Scalability Study of the Ad hoc On-Demand Distance Vector Routing Protocol

Sung-Ju Lee, Elizabeth M. Belding-Royer, and Charles E. Perkins

Abstract—**As mobile networking continues to experience increasing popularity, the need to connect large numbers of wireless devices will become more prevalent. Many recent proposals for ad hoc routing have certain characteristics which may limit their scalability to large networks. This paper examines five different combinations of modifications which may be incorporated into virtually any on-demand protocol in order to improve its scalability. The scalability of current on-demand routing protocols is evaluated through the selection of a representative from this class of protocols. The performance of the un-modified on-demand protocol is compared against that of it combined with each of the scalability modifications. Each scheme's behavior is analyzed in networks as large as 10,000 nodes through detailed simulation. Based on the observations, conclusions are drawn as to the expected scalability improvement which can be achieved by each modification.**

I. INTRODUCTION

RECENT advances in the portability, power, and capabilities of wireless devices and applications have resulted in the proliferation and increased popularity of these devices. As the number of users continues to grow, wireless routing protocols will be required to scale to increasingly larger populations of nodes. Conference networking scenarios can require the formation of networks on the order of tens to hundreds of nodes, while many military applications can involve thousands to tens of thousands of nodes. Furthermore, as the deployment of wireless networks becomes more widespread, new applications may encourage the formation of large ad hoc networks. For instance, sensor networks may include thousands of sensors which must be able to self-configure and establish routes. Similarly, military battlefield operations often require the formation of ad hoc networks containing hundreds to thousands of soldiers and personnel.

There have been many recent proposals of unicast routing protocols for ad hoc mobile networks [1], [15], [24], [25], [28], [35], [37]. Many of these publications include simulations of the protocols they describe, illustrating the performance of the protocol. To determine the relative merits and strengths of the various protocols, studies have been performed which simulate the protocols under various input conditions [5], [8], [14], [19]. While these simulations and studies are informative in evaluating the performance of the protocols for relatively small numbers of nodes (i.e., 50 nodes), they do not show how any of the protocols scales to larger node populations. The simulations presented in [28] and [25] evaluate the Ad hoc On-Demand Distance Vector (AODV) Routing protocol and the Zone Routing Protocol (ZRP), respectively, for networks as large as 1,000 nodes, and are simulations of some of the largest network sizes to date. Because ad hoc routing protocols could be used in networks containing a large number of nodes, it is important to know how these protocols will scale and perform in these scenarios.

Many of the proposed protocols for ad hoc networks [1], [15], [24], [28], [37] use a broadcast route discovery mechanism whereby a route request is flooded across the entire network. While the impact of such route discovery floods may be limited in small networks, the impact will be significantly greater for larger networks. When a link break in an active route occurs, many of these protocols [15], [24], [28] require that an error notification be sent to nodes that were using that link. Again, for small networks with limited network diameters, this route error message can be propagated back to a source node relatively quickly, and some repair action can be taken. However, as the network diameter and average path length increase, the error message may have to propagate across tens of hops to reach the source node. For such large networks, or even smaller networks with rapidly moving nodes, it is likely that the source node will be unable to make a repair before another link in the route breaks. Hence, this mechanism may prove ineffective for more stressful scenarios.

There are other approaches to unicast routing in ad hoc mobile networks than those previously described which may prove more scalable; however, each of these methods also has its limitations. Clustering and hierarchical ad-

---

Sung-Ju Lee is with the Internet Systems & Storage Labs, Hewlett-Packard Laboratories, Palo Alto, CA 94304. E-mail:sjlee@hpl.hp.com.

Elizabeth M. Belding-Royer is with the Department of Computer Science, University of California, Santa Barbara, CA 93106. E-mail:ebelding@cs.ucsb.edu.

Charles E. Perkins is with the Communications System Laboratory, Nokia Research Center, Mountain View, CA 94043. E-mail:charliep@iprg.nokia.com.

dressing methods have long been known for attempting to increase protocol scalability [3], [7], [10], [13], [18], [20], [30], [34]. Clustering protocols group nodes into *clusters* based on their proximity to each other. Each cluster generally has a cluster leader, which is the representative of the nodes in its cluster. The cluster leader typically participates in the network routing protocol, freeing the other network nodes of this burden. Routes in clustered networks may often be recorded hierarchically between clusters, as in [7]. These logical hierarchical paths may be longer lived than routes which utilize flat addressing, because any gateway between two clusters can be used to route between the clusters. This may result in fewer route reconstructions, and hence also reduce the number of on-demand control messages required to maintain the routes. Cluster-based protocols, however, have their drawbacks. They require periodic messaging from each network node in order to maintain the clusters. This periodic messaging results in higher processing and control packet overhead, as well as increased bandwidth utilization and longer delays. Moreover, if the protocol constrains routes to traverse cluster leaders, longer path lengths will be required. This is another contributor to increased bandwidth utilization. Finally, there may be complications when the cluster leaders fail or give up their cluster leader status.

Instead of performing routing on-demand, other protocols have instead been based on modified versions of either the distance-vector [21] or link-state [22] routing algorithms. Because both distance-vector and link-state algorithms not only use periodic updates, but also triggered updates in the event of a change in link status, they are not well-suited for mobile networks. A network composed of moderately moving nodes results in a high number of triggered updates, consuming bandwidth and making route convergence difficult, if not impossible. The protocols described in [4], [9], [16], [23], [26], [27], [36] each present a modified version of one of these protocols. For instance, [16], [26] and [36] each utilize a prioritized connectivity information exchange algorithm, whereby information about parts of the network more distant from the sending node is sent less frequently than information about neighboring nodes. [16] and [36] apply this technique to the distance-vector algorithm, while [26] modifies the link-state protocol. A similar approach [4] sends update information to only those nodes that actually need the information. These protocols have the benefit of a reduction in routing update overhead as compared with the basic link-state and distance-vector algorithms. However, they still have the drawback that they require updates based on node movement, which can result in a large amount of control overhead and bandwidth consumption in a mobile network.

A different approach to route finding is taken by the Core Extraction Distributed Ad Hoc Routing (CEDAR) algorithm [35]. CEDAR is an algorithm that builds a set of nodes (i.e., a *core*) to perform route computation. Using the local state information, a minimum dominating set of the network is approximated to form the core. CEDAR establishes QoS routes that satisfy bandwidth requirements using the directionality of the core path. Link state and bandwidth availability is exchanged to maintain important information for computing QoS routes. Although CEDAR builds a core infrastructure that yields low overhead, the protocol is fairly complex and difficult to implement. The problem of calculating the minimum dominating set and the core is known to be NP-hard [11].

Finally, [1] and [25] are variations of on-demand routing protocols which attempt to increase scalability through other methods. The Relative Distance Micro-discovery Ad Hoc Routing (RDMAR) protocol allows for local repair of link breaks in active routes [1], and the ZRP protocol maintains route information to *all* nodes within a “zone” [25]. ZRP is a hybrid protocol which maintains the route information for the zone via a link-state or distance-vector protocol and then applies the on-demand technique communication for nodes outside the zone. These protocols may reduce the number of route discovery floods required by a source node by either repairing link breaks locally where they occur (RDMAR) or by maintaining routes to some destinations before they are actually needed (ZRP). Nevertheless, the protocols still suffer from the same disadvantage as the class of on-demand protocols whereby efficiency drops as the number of source-destination pairs increases, due to the likely requirement of a route discovery flood.

This paper evaluates the scaling potential of on-demand ad hoc routing protocols by comparing a base routing protocol with the performance of it combined with various modifications. The Ad hoc On-Demand Distance Vector (AODV) Routing protocol [28], [29] is used as a representative of on-demand routing protocols. AODV was chosen because it is currently one of the leading protocols for routing in ad hoc mobile networks. The scalability of AODV is investigated by evaluating its performance in networks as large as 10,000 nodes. Then, three methods for improving the scalability of ad hoc routing protocols are described and integrated into the AODV protocol for their evaluation. The modifications include an expanding ring search for route discoveries initiated by a source node, a query localization protocol (proposed in [6]) which also attempts to prevent the flooding of route requests, and the local repair of link breaks in active routes. Further, the methods for preventing discovery floods are each in turn combined with the local repair mechanism, to yield a total of five possible improvement algorithms. Each of these modification combinations is evaluated, through detailed simulation, in networks of up to 10,000 nodes, and compared with the results achieved by the un-modified AODV protocol. The purpose of this paper is to study the routing behavior of on-demand routing protocols in large scale networks, and investigate how enhancement strategies affect the performance. The contribution of this work is the analysis of the scalability characteristics of the AODV routing protocol, the addressing of the possible problems of on-demand routing in large networks, and the presentation of results and insights that suggest future directions of research for scalable ad hoc routing protocols.

The rest of the paper is organized as follows. Section II presents an overview of the AODV routing protocol. Section III then describes the proposed modifications to the protocol to improve its scalability. Then, Section IV describes the simulations used to test the modifications, as well as presents the results achieved by these simulations. Section V details observations which can be drawn from the simulations, and finally, Section VI concludes the paper.

II. OVERVIEW OF THE ROUTING PROTOCOL

The routing protocol utilized for the scalability study is the Ad hoc On-Demand Distance Vector (AODV) protocol [28], [33]. AODV is an on-demand protocol which is capable of providing unicast, multicast, and broadcast communication. For the purposes of this study, its unicast operation is focused upon. Route discovery is based on a route request/route reply query cycle. Once discovered, a route is maintained as long as needed by the source. To guarantee loop freedom, AODV utilizes per node sequence numbers. A node increments the value of its sequence number whenever there is a change in its local connectivity information.

A. Route Discovery

Route discovery begins when a source node needs a route to some destination. It places the destination IP address and last known sequence number for that destination, as well as its own IP address and current sequence number, into a Route Request (RREQ). It then broadcasts the RREQ and sets a timer to wait for a reply.

When a node receives the RREQ, it first creates a *reverse route entry* for the source node in its route table. It then checks whether it has an unexpired route to the destination node. In order to respond to the RREQ, the node must either be the destination itself, or it must have an unexpired route to the destination whose corresponding sequence number is at least as great as that contained in the RREQ. If neither of these conditions are met, the node rebroadcasts the RREQ.

On the other hand, if it does meet either of these conditions, the node then creates a Route Reply (RREP) message. It places the current sequence number of the destination, as well as its distance in hops to the destination, into the RREP, and then unicasts this message back to the source. The node from which it received the RREQ is used as the next hop. When an intermediate node receives the RREP, it creates a *forward route entry* for the destination node in its route table, and then forwards the RREP to the source node. Once the source node receives the RREP, it can begin using the route to transmit data packets to the destination. If it later receives a RREP with a greater destination sequence number or equivalent sequence number and smaller hop count, it updates its route table entry and begins using the new route.

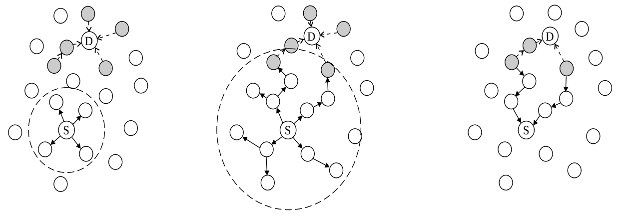
If the source node does not receive a RREP by the time its discovery timer expires, it rebroadcasts the RREQ. It attempts discovery up to some maximum number of times. If no route is discovered after the maximum number of attempts, the session is aborted.

B. Route Maintenance

An active route is defined as a route which has recently been used to transmit data packets. Link breaks in non-active links do not trigger any protocol action. However, when a link break in an active route occurs, the node *upstream* of the break determines whether any of its neighbors use that link to reach the destination. If so, it creates a Route Error (RERR) packet. The RERR contains the IP address of each destination which is now unreachable, due to the link break. The RERR also contains the sequence number of each such destination, incremented by one. The node then broadcasts the packet and invalidates those routes in its route table.

When a neighboring node receives the RERR, it in turn invalidates each of the routes listed in the packet, *if* that route used the source of the RERR as a next hop. If one or more routes are deleted, the node then goes through the same process, whereby it checks whether any of its neighbors route through it to reach the destinations. If so, it creates and broadcasts its own RERR message.

Once a source node receives the RERR, it invalidates the listed routes as described. If it determines it still needs any of the invalidated routes, it re-initiates route discovery for that route.



(a) RREQ: TTL = 1 (b) RREQ: TTL = 3 (c) Path of RREPs to Source

Fig. 1. Example of an expanding ring search.

III. MODIFICATIONS

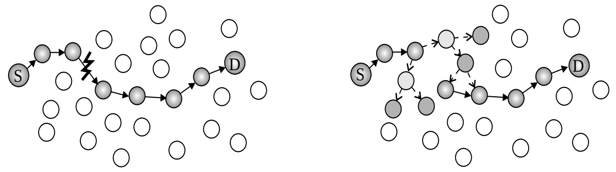
The scalability of many on-demand routing protocols may be limited due to a couple of important factors. The first is the need for flooding each RREQ. In small networks, flooding the RREQ across the network has a limited impact due to the relatively few nodes in the network. However, as networks grow to thousands and tens of thousands of nodes, flooding the entire network each time a route needs to be discovered consumes significant processing power at each network node, as well as excessive bandwidth during the floods.

As path lengths increase and as node mobility speeds rise, breaks in active routes occur with increasing frequency. Requiring an error message to be sent to the source node for each link break may result in an overwhelming number of route repairs by the source node. Particularly for high mobility and/or long path lengths, it may be true that the source node barely has time to rediscover a route before that route suffers from another link break.

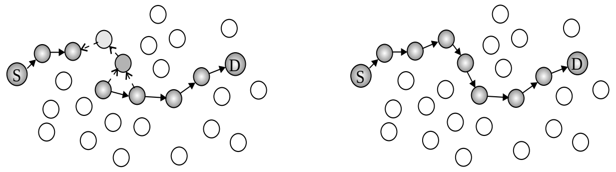
Because of these characteristics, on-demand routing protocols may not scale well to networks of large numbers of nodes and high mobility. To improve their scalability, the following modifications are offered. The expanding ring search and query localization can be used to reduce the area searched during a route discovery, and hence prevent flooding of the network. The current Internet draft specification of AODV [29] recommends such an expanding ring search be used for route discoveries. Local repair can also be used to provide immediate patching of breaks in active routes. Finally, the expanding ring search and query localization can be combined with local repair to provide increased scalability in both of these domains.

A. Expanding Ring Search

An expanding ring search works by searching successively larger areas, centered around the source node, until a node with a route to the destination is located. The basic premise behind the expanding ring search is to find some local node with a route to the destination, and thereby avoid flooding the entire network in search of such a route. Using an expanding ring search, the initial RREQ has a small "time to live (TTL)" value, i.e., two hops. Each time the RREQ is rebroadcast, the sending node decrements the TTL. Once the TTL reaches zero, the RREQ is no longer forwarded. The source node waits the discovery period for a RREP to be returned. If it has not received a RREP by the end of the discovery time, it initiates a new RREQ with the TTL incremented. This process continues until a threshold TTL value is reached. After this point, if no route has been located, the RREQ is flooded across the network. Figure 1 illustrates an example of an expanding ring search. In the figure, the shaded nodes indicate nodes which have a route to the destination. In a larger network with more nodes than that illustrated, the number of nodes undisturbed by the route query would be greater.



(a) Link Break in Active Route (b) Propagation of RREQ, κ=2



(c) Path of RREPs (d) Repaired Route

Fig. 2. Example of query localization.

To optimize the expanding ring search, the discovery time can be calculated so that it is proportional to the size of the area being searched. For instance,

$$\texttt{rte\_disc\_tmo} = 2 \times \text{TTL} \times \texttt{node\_traversal\_time}$$

results in the route discovery timeout being directly proportional to the TTL used for that discovery. Here the node traversal time is an approximation of the time required by the node to process and transmit a packet.

When re-discovering a route after a link break, the source places the last known hop count to the destination in the TTL field of the RREQ. If no route is found in this attempt, the TTL is increased by the TTL increment value. The TTL is increased on each subsequent route discovery attempt until the TTL threshold is reached. After this point, the RREQ is simply flooded to the entire network.

Utilizing the expanding ring search, a tradeoff exists between both the latency in finding the route (if it is not located on the first attempt) and the number of times local nodes receive the RREQ, and the drawback of flooding the entire network.

B. Query Localization

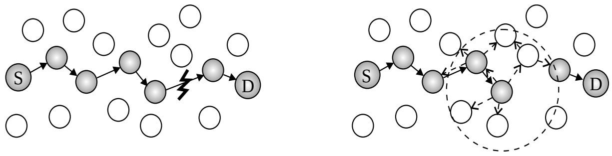
The query localization technique was developed by Castaneda and Das and described in [6]. Query localization is a method by which the flooding of the RREQ is restricted to some area that is based on the previously known route to the destination node. Hence the RREQ is not actually flooded at all, but instead is limited to a specific region of the network. Reference [6] presents two different techniques for performing query localization. For the purposes of this paper, method 2 (*Exploiting node locality*) is selected. This method assumes that the destination has traveled a bounded distance from its previous location, and hence can be found within some small number of hops from the most recently used route to it. To enable query localization, a counter is placed in the RREQ packet. Whenever a node that was not on the previous route to the destination receives the RREQ, it increments the counter. Conversely, when a node that was previously on the route to the destination receives the RREQ, it resets the counter to zero. Once the counter exceeds the threshold value κ, the RREQ is dropped.

On the initial route discovery for a destination, κ is set to the network diameter, so that the RREQ traverses the entire network. For a route repair, however, κ is initialized to a small value, i.e., two. If a route to the destination is not located on the first attempt, the value of κ may be increased until some maximum value is reached. Figure 2 illustrates an example of query localization. In the figure the last known route between the source and destination is highlighted. On the repair route discovery, κ is set to two. The shading of the nodes indicates their distance from the previously known route to the destination. As is evident from the figure, many of the network nodes do not need to receive the RREQ, and the query is able to be contained.

As with the expanding ring search, the drawback of the query localization protocol occurs when a route to the destination is not located on the first attempt. This results in certain nodes being repeatedly queried, as well as an increased route acquisition latency.

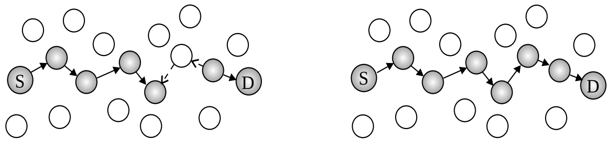
C. Local Repair

Local repair of link breaks in active routes is another approach to increasing scalability. In the current AODV specification, when a link break in an active route occurs, the node *upstream* of the break creates a Route Error (RERR) message listing all the destinations which have become unreachable due to the break. It then sends this message to its upstream neighbors, as described in Section II-B. If, instead of sending an error message to the source node, the upstream node attempts to repair the broken link itself, fewer data packets may be lost and the link can be repaired without the source node (and other upstream nodes) being disturbed. For short routes, local repair may not have any significant performance advantages. But for large networks with increasingly longer routes (e.g., 10 or more hops), it is likely that link breaks will occur so frequently that it will be nearly impossible for the source node to keep up with all the necessary repairs.



(a) Link break in active route

(b) Broadcast RREQ with small TTL



(c) Propagation of RREP

(d) Repaired route

Fig. 3. Example of local repair.

A node upstream of a link break that attempts to repair the route does so by broadcasting a RREQ with a TTL set to the last known distance to the destination, plus an increment value. This TTL value is used so that only the most recent whereabouts of the destination will be searched, which prevents flooding the entire network. The upstream node places the sequence number of the destination, incremented by one, into the RREQ. This prevents nodes further upstream on the route from replying to the RREQ, which would form a loop. Figure 3 illustrates an example of a local repair.

If a route to the destination is not located on the first attempt, a RERR message is sent back to the source node, and route re-discovery continues as described in Section II-B.

D. Combining the Modifications

The above modifications can be combined in various ways for increased protocol scalability. Specifically, the expanding ring search and local repair can work together, as can query localization and local repair. The expanding ring search and query localization are used to optimize route discoveries initiated by a source node, while local repair is used to decrease the number of route discoveries which a source node must initiate. The local repair, when combined with those two modifications, operates in the same fashion as previously described. One attempt at the repair is made locally. If this attempt is unsuccessful,

TABLE I
SUMMARY OF ROOM SIZES.

# of Nodes	Room Size (m^2)	Average # of Neighbors
50	1000 × 1000	7.32
100	1500 × 1500	7.46
500	3500 × 3500	7.33
1000	5000 × 5000	7.69
5000	11500 × 11500	7.22
10000	16000 × 16000	7.50

a RERR message is sent back to the source.

IV. SIMULATIONS

A. Environment

The simulations used to evaluate the scalability of AODV and its modifications were implemented within the GloMoSim library [38]. The GloMoSim library is a scalable simulation environment for wireless network systems using the parallel discrete-event simulation capability provided by PARSEC [2]. The simulations model networks between 50 and 10,000 mobile hosts placed randomly within the simulation area. The simulation boundary and average connectivity for each simulated number of nodes are shown in Table I. The room size for each simulation was chosen so as to keep the node density approximately constant in the different size networks. Instead of holding the room size constant and increasing the node population density, the node density was held constant in the simulations because it was desired to investigate the scalability of networks in terms of increasing the room size, as opposed to increasing the density. Increasing density causes congestive failures not closely related to routing protocol performance. Furthermore, studies have shown that the ad hoc network performance is optimal when the average number of neighbors is between six and eight [17], [32].

The radio propagation range for each node is 250 meters and channel capacity is 2 Mb/s. Each simulation is executed for 300 seconds of simulation time. Because of the long real-time simulation run for large network experiments, only five runs for each scenario were performed. The results of these runs were averaged together to produce the resulting graphs.

A.1 Channel and Radio Model

A free space propagation model [31] with a threshold cutoff was used in the experiments. In the free space model, the power of a signal attenuates as $1/d^2$, where d is the distance between radios. In the radio model, capture is assumed, whereby a radio has the ability to lock onto a sufficiently strong signal in the presence of interfering signals. If the capture ratio (the minimum ratio of an arriving packet's signal strength relative to those of other colliding packets) [31] is greater than the predefined threshold value, the arriving packet is received while other interfering packets are dropped.

A.2 Medium Access Control Protocol

The IEEE 802.11 MAC protocol with Distributed Coordination Function (DCF) [12] is used as the MAC layer in the experiments. DCF is the basic access method used by nodes to share the wireless channel under independent ad hoc configuration. The access scheme is Carrier Sense Multiple Access/Collision Avoidance (CSMA/CA) with acknowledgments. Optionally, the nodes can make use of Request-To-Send/Clear-To-Send (RTS/CTS) channel reservation control frames for unicast, virtual carrier sense, and fragmentation of packets larger than a given threshold. By setting timers based upon the reservations in RTS/CTS packets, the virtual carrier sense augments the physical carrier sense in determining when mobile nodes perceive that the medium is busy. Fragmentation is useful in the presence of high bit error and loss rates, as it reduces the size of the data units that need to be retransmitted.

A.3 Traffic Pattern

A traffic generator was developed to simulate constant bit rate sources. The size of data payload is 512 bytes. Twenty data sessions with randomly selected sources and destinations are simulated. Each source transmits data packets at a rate of four packets/sec. The number of data sessions was held constant to limit the number of variables in the experiment, and because of the time required to run the large simulations with more data sessions.

A.4 Mobility Pattern

The random waypoint model [15] is utilized as the mobility model. In this model, a node selects a random destination within the terrain range and moves towards that destination at a speed between the pre-defined minimum and maximum speed. Once the node arrives at the destination, it stays at its current position for a pause time. After being stationary for the pause time, it randomly selects another destination and speed and then resumes movement. The minimum speed for the simulations is 0 m/s while the maximum speed is 10 m/s. The selected pause time is 30 seconds.

A.5 Parameter Values

Table II gives a summary of the chosen parameter values. The network diameter (`net_diameter`) for the sim-

TABLE II

PARAMETER VALUES.

	Parameter	Value
General	`net_diameter`	35, 70
	`node_traversal_time`	40 ms
Expanding Ring Search	`ttl_start`	1
	`ttl_increment`	2
	`ttl_threshold`	7
Query Localization	κ	2
	κ_2	$\kappa \times 2$
	τ	10 sec
Local Repair	`local_add_ttl`	2

TABLE III

PROTOCOL ABBREVIATIONS.

Protocol Combination	Abbreviation
AODV	AODV
AODV and Expanding Ring Search	AODV-ERS
AODV and Query Localization	AODV-QL
AODV and Local Repair	AODV-LR
AODV, Expanding Ring Search and Local Repair	AODV-ERS-LR
AODV, Query Localization and Local Repair	AODV-QL-LR

ulations represents the approximate diameter of the network, and is used for setting the TTL value of broadcast control packets. It is also a factor in the calculation of how long a node should wait to receive a RREP after sending another RREQ. If the RREQ is broadcast across the network, the reception of the RREP may take longer for large networks than for small. The setting of the `net_diameter` variable to 35 for small networks (50, 100, 500, and 1,000 nodes) and 70 for the larger networks (5,000 and 10,000 nodes) provides an upper bound of the actual network diameter for these networks.

The `node_traversal_time` represents an estimation of the processing time of a packet at a given node. It is also used for estimating the period of time a source node should wait to receive a RREP after broadcasting a RREQ.

The values selected for the modification parameters represent a tradeoff between minimizing the number of searches required to locate a given destination, and reducing the number of nodes that must receive and process the RREQ packet. For the expanding ring search, `ttl_start`, the initial TTL value of the RREQ is set to one. Each time a reply is not received, the TTL is incremented by `ttl_increment`, until the threshold value (`ttl_threshold`) is reached. After that point, the RREQ is broadcast across the network. When rediscovering routes, the initial TTL is the last known hop count to the destination by the source.

The value of κ for query localization represents the number of hops the RREQ is allowed to travel off the previously known path to the destination. The initial value of κ is set to two. If no reply is received, the value of κ is doubled for the second attempt. The value of τ is ten seconds. If a node has been part of the most recent route for the past τ time units and receives the RREQ, it resets the κ to zero.

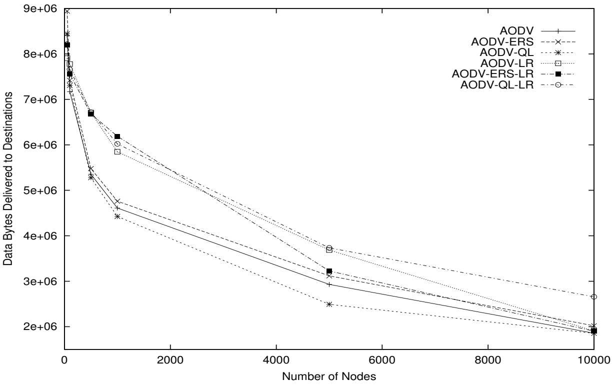


Fig. 4. Throughput.

Finally, the `local_add_ttl` parameter is used for a local repair. It represents the value added to the previously known distance to the destination. This sum is used as the TTL of the RREQ for the local repair.

Among the runs that were performed with varied parameter values for expanding ring search, query localization, and local repair, the values that yielded the best results are presented.

B. *Results and Analysis*

The following sections present the results achieved by the different protocol combinations. Table III indicates the abbreviation associated with each protocol combination in the following figures.

B.1 Throughput

Figure 4 shows each scheme's throughput performance, where throughput is calculated to be the number of data bytes delivered to destination hosts. The figure shows that the ability of the protocols to deliver packets to their destination degrades as the network size becomes larger. The path length is greater in larger networks because the sim-

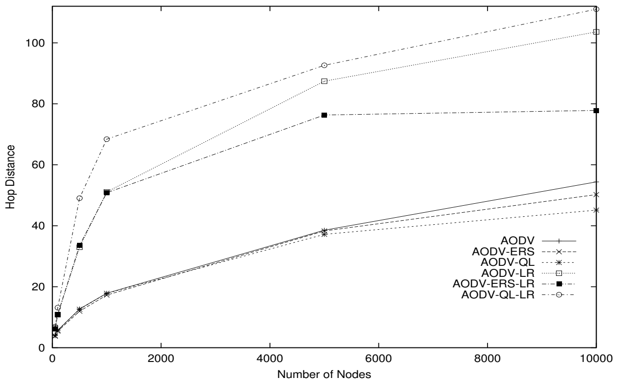


Fig. 5. Path length.

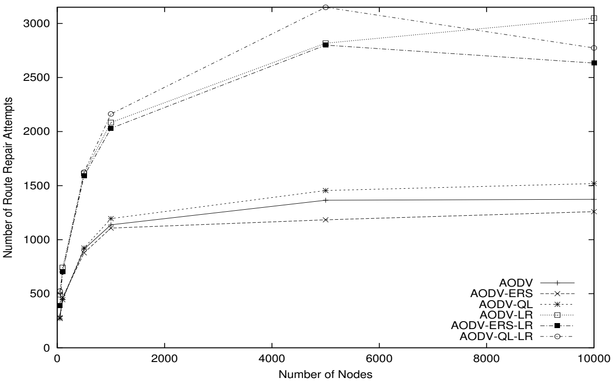


Fig. 6. Number of route repair attempts.

ulation area and the number of nodes increase while the average number of neighbors is kept relatively constant (see Table I). Routes are more prone to disconnections in mobile networks when path lengths are longer. Because any link failure along the path results in the inability of the source to reach the destination, longer routes have a greater probability of route disconnection than shorter hop routes. An increased route length in larger multihop networks is a characteristic not only of on-demand routing protocols, but any routing protocols such as table driven algorithms (i.e., distance vector and link state) and hierarchical clustered routing protocols. It is observed that performing route repair locally improves throughput. Since nodes closer to the destination than the source initiate route rediscovery, new routes are repaired more quickly and fewer data packets are dropped. It is interesting to note that AODV-QL has the poorest throughput. The main purpose of query localization is to exploit node locality and reduce the number of routing message transmissions. Localizing the query, however, has the risk of not being able to establish the route.

The path length of each scheme is presented in Figure 5. The route length is measured by calculating the distance between the source and destination when the route is constructed. The measure includes the first discovered route for both the construction of new routes, and the repair of broken routes. It is observed that schemes that utilize the local repair technique yield longer paths. For protocols that do not use local repair, only the source node can reconstruct routes. When a source rediscovers routes with a request/reply cycle, a new route is obtained based on current network information such as hop count, route freshness, node location, network topology, etc. On the other hand, in local repair schemes, the node immediately upstream of the disconnected link initiates a route reconstruction. Because of the possibility that the destination has actually moved closer to the source node, but the distance between the node reconstructing the route and destination has increased, path lengths tend to grow as intermediate nodes repair routes.

Longer path lengths naturally result in more route breaks and more route recoveries, as shown in Figure 6. Local repair schemes have more route reconstruction attempts for the following two reasons. First, longer routes can fail more easily than shorter routes. Second, no RERR message is sent upstream to the source in local repair. If a link upstream of the previously broken link becomes disconnected while a new route is being discovered, another local repair procedure is initiated. This results in more route reconstruction attempts by local repair schemes. It is also interesting to note that the number of repair attempts does not significantly increase between 5,000 and 10,000 nodes. This is due to the difficultly of all the protocols in maintaining routes with such a large path length. In these scenarios, many sessions are forced to abort due to the inability to maintain a route. Hence, with fewer sessions being maintained, fewer route recoveries are necessary.

B.2 Control Message Overhead

The routing message overhead is presented as the number of control message transmissions in Figure 7. Each hop-wise transmission of a control message by a node is counted as one transmission. As expected, AODV without modification has the most control packet transmissions. Local repair schemes have less control overhead compared with schemes that perform route repair by sources. AODV-QL-LR has the least control overhead among all protocols. Local repair schemes reduce the number of RREQ transmissions. As shown in Figure 8, the percentage of RREQ transmissions among all routing packet transmissions (RREQ, RREP, and RERR) by local repair schemes

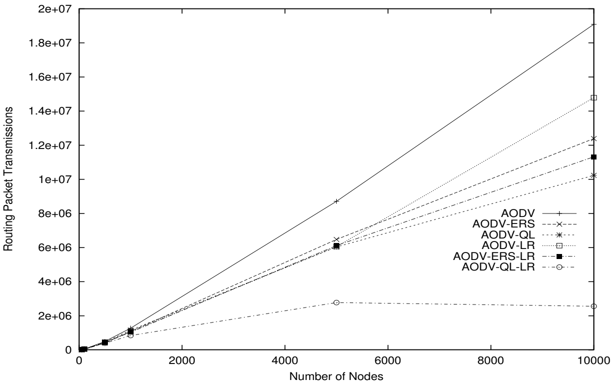


Fig. 7. Routing message overhead.

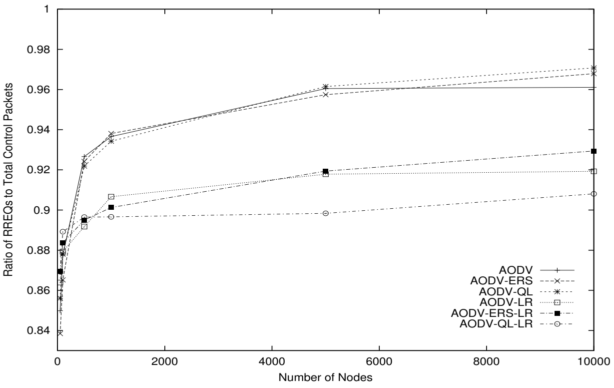


Fig. 8. Percentage of RREQ transmissions among control packet transmissions.

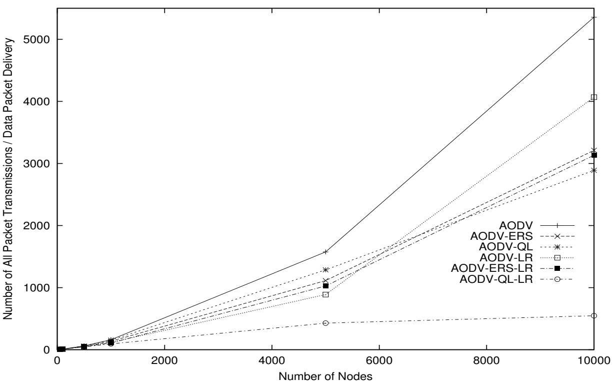


Fig. 9. Number of all packet transmissions per data packet delivery.

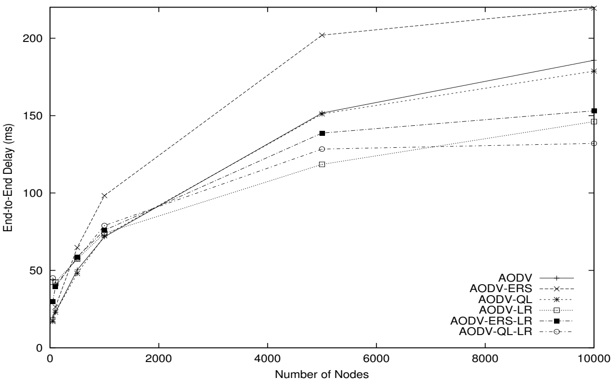


Fig. 10. End-to-end delay.

is lower than source repair schemes.

In order to evaluate protocol efficiency, the number of all packet (i.e., data, RREQ, RREP, and RERR) transmissions per data delivery is investigated. Because link layer protocols for ad hoc networks are contention-based, this measure is very important for protocol analysis. The measure is presented in Figure 9. The scope and the ranking of protocols are similar to those of Figure 7. Note the large number of packet transmissions per successful delivery at high node populations shown in Figure 9. This ratio, which can grow as large as 5,000, indicates the drastic need for work in a crucial area affecting the scalability of AODV, and probably all known ad hoc on-demand routing protocols, to large network populations. The ratio should be brought down by three orders of magnitude; such a reduction will probably also be accompanied by a proportional increase in the packet delivery fraction, which is sometimes as low as 15%. Work towards developing techniques for quickly re-establishing valid routes is likely to be of the highest importance for improving the scalability of ad hoc networks.

B.3 Latency

The end-to-end delay of each protocol is reported in Figure 10. Schemes that utilize the local repair technique have shorter delays. Protocols in which sources initiate route repair have longer end-to-end delays because of longer route re-establishment latency. To recover a broken route, a RERR packet must first be delivered from the node upstream of the broken link to the source of the route. The RREQ must then be broadcast from the source to the destination, and a RREP consequently has to be transmitted back to the source. Data packets are buffered at the source node during this process and this duration of time adds to the end-to-end delay. In local repair schemes, on the other hand, the node upstream of the disconnected link initiates an immediate route reconstruction. Since route rediscovery is done locally, less time is needed to search for and obtain a new route. Local repair schemes can, therefore, yield shorter delays. Note that AODV-ERS has the longest delay because a route may not be built in the initial attempt (i.e., TTL = 1 or last known hop count of the route). Among local repair schemes, AODV-ERS-LR has the longest delay for the same reason.

V. OBSERVATIONS

In the previous sections, the scalability characteristics of on-demand routing protocols has been studied. These protocols are known to generally perform well in mobile multihop networks. It was shown that routing in ad hoc networks of tens of thousands of nodes is extremely difficult. In large networks, path lengths are longer compared with those in small networks (i.e., 50 or 100 nodes). Because network hosts are capable of mobility, longer routes are more prone to disconnection since a single link failure results in a broken route. Each route invalidation invokes a route repair process and burdens the network with control messages. Worse, because there are generally multiple hops between a source and destination, and because nodes are mobile, many route discoveries are unsuccessful. Although the flooded RREQ packet reaches the destination or intermediate nodes with routing information to the destination, the unicast RREP packet may not reach the source due to link breaks. Even when the RREP packet survives to reach the source, the route may break shortly afterwards and the source will need to initiate another route discovery. Therefore, maintaining routes with many hops in mobile ad hoc networks is a difficult challenge.

This paper introduces three techniques and applies five different modification combinations to improve AODV scalability. The expanding ring search reduces the routing message overhead, but yields longer delays because of initial route discovery failures. Query localization also decreases control overhead, but it has poor throughput performance due to low route repair rate. This is especially true when routes have long distances. Local repair proves to be effective in enhancing AODV's performance in large networks. Because route repair is localized, new routes are found more quickly than source initiated route discoveries. Consequently, packet drops are minimized. Local repair works efficiently with the expanding ring search and query localization to reduce control message overhead. The drawback of local repair, however, is that multiple repairs for the same route can be present at the same time.

Local repair may benefit from some mechanism to reduce the growth in path lengths which result from this method. One possible solution is to combine local repair with a RERR unicast back to the source. If a link breaks in an active route, the node upstream of that break could repair the route using local repair, and then send a RERR message back to the source. In this way, as the upstream node continues to receive data packets while the RERR is traveling to the source node, the data packets can still be forwarded to the destination. When the source receives the RERR, it can decide whether to reinitiate route discovery to look for a better route in order to reduce the length of the route if it has increased significantly. This method will result in fewer dropped packets than not using local repair while also reducing the increase in path lengths which results from local repair. The cost is another increase in the number of broadcast control messages.

VI. CONCLUSION

This paper has evaluated the scalability of on-demand ad hoc routing protocols by selecting a representative from this set of protocols and simulating it in networks of up to 10,000 nodes. To improve the performance of on-demand protocols in large networks, five modification combinations have been separately incorporated into an on-demand protocol, and their respective performance has been studied. It has been shown that the use of local repair is beneficial in increasing the number of data packets that reach their destinations. Expanding ring search and query localization techniques seem to further reduce the amount of control overhead generated by the protocol, by limiting the number of nodes affected by route discoveries.

While the performance improvements of the modifications have only been demonstrated with the AODV protocol, we believe that other on-demand ad hoc routing protocols will show similar improvements when incorporated with the modifications we studied. The verification of this belief, however, remains future work.

Scalability in ad hoc mobile networks is inherently difficult due to the mobility of the nodes and the transience of network links. Work on large-scale ad hoc networks is likely to uncover techniques that would be valuable for stabilizing routing protocols in the Internet at large, leading to faster route convergence and reduced route flaps. Creating ad hoc routing protocols which experience minimal performance degradation when used in increasingly large networks is a challenge, and there remains a significant amount of work to reach this goal.

REFERENCES

[1] G. Aggelou and R. Tafazolli, "RDMAR: A Bandwidth-efficient Routing Protocol for Mobile Ad Hoc Networks," *Proceedings of the ACM WoWMoM'99*, pp. 26-33, Seattle, WA, August 1999.

[2] R. Bagrodia, R. Meyer, M. Takai, Y. Chen, X. Zeng, J. Martin, and H. Y. Song, "PARSEC: A Parallel Simulation Environment for Complex Systems," *IEEE Computer*, vol. 31, no. 10, pp.77-85, October 1998.

[3] S. Basagni, “Distributed Clustering for Ad Hoc Networks,” *Proceedings of the IEEE I-SPAN’99*, pp. 310-315, Perth, Western Australia, June 1999.

[4] D. Beyer, M. Frankel, J. Hight, D. Lee, M. Lewis, R. McKenney, J. Naar, R. Ogier, N. Shacham, and W. Zaumen, “Packet Radio Network Research, Development and Application,” *Proceedings of the SHAPE Packet Radio Symposium*, 1989.

[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 ACM/IEEE MobiCom’98*, pp. 85-97, Dallas, TX, October 1998.

[6] R. Castaneda and S. R. Das, “Query Localization Techniques for On-demand Routing Protocols in Ad Hoc Networks,” *Proceedings of the ACM/IEEE MobiCom’99*, pp.186-194, Seattle, WA, August 1999.

[7] C.-C. Chiang, H.-K. Wu, W. Liu, and M. Gerla, “Routing in Clustered Multihop, Mobile Wireless Networks with Fading Channel,” *Proceedings of the IEEE SICON’97*, pp. 197-211, Singapore, April 1997.

[8] S. R. Das, C. E. Perkins, and E. M. Royer, “Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks,” *Proceedings of the IEEE INFOCOM 2000*, pp. 3-12, Tel Aviv, Israel, March 2000.

[9] J. J. Garcia-Luna-Aceves and M. Spohn, “Source-Tree Routing in Wireless Networks,” *Proceedings of the IEEE ICNP’99*, pp. 273-282, Toronto, Canada, November 1999.

[10] M. Gerla and J. T.-C. Tsai, “Multicluster, Mobile, Multimedia Radio Network,” *ACM/Baltzer Wireless Networks*, vol. 1, no. 3, pp. 255-265, March 1995.

[11] S. Guha and S. Khuller, “Approximation Algorithms for Connected Dominating Sets,” *Algorithmica*, vol. 20, no. 4, pp. 374-387, April 1998.

[12] IEEE Computer Society LAN MAN Standards Committee, *Wireless LAN Medium Access Protocol (MAC) and Physical Layer (PHY) Specification*, IEEE Std 802.11-1997. The Institute of Electrical and Electronics Engineers, New York, NY, 1997.

[13] M. Jiang, J. Li, and Y. C. Tay, “Cluster Based Routing Protocol (CBRP) Functional Specification,” *IETF Internet Draft*, draft-ietf-manet-cbrp-spec-01.txt, August 1999 (Work in Progress).

[14] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, “Scenario-based Performance Analysis of Routing Protocols for Mobile Ad-hoc Networks,” *Proceedings of the ACM/IEEE MobiCom’99*, pp. 195-206, Seattle, WA, August 1999.

[15] D. B. Johnson and D. A. Maltz, “Dynamic Source Routing in Ad Hoc Wireless Networks,” In *Mobile Computing*, edited by T. Imielinski and H. Korth, pp. 153-181, Kluwer Academic Publishers, 1996.

[16] J. Jubin, “Uphill Tier Routing, Less Frequent Tier Data Updating, and Larger Networks,” *SURAN* Program Technical Note (SRNTN) 2, Richardson, TX, Rockwell Inc., 1983.

[17] L. Kleinrock and J. Silvester, “Optimum Transmission Radii for Packet Radio Networks or Why Six is a Magic Number,” *Proceedings of National Telecommunications Conference*, pp. 4.3.2-4.3.5, Birmingham, AL, December 1978.

[18] G. Lauer, “Hierarchical Routing Design for SURAN,” *Proceedings of the IEEE ICC*, pp. 93-102, Toronto, Canada, June 1988.

[19] S.-J. Lee, W. Su, J. Hsu, M. Gerla, and R. Bagrodia, “A Performance Comparison Study of Ad Hoc Wireless Multicast Protocols,” *Proceedings of the IEEE INFOCOM 2000*, pp. 565-574, Tel Aviv, Israel, March 2000.

[20] C. R. Lin and M. Gerla, “Adaptive Clustering for Mobile Wireless Networks,” *IEEE Journal on Selected Areas in Communications*, vol. 15, no. 7, pp. 1265-1275, September 1997.

[21] G. S. Malkin and M. E. Steenstrup, “Distance-Vector Routing,” In *Routing in Communications Networks*, edited by M. Steenstrup, pp. 83-98, Prentice Hall, 1995.

[22] J. Moy, “Link-State Routing,” In *Routing in Communications Networks*, edited by M. Steenstrup, pp. 135-157, Prentice Hall, 1995.

[23] S. Murthy and J. J. Garcia-Luna-Aceves, “An Efficient Routing Protocol for Wireless Networks,” *ACM/Baltzer Mobile Networks and Applications*, vol. 1, no. 2, pp. 183-197, October 1996.

[24] V. D. Park and M. S. Corson, “A Highly Adaptive Distributed Routing Algorithm for Mobile Wireless Networks,” *Proceedings of the IEEE INFOCOM’97*, pp. 1405-1413, Kobe, Japan, April 1997.

[25] M. R. Pearlman and Z. J. Haas, “Determining the Optimal Configuration for the Zone Routing Protocol,” *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1395-1414, August 1999.

[26] G. Pei, M. Gerla, T.-W. Chen, “Fisheye State Routing: A Routing Scheme for Ad Hoc Wireless Networks,” *Proceedings of the IEEE ICC 2000*, pp. 70-74, New Orleans, LA, June. 2000.

[27] C. E. Perkins and P. Bhagwat, “Highly Dynamic Destination-Sequenced Distance-Vector Routing (DSDV) for Mobile Computers,” *Proceedings of the ACM SIGCOMM’94*, pp. 234-244, London, UK, September 1994.

[28] C. E. Perkins and E. M. Royer, “Ad-hoc On-Demand Distance Vector Routing,” *Proceedings of the IEEE WMCSA’99*, pp. 90-100, New Orleans, LA, February 1999.

[29] C. E. Perkins, E. M. Royer, and S. R. Das, “Ad Hoc On Demand Distance Vector (AODV) Routing,” *IETF Internet Draft*, draft-ietf-manet-aodv-08.txt, March 2001 (Work in Progress).

[30] R. Ramanathan and M. Steenstrup, “Hierarchically-Organized, Multihop Mobile Wireless Networks for Quality-of-Service Support,” *ACM/Baltzer Mobile Networks and Applications*, vol. 3, no. 1, pp. 101-119, June 1998.

[31] T. S. Rappaport, *Wireless Communications: Principles and Practice*, Prentice Hall, Upper Saddle River, NJ, October 1995.

[32] E. M. Royer, P. M. Melliar-Smith, and L. E. Moser, “An Analysis of the Optimum Node Density for Ad hoc Mobile Networks,” *Proceedings of the IEEE ICC’01*, pp. 857-861, Helsinki, Finland, June 2001.

[33] E. M. Royer and C. E. Perkins, “Multicast Operation of the Ad-hoc On-Demand Distance Vector Routing Protocol,” *Proceedings of the ACM/IEEE MobiCom’99*, pp. 207-218, Seattle, WA, August 1999.

[34] N. Shacham, “Hierarchical Routing in Large, Dynamic Ground Radio Networks,” *Proceedings of the HICSS’85*, pp. 292-301, Honolulu, HI, January 1985.

[35] P. Sinha, R. Sivakumar, and V. Bharghavan, “CEDAR: a Core-Extraction Distributed Ad hoc Routing Algorithm,” *Proceedings of the IEEE INFOCOM’99*, pp. 202-209, New York, NY, March 1999.

[36] J. Stevens, “Spreading Connectivity Information out over Multiple PROP Periods, and Timeliness of Information,” *SURAN Program Technical Note (SRNTN) 21*, Richardson, TX, Rockwell Inc., May 1985.

[37] C.-K. Toh, “Associativity-Based Routing for Ad-Hoc Mobile Networks,” *Wireless Personal Communications Journal*, vol. 4, no. 2, pp. 103-139, Kluwer Academic Publishers, March 1997.

[38] UCLA Parallel Computing Laboratory and Wireless Adaptive Mobility Laboratory, *GloMoSim: A Scalable Simulation Environment for Wireless and Wired Network Systems*, http://pcl.cs.ucla.edu/projects/glomosim.