A Unicast Stable Path Routing Protocol for Mobile Ad hoc Networks based on the Inverse of Link Expiration Time

Natarajan Meghanathan
Dept. of Computer Science, Jackson State University
1400 John R. Lynch Street, Jackson, MS 39217, USA
Tel: 1-601-979-3661 E-mail: natarajan.meghanathan@jsums.edu

Received: October 15, 2011 Accepted: November 30, 2011 Published: December 18, 2011
DOI: 10.5296/npa.v3i3.1018 URL: http://dx.doi.org/10.5296/ npa.v3i3.1018

Abstract

The high-level contribution of this paper is the design and development of a unicast stable path routing protocol (referred to as SILET) that determines long-living routes based on the Inverse of the Link Expiration Times (LET) for Mobile Ad hoc Networks (MANETs). From a graph theoretical perspective, we model SILET as a minimization problem to discover stable routes with the lowest value for the sum of the link weights; the weight of a link defined as 1 plus the inverse of the LETs. The inclusion of '1' in the link weights helps to reduce the number of constituent links of the paths (i.e. the hop count). Performance comparison of SILET with FORP, DSR and ABR routing protocols in ns-2 illustrate that SILET discovers routes whose lifetimes are significantly longer compared to that of DSR and ABR and at most 20-40% lower than that of FORP; the hop counts of SILET routes are at most 4% larger than the minimum hop count; and the end-to-end delay per data packet for SILET routes is the lowest among all the routing protocols simulated – an indication that SILET effectively minimizes the stability-hop count tradeoff observed with the currently available MANET unicast routing protocols.

Keywords: Link Expiration Time, Mobile Ad hoc Networks, Path Lifetime, Stability-Hop Count Tradeoff, Unicast Routing Protocol

1. Introduction

A Mobile Ad hoc Network (MANET) is a dynamic distributed system of wireless nodes moving independently of each other. A MANET operates with several constraints [1] – the wireless nodes are battery charged, the bandwidth is limited and the wireless medium is prone to interference and collisions. Because of all these constraints, wireless nodes often operate with a limited transmission range and two nodes can communicate directly if and only if they lie within the transmission range of each other. Nodes that are outside the transmission range of each other have to route their packets through one or more intermediate nodes; thus, multi-hop communication is more common in MANETs. As the network topology changes dynamically with time due to node mobility, the route between a pair of nodes, hereafter referred to as the source-destination (*s-d*) route, do not often exist for the entire communication session. Routes have to be frequently reconfigured.

Based on the strategy adopted to determine and maintain routes, MANET routing protocols can be categorized as proactive or reactive routing protocols [2]. Proactive routing protocols determine routes between every pair of nodes, irrespective of the requirement, through network-wide broadcast of table updates [3] or link state updates [4]. Reactive (or on-demand) routing protocols determine routes between an *s-d* pair only when required (i.e., when the source node *s* has data to send to the destination node *d* and is not aware of a route to *d*). In the presence of a dynamically changing network topology, if the number of *s-d* pairs is not significantly high, reactive routing is a preferred strategy as it incurs relatively less control overhead [5][6]. On the other hand, if any two nodes can become an *s-d* pair and communicate, reactive routing may involve frequent flooding of the network for on-demand route discovery. Nevertheless, a majority and most of the recent MANET routing protocols proposed in the literature are on-demand in nature. We will also focus on on-demand MANET routing for the rest of this paper.

Reactive routing protocols discover routes through a global broadcast query-reply cycle according to which Route Request (RREQ) control messages will be flooded through the entire network, with each node broadcasting a RREQ message exactly once to all of its neighbors. The destination node receives the RREQ message through one or more paths, applies the route selection principles of the particular routing protocol and then sends a Route Reply (RREP) message on the chosen route. Reactive routing protocols have been proposed based on different route selection principles [2] – the primary two among these being minimum-hop based and stability-based. The minimum-hop based routing protocols and their variants also yield lower energy consumption [7]. The stability-based routing protocols attempt to minimize the number of route changes and determine routes that exist for a longer lifetime. Stable routes are desired for Quality-of-Service (QoS) in communication so that packets can be routed incessantly through paths that offer the required quality [8]. Frequent route changes introduce jitter and out-of-order packet delivery at the destination. Also, flooding-based route discoveries are expensive in terms of energy consumption and bandwidth usage. Thus, like hop count, the stability of routes is very much a desired characteristic for MANET routing protocols.

In this research, we re-visit stability-based unicast routing protocols and explore the possibility of determining stable routes that can exist for a longer time and do not require frequent reconfiguration. In this pursuit, we examine the possibility of using the 'Predicted Link Expiration Time (LET)' concept of [9] that has been actually employed for stable unicast routing in the Flow-Oriented Routing Protocol (FORP) [10]. The FORP protocol has been observed to yield stable routes that have even twice the lifetime of the minimum-hop routes [7] discovered using the well-known Dynamic Source Routing (DSR) protocol [24]. However, the tradeoff is that the hop count of FORP routes is significantly larger (twice or more) than the minimum hop count [7]. A close analysis behind this stability-hop count tradeoff from a graph theoretic perspective reveals that FORP has been formulated as a Max-Min problem wherein the weight of a link is the predicted LET; the weight of a path (referred to as the Route Expiration Time, RET) is the minimum of the LET of the constituent links and FORP tries to choose the path with the maximum RET. Max-Min based routing on a weighted network graph (in the case of FORP, the LETs are the edge weights) leads to a maximum spanning tree [11] and the hop count of paths between any two nodes on a maximum spanning tree is bound to be quite larger than the minimum hop count. Section 3 explains more about the inherent flaw in the design of the FORP protocol.

In this paper, we explore the use of a minimization-based optimization function that can yield us stable source-destination (*s-d*) paths and at the same time does not significantly increase the hop count of the *s-d* paths. In this pursuit, we propose using the inverse of the LET values as part of the link weights. The actual value for a link weight is 1 plus the inverse of the LET of the link. The weight of a path is the sum of all the link weights. Since we are using the inverse of the LET as part of the link weights, the path with the minimum weight is likely to be more stable. The reason to include a '1' to the link weight is to account for the number of links on the *s-d* paths; when the destination node chooses the path with the minimum weight, the hop count of the chosen stable path is also more likely to be close to the minimum. The above explanation forms the hypothesis of our research in this paper. We call our proposed protocol as the Stable routing protocol based on Inverse of the Link Expiration Times, with the acronym SILET. The twin objectives of SILET are to determine stable source-destination paths with minimum hop count. Through extensive simulation analysis (conducted in ns-2 [12]), we prove that our hypothesis is valid and the twin objectives of SILET are accomplished. We also observe that SILET effectively minimizes the stability-hop count tradeoff and this leads to a relatively lower end-to-end delay per data packet.

The rest of the paper is organized as follows: Section 2 describes related work on stable path routing. Section 3 presents the objective, motivation and the optimization function of the proposed SILET routing protocol. Section 4 presents the sequence of steps illustrating the route discovery, acquisition and maintenance procedures of the SILET protocol. Section 5 reviews the routing protocols that have been compared with SILET in this paper. In addition to SILET, we simulate and compare the performance of (i) Dynamic Source Routing protocol, DSR [24]; (ii) Associativity-Based Routing protocol, ABR [25]; and (iii) Flow-Oriented Routing Protocol, FORP [10]. Section 6 presents the simulation environment and the simulation results – comparing the routing protocols with respect to three metrics: path lifetime, hop count and end-to-end delay per data packet. Section 7 discusses the advantages with the multi-objective optimization of SILET and the scenarios where the protocol can be applied. Section 8 draws conclusions and presents future work. Throughout the paper, the terms ‘path’ and ‘route’, ‘link’ and ‘edge’, ‘packet’ and ‘message’ are used interchangeably. They mean the same.

2. Related Work

Research works [13][14][15] propose a Min-Max approach based routing for communication in MANETs. In [13], the authors take into account the predicted link expiration time (LET) and the residual battery charge of the nodes while computing stable and reliable routes that exist for a longer time without link failures (due to node mobility) and node failures (due to battery power exhaustion). Accordingly, the predicted lifetime of a node is the product of its residual battery charge and the sum of the LETs of all links adjacent to the node. The predicted lifetime of a route is then the minimum of the predicted lifetime of the intermediate nodes on the route and a route with a larger predicted lifetime is naturally preferred for stable and reliable routing. In [14], the authors propose a signal strength-estimate driven stable path routing protocol wherein the estimated signal strength of the Route Request (RREQ) packets is recorded in the RREQ packets itself at each forwarding node. The estimated signal strength of a path is the minimum of the estimated signal strength of the constituent links on the path as included by the forwarding nodes. The destination chooses the path with the largest estimated signal strength and sends back a Route Reply (RREP) packet on the chosen path. In [15], the velocity of the intermediate nodes is considered as the routing criteria to form a stable path. The bottleneck velocity of a path is the maximum velocity of any of the intermediate nodes on the path. The path with the minimum bottleneck velocity is chosen by a destination node to connect to the source node. Again, either a Min-Max or a Max-Min approach based route discovery only leads to minimum or maximum spanning trees respectively and the hop count per source-destination path on such trees could be significantly larger.

Research works [16][17] characterize the stability of a link using the strength of the beacon signals periodically exchanged in a neighborhood. The Signal Stability-based Adaptive (SSA) unicast routing protocol [16] characterizes the links into two classes: strong and weak links. Nodes are required to periodically exchange beacons in the neighborhood. The network operates with two thresholds for signal strength: threshold for strong links P_{th}^{strong} and threshold for signal reception P_{th}^{rec}, with $P_{th}^{strong} > P_{th}^{rec}$. If the strength of beacon signal received from a neighbor node exceeds P_{th}^{strong}, then a node categorizes the link with the neighbor node as a strong link. If the strength of the beacon signal is below P_{th}^{strong}; but above P_{th}^{rec}, the link is characterized as a weak link. During a broadcast route discovery process, the source node attempts to determine a stable route to the destination only comprising of the strong links. If a route cannot be determined based on strong links, another broadcast route discovery process is launched to determine routes considering all the links in the network. In networks of low and moderate density, there will be several instances of failure to find a stable route comprising only of strong links and a second broadcast route discovery has to be launched to find a route involving both strong and weak links. This will add considerable control overhead to the network.

The Route-lifetime Assessment Based Routing (RABR) unicast protocol [17] works by computing a metric called "link affinity" for each link based on the average change in the signal strength of the beacons received within a time window during the recent past. If the average change in the signal strength is positive, then the nodes are assumed to be approaching each other and the affinity of the link is assigned to a high value (theoretically, ∞). If the average change in the signal strength is negative, then the affinity value of the link is the ratio computed as the difference between the minimum threshold for the signal strength required for a link between two nodes to exist and the signal strength of the most recently received beacon divided by the average change in the signal strength. During on-demand route discovery, the affinity values of the links get propagated in the route search messages and the destination prefers a path with the largest affinity. The affinity value for a path is the minimum of the affinity values of its constituent links. An adaptation of RABR for zone-based routing has been recently proposed as the ZLERP (Zone and Link Expiry based Routing) protocol [18] for MANETs.

3. SILET Routing Protocol – Objective, Motivation and Optimization Function

3.1 Objective and Assumptions

The objective of the proposed unicast stable path routing protocol based on the inverse of the predicted link expiration times (SILET) is to determine long-living source-destination routes with hop count close to that of the minimum, thereby effectively neutralizing the stability-hop count tradeoff. The key assumptions behind the design and working of SILET are as follows:

- The network is homogeneous in nature and that all nodes operate with an identical and fixed transmission range, denoted as R in equation (1).
- Nodes periodically exchange beacons in the neighborhood. The beacon message broadcast by a node includes the current location of the node, the velocity at which the node is moving and the direction of movement of the node (denoted as the angle subscribed with respect to the positive X-axis).
- A node can determine the identification (ID) of the neighbor node that sends a beacon or protocol message through the underlying Medium Access Control (MAC) protocol and resolve the learnt hardware address to a network level logical address (say the IP address) using the Reverse Address Resolution Protocol (RARP) [19].
- A node obtains its location information through Global Positioning Scheme (GPS) [20] or any other relevant location service schemes (e.g. [21]).

3.2 Network Model and Optimization Function for SILET

The network model used in this research is described as follows:

- We adopt the unit-disk graph model [22] according to which there exists a link between two nodes *i* and *j* if and only if the distance between the two nodes is less than or equal to the fixed transmission range, *R*.

- The predicted link expiration time (LET) of a link $i - j$ between two nodes *i* and *j*, currently at (X_i, Y_i) and (X_j, Y_j), and moving with velocities v_i and v_j in directions θ_i and θ_j (with respect to the positive X-axis) is computed using the formula (equation 1) proposed in [9]:

$$LET(i,j) = \frac{-(ab+cd)+\sqrt{(a^2+c^2)R^2-(ad-bc)^2}}{a^2+c^2} \tag{1}$$

where $a = v_i*\cos\theta_i - v_j*\cos\theta_j$; $b = X_i - X_j$; $c = v_i*\sin\theta_i - v_j*\sin\theta_j$; $d = Y_i - Y_j$

- At any moment, every node maintains a LET-table comprising of the estimates of the LET values to each of its neighbor nodes based on the latest beacons received from the neighbor node.

- The weight of a link (edge) between nodes *i* and *j* is given by equation (2) below:

$$Weight(i,j) = 1 + \frac{1}{LET(i,j)} \tag{2}$$

The weight of a path *p* between vertices (nodes) v_0 and v_k, going through a sequence of intermediate nodes $v_1, v_2, \ldots, v_{k-1}$ is the sum of the weights of the constituent edges (v_0, v_1), (v_1, v_2), …, (v_{k-1}, v_k) of the path, denoted formally as:

$$Weight(p = v_0, v_1, \ldots, v_k) = \sum_{i=0}^{i=k-1} Weight(v_i, v_{i+1}) = \sum_{i=0}^{i=k-1}\left[1 + \frac{1}{LET(v_i, v_{i+1})}\right] \tag{3}$$

If **P** is the set of paths from which the optimal source-destination path is to be chosen from, the destination node chooses the path that has the minimum weight where the weight of each path in **P** is computed according to equation (3). More formally, the optimization function F_{opt} of SILET can be denoted as shown in equation (4) below:

$$F_{opt}^{SILET} = \underset{\forall p \in P}{Min}\left[Weight(p = v_0, v_1, \ldots, v_k)\right] = \underset{\forall p(=v_0, v_1, \ldots, v_k) \in P}{Min}\left[\sum_{i=0}^{i=k-1}\left[1 + \frac{1}{LET(v_i, v_{i+1})}\right]\right] \tag{4}$$

3.3 Analysis of the Optimization Function for FORP

The LET equation shown in (1) has been earlier used in the Flow-Oriented Routing Protocol (FORP) to determine stable unicast paths [10]. The FORP protocol has also been observed to yield the sequence of longest-living paths among contemporary unicast stable path routing protocols for MANETs [7][23]. FORP has been observed to determine stable routes whose lifetime is even twice the lifetime of the routes determined by the minimum-hop based Dynamic Source Routing (DSR) protocol [24]. However, the tradeoff observed in these performance comparison studies is that FORP incurs a significantly larger hop count, almost twice (or even more) the minimum hop count. When we analyze the reason behind this stability-hop count tradeoff of FORP, we observe the following flaw in its design and optimization function:

As part of the route discovery procedure of FORP, once the destination receives several Route Request (RREQ) packets containing the LET of the links traversed through, the destination chooses the path that has the maximum Route Expiration Time (RET). The RET of a path is the minimum of the LETs of the constituent links of the path. Using the same symbols/ notations employed for the optimization function for SILET as in equations (3) and (4), the optimization function of FORP can be formally represented as shown in equation (5):

$$RET(p = v_0, v_1, ..., v_k) = \underset{i=0}{\overset{k-1}{Min}}\left[LET(v_i, v_{i+1})\right] \qquad (5)$$

$$F_{opt}^{FORP} = \underset{\forall p \in P}{Max}\left[RET(p)\right] = \underset{\forall p(=v_0, v_1, ..., v_k) \in P}{Max}\left[\underset{i=0}{\overset{k-1}{Min}}\left[LET(v_i, v_{i+1})\right]\right] \qquad (6)$$

As observed in equation (6), the optimization function of FORP is basically a Max-Min function and if this function is employed on a network graph model with the LETs representing the edge weights, it would lead to the determination of a maximum spanning tree. The source-destination paths embedded in the maximum spanning tree of LETs would have a longer lifetime; but, the hop count of such paths would be significantly longer. It is evident from Graph Theory [11], that the minimum or maximum spanning trees (determined using algorithms like the Prim's or Kruskal's algorithms) on a weighted network graph are not the same as a minimum-weight tree (determined using algorithms like the Dijkstra's algorithm or the Bellman-Ford algorithm). In order to minimize the hop count of source-destination paths, what we need to employ is a minimization-based optimization function and not a Max-Min function.

The above analysis behind the stability-hop count tradeoff of FORP lead us to design an optimization function that would still give us long-living source-destination paths and at the same time minimize the hop count. We cannot directly use the LETs as the edge weights for our optimization as it would only lead to a maximization-based optimization function and we are likely to still incur larger hop count paths. So, we decided to use the inverse of the LETs as part of the edge weights. We then thought that just an optimization function that minimizes the sum of the inverse of the LETs of the constituent edges will not necessarily minimize the hop count of the paths. Hence, to account for the number of edges that constitute the path, we decided to add a '1' to the weight of an edge. Thus, the weight of an edge is 1 plus the inverse of the LET of the edge, as shown in equation (2). The weight of a path, shown in equation (3), is the sum of the weights of the constituent edges of the path. SILET discovers source-destination paths that have the lowest path weight – such paths have been observed in our simulations to have a longer lifetime, the lowest end-to-end delay per data packet and hop count close to that of the minimum.

4. Design of the SILET Routing Protocol

4.1 Propagation of the Route Request (RREQ) Message

When a source node has data to send to a destination node and is not aware of any next hop node to send to or has just purged the route entry for the destination node because of receiving a Route Error Message (described more in Section 4.4), the source node initiates a request-reply flooding cycle to determine a route to the destination node. The source node constructs a Route Request (RREQ) message and broadcasts the message to its neighbor nodes. The structure of the RREQ message is shown in Figure 1.

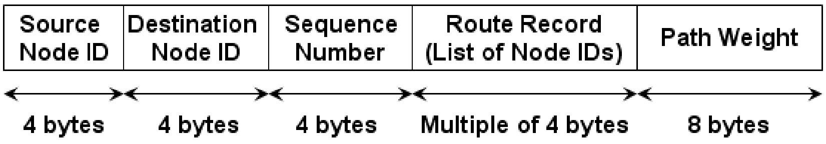


Figure 1: Structure of the Route Request (RREQ) Message

The RREQ message includes the ‘Source Node ID’, ‘Destination Node ID’, ‘Sequence Number’, ‘Route Record’ and ‘Path Weight’. The Sequence Number field is a monotonically increasing value (incremented by 1) and is used to keep track of the RREQ messages meant to discover route between a particular source-destination pair. The Sequence Number field guides the intermediate forwarding nodes to detect redundant copies of the RREQ messages and also prevents these messages from getting struck in broadcast loops. With flooding, each node in the network broadcasts the RREQ message exactly once. The ‘Route Record’ field is used to store the sequence of nodes traversed by the RREQ message on its path from the originating source node. The ‘Path Weight’ stores the sum of all the weights of the links through which the RREQ message has traversed starting from the source node. As described in Section 2, the weight of a link is 1 plus the inverse of the predicted Link Expiration Time (LET) value of the link at the current instant (refer to equations 1 and 2 in Section 2). The source node inserts its own ID in the Route Record field and initializes the Path Weight field with a value of 0.

When an intermediate node (not the destination node) receives a RREQ message for the first time (identified using the trio of *Source Node ID*, *Destination Node ID* and *Sequence Number* fields), the intermediate node updates the *Path Weight* value in the RREQ message by adding to it the value of weight of the link to the upstream node from which the message was received. The value of the weight of the link is 1 plus the inverse of the Link Expiration Time (LET) value currently estimated by the intermediate node for its link with the upstream node that sent the RREQ message. The intermediate node also appends its own ID to the list of node IDs in the *Route Record* field of the RREQ message. The message is then broadcast to the neighborhood. If an intermediate node receives a RREQ message that it has already seen, the message is simply dropped.

4.2 Route Selection and Propagation of the Route Reply (RREP) Message

When the destination node receives a RREQ message, the node updates the *Path Weight* value in the message by adding to it the weight of the link (1 plus the inverse of the LET value) to the upstream node that sent the message. After receiving the first RREQ message for a particular request-reply flooding cycle initiated by the source node, the destination node waits for a certain time to receive several other RREQ messages. The destination node computes the *Path Weight* values for every RREQ message received within the waiting time period. The destination node then chooses the RREQ message that has the minimum *Path Weight* value among all such RREQ messages received and generates a Route Reply (RREP) message that is to be sent on the reverse path of the sequence of nodes through which the RREQ message traversed. To facilitate this, the destination node copies the contents of the *Route Record* field of the RREQ message to the *Route Record* field of the RREP message. The structure of the RREP message is shown in Figure 2.

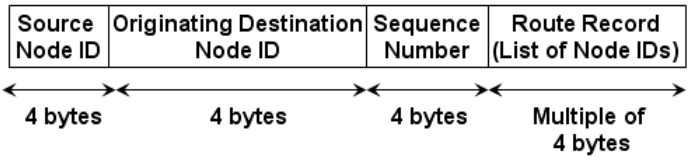


Figure 2: Structure of the Route Reply (RREP) Message

Each node in the network maintains a unicast routing table that is an ordered entry of <*key*, *value*> pairs, where the <*key*> is the <*Destination Node ID*> and the <*value*> is the tuple <*Next Hop Node ID*, *List of Upstream Nodes*, *Source Node ID*, *RREQ-RREP Sequence Number*>. The structure of a route entry maintained in the unicast routing table of an intermediate forwarding node is shown in Figure 3. The *Next Hop Node ID* corresponds to the node that lies on the minimum weight path to the *Destination Node ID*. The *RREQ-RREP Sequence Number* and the *Source Node ID* fields correspond to the Sequence Number field value of the latest RREQ-RREP broadcast cycle initiated by the *Source Node ID*, through which the route entry was updated. The intermediate forwarding node also includes the upstream node (on the path towards the *Source Node* ID, listed in the *Route Record* field of the RREP message) to the *List of Upstream Nodes* and forwards the RREP message to that node. The purpose of maintaining a *List of Upstream Nodes* at an intermediate node is to inform all these nodes of a link failure, if it happens, with the *Next Hop Node ID* so that these upstream nodes do not use the intermediate node to forward the data packets meant for the *Destination Node ID*. The propagation of a Route Error Message (RERR) triggered due to link failure is explained in Section 4.4.

Key	Value			
Destination Node ID	Next Hop Node ID	List of Upstream Nodes	Source Node ID	RREQ-RREP Sequence Number

Figure 3: Structure of the SILET Unicast Routing Table Maintained at an Intermediate Node

Key		Value
Destination Node ID	RREQ-RREP Sequence Number	Next Hop Node ID

Figure 4: Structure of the SILET Unicast Routing Table Maintained at a Source Node

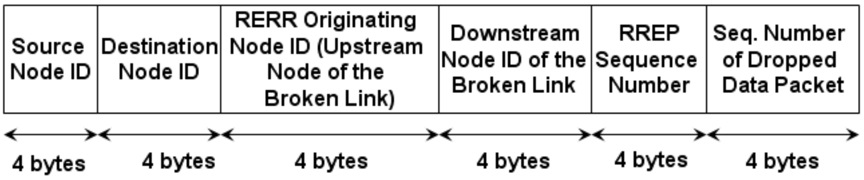
The unicast routing table at the source node consists of <*key*, *value*> pairs, where the <*key*> is the tuple <*Destination Node ID*, *RREQ-RREP Sequence Number*> of the latest broadcast query-reply flooding cycle and the <*value*> is the *Next Hop Node ID* through which packets meant for the *Destination Node ID* are sent. The *Next Hop Node ID* is updated corresponding to the ID of the neighbor node that forwarded the RREP message originating from the *Destination Node ID*. Figure 4 illustrates the structure of the unicast routing table maintained at the source node.

4.3 Route Acquisition and Data Transmission

After launching the broadcast of the RREQ messages as part of the route discovery initiative, the source node waits for a *Route Acquisition Time* to receive a RREP message from the destination node. If no RREP message is received within the *Route Acquisition Time*, the source node broadcasts the next RREQ message (Sequence Number field value incremented by 1) to its neighbors. If a RREP message is received within the *Route Acquisition Time*, the source node starts sending the data packets using the path established through the RREQ-RREP broadcast cycle. The *Route Acquisition Time* is dynamically reset based on the time it took to receive the RREP messages as part of the successful route discovery initiatives in the recent past.

4.4 Route Maintenance

An *s-d* path is broken when even one of the links is broken. The failure of a link is typically detected when a forwarding node could not successfully transmit a data packet to the corresponding next hop node with which the link is broken. Since SILET requires nodes to periodically exchange beacons in the neighborhood, the failure of a link could also be detected when an upstream node fails to receive beacons from the downstream node of the broken link. In either case, the upstream intermediate node of the broken link initiates a Route Error (RERR) message and sends it to all the nodes in the *List of Upstream Nodes* of the routing table entries that have the downstream node (with which the link is broken) as the next hop node. The RERR message gets propagated further all the way to the source nodes of the *s-d* sessions that had paths involving the broken link. The structure of the RERR message is shown in Figure 5. The RERR message is also used to piggyback information to the source node of an *s-d* session about the failure to forward a data packet towards its destination node; the *s-d* session is identified using the *Source Node ID*, *Destination Node ID* and *Sequence Number* (of the Data packet and RREP) fields.



Source Node ID	Destination Node ID	RERR Originating Node ID (Upstream Node of the Broken Link)	Downstream Node ID of the Broken Link	RREP Sequence Number	Seq. Number of Dropped Data Packet
4 bytes	4 bytes	4 bytes	4 bytes	4 bytes	4 bytes

Figure 5: Structure of the Route Error Message (RERR)

5. Review of the Unicast Routing Protocols Simulated in this Paper

In addition to SILET, we simulate three MANET unicast routing protocols: (i) Dynamic Source Routing (DSR) protocol [24], (ii) Associativity-Based Routing (ABR) protocol [25] and (iii) Flow-Oriented Routing Protocol (FORP) [10]. The following sections briefly review the working of each of these three routing protocols.

5.1 Dynamic Source Routing (DSR) Protocol

As the name indicates, DSR [24] is a source-routing based protocol wherein the entire path information is recorded in the header of each data packet; thus, intermediate nodes do not need to store up-to-date routing information in their forwarding tables. Also, DSR does not require periodic beacon exchange in the neighborhood unlike the stability-based routing protocols. Route discovery is through the traditional route request-reply flooding cycle. The destination responds with a Route Reply (RREP) message by choosing the Route Request (RREQ) message that traversed through the minimum number of hops. Any tie is arbitrarily broken. The Route Record field in the RREP message is stored by the source node as part of its route information to the destination node and is included in the header of every data packet sent after the successful route discovery. DSR also supports route caching and promiscuous listening wherein an intermediate node stores the route learnt to a destination node and responds with a RREP for a RREQ message instead of letting the RREQ to propagate all the way to the destination. But, in networks of moderate and high mobility, cached routes are more likely to become stale by the time they are used. Promiscuous listening also leads to significant energy loss as the node has to stay turned on all the time. Hence, we disable both the cache routing and promiscuous listening modules of DSR in our simulations.

5.2 Associativity-Based Routing (ABR) Protocol

ABR [25] is a stability-oriented unicast routing protocol for MANETs. Here, a node estimates the stability of a link with a neighbor node in the form of 'Associativity ticks', which is basically the number of beacons received from the neighbor node since a link was formed between the two nodes. The 'Associativity Threshold' of a link is given by $2R/v_{rel}$ where R is the transmission range of the nodes and v_{rel} is the relative velocity between the constituent end nodes of the link. A node characterizes a link with its neighbor node as a strong link if the number of beacons received from the neighbor node is at least the Associativity Threshold for the link; otherwise, the link is characterized to be a weak link. Route discovery in ABR is through the classical broadcast query-reply cycle with the RREQ message containing two counters – one counter for the number of strong links through which the message has passed through and another counter for the number of weak links through which the message has passed through. These two counters are initialized to zero by the source node. When an intermediate node sees the RREQ message for the first time, it increments the counter for the number of strong or weak links depending on the nature of its link to the upstream node that sent it the RREQ message, and further propagates the message in its neighborhood. The destination node waits to receive one or more RREQ messages and chooses the RREQ message that traversed through the maximum proportion of stronger links. A RREP message is sent back to the source node on the chosen path.

5.3 Flow-Oriented Routing Protocol (FORP)

FORP [10] is a stability-based routing protocol that uses the link expiration time (LET) as the routing metric. The LET of a link is given according to equation (1). The Route Expiration Time (RET) for a path is the minimum of the LETs of the constituent links. During the broadcast route discovery process, the RET of the paths, starting from the source node, traversed by the RREQ messages is constantly updated by the intermediate nodes before forwarding the message further. An intermediate node updates the RET value in the RREQ message received for the first time, if the LET of the link to the upstream node from which the message was received is lower than the RET currently recorded in the message; otherwise, the message is simply forwarded. The destination node receives RREQ messages through several paths; the path with the largest RET is chosen and a RREP is sent to the source node along the chosen path.

6. Simulations

The performance of SILET has been compared with the DSR [24], ABR [25] and FORP [10] protocols. The simulations were conducted in the ns-2 simulator (v. 2.32) [12]. While we used the implementation of DSR that came with ns-2, we implemented the SILET, ABR and FORP protocols in the ns-2 simulator. The network dimensions are 1000m x 1000m. The transmission range per node is 250m and is the same for all the nodes in the network (i.e. we assume a homogeneous MANET). Network density is varied by conducting simulations with 50 nodes (low density) and 100 nodes (high density). The nodes are initially assumed to be uniform-randomly distributed in the network. The MAC (Medium Access Control) layer uses the distributed co-ordination function (DCF) of the IEEE Standard 802.11 [26] for wireless LANs.

Traffic sources are continuous bit rate (CBR)-based. The number of source-destination (*s-d*) sessions run simultaneously is 15. The starting times of these *s-d* sessions is uniformly distributed between 1 to 50 seconds. Data packets are 512 bytes in size and the packet sending rate is 4 data packets per second. While distributing the source-destination roles for each node, we saw to it that a node does not end up a source of more than two sessions and/or also not as a destination for more than two sessions.

Nodes move according to the Random Waypoint mobility model [27] with each node moving independent of the other nodes in the network. A node starts moving from an arbitrary location to a randomly chosen destination location within the range [0…1000m, 0…1000m], corresponding to the 1000m x 1000m network, and moves to the chosen location at a speed uniform-randomly picked from the range [0,…, v_{max}] where v_{max} represents the maximum node velocity. The v_{max} values used in the simulations are 5 m/s, 25 m/s and 50 m/s representing scenarios of low, moderate and high node mobility respectively. Pause time is 0 seconds. For a given condition of network density and v_{max} values, 5 different mobility profiles were generated. The performance results shown in Figures 6 through 8 are average values when the 15 *s-d* sessions are run over the 5 mobility profiles generated for every combination of network density and node mobility (v_{max}) values. So, basically, each data point displayed in Figures 6 through 8 is an average of 15 * 5 = 75 values.

The chosen speed and density values represent a range of scenarios: With the maximum node velocity (v_{max}) values of 5, 25 and 50 m/s chosen for the simulations, the average velocity of a node is roughly about 2.5 m/s, 12.5 m/s and 25 m/s respectively and these values correspond to 9 km/hour, 45 km/hour and 90 km/hour, typical of the highway speeds. The chosen node density of 50 nodes and 100 nodes in a network of dimensions 1000m x 1000m, with each node having a transmission range of 250m, corresponds to operating with about 10 and 20 neighbors per node respectively. From the simulation results, we notice that the average hop count per path does not change much with node velocity and varies slightly with node density. The path lifetime varies much with node velocity; but, does not significantly change with node density. It is the end to end delay per data packet that changes much with node density (due to the impact of the forwarding load on the nodes and interference in the neighborhood). With 10 and 20 neighbors per node, we are able to more effectively capture the impact of the above two factors on the end-to-end delay per data packet.

The stability-based routing protocols use beacon control messages to let each node advertise its presence to neighbors. Beacons are exchanged every one second. For ABR, we will let each node to mention its current velocity in the beacon message to neighbors. This will help a neighbor node to calculate the relative velocity between itself and the node sending the beacon message. For FORP and SILET, we will let each node to send information about its location and current velocity in the beacons. Each node will also keep track of the most recently advertised location of its neighbor nodes. This will help to determine the direction in which the neighbor node is moving.

6.1 Performance Metrics

The performance metrics evaluated through the simulations are the following:

- ***Path Lifetime*****:** For every source-destination (*s-d*) path that is determined during the simulation session, we keep track of the duration the *s-d* path exists. The path lifetime is the average value of the duration of the *s-d* paths, over the entire simulation time, across all the simulation runs corresponding to a particular combination of network density and node mobility.

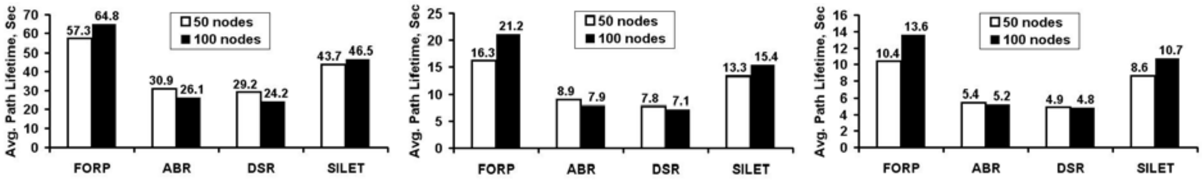
- ***Hop Count per Source-Destination Path*****:** This is the time averaged value of the hop count (number of edges) of the source-destination (*s-d*) paths computed over the entire simulation session. For example, if we used a sequence of three *s-d* paths with a lifetime and hop count of (15 seconds, 4 edges), (22 seconds, 3 edges) and (13 seconds, 5 edges) spanning the entire simulation time of 50 seconds, the time-averaged value for the hop count per *s-d* path is (15*4 + 22*3 + 13*5)/50 = 3.82 and not simply 4.0 (the average of 4, 3 and 5 – the number of edges in each of these paths).

- ***End-to-End Delay per Data Packet*****:** This is the average of the delay incurred by the data packets that originate at the source node and delivered at the destination node. The delay incurred by a data packet includes all the possible delays – the buffering delay due to the route acquisition latency, the queuing delay at the interface queue to access the medium, transmission delay, propagation delay, and the retransmission delays due to collisions at the MAC (Medium Access Control) layer.

The standard deviation (std. dev.) values of the simulations conducted for different performance metrics under different maximum node velocity (v_{max}) values are displayed next to the v_{max} value in the Figures 6 through 8. The std. dev. value reported for a metric under a particular condition is the maximum among all those observed for the different protocols simulated under that condition.

6.2 Path Lifetime

The proposed SILET protocol discovers paths that are significantly more stable than the paths discovered using the minimum-hop DSR and the stability-based ABR (refer to Figure 6). The lifetime of SILET paths is about 90% to 125% (high density networks) and 50% to 76% (low density networks) more than the lifetime of DSR paths. Similarly, the lifetime of SILET paths is about 80% to 105% (high density networks) and 40% to 60% (low density networks) more than the lifetime of ABR paths. For a given network density, the percentage difference in the lifetime of the SILET paths, when compared with that of the DSR and ABR paths, increases with increase in node mobility. Also, for a given node mobility (i.e., value of v_{max}), the percentage difference in the lifetime per path for ABR and DSR vis-à-vis SILET increases significantly (by 40% to 50%) as we double the number of nodes in the network from 50 to 100. Thus, SILET discovers relatively more stable paths compared to that of ABR and DSR with increase in network density and/or node mobility.



v_{max} = 5 m/s (std. dev. = ±2.6 sec) v_{max} = 25 m/s (std. dev. = ±1.6 sec) v_{max} = 50 m/s (std. dev. = ±0.7 sec)

Figure 6: Average Path Lifetime

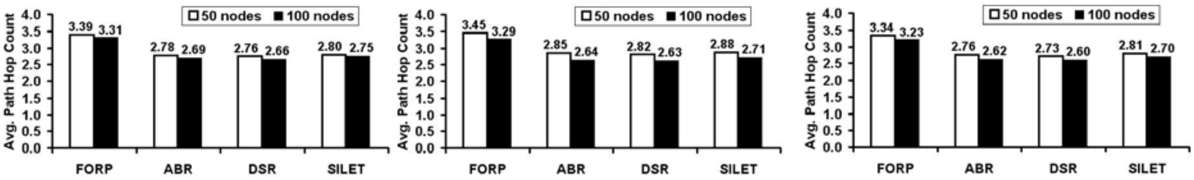
Though, FORP discovers the most stable paths, the lifetime of SILET paths is only about 20%-30% (low density network) and 26%-40% (high density network) lower than that of FORP paths. As described in Section 6.3 on path hop count, SILET compensates the relatively lower lifetime per path (compared to FORP) with a relatively lower hop count per path. Unlike the trend observed for SILET vis-à-vis ABR and DSR, for a given network density, the percentage difference in the lifetime per path for SILET and FORP decreases with increase in node mobility. Thus, for a given network density, with increase in node mobility, the lifetime per SILET path gradually approaches the lifetime per FORP path. Also, for a given node mobility, the percentage difference in the lifetime per SILET path compared to the FORP path increases only marginally (by at most 15%) as we double the network density. Thus, SILET accomplishes its objective of incurring path lifetime closer to that of FORP and at the same time incurs lower hop count and lower end-to-end delay per data packet as discussed in the subsequent sections.

For a given node mobility, the lifetime for the FORP and SILET paths increase with increase in network density and the lifetime for the ABR and DSR paths decrease with increase in the number of nodes in the network. The decrease in the path lifetime for DSR paths (with increase in network density) can be attributed to the Edge Effect problem [28] according to which the physical Euclidean distance between the constituent links of a minimum hop path is more likely to be closer to the transmission range of the node. Due to the presence of multiple alternate nodes to choose from for the next hop node that will be part of the minimum hop paths, the Edge Effect problem is more dominant in high-density networks. If the physical distance between the end nodes of a link at the time of formation of the minimum hop paths is closer to the transmission range per node, the probability that such a link will exist for the near future is highly likely to be very low. That is the two end nodes of the links of a minimum hop path are highly likely to separate at any time in the near future. In the case of ABR, especially in high-density networks, there is possibility of finding multiple paths with the same larger percentage of stable links; the destination breaks the tie by choosing paths with lower hop count and such paths are also prone to the Edge Effect. The FORP and SILET links are not much prone to Edge Effect, because the physical distance between the constituent end nodes of the links of their paths is not close to the transmission range and the links are carefully selected based on the predicted LET. Since, the number of alternative links increases with increase in network density, FORP and SILET protocols get more choices of links to select from during route discovery.

6.3 Hop Count per Source-Destination Path

The average hop count per FORP path is the largest of all the protocols simulated (refer to Figure 7) and is relatively about 20% to 25% larger. The larger is the hop count per path, the larger will be the energy consumption per path as well the bandwidth usage. Also, in networks of low mobility, when the route discovery control overhead is less dominating, the larger the hop count per path, the larger will be the end-to-end delay per data packet (as observed in the results for Section 6.4). Hence, MANET routing protocols are always desired to have a lower hop count per path. However, as we observe in the results from Section 6.2 as well as this section, there is a stability-hop count tradeoff. FORP is able to determine paths that have the longest lifetime; however, the hop count of such long-living stable paths is also the largest among the routing protocols.

SILET has been proposed with the objective of neutralizing the stability-hop count tradeoff as much as possible. The simulation results illustrate that SILET does accomplish its objective to a significant extent. The hop count of SILET paths is only at most 4% larger than the hop count of the minimum-hop based DSR protocol, a classical well-studied MANET routing protocol and is at most 3% longer than the paths of the stability-based ABR, another classical well-studied MANET routing protocol. The hop count of FORP paths is about 20% larger than the hop count per path incurred for SILET. Thus, SILET determines stable paths that are off by about 20% to 40% with respect to path lifetime compared to FORP; the hop counts of these stable paths is about 20% lower than that of FORP. As observed in Section 6.2 for path lifetime, the lifetime of SILET paths is significantly larger than the lifetime of DSR and ABR paths. Thus, SILET need not be just considered as an alternative MANET routing protocol to yield lower hop count per path; SILET can hereafter be used (instead of DSR or ABR) as the routing protocol that will yield minimum hop count stable paths that will exist for a significantly longer lifetime.



v_{max} = 5 m/s (std. dev. = ± 0.32) v_{max} = 25 m/s (std. dev. = ± 0.35) v_{max} = 50 m/s (std. dev. = ± 0.28)

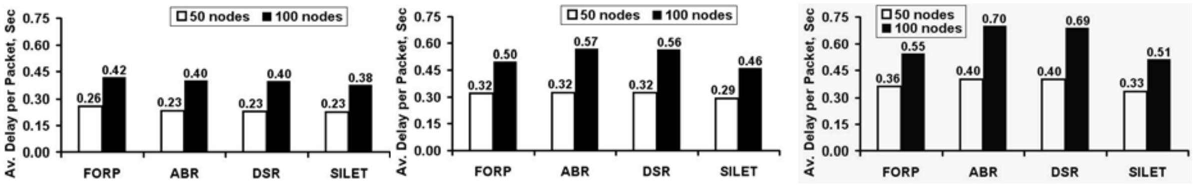
Figure 7: Average Hop Count per Source-Destination (*s*-*d*) Path

For a given node mobility (i.e., value of v_{max}), all the four routing protocols are able to reduce the hop count per path with increase in node density. This is because, in high-density networks, the route discovery packets are likely to traverse through multiple paths in the network and if there is a tie among the best paths chosen according to a route selection principle (for e.g., to minimize the sum of the inverse of the LETs), the tie is broken by choosing the minimum hop count path among the best paths.

6.4 End-to-End Delay per Data Packet

Owing to its relatively lower hop count per path and longer lifetime per path, for all the scenarios, SILET incurs the lowest end-to-end delay per data packet among the protocols studied in the simulations (refer to Figure 8). The end-to-end delay per data packet depends on several factors: (i) route acquisition time – the time it takes to successfully discover a source-destination route, (ii) the route discovery control overhead – the route discovery packets are given priority in the queues at the nodes; so, if a routing protocol incurs repeated path failures leading to frequent route discoveries, then the data packets suffer excessively delay waiting at the queues of the nodes and (iii) hop count – larger the hop count, it is more likely that the end-to-end delay per data packet will also be larger. The direct impact of hop count on end-to-end delay is predominant in low mobility scenarios when the other two factors (i.e., (i) and (ii)) mentioned above are less dominating. The end-to-end delay per data packet is also dependent on the channel access delay, which is the time it takes for a node to successfully gain access to the channel in order to transmit the data packet to a downstream node on the *s*-*d* path. The channel access delay starts to influence the end-to-end delay in high-density networks, as also explained later in this section.

Since FORP has larger hop count per source-destination path, the direct impact can be observed in low mobility networks, where the end-to-end delay per data packet when sent over FORP routes is about 15% larger than that of the SILET routes. As DSR and ABR also discover minimum hop or close to minimum hop routes, the end-to-end delay per data packet for DSR and ABR routes in low mobility scenarios is only about 6% larger than that of SILET routes. However, as we increase node mobility (v_{max} increased from 5 m/s to 25 m/s and 50 m/s), the topology of the network changes dynamically, triggering frequent route failures in case of the minimum-hop based DSR as well as for ABR. This significantly increases the route discovery control overhead as well as the route acquisition time. On the other hand, since FORP and SILET are able to use the predicted LET as part of their link weights, these two protocols suffer far less route failures with increase in node mobility. Thus, we observe FORP routes to start incurring lower end-to-end delay per data packet than the DSR and ABR routes. SILET routes continue to incur the lowest end-to-end delay per data packet. In moderate mobility scenarios (v_{max} = 25 m/s), the end-to-end delay per data packet for the three routing protocols (ABR, DSR and FORP) is about 10% to 20% more than that of SILET routes. However, as the mobility is increased further (v_{max} = 50 m/s), the end-to-end delay per data packet for ABR and DSR routes is 20% (low density networks) to 36% (high density networks) larger than that of SILET routes.



v_{max} = 5 m/s (std. dev. = ±0.020 s) v_{max} = 25 m/s (std. dev. = ±0.023 s) v_{max} = 50 m/s (std. dev. = ±0.028 s)

Figure 8: Average End-to-End Delay per Packet

For a given network density, the percentage difference between the end-to-end delay per data packet for FORP and SILET routes decreases with increase in node mobility. This can be attributed to the ability of FORP to discover routes that have longer lifetime than SILET. However, since the percentage difference in the lifetime of SILET paths and FORP paths also decreases with increase in node mobility and SILET uses paths with relatively lower hop count, the end-to-end delay per data packet for SILET is about 10% lower than that of FORP routes even in conditions of high network density and high node mobility. For a given node mobility, the end-to-end delay per data packet for all the four routing protocols increases with increase in network density and this can be attributed to the relatively longer time it takes to successfully access the channel in high-density networks compared to low-density networks.

7. SILET – Advantages with Multi-objective Optimization and Applications

By including both the LET and the hop count together in the optimization function, we are able to simultaneously optimize both the hop count as well as the lifetime of the paths. In other words, the proposed SILET routing protocol falls under the MANET routing protocols based on multi-objective optimization and attempts to minimize the stability-hop count-delay tradeoff observed with the classical single-objective based MANET routing protocols such as DSR and FORP. DSR is minimum-hop based, incurring relatively lower delay (attributable somewhat to the lower hop count); however, DSR routes are highly unstable due to the weak links constituting the minimum hop paths. On the other hand, FORP paths are highly stable due to the relatively strong links chosen taking into consideration their predicted LET; but, such stable paths have a much larger hop count as well as suffer a longer delay. With SILET, we are trying to determine stable paths that have a hop count closer to that of being the minimum and at the same time the SILET paths do not incur significantly longer delay.

Stable paths with lower end-to-end delay are preferable for real-time applications where the packets should reach the destination within a certain time and at the same time, the packets should suffer minimum jitter (i.e., variations in the inter-packet delay). To minimize jitter, it would be better to route the data packets on the same path, as much as possible. If the packets go through the same sequence of nodes, the variations in the delay suffered by the successive packets reaching the destination would be minimal, thus leading to lower jitter. If we route the packets through different paths (due to frequent route failures), then there would be appreciable variations in the delay suffered by the packets reaching the destination. This is not preferable for multi-media real-time applications that prefer packets to be routed through paths that incur zero or minimum jitter. Routing on stable minimum hop paths would also lead to lower energy consumption across the network as only very few nodes are used for routing and there is no need for frequent network-wide flooding-based route discoveries.

8. Conclusions and Future Work

Our primary contribution in this paper is the design and development of a unicast routing protocol that discovers stable routes based on the inverse of the link expiration times (LET), referred to as SILET. The optimization function of SILET has been carefully chosen to simultaneously minimize the hop count of the routes as well as maximize their lifetime. The weight of a link is ‘1’ plus the inverse of the LET of the link and SILET targets to determine routes that have minimum value for the sum of the link weights. The term ‘1’ in the link weight helps to minimize the hop count while the term ‘inverse of the LET’ in the link weight helps to determine stable routes.

Through simulation studies, we observe that SILET effectively minimizes the stability-hop count tradeoff that has been predominantly observed in well-known minimum-hop based (DSR) and stability-based routing protocols (FORP). A minimum-hop based protocol like DSR incurs more route transitions (lower route lifetime) compared to FORP; whereas, FORP routes have significantly larger hop count than the minimum-hop based DSR routes. As a result of this stability-hop count tradeoff, different categories of routing protocols had to be chosen for different scenarios. In networks of limited node mobility, minimum-hop based routing is considered preferable as there are fewer global route discoveries (lower control overhead) and the hop count is also lower; whereas, in networks of high node mobility, stability-based routing appears to be better as the number of global route discoveries is relatively lower. But, real-time performance metrics such as the end-to-end delay per data packet suffer as a result of larger hop count and/or larger route discovery control overhead. With the development of SILET, the stability-hop count tradeoff has been effectively minimized and the routing protocol can be used for both low mobility and high mobility scenarios; SILET incurs the lowest end-to-end delay per data packet for all combinations of mobility and density scenarios (low and high) simulated in this paper. Thus, SILET is a valuable addition to the MANET literature and is truly the first such unicast routing protocol to effectively minimize the stability-hop count tradeoff.

We are currently working on developing a multicast version of SILET that can use the same weight function for the links and try to discover stable multicast trees that have a lower hop count per source-receiver path. In future, we plan to implement the MANET energy consumption models [30] and compare the stability-based protocols like FORP, ABR and

SILET with respect to metrics such as node lifetime (time of first node failure due to battery exhaustion), network lifetime (time of network disconnection due to node failures) and fairness of node usage. We also plan to evaluate the minimum-hop and stability-based MANET routing protocols under different mobility models (like the Random Direction Model, Random Walk Model and Gauss-Markov Model) [29] and rank the routing protocols with respect to hop count, delay, route lifetime as well as the stability-hop count tradeoff.

9. Acknowledgments

The work leading to this paper was partly funded through the U.S. National Science Foundation (NSF) CCLI/TUES grant (DUE-0941959) on “Incorporating Systems Security and Software Security in Senior Projects.” The views and conclusions contained in this paper are those of the authors and do not represent the official policies, either expressed or implied, of the funding agency.

10. References

[1] C. Siva Ram Murthy and B. S. Manoj, *Ad Hoc Wireless Networks: Architectures and Protocols*, Prentice Hall, 1st Edition, June 2004.

[2] N. Meghanathan, “Survey and Taxonomy of Unicast Routing Protocols for Mobile Ad hoc Networks,” *The International Journal on Applications of Graph Theory in Wireless Ad hoc Networks and Sensor Networks*, vol. 1, no. 1, pp. 1-21, December 2009.

[3] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination Sequenced Distance Vector Routing for Mobile Computers,” *Proceedings of ACM SIGCOMM (Special Interest Group on Data Communications*), pp. 234-244, October 1994, http://dx.doi.org/10.1145/190314.190336.

[4] P. Jacquet, P. Muhlethaler, T. Clausen, A. Laouiti, A. Qayyum and L. Viennot, “Optimized Link State Routing Protocol for Ad Hoc Networks,” *Proceedings of the IEEE International Multi Topic Conference*, pp. 62-68, Pakistan, December 2001, http://dx.doi.org/10.1109/INMIC.2001.995315.

[5] J. Broch, D. A. Maltz, D. B. Johnson, Y. C. Hu and J. Jetcheva, “A Performance Comparison of Multi-hop Wireless Ad hoc Network Routing Protocols,” *Proceedings of the 4th Annual International Conference on Mobile Computing and Networking,* pp. 85-97, October 1998, http://dx.doi.org/10.1145/288235.288256.

[6] P. Johansson. T. Larson, N. Hedman, B. Mielczarek and M. DegerMark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad hoc Networks,” *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking,* pp. 195-206, August 1999, http://dx.doi.org/10.1145/313451.313535.

[7] N. Meghanathan, “Exploring the Stability-Energy Consumption-Delay-Network Lifetime Tradeoff of Mobile Ad Hoc Network Routing Protocols,” *Journal of Networks*, vol. 3, no. 2, pp. 17-28, February 2008, .

[8] N. Meghanathan, “Design of a Quality of Service-based Multicast Routing Protocol for

Mobile Ad hoc Networks," *Journal of Theoretical and Applied Information Technology*, vol. 13, no. 2, pp. 117-124, March 2010.

[9] W. Su, S-J. Lee and M. Gerla, "Mobility Prediction and Routing in Ad hoc Wireless Networks," *International Journal of Network Management*, vol. 11, no. 1, pp. 3-30, 2001, http://dx.doi.org/10.1002/nem.386.

[10] W. Su and M. Gerla, "IPv6 Flow Handoff in Ad hoc Wireless Networks using Mobility Prediction," *Proceedings of the IEEE Global Telecommunications Conference*, pp. 271-275, December 2009, http://dx.doi.org/10.1109/GLOCOM.1999.831647.

[11] T. H. Cormen, C. E. Leiserson, R. L. Rivest and C. Stein, *Introduction to Algorithms*, 3rd Edition, MIT Press, September 2009.

[12] K. Fall and K. Varadhan, "ns notes and documentation," The VINT Project at LBL, Xerox PARC, UCB, and USC/ISI, http://www.isi.edu/nsnam/ns, August 2001.

[13] K. Kant and L. K. Aswathi, "Stable Link Based Multicast Routing Scheme for MANET," *Proceedings of the International Conference on Computational Intelligence & Comm. Networks*, pp. 296-300, Nov. 2010, http://dx.doi.org/10.1109/CICN.2010.68.

[14] M. H. Mamoun, "A Novel Technique for the Route Selection in DSR Routing Protocol," *International Journal of Video and Image Processing and Network Security*, vol. 11, no. 3, pp. 1-4, June 2011.

[15] N. Meghanathan, D. Thomas and E. S. Addison, "The Beaconless Node Velocity-based Stable Path Routing Protocol for Unicasting and Multicasting in Mobile Ad hoc Networks," *Journal of Emerging Trends in Computing and Information Sciences*, vol. 2, no. 1, pp. 49-61, January 2011.

[16] 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*, pp. 36-45, February 1997, http://dx.doi.org/10.1109/98.575990.

[17] S. Agarwal, A. Ahuja, J. P. Singh and R. Shorey, "Route-lifetime Assessment based Routing (RABR) Protocol for Mobile Ad hoc Networks," *Proceedings of the IEEE International Conference on Communications (ICC)*, vol. 3, pp. 1697-1701, June 2000, http://dx.doi.org/10.1109/ICC.2000.853783.

[18] S. S. Manvi, M. S. Kakkasageri, S. Paliwal and R. Patil, "ZLERP: Zone and Link Expiry based Routing Protocol for MANETs," *International Journal of Advanced Networking and Applications*, vol. 2, no. 3, pp. 650-655, 2010.

[19] C. M. Kozierok, *A Comprehensive, Illustrated Internet Protocols Reference*, No Starch Press, October 2005.

[20] B. Hofmann-Wellenhof, H. Lichtenegger and J. Collins, *Global Positioning System: Theory and Practice*, 5th ed., Springer, September 2004.

[21] W. Keiss, H. Fuessler and J. Widmer, "Hierarchical Location Service for Mobile Ad hoc Networks," *ACM SIGMOBILE Mobile Computing and Communications Review*, vol. 8, no. 4, pp. 47-58, 2004, http://dx.doi.org/10.1145/1052871.1052875.

[22] F. Kuhn, T. Moscibroda and R. Wattenhofer, "Unit Disk Graph Approximation," *Proceedings of the Workshop on Foundations of Mobile Computing*, pp. 17-23, October 2004, http://dx.doi.org/10.1145/1022630.1022634.

[23] N. Meghanathan, "Path Stability based Ranking of Mobile Ad Hoc Network Routing

Protocols," *ISAST Transactions Journal on Communications and Networking*, vol. 1, no. 1, pp. 66-73, August 2007.

[24] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-hop Wireless Ad hoc Networks," *Ad hoc Networking*, edited by Charles E. Perkins, Chapter 5, pp. 139-172, Addison-Wesley, 2001.

[25] C-K. Toh, "Associativity-Based Routing for Ad hoc Mobile Networks," *Wireless Personal Communications: An International Journal*, vol. 4, no. 2, pp. 103-139, March 1997, http://dx.doi.org/10.1023/A:1008812928561.

[26] G. Bianchi, "Performance Analysis of the IEEE 802.11 Distributed Coordination Function," *IEEE Journal of Selected Areas in Communication*, vol. 18, no. 3, pp. 535-547, March 2000, http://dx.doi.org/10.1109/PIMRC.2006.254094.

[27] C. Bettstetter, H. Hartenstein and X. Perez-Costa, "Stochastic Properties of the Random-Way Point Mobility Model," *Wireless Networks*, pp. 555-567, vol. 10, no. 5, September 2004, http://dx.doi.org/10.1023/B:WINE.0000036458.88990.e5.

[28] G. Lim, K. Shin, S. Lee, H. Yoon, and J. S. Ma, "Link Stability and Route Lifetime in Ad hoc Wireless Networks," *Proceedings of the International Conference on Parallel Processing Workshops*, pp. 116-123, August 2002, http://dx.doi.org/10.1109/ICPPW.2002.1039720.

[29] T. Camp, J. Boleng and V. Davies, "A Survey of Mobility Models for Ad Hoc Network Research," *Wireless Communication and Mobile Computing*, vol. 2, no. 5, pp. 483-502, September 2002.

[30] L. M. Feeney, "An Energy Consumption Model for Performance Analysis of Routing Protocols for Mobile Ad hoc Networks," *Journal of Mobile Networks and Applications*, vol. 3, no. 6, pp. 239-249, June 2001, http://dx.doi.org/10.1023/A:1011474616255.

Copyright Disclaimer