Title
Proactive Path Maintenance in Regions of Interest

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2008-09-29

Proactive Path Maintenance over Regions of Interests in MANETs

Dhananjay Sampath† and J.J. Garcia-Luna-Aceves † ‡
† Computer Engineering Department, University of California, Santa Cruz, 1156 High Street, Santa Cruz, CA 95064, USA
‡ Palo Alto Research Center (PARC), 3333 Coyote Hill Road, Palo Alto, CA 94304, USA
Email:{dsampath, jj}@soe.ucsc.edu

***Abstract*—We present Elliptic Demarcation of Information Transfer (EDIT) as a scheme to maintain paths between a source and destination more robustly and limit signaling overhead incurred in mobile ad hoc networks (MANET). EDIT establishes regions of interest on demand, based on distances between a relay node and a source-destination pair, and maintains proactive signaling within this region of interest. We prove the correctness of EDIT, which is based on a progressive sequence numbering scheme, and show that the elliptical regions of interest built by EDIT are more efficient than the traditional use of TTLs, which establish circular, undirected boundaries around destinations. Simulation results comparing EDIT against location aided routing (LAR)[1], AODV[2] and OLSR[3] indicate that EDIT enables on-demand routing that is far more efficient than AODV and OLSR, and is comparable to LAR, but without the need for geo-location information.**

I. INTRODUCTION

Many routing protocols have been proposed for mobile ad hoc networks (MANET) to date. The basic routing schemes can be classified as proactive and on-demand. In the basic proactive routing scheme, either link-state or distance updates propagate throughout the network for nodes to update their routing-table entries for all destinations. In on-demand routing schemes, route requests are disseminated throughout the network to find destinations of interest. In large MANETs. the frequent propagation of signaling packets, owing to mobility and interference, throughout the entire network, is a detriment to network performance.

Several approaches have been proposed to reduce the communication overhead incurred in updating routing tables and maintaining path information. These approaches include organizing the network into clusters, reducing the rate at which signaling packets propagate away from the origin of an update, and using geo-location information to direct packets to particular regions of the network.

Examples of schemes that reduce the rate at which signaling is propagated away from destinations also date back to PRNET times [4]. Recent approaches include expanding-ring search optimizations used in on-demand routing protocols. Another example is the fisheye state routing [5] scheme. The OPRAH protocol [6] uses an opportunistic routing scheme that exploits promiscuous listening by radios to cache route information and impose constraints on the forwarding of this information within a given threshold. The limitation with threshold-control schemes is that they may not provide enough accuracy when sources and destinations are far apart, and information about destinations or links propagates in all directions, not just towards those nodes that need the information.

The location aided routing (LAR) scheme is a well-known approach based on location information to direct the transmission of route requests. However, most location-based schemes, including LAR, are limited by their need to use such devices as GPS for the localization of nodes which does not perform well when line of sight to satellites is not available.

Our work is motivated by the performance gains attained with the use of location-information in LAR and similar schemes, the reduction in signaling overhead in recent threshold-based approaches like OPRAH, and the fact that routing protocols in the past have focused on maintaining routing entries for destinations, rather than source-destination pairs.

This paper introduces the Elliptic Demarcation of Information Transfer (EDIT) approach to routing in MANETs. EDIT consists of maintaining routing-table entries for source-destination pairs in a way that the dissemination of signaling packets is confined as much as possible within regions of interest, that are defined in a distributed manner by the distance that the node computing the region has, to both the source and destination of a given route. Unlike prior threshold schemes, a region of interest in EDIT is inherently an elliptical zone with a source and a destination as the foci of an ellipse, rather than an undirected ring-shaped zone as in prior threshold-based schemes based solely on distances away from destinations or sources of updates.

Section III shows that EDIT maintains loop-free routes within the regions of interest. Section IV presents the results of simulation experiments comparing EDIT and EDIT-GPS, where EDIT-GPS utilizes the notion of region of interest but uses GPS information to reduce the dissemination of control messages. We compare these against AODV, DSLR, OLSR and LAR. AODV and OLSR are representatives of on-demand and proactive routing protocols that are the most prominent in the research community. LAR is a very efficient location-aware routing protocol. DSLR [7] is a loop-free routing protocol that stands for Destination-controlled Source-sequenced Labeled

---

[1] Work partially sponsored by ARO under grants W911NF-04-1-0224 and W911NF-05-1-0246, by NSF under grant CNS-0435522, by DARPA through Air Force Research Laboratory (AFRL) Contract FA8750-07-C-0169, and by the Baskin Chair of Computer Engineering.

1-4244-2575-4/08/$20.00 ©2008 IEEE

Notation	Description
s, d, r, a	nodes
D^j_i	distance between nodes i and j
$D^j_{i,k}$	distance between j and i as seen by k
$D^{'}_{i,j}$	estimated distance for (i, j)
ϵ	shape constant
RREQ, RREP	Route Request, Reply
SSL	Source-controlled Sequence Labels
RSL	Relay-controlled Sequence Labels
DSL	Destination-controlled Sequence Labels
SUL	Sequenced Update Label
$sn(SSL^s_d)$	sequence number set by SSL^s_d
$sn(RSL^s_d)$	sequence number set by RSL^s_d
$sn(DSL^s_d)$	sequence number set by DSL^s_d
M_C	Mobility counter
$\epsilon^r_{s,d}$	Mobility threshold
Δ	rate constant for change of eccentricity
δ_n	next hope for node n
$ID(t)^a_{ab}$	denotes ID from SUL(a, id) at time t
$SID^a_d(t)$	denotes the start identifier for d at a at time t
$S_i(G)$	denotes the successor graph for node i
P_{ai}	path from a to i
Q_i	region covering node i
P^{*a}_d	shortest path between a, d

TABLE I
NOTATIONS USED IN THE PAPER

Routing, that uses the underlying DAG constructed with the help of signaling messages such as RREQs and RREPs (as in AODV) and is one of the first loop free routing algorithms to utilize Source-sequenced Labels to route correctly and efficiently. We use this frame work as our baseline for EDIT.

II. THE EDIT PROTOCOL

In EDIT, regions of interest are defined on the basis of source-destination pairs, rather than on the basis of destinations alone. Should no path exist between the two, the algorithm defaults to a ring-based route discovery. The basic idea behind this being that, once a route is established the algorithm tries to ensure that this path is maintained robustly against active topological changes. A node within the region of interest of a source-destination pair forwards signaling packets, for the pair, and nodes just outside the region do not. Nodes s, d, say the source and the destination, form the two foci of an ellipse defining the region of interest for the routing information pertaining to the path from s to d. A node, r, decides whether or not it belongs to a certain region of interest for the pair based on the relationship between the length of the known route between s and d and the distance between r, to s and to d.

Let D^i_j denote the distance from node i to node j. While the distance metric can be any arbitrary link weight, in this paper we focus on the case in which it is the hopcount. The region of interest for a source-destination pair (s, d) consists of all the nodes that satisfy the following condition, which defines an elliptical region of nodes around a possibly shortest route from s to d.

Definition *Elliptical Boundary Condition (EBC):* Node r is in the region of interest of the source-destination pair (s, d) if

$$D^r_d + D^r_s \leq D^s_d + \epsilon \tag{1}$$

ϵ in the above equation is defined as the *shape-constant* that determines the shape of the ellipse and the scope of additional nodes that are to be added into the region. ϵ values are analogous to the eccentricity of an ellipse. The eccentricity typically ranges from 0 (circle) to 1(parabola) and is defined as the ratio between the distance of each of the source, destination node to the midpoint between them and the distance between the midpoint and the maximal hop a message can travel across this path. The shape-constant describes the additional distance between the focii and the fringe of the ellipse, in other words the *linear eccentricity*. To use EBC, node r needs to maintain three distances for a given source-destination pair. In the following sections, we describe an implementation of EDIT based on source-sequenced labels for the maintenance of loop-free routes on-demand, and a combination of proactive and on-demand signaling taking advantage of EBC.

The routing mechanism in EDIT involves two phases, the discovery of routes, which is carried out as in traditional on-demand routing schemes, and the establishment of routes, which is done proactively within regions of interest. EBC is used to confine signaling packets within regions of interest.

1) Route Discovery: The source node broadcasts a route request (RREQ) with a source sequenced label (SSL) to uniquely identify the request and each RREQ and route reply (RREP) relayed by any node, carries the relay-sequenced label (RSL) which is maintained independently by the relaying node. The RREQ also typically contains the D^r_s (distance between the relay and the source) and carries the D^s_d last known distance between the source and destination.

2) Route Establishment: When the RREQ reaches a destination, the destination generates a RREP and broadcasts it reliably. This RREP contains the SSL, the RSL used in RREQ and also the distance between the relay and the destination (D^d_r). Broadcasting the RREP might seem inefficient at first sight but this is infact necessary to actively maintain the region of interest and deliver successfully under highly mobile scenarios. Using simulations and analysis we show how proactive updates, in these regions, manage to be efficient in-spite of additional dissemination.

A. Control Signaling

The RREQ sent by a node r contains: (SSL, RSL, d, TTL, (D^r_s), (D^r_d), $D^r_{s,d}$)

The SSL uniquely identifies the request by the identifier of the source originating the RREQ and a sequence number generated by the source. The propagation of such a RREQ results in the formation of a directed acyclic graph (DAG) rooted at the source and ordered by the SSL and the destination identifier. The RSL is significant only in the neighborhood of the node sending the RREQ, and it equals the value of the SSL of the node forwarding the RREQ. RREP packets, in addition to the fields mentioned in the RREQ, contain the distance information between the source and the destination, D^d_s. Node

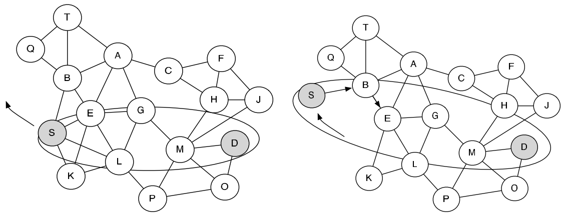

Fig. 1. (a) Source node moving away, (b) Source node broadcasts to neighbors causing the region to expand.

r accepts a RREP in response to a RREQ it originates or relays, regarding destination d, only if the RREP specifies the SSL and RSL that r used in its last RREQ for d.

The following five rules are used to search for routes to a destination d.

Rule 1: Node r increments $sn(RSL_d^r)$ and decrements the TTL of the RREQ each time it relays or originates a RREQ.

Rule 2: If node s requires a route to destination d, it issues a RREQ identified by SSL $(s, sn(SSL_d^s))$. At node s, the RSL and the SSL for d are identical.

Rule 3: If some node $r \neq d$ receives RREQ with SSL $(s, sn(SSL_d^s))$ from neighbor x, it caches the RSL in the RREQ $(x, sn(RSL_d^x))$. Node r processes a RREQ identified by an SSL only once. If it forwards a RREQ to its neighbors, it uses the same SSL created by the origin of the RREQ, s, and its own RSL $(r, sn(RSL_d^r))$.

Rule 4: When node d receives a RREQ from neighbor r that was issued by source s for d itself, it sends a RREP carrying the same SSL $(s, sn(SSL_d^s))$ of the RREQ, and the RSL $(r, sn(RSL_d^r))$.

Rule 5: When node r receives a RREP for destination d identified by SSL $(s, sn(SSL_d^s))$ and carrying RSL $(r, sn(RSL_d^r))$, the node can update its routing table (if it is feasible, ie., the update satisfies all other rules) to update its routing table. If it does so, then it must send a RREP to any neighbor x that sent a RREQ to r and for which r has cached an RSL noting the reception of a RREQ for d. ■

An update is similar to a RREQ packet but has an additional flag set, differentiating between the two messages. The following additional rules are used to ensure that the proactive signaling of the update messages correctly modify the routing tables:

Rule 6: When node r receives a RREQ or RREP for d that is valid (i.e., it has an SSL or RSL that is most recent), it updates its distance table with the appropriate information for d, and updates its routing table as needed.

Rule 7: When destination d issues an update for itself, it increments the DSL for d. No other node changes the DSL.

Rule 8: When node r receives an update message for d that is valid (i.e., it has a more recent SSL or DSL), it updates the distance table[1] and/or the information in the routing table, and sends an update of its own if its distance to d or s has changed. ■

B. Adaptive Regions of Interest

The regions of interest adapt to moving nodes by applying heuristics to distance information learned by the nodes in the region. The region of interest has a constant churn of nodes that move in and out of it at different rates. New nodes enter a region of interest, while some nodes no longer fall within the transient elliptical boundary, changing along with the SD-pair. Consider the topology in Figure 1. The nodes in Figure 1 establish a region of interest in which the destination sends out periodic updates within the region. Owing to the mobility of the nodes, the elliptical region self-organizes itself.

In the following subsections we describe some of the heuristics applied to make the regions of interest more adaptive to the changing scenarios.

1) Epsilon Learning: Epsilon (ϵ) is the shape-constant of the ellipse that determines the shape of the ellipse. As can be observed, the size of the region of interest is a function of the mobility of the nodes within the region. If the nodes in the region move faster, then a wider region would cover the paths between SD-pairs better. The rate of mobility is determined by each relay node using its one-hop neighbor information. At each hop, messages modify a counter, M_C (added to all or any of the packets based on required granularity), incrementing it if all its one-hop neighbors have changed and decrementing it if not. The source or the destination examine if the counter has exceeded a certain 'mobility threshold' that is a function of the hop count and the counter. For instance, a heuristic could be that if the counter value is less than half of the hop count of the path it has traveled, then the eccentricity is reduced to allow for a narrower region of dissemination. This allows the ellipse to grow or shrink based on the mobility of the nodes within the region.

2) Motion Determination Using Distance Information: We use a notion of trilateration, to approximate the motion of the node using distances(lengths) instead of points. Since all the information needed to determine is described by using hop-counts, we use trilateration to detect if a node is moving away from a region or towards it. It is important for nodes within the region of interest to have some opportunistic prediction based on previously learnt locations and expected mobility of the source-destination pair. Traditionally, prediction is dependent on a velocity vector of mobile nodes. However, given that EDIT does not use any location based service, we use geometric properties of triangles to determine the direction of motion within the region, while the velocity is determined by the rate of increase or decrease in the hop-count between successive updates.

Three possible cases that arise depending on the distances, between the source-destination-relay, which form a triangle. As it can be seen from Table II, if two out of three distances change, the node that is included between these two sides is undergoing a variation in distance. If the two distances increase then the node is moving farther away, while if both of them decrease the node is moving closer. The distance relation owing to the property of ellipses ensures that the increase in any one of the distances also affects at least

---

[1] distance table is defined in section II-E

D_s^d	D_s^r	D_d^r	**Inference**
increase	increase	-	s is moving away from the region. increase estimated D_s^d
increase	-	increase	d is moving away from the region. increase estimated D_s^d
-	increase	increase	Relay is moving away. Re-compute EBC
decrease	decrease	-	s is moving closer inside the region. reduce estimated D_s^d
-	inc/dec	-	s is moving. Change value of estimated D_s^d using last known D_d^r
-	-	inc/dec	d is moving. Change value of estimated D_s^d using last known D_s^r

TABLE II
TRILATERATION USING DISTANCE INFORMATION

one another. If it does not, then the elliptical region reorganizes. This information is used by the nodes lying within the region to determine if newer nodes are to be included into the forwarding zone while older nodes that no longer are a part of the forwarding zone stop relaying. Moreover, increase in distances is used to predict distance between source and destination, which determines the increase or decrease in size of the ellipse. An increase in D_d^s is always updated by messages propagated either by source or destination.

C. Information Maintained at Each Node

Each node maintains routing information about active source-destination pairs to carry out EDIT signaling. If any node lies within some k hops from the path between the source and the destination, for which it has seen a request or a reply within its timeout period, then it is said to be active for that SD-pair. The routing information at node r for the source-destination pair (s, d) consists of the following: (s, d, D_s^r, D_d^r, $D_{s,d}^r$, $sn(SSL_d^r)$, $sn(DSL_d^r)$, $sn(RSL_d^r)$, $\epsilon_{s,d}^r$, M_C).

Node r also maintains a *distance table* specifying the same routing information listed above but reported by each known neighbor. Node r also maintains a cache of the RSLs it has received in RREQs from each of its neighbors over the past few seconds.

III. PROTOCOL ANALYSIS

A. Loop Freedom

Given that update messages contain information that may result in a change in the successor relationship for any given destination, it is important to ensure that the updates also remain loop-free. The source-destination pair that triggers these updates enforces ordering using the DSL or SSL.

The *Sequenced Update Label (SUL)* SUL is identified by the pair *(node, ID)*, where the node is either a source or a destination, and the ID is a sequence number created by that node, which makes the SUL a globally unique identifier.

Theorem 1 in [7] shows that the RREQs and the RREPs remain loop-free if specific rules are enforced. The following theorem extends the proof in [7] to handle proactive signaling by means of SULs.

Theorem 3.1: If rules 1 to 8 are followed, then RREQs, RREPs and Update messages do not loop in an error-free network.

Proof: It is shown in [7] that RREQs and RREPs are loop free if rules 1 to 5 are followed. Therefore, it suffices to show that the update messages are also loop free if the added rules are also followed. The update messages traverse the DAG built similar to the RREQs . According to Rule 8, a node relays an update message only once. Therefore, the update message traverses a directed tree and hence any given update message identified by SUL(d, ID_d) traverses a path free of loops. ■

Each of the nodes also stores the SUL and the corresponding RSL for the SUL. Note that, while the update messages do not loop endlessly, it is possible that the update messages can trigger a loop by causing the node to change its successor for a given destination. This is prevented by enforcing a safe loop-free condition before accepting an update message as is required by *Rule 8*, where an update message is processed only if the update is feasible. The update is feasible only if the locally cached RSL for a SUL satisfies the loop-freedom condition.

1) Sufficient Condition for Loop freedom: We follow the same reasoning used in [7] to show that the DAG constructed by the update messages, similar to the RREQ-DAG, is loop free as long as every node in the region is engaged in only one route computation for a given source-destination pair. We use the following notation: $ID(t)_{db}^a$ denotes ID from the RSL*(a, id)* in the update for d sent by b to a.

Update Sequence-Number Condition (USC): Node a can change the routing table entry for the current successor to node b by processing an Update message if $ID_{db}^a(t) \geq SID_d^a(t)$ where $SID_d^a(t)$ is the start identifier used to check if a neighbor can be used as a loop-free successor. The $SID_d^a(t)$ is set to the last known value of $ID_d^a(t)$ which is denoted as $ID_d^{*a}(t)$.

Lemma 3.2: $SID_d^a(t_1) \leq SID_d^a(t_2)$, where $t_1 < t_2$.

Proof: If an update message does not cause a change in successor till some time t_2 then, $SID_d^a(t_1) = SID_d^a(t_2)$. An update is accepted only if $ID_d^a(t_2) > ID_d^a(t_1)$, which implies that $SID_d^a(t_2) = ID_d^{*a}(t_2)$. Therefore, $SID_d^a(t_2) > SID_d^a(t_1)$ and hence the Lemma is true. ■

Theorem 3.3: If nodes use USC to change successors, no routing table loops can form.

Proof: The proof is by contradiction. Assume that, before time t, the directed successor graph for destination d, which we denote by $S_d(G)$ is loop-free at every instant, and a loop is formed at time t. A loop can be formed only if some node changes its successor at time t to a node that is upstream of itself in $S_d(G)$.

Assume that the loop is formed when some node i makes node a its new successor $S_d^i(t)$ after processing the input event that caused it do so at time t, where $b = S_d^i(t_b) \neq a$ and $t_b < t$. Let $P_{ai}(t)$ consist of chain of nodes $a = s[1, new], s[2, new]..., s[k, new], ..i$. Now, $P_{ad}(t)$ must include $P_{ai}(t)$ which would mean that the path to the destination now contains the path added as a result of the new input after time t. To understand the notation better, $s[k, new]$ indicates that the node $s[k, new]$ is k hops away from the source in the path $P_{ai}(t)$ at time t and has node $s[k+1, new]$ as its successor at time t. The time at which the node $s[k, new]$ sets its successor $S_d^{s[k,new]} = s[k+1, new]$ is denoted by $t_{s[k+1,new]}$, where $t_{s[k+1,new]} \leq t$ which is the last time that the node updates its routing table. Therefore, $S_d^{s[k,new](t_{s[k+1,new]})} = S_d^{s[k,new]}(t)$. Given that any node joining the path P_{ad} does not change its successor after time t, it is true that

$$SID_d^{s[k,new]}(t_{s[k+1,new]}) = SID_d^{s[k,new]}(t) \tag{2}$$

Node $s[k, new]$ sends the last reply before time t to node s[k-1,new] and is denoted as $t_{s[k+1,old]}$. The successor of the node $s[k, new]$ is $s[k+1, old]$ at time $t_{s[k+1,old]}$. Note that $t_{s[k+1,old]} \leq t_{s[k+1,new]} \leq t$, and $s[k+1, old]$ need not be same as $s[k+1, new]$. It is also true that $SID_d^{s[k,new]}(t_{s[k+1,old]}) \leq SID_d^{s[k,new]}(t_{s[k+1,new]})$. From rules 6 thru 8 and Lemma 3.2, when a node relays an update message to node $s[k-1, new]$ at time $t_{s[k+1,old]}$ after some time t_k, it must be true that

$$id(s[k, new])_d^{s[k,new]}(t_k) < ID_d^{*s[k,new]}(t_{s[k+1,old]}) \tag{3}$$

We have that USC is satisfied when any node changes its successor, i.e., $s[k, new]$ makes $s[k+1, new]$ as its successor, where both the nodes are actually in the path P_{ad} at time $t_{s[k+1,new]}$. Therefore, it must be true that

$$\begin{aligned} id(s[k, new])_{ds[k+1,new]}^{s[k,new]}(t) = \\ id(s[k, new])_{ds[k+1,new]}^{s[k,new]}(t_{s[k+1,new]}) \geq \\ SID_d^{s[k,new]}(t_{s[k+1,new]}) \end{aligned} \tag{4}$$

Since a loop is formed after t, P_{ad} exists. We now derive the following inequalities along the path $P_{ai} \subset P_{ad}$ at time t, if nodes satisfy the USC when switching successors.

$$\begin{aligned} ID_d^{*s[k,new]}(t_{s[k+1,old]}) \leq ID_d^{*s[k,new]}(t_{s[k+1,new]}) \\ = SID_d^{s[k,new]}(t_{s[k+1,new]}) \leq id(s[k, new])_{ds[k,new]}^{s[k+1,new]}(t) \\ = id(s[k, new])_d^{s[k+1,new]}(t_{s[k+1,old]}) \\ \vdots \\ ID_d^{*i}(t_b) \leq ID_d^{*i}(t) = SID_d^i(t) \end{aligned} \tag{5}$$

The invariant conditions along this path lead to the erroneous conclusion that $SID_d^i(t) < SID_d^i(t)$. Therefore, no loops can be formed when USC is applied. ■

B. *Efficiency of Elliptical Regions of Interest*

An efficient search area for a mobile node, given its approximate location was shown to be an elliptical region in [8] with the center of this region located on the last known position of the node. Yosy, Panchapakesan [9] showed that the probability of a given independent observation falls within σ standard deviations of the ellipse computed from the χ^2 cumulative distribution, increases with higher standard deviations about the true position. Using the χ^2 location estimation problem and with $1-\sigma$ and $2-\sigma$ support planes they demonstrate that the area obtained using the projection of similarly described distribution is not circularly symmetric and the exact shape so obtained is an ellipse.

This observation leads to the intuition that a more efficient search area for randomly moving node in a ad-hoc network can be best described by the elliptical region of σ standard deviations from its true position. The search area for each node in the path between a source-destination pair will then be a series of elliptical regions around the approximate positions of each of these nodes. In the following theorem we show that an efficient search area for a given source-destination pair, within which proactive updates are propagated, is the ellipse covering the individual elliptical search regions of each of the nodes that are in the shortest path between the source-destination. We use the following terminology: Q_a, Q_b, Q_c, represent the Elliptical region surrounding the node a, b, c, respectively. Q denotes the covering ellipse. P_{da} represents a path between a and d. The symbol $\succ$ denotes the covering relation.

Theorem 3.4: Q is a efficient search region of dissemination for active flow between a and d, provided $Q \succ Q_n | n \in P_d^{*a}$

Proof: We know that Q_a, Q_b are the efficient search regions covering the nodes a and b separately. If both a and b are considered together, the search area including both the nodes can be described as $Q_{a,b} \succ (Q_a U Q_b)$. Since P_{ad} is the path between A and B, Q_{ab} covers the path P_{ad}. Construct $Q_{a,b}$ such that the midpoint of P_{ad} denoted as o, is the centre of the χ^2 distribution. The elliptical region surrounding o is an efficient search region by the result of [9]. Therefore, $Q_{a,b}$ is the search region covering nodes a and b.

Similarly, we can prove that $Q_{a,b,c,..z}$ is an efficient search region covering the path P_{az} where $Q_{a,b,c,...z} \succ (Q_a U Q_c U Q_c ... U Q_z)$. Hence, $Q \succ Q_n | n \in P_d^{*a}$ is an efficient search region of dissemination, where P_d^{*a} is the shortest path between A and D. ■

IV. PERFORMANCE EVALUATION

We carried out simulation experiments using the Qualnet simulator [10] to compare the performance of EDIT against that of AODV, OLSR, and LAR. Scenarios included, 100 nodes with 10,30 active flows for a duration of 900s over 10 random seeds in a 1500 sq.m area at a data rate of 1pps and random waypoint mobility. We use the following metrics in our comparison: (a) Control overhead ratio, which is the ratio of the control packets to the total data packets received; (b) latency, which is the delay measured end to end; and (c) the delivery ratio, which indicates the number of packets

Flows	10	30	10	30	10	30
Protocols	Delivery Ratio		Control Overhead		Latency	
AODV	0.988± 0.004	0.640± 0.040	1.823± 0.127	21.106± 3.518	0.04± 0.001	2.757± 0.484
DSLR	0.991± 0.050	0.689± 0.260	2.191± 0.355	15.801± 3.029	0.03± 0.002	2.495± 0.294
EDIT epsi	0.991± 0.002	0.710± 0.026	1.810± 0.212	13.238± 1.638	0.03± 0.005	2.183±0.292
EDIT gps	0.993± 0.002	0.727± 0.012	0.980± 0.400	11.610± 1.715	0.03± 0.002	1.520± 0.012
LAR	0.993± 0.004	0.726± 0.015	1.019± 0.516	11.960± 1.642	0.03± 0.004	1.570± 0.012
OLSR	0.861± 0.001	0.667± 0.047	7.633± 1.091	17.635± 3.518	0.1± 0.015	3.043± 0.277

TABLE III
PERFORMANCE AVERAGE OVER ALL PAUSE TIMES FOR 100 NODES

delivered per flow. The vertical bars in the graphs indicate the 95% confidence interval and the symbols indicate the mean value over various pause times. The simulation results show that EDIT performs better in conditions of high mobility. EDIT-GPS performs slightly better than EDIT, because of the intrinsic location based scoping. However, it must be noted that EDIT without GPS still performs quite close to EDIT-GPS and LAR, except for latency, where it takes longer to deliver packets than the GPS-based approaches.

As can be seen from the figures, the control overhead of such a scheme is comparable to schemes that utilize GPS or any other location based services, even under high mobility. This is typically owing to the local-maxima attained by the GPS based protocols and also owing to the bootstrapping problem of location to a node identifier in GPS based schemes. Also note in Fig 2 that the reactive routing protocols repeatedly flood the regions while the purely proactive schemes waste considerable bandwidth using excessive propagation of topology control messages. Combined with the collision issues, the larger control overhead correspondingly reduces the delivery ratio of the each of these protocols and we again find in Fig 2 that EDIT outperforms both the purely reactive and proactive schemes and performs as good as schemes with GPS. EDIT-GPS does not offer much more gain than EDIT or LAR does. This also gives us an insight into the fundamental problem of using GPS based device and the local maxima problem. However, DSLR fails to utilize any information about regions of interest and with increasing flows in the network, DSLR starts to perform perform poorly in spite of offering hop-by-hop loop freedom.

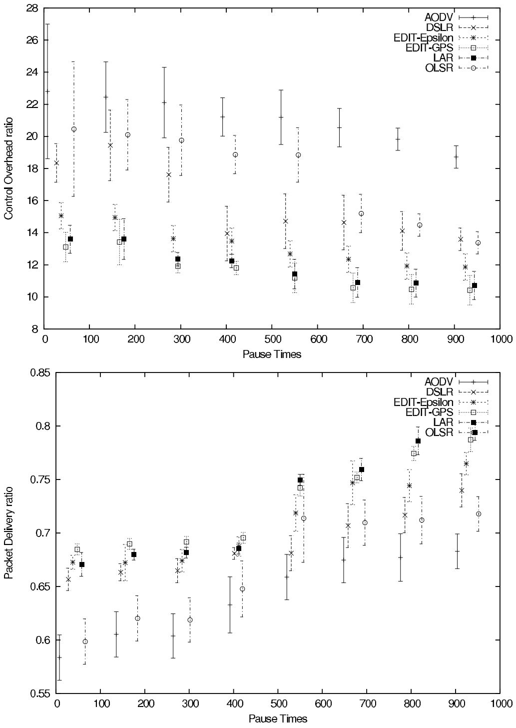

Fig. 2. Control Overhead, Delivery Ratio

V. CONCLUSION

We presented EDIT, a routing scheme based on on-demand establishment of routes and proactive maintenance of such routes using two main mechanisms: (a) source sequenced labels that ensiure loop freedom; and (b) the dynamic establishment of regions defined entirely by the distances to the source and destination of a route of interest. We proved the correctness of EDIT and used detailed simulations to show that EDIT is more efficient than traditional on-demand and proactive routing schemes, and that it is almost as efficient as routing schemes based on location information.

REFERENCES

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

[2] C. E. Perkins and E. M. Royer, "Ad-hoc on-demand distance vector routing," in *2nd /E Wksp. Mobile Comp. Sys. and Apps.*, Feb 1999, pp. 90–100.

[3] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)." United States: RFC Editor, 2003.

[4] R.E.Kahn, S.A.Gronemeyer, J.Burchfiel, and R.C.Kunzelman, "Advances in packet radio technology," in *Proc. IEEE*, vol. vol. 66. pp. 1468-1496, November 1978.

[5] G. Pei, M. Gerla, and T. Chen, "Fisheye state routing in mobile ad hoc networks," in *Proc. ICDCS Workshop on Wireless Networks and Mobile Computing*, 2000.

[6] C. Westphal, "Opportunistic routing in dynamic ad hoc networks: the oprah protocol," in *Proc. of IEEE MASS'06*, 2006.

[7] H. Rangarajan and J. Garcia-Luna-Aceves, "On-demand loop-free routing in ad hoc networks using source-sequence numbers," in *2nd IEEE International Conference on Mobile Ad-hoc and Sensor Systems (MASS)*, Nov 2005.

[8] T. Palsson, "Determination of search areas for maritime search and rescue operations," in *The Analytic Sciences Corporation*, Oct 1976.

[9] G. Yosy and S. Panchapakesan, "Comments on circular vs. elliptical errors in automatic location identification," in *Vehicular Technology Conference, 1998.*, May 1998.

[10] S. N. Technologies, "Qualnet simulator." in *http://www.scalablenetworks. com*, 2004.