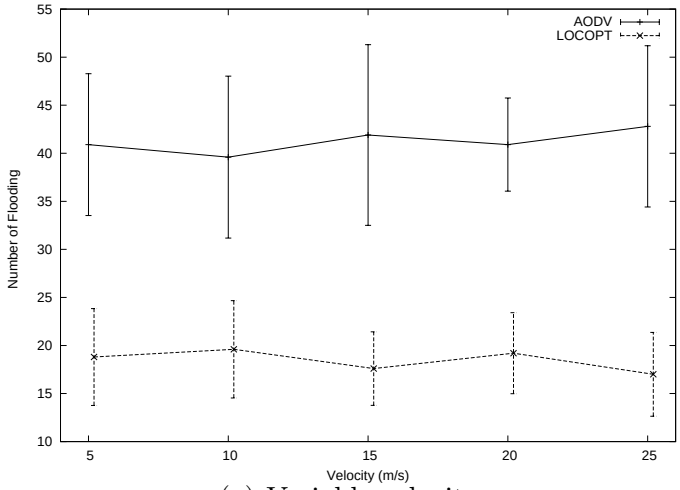
(c) Variable traffic load

Figure 3.13: Path Stretch (PS)

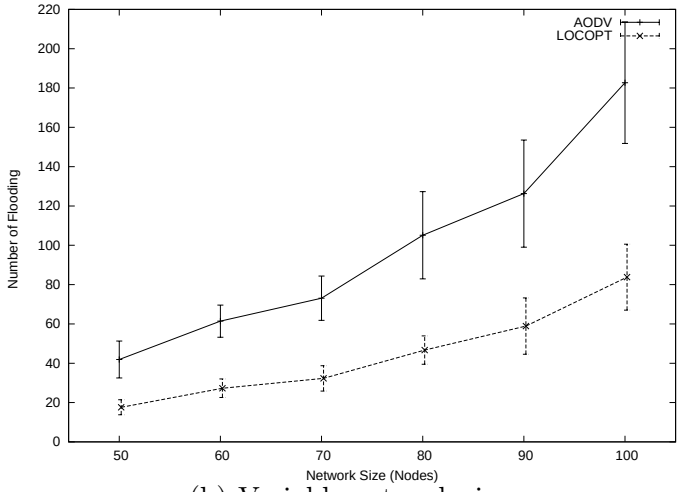
level has no impact on the number of flooding attempt. In fact, mobility level greatly affects the number of flooding events. The reason we obtain flat curves in this graph is that simulation duration in the experiments is adjusted inversely proportional with the maximum velocity level, as described in Section 3.8.3.

In the second set of experiments in this section, we consider impact of network sizes. Figure 3.14(b) depicts the number of flooding attempts for both compared mechanisms under different network sizes. According to the figure, in case of pure AODV, there are 42, 61, 73, 105, 126, and 183 flooding, on average, for network sizes of 50, 60, 70, 80, 90, and 100 respectively. When LocOpt is in effect, these values are 18, 27, 32, 47, 59, and 84 respectively, which means there is almost a fixed gain by about 56% in terms of reduction of the number of flooding.

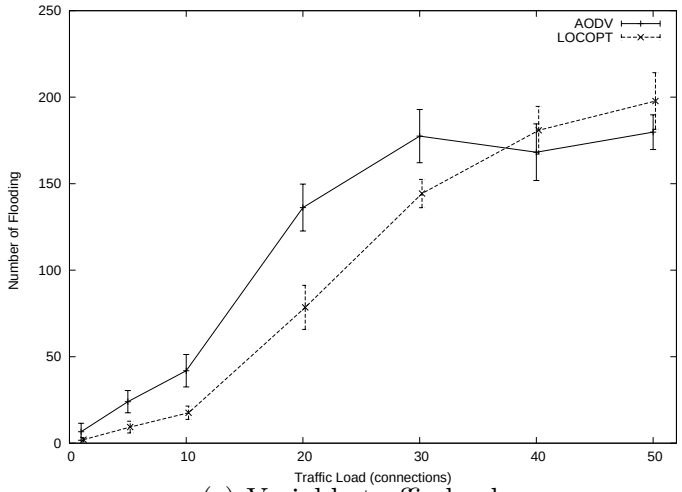
In the third set of experiments on "the number of flooding events", we investigate the impact of data traffic load parameter. Figure 3.14(c) indicates relationship between the number of flooding events and data traffic load for both compared mechanisms in networks including 50 nodes, when mobility setting is 15 m/s. It can be deduced from the figure that the proposed LocOpt scheme reduces the number of flooding performed in the network, when data load is less than 30 connections. Notice that 30 connections represent extremely high loads, given that networks include 50 nodes. It is not reliable to compare mechanisms based on mean values at higher loads due to overlapping error bars.



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.14: Number of Flooding Events (NFE)

3.8.5.3 Expansion versus LocOpt

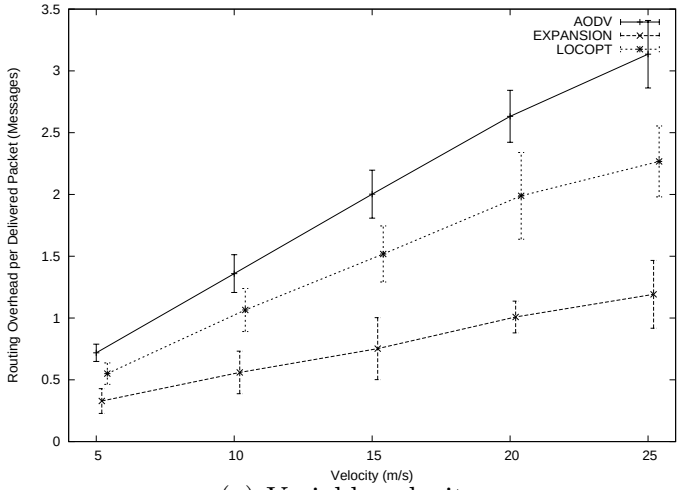
In this part, we compare the proposed *Expansion* scheme with the proposed *LocOpt* scheme from the point of several performance metrics. Since we already analyzed performance of pure AODV with respect to both the proposed schemes in the preceding parts, here we just compare performance of the *Expansion* and the *LocOpt* against each other.

Normalized Routing Load (NRL):

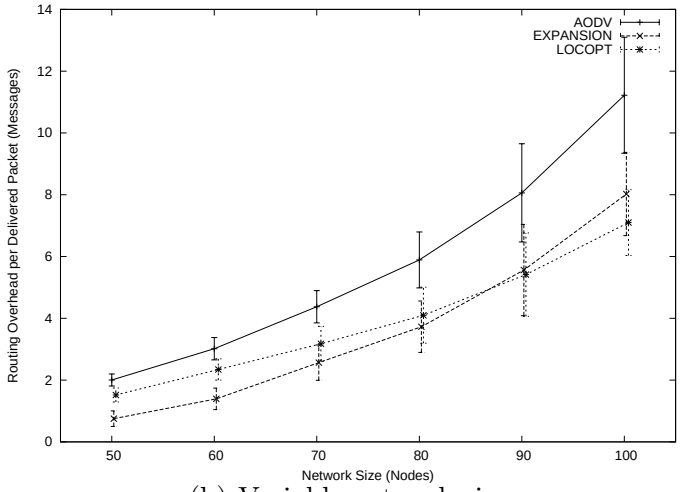
In this set of experiments, we investigate NRL of AODV+Expansion and AODV+LocOpt under different assumptions of node mobility level, network size, and data load. Figure 3.15(a) depicts the relationship between NRL and mobility levels. It is clear from the figure that the proposed *LocOpt* scheme yields more NRL than the proposed *Expansion* scheme does, regardless of node mobility level. For example, when nodes are moving at 25 m/s, AODV+LocOpt utilizes 2.27 control packets per data packet delivered, while AODV+Expansion uses only 1.50. This is because the *LocOpt* scheme additionally performs a route optimization process.

Figure 3.15(b) shows NRL with respect to increasing network size. It can be deduced from the figure that the *Expansion* scheme yields less NRL than what the *LocOpt* does when network size is 50-node and 60-node. However, in larger network sizes, it is not possible to make reliable comparisons based on mean values of experiment trials due to overlapping error bars.

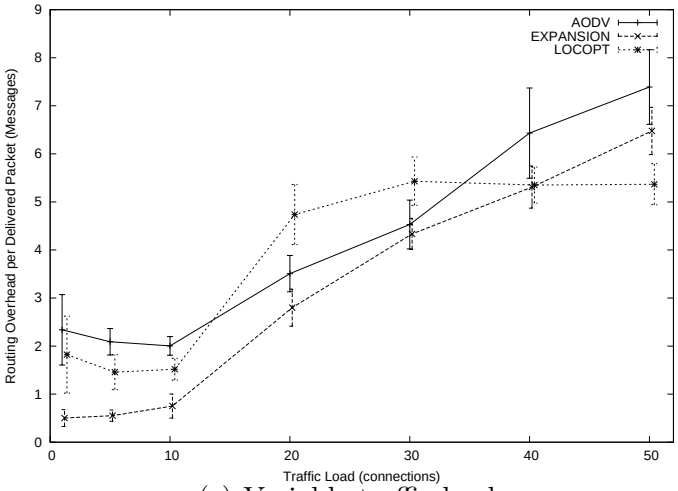
Figure 3.15(c) indicates NRL as data traffic load increases. The curves belong to the *LocOpt* and the *Expansion* schemes cross each other at around traffic load of 40



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.15: Normalized Routing Load (NRL)

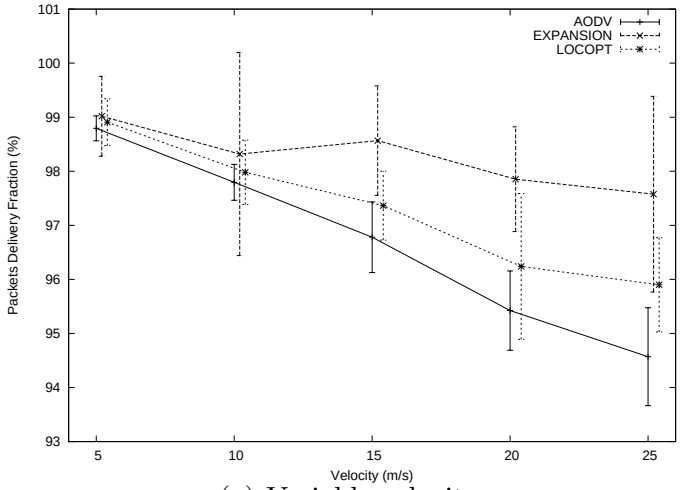
connections. When traffic load is less than 40 connections, the *Expansion* scheme has less NRL than the *LocOpt* does by up to 73%, but at extremely high traffic loads (such as 50 connections in a 50-node network) the *LocOpt* has lower NRL than the *Expansion* does by about 17%.

Packet Delivery Fraction (PDF):

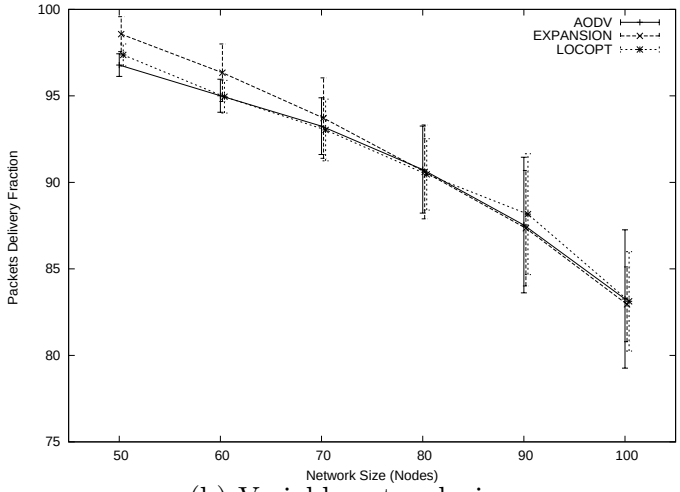
In this set of experiments, we compare performance of the proposed *Expansion* and *LocOpt* schemes on packet delivery fraction (PDF) under different mobility levels, network sizes, and traffic loads. Figure 3.16(a) shows PDF for pure AODV, AODV+Expansion, and AODV+LocOpt at different mobility levels, when network consists of 50 nodes and 10 data connections. According to the figure, the *Expansion* has higher PDF, on average, than what the *LocOpt* has, regardless of mobility level. However, this is not enough to make reliable comparative conclusions since corresponding error bars overlap.

In the second set of PDF experiments, we consider the impact of network sizes. As seen in Figure 3.16(b), PDF decreases for all schemes as network size gets larger due to increasing contention and packet losses. It is difficult to compare mechanism since mean values are too close to each other, as well as corresponding error bars overlap.

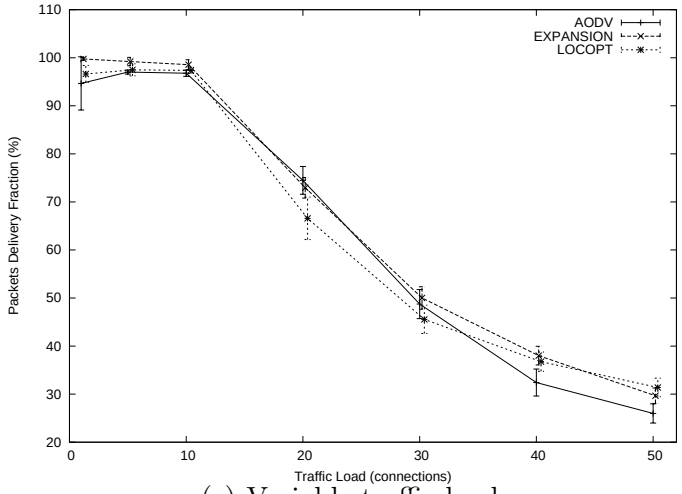
In the third set of PDF experiments, we investigate the effect of data load of networks. According to Figure 3.16(c), PDF decreases as data load increases for both mechanisms, obviously because of increasing contention. It is not possible to compare both the proposed schemes when data load is higher than 30 connections due to overlapping error bars. However, it can be deduced from the figure that the *Expansion* scheme yields better PDF than *LocOpt* when data loads is less than 30 connections.



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.16: Packet Delivery Fraction (PDF)

Path Stretch (PS):

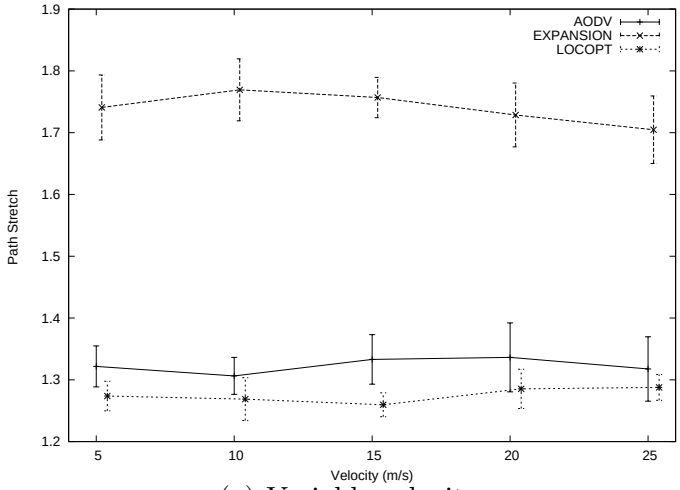
We see from the preceding analysis that the *Expansion* scheme usually beats the *LocOpt* scheme in terms of both normalized routing load (NRL) and packet delivery fraction (PDF). In this set of experiments, we compare the proposed *Expansion* and *LocOpt* schemes from the point of path optimality. Figure 3.17(a) shows that the path stretch of AODV+Expansion are higher than those provided by AODV+LocOpt. For example, when mobility setting is 5 m/s, path stretch of AODV+Expansion is about 1.75, as it is about 1.27 in case of AODV+LocOpt. Thus, it can be concluded that the *LocOpt* scheme significantly outperforms the *Expansion* scheme in terms of path optimality.

Figure 3.17(b) indicates the relationship between path optimality and network size for the proposed schemes. It is clear from the figure that the *LocOpt* scheme has better path stretch than the *Expansion* scheme does even in large network sizes.

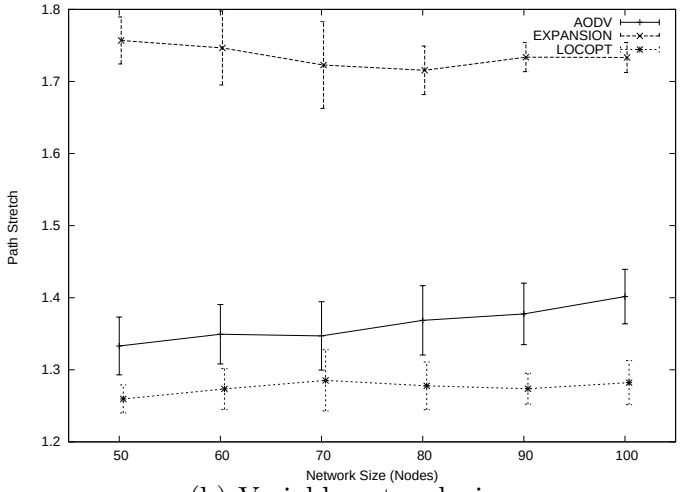
Figure 3.17(c) shows path optimality with respect to data traffic load. It is obvious from the figure that the *LocOpt* scheme outperforms the *Expansion* scheme in terms of having lower path stretch at any data load.

Number of Flooding Events (NFE):

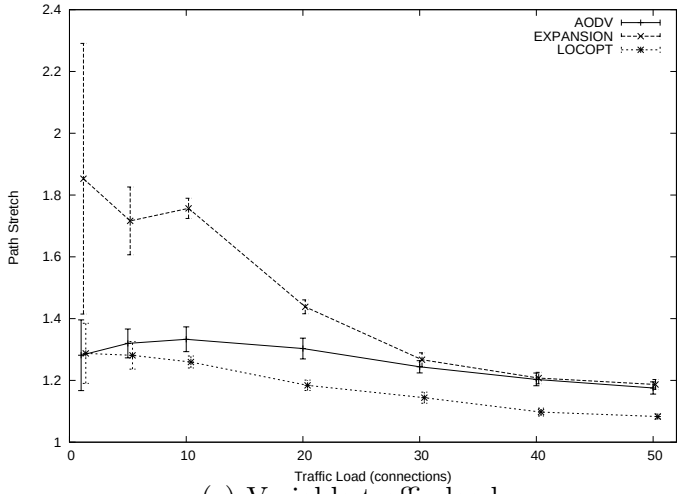
In this set of experiments, we compare the proposed *Expansion* and *LocOpt* schemes from the point of the number of flooding performed in networks. Figure 3.18(a) indicates the number of flooding for pure AODV, AODV+Expansion, and AODV+LocOpt under different mobility levels. According to the figure, when nodes move at 5 m/s, there are about 32 flooding, on average, performed in the existence of *Expansion* scheme, while this number is around 18 in case of *LocOpt*. Nevertheless, overlapping



(a) Variable velocity



(b) Variable network size



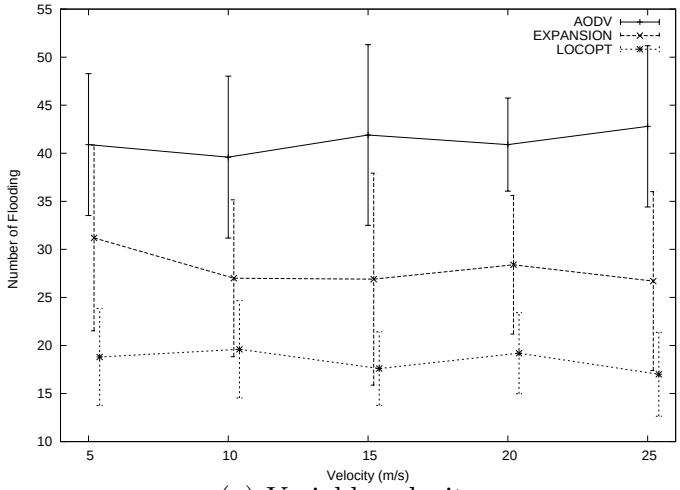
(c) Variable traffic load

Figure 3.17: Path Stretch (PS)

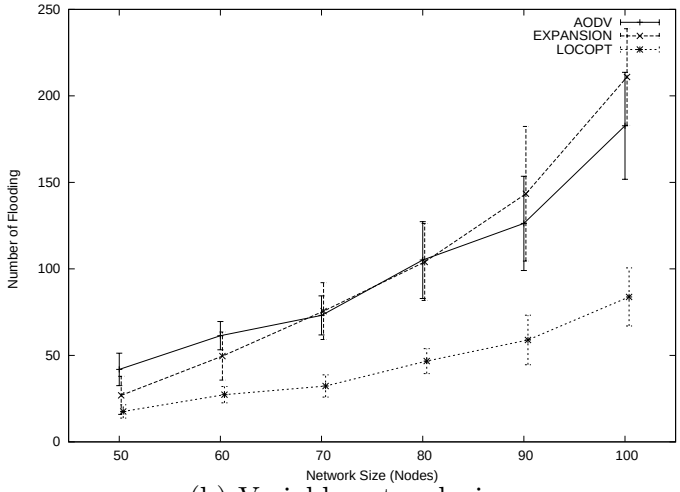
error bars prevent us from comparing these schemes based on these mean performance values of experiment trails.

In the second set of "flooding" experiments, we consider the impact of network sizes. Figure 3.18(b) indicates the number of flooding performed as network size increases. It is clear from the figure that the *LocOpt* scheme produces less number of flooding than the *Expansion* does in any network size. For example, when the LocOpt is in effect, the number of flooding, on average, are 18, 27, 32, 47, 59, and 84 for the network sizes of 50, 60, 70, 80, 90, and 100 nodes respectively. When the Expansion is in effect, the number of flooding for the given network sizes are 27, 50, 76, 104, 143, and 211 respectively, which means there is reduction in the number of flooding by 33%, 46%, 58% 55% 59% and 60% respectively.

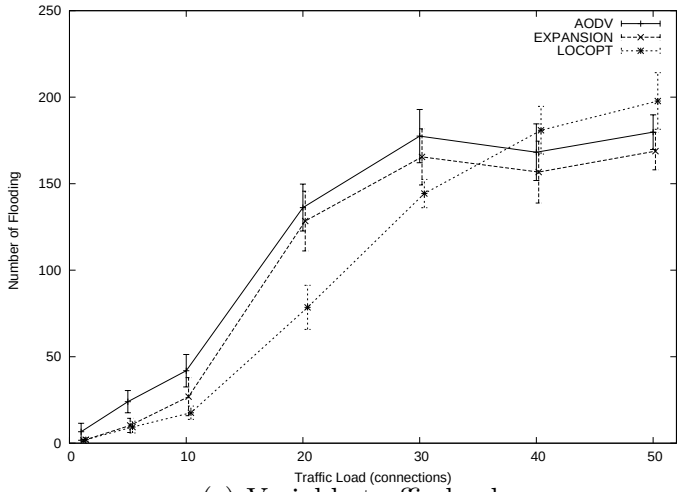
In the third set of "flooding" experiments, we investigate the effect of different data loads. Figure 3.18(c) shows the relationship between the number of flooding and data traffic load for both the proposed schemes. It is seen in the figure that the number of flooding grows, as the number of data connections increases. Since error bars in the figure generally overlap each other, we can not compare reliably the proposed schemes based on given mean values in this set of experiments.



(a) Variable velocity



(b) Variable network size



(c) Variable traffic load

Figure 3.18: Number of Flooding Events (NFE)

3.9 Formal Analysis of Scheme Invariants

In this section we seek to prove that the Expansion and LocOpt schemes are both loop-free. To make the argument formally, we will build on the notation of Sections 1.3.1 (pp. 9 and the earlier results of Section 2.10 (102).

3.9.1 Expansion Scheme is Loop-Free

Remark 4. *In the* **Expansion** *scheme, link breakage is preempted by three concurrent strategies: (i) bridging upstream nodes to the downstream endpoint of the weak link, (ii) bridging the destination to the upstream endpoint of the weak link, and (iii) splicing in a relay node to replace the week link with a path of length 2. To show that AODV continues to be loop free even after being extended with the Expansion scheme, the possibility of loop formation in these three strategies needs to be addressed.*

Definition 7. *Given a network* $G(t) = (V, E(t))$ *and destination* $d \in V$*. We say the distance field of routing tables:*

D1 **reflect unit edges***: if for all* $u, v \in V$

$$u(t).T[d].next = v \Rightarrow u(t).T[d].dist = v(t).T[d].dist + 1.$$

D2 **strictly monotonic***: if for all* $u, v \in V$

$$u(t).T[d].next = v \Rightarrow u(t).T[d].dist < v(t).T[d].dist$$

D3 **weakly monotonic***: if for all* $u, v \in V$

$$u(t).T[d].next = v \Rightarrow u(t).T[d].dist \leqslant v(t).T[d].dist$$

Clearly these relations are progressively weaker assertions:

$$(D1) \Rightarrow (D2) \Rightarrow (D3).$$

Lemma 4. *If the distance field of routing tables is initialized to weakly-monotonic values, AODV will provide loop-free routes.*

Proof. When, as part of route discovery, a RREQ message reaches an initialized node, this node responds only if its *dsn* is at least what is specified in the request. If the node responds, a new arc augments the existing flow tree to the destination, but no loop forms. If the node does not respond, then the request proceeds until it finds either: the destination, or a node with a fresh enough routing table entry for the destination. The distance field only plays a role in the setting of forwarding pointers in situations where more than one choice exists. Thus, weakly monotonic values in the distance may result in suboptimal choices for paths (from the point of view of hop count to destination), but not in loops; the loop free property is guaranteed by the *dsn*, and not by the distance field. □

Definition 8. *For any two properties* $X, Y \in \{D1, D2, D3\}$*, where* $X \Rightarrow Y$*, we say the distance field of routing tables is* **almost-always** X **/ sometimes** Y *if*

- *It is always either X or Y, and*
- *the duration of any contiguous time interval in which it is Y is finite.*

Lemma 5. *AODV+Expansion is almost-always strictly monotonic / sometimes weakly monotonic.*

Proof. AODV+Expansion fails to be strictly monotonic immediately after an expansion event occurs. However, the upstream node of the weak link initiates a distance update in this case. When the distance update process completes, the distances along the flow path are strictly monotonic. □

Lemma 6. *A loop cannot arise in the process of bridging upstream nodes to the downstream endpoint of a weak link (upon receiving a SHRINK0), in the AODV+Expansion scheme.*

Proof. Suppose the weak link is (u, v) and the flow is to node d, and

$$u \overset{t,d}{\rightarrow} v.$$

Node v broadcasts an EXP0 at time t_0, which is received by a node w t time t_1. Since AODV guarantees strict monotonicity of distance field, and by Lemma 5, AODV+Expansion is almost-always strictly monotonic / sometimes weakly monotonic. According to the expansion protocol, the EXP0 message M is acted on by node w if and only if the following strict inequality holds:

$$w(t_1).T[d].dist < M.dist$$

This guarantees that w acts on the EXP0 message if and only if either: (i) the flow from w to d passes through v (and hence w is upstream of v, or (ii) node w is on a parallel independent flow to d. In both these instances the distance inequality

constraint ensures that acting on the EXP0 and adding the forwarding link (w, v) will not cause the formation of a routing loop. □

Lemma 7. *A loop cannot arise in the process of bridging the destination to the upstream endpoint of the weak link (upon receiving a EXP2 from the destination), in the AODV+Expansion scheme.*

Proof. Suppose the weak link is (u, v), the flow is to node d, and

$$u \overset{t,d}{\rightarrow} v.$$

Node u broadcasts EXP1, and d responds with an EXP2, as per the protocol specification. Note that packets destined to d are never forwarded further upon arriving at node d. Hence, making a direct link from u directly to the destination d cannot result in the creation of a packet forwarding loop. □

Lemma 8. *A loop cannot arise in the process of distance update, in the AODV+Expansion scheme.*

Proof. By Lemma 5, AODV+Expansion scheme is almost-always strictly monotonic / sometimes weakly monotonic. Then by Lemma 4 advertisement of artificially low distances causes inefficiencies in route selection, but it cannot cause loop formation, since the distance update mechanism does not involve the manipulation of forwarding pointers. □

Lemma 9. *A loop cannot arise in the process of splicing in relay node (upon receiving a SHRINK3) to replace the weak link with a path of length 2, in the AODV+Expansion scheme.*

Proof. We assume AODV is loop-free. Suppose to contradiction that the splicing were to cause a loop. Since Expansion is pre-emptive, the splicing of relay node w into a weak link (u, v) occurs while (u, v) is still active. Then the loop must have been present already at the start of the splicing operation. This contradicts that AODV is loop-free. □

Theorem 4. *If AODV is loop-free, the AODV+Expansion is loop-free.*

Proof. Suppose Expansion causes a loop. Consider the first expansion event when this occurs. Either it occurs when we are (i) bridging upstream nodes to the downstream endpoint of the weak link, or (ii) bridging the destination to the upstream endpoint of the weak link, or (iii) splicing in a relay node to replace the week link with a path of length 2, or (iv) because of the distance update process. But it cannot occur because of (i), as was shown in Lemma 6; and it cannot happen because of (ii), as was shown in Lemma 7; and it cannot happen because of (ii), as was shown in Lemma 7; and it cannot happen because of (iii), as was shown in Lemma 9; and it cannot happen because of (iv), as was shown in Lemma 8. The contradiction proves the theorem. □

3.9.2 LocOpt Scheme is Loop-Free

Remark 5. *In the* **LocOpt** *scheme, link breakage is addressed after it has occurred rather than preemptively. The strategy is to establish a new path of length 2 to the downstream endpoint of the broken link through a bridge node that is outside of the flow tree; then use distance update to the source(s) to trigger concurrent multihop shrinking at all sources. Since monotonicity To show that AODV continues to be loop free even after being extended with the Expansion scheme, the possibility of loop*

formation in these three strategies needs to be addressed.

Lemma 10. *A loop cannot arise in the process of making a new path of length 2 to the downstream endpoint of the broken link through a bridge node that is outside of the flow tree, in the AODV+LocOpt scheme.*

Proof. Suppose the link (u, v) breaks on a flow to d, that is there exist times $t < t'$ such that

$$u(t).T[d].next = v$$

and

$$u(t').T[d].next = \text{undefined}.$$

Recall that AODV+LocOpt begins the process of addressing the link breakage by splicing in a new node w. Let us assume that this process starts at time $t_0 > t'$ completes by time $t_1 > t_0$. First, LocOpt insists that w can only be chosen provided $w(t_0).T[d].next$ is undefined (no route entry). Second, LocOpt does not apply the forwarding pointer updates if the upstream node's dsn value changes during the local recovery process. This ensures

$$u(t_0).T[d].dsn = u(t_1).T[d].dsn$$

In order for a loop to form, some node b

$$u \overset{t_0,d}{\rightarrow} b$$

must become upstream of u in the period of time $[t', t_1]$ between when the link breaks and when LocOpt's recovery process completes. For this to position reversal to hap-

pen, b must have either initiated or received a *global* discovery for d and forwarded this to a node a upstream of u; local recoveries could not cause a position reversal because standard AODV local recovery operation is entirely replaced with LocOpt's custom operation sequence (which use a special request message). If at time $j_0 > t'$ node b forwarded the RREQ (and did not respond to it, despite having a valid route to d) then the dsn in the RREQ message M must have been strictly greater than the dsn in b's routing table entry for destination d:

$$b(j_0).T[d].dsn < M.dsn$$

If node a replied to the RREQ upon receiving message M at time $j_1 \in (j_0, t_1)$ then the node's routing table must have an entry for d with a dsn at least as large as the one specified in M:

$$M.dsn \leqslant a(j_1).T[d].dsn$$

Since b was downstream from u,

$$u(t_0).T[d].dsn \leqslant b(t_0).T[d].dsn$$

By Lemma 1,

$$b(t_0).T[d].dsn \leqslant b(j_0).T[d].dsn$$

and

$$a(j_1).T[d].dsn \leqslant a(t_1).T[d].dsn$$

The above five inequalities concerning dsn combine to yield

$$u(t_0).T[d].dsn < a(t_1).T[d].dsn$$

Since $u(t_0).T[d].dsn = u(t_1).T[d].dsn$, this implies

$$u(t_1).T[d].dsn < a(t_1).T[d].dsn$$

But then by Lemma 2, node u cannot be downstream of a at time t_1 when the splicing operation completes. It follows that a loop cannot arise in the process of making a new path of length 2 to the downstream endpoint of the broken link through a bridge node that is outside of the flow tree. □

Lemma 11. *A loop cannot arise in the process of distance update, in the AODV+LocOpt scheme.*

Proof. The proof is completely analogous to proof of 8. □

Lemma 12. *A loop cannot arise in the process of multihop shrinking, in the AODV+LocOpt scheme.*

Proof. Follows immediately from Theorem 3. □

Theorem 5. *If AODV is loop-free, the AODV+LocOpt is loop-free.*

Proof. AODV+LocOpt addresses the breakage of a link by establishing a new path of length 2 from the upstream to the downstream endpoint of the broken link (through a bridge node that is outside of the flow tree); by Lemma 10, this cannot cause a loop. It then uses the distance update to the source(s) to trigger concurrent multihop shrinking at all sources; by Lemmas 11 and 12, this cannot cause a loop to form. This theorem is proved. □

CHAPTER 4

DYNAMIC POWER DISTRIBUTION

4.1 Statement of the Problem

Historically, reconciling the gap between power availability and power supply in MANETs has involved addressing the following issues: (i) improving the power *efficiency* in the system; and (ii) preventing the system deconstruction due to un*fair* power usage [60]. To address these concerns, a budgeted power model was proposed in an earlier study [10, 11], normalizing the measurement of "efficiency" and "fairness" by using a model in which every connection is assigned a fixed power utilization budget. In practice, the magnitude of the budget reflects the connection's priority, or equivalently, the benefit that the system derives by maintaining the connection. In consumer MANETs, for example, this benefit might be based on financial incentives provided by a satisfied customer who will continue to pay for the connection service. In military MANETs the benefit might reflect the extent to which the connection is essential to achieving a positive outcome in some coordinated systemic mission objective.

In this chapter of the thesis, we are interested in the efficient distribution of a budgeted power among constituents nodes of a connection(i.e. route) in MANET environment, with the objective of increasing connection lifetime in the face of mobility. A con-

nection's lifetime is typically taken to be the time interval during which all of the connection's constituent links are operational. A link in a connection ceases to be operational when the autonomous movement of one of the link's endpoints causes it to fall out of transmission range of the other endpoint as illustrated in Figure 4.1. To increase lifetime of a connection, such an anticipated link breakage should be prevented from happening, or at least postponed to a later time as much as possible.

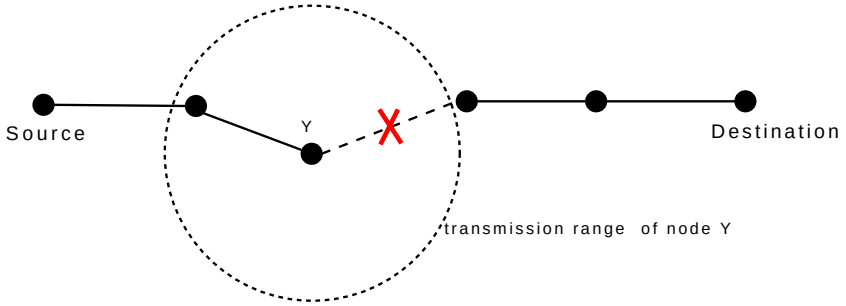


Figure 4.1: An example of link breakage on a connection

4.2 Motivation

Expected benefits from increasing lifetime of a connection are listed below:

- *Preventing service interruptions.* There may be some applications which are sensitive to service interruptions on the connections. Any increment in connection lifetime is favorable for this type of applications.

- *Decrease in routing overhead.* In order to establish a connection between a pair of source and destination nodes, it is generally required to initiate a network-wide route discovery process, which a very expensive operation in terms of control traffic incurred. As the lifetime of connections in a network increases, the expected number of route discovery operation decreases.

- *Decrease in packet losses.* There may be several reasons for losing packets in a network, however, from the point of our interest, losing packets because of the disconnection on the routes can be partly eliminated by increasing the connection lifetime.

- *Decrease in average delay:* The average delay in packet delivery could be high in case of link breakages since it takes time to construct a new route instead of the broken one. Therefore, increased connected lifetime can decrease the expected average delay in packet delivery.

4.3 Related Work

4.3.1 Budgeted Power

The budgeted power model was initially introduced by Khan in 2006 [10]. Later, in [27], the authors considered the opportunities afforded by such a model vis-a-vis minimizing connection bit error rate (BER), and presented a distributed scheme which successfully minimize a the BER of a connection by continuously reapportioning its fixed power budget among the constituent (static) nodes of the connection.

As an objective, lifetime maximization has been interpreted in one of two ways: network lifetime maximization, and connection lifetime maximization.

4.3.2 Network Lifetime

The lifetime of a network is most frequently defined as the time interval for which the network is a connected graph. Broadly speaking, the network may partition (becoming disconnected) when one of two events occurs: (i) the autonomous movement of a node causes some of its incident link(s) to fail due to a shortage of transmission power, or (ii) some node exhausts its energy supply sufficiently so that some of its incident links fail. Most prior research on network lifetime attempts to delay the onset of these two types of events—the most frequent emphasis being on event type (ii), see e.g. works of Abbas et al. in 2006 [6], and Kim and Jang in 2006 [28]–by extending the network routing protocol to make it energy-aware and using a route selection strategy that facilitates optimization with respect to the network's lifetime.

4.3.3 Connection Lifetime

Somewhat analogously, a connection's lifetime is typically taken to be the time interval during which all of the connection's constituent links are operational. A link in a connection ceases to be operational when one of two events occurs: (a) the autonomous movement of one of the link's endpoints causes it to fall out of transmission range of the other endpoint, or (b) one of the two endpoints exhausts its energy supply, causing the other endpoint fall out of transmission range. Most prior research on connection lifetime attempts to delay the onset of these two types of events—the most frequent emphasis being on events of type (a), see e.g. Dube et al.'s 1997 paper [20], Toh's 1997 work [52], Lim's 2002 study [34], Das et al.'s 2000 paper [18], and Lenders et al.'s 2006 article [20, 52, 34, 18, 31]. The main approach has been (as

was the case in research on network lifetime) to extend the network routing protocol by making it energy-aware, and then to make route selection sensitive to connection lifetime maximization. In [20, 52, 34, 18], for example, the authors proposed new routing protocol extensions based on finding the path which probably has longest lifetime among many possible paths from source to destination. In these studies, the route selection process was based on the received signal strength at each of the nodes [20, 34], transmission range and the relative speed of the nodes [52], or recent changes in signal strength [31].

4.3.4 Topology Control

Another common application area of transmission power control is topology control of wireless networks. The topology of a wireless ad hoc network is defined as the combination of communication links between node pairs used explicitly or implicitly by a routing protocol [44]. Then, we can define topology control (TC) as a mechanism to obtain the desired topology according to intended objective(s) by changing the transmit power level of the nodes.

One of the earliest and most referenced work in this area is that of Rudolph and Meng [46], in 1999, the authors of which described a distributed position-based TC algorithm with the objective of minimizing total power consumption in mobile wireless networks. In this work, basically, the authors introduced the notion of relay region and enclosure that can be defined as follows: For any node i that intends to transmit to node j, node j is declared to lie in the relay region of a third node r, if node i will consume less power when it sends packets through relay node r instead of transmitting directly to node j. The enclosure of node i is then defined as the

union of the complement of relay regions of the nodes which are i's neighbor but not lying in the relay regions of previously found nodes. It is shown that the network is strongly connected if every node maintains links with the nodes in its enclosure and the resulting topology is a minimum power topology, which includes the minimum-power paths from each node to a selected master node. The efficiency of the proposed algorithm greatly depends on the accuracy of the assumed propagation loss model since determination of the relay region of each node is performed analytically based on location information provided by GPS and the assumed propagation loss model. To handle limited mobility, it is suggested that each node periodically repeats the protocol according to updated location information. In contrast to many contemporary studies, astonishingly, this relatively older work didn't ignore the power consumed at the receiver side of a transmission, which is a subject that will be discussed briefly towards the end of the paper.

Another famous study, having a similar objective, was presented by Ramanathan and Rosales-Hain [44] in 2000 . In this work, the authors proposed two centralized algorithms to minimize the maximum power used per node while keeping the (bi)connectivity of the network. The first algorithm, CONNECT, is a simple greedy algorithm, similar to minimum spanning tree algorithm, that iteratively merges different components until only one remains. Initially, each node is a component by itself. Then node pairs are selected in non-decreasing order of the distance between them. If the selected nodes are in different components, then the transmit power of each node is adjusted to the minimum level that satisfies connectivity between them. This process continues until the whole network becomes connected under the assumption that network connectivity can be achieved without exceeding the maximum allowable transmission powers. Augmenting a connected network to a bi-connected network is

performed by BICONN-AUGMENT algorithm, which uses the same idea as in CONNECT to iteratively build the bi-connected network. Finally, a post-processing phase can be applied to ensure per-node minimality by deleting redundant connections [44].

Another noteworthy study in this area, dealing with energy efficiency and network connectivity, is Cone Based Topology Control algorithm (CBTC) proposed by Wattenhofer et al. [54] in 2001. It is assumed in this study that nodes have ability to determine direction of incoming signals. The algorithm works in two phases. In the first phase, the nodes incrementally increase their transmission power until they find a neighbor within a certain degree (α) of slice centered on themselves. In the second phase, the nodes eliminate costly links, which are the links that can be reached through closer neighbors. This algorithm guarantees connectivity if α is smaller than 2pi/3, and if the original network is connected, which is the graph obtained by maximum transmitting power. Also, the algorithm bounds maximum node degree.

Aiming to maintain network connectivity, Li et al. [33] propose a topology control algorithm, in 2003, called Local Minimum Spanning Tree (LMST) that have the following three nice properties: (i) network connectivity is preserved in the derived topology under the LMST, (ii) the maximum node degree is bounded by 6 in the resulting topology, (iii) the resulting network graph can be made symmetric by removing unidirectional links without impairing network connectivity. The authors prove these properties analytically and support their claim by simulation results. The property (i) says that if the original graph, which is obtained at maximum transmit power of the nodes, is connected initially, then LMST does not impair connectivity constraint besides providing other nice properties. However, notice that if the original graph is not connected in the beginning, then there is nothing to do by LMST in order to

make the network connected. Regarding second property, the maximum node degree is important in wireless networks because it is a measure of contention level, as we will discuss in the next section. The third property (i.e. bidirectionality) may become an important issue in some cases. For instance, AODV benefits from bidirectional links while performing its routing task. The proposed LMST algorithm in [33] works as follows: Each node initially advertises/broadcasts its location information provided by GPS or another technology to its neighbors within its transmission range. Then each node calculates its local MST of neighborhood, rooted to itself, by weighting links as the Euclidian link distance. Finally, nodes adjust their transmission power as just enough to cover only 1-hop neighbors on their local MST tree. Notice that this may not result in global MST, but it is shown in another study [reference] that its properties are good enough to that of global MST's properties, and it is a sub-graph of the global MST. LMST is developed for the static or low mobile networks.

Another TC algorithm, XTC, guarantying the network connectivity as long as the maxpower communication graph is connected, is presented by Wattenhofer and Zollinger [55] in 2003. XTC constructs the topology by taking into account the link quality among the neighbor nodes. The algorithm basically works as follows: Firstly, nodes order their neighbors according to link quality which can be estimated from received signal strength, and keep this neighbor-order information. Then the calculated neighbor-order information is exchanged among all neighbor nodes. Thus, every node also obtains the neighbor-order information of its neighbors. By comparing the collected neighbor-order information, a node u forms a direct link to its neighbor v only if there is no better way to reach v through other nodes. It is shown that the resulting topology keeps connectivity, provides only bidirectional links, and bounds the node degree by 6.

Santi [48] provides a good survey, in 2005,regarding topology control algorithms, as investigating and classifying them from different perspectives.

This work diverges and extends the earlier investigations of the authors [27] in two very significant ways: First, this paper considers *mobile* nodes instead of merely a static snapshots of a dynamic network. Second, our objective is to leverage the ability to dynamically distribute a connection's power budget to *maximize expected connection lifetime*, rather than to minimize connection BER as was the focus in our earlier work [27]. We will compare our proposed lifetime-maximizing schemes with the connection lifetimes exhibited under the BER-minimizing power distribution scheme developed in [27]. By doing so we shall quantify the extent to which the two objectives (lifetime maximization and BER minimization) are in opposition. Finally, we will refine our simulation-based analysis to consider *the traffic required in operating the control protocols themselves.*

4.4 Objectives

The prior research described in the preceding section addresses those situations in which there is a very limited amount of energy available at each node, as is the case for example, in sensor networks or networks consisting of small mobile devices. Additionally, the strategies adopted in the prior research have centered on the routing layer, vis-a-vis energy-aware path selection. In contrast, we consider a model in which each connection's power requirements are modest compared to battery capacities, and where there are many connections which have been prioritized relative to each other by assigning each connection its own fixed *power budget* (i.e. a cumulative energy

utilization rate). Our model better reflects the realities of battlefield settings in which the MANET nodes are unmanned autonomous vehicles [32, 13].

We presume that energy supplies at the nodes are renewable. Specifically, we assume that each connection in a network has a certain ***power consumption budget*** which can be distributed among the nodes on the connection [10, 11]. We describe new power budget distribution schemes which seek to increase connection robustness in mobile environments.

This work thus begins at the point where the research efforts on energy-aware routing end. We assume throughout that the problem of route selection has been resolved in some manner, e.g. by one of the schemes cited in the previous section. Because all schemes we compare implicitly make use of the (unspecified) underlying routing protocol *in the same manner*, the effects of any inefficiencies in routing are reflected identically across the schemes. We make the idealized assumption that node mobility is insignificant when compared to routing convergence times. In reality, of course, this is not always true, but our conclusions regarding the *relative* lifetime enhancing merits of the various power distribution schemes are not influenced by the assumption. Because our study takes routing for granted, we do not investigate on the possibility of extending the lifetime of a connection post link failure through routing-level connection recovery processes. Rather, we restrict ourselves to maximizing expected connection lifetime by minimizing the probability of link breakage.

We describe dynamic schemes that continuously redistributes the power budget assigned to a connection among its constituent nodes, with the objective of postponing events of type (a), thereby maximizing the connection's expected lifetime in the face of node mobility. We assume—as other similar investigations have, that each node is

able to send with dynamically tunable transmission power [51, 11, 10, 27]. Each connection's power requirements are assumed to be modest relative to energy availability at the nodes, and connections are prioritized relative to one another by means of their power budgets. The proposed dynamic power distribution protocol is implemented on top of a routing protocol that is responsible for providing a multi-hop path between s and t, within total power budget constraints.

To define the problem more specifically, consider a single connection between a source node s and a destination node t, and assume that a transmission power budget P has been specified for this connection. The **Fundamental Questions** to be answered are:

Q1. Can one design a lifetime-maximizing power distribution scheme(s) which will dynamically distribute the connection's power budget among the constituent nodes?

Q2. How do these lifetime-maximizing power distribution scheme(s) perform relative to a scheme which simply allocates power in a uniform static manner among the constituent nodes?

Q3. How do these lifetime-maximizing power distribution scheme(s) perform relative to a scheme which dynamically distributes power with the objective of minimizing end to end connection bit error rate? Are BER minimization and lifetime maximization two competing objectives?

Q4. What is the control traffic overhead of these lifetime-maximizing power distribution scheme(s), or more precisely, how do their performance depend on the parameters which govern their control traffic overhead? Answering this ques-

tion requires us to again compare the relative increases achieved in expected connection lifetimes (Q1,Q2,Q3), but under the additional requirement that all schemes use comparable resources for control traffic.

4.5 Network Model

We consider a wireless ad hoc network consisting of N nodes equipped with omni-directional antennas that can dynamically adjust their transmission power. We model this network as a geometric graph $G = (V, E)$, where V is the set of nodes and E is the set of edges. Each node is assigned a unique ID i in $\{1, \ldots, |V|\}$, and node i can send data with a dynamically tunable transmission power.

Wireless propagation suffers severe attenuation [19]. If node i transmits with power $P_t(i)$, the power of the signal received by node j is given by $P_{rcv}(j) = \frac{P_t(i)}{c \times d_{ij}^{\alpha}}$, where d_{ij} is the distance between nodes i and j, and α, c are both constants, and usually $2 \leq \alpha \leq 4$ (See [19]). In order to correctly decode the signal at the receiver side, it is required that $P_{rcv}(j) \geqslant \beta_0 \times N_0$, where β_0 is the required signal to noise ratio (SNR) and N_0 is the strength of the ambient noise. We denote the minimum signal power at which node i is able to decode the received signal as $P_{min} = \beta_0 \times N_0$.

4.6 Power Distribution Schemes

The following sequence of observations are the intuitive foundation for the first power distribution scheme we propose, *Sqr*:

1. When a multi-hop connection fails, it does so because at least one of its constituent links has failed.

2. Consider the point in time T when the first link failure occurs. Suppose that one link L_1 of the connection has failed at T while another link L_2 still survives. Then the power budget must have been distributed suboptimally, since giving L_2's endpoints (infinitesimally) less power, and L_1's endpoints (infinitesimally) more power would have yielded a longer lifetime for the connection.

3. Thus, for connection lifetime to be maximized, power must be distributed in such a manner that at the point in time when the connection fails, *all* of its constituent links fail simultaneously.

4. If all nodes have the same sensitivity threshold P_{min} then item (3) implies that the power budget must be distributed in such a manner that the received signal power at each node in the connection is the same.

5. Suppose the connection has power budget P and the distance between nodes j and $j+1$ of the connection is d_j (for $j = 1, \ldots, N-1$). If node j transmits with power $P \cdot d_j^2 / \sum_{i=1}^{N-1} d_i^2$, then all nodes will receive the transmission from their upstream neighbor at the same power level, thus satisfying the conclusion of item (4). Additionally, the total power consumption attributable to the connection will be precisely P.

In what follows, we will refer to the dynamic power redistribution scheme deduced above as the *Sqr* scheme:

4.6.1 Sqr Scheme

Under this power distribution scheme, the power is allocated based on the square of the distance to the next hop along the path towards the destination node. Specifically, given a connection between nodes s and t with length $N-1$ hops and a total power budget P, each node j will be allocated

$$P_{sqr}(j) = P \cdot d_j^2 / \sum_{i=1}^{N-1} d_i^2,$$

where d_j is the distance from node j to node $j+1$ along the path.

The protocol runs continuously to keep power values updated in light of node mobility. The protocol strives to keep received signal power at each node identical, thereby ensuring all links have the same stability. The implementation is as follows:

In phase 1, the source node initiates a control message including its Tx power information and sends it to the next hop towards the destination node. At the next hop the receiver deduces the distance to the sender by comparing Tx and Rx levels, then inserts its own Tx level into the message, updates the cumulative square-distance field, and forwards it to the next hop. This process repeats; when the message is delivered to the destination, $\sum d_i^2$ is known. In phase 2, the destination initiates another control message containing $\sum d_i^2$ and sends it to the source node, allowing all transit nodes to learn the value. In order to prevent budget violations, only reductions of power are carried out in phase 2; increases are carried out in a third phase, in reaction to a control message from the source to the destination.

4.6.2 Uniform Scheme

Given an N-node connection between nodes s and t having total power budget P, the *Uniform* power distribution scheme allocates power uniformly to each of the $N-1$ nodes (excluding the destination node) $P_{unif}(j) = \frac{P}{N-1}$.

4.6.3 MinBER Scheme

This power budget distribution protocol was originally described by the authors in [27]. The protocol operates on *all* (overlapping) consecutive triplets of nodes within the connection (s,t). Within each triplet, we denote the nodes as the upstream node, the central node, and the downstream node—this naming convention is illustrated in Figure 4.2.

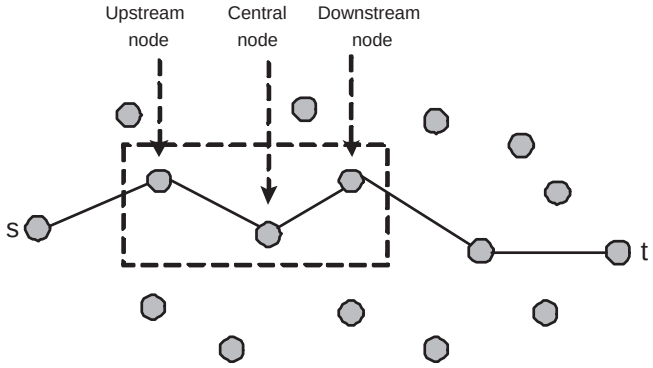


Figure 4.2: Multi-hop path description

A node enters the protocol by simultaneously sending an Update message to its upstream and downstream neighbors. The Update message describes its present transmission strength. A node receiving an update uses its contents and the actual received signal strength to deduce an estimate of the distance to the sender of the Update. Thus each node (viewed in its central role) maintains estimates of distance

to upstream and downstream nodes. When the central node receives an Update message informing it of the transmission power and (implicitly) distance to a neighbor, it determines the optimal redistribution of power between itself and the upstream node. This local optimization is computed using the analytic model of BER model presented in [27]. In effect the central node acts greedily to minimize the BER of the two hop sub-path from its upstream neighbor to its downstream neighbor. If the local optimization shows that a significant redistribution of power is required, and this redistribution will not cause the received signal strength to drop below P_{min} at any node, then the central node is able to draw power downstream or push power upstream. The power reallocation process is negotiated concurrently between all (overlapping) triplets of nodes via a distributed protocol. The protocol is said to "converge" when the total power exchanged drops below a specified threshold. Since we are interested in comparing the quality of power distribution decisions of various schemes, we assume that the *minBER* algorithm converges in timeframes significantly shorter than those involved in node mobility and routing; we denote the resulting power distribution at node j as $P_{BER}(j)$.

4.7 Experimental Setup

In this part of the experiments, we assume that the control traffic incurred by the schemes is free, and also convergence time of the schemes is immediate. The effects of relaxation of this assumptions are evaluated in another set of experiments in the following sections.

Initial network design. In our simulations, we consider connections where the nodes were placed randomly according to the following inductive process: If node j is located at (x_j, y_j), then node $j+1$ is located at $(x_{j+1}, y_{j+1}) = (x_j + d_x, y_j + d_y)$ where d_x, d_y are uniformly distributed in the interval $[0, D]$. For most experiments (except when considering the effect of node density) we took $D = 100$ meters.

Mobility model. Nodes are allowed to move according to a Cartesian random walk mobility model [14]. Each node has five possible directions in which to move at each time step, of which one is selected uniformly at random: it may go north, south, east, or west with velocity v, or to stay at the current position until next time step. For instance, movement frequency = 100 means that the nodes move after every 100 packets handled at a node.

Performance measures. Starting with an initial network, we generate a movement sequence for the nodes. Then we simulate this movement sequence under each of the three power distribution schemes and note the time at which the connection fails (i.e. one of the constituent links fails) for each of the schemes. We denote these times as T_{Unif}, T_{Sqr}, T_{BER} and compute the advantage or "gain" enjoyed by the *Sqr* scheme over the *Unif* scheme as

$$\mathrm{G}_{Unif} = T_{Sqr}/T_{Unif}$$

, and the gain over the *minBER* scheme as

$$\mathrm{G}_{BER} = T_{Sqr}/T_{BER}$$

. The preceding experiment is carried out repeatedly, in 10^4 independent trials, where each trial begins with a different random initial network and movement sequence. The

aggregate performance metrics are computed as averages over 10^4 trials:

$$\text{Expected Gain over Uniform} = E[T_{Sqr}/T_{Unif}]$$

$$\text{Expected Gain over minBER} = E[T_{Sqr}/T_{BER}].$$

System and environmental parameters. We explored the impact of the following situational parameters on the above two performance metrics:

- *Number of nodes N:* We vary the number of nodes on the path ranging from few (5) to many (25) nodes.
- *Power budget P:* We consider connection power budgets ranging from small (2.0 Watts) to large (10.0 Watts).
- *Initial node density δ:* We vary node density in the initial network from sparse (0.15 nodes/m) to dense (0.4 nodes/m). This is achieved by taking $N = 500 * \delta$ and scaling the network geometry proportionally so that the initial network is bounded by the 500 meter square.
- *Mean node velocity v:* We consider velocity of the nodes ranging from slow (0.5 meters/sec; 1.1 miles/hr) to rapid (4.0 meters/sec; 9 miles/hr).
- *Signal attenuation exponent* α is taken as 2, appropriate to our connection distance scales in free space.
- P_{min}, the minimum signal power at which a receiver is able to decode a signal is taken as 10 mW.

4.8 Simulation Results and Analysis

We conducted experiments to quantify the influence of the number of nodes N, connection power budget P, initial node density δ, and mean node velocity v, on the expected gain of *Sqr* over the *Uniform* and *minBER* schemes. The error bars on each graph below show the width of (plus/minus) one standard deviation from the expected values of the gains.

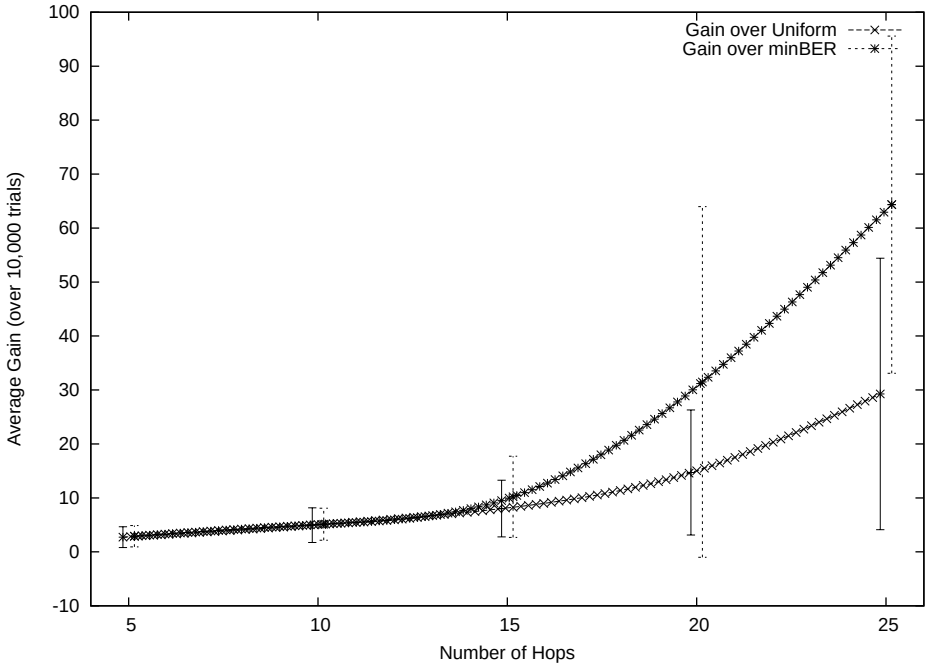


Figure 4.3: The Influence of Number of Hops on Gain

Varying the connection size. In the first set of experiments, we varied the connection size from 5 to 25 nodes, while keeping all other variables fixed: the power budget (P) was fixed at 5.0W, the mean velocity of the nodes (v) was fixed at 1.0 m/sec, and the initial mean node density was one node every 50m. As can be seen in Figure 4.3, *Sqr* enjoys a linearly growing gain over both *Uniform* and *minBER* for small connections sizes (i.e. when $N < 15$), outperforming them both by a factor of

2.7 when $N = 5$, a factor of 4.9 when $N = 10$ and a factor of 8.5 when $N = 15$. As connection size grows beyond this threshold, the minBER scheme's power allocation decisions deviate significantly from the lifetime maximization scheme, and the rate at which the latter's gains increase begins to grow super-linearly. The reason for this divergence is explainable as follows. Since we are holding the connection's power budget constant while increasing N, when N becomes sufficiently large energy becomes scarce, and the power distribution which minimizes BER is markedly different from the power distribution which maximizes connection lifetime.

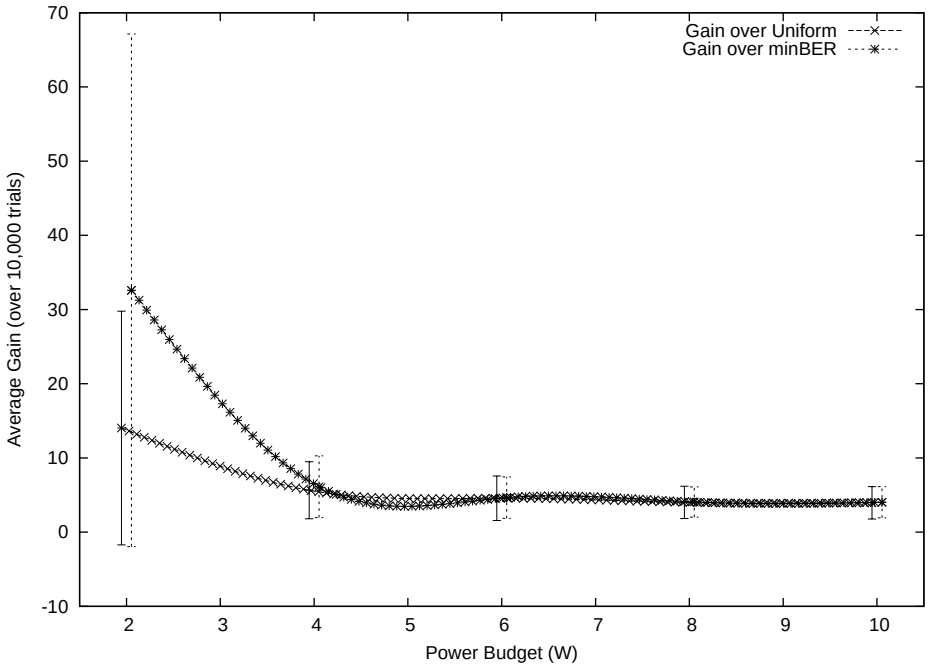


Figure 4.4: The Influence of Power Budget on Gain

Varying the power budget. In the second set of experiments, we varied the connection's power budget from 2.0W to 10.0W, while keeping all other variables fixed: the number of nodes (N) was fixed at 10, the mean velocity of the nodes (v) was fixed at 1.0 m/sec, and the initial mean node density was one node every 50m. As can be seen in Figure 4.4, the *Sqr* scheme enjoys a very significant (factor of

32) gain over *minBER* and factor of 14 gain over the *Uniform* scheme in low power budget settings (2.0W). As power budgets increase, the gain that *Sqr* enjoys over both schemes declines near-linearly. Once the power budget exceeds a threshold of 4.0W, any further increases of the power budget do not appear to separate *Sqr*'s factor of 3.9 advantage over *Uniform* and *minBER*.

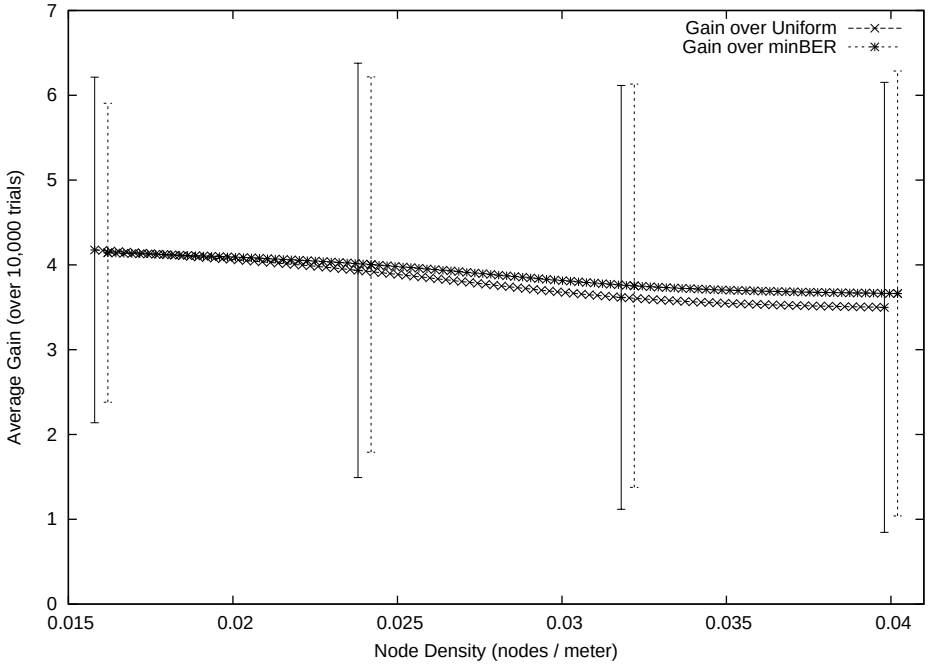


Figure 4.5: The Influence of Node Density on Gain

Varying the node density. In the third set of experiments, we varied the node density from one node every 66m ($\delta = 0.15$ nodes/meter) to one node every 25m ($\delta = 0.40$ nodes/meter), while keeping all other variables fixed: the power budget (P) was fixed at 5.0W, the mean velocity of the nodes (v) was fixed at 1.0 m/sec. The number of nodes N was taken to be $500/\delta$, and the initial configuration was rescaled proportionately so that it was bounded by a 500m by 500m square. As can be seen in Figure 4.5, *Sqr* enjoys a significant gain in connection lifetime (a factor of 4.1) over both *Uniform* and *minBER*, although as node density increases, the gain

is gradually reduced, becoming only a factor of 3.5 when the node density reaches $\delta = 0.4$ nodes per meter. Altering node density does not appear to separate *Sqr*'s advantage over *Uniform* from *Sqr*'s advantage over *minBER*.

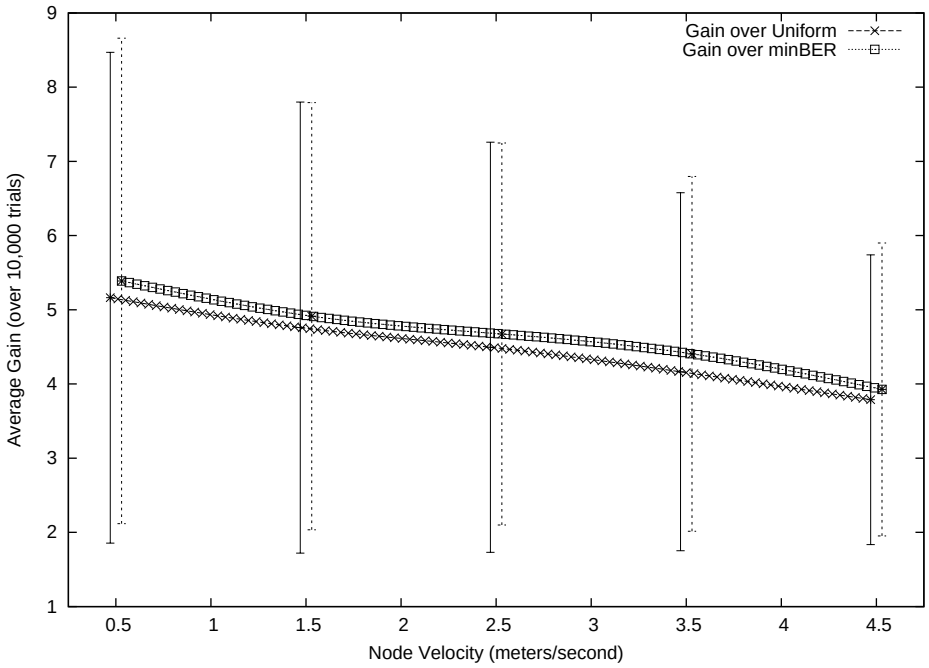


Figure 4.6: The Influence of Node Velocity on Gain

Varying the node velocity. In the final set of experiments, we varied the mean node velocity from 0.5 meters/sec to 4.5 meters/sec, while keeping all other variables fixed: the power budget (P) was fixed at 5.0W, and the initial mean node density was one node every 50m. As can be seen in Figure 4.6, *Sqr* enjoys a significant factor of 5.2 gain in connection lifetime over both *Uniform* and *minBER*, although as node velocity increases, the gain is gradually reduced, becoming only a factor of 3.8 when the node velocity reaches 4.5 meters/sec. Altering node velocity does not appear to separate *Sqr*'s advantage over *Uniform* from *Sqr*'s advantage over *minBER*.

4.9 More Power Distribution Schemes

The experimental setup and analysis of previous simulations were based on two assumptions that (i) control traffic incurred by the schemes is free, and (ii) the convergence time of the schemes is ignorable. Under this assumptions, all links should break simultaneously for the *Sqr* scheme, because Sqr scheme equalizes the received signal strength at each node. However, when these assumption are relaxed, the performance of the proposed power distribution scheme, i.e., *Sqr*, becomes no longer so good over the other schemes. This is because links under Sqr scheme have different level of robustness in the face of node movements. More specifically, shorter links become much more fragile against mobility than longer links are, since shorter links have a smaller radius of safe region, in which corresponding downstream nodes can move without causing link break. We will describe this issue on a concrete example in the following part.

4.9.1 More Investigation on Sqr scheme

In this part, we will show the shortcoming of Sqr scheme on a concrete example. To do this, we suppose a 2-hop connection between source s and destination t through intermediate hop u, with a power budget P_{budget} assigned for this connection. Figure 4.7 illustrates such a simple connection model. Lets further suppose that Source (i.e. node s) and Destination (i.e. node t) are not moving, and the distance between them is D. To reveal the impact of different distance settings between the nodes, let's suppose node u moves on a straight line connecting s and t, resulting in variable distance between Source s and node u, which is denoted as d_{su}.

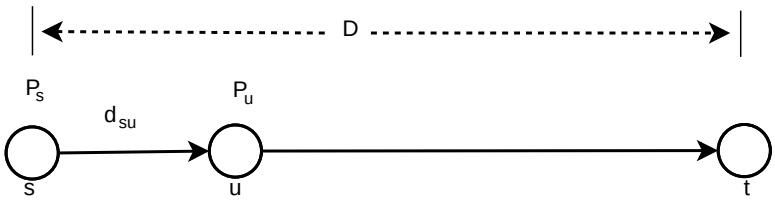


Figure 4.7: Power distribution

Power distribution in Sqr scheme:

From the context of the given connection model above, we will calculate transmission power of node s and node u for Sqr scheme. Sqr scheme distributes the power budget among transmitting nodes proportionally with square of distances between them. Formally, the transmission power of node s and node u are calculated based on Equations 4.1 and 4.3 respectively.

$$P_{Sqr}(s) = P_{budget} \cdot \frac{d_{su}^2}{d_{su}^2 + (D - d_{su})^2} \tag{4.1}$$

$$P_{Sqr}(u) = P_{budget} - P_{Sqr}(s) \tag{4.2}$$

$$= P_{budget} \cdot \frac{(D - d_{su})^2}{d_{su}^2 + (D - d_{su})^2} \tag{4.3}$$

The graphical representation of these equations is shown in Figure 4.8, setting $P_{budget} = 10W$ and $D = 10m$.

Now, we introduce a new metric called "Safe distance".

Definition 9. *(Safe Distance)*

Safe distance (SD) is the minimum range that a node must move in order to cause a disconnection from its upstream hop. In other words, SD of a node equals the

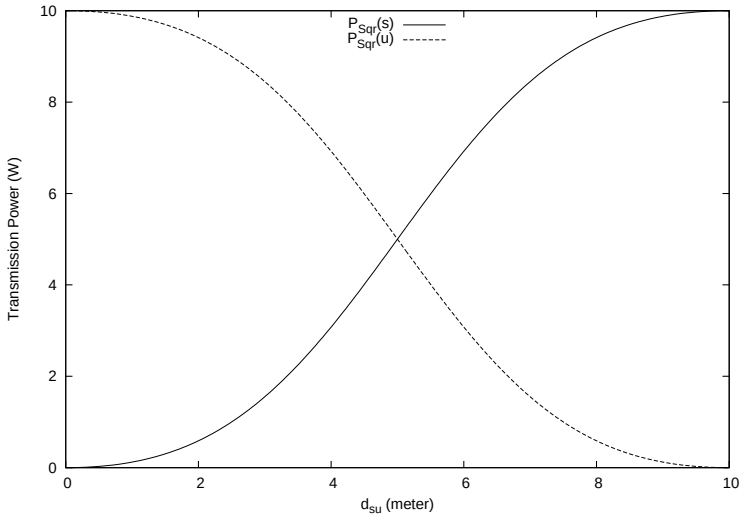


Figure 4.8: Power distribution for Sqr scheme

difference between the transmission radius of its upstream node and the distance to its upstream node. Formally, Safe Distance of a node i, SD(i), whose upstream node is j, is given by Equation 4.4.

$$SD(i) = \sqrt[\alpha]{\frac{P_j}{P_{min}}} - d_{ij} \tag{4.4}$$

where P_j transmission power of node j, P_{min} is the minimum received signal power required in order to correctly decode the received signal, d_{ij} is the distance between nodes i and j, and α is attenuation constant of the transmission medium, and usually $2 \leq \alpha \leq 4$ (See [19]).

According to Definition 9, SDs of node u and node t for Sqr scheme in our connection model can be calculated as in Equations 4.6 and 4.8 respectively.

$$SD_{Sqr}(u) = \sqrt[\alpha]{\frac{P_{Sqr}(s)}{P_{min}}} - d_{su} \tag{4.5}$$

$$= \sqrt[\alpha]{\frac{P_{budget}}{P_{min}} \cdot \frac{d_{su}^2}{d_{su}^2 + (D - d_{su})^2}} - d_{su} \tag{4.6}$$

$$SD_{Sqr}(t) = \sqrt[\alpha]{\frac{P_{Sqr}(u)}{P_{min}}} - (D - d_{su}) \tag{4.7}$$

$$= \sqrt[\alpha]{\frac{P_{budget}}{P_{min}} \cdot \frac{(D - d_{su})^2}{d_{su}^2 + (D - d_{su})^2}} - (D - d_{su}) \tag{4.8}$$

Figure 4.9 shows SDs of node u and node t, based on Equations 4.6 and 4.8. According to the figure, when two nodes are too close to each other (e.g. $d_{su} \approx 0$ or $d_{su} \approx D$, where $D = 10$), SDs of the nodes become significantly different, resulting in unbalanced SDs. This is the main shortcoming of the Sqr scheme, and this effect becomes much more severe when the SD of one node becomes very low, which causes immediate disconnections due to mobility.

Power distribution in Safe scheme:

To fix aforementioned drawback of Sqr scheme, we propose another power distribution scheme, called Safe, which distributes the power budget among transmitting nodes so that the SDs of all nodes become equal. Nevertheless, it is difficult to construct a mathematical formula to calculate transmission power of nodes in this scheme. Instead, required power levels are calculated by numeric calculation methods at the Destination node once all distance information is obtained. Details of the scheme is

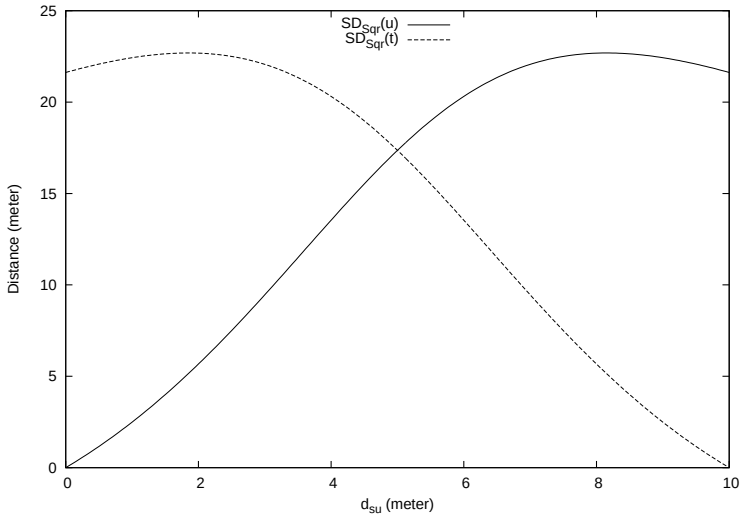


Figure 4.9: Safe Distances of node u and node t for Sqr scheme

described in the corresponding section.

Yet, we can construct a formal representation of SDs in our 2-hop connection example as follows. Since we know that the Safe scheme equalizes SDs of all nodes, the SDs of node u and node t should be equal, which gives the constraint:

$$\sqrt[\alpha]{\frac{P_{Safe}(s)}{P_{min}}} - d_{su} = \sqrt[\alpha]{\frac{P_{Safe}(u)}{P_{min}}} - (D - d_{su}) \tag{4.9}$$

$$\sqrt[\alpha]{P_{Safe}(s)} - \sqrt[\alpha]{P_{Safe}(u)} = \sqrt[\alpha]{P_{min}}.(2d_{su} - D) \tag{4.10}$$

For simplicity, lets assume $\alpha = 2$, and then take the square of both sides.

$$P_{Safe}(s) + P_{Safe}(u) - 2\sqrt{P_{Safe}(s).P_{Safe}(u)} = P_{min}.(2d_{su} - D)^2 \tag{4.11}$$

$$P_{budget} - 2\sqrt{P_{Safe}(s).(P_{budget} - P_{Safe}(s))} = P_{min}.(2d_{su} - D)^2 \tag{4.12}$$

$$\sqrt{P_{Safe}(s).(P_{budget} - P_{Safe}(s))} = \frac{1}{2}(P_{budget} - P_{min}.(2d_{su} - D)^2) \tag{4.13}$$

$$P_{Safe}(s)^2 - P_{budget}.P_{Safe}(s) + \frac{1}{4}(P_{budget} - P_{min}.(2d_{su} - D)^2)^2 = 0$$

The third term in this equation contains constants (i.e. P_{budget} and P_{min}) and a variable d_{su}. Let's represent it as a single variable $c(d_{su}) = \frac{1}{4}(P_{budget} - P_{min}.(2d_{su} - D)^2)^2$. This gives us

$$P_{Safe}(s)^2 - P_{budget}.P_{Safe}(s) + c(d_{su}) = 0 \tag{4.14}$$

The solution of the quadratic Equation 4.14 gives us the transmission power of Source node s for the Safe scheme, as shown in Equation 4.15.

$$P_{Safe}(s) = \begin{cases} \frac{P_{budget}}{2} - \frac{1}{2}\sqrt{P_{budget}^2 - 4c(d_{su})} & when \quad d_{su} < D/2 \\ \frac{P_{budget}}{2} + \frac{1}{2}\sqrt{P_{budget}^2 - 4c(d_{su})} & when \quad d_{su} > D/2 \end{cases} \tag{4.15}$$

Figure 4.10 indicates transmission power of the nodes for both Sqr and Safe schemes. According to the figure, Safe scheme changes transmission power of the nodes linearly

as the relay node, i.e. node u, moves towards the Destination. On the other hand, the Sqr scheme changes the transmission power of the nodes quadratically.

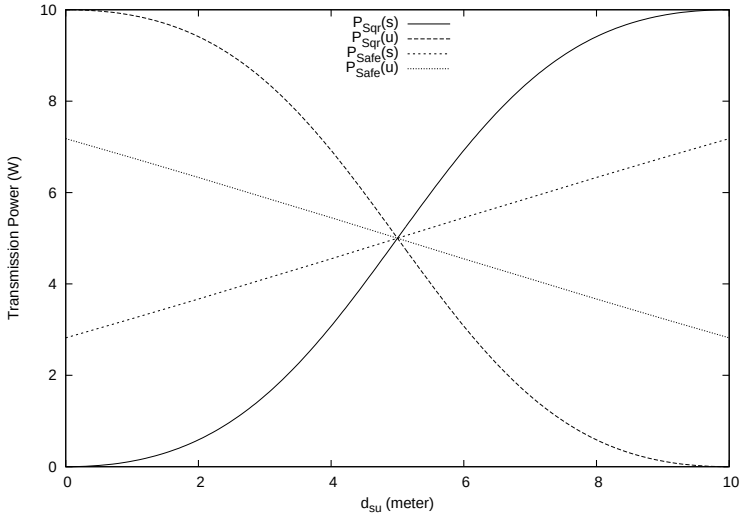


Figure 4.10: Power distribution for both Sqr and Safe schemes

Figure 4.11 depicts SDs of the nodes for both Sqr and Safe schemes. It is obvious in the figure that the Safe scheme yields almost constant and uniform SDs for all nodes, which shows that the Safe scheme is able to remove the aforementioned deficiency of the Sqr scheme (i.e. yielding unbalanced SDs).

This is because, in spite of the fact that Sqr scheme equalizes received signal strength and thus yields the same quality of links at each node, the robustness of links against node movements becomes different. More specifically, shorter links become much more fragile in the face of mobility than longer links does, since shorter links have a smaller radius of safe region, in which downstream node con move without causing link break.

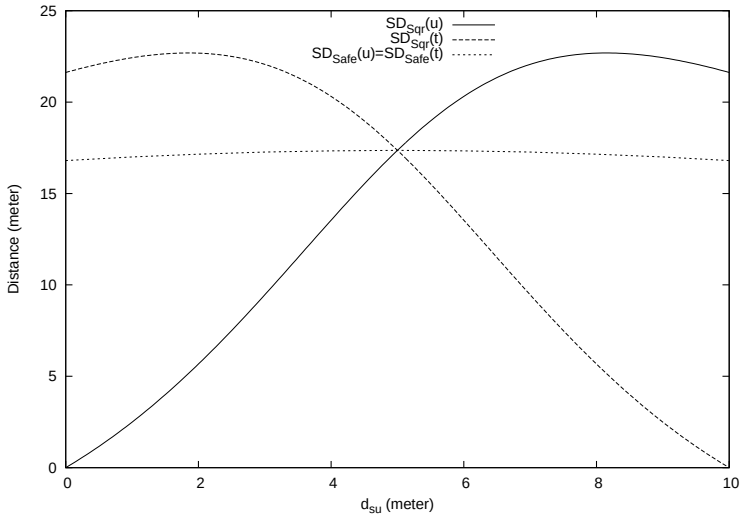


Figure 4.11: Safe distances of node u and t for Sqr and Safe schemes

4.9.2 Safe Scheme

The power budget is distributed among the nodes so that the safe distance of every node is equal, where safe distance is defined as the minimum distance for a node to go out of the coverage area of its upstream neighbor (see Figure 4.12).

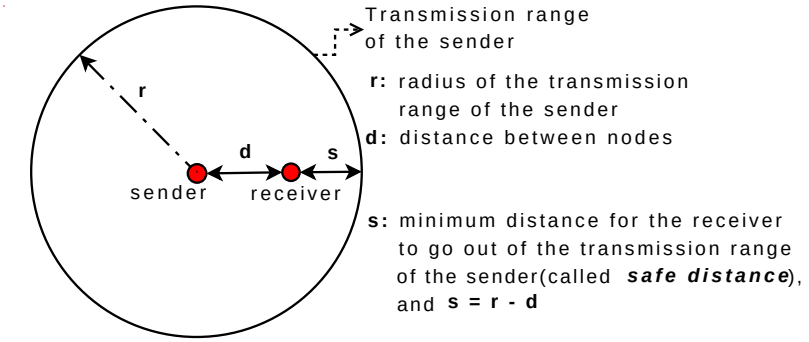


Figure 4.12: Safe scheme description

Implementation of this protocol is similar to *Sqr* scheme's implementation, however because computing the safe distance of each node requires all distance values between nodes to be known (not just their cumulative value as in *Sqr* scheme), the message initiated by the source node in phase 1 keeps a distance vector rather than just

one cumulative value. This makes the resulting message grow linearly in size as it traverses the connection. In addition, in phase 2, the destination must inform each node individually about its new power Tx values. Because of these reasons, control message traffic is higher and convergence time is longer than for *Sqr*.

4.9.3 ModSqr Scheme

This scheme is a modification of the *Sqr* scheme with the following power distribution formula.

$$P_{sqr}(j) = P \cdot (d_j + c)^2 / \sum_{i=1}^{N-1} (d_i + c)^2,$$

where c is a constant (taken as 3 meters in this study), d_j is the distance from node j to node $j + 1$ along the path. Note that *ModSqr* is a continuous parametric family spanning the *Sqr* ($c = 0$) and *Uniform* ($c = \infty$). The control traffic required to implement *ModSqr* is comparable to that of *Sqr* since only a cumulative value needs to be conveyed.

4.10 Control Traffic Overhead

In this part of experimental setup, we will only explain what changed with respect to previous experiments. The rest of the setup remains the same as in Section 6.

4.10.1 Experimental Setup

Initial network design. The simulation is realized at packet level such that each node handles one either data or control packet at each simulation step, assuming that there is spatial reuse. We considered two different lower layer implementations in these experiments:

- *Soft TDM (Time Division Multiplexing) Mode:* There are an ε fraction of the time slots assigned to carry control traffic, but these slots can be used for data traffic if there is no control packet to be forwarded.
- *Hard TDM Mode:* As above, except control slots cannot be reallocated to carry data packets.

Performance measures. As performance, we take the number of data packets received by the destination node before the connection breaks. The gain of a power distribution scheme is calculated by comparing its performance against that of the *Uniform* scheme. We conduct 10^3 independent trials, each trial beginning with a different random initial network and movement sequence; aggregate metrics are then computed as averages over the trials:

$$\text{Gain}_A^* = E[\frac{\text{Number of the data messages of Scheme A}}{\text{Number of the data messages of Unif}}]$$

System and environmental parameters. We explore the impact of these situational parameters on gain of each scheme:

- *Initial node density δ:* We vary node density in the initial network from sparse (0.01 nodes/m) to dense (0.05 nodes/m). This is achieved by taking $N = 500 * \delta$ and scaling the network geometry proportionally so that the initial network is bounded by the 500 meter square.
- *Control time slots percentage ϵ:* This parameter determines what percentage of total time slots are assigned for control traffic, ranging from 0 to 100. We fixed this value as 0.2 (20%) for all experiments except the last experiment, gain verus ϵ.

Fixed parameters. We consider *movement frequency* to be fixed, at one movement every 200 packet transmissions, for each node. The *Signal attenuation exponent* α was taken as 2, appropriate to our connection distance scales in free space. The minimum signal power at which a receiver is able to decode a signal is taken as $P_{min} = 10$ mW.

4.10.2 Simulation Results and Analysis

We conducted experiments to quantify the influence of the number of nodes N, connection power budget P, initial node density δ, and control traffic overhead ϵ, on the expected gain of the schemes over the *Uniform* scheme.

Note: In cases where pairs of curves in our graphs lie within one standard deviation of each other's points, we provide the ***correlation coefficient*** between the per-

trial gains of the two schemes, so as to justify any conclusions based on the relative geometry of the two curves.

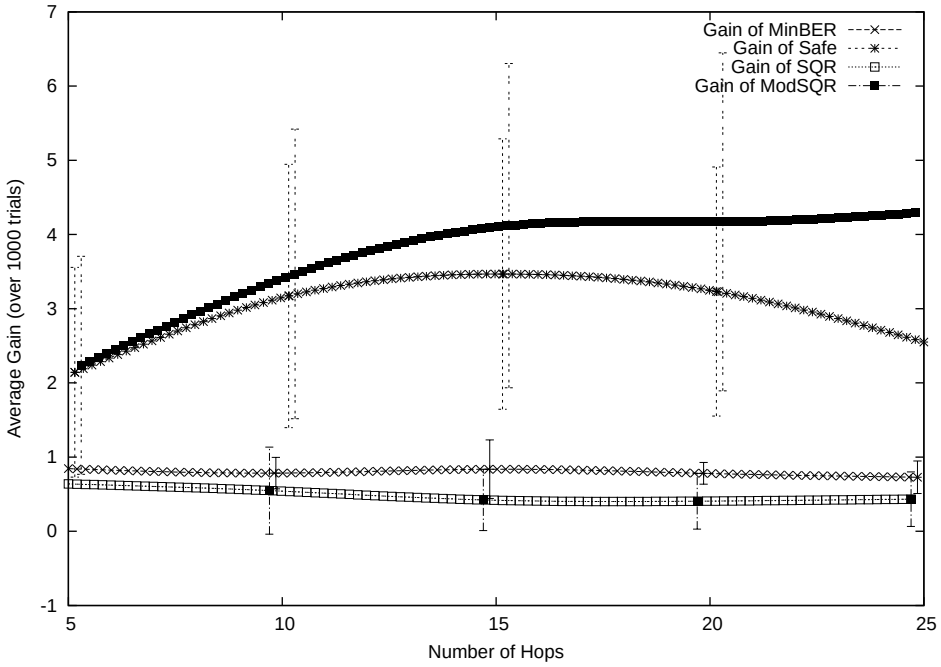


Figure 4.13: The Influence of Number of Hops on Gain in Soft TDM Mode

A. Varying the connection size. In the first set of experiments, we varied the connection size from 5 to 25 nodes, while keeping all other variables fixed: the power budget (P) was fixed at 10.0W, and the initial mean node density was one node every 100m.

There are three significant factors (which depend on the number of hops) that affect gain: (i) the collective resources available to the nodes to compensate for weak links, (ii) the convergence time of the control protocol (especially for end-to-end operations), and (iii) control traffic overhead of the schemes. For *ModSqr* scheme, as the number of hops increases, the factor (i) increases faster than (ii), while (iii) is remains constant per node. *ModSqr* enjoys a linearly growing gain up to the connection size of 15, after which the first two factors balance each other, and the curve becomes flat as

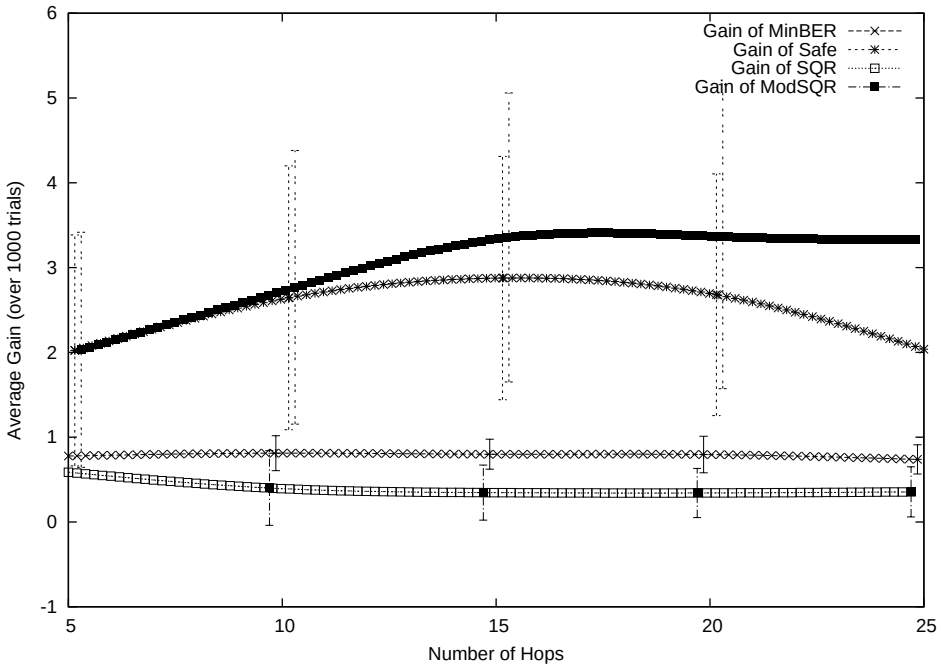


Figure 4.14: The Influence of Number of Hops on Gain in Hard TDM Mode

in Figure 4.13 and 4.14. For *Safe* scheme, factor (ii) begins to dominate factor (i) because increased message complexity causes the convergence time of *Safe* scheme to grow quadratically in connection size. The correlation between *ModSqr* and *Safe* was (0.98), showing that the former genuinely outperforms the latter, in spite of the fact that the two curves lie within each other's error bars. The gain of *Minber* is independent of the connection size because it works within triplets and it does not use an end-to-end protocol. Finally, *Sqr* and *minBER* were the worst scheme in terms of gain as seen in the figures. *Sqr* performs poorly because it leaves short connection links disproportionately vulnerable to endpoint mobility. *minBER* performs poorly because its objective is orthogonal to expected longevity. The correlation between *MinBER* and *Sqr* was (0.21) indicating that the schemes cannot be compared to each other, as they lie within each other's error bars.

B. Varying the power budget. In the second set of experiments, we varied the

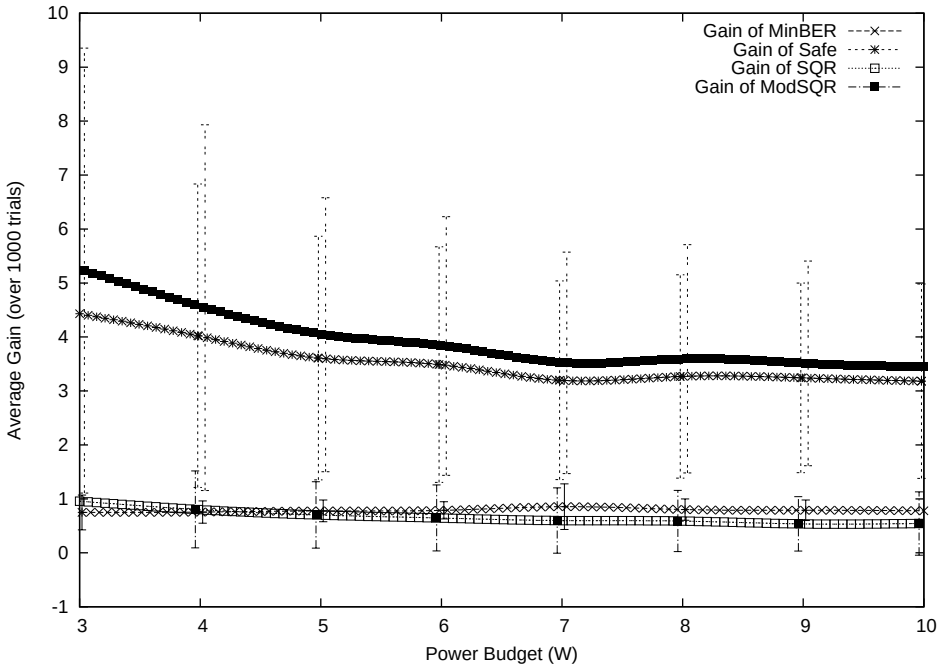


Figure 4.15: The Influence of Power Budget on Gain in Soft TDM Mode

connection's power budget from 3.0W to 10.0W, while keeping all other variables fixed: the number of nodes (N) was fixed at 10, and mean node density was one node every 100m.

Clearly the gains of the schemes should approach 1 as the power budget goes to infinity. This intuition is born out in Figures 4.15 and 4.16. In spite of this, ModSqr and Safe schemes outperform the Sqr and minBER schemes, providing a gain of over 300% even for very high power budgets.

C. Varying the node density. In the third set of experiments, we varied the node density from one node every 100m ($\delta = 0.01$ nodes/meter) to one node every 20m ($\delta = 0.05$ nodes/meter), while keeping all other variables fixed with the power budget (P) of 10.0W. The number of nodes N was taken to be 500δ, and the initial configuration was rescaled proportionately so that it was bounded by a 500m by 500m

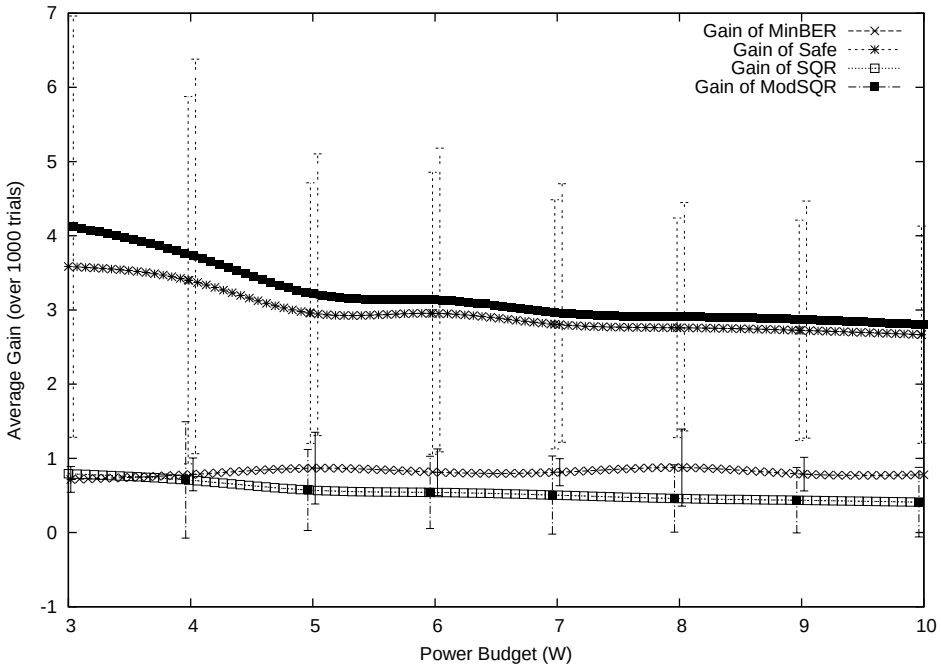


Figure 4.16: The Influence of Power Budget on Gain in Hard TDM Mode

square.

This experiment is closely related to the investigations of hop size since changing node density implies changing hop size of a connection. The results of both experiments thus look similar (Figure 4.17 and 4.18) and the interpretations made in the hop-size experiment are also valid here. The correlation between *ModSqr* and *Safe* was (0.95), showing that the former genuinely outperforms the latter, in spite of the fact that the two curves lie within each other's error bars. The *Sqr* scheme exhibits better performance when node density is low since when the nodes are denser the probability of having short links gets higher. As noted earlier, *Sqr* causes nodes with short links to adjust their power level so that receiver nodes are at the precipice of the transmission radius (i.e. their safe distance is very small); this yields early disconnections of short links.

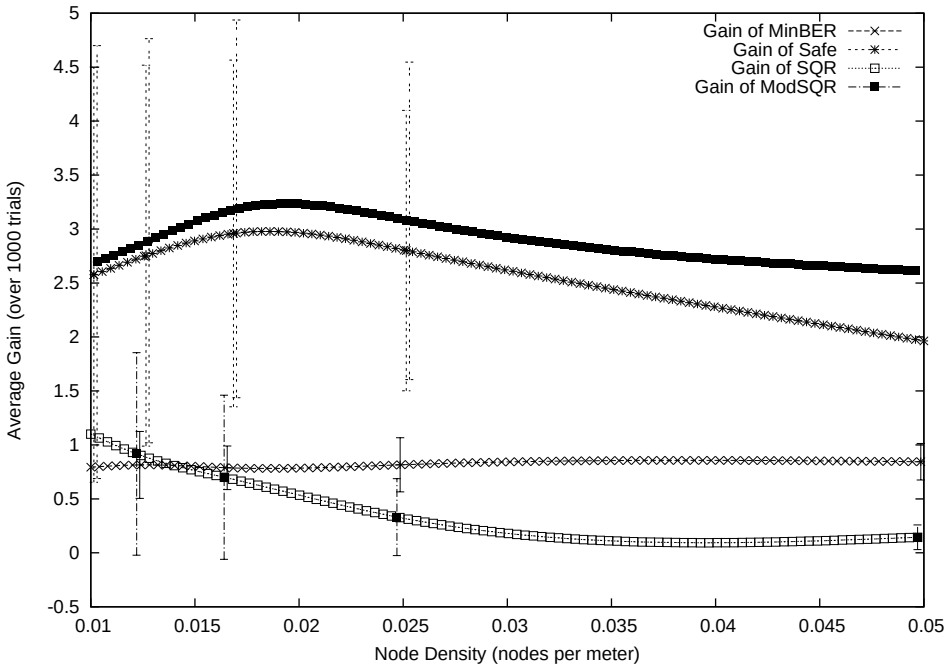


Figure 4.17: The Influence of Node Density on Gain in Soft TDM Mode

D. Varying the control traffic overhead. In the final set of the experiments, we varied the control traffic overhead by changing fraction of slots (ε) allocated to control traffic, while keeping other variables fixed: power budget (P) was fixed at 10.0W, the number of nodes (N) was fixed at 10 , and the mean node density was one node every 100m.

Figure 4.19 shows the effect of *Epsilon* on the gain in Soft TDM mode. *ModSqr* and *Safe* show similar response herein, but *ModSqr*'s gain is higher than *Safe*'s gain since *Safe* has higher control traffic (and hence longer convergence times) than *ModSqr*. Note that the correlation between *ModSqr* and *Safe* was (0.97), showing that the former genuinely outperforms the latter, in spite of the fact that the two curves lie within each other's error bars. Both curves initially rise linearly up to around ε=0.25, then remain flat. When the ε is too small, there are few control time slots and the schemes run and converge slower than the side effects of node mobility, resulting in

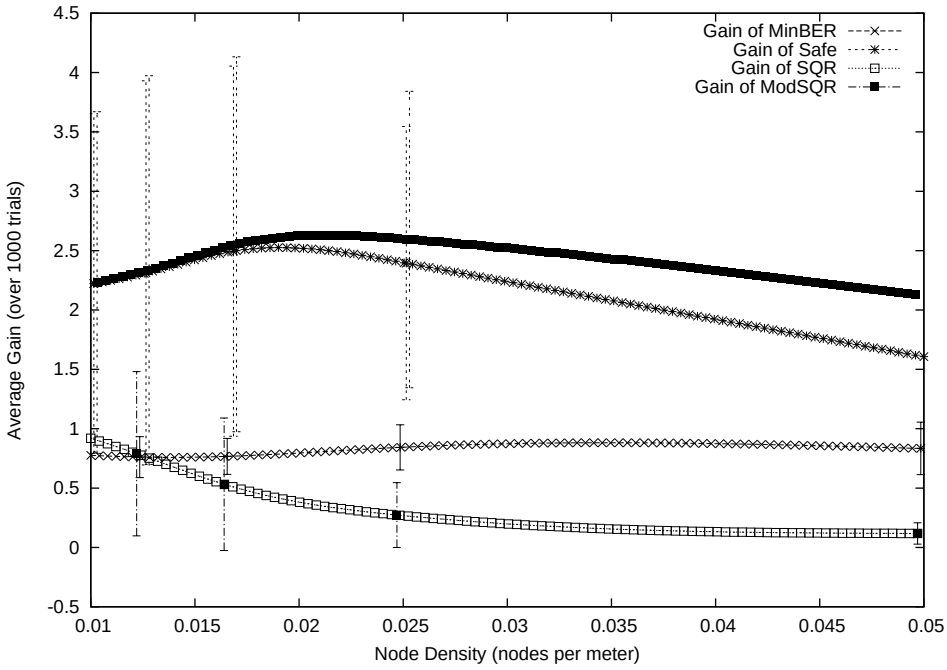


Figure 4.18: The Influence of Node Density on Gain in Hard TDM Mode

lower performance. Once the percentage of control time slots reaches a reasonable value (at about ε=0.25) the schemes obtain maximum gain level and increasing ε beyond this yields nothing in terms of the gain. Assigning too large an ε is both useless and *harmless* due to fact that the schemes is already converged and the nodes are allowed to handle data packets in the absence of control packets respectively in Soft TDM. In the same figure, the gain of *MinBER* decreases linearly as ε increases, and when ε=1.0 its gain becomes zero; this is understandable since in *minBER* the nodes always have control messages to send and will thus swamp out data if given the opportunity. The correlation between *minBER* and *Sqr* was low (0.13), indicating that the two schemes are incomparable, since their curves lie within each other's error bars. Figure 4.20 considers the effect of varying ε in Hard TDM mode. The correlation between *ModSqr* and *Safe* was (0.99). According to the figure, there are tradeoffs for *ModSqr* and *Safe* schemes such that they reach their maximum gain

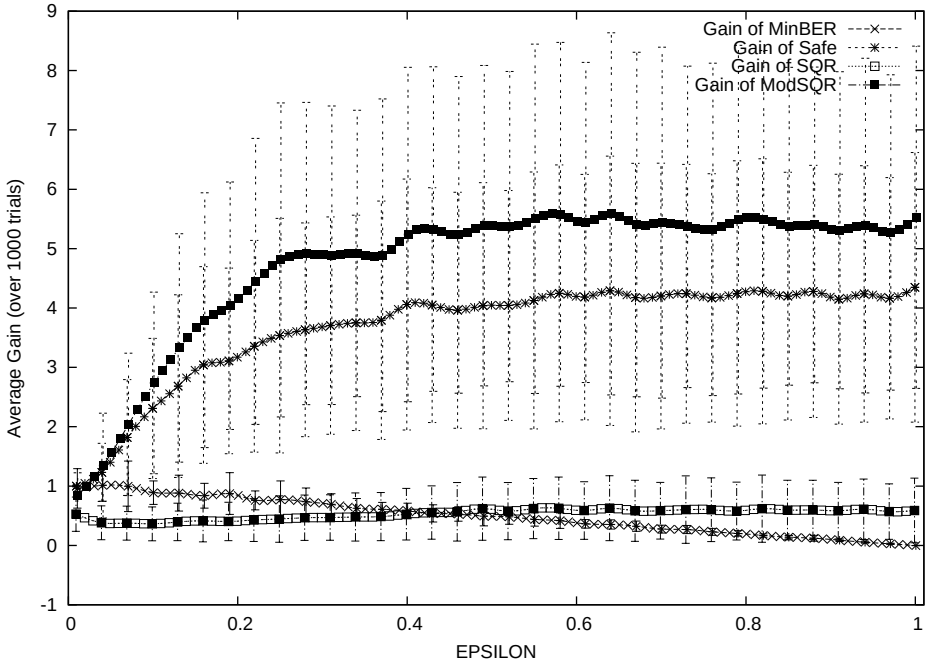


Figure 4.19: The Influence of Control Traffic on Gain in Soft TDM Mode

values when ε=1/4. As ε increases to 1.0 the gains of the all schemes decrease linearly and approach 0, since in Hard TDM control slots will swamp out data slots if ε is set too high.

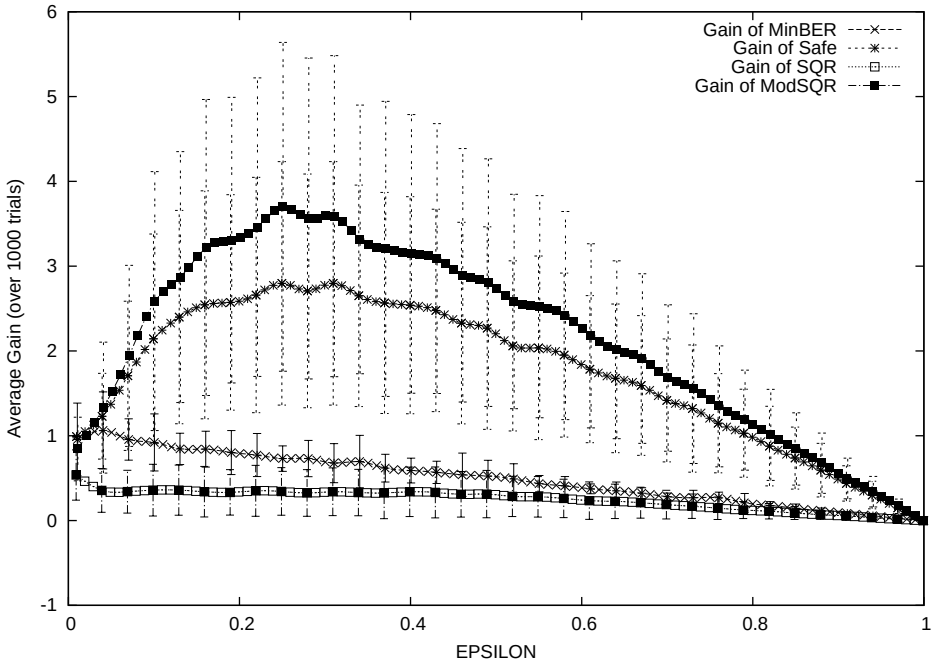


Figure 4.20: The Influence of Control Traffic on Gain in Hard TDM Mode

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

The ideas and schemes developed in this thesis address the problem of dynamic optimization of connections in MANETs. Two broad and complementary classes of approaches are considered ;

- **Variable topology, fixed power** approaches which assume that the transmission power of the nodes is kept fixed, but the topology of the connections is modifiable during their lifetimes. The schemes presented in Chapter 2 and 3 belong to this group.
- **Variable power, fixed topology** approaches which assume that the topological structure of the connection must be kept fixed, but the transmission power levels used by constituent nodes is adjustable. The schemes proposed in Chapter 4 fall into this group.

5.1 Route Optimization

In Chapter 2, we proposed and evaluated two different route optimization schemes called **1-hop Shrink** and **Multihop Shrink** as extensions to Ad hoc On Demand Distance Vector (AODV). These schemes sought to counteract the inefficiencies in

connection topology arising from node mobility. More specifically, they try to optimize active connections in terms of route length by eliminating inessential hops on paths. The proposed methods use special packets to detect and modify suboptimal active routes rather than relying on promiscuous mode. Although they have the same objective (i.e. route optimization), there are two major differences between the approaches they use to achieve the intended objective:

- *Unicasting vs. broadcasting:* 1-hop Shrink uses unicast packets while performing its actions, whereas Multihop Shrink uses broadcast packets.
- *Triplet nodes vs. end-to-end:* 1-hop Shrink is able to detect shortcuts within triplet of nodes, whereas Multihop Shrink can detect shortcuts between any pair of nodes on an active connection.

Through extensive ns2 simulations, we showed that the *1-hop Shrink* method improves path optimality and performs well with respect to routing overhead incurred. The experimental results showed that the proposed *1-hop Shrink* scheme decreases end-to-end delay experienced in packet delivery, as well as provides power savings in packet transmission due to reduction of hop counts. In addition, dynamic route optimization on active connections yielded increased connection lifetimes.

Analysis of ns2 simulations demonstrated that the proposed Multihop Shrink scheme yields significant improvements over classical AODV in terms of path optimality. It also reduces the number of repair-related global route discoveries experienced by long lived connections. Indeed, the simulations indicate that the control traffic needed to operate our Multihop Shrink scheme can be amortized against the corresponding reduction in repair-related global route discovery traffic. Thus, the Multihop Shrink

modification to AODV achieves intended objective and exhibits less control traffic than even standard AODV. We also showed through experiments that the proposed scheme enjoys several improvements in related performance metrics, such as end-to-end delay and power consumption. Our conclusions continue to hold scalably over a wide range of mobility settings scenarios.

In comparing the 1-hop Shrink and Multihop Shrink schemes to each other, we observed that both yield almost the same level of improvement in path optimality, energy consumption, and control traffic. However, we determined that the Multihop Shrink scheme outperformed the 1-hop Shrink scheme in terms of packet delivery fraction. The experiments also indicated that 1-hop Shrink scheme outperformed Multihop Shrink slightly, in both end-to-end delay and average connection lifetime.

5.2 Route Maintenance

In Chapter 3, we proposed and evaluated two different schemes called **Expansion** and **LocOpt**. Both schemes aim to address link failures that may occur due to node mobility. The most significant difference between the two schemes is the time at which they act, relative to link breakage. The former scheme performs its role just before a potential link failure, with the objective of preventing the breakage. The latter scheme operates after a link failure, and aims to heal the breakage by performing a local recovery operation. Both schemes improve the scalability of a network because they are local (as opposed to global) solutions the problem of link failure.

The *Expansion* scheme tries to prevent link breaks on active connections by detecting weak links (through measurements of the received signal strength of incoming data

packets), and inserting an extra relay node between its endpoints. Subpaths which include a *weak* link are eliminated whenever possible, by directly connecting the downstream endpoint of the weak link to an upstream node, or the upstream endpoint of the weak link to the final destination. The advantages of the proposed scheme are that it decreases overall control traffic overhead, while retaining high packet delivery fraction. The scheme pays for these improvements through a modest degradation in path optimality. These conclusions were reached by analyzing extensive simulation experiments using ns2.

The *LocOpt* scheme, in contrast with prior local recovery approaches, achieves both link recovery and route optimization simultaneously. In local recovery, a broken link is patched by inserting an extra relay node between its endpoints. Since successive local recovery operations may produce unnecessarily long paths, a secondary route optimization process is initiated. We showed through extensive ns2 simulation experiments that LocOpt makes use of significantly *less control traffic* than standard AODV, while exhibiting connection routes that are *shorter* than those of AODV. Thus, we showed that reactive routing protocols do not contain an inherent tradeoff between control traffic and route optimality. These conclusions were shown to hold scalably as one varies situational parameters such as network size, number of connections, and node mobility.

5.3 Connection Lifetime

In Chapter 4, we considered the problem of how best to distribute a fixed power budget among the nodes of a connection, so as to increase expected connection life-

times. We developed a new dynamic power redistribution scheme, called Sqr, which yields gains in lifetime that range from a factor of 3 to a factor of 30 (relative to static uniform distributions), depending on the particular values of system and environmental parameters. The expected gain in lifetime was seen to be most sensitive to power budget and connection size, and relatively insensitive to initial node density and mean node velocity.

We also compared dynamic power distribution schemes that minimize end-to-end BER with those seeking to maximize connection lifetime. The objectives of BER minimization and lifetime maximization were most starkly in opposition in settings where the connection power budget is low. In these settings, the distribution of power is most sensitive to the choice of objective. Connections which are declared low priority (and hence assigned low power budgets) cannot expect to simultaneously optimize their power distributions with respect to both BER and expected lifetime, since the two criteria are seen to be in opposition. High-priority connections, on the other hand can "have their cake and eat it too", since with large power budgets, the relative advantage of *Sqr* over *minBER* (with respect to connection lifetime) is diminished.

The *Sqr* scheme equalizes received power levels at all nodes, and so is theoretically optimal in settings where the power distribution protocols operate instantaneously. In practice, however, it is seen to perform poorly when convergence time take into consideration. This realization motivated the design of the *Safe* scheme, which seeks to equalize the safe distances of nodes–that is, the distance that any node can move without breaking the connection. While the *Safe* scheme computes an optimal solution, it is crippled by its high communication complexity. This motivated the development

of the *ModSqr* scheme, which captures the gains of *Safe* without incurring the high communication costs.

We implemented all schemes in a discrete event simulator, under two different lower layer models: (i) *Soft TDM mode*, where there are time slots for control traffic, but these slots can be used to send data traffic if there is no control packet to be forwarded, (ii) *Hard TDM mode*, where there are time slots assigned for control traffic, but nodes cannot use the slots for data packets, even if there is no control packet to send. The investigations demonstrate that taking control traffic overhead and convergence time of protocols play a critical role in the analysis of the proposed schemes.

5.4 Future Work

We categorize our intended future work for each class of approaches:

Variable topology, fixed power investigations: While developing the proposed schemes within this group, the most significant assumption we made was that links between nodes are bidirectional. This was a natural result of having fixed and uniform transmission power, and is a reasonable assumption in a radio propagation model which is hybrid of the free space and two-ray ground reflection models. However, links between nodes may not always be bidirectional in reality, due to issues of shadowing, obstructions, reflections, and interference of wireless signals. We are planning to extend our work by taking into account the fact that links may not be bidirectional.

Secondly, in investigating the impact of data traffic load on the performance of our proposed schemes, we considered independent point-to-point connections. In future,

we would like to assess our proposed schemes in point-to-multipoint and group multicast settings.

Thirdly, we implemented our proposed schemes as extensions to the popular reactive routing protocol, AODV. There is also another subclass of reactive routing, namely source routing based protocols, such as DSR. In this type of reactive routing protocols, the source node inserts the IP addresses of all nodes along the route into data packet headers, so other nodes receiving a data packet can infer the next hop from header of received data packet. We are planning to adapt our proposed schemes to this type of reactive protocols, and extend them to leverage having all node IDs along the route.

Fourth, while measuring the scalability of our proposed schemes, we have considered simulations of MANETs consisting of 100 nodes. Since the problems we are trying to address in our proposed schemes are much more severe for larger networks, we would like to measure their performance at bigger scales (both in terms of the number of nodes and deployment area).

Fifth, we have shown through formal argument that the proposed schemes do not harm loop free property of AODV. However, we would also like to confirm this by implementing our schemes in a model checker such as SPIN.

Sixth, we are planning to implement our schemes in real test-beds, and conduct real performance testing.

Finally, in the proposed route optimization schemes, we sought to find 1-hop shortcuts (i.e. direct connections), if any, between node pairs on a connection. In other words, our proposed schemes replace n-hop subconnection with a direct link. However, there may be shortcuts containing more than 1-hop, which are able to shorten the path.

We would like to extend our research by taking into account these types of shortcuts. In addition, once a shortcut is applied, the eliminated hops are not taken into account in seeking future shortcuts, since they are no longer lying on the connection. In our future work, we will consider how to use these eliminated nodes.

Variable power, fixed topology investigations: While developing power distribution schemes, we assumed that there is no upper or lower bound for node transmission power. However, this is not a case in reality. We intend to lift this assumption, assess our schemes, and adapt them accordingly.

Second, we considered the case of single connection in our evaluations. We are planning to apply our schemes networks with many simultaneous point-to-point connections between different source and destination nodes.

Third, in our work to date, we did not take into consideration the difficulties in communication over unidirectional links, which is the case owing to nonuniform transmission power settings. We are planning to modify and extend the communication protocol of our proposed power distribution schemes to take into account unidirectional links.

Finally, we assumed that nodes are able to adjust their transmission power for each packet they transmit. This may not be possible in current devices, as has been noted by Kawadia and Kumar [26]. Therefore, we are planning to make more restrictive assumptions on the temporal frequency of control.

REFERENCES

[1] *Gnuplot software*. http://www.gnuplot.info/, access date: April 2010.

[2] *IEEE 802.11 standard*. http://www.ieee802.org/11/, access date: April 2010.

[3] *Network Simulator, ns2*. http://www.isi.edu/nsnam/ns/, access date: April 2010.

[4] *Network Simulator, ns2*. http://nsnam.isi.edu/nsnam/index.php/Main_Page, access date: April 2010.

[5] *Network Simulator, ns2*. http://www.isi.edu/nsnam/ns/doc/everything.html, access date: April 2010.

[6] A. Abbas, J. Bahi, and A. Mostefaoui. Improving wireless ad hoc networks lifetime. Proc. of the IEEE Int. Conference on Sensor Networks, Ubiquitous, and Trustworthy Computing (SUTC), 2006.

[7] M. Al-Shurman, S.-M. Yoo, and S. Park. A performance simulation for route maintenance in wireless ad hoc networks. In *ACM-SE 42: Proceedings of the 42nd annual Southeast regional conference*, pages 25–30, New York, NY, USA, 2004. ACM.

[8] A. M. Alshanyour and U. Baroudi. Bypass AODV: improving performance of ad hoc on-demand distance vector (aodv) routing protocol in wireless ad hoc networks. In *Ambi-Sys '08: Proceedings of the 1st international conference on Ambient media and systems*, pages 1–8, ICST, Brussels, Belgium, Belgium, 2008. ICST (Institute for Computer Sciences, Social-Informatics and Telecommunications Engineering).

[9] B. Awerbuch and D. Peleg. Online tracking of mobile users. *J. ACM*, 42(5):1021–1058, 1995.

[10] G. B. Brahim and B. Khan. Budgeting power: Packet duplication and BER reduction in wireless ad hoc networks. International Wireless Communications and Mobile Computing Conference, IWCMC, Vancouver, Canada, 2006.

[11] G. B. Brahim, B. Khan, A. Al-Fuqaha, and M. Guizani. Using energy efficient overlay to reduce packet error rates in wireless ad hoc networks. International Conference on Communications, ICC, 2006.

[12] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu, and J. Jetcheva. A performance comparison of multihop wireless ad hoc network routing protocols. pages 85–97, 1998.

[13] T. X. Brown, B. Argrow, C. Dixon, S. Doshi, R.-G. Thekkekunnel, and D. Henkel. Ad hoc UAV ground network. pages 20–23. AIAA 3rd Unmanned Unlimited Technical Conference, Chicago, IL, 2004.

[14] T. Camp, J. Boleng, and V. Davies. A survey of mobility models for ad hoc network research. *Wireless Communication and Mobile Computing (WCMC): Special issue on Mobile Ad Hoc Networking: Research, Trends and Applications*, 2(5):483–502, 2002.

[15] I. D. Chakeres and L. Klein-Berndt. AODVjr, AODV simplified. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):100–101, 2002.

[16] Z. Cheng and W. B. Heinzelman. Exploring long lifetime routing (LLR) in ad hoc networks. In *MSWiM '04: Proceedings of the 7th ACM international symposium on Modeling, analysis and simulation of wireless and mobile systems*, pages 203–210, New York, NY, USA, 2004. ACM.

[17] S. Corson and J. Macker. *Internet Engineering Task Force, IETF.* http://www.ietf.org/rfc/rfc2501.txt, January 1999.

[18] S. K. Das, A. Mukherjee, S. Bandyopadhyay, K. Paul, and D. Saha. Improving quality-of-service in ad hoc wireless networks with adaptive multi-path routing. GLOBECOM, 2000.

[19] Q. Dong and S. Banerjee. Minimum energy reliable paths using unreliable wireless links. MobiHoc'05, Urbana-Champaign, Illinois, May 25-27, 2005.

[20] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi. Signal stability based adaptive routing (SSA) for ad-hoc mobile networks. *IEEE Personal Communications*, 4:36–45, 1997.

[21] H. T. Friis. A note on a simple transmission formula. *Proc. IRE*, vol.34:pages 254, 1946.

[22] V. C. Giruka, M. Singhal, and S. P. Yarravarapu. A path compression technique for on-demand ad-hoc routing protocols. In *Mobile Ad-hoc and Sensor Systems (MASS) 2004*, 2004.

[23] C. Gui and P. Mohapatra. Short: self-healing and optimizing routing techniques for mobile ad hoc networks. In *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile ad hoc networking & computing*, pages 279–290, New York, NY, USA, 2003. ACM.

[24] X. Jia, L. Qian-mu, Z. Hong, and L. Feng-yu. Model and analysis of path compression for mobile ad hoc networks. *Computers and Electrical Engineering*, In Press, Corrected Proof, 2009.

[25] D. B. Johnson, D. A. Maltz, and J. Broch. DSR: The dynamic source routing protocol for multihop wireless ad hoc networks. In *Ad Hoc Networking, edited by Charles E. Perkins, Chapter 5*, pages 139–172. Addison-Wesley, 2001.

[26] V. Kawadia and P. R. Kumar. Principles and protocols for power control in wireless ad hoc networks, 2005.

[27] B. Khan, G. B. Brahim, A. Al-Fuqaha, and M. Guizani. Minimizing wireless connection BER through the dynamic distribution of budgeted power. Proceedings of IEEE Globecom, 2006.

[28] M. Kim and J. Jang. AODV based energy efficient routing protocol for maximum lifetime in MANET. Proc. of the Adv. International Conference on Telecommunications and International Conference on Internet and Web Applications and Services (AICT/ICIW), 2006.

[29] J. F. Kurose and K. W. Ross. *Computer Networking: A Top-Down Approach*. Addison Wesley, 2009.

[30] S. M. Lee and K. Kim. *An Effective Path Recovery Mechanism for AODV Using Candidate Node*, volume 4331, pages 1117–1125. LNCS, ISPA Workshops, 2006.

[31] V. Lenders, J. Wagner, and M. May. Analyzing the impact of mobility in ad hoc networks. In *REALMAN '06: Proceedings of the 2nd international workshop on Multi-hop ad hoc networks: from theory to reality*, pages 39–46, New York, NY, USA, 2006. ACM.

[32] P. J. Lewis, M. R. Torrie, and P. M. Omilon. Applications suitable for unmanned and autonomous missions utilizing the Tactical Amphibious Ground Support (TAGS) platform. In G. R. Gerhart, C. M. Shoemaker, & D. W. Gage, editor, *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, volume 5422 of *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, pages 508–519, Sept. 2004.

[33] N. Li, J. C. Hou, and L. Sha. Design and analysis of an MST-based topology control algorithm. IEEE INFOCOM, 2003.

[34] G. Lim. Link stability and route lifetime in ad-hoc wireless networks. In *ICPPW '02: Proceedings of the 2002 International Conference on Parallel Processing Workshops*, page 116, Washington, DC, USA, 2002. IEEE Computer Society.

[35] G. Liu, K.-J. Wong, B.-S. Lee, B.-C. Seet, C.-H. Foh, and L. Zhu. PATCH: A novel local recovery mechanism for mobile ad hoc networks. In *Proc. of IEEE Vehicular Technology Conference*, pages 2995–2999, 2003.

[36] S. Medidi and J. Wang. A fault resilient routing protocol for mobile ad-hoc networks. In *WIMOB '07: Proceedings of the Third IEEE International Conference on Wireless and Mobile Computing, Networking and Communications*, page 41, Washington, DC, USA, 2007. IEEE Computer Society.

[37] C. S. R. Murthy and B.S.Manoj. *Ad Hoc Wireless Networks Architectures and Protocols*. Prentice Hall PTR, 2004.

[38] M. Pan, S.-Y. Chuang, and S.-D. Wang. Local repair mechanisms for on-demand routing in mobile ad hoc networks. In *PRDC '05: Proceedings of the 11th Pacific Rim International Symposium on Dependable Computing*, pages 317–324, Washington, DC, USA, 2005. IEEE Computer Society.

[39] S. Park and B. Voorst. Anticipated route maintenance (ARM) in location-aided mobile ad hoc networks. Globecom, San Antonio, TX, USA, 2001.

[40] C. Perkins, E. B-Royer, and S. Das. Ad hoc on-demand distance vector (AODV) routing. Internet Engineering Task Force, 2003.

[41] C. Perkins, C. E. P. Nokia, E. M. Royer, S. R. Das, and M. K. Marina. Performance comparison of two on-demand routing protocols for ad hoc networks. *IEEE Personal Communications*, February:16–28, 2001.

[42] C. E. Perkins and E. M. Royer. Ad-hoc on-demand distance vector routing. In *IEEE WORKSHOP ON MOBILE COMPUTING SYSTEMS AND APPLICATIONS*, pages 90–100, 1999.

[43] L. Qin and T. Kunz. Proactive route maintenance in DSR. *SIGMOBILE Mob. Comput. Commun. Rev.*, 6(3):79–89, 2002.

[44] R. Ramanathan and R. Rosales-Hain. Topology control of multihop wireless networks using transmit power adjustment. IEEE Infocom, 2000.

[45] T. Rappaport. *Wireless Communications: Principles and Practice*. Prentice Hall PTR, Upper Saddle River, NJ, USA, 2001.

[46] V. Rudolph and T. Meng. Minimum energy mobile wireless network. *IEEE Journal of Selected Areas in Communications*, 17(8), 1999.

[47] M. Saito, H. Aida, Y. Tobe, and H. Tokuda. A proximity-based dynamic path shortening scheme for ubiquitous ad hoc networks. In *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS04*, 2004.

[48] P. Santi. Topology control in wireless ad hoc and sensor networks. *ACM Comput. Surv.*, 37(2):164–194, 2005.

[49] R. Shi and Y. Deng. An improved scheme for reducing the latency of AODV in ad hoc networks. In *ICYCS '08: Proceedings of the 2008 The 9th International Conference for Young Computer Scientists*, pages 594–598, Washington, DC, USA, 2008. IEEE Computer Society.

[50] M. F. Sjaugi, M. Othman, and M. F. A. Rasid. A new distance based route maintenance strategy for dynamic source routing protocol. *Journal of Computer Science*, 4(3):172–180, 2008.

[51] A. Srinivas and E. Modiano. Minimum energy disjoint path routing in wireless ad-hoc networks. MobiCom'03, San Diego, California, September 14-19, 2003.

[52] C.-K. Toh. Associativity-based routing for ad-hoc mobile networks. *Wireless Personal Communications*, 4(2):1–36, 1997.

[53] Y.-C. Tseng, Y.-F. Li, and Y.-C. Chang. On route lifetime in multihop mobile ad hoc networks. *IEEE Transactions On Mobile Computing*, 2(4), 2003.

[54] R. Wattenhofer, L. Li, P. Bahl, and Y.-M. Wang. Distributed topology control for power efficient operation in multihop wireless ad hoc networks. IEEE Infocom, 2001.

[55] R. Wattenhofer and A. Zollinger. XTC: A practical topology control algorithm for ad-hoc networks. In *4th International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, 2003.

[56] S. Wu, S. Ni, Y. Tseng, and J. Sheu. Route maintenance in a wireless mobile ad hoc network. In 33rd Hawaii International Conference on System Sciences, Maui, 2000.

[57] X.Wu, H. Sadjadpour, and J. J. Garcia-Luna-Aceves. From link dynamics to path lifetime and packet-length optimization in manet. *Wireless Networks, Springer Science+Business Media, LLC*, 2007.

[58] Y.-S. Yen, H.-C. Chang, R.-S. Chang, and H.-C. Chao. Routing with adaptive path and limited flooding for mobile ad hoc networks. *Computers and Electrical Engineering*, 36(2):280 – 290, 2010.

[59] M. G. Zapata. Shortcut detection and route repair in ad hoc networks. In *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, pages 237–242, Washington, DC, USA, 2005. IEEE Computer Society.

[60] Y. Zhang and L. Cheng. Cross-layer optimization for sensor networks. New York Metro Area Networking Workshop, New York, September 12, 2003.

INDEX