On Optimizing Route Discovery of Topology-based On-demand Routing Protocols for Ad Hoc Networks

Boon-Chong Seet, Bu-Sung Lee, and Chiew-Tong Lau

Abstract: **One of the major issues in current on-demand routing protocols for ad hoc networks is the high resource consumed by route discovery traffic. In these protocols, flooding is typically used by the source to broadcast a route request (RREQ) packet in search of a route to the destination. Such network-wide flooding potentially disturbs many nodes unnecessarily by querying more nodes than is actually necessary, leading to rapid exhaustion of valuable network resources such as wireless bandwidth and battery power. In this paper, a simple optimization technique for efficient route discovery is proposed. The technique proposed herein is location-based and can be used in conjunction with the existing Location-Aided Routing (LAR) scheme to further reduce the route discovery overhead. A unique feature of our technique not found in LAR and most other protocols is the selective use of unicast instead of broadcast for route request/query transmission made possible by a novel reuse of routing and location information. We refer to this new optimization as the UNIQUE (UNIcast QUEry) technique[1]. This paper studies the efficacy of UNIQUE by applying it to the route discovery of the Dynamic Source Routing (DSR) protocol. In addition, a comparative study is made with a DSR protocol optimized with only LAR. The results show that UNIQUE could further reduce the overall routing overhead by as much as 58% under highly mobile conditions. With less congestion caused by routing traffic, the data packet delivery performance also improves in terms of end-to-end delay and the number of data packets successfully delivered to their destinations.**

Index Terms: **Ad hoc networks, on-demand routing, route discovery, flooding, optimization.**

I. INTRODUCTION

Mobile ad hoc network (MANET) is a rapidly deployable, multihop wireless network where nodes communicate with each other without relying on any preplaced fixed network infrastructure. In such networks, every node must act as a router to forward packets for other peer nodes. The nodes may move about freely, causing frequent and unpredictable topology changes. In the light of such non-deterministic node behavior, a central challenge to the design of MANET routing protocols is to devise an efficient algorithm to handle the frequent requests for new routes to be established.

The Internet Engineering Task Force (IETF) chartered the MANET Working Group [1] to develop standard IP routing support for ad hoc networks. Many routing protocols have been proposed. Most of the proposals can be divided into proactive (i.e., OLSR [2] and TBRPF [3]), reactive (i.e., DSR [4] and AODV [5]), and hybrid protocols (i.e., ZRP [6]). Proactive protocols in general rely on periodic or event-driven control packet updates to maintain information about the routing paths available in the network even if these paths are not currently used. A potential drawback with these protocols is that the maintenance of unused paths might use up a significant portion of the bandwidth if the network topology changes frequently.

On the other hand, reactive protocols maintain only the paths in use and the paths to initially unknown destinations are discovered only *on-demand*. Fewer control packets are therefore generated by route maintenance. Route discovery, however, in addition to adding an initial delay to packet delivery, may still generate a significant amount of control traffic. This is because flooding is typically used by the source to broadcast a route request (RREQ) packet in search of a route to the destination. Such network-wide flooding potentially disturbs many nodes unnecessarily by querying more nodes than is actually necessary, leading to rapid exhaustion of valuable network resources such as wireless bandwidth and battery power.

Hybrid protocols exhibit properties of both proactive and reactive protocols. Typically, proactive routing is used in a local zone while reactive routing is used between zones. These protocols can provide a better trade-off between routing overhead and delay, but this trade-off is subject to the size and dynamics of the zone. Adapting the zone size for optimal trade-off in a time-varying environment, i.e., where network mobility and density changes with time, is a challenging task to be overcome. The protocols described so far are known as *topology-based routing* protocols because they rely on information about links that exist in the network to make routing decisions.

Another class of protocols that rely *solely* on information about physical node position to perform packet forwarding is the so-called *position-based routing* protocols, i.e., GPSR [7] and DREAM [8]. The main prerequisite for routing with such protocols is that each node must know the position information of: i) its own; ii) the packet's destination; and iii) its 1-hop neighbors. GPS [9] or other techniques such as [10] are used for each node to obtain its own position. To learn the position of other nodes such as the destination, a *location service* (i.e.,

---

Manuscript received April 25, 2002; approved for publication by Suresh Subramaniam, Division III Editor, June 17, 2003.

B.-C. Seet is with the Network Technology Research Centre, Nanyang Technological University, Singapore, email: ebcseet@ntu.edu.sg.

B.-S. Lee and C.-T. Lau are with the School of Computer Engineering, Nanyang Technological University, Singapore, email: {ebslee,asctlau} @ntu.edu.sg.

This work was supported in part by the Singapore Research and Education Networks, under grant M47700003, and the Defense Science and Technology Agency, under grant P0D0001523.

[1] Earlier versions of this work appeared in [22, 23]. Our recent survey on route discovery optimization techniques can be found in [24].

1229-2370/03/$10.00 © 2003 KICS

GRID [11]) is used. Though position-based routing does not require the discovery and maintenance of routes, it does require the discovery and maintenance of node positions. Since its operations rely *solely* on position information, the timeliness and currency of such information become *very critical* to the performance of positional-based protocols. Furthermore, as with link updates in topology-based proactive protocols, the efficiency of location updates in position-based protocols is another issue that needs to be duly addressed. A survey on position-based routing can be found in [12], while a good reference on location updates by Stojmenovic can be found in [13].

This paper focuses on *topology-based on-demand routing* protocols. However, we have explored using physical node position (or location) information in a new optimization technique which we call the UNIQUE technique, to decrease the high overhead of flooding-based route discovery typical of these protocols. The proposed technique can be used in conjunction with the existing LAR scheme to further reduce the route discovery overhead. This paper studies the efficacy of UNIQUE by applying it to the route discovery of the DSR protocol. In addition, a comparative study is made with a DSR protocol optimized with only LAR.

The rest of the paper is organized as follows. Section II provides an overview of DSR. Section III reviews some existing work on optimizing route discovery. Section IV describes the UNIQUE technique. Section V explains the incorporation of UNIQUE and LAR into DSR. Performance evaluation by simulation is presented in Section VI and concluding remarks are made in Section VII.

II. REVISITING DSR

DSR [4] is a topology-based on-demand routing protocol for ad hoc networks based on the concept of source routing. With source routing, the source node first determines the complete routing path to the destination. The path information is then encapsulated into every packet that is meant for that destination. The main benefit of source routing over conventional hop-by-hop routing (i.e., table-driven protocols such as DSDV [14]) is that it frees the intermediate nodes from having to make routing decisions, because the complete path information is explicitly specified in the packet. The intermediate nodes simply examine the path information and forward the packet accordingly. Two major components in DSR protocol are: Route Discovery and Route Maintenance.

A. Route Discovery

Typically when a source node has a packet to send but without a route to the destination, it performs route discovery by flooding the network, i.e., querying every reachable node, with a Route Request (RREQ) message seeking a route to the destination. The source broadcasts the RREQ, which is then forwarded (re-broadcast) by other nodes until the RREQ arrives at the destination. Each intermediate node receiving this message for the first time appends its address to and rebroadcast the message to its neighbors. Once the RREQ reaches the destination, a Route Reply (RREP) message carrying the accumulated route, which holds the series of hops taken by the RREQ, is unicast to the source via the reversed path. The source then sends out the packet using this route, and caches the route information for subsequent delivery of packets to the same destination. Fig. 1 illustrates how the network-wide flooding of RREQ results in every node in the network to participate in the route discovery.

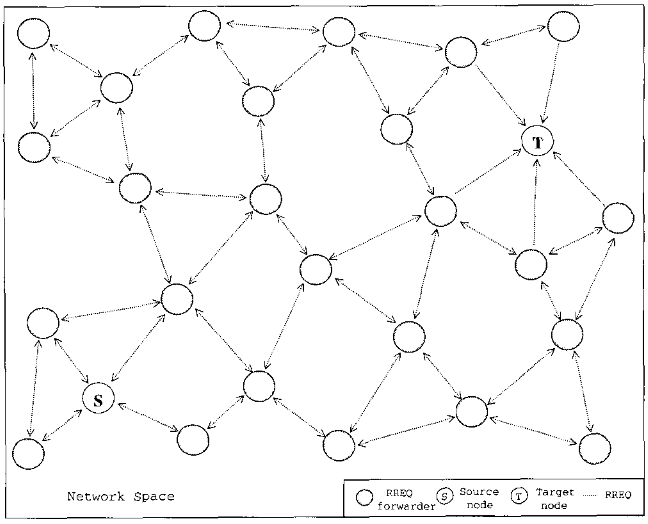


Fig. 1. DSR: Every node participates in the route discovery.

B. Route Maintenance

Route maintenance is used for detecting broken links in the routes in use, due to, i.e., changes in network topology. Each sending node on a route is responsible for sensing whether the link to its downstream neighbor is broken. If so, the node informs the source by sending a Route Error (RERR) message to indicate the nodes at both ends of the hop in error. The source then eliminates the erroneous hop from its cache by truncating all routes that contain this hop, and uses another cache route to send packets to destination. If no cache route is available, the source performs route discovery again to acquire a new route to the destination.

III. ROUTE DISCOVERY OPTIMIZATION

Given that bandwidth is a limited resource in ad hoc networks, it is essential to minimize the bandwidth consumed by non-data traffic. DSR attempts to decrease the overhead of route discovery by limiting the frequency and spread of route request messages through the following optimization techniques:

- ***Intermediate node cache replies***: If an intermediate node receives a RREQ for which it has a route, it may respond with a RREP instead of re-broadcasting the RREQ. This not only prevents the RREQ from flooding the network, but also reduces the latency of route discovery. However, the route acquired may be broken due to node movements, and therefore result in a routing error when packets attempt to use the route to reach their destination.
- ***Promiscuous route learning***: Nodes with their interfaces operating in promiscuous mode may eavesdrop on packets

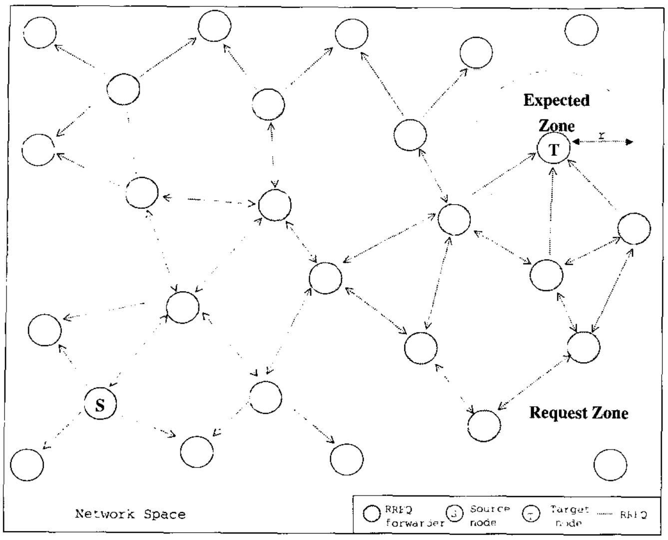


Fig. 2. LAR-1: Only nodes inside the request zone participate in the route discovery.

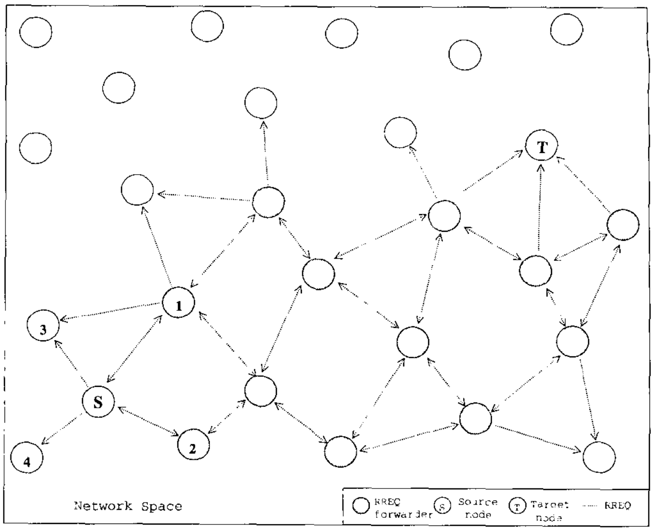


Fig. 3. LAR-2: Only nodes that can forward the RREQ closer to the target participate in the route discovery.

"passing-by," i.e., packets that are not addressed to them but could be overheard, and cache the route information they learned for their own future use. Nodes may thus perform fewer route discoveries as the routes needed to send their packets are already in their caches.

- ***Ring zero technique***: This optimization allows the source to query only its *1-hop* neighbors during the route discovery. If a neighbor that receives the RREQ is the destination, or knows a route to the destination, the neighbor replies to the source. The RREQ will be dropped by these nodes after it has been processed, and will never be rebroadcasted.

Besides the optimizations from the DSR protocol, there are a number of optimizations for route discovery proposed by other researchers (i.e., [15]–[18]). Two well-known optimization techniques in literature are presented in the following:

A. Query Localization

Castaneda and Das [15] proposed a query localization technique that utilizes prior routing histories for route discovery. The technique assumes that the topology has not changed drastically soon after a route failure, and therefore many nodes on the previous route are still useful for reconstructing a new route to the destination. Two heuristics are presented. The first assumes that the new route will not be very different from the previous route, at the most by k nodes different (path locality). The second assumes that the destination will be within k hops away from any nodes on the previous route (node locality). In both schemes, each query carries a counter that is initialized to zero, and then incremented each time it encounters a node that is not on the previous route. If it does encounter a node on the previous route, then only the second scheme will reset the counter to zero. Once the counter exceeds the threshold k, the query will be discarded. This technique is used for *route reconstruction*. For a route discovery with no previously known route to the destination, the query will be flooded throughout the network.

B. Location-Aided Routing

Ko and Vaidya [16] proposed the Location-Aided Routing (LAR) scheme. As with our UNIQUE technique, to be introduced later, this scheme uses the location of nodes from GPS to restrict the region of RREQ flood. Two LAR variants are proposed. Fig. 2 illustrates the concepts of LAR-1. From knowing physical location L and average speed v of the destination at time $t0$, the source can define at time $t1$, a circular region of radius $v(t1 - t0)$ known as the "expected zone," which is an area where the destination is expected to be found. The source further defines the smallest rectangle that includes both expected zone and itself as the "request zone," whose coordinates are sent along with the RREQ. Only nodes inside the request zone forward the RREQ. Others discard them when they are received.

In LAR-2, the source uses location information to compute its physical distance to the destination, and then attaches this distance value, and the destination location into the RREQ. An intermediate node that receives this RREQ computes its own distance to the destination, and compares with the distance value in the RREQ. If the node finds that it resides closer to the destination than the node from which it receives the RREQ, then it overwrites the previous distance value with its own value and forwards the RREQ. Otherwise, the RREQ is discarded. Fig. 3 illustrates the concept of LAR-2.

The RREQ broadcast by source S is received by nodes 1 to 4, but only nodes 1 and 2 forward the RREQ because they are closer to target T, than is from S to T. Hence, the RREQ only gets progressively closer to the target after each forwarding and not further away from it. In both variants, if the source does not receive a RREP after timeout, it resorts to flooding in a new attempt of route discovery for the same destination.

IV. UNIQUE

This paper proposes a simple new location-based optimization technique which can be used in conjunction with LAR to

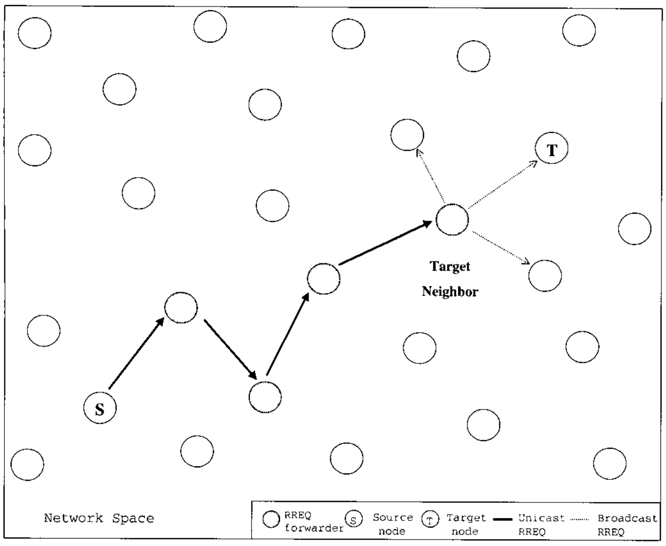


Fig. 4. UNIQUE: Fewer nodes participate in the route discovery.

further reduce the route discovery overhead. Our work is motivated by the observation that LAR would still incur a high overhead if the source and target of the route discovery are physically far from each other, leading to a large RREQ forwarding zone (flooding area) and renders LAR ineffective. The high overhead in the circumstance mentioned above provided us the impetus to propose the UNIQUE technique. An illustration of the proposed technique is provided in Fig. 4.

A. *Route Discovery with UNIQUE*

Notice from Fig. 4 that UNIQUE has far fewer nodes involved in the route discovery as compared to DSR and LAR shown in Figs. 1, 2, and 3 respectively. As an example, assume that source S is aware of target T's location, and the location of nodes to which it has a route. Then S will *unicast* (in contrast to broadcast) a RREQ using an existing route to some node in the vicinity of T. We refer to this node as *target neighbor* (TN) and the route used to send the RREQ as *unicast query path* (UQPath).

If the routing protocol, i.e., DSR, being optimized permits the RREQ to be replied by intermediate nodes with a route to the destination, then each intermediate node visited by the RREQ will check if it has a route to T. If so, the node returns a RREP to S and the route discovery is completed. Otherwise, the RREQ is forwarded to the next hop on the *UQPath*. This repeats until the RREQ reaches TN. If a route to T still cannot be found, TN forwards the RREQ to T as a broadcast message. Since the RREQ broadcast from TN is often closer to T than a RREQ broadcast from S (like LAR), the number of RREQ and corresponding RREP generated can be reduced. To further decrease the overhead, the broadcast from TN is directed towards T using a technique similar to LAR-2 but adapted for use by UNIQUE to within close vicinity of the destination.

In the event of a broken link, i.e., due to node mobility, on the *UQPath* when the RREQ is being forwarded towards TN, the forwarding node that detects the broken link will return a RERR to the source which then proceeds to select another *UQPath* for resending the RREQ. If no appropriate path is available, the source can use an alternative location-based technique such as LAR before resorting to flooding since the destination location is already known. Note that the type and number of alternative techniques being used is a matter of implementation choice and not part of the UNIQUE specification.

B. *UQPath Selection*

In a network with moderate to high node density, each node can be expected to have a sizeable number of neighbors, i.e., including the destination node. Intuitively, it is thus likely that the source will have a route to at least one neighbor node of the destination that can be used as the *UQPath*.

From its route cache (or routing table), source S determines whether a routing path to some node n can be used as the *UQPath*. This is done by finding and comparing the $DIST_{n-T}$ and $DIST_{S-T}$, which are distances to the target T from n and S respectively. The calculation of distance between any two nodes is based on their location coordinates. For instance, the distance between n and T can be found by: $\| DIST_{n-T} \| = \sqrt{(X_n - X_T)^2 + (Y_n - Y_T)^2}$, where (X_n, Y_n) and (X_T, Y_T) are n and T's locations respectively.

For n to be considered as a target neighbor TN, the condition $DIST_{n-T} \leq 0.5 DIST_{S-T}$ must be satisfied. The distance between n and T must be less than, or at most, equal to $1/2$ the distance between S and T. This is to ensure that n not only lies between S and T, but also lies closer to T than to S. The path to a TN with the least distance to T is selected as the *UQPath*. However, if more than one candidate path exists, i.e., two or more nodes with similar least distance to T, the path with minimum hops is chosen. If multiple paths have the same minimum-hop count, then one of the paths is arbitrarily selected.

C. *UNIQUE with Query Redirection*

Generally it can be expected that when the RREQ on the *UQPath* edges closer to the destination, more up-to-date information about the destination location will be learned. This information should be used to minimize the impact of any inaccurate location information about the destination known by the source. We thus propose an additional flexibility to allow the RREQ to be redirected by an intermediate node to the newer location when such information is available. For instance, before an intermediate node *n* forwards the RREQ to the next hop on the current *UQPath*, if *n* knows of more recent location of the destination, *n* can select a new *UQPath* for the RREQ, i.e., changing the next hop to which it will forward the RREQ, so that the RREQ will be redirected to this newer location. Selection of this new path is based on the newer destination location and the path selection criteria provided in Section IV-B.

V. PROTOCOL DESCRIPTION

In this paper, four protocols are being studied. They are: a simple flooding protocol (FLOOD), the DSR protocol, DSR protocol optimized with LAR (DSR-L), and DSR protocol optimized with both LAR and UNIQUE (DSR-LU). We use DSR as our base protocol for evaluating LAR and UNIQUE because

```
1. If S has UQPath to TN, then perform route discovery (RD) by
   UNIQUE and go to step 2. Otherwise, go to step 3.
2. If a route to T is found, END. Otherwise, go to step 6.
3. Perform RD by Ring-Zero.
4. If a route to T is found, END. Otherwise, go to step 5.
5. If S has location of T, then perform RD by LAR, else go to step 7.
6. If a route to T is found, END. Otherwise, go to step 7.
7. Perform RD by flooding. END.
```

Fig. 5. Algorithm for discovering a route in DSR-LU. Note that steps 3–6 also give the route discovery algorithm of DSR-L.

DSR is among the most efficient protocols in terms of routing packet overhead mainly due to several native optimization techniques it employs to improve its efficiency (see in Section III). A DSR protocol running with these optimizations thus provides a more challenging base than a simple flooding protocol for evaluating the efficacy of LAR and UNIQUE. In the following, we highlight the main features of route discovery in each of these protocols.

A. FLOOD

In this simple flooding protocol, the RREQ broadcast by the source will be rebroadcasted by every reachable non-destination node throughout the network. Only the destination node is permitted to reply to the RREQ.

B. DSR

As mentioned, the DSR protocol in this study operates with all its native optimizations, including route discovery optimizations such as intermediate cache replies, promiscuous route learning, and ring-zero technique. There are two stages to the route discovery. In the initial stage, the source performs a 1-hop route discovery using *Ring Zero* technique. If a RREP is obtained within a timeout period, the route discovery ends. Otherwise, the source initiates a network-wide broadcast of the RREQ.

C. DSR-L

DSR-L is the DSR protocol in Section V.-B incorporated with LAR. Note that we are only using LAR-2 since we found that it performed better than LAR-1 during our preliminary studies, which concurs with the results reported by Ko and Viadya [16].

In DSR-L, the LAR scheme is used when Ring-Zero technique is unsuccessful. If the follow-up attempt by LAR is still unable to discover a route to the destination, the RREQ will be broadcasted throughout the network.

DSR-L is incorporated with a piggyback mechanism proposed by LAR's authors in [16] for propagation of location information: Each node piggybacks its own location onto any packet it sends or forwards. Since DSR-L uses *source routing*, each packet will carry the location and address of node in the source route. A node receiving the packet can then learn the location of *every* other node through which the packet has visited. Note that the location information can be learned not only by nodes on the route but also by other nodes in the area when *promiscuous learning* is used. Initially, DSR-L performs route discovery in a way similar to DSR as the location of other nodes may not yet be known. However, as time progresses, more information about others' location will be available, resulting in more route discoveries to be performed using the LAR scheme.

Table 1. Simulation parameters.

Simulation Parameters	Settings
Mobility model	Random waypoint
Traffic model	20 CBR connections
Network space	1500m x 300m
Number of nodes	50 nodes
Maximum node speed	20 m/s
Packet sending rate	4 packets/s
Data payload	512 bytes

D. DSR-LU

DSR-LU features a combination of LAR and UNIQUE. Fig. 5 shows the algorithm used by DSR-LU to obtain a route. Notice that UNIQUE is used as the first choice of technique for route discovery instead of Ring-Zero. The reason is because in UNIQUE, only the next node on the *UQPath* will react to the RREQ, whereas in Ring-Zero, potentially all nodes within the radio range of the source will react, leading to a high routing packet overhead. UNIQUE can thus be more bandwidth efficient, especially in dense networks, and so will be used whenever possible over Ring-Zero.

Notice also that if a route cannot be found within the timeout period using UNIQUE, flooding will be performed instead of LAR. This design choice is made based on our observation that LAR usually does not perform well after an unsuccessful attempt by UNIQUE. The reason could be attributed to inaccurate location information. Flooding is still performed as a result of unsuccessful attempts by LAR. If flooding is performed immediately after UNIQUE, we might reduce not only the packet delay, but also the routing overhead as a result of less overlapping search areas.

VI. PERFORMANCE EVALUATION

This paper evaluates the performance of the four protocols under study using the *ns*-2 [19] network simulator with MONARCH wireless extensions [20]. Table 1 gives a summary of the simulation parameters used in this study. Note that these are the same parameters used in literature to evaluate many ad hoc routing protocols.

Movement scenarios are generated based on random waypoint model [21] characterized by a pause time, which is varied between 0 and 900 seconds to effect different relative speeds between the nodes. A 0 second pause time corresponds to continuous motion, while a 900 seconds pause time, which is equivalent to the total time of simulation, corresponds to no motion.

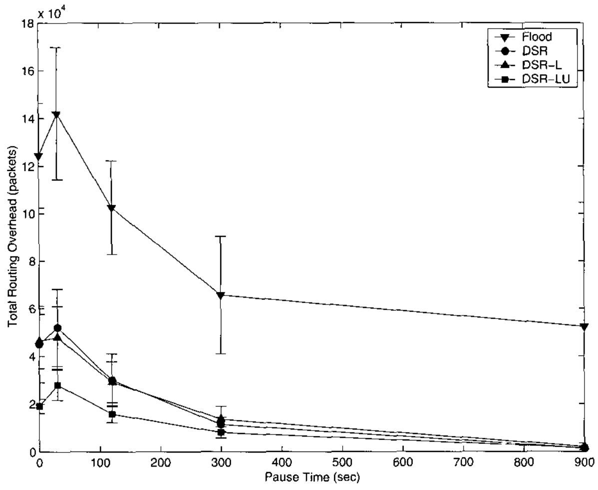


Fig. 6. Total routing overhead.

We simulate each pause time with 10 movement scenarios, and present the mean of each performance metric (with their 95% confidence interval) over these 10 runs. The metrics we use in this study are:

- ***Total routing overhead*** - The sum of routing packets generated by each node, including Route Request (RREQ), Route Reply (RREP), and Route Error (RERR) packets. Each hop-wise transmission is counted as one transmission.
- ***End-to-end delay*** - The average time from originating a data packet at the source to delivering the packet to its final destination. This includes all possible delays such as route discovery latency, queuing delay at interface queue, propagation, and retransmission delays in MAC and physical layers.
- ***Path optimality*** - The difference in hop-length between the path data packet took to reach its final destination and the shortest path that physically existed at the time of transmission.
- ***Packet delivery ratio*** - The ratio of number of data packets successfully delivered to the destinations to the number of data packets originated by the sources.

A. Total Routing Overhead

Fig. 6 shows the sum of routing packets generated by each protocol as a function of pause time (node mobility). In all protocols, the routing packet decreases with mobility or increases when pause time is greater than 0s. This is generally expected since lower mobility increases the stability of the routes, and consequently requires fewer route reconstructions once they are established. Also expectedly, FLOOD generates the most routing packets among the protocols. DSR, with its native optimizations, significantly reduces the routing overhead of FLOOD by at least three fold.

Somewhat unexpected is the result of DSR-L, which is not much better than DSR, despite the incorporation of LAR. The reason is because DSR's optimization techniques have been effective in reducing the frequency and spread of flooding, and therefore limits the potential contribution of the LAR scheme. For example, some flooding can be averted when using the Ring-Zero technique. DSR's promiscuous listening enables nodes to perform fewer route discoveries, and increases their responsiveness to route requests from other nodes, thereby curbing the spread of these broadcast messages. Thus, while LAR certainly has advantages over a simple flooding protocol, it may not have much to offer when used with a sophistically optimized protocol such as DSR. Fig. 7 shows the composition of routing packets generated by each protocol when pause time = 0s.

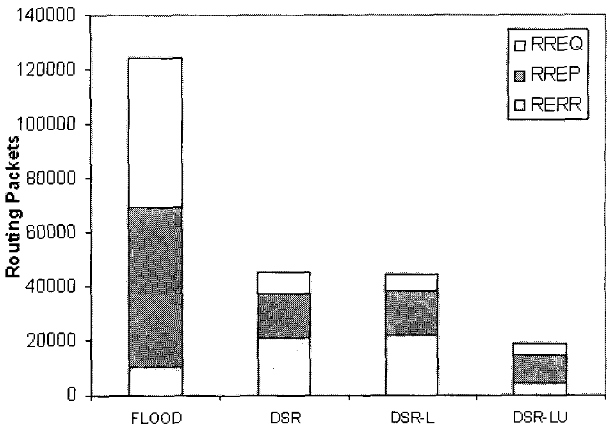


Fig. 7. Composition of routing packets.

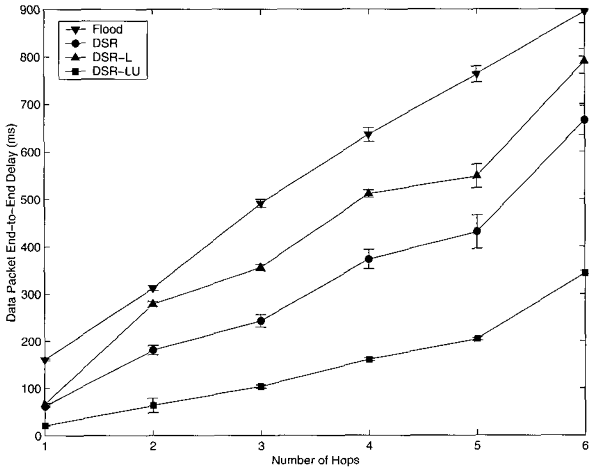


Fig. 8. End-to-end delay as a function of hops.

With UNIQUE, the overhead of DSR can be further reduced, as illustrated by DSR-LU. For instance, with highest mobility (pause time = 0s), DSR-LU's routing overhead is further reduced by 57.5% (w.r.t. DSR). In terms of individual packet types, RREQ is reduced by 39.3%, RREP by 39.7% and RERR by 78.8%. The margin of reduction in RERR is probably most interesting and can be attributed to fewer invalid cached routes used for *data packet* delivery. Recall that in DSR, except for routes that are in use, other routes acquired and cached are *not maintained.* If these routes are used to deliver data packets at a later point in time, and if node mobility is sufficiently high, the possibility of encountering a broken link on these routes can be

Table 2. Average end-to-end delay.

	Delay (ms)
DSR	212.77
DSR-L	270.91
DSR-LU	98.60
FLOOD	437.91

Table 3. Average extra hops incurred per data packet.

	No. of hops incurred
DSR	0.200
DSR-L	0.201
DSR-LU	0.248
FLOOD	0.464

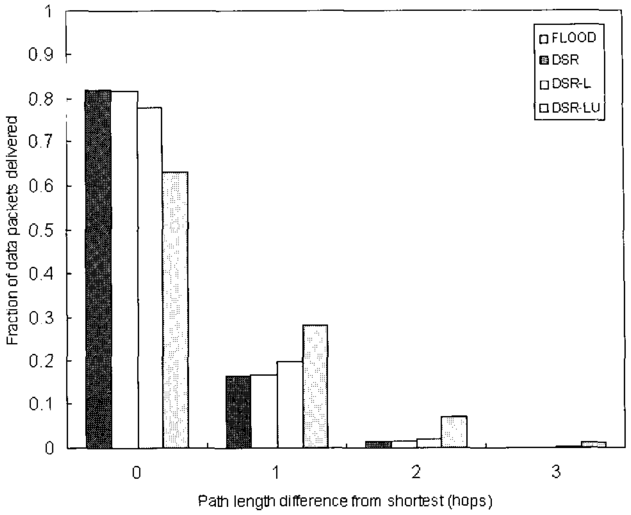


Fig. 9. Path optimality.

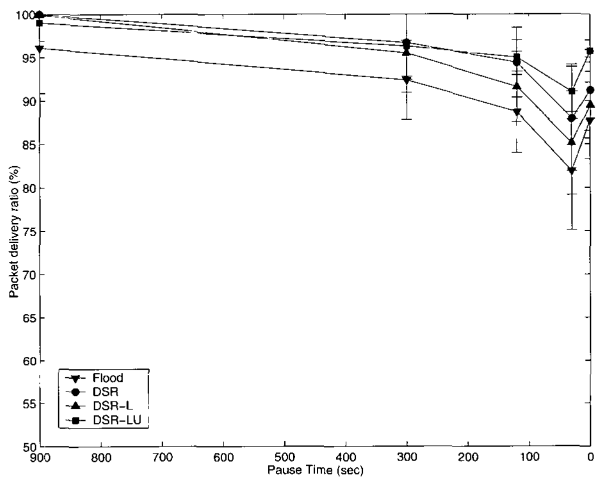


Fig. 10. Packet delivery ratio.

non-negligible. With UNIQUE, however, the cached routes have been better maintained. The reason is because the RREQ sent by UNIQUE through a cached route also acts like a *probe packet*, verifying the connectivity of each link on the route it travels. Thus, some invalid cached routes can be discovered early and removed from the nodes' caches before they can be used by data packets to reach their destinations, causing subsequent routing errors.

B. End-to-End Delay

Fig. 8 shows the end-to-end delay as a function of hops. A data point represents the average delay a packet incurs for traveling a certain hop count to reach its destination. The data path length on average (in hops) is about 3. Table 2 further shows the overall average end-to-end delay. Results are shown for the worst-case scenario at highest mobility (pause time = 0s). Under this scenario, the data packets experience their highest delays dominated by network congestion due to excessive routing traffic. FLOOD gives the worst delay performance as expected, with the delay increasing at a rate faster than other protocols with increasing number of hops to the destination. On the other hand, the lower routing load arising from the use of UNIQUE has allowed DSR-LU to deliver packets to their destinations in the shortest time, i.e., fewer routing packets reduces channel contention, while less backlog in interface queue accelerates packet delivery. From Table 2, the average end-to-end delay incurred in DSR-LU is 53.7% and 77.5% lower than DSR and FLOOD respectively. Between DSR and DSR-L, the latter exhibits a higher delay when packets are required to travel more hops to reach their destinations. This result is not surprising. Recall from Fig. 6, DSR-L has a similar routing load as DSR, and thus experiences a similar level of congestion. However, its packet size is larger due to the location information it carries, which increases as more hops are traveled. Consequently, packets queuing for transmission by the node must wait longer since the backlog in the interface queue will drain away more slowly.

C. Path Optimality

Fig. 9 shows the difference between the actual number of hops taken and optimal number of hops required by *data packets* to reach their destinations. A difference of 0 means the packets have taken an optimal (shortest) path, whereas a difference > 0, i.e., 1, 2, 3, etc., indicates the number of extra hops taken by the packets.

Based on the results in Fig. 9, Table 3 further shows the average extra hops incurred per packet found using the expression: $\sum(F_n * n)/\sum F_n$ where n is the number of extra hops incurred (path length difference from shortest) and F_n is the fraction of packets incurring n hops. Note that only packets within $0 \leq n \leq 3$ are considered. Packets with $n > 3$ can be neglected without loss of generality since their numbers are comparatively insignificant.

From Table 3, it can be observed that with the exception of FLOOD, all other protocols have quite similar path optimality. The performance of DSR-LU is better than expected. Although the UNIQUE in DSR-LU selects the shortest existing path for route discovery, the final path that it discovers and uses for packet delivery might not be the shortest available in the network. However, due to the intelligent *route shortening* feature of the underlying DSR protocol (one of the DSR's route maintenance optimization technique), the initially less-than-optimal path can become optimal over time.

The result of FLOOD is a little surprising as one would expect it to give the best path optimality since it searches the entire network for all possible paths. However, note that the shortest path discovered is only shortest *at the time of discovery*. As topology may change with time and/or mobility, the path that is initially optimal may no longer be optimal. Since this simple protocol does not incorporate any optimizations, including DSR's route shortening feature, even though a shorter path may exist, all data packets will continue to go through the path discovered initially.

D. Packet Delivery Ratio

Fig. 10 shows the success of delivering data packets to their destinations. In general, the fraction of data packets successfully delivered during low mobility approaches close to 100%. At higher mobility, packet delivery deteriorates mainly because of two reasons: i) more packets are dropped in transit due to route disconnections, i.e., a packet can be dropped by an intermediate node when the primary route fails and no other routes exist between itself and the destination; ii) packets are also dropped when the interface queue becomes full as a result of more routing traffic introduced by the increase in route discoveries. For protocols with low routing load such as DSR-LU, the impact from the latter is less compared to other protocols and consequently leads to better packet delivery performance.

VII. CONCLUSION

This paper proposes the UNIQUE optimization technique for route discovery of topology-based on-demand routing protocols. The key idea is to send the route request as a *unicast* message using an existing route that will bring the route request closer to the destination with the aid of location information. The proposed technique can be used in conjunction with the existing LAR optimization technique to allow even better efficient use of bandwidth by reducing significantly the routing traffic due to route discovery performed typically by flooding.

A comparison study is made with LAR using the DSR as the base routing protocol. An important finding from the study is that LAR while certainly could benefit a simple flooding protocol in terms of overhead reduction, it may not be of much benefit to a fully-optimized DSR protocol for reducing its overhead further. DSR's own optimization techniques such as Ring Zero and promiscuous route learning are found effective in curbing the frequency and spread of flooding in route discovery. Consequently, the contribution of LAR to DSR is only marginal.

However, when UNIQUE is used in conjunction with LAR, the overhead of DSR is further reduced significantly. Other performance metrics used such as packet delivery ratio and end-to-end delay have also shown improvements as a result of decreased network congestion due to routing traffic. Although UNIQUE is studied using DSR in this paper, it can also be used by other topology-based on-demand routing protocols to achieve better efficiency in route discovery.

REFERENCES

[1] IETF MANET Working Group, http://www.ietf.org/html.charters/manet-charter.html.

[2] T. Clausen and P. Jacquet, "Optimized link state routing protocol," *Internet Draft*, draft-ietf-manet-olsr-10.txt, May 2003.

[3] R. Ogier , M. Lewis, and F. Templin, "Topology dissemination based on reverse path forwarding," *Internet Draft*, draft-ietf-manet-tbrpf-08.txt, Apr. 2003.

[4] B. Johnson, D. A. Maltz, and Y.-C. Hu, "The dynamic source routing protocol for mobile ad hoc networks," *Internet Draft*, draft-ietf-manet-dsr-09-txt, Apr. 2003.

[5] E Perkins, E. M Royer, and S. R Das, "Ad hoc on-demand distance vector routing," *Internet Draft*, draft-ietf-manet-aodv-13.txt, Feb. 2003.

[6] Z. Haas, M. Pearlman, and P. Samar, "The zone routing protocol," *Internet Draft*, draft-ietf-manet-zrp-04.txt, Jul. 2002.

[7] B. Karp and H. T. Kung, "Greedy perimeter stateless routing for wireless networks," in *Proc. ACM/IEEE Mobicom'00*, Boston, MA, Aug. 2000, pp. 243–254.

[8] S. Basagni *et al.*, "A distance routing effect algorithm for mobility (DREAM)," in *Proc. ACM/IEEE Mobicom'98*, Dallas, TX, USA, 1998, pp. 76–84.

[9] G. Dommety and R. Jain, "Potential networking applications of global positioning systems (GPS)," *Technical Report TR-24*, Computer Science Department, The Ohio State University, Apr. 1996.

[10] S. Capkun, M. Hamdi, and J. P. Hubaux, "GPS-free positioning in mobile ad-hoc networks," *J. Cluster Computing*, vol. 5, no. 2, Apr. 2002.

[11] J. Li *et al.*, "A scalable location service for geographic ad hoc routing," in *Proc. ACM/IEEE Mobicom'00*, Boston, MA, 2000, pp. 120–130.

[12] Mauve, J. Widmer, and H. Hartenstein, "A survey on position-based routing in mobile ad hoc networks," *IEEE Network Magazine*, pp. 30–39, Nov. 2001.

[13] Stojemnovic, "Location updates for efficient routing in ad hoc wireless networks," *Handbook of Wireless Networks and Mobile Computing*, John Wiley and Sons, pp. 451–471, Jan. 2002.

[14] C. Perkins and P. Bhagwat, "Highly dynamic destination sequenced distance-vector routing for mobile computers," *Computer Communications Review*, pp. 234–244, Oct. 1994.

[15] R. Castaneda and S. R. Das, "Query localization techniques for on-demand routing protocols in ad hoc networks," *ACM/Kluwer J. of Wireless Networks*, vol. 8, no. 2, pp. 137–151, 2002.

[16] Y. Ko and N. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," *ACM/Kluwer J. of Wireless Networks*, vol. 6, no. 4, pp. 307–321, 2000.

[17] J. Sucec and I. Marsic, "An application of parameter estimation to route discovery by on-demand routing protocol," in *Proc. Distributed Computing Systems*, Mesa, AZ, Apr. 2001.

[18] J. Li and P. Mohapatra, "PANDA: An approach to improve flooding based route discovery in mobile ad hoc networks," *Technical Report CS5-2002-29*, Computer Science Department, UC Davis.

[19] Network Simulator (ns-2), http://www.isi.edu/nsnam/ns/.

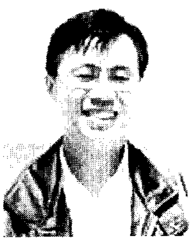
[20] MONARCH Wireless Extensions to ns-2, http://www.monarch.cs.rice.edu/cmu-ns.html.

[21] B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, Chapter 5, pp. 153–191, Kluwer Academic Publishers, 1996.

[22] B.-C. Seet, B.-S. Lee, and C.-T. Lau, "Route discovery optimisation for dynamic source routing in mobile ad hoc networks," *IEE Elec. Lett.*, vol. 36, no. 23, pp. 1963–1964, 2000.

[23] B.-C. Seet, B.-S. Lee, and C.-T. Lau, "Study of a unicast query mechanism for dynamic source routing in mobile ad hoc networks," *Lecture Notes on Computer Science (LNCS)*, Springer-Verlag, Berlin, vol. 2094, pp. 168–176, 2001.

[24] B.-C. Seet, B.-S. Lee, and C.-T. Lau, "Route discovery optimization techniques in ad hoc networks," *Handbook of Ad Hoc Wireless Networks*, ed. by M. Ilyas, Chapter 17, CRC Press, Dec. 2002.



Boon-Chong Seet received his B.Eng. degree from School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore in 1999. He is presently a technical member of Network Technology and Research Centre, and is working towards his Ph.D in School of Computer Engineering, NTU. His current research interests are in routing and mobility management for ad hoc wireless networks.

Bu-Sung Lee received his B.Sc. and Ph.D from Electrical and Electronics Department, Loughborough University of Technology, UK in 1982 and 1987 respectively. He is a member of Asia Pacific Advance Network (APAN) and an Associate of Singapore Research and Education Networks (SingAREN). He is also an active member of several national standards organization such as the National BioMedical Grid Task Force. Presently, he is an Associate Professor and Head of Division of Computer Communications in the School of Computer Engineering, Nanyang Technological University, Singapore. His current research interests include network protocol and management, broadband networks, distributed systems and in particular Grid Computing.

Chiew-Tong Lau received received his B.Eng. degree from Lakehead University in 1983, and M.A.Sc and Ph.D. degrees in Electrical Engineering from the University of British Columbia in 1985 and 1990 respectively. He is currently an Associate Professor and Head of Division of Computing Systems in the School of Computer Engineering, Nanyang Technological University, Singapore. His main research interests are in wireless communications.