Performance comparison of two route optimisation schemes for AODV in MANETs

Bilal Khan*

Department of Sociology,
University of Nebraska-Lincoln,
Lincoln NE 68588, USA
Email: bkhan2@unl.edu

Zeki Bilgin*

Department of Security,
Gumushane, 29000, Turkey
Email: zbilgin@egm.gov.tr
*Corresponding authors

Abstract: In this study, we extend performance evaluation of two different 'route optimisation' schemes proposed in Bilgin and Khan (2010) and Bilgin et al. (2010), with new performance metrics like normalised energy consumption, packet delivery fraction, average end-to-end delay, and average connection lifetime. In addition, we provide performance comparison of this two schemes against each other with pure AODV as a reference. The compared schemes are implemented as extension of AODV in ns-2 packet level network simulator. The first scheme, called *1-hop shrink*, periodically checks if there is any shortcut (i.e., direct connection) between *two endpoints of each successive and overlapped triplet* of nodes along active routes. On the other hand, the second scheme, called *multi-hop shrink*, periodically checks if there is a shortcut between *any two nodes* on a route by broadcasting a special packet starting from at the third node on active connections.

Keywords: route optimisation; mobility; MANET.

Reference to this paper should be made as follows: Khan, B. and Bilgin, Z. (2016) 'Performance comparison of two route optimisation schemes for AODV in MANETs', *Int. J. Communication Networks and Distributed Systems*, Vol. 17, No. 1, pp.76–102.

Biographical notes: Bilal Khan is a Professor in the Department of Sociology at University of Nebraska-Lincoln, and previously employed in the Department of Mathematics and Computer Science at John Jay College of Criminal Justice, City University of New York (CUNY). He received his BS degree in Mathematics and Computer Science from Massachusetts Institute of Technology (MIT), MS degree in Computer Science from John Hopkins University, and PhD in Mathematics from CUNY. He is the author of over 90 papers in mathematics and networks. His research interest includes social network simulation, wireless ad-hoc networks, system and network security, and discrete dynamical systems.

Zeki Bilgin has PhD in Computer Science at City University of New York (CUNY), as well as MSc and Bachelor degrees in Electrical Engineering from Gazi University, Turkey. He has been involved in many projects related to telecommunication, networking, and video surveillance in Turkish Government. His research interest covers wireless communication, networking, and information security.

This paper is a revised and expanded version of a paper entitled 'A dynamic route optimization mechanism for AODV in MANETs' presented at International Conference on Communications (ICC), Cape Town, South Africa, 23–27 May 2010 and paper entitled 'Only the short die old: route optimization in MANETs by dynamic subconnection shrinking' presented at the International Wireless Communications and Mobile Computing Conference (IWCMC), Caen, France, 28 June to 2 July 2010.

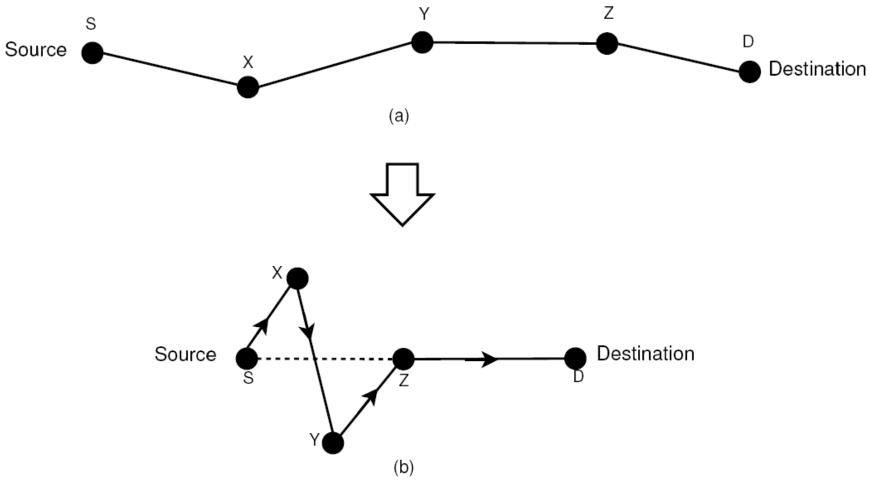
1 Introduction

A mobile ad hoc network (MANET) is a special type of wireless network, in which each node acts as a router, and connections between source and destination pairs are constructed in multi-hop fashion without requiring any telecommunication infrastructure. Thus, issues of routing in this type of networks become much more challenging especially when network topology is highly dynamic. There are plenty of research articles dealing with several issues of routing from different perspectives in wireless mesh or ad hoc networks (Misra et al., 2010, 2014, 2012; Misra and Rajesh, 2011).

Reactive routing protocols like AODV and DSR, in MANETs, initiate route construction processes on-demand between source and destination nodes. In a route establishment process, even when there are many possible alternative routes between related source-destination pair, underlying routing protocol usually selects the one which is the shortest in terms of hop count. This tendency relies on the fact that a shorter route has many advantages with respect to a longer one. For instance, as the hop count on a path decreases, the throughput achieved along this path gradually increases (Li et al., 2001; Sadagopan et al., 2003), the expected path lifetime becomes longer (Cheng and Heinzelman, 2004; Lenders et al., 2006; Lim, 2002; Raw et al., 2012; Sadagopan et al., 2003; Tseng et al., 2003; Wu et al., 2007), the energy efficiency increases (Li, 2009), the reliability of path increases (Mamoun, 2011), end-to-end (E2E) delay in packet delivery reduces (Kolar et al., 2005; Wu et al., 2000; Zhai and Fang, 2006), and the number of transmission required for delivering a packet from source node to destination node reduces (Kolar et al., 2005; Zhai and Fang, 2006).

Although many routing protocols try to employ shortest path approach for routing because of the benefits mentioned above, they mostly fail to manage this due to fact that a path which is the shortest at the beginning of a connection usually does not remain as the shortest during lifetime of this connection due to mobility of nodes causing highly dynamic topology in MANET, as illustrated in Figure 1.

Figure 1 (a) Initially optimal route (b) How mobility causes sub-optimality



Main contribution. This work has some unique contributions such that we, first of all, present a *comprehensive* survey of route optimisation algorithms in the literature, and then introduce a *mathematical model* of the route optimisation problem in MANETs. Later on, we perform *performance comparison* of two different route optimisation algorithms proposed by Bilgin and Khan (2010) and Bilgin et al. (2010). In doing this, we bring a *new* approach calculating correlation coefficients of mean values of compared curves, with the aim of checking reliability of comparisons in case of overlapping error bars. Moreover, for each schemes compared, we perform *new* performance measurements with *new* metrics like normalised energy consumption, packet delivery fraction (PDF), average E2E delay, and average connection lifetime (ACL).

2 Prior related work

Although a few researchers have considered the problem of dynamic route optimisation in the context of reactive routing protocols in MANETs, the problem has not received widespread attention. In a recent paper, Sareen and Sharma (2014) provide comparative survey of some of this works. In this section, we present a comprehensive survey of related prior research on this issue. The approaches can be classified into two groups:

1 those based on *promiscuous mode*

2 *special-packet-based approaches*.

2.1 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. (2000), which considers several dynamic route optimisation 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 summarise their route optimisation 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 neighbourhood, 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. (2000), other studies (Giruka et al., 2004; Gui and Mohapatra, 2003; Yen et al., 2010) 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 (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 optimisation into the MAC layer header of the packets, and enable the nodes to retrieve information from these headers during MAC layer interactions, *without utilising promiscuous mode*. However, such a solution requires complex cross-layer design, as well as a modification of packet header formats.

Gui and Mohapatra (2003) 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. (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 work of Yen et al. (2010) contains a similar route optimisation mechanism to the SHORT (Gui and Mohapatra, 2003). Unfortunately, the authors do not provide any experimental results regarding the effectiveness of the proposed path optimisation mechanism.

As the most recent work in this group, Liang and Wakahara (2014) propose a selective route shortening method by implementing *support vector machine* (SVM) in order to predict the quality of shortening links, and classify potential shortening opportunities into either non-preferred or preferred. It is assumed in this work that all nodes operate in promiscuous mode, and moreover, they have the location information of both themselves and their neighbours.

2.2 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. (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 optimisation 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.

Bilgin and Khan (2010) introduce a similar scheme with the AODV version of the approach taken by Saito et al. in that they too consider shortening triplets (i.e., two-hop sub-paths) into direct links (i.e., one-hop) by eliminating the middle node. However, their work diverges from Saito et al.'s in several aspects:

1 Bilgin and Khan amortise tests against data traffic, rather than using proximity triggered approach

2 they have resolved a deficiency in their implementation that prevents the destination node from being able to activate the optimisation process

3 Bilgin and Khan's protocol operates E2E, rather than on triplets of nodes of interest.

This protocol is one of the two route optimisation schemes that we select for performance comparison in our own work here.

Kolar et al. (2005) introduce a different approach exploiting advantages of directional antennas for route compaction. The proposed approach aims to shorten paths by eliminating some of nodes on routes via using long range directional antennas instead of omnidirectional antennas.

Similarly, Zapata (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 neighbours 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[1].

Bilgin et al. (2010) put forward a dynamic route optimisation scheme which uses similar ideas with the Zapata's mechanism. However, Bilgin and Khan's protocol does not fall prey to the aforementioned pitfall in Zapata's algorithm. Moreover, Bilgin and Khan validates their protocol by experimental studies conducted using extensive packet level simulations with ns2. This is the second algorithm that we choose for performance comparison in our own article here.

Huang et al. (2011) present a route optimisation algorithm partly similar to Bilgin and Khan's (2010) work such that the source node of an active connection initiates the mechanism at probabilistic intervals by producing and broadcasting a special packet S-0 with TTL of 1. The nodes receiving this special packet check if they have a shorter route to defined destination in S-0 by comparing hop count distances. If there is such a node, then the route is redirected through this node regardless of it is on active route. However, this approach includes a risky as follows: it is likely that there may be nodes caching route information for a period of time towards a specific destination, but not valid in reality due to dynamic environment of MANETs. Redirecting an active route through such a misleading node could lead a dead end or cause loop.

Revathi and Rangaswamy (2012) propose a dynamic route shortening and repairing mechanism as extension of AODV. Their proposed mechanism shortens active routes by replacing certain parts of paths with shorter alternatives that are not on active route. Additionally, in case of a link failure, the mechanism employs local route repair process instead of global route reconstruction.

3 Mathematical model

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 (Perkins et al., 2003), 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 ℓ: $V \times \Re \to \Re^2$ be a function which describes the two-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

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

where

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

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

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 $d_A(p)$ 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

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

and

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

This is the quantity *stretch*(*A*) is what we seek to evaluate and minimise. 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 minimise 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}.

3.2 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:

1. if the received signal strength is lower than a predefined threshold level[2], then the link the data packet last traversed is labelled a *weak link*, and regarded as likely to break soon
2. 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).

4 Compared solutions

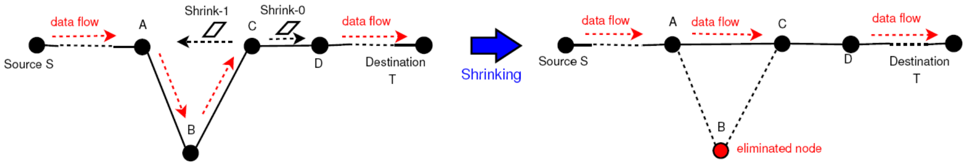
In this part, summary of the two compared route optimisation schemes are given. The first scheme is named *1-hop shrink* and the second one is named *multi-hop shrink*. Both the schemes are implemented as extension of AODV.

4.1 The 1-hop shrink scheme

The *1-hop* 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. To do this, the *1-hop 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).

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

Figure 2 The shrink mechanism (see online version for colours)



From the vantage point of the source node: the source node S generates (an additional) special packet of type shrink-0[3] after every certain amount of data packets (e.g., 4,8,16, or 32) has been sent. A shrink-0 packet contains the following fields:

- the IP address of the previous node in the connection
- 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 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). 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.

4.2 The multi-hop shrink scheme

The *multi-hop shrink* extends the previously described method of *1-hop shrink*, in two crucial ways:

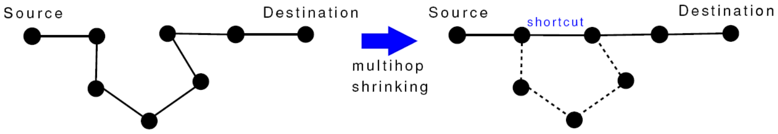
1 The new mechanism does not confine itself to successive *triplets* of nodes, thereby making it possible to make dramatic global reductions in path length.

2 The control traffic overhead incurred by the proposed multi-hop shrinking is approximately half of what was incurred by its predecessor, 1-hop shrinking.

Like 1-hop shrinking, the objective of the multi-hop shrink is to shorten unnecessarily long connections by eliminating inessential hops as illustrated in figure.

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. 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 multi-hop 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 3).

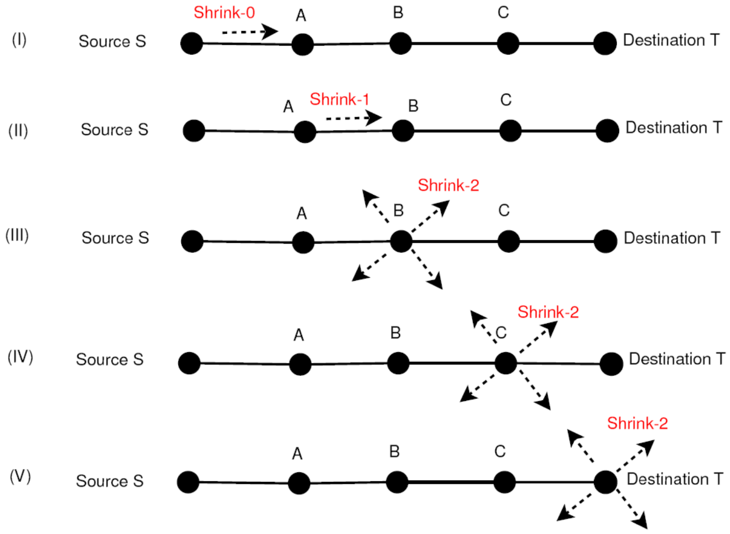
Figure 3 Main idea of *multi-hop shrink* (see online version for colours)



The multi-hop 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.

As the shrink packet travels downstream, nodes that see the special packet attempt to discover a shortcut to *upstream* nodes on the connection[4]. In summary, the *multi-hop shrink* mechanism achieves its goal of optimising connection topology by replacing inefficient multi-hop sub-connections with a direct 1-hop connection thereby eliminating unnecessary relay node(s). The protocol that achieves this will be described with the aid Figure 4.

Figure 4 How multi-hop shrinking works (see online version for colours)



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 recognise 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 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 recognises 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 recognises 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 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[5].

4.3 Determining periodicity of the schemes

The periodic sending of the special packets (i.e., shrink packets) both in *1-hop shrink* and *multi-hop shrink* schemes 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 (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[6]. In what follows, however, for simplicity of exposition, we will not consider this space optimisation. A natural question that arises concerns the effect of the choice of p on the performance of the optimisation schemes. This question has been investigated in Bilgin and Khan (2010) and Bilgin et al. (2010), where the authors have considered $p = 4$, 8, 16, 32. In general we will denote the shrinking mechanisms when $p = \alpha$ as *shrink-*α and *MShrink-*α respectively.

5 Performance analysis via simulation

The compared schemes have been implemented in the network simulator ns-2.

5.1 Experimental setup

The performance of the *1-hop shrink* and the *multi-hop shrink* schemes are compared against each other (with pure AODV as a reference) for the following network size, mobility models, and traffic/connection pattern as follows:

- *Networks:* networks comprised of 50 and 100 nodes are investigated. Node density was kept fixed at 50 nodes per 700 m × 700 m. Initial placement of the N nodes was taken to be uniform random in a square of side $700\sqrt{N/50}$ metres.
- *Traffic patterns:* the traffic connections are initiated between randomly chosen source-destination pairs chosen uniformly from the N nodes. The number of connections C is kept fixed at ten connections. Traffic sources generate 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 sub-section.

5.2 Mobility model

We would like to investigate the performance of the *1-hop shrink* and *multi-hop shrink* mechanisms under different mobility levels, including high mobility, low mobility, and static cases. To do this, we have modified the random waypoint mobility model, which is provided by *ns*2, as follows.

*ns*2 allows us to produce random movement scenarios based on the random way point mobility (RWP) model with pause times (Camp et al., 2002). 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 compared optimisation 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 compared 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 RWP 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 1,200 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.

5.3 Performance metrics

To compare the performance of the route optimisation schemes, we used the following metrics:

- *Normalised path length* (NPL) is measured by noting for each data packet delivered to its destination, both the number of hops that packet travelled, 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 compared schemes is to optimise path lengths.
- *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 optimisation process.
- *Normalised 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 compared algorithms, and therefore, it can be regarded as a measure of the cost of the schemes.
- *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 optimisation 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 E2E delay* – 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 Kolar et al. (2005) and Zhai and Fang (2006), dynamic route optimisation is expected to yield reduction in delay of packet delivery. This metric is the basis of verifying the extent to which the expectations of latency improvements are borne out in practice.
- *Normalised 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 ten 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.

5.4 Results and analysis

In this sub-section, we compare performance of the *1-hop shrink* and the *multi-hop shrink* schemes against each other (with pure AODV as a reference). As mentioned earlier, each scheme has four variants, i.e., SHRINK4, SHRINK8, SHRINK16, SHRINK32 for the *1-hop shrink*, and MSHRINK4, MSHRINK8, MSHRINK16, MSHRINK32 for the *multi-hop shrink*. SHRINK/MSHRINKα (where α is 4, 8, 16, and 32) means that an E2E shrink operation is initiated by source node after every α data packets are sent by the source node. Therefore, SHRINK4/MSHRINK4 and SHRINK32/MSHRINK32 represent the most and least frequent shrink operations, respectively, while SHRINK8/MSHRINK8 and SHRINK16/MSHRINK16 are intermediary between them. Rather than considering all 4 variants of each of the Shrink and MShrink, we report here on only representative version of each, namely SHRINK16 and MSHRINK16.

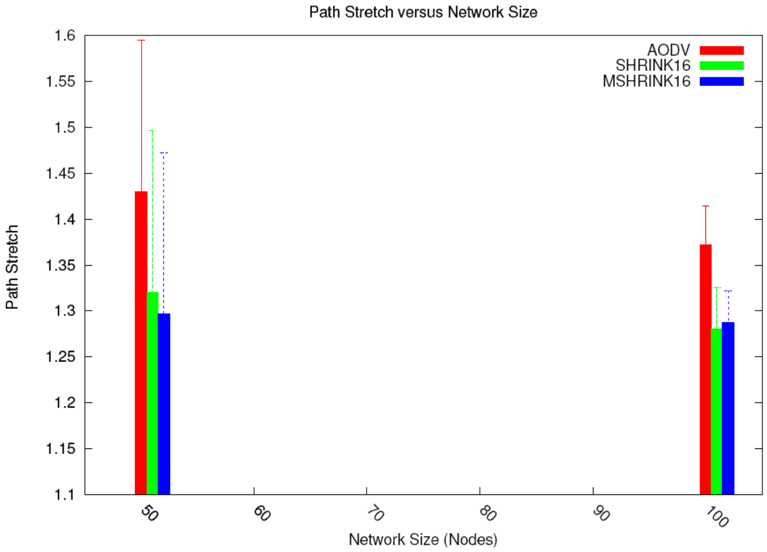
5.4.1 Normalised path length (NPL)

In this set of experiments, we measured the optimality of paths for SHRINK16, MSHRINK16, and pure AODV under different mobility level settings and network regimes. The results are the average of ten trials, each of which consists of about 50,000 data packets processed in networks including either 50 nodes or 100 nodes, with ten traffic connections. The outcomes are normalised 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.

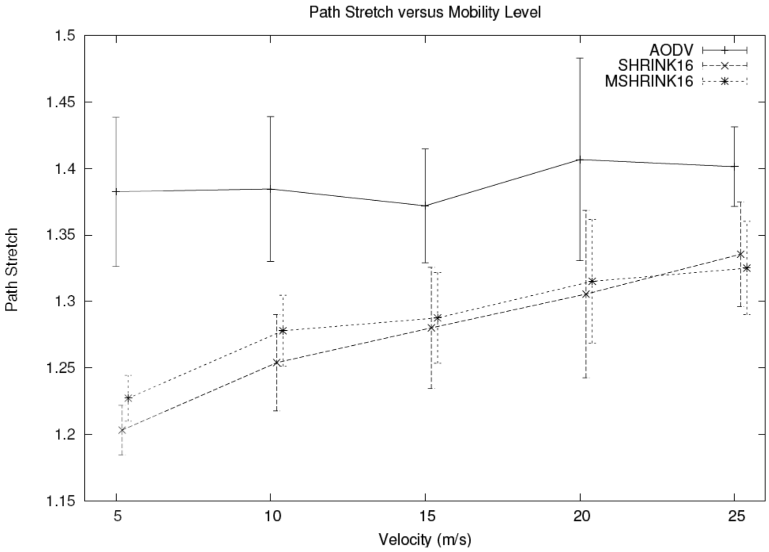
Figure 5(a) depicts path optimality as network size changes from 50 nodes to 100 nodes while maximum velocity is 15 m/s. At this point, we should notice that many error bars in the figure overlap each other, which means it may not be justifiable to compare the tested mechanisms based on mean values of 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 the other one (e.g., MSHRINK16), even though average value of the former (e.g., SHRINK16) is worse than average value of the latter (e.g., MSHRINK16). 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.

Figure 5 Normalised path length (NPL), (a) v=15, c=1 (b) n=100, c=10 (see online version for colours)



(a)



(b)

Table 1 includes correlation coefficients of trials underlying Figure 5. 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 1 Correlation coefficients for curves in Figure 5

vel =15 m/s	*50 nodes*			*100 nodes*		
	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 5(a), 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.

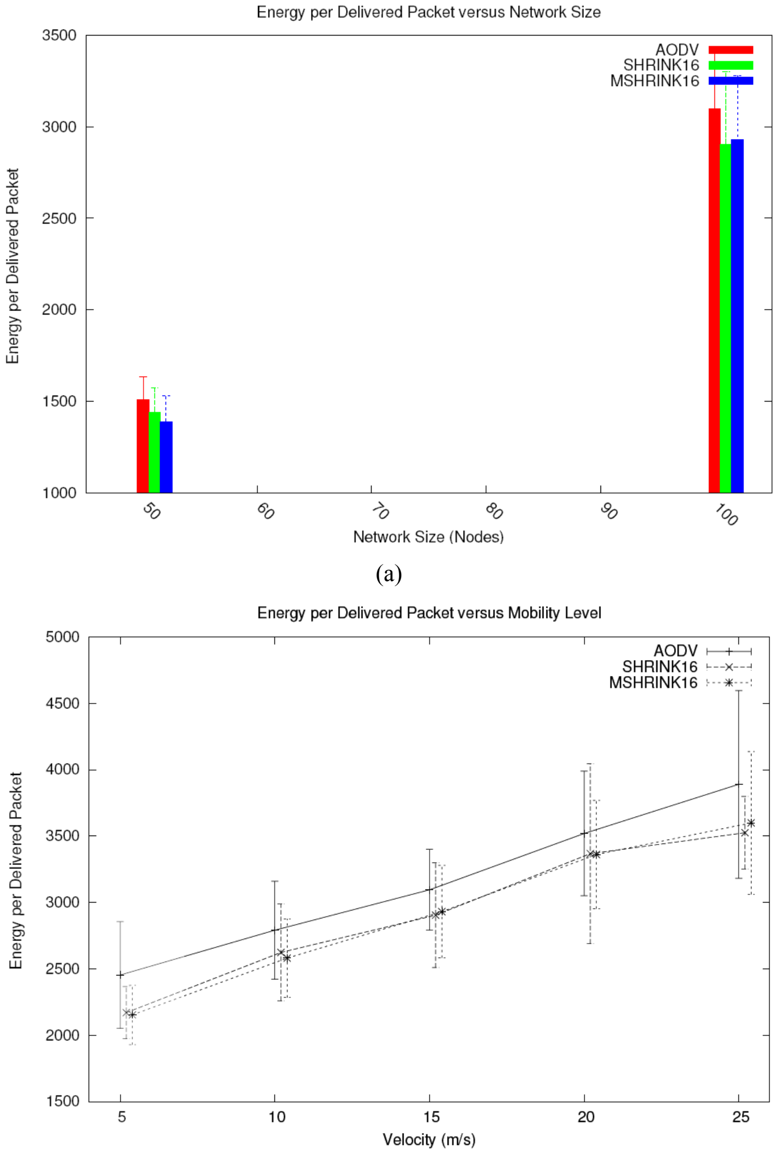
Figure 5(b) shows the NPL at different mobility levels, for a network of 100 nodes with ten traffic connections. According to the figure, both the proposed SHRINK16 and MSHRINK16 schemes have lower path stretch than pure AODV at all mobility levels. Nevertheless, the schemes approach the performance 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.

5.4.2 Normalised 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 6(a) depicts the performance of the schemes in networks having different sizes. According to the figure, pure AODV requires 1,508 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. 1,386 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 3,095 units, 2,901 units, and 2,930 units respectively.

Before interpreting these results, we need to determine whether we can use mean values of experimental trials by calculating corresponding correlation coefficients for the trials underlying the curves in the figure. Table 2 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.

Figure 6 Normalised energy consumption (NEN), (a) v = 15, c = 10 (b) n = 100, c = 10 (see online version for colours)



(a)

(b)

Table 2 Correlation coefficients for curves in Figure 6

vel = 15 m/s	*50 nodes*			*100 nodes*		
	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

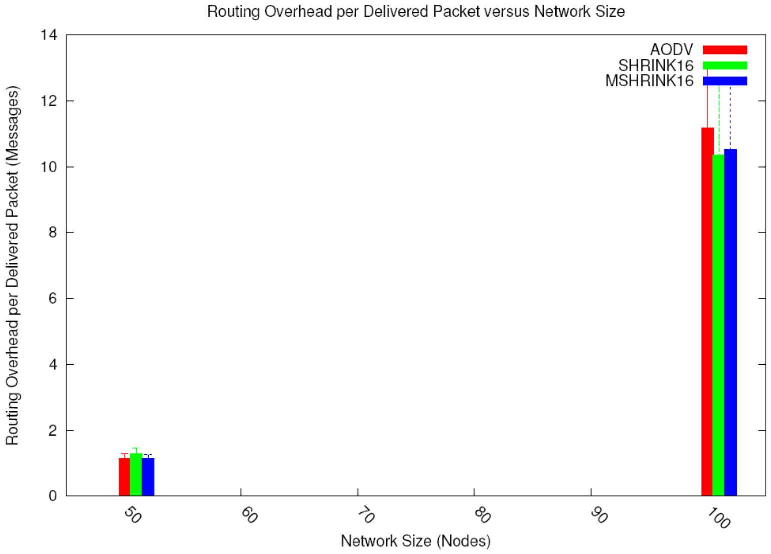
Since correlation is significant, it can be deduced from Figure 6(a) that both SHRINK16 and MSHRINK16 schemes reduce energy consumption in comparison to pure AODV. Besides this, MSHRINK16 and SHRINK16 are incomparable in 50-node and 100-node networks. Figure 6(b) depicts average energy consumption per 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.

5.4.3 Normalised routing load (NRL)

In this set of experiments, we compared the traffic load incurred by the proposed SHRINK16 and MSHRINK16 schemes. Figure 7(a) indicates overall routing load, normalised by the number of data packets delivered to destinations, for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 mechanisms in different network sizes.

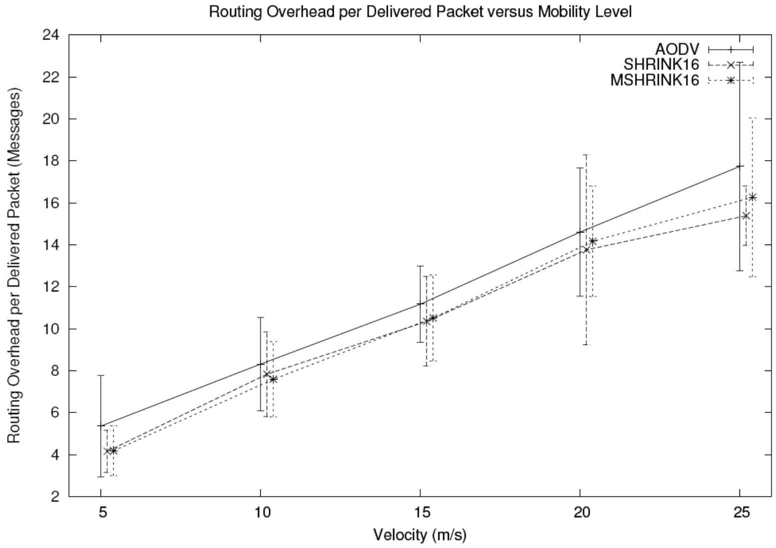
Table 3 contains the correlation coefficients for trials underlying the curves in Figure 7. It can be deduced from the tables that the performance of the schemes may be compared using the data from the larger 100-node network experiments, since correlation coefficients between SHRINK16 and MSHRINK16 are 0.967 for 100-node network experiments (but are only 0.760 for 50-node network). Figure 7(b) 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.

Figure 7 Normalised routing load (NRL), (a) v = 15, c = 10 (b) n = 100, c = 10 (see online version for colours)



(a)

Figure 7 Normalised routing load (NRL), (a) v = 15, c = 10 (b) n = 100, c = 10 (continued) (see online version for colours)



(b)

Table 3 Correlation coefficients for curves in Figure 7(a)

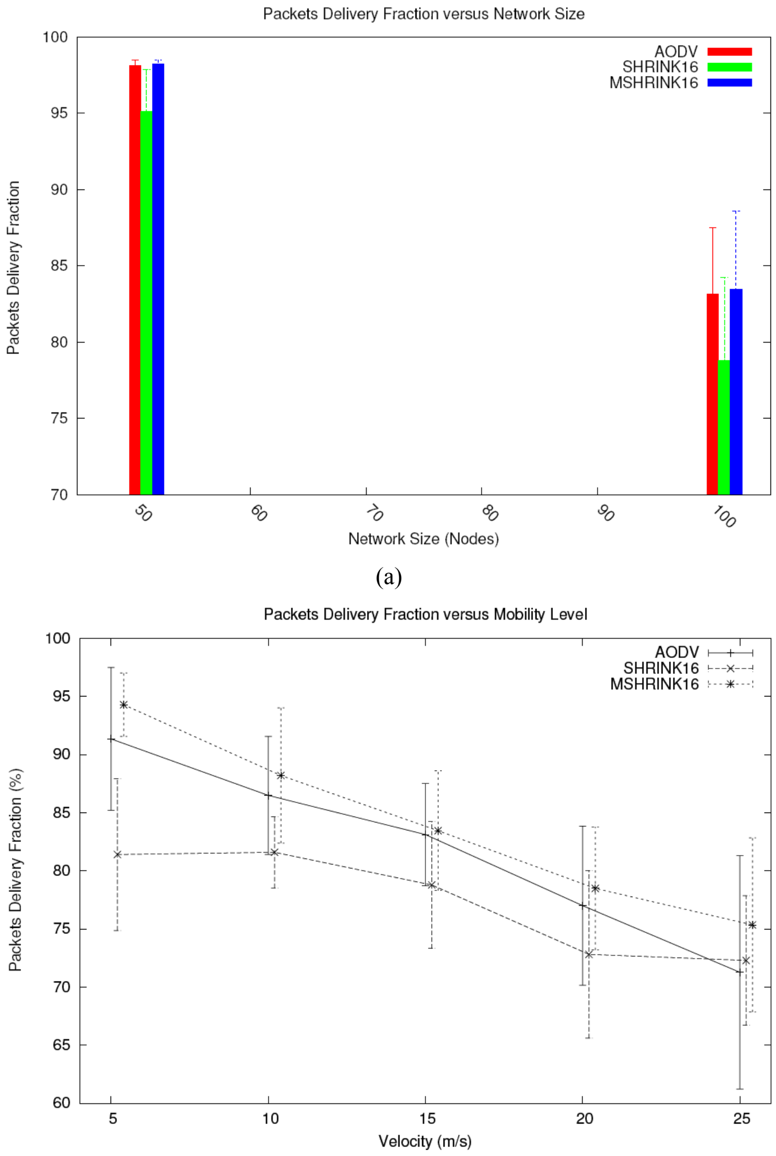
vel =15 m/s	*50 nodes*			*100 nodes*		
	AODV	*SHRINK16*	*MSHRINK16*	*AODV*	*SHRINK16*	*MSHRINK16*
AODV	1	0.705	0.783	1	0.765	0.838
SHRINK16	-	1	0.760	-	1	0.967
MSHRINK16	-	-	1	-	-	1

5.4.4 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 8(a) indicates PDF for different network sizes when maximum velocity is set at 15 m/s.

Table 4 contains correlation coefficients for trials corresponding to Figure 8. The table shows it is not justifiable to compare average performance of the schemes in 50-node networks. In contrast, the coefficients are close to 1 for large network experiments, e.g., the 100-node network. From Figure 8(b) and Table 4, 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.

Figure 8 PDF, (a) v = 15, c = 10 (b) n = 100, c = 10 (see online version for colours)



(a)

(b)

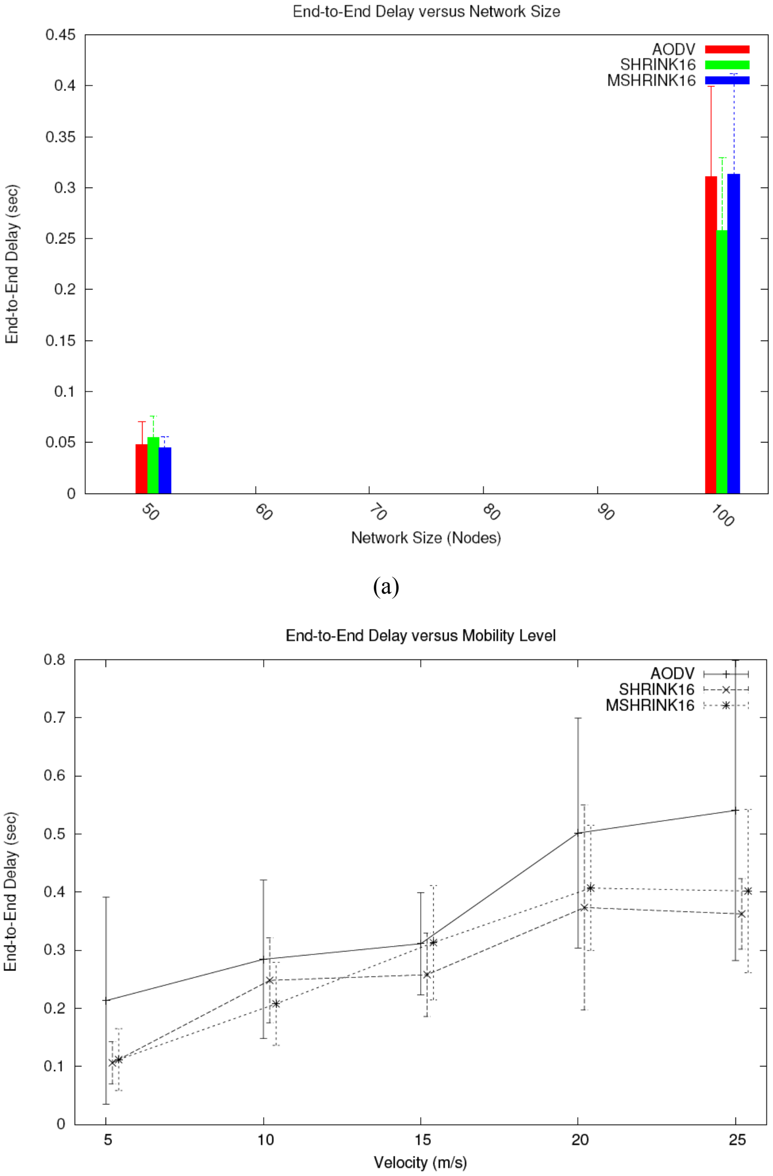
Table 4 Correlation coefficients for curves in Figure 8

vel =15 m/s	*50 nodes*			*100 nodes*		
	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

5.4.5 Average E2E delay

In this set of experiments, we measured average E2E delay in packet delivery to destinations. Figure 9(a) depicts average E2E delay for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 on different sized networks, based on ten trials, in which each trial consisted of approximately 50,000 data packets being delivered on ten different connections.

Figure 9 Average E2E delay, (a) v =15, c =10 (b) n =100, c = 10 (see online version for colours)



(a)

(b)

Table 5 includes correlation coefficients for Figure 9. According to the table, the correlation coefficient between SHRINK16 and MSHRINK16 is 0.463 in 50-node network, and is 0.709 in 100-node network, indicating it is difficult to justify comparing these schemes based on mean performance values.

Table 5 Correlation coefficients for curves in Figure 9

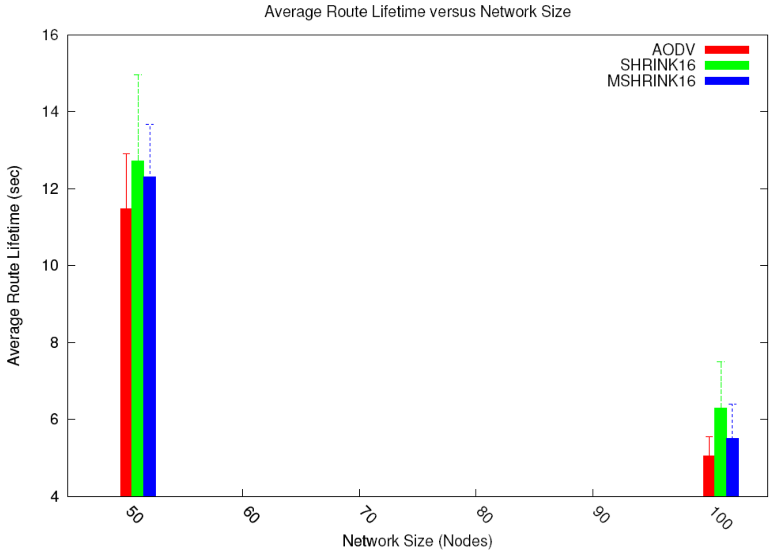
vel =15 m/s	*50 nodes*			*100 nodes*		
	AODV	*SHRINK16*	*MSHRINK16*	*AODV*	*SHRINK16*	*MSHRINK16*
AODV	1	0.919	0.501	1	0.827	0.343
SHRINK16	-	1	0.463	-	1	0.709
MSHRINK16	-	-	1	-	-	1

5.4.6 ACL

In this set of experiments, we measured the average lifetime of active connections. Figure 10 shows ACL for pure AODV, AODV+SHRINK16, and AODV+MSHRINK16 in different network sizes. Table 6 shows corresponding correlation coefficients for trials underlying the curves in Figure 10. 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.

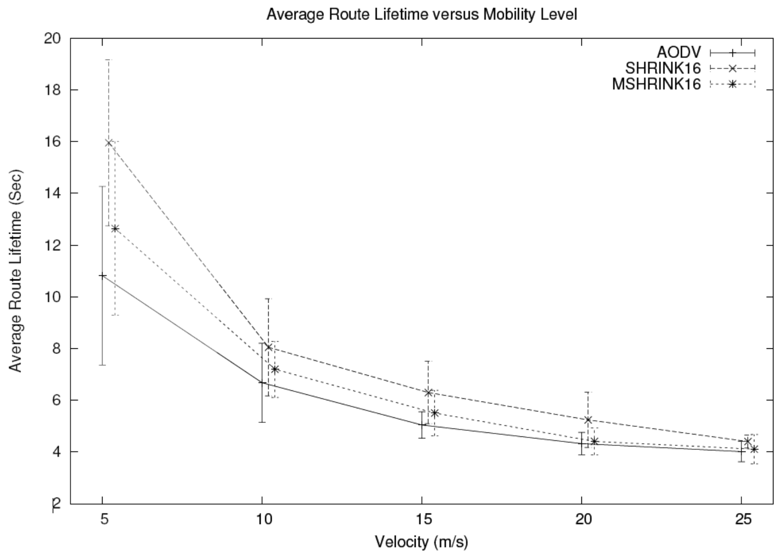
According to Figure 10(a), applying SHRINK16 yields longer-lived routes in comparison with pure AODV and AODV+MSHRINK16. This can also be seen from Figure 10(b), which shows the ACL at different mobility levels for the larger 100-node network experiments.

Figure 10 ACL, (a) v = 15, c = 10 (b) n = 100, c = 10 (see online version for colours)



(a)

Figure 10 ACL, (a) v = 15, c = 10 (b) n = 100, c = 10 (continued) (see online version for colours)



(b)

Table 6 Correlation coefficients for curves in Figure 10

vel =15 m/s	*50 nodes*			*100 nodes*		
	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

6 Conclusions

In this paper, we evaluated and compared two different route optimisation schemes called *1-hop shrink* and *multi-hop 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 optimise active connections in terms of route length by eliminating inessential hops on paths. The compared 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 optimisation), 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 multi-hop shrink uses broadcast packets
- *Triplet nodes vs. E2E:* 1-hop shrink is able to detect shortcuts within triplet of nodes, whereas multi-hop shrink can detect shortcuts between any pair of nodes on an active connection.

In comparing the 1-hop shrink and multi-hop 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 multi-hop shrink scheme outperformed the 1-hop shrink scheme in terms of PDF. The experiments also indicated that 1-hop shrink scheme outperformed multi-hop shrink slightly, in both E2E delay and ACL.

References

Bilgin, Z. and Khan, B. (2010) 'A dynamic route optimization mechanism for aodv in manets', *2010 IEEE International Conference on Communications (ICC)*, pp.1–5.

Bilgin, Z., Khan, B. and Al-Fuqaha, A. (2010) 'Only the short die old: route optimization in manets by dynamic subconnection shrinking', *Proceedings of the 6th International Wireless Communications and Mobile Computing Conference, IWCMC'10*, ACM, New York, NY, USA, pp.681–685.

Camp, T., Boleng, J. and Davies, V. (2002) '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*, Vol. 2, No. 5, pp.483–502.

Cheng, Z. and Heinzelman, W.B. (2004) 'Exploring long lifetime routing (LLR) in ad hoc networks', *MSWiM '04: Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ACM, New York, NY, USA, pp.203–210.

Giruka, V.C., Singhal, M. and Yarravarapu, S.P. (2004) 'A path compression technique for on-demand ad-hoc routing protocols', *Mobile Ad-Hoc and Sensor Systems (MASS).*

Gui, C. and Mohapatra, P. (2003) 'SHORT: self-healing and optimizing routing techniques for mobile ad hoc networks', *MobiHoc '03: Proceedings of the 4th ACM international symposium on Mobile Ad Hoc Networking & Computing*, ACM, New York, NY, USA, pp.279–290.

Huang, L., Wang, F., Yan, G. and Duan, W. (2011) 'An efficient dynamic route optimization algorithm for mobile ad hoc networks', *Procedia Environmental Sciences*, Vol. 11, Part A, pp.518–524, *2nd International Conference on Challenges in Environmental Science and Computer Engineering (CESCE).*

Kolar, V., Rogers, P. and Abu-Ghazaleh, N. (2005) 'Route compaction for directional route discovery in manets', *IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, 2005. (WiMob'2005)*, Vol. 3, pp.101–108.

Lenders, V., Wagner, J. and May, M. (2006) 'Analyzing the impact of mobility in ad hoc networks', *REALMAN '06: Proceedings of the 2nd International Workshop on Multi-Hop Ad Hoc Networks: From Theory to Reality*, ACM, New York, NY, USA, pp.39–46.

Li, J., Blake, C., De Couto, D.S., Lee, H.I. and Morris, R. (2001) 'Capacity of ad hoc wireless networks', *Proceedings of the 7th Annual International Conference on Mobile Computing and Networking, MobiCom '01*, ACM, New York, NY, USA, pp.61–69.

Li, X. (2009) 'Hop optimization and relay node selection in multi-hop wireless ad-hoc networks', in Vicat-Blanc Primet, P., Kudoh, T. and Mambretti, J. (Eds.): *Networks for Grid Applications*, Vol. 2, *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Eng.*, pp.161–174, Springer Berlin Heidelberg.

Liang, Z. and Wakahara, Y. (2014) 'Automatic route shortening based on link quality classification in ad hoc networks', in Swiatek, J., Grzech, A., Swiatek, P. and Tomczak, J.M. (Eds.): *Advances in Systems Science*, Vol. 240, *Advances in Intelligent Systems and Computing*, pp.345–354, Springer International Publishing.

Lim, G. (2002) 'Link stability and route lifetime in ad-hoc wireless networks', *ICPPW '02: Proceedings of the 2002 International Conference on Parallel Processing Workshops*, IEEE Computer Society, Washington, DC, USA.

Mamoun, M.H. (2011) 'A proposed route selection technique in dsr routing protocol for manet', *International Journal of Engineering & Technology*, Vol. 11, No. 2, pp.10–13.

Misra, S. and Rajesh, G. (2011) 'Bird flight-inspired routing protocol for mobile ad hoc networks', *ACM Trans. Auton. Adapt. Syst.*, Vol. 6, No. 4, pp.25:1–25:37.

Misra, S., Dhurandher, S.K., Obaidat, M.S., Gupta, P., Verma, K. and Narula, P. (2010) 'An ant swarm-inspired energy-aware routing protocol for wireless ad-hoc networks', *Journal of Systems and Software*, Vol. 83, No. 11, pp.2188–2199, Interplay between Usability Evaluation and Software Development.

Misra, S., Ghosh, T.I. and Obaidat, M.S. (2014) 'Routing bandwidth guaranteed paths for traffic engineering in wimax mesh networks', *International Journal of Communication Systems*, Vol. 27, No. 11, pp.2964–2984.

Misra, S., Krishna, P., Bhiwal, A., Chawla, A., Wolfinger, B. and Lee, C. (2012) 'A learning automata-based fault-tolerant routing algorithm for mobile ad hoc networks', *The Journal of Supercomputing*, Vol. 62, No. 1, pp.4–23.

Perkins, C., Belding-Royer, E. and Das, S. (2003) *Ad Hoc on-Demand Distance Vector (AODV) Routing*, RFC 3561, RFC [online] http://www.rfc-editor.org/rfc/rfc3561.txt (accessed 18 April 2015).

Rappaport, T. (2001) *Wireless Communications: Principles and Practice*, Prentice Hall, PTR, Upper Saddle River, NJ, USA.

Raw, R.S., Toor, V. and Singh, N. (2012) 'Estimation and analysis of path duration in vehicular ad hoc networks using position-based routing protocol', *IJCA Special Issue on Issues and Challenges in Networking, Intelligence and Computing Technologies*, Vol. ICNICT, No. 3, pp.34–39.

Revathi, S. and Rangaswamy, T. (2012) 'Dynamic route shortening and route repairing mechanism for mobile ad hoc networks', *Journal of Computer Science*, Vol. 8, No. 8, pp.1212–1218.

Sadagopan, N., Bai, F., Krishnamachari, B. and Helmy, A. (2003) 'PATHS: analysis of path duration statistics and their impact on reactive manet routing protocols', *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking &Amp; Computing, MobiHoc '03*, ACM, New York, NY, USA, pp.245–256.

Saito, M., Aida, H., Tobe, Y. and Tokuda, H. (2004) 'A proximity-based dynamic path shortening scheme for ubiquitous ad hoc networks', *Proceedings of IEEE International Conference on Distributed Computing Systems (ICDCS04)*.

Sareen, K. and Sharma, D. (2014) 'Survey on path optimization in ad hoc networks', *International Journal of Information and Computation Technology*, Vol. 4, No. 1, pp.91–98.

Tseng, Y-C., Li, Y-F. and Chang, Y-C. (2003) 'On route lifetime in multihop mobile ad hoc networks', *IEEE Transactions on Mobile Computing*, Vol. 2, No. 4, pp.366–376.

Wu, S., Ni, S., Tseng, Y. and Sheu, J. (2000) 'Route maintenance in a wireless mobile ad hoc network', *33rd Hawaii International Conference on System Sciences*, Maui.

Wu, X., Sadjadpour, H.R. and Garcia-Luna-Aceves, J.J. (2007) 'From link dynamics to path lifetime and packet-length optimization in MANETs', *Wireless Networks*, Vol. 15, No. 5, pp.637–650.

Yen, Y-S., Chang, H-C., Chang, R-S. and Chao, H-C. (2010) 'Routing with adaptive path and limited flooding for mobile ad hoc networks', *Computers and Electrical Engineering*, Vol. 36, No. 2, pp.280–290, Wireless ad hoc, Sensor and Mesh Networks.

Zapata, M.G. (2005) 'Shortcut detection and route repair in ad hoc networks', *PERCOMW '05: Proceedings of the Third IEEE International Conference on Pervasive Computing and Communications Workshops*, IEEE Computer Society, Washington, DC, USA, pp.237–242.

Zhai, H. and Fang, Y. (2006) 'Impact of routing metrics on path capacity in multirate and multihop wireless ad hoc networks', *Proceedings of the 2006 IEEE International Conference on Network Protocols, ICNP '06*, IEEE Computer Society, Washington, DC, USA, pp.86–95.

Notes

[1] 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.

[2] 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 250 m, then threshold level corresponds 225 m, when we consider the two-ray ground reflection model (Rappaport, 2001) as implemented in ns2.

[3] Here the 0 represents the status of a flag in the packet header.

[4] 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 A5□ can be made (see Section 3.2), the shrinking operation is initiated by the third node within the connection, rather than at the source. In settings where assumption A5* cannot be made, identifying the third node may not be feasible: the multi-hop shrink scheme still operates correctly, but is slightly more wasteful in terms of control traffic overhead.

[5] 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 increases the time required to process each data packet, since random number generation takes time.

[6] Admittedly, in this space optimization, the operation of the random number generator consumes time.