Performance Comparison of Two Location Based Routing Protocols for Ad Hoc Networks

Tracy Camp, Jeff Boleng, Brad Williams, Lucas Wilcox, William Navidi
Department of Math. and Computer Sciences
Colorado School of Mines
Golden, CO 80401
tcamp, jboleng, bwilliam, wnavidi, and lwilcox@mines.edu

***Abstract*— In recent years, many location based routing protocols have been developed for ad hoc networks. This paper presents the results of a detailed performance evaluation on two of these protocols: Location-Aided Routing (LAR) and Distance Routing Effect Algorithm for Mobility (DREAM). We compare the performance of these two protocols with the Dynamic Source Routing (DSR) protocol and a minimum standard (i.e., a protocol that floods all data packets). We used NS-2 to simulate 50 nodes moving according to the random waypoint model. Our main goal for the performance investigation was to stress the protocols evaluated with high data load during both low and high speeds. Our performance investigation produced the following conclusions. First, the added protocol complexity of DREAM does not appear to provide benefits over a flooding protocol. Second, promiscuous mode operation improves the performance of DSR significantly. Third, adding location information to DSR (i.e., similar to LAR) increases both the network load and the data packet delivery ratio; our results conclude that the increase in performance is worth the increase in cost. Lastly, our implementation of DREAM provides a simple location service that could be used with other ad hoc network routing protocols.**

I. INTRODUCTION

An ad hoc network is a set of wireless mobile nodes (MNs) that cooperatively form a network without specific user administration or configuration. Each node in an ad hoc network is in charge of routing information between its neighbors, thus contributing to and maintaining connectivity of the network. Since ad hoc networks have proven benefits, they are the subject of much current research. Many unicast routing protocols have been proposed for ad hoc networks; a performance comparison for a few of the protocols are in [1] and [2]. Some of the unicast routing protocols for an ad hoc network use location information in the routing protocol in an effort to improve the performance of unicast communication. A few of the proposed algorithms include the Location-Aided Routing (LAR) algorithm [3], the Distance Routing Effect Algorithm for Mobility (DREAM) [4], the Greedy Perimeter Stateless Routing (GPSR) algorithm [5], and the Geographical Routing Algorithm (GRA) [6].

This paper is the first to provide a detailed, quantitative evaluation comparing the performance of two location based ad hoc network routing protocols: LAR and DREAM. Simulation results on LAR, DREAM, and other location based protocols exist on the individual protocols; however, since these simulation results are based on different simulation environments, different simulation parameters and even different network simulators, the performances are not comparable. We compare the simulation results for LAR and DREAM with the Dynamic Source Routing (DSR) protocol [7], a unicast routing protocol that does not use location information. We chose DSR since it performs well in many of the performance evaluations of unicast routing protocols (e.g. [1], [2]).

The NS-2 code used in our simulations of DSR was obtained from [8]; we wrote the NS-2 code used in our simulations of LAR and DREAM. During implementation, we followed the protocol descriptions provided for LAR in [3] and DREAM in [4]. When implementation questions occurred, we contacted the protocol authors for guidance. We discuss the implementation decisions made and protocol parameters chosen in the description of each protocol. Some of the simulation results we present are different from previously reported results; we discuss the reasons for the differences in this paper. Lastly, at the end of this paper, we list five conclusions which summarize our findings.

II. PROTOCOLS STUDIED

A. Dynamic Source Routing (DSR)

DSR is a source routing protocol which determines routes on demand [7]. In a source routing protocol, each packet carries the full route (a sequenced list of nodes) that the packet should be able to traverse in its header. In an on demand routing protocol (or reactive protocol), a route to a destination is requested only when there is data to send to that destination and a route to that destination is unknown or expired. In the evaluation of DSR, both [1] and [2] only locate routes that consist of bi-directional links. (Although DSR does not require bi-directional links in the protocol, IEEE 802.11 requires bi-directional links in the delivery of all non-broadcast packets.) The version of DSR in our study also only locates bi-directional links. In other words, a route reply packet containing the complete route from S to D is sent along the reverse route to S.

MNs using DSR may operate in promiscuous mode. In promiscuous mode, an MN can learn potentially useful routes

This work supported in part by NSF Grants ANI-9996156 and ANI-0073699.

TABLE I
DSR CONSTANTS

Timeout for 1 hop route request	30 ms
Retransmit route request	500 ms
Size of header with n addresses	4n + 4 bytes
Buffer size	64 packets
Packet lifetime in buffer	30 s
Max rate for route replies	1/s

by listening to packets not addressed to it. Simulation results on DSR presented in [1] use promiscuous mode operation, while simulation results on DSR presented in [2] do not use promiscuous mode operation. Contrary to comments in [2], we discovered that including promiscuous mode operation in DSR significantly reduced control overhead and significantly increased delivery ratio at higher speeds. However, as noted in [2], promiscuous mode operation is power consuming. Thus, we chose to present both promiscuous mode operation and non-promiscuous mode operation in our simulation results for DSR.

A version of DSR from [8] was used for our simulations. The constants chosen for DSR's parameters are the same as those used in [1] and [2] (see Table I). Note that although the time to hold a packet awaiting a route is 30 seconds, the results in [1] and herein never hold a packet for longer than 16 seconds. That is, 4 packets are transmitted every second and the buffer size for holding packets is 64; thus, no more than 16 seconds of packets can be held.

B. Location Aided Routing (LAR)

1) Protocol Overview: Like DSR, LAR [3] is an on-demand source routing protocol. The main difference between LAR and DSR is that LAR sends location information in all packets to (hopefully) decrease the overhead of a future route discovery. In DSR [7], if the neighbors of S do not have a route to D, S floods the entire ad hoc network with a route request packet for D. LAR uses location information for MNs to flood a route request packet for D in a *forwarding zone* instead of in the entire ad hoc network. (The term forwarding zone in this paper is defined the same as the term request zone in [3].) This forwarding zone is defined by location information on D. The authors of [3] propose two methods used by intermediate nodes between S and D to determine the forwarding zone of a route request packet.

In method 1, which we call LAR Box, a neighbor of S determines if it is within the forwarding zone by using the location of S and the expected zone for D. The expected zone is a circular area determined by the most recent location information on D, (X_D, Y_D), the time of this location information, (t_0), the average velocity of D, (V_{avg}), and the current time, (t_1). This information creates a circle with radius $R = V_{avg} \times (t_1 - t_0)$ centered at (X_D, Y_D). The forwarding zone is a rectangle with S in one corner, (X_S, Y_S), and the circle containing D in the other corner.

If a neighbor of S determines it is within the forwarding zone, it forwards the route request packet further. An MN that is not a neighbor of S determines if it is within the forwarding zone by using the location of the neighbor that sent the MN the route request packet and the expected zone for D based on the most recent available information. Thus the forwarding zone and the expected zone adapt during transmission. (This adaptation is mentioned in [3] as a possible optimization to the LAR protocol.)

In method 2, which we call LAR Step, an intermediate MN determines if it is within the forwarding zone if the MN is closer to D than the neighbor that sent the MN the route request packet. Specifically, if the distance of the neighbor that sent the MN the route request packet to D is S_{dist}, and the distance of the MN that received the route request packet to D is C_{dist}, then the MN will forward the route request packet if $C_{dist} \leq S_{dist}$.

In both LAR Box and LAR Step, [3] offers the option to increase or decrease the size of the forwarding zone via an error factor, δ. With this error factor, the above formulas become:

$$\begin{array}{ll} \text{LAR Box:} & R = (V_{avg} \times (t_1 - t_0)) + \delta \\ \text{LAR Step:} & C_{dist} \leq (S_{dist} + \delta) \end{array}$$

Both LAR Box and LAR Step include a two stage route discovery method. In the first stage, the route request packet is forwarded according to either LAR Box or LAR Step. If a route reply packet is not received within the route request timeout period, then a second route request packet is flooded through the entire ad hoc network. If a route reply packet is (again) not received within the route request timeout period, then D is considered unreachable. If D remains unreachable for 30 seconds, packets for D are dropped.

2) Implementation Decisions: Unlike the performance results on LAR presented in [3], we evaluated all the variations and optimizations (except the alternative definitions of the forwarding zone) proposed in [3]. These optimizations include adaptation of the request zone based on more recent location information (discussed in Section II-B.1), propagation of location and speed information in every packet transmitted, and local search for route repair. The results presented in this paper include two of these three optimizations. We did not include the local search optimization (see [3]) in our simulations since the performance results in doing so were unsatisfactory. When an intermediate MN attempts a local search, data packets are held at the intermediate MN (instead of dropped) in the hope that a route discovery call by the intermediate MN will prove beneficial. Waiting at the intermediate MN increases end-to-end delay substantially; specifically, if a route isn't discovered, the data packet may wait a full 30 seconds at the intermediate MN.

In our LAR implementation, as in DSR, a source asks its neighbors for a route to a destination before transmitting a route request in the forwarding zone. Although this feature is not mentioned in [3], we found that including this feature improved the performance results. Lastly, although not mentioned as a possible variation in [3], we evaluated allowing an intermediate MN to respond to a route request (if a route is available). However, a route reply from an intermediate MN does not update the source with recent location information on the destination; thus, the source floods route requests more often when this variation is used. Without allowing an intermediate MN to respond to a route request, the benefits of promiscuous mode operation are significantly reduced. Thus, our performance results on LAR are for non-promiscuous mode operation.

In the LAR protocol, route errors are generated when a route breaks; since a MAC layer does not exist in the original LAR implementation (see [3]), details on how route errors are generated are missing. In our implementation of LAR, following the implementation of DSR, route errors in LAR are discovered by the MAC layer via link layer feedback at the transmitting node. When a route error is discovered, a route error message is unicast to S along the reverse source route. Lastly, when a route error occurs, the MN that discovers the error looks in its cache for another route from itself to D. In other words, similar to DSR, the MN forwards the packet along a new route if another route is available. Our implementation of LAR in NS-2 includes the error factor, δ.; however, following the results presented in [3], we set the error factor to zero in all our simulations. Table II lists the constants used in our implementation of LAR.

TABLE II

LAR Constants

Timeout for 1 hop route request	30 ms
Route request timeout	500 ms
Forwarding error factor (δ)	0.0
Size of header with n addresses	4n + 40 bytes
Buffer size	64 packets
Packet lifetime in buffer	30 s

C. DREAM

1) Protocol Overview: Unlike DSR and LAR, DREAM is not an on demand routing protocol [4]. Instead, each MN in this proactive protocol maintains a location table for all other nodes in the ad hoc network. To maintain the table, each MN transmits location packets to nearby MNs in the ad hoc network at a given frequency and to faraway MNs in the ad hoc network at another lower frequency. Since faraway MNs *appear* to move more slowly than nearby MNs, it is not necessary for an MN to maintain up-to-date location information for faraway MNs. Thus, by differentiating between nearby and faraway MNs, DREAM attempts to limit the overhead of location packets.

Each location packet (LP), which updates location tables, contains the coordinates of the source node based on some reference system, the source node's speed, and the time the LP was transmitted. Suppose a source node S needs to send data to a destination D. In DREAM, S first calculates a circle around the most recent location information for D, using the last known speed. The radius is $R = V_{max} \times (t_1 - t_0)$ centered at (X_d, Y_d).

Once the circle is calculated, S defines its forwarding zone (a cone) to be the region enclosed by an angle whose vertex is at S and whose sides are tangent to the circle calculated for D. Similar to LAR, S sends a packet for D to all its neighbors in the forwarding zone; however, in DREAM, the packet is a data packet not a route request. Each of these neighbors then compute their own forwarding zones, based on their own location tables, and forward the packets accordingly. When D receives a data packet, D returns an ACK packet. The ACK packet is sent to S in the same manner as the data packet was sent to D.

An ACK packet may not be received by S due to the following reasons: there is no route to the destination from the source (i.e., no neighbors in the calculated cone), there is no route to the source from the destination, or there is an error in transmission (e.g., a queue overflow due to congestion). If S does not receive an ACK packet within a timeout period, then S resorts to a recovery procedure. In our implementation of DREAM in NS-2, following the work done in [4], the recovery procedure floods the data packet to D. If D receives a flooded data packet, D does not return an ACK packet. Lastly, DREAM defines a timeout value on location information. If the location information is older than the limit specified, then S immediately resorts to the recovery procedure (i.e., flooding).

2) Implementation Decisions: In our first implementation of DREAM, the cone angle was often so small that no neighbors existed in the forwarding zone. Although it is not discussed in [4], the simulation results presented there are based on DREAM using a minimum cone angle of 30 degrees [9]. Thus, we added a minimum cone angle of 30 degrees to our implementation of DREAM in NS-2.

We evaluated all the optimizations proposed in [4] for DREAM and also evaluated other variations of the protocol in an attempt to improve the performance of the protocol. In one optimization, an MN transmits location packets (LPs) adaptively based on when the MN has moved a specified distance from its last update location. Although this optimization is proposed in [4], it is not evaluated and details on how to implement this optimization are not provided. Our solution for the transmission of LPs follows:

transmit nearby LP:	$T_{range}/\alpha * 1/\nu = T_{range}/(\alpha\nu)$
transmit faraway LP:	one for every X nearby LPs; sent at least every Y seconds

where T_{range} is the transmission range of the MN, ν is the average velocity of the MN, and α is a scaling factor. In our simulations, we set α to 10, X to 13, and Y to 23 seconds. (We optimized these three values via numerous simulation trials.) To avoid LPs being transmitted by MNs at the same time (and, thus, colliding), MNs offset the transmission of their LPs randomly.

In the performance results on DREAM presented in [4], LPs are transmitted periodically and these packets are sent to nearby MNs (100 meters or closer) at a higher frequency than

TABLE III
DREAM Constants

Minimum cone angle	30 degrees
Nearby MN defined as within	1 hop
α for nearby LPs	10
X for faraway LPs	13
Y for faraway LPs	23 seconds
LP update offset	0.01 seconds
Location table entry timeout	46 seconds
Timeout for receiving ACK	500 ms

to faraway MNs. (Note that LPs to faraway MNs also update nearby MNs.) We compared sending LPs periodically (as done in [4]) with our solution for the transmission of LPs. We found that our solution reduces the total packets transferred in the simulation by 19% and that the data packet delivery ratio of the two solutions is approximately equivalent (i.e., within 1% of each other).

In the DREAM protocol, nearby MNs are categorized by distance. A variation of the protocol is to specify nearby MNs as being within a given number of hops. We compared the performance of defining a nearby MN as being within 100 meters versus being within one hop and discovered one hop slightly improves the results of the protocol. Thus, unlike the results presented in [4], our implementation of DREAM defines nearby MNs as one hop neighbors.

In the DREAM protocol, each ACK packet is sent to the source via the DREAM protocol. We attempted to reduce the flooding of ACK packets in the forwarding zone by sending each ACK via the reverse source route, which is gathered by the data packet. While this variation of the DREAM protocol does reduce the total packets transmitted by 6% without a large decrease in the data packet delivery ratio (i.e., the decrease is only 0.8%), this variation of the DREAM protocol adds 11% to end-to-end delay. The increase in end-to-end delay occurs because a unicast ACK is less likely to be delivered than a flooded (in the forwarding zone) ACK; thus, an ACK timeout is more likely to occur. Due to the large increase in end-to-end delay, we chose to not include this variation in our simulation results. Table III lists the constants used in our implementation of DREAM.

III. SIMULATION ENVIRONMENT

Table IV lists the simulation parameters that we used along with those of [1] and [2] (the random scenarios). We compare our choices with the choices made in [1] and [2] in order to validate our choices and to illustrate the differences in these three performance investigations of ad hoc network routing protocols. Our main goal was to *stress the protocols* with high data load during both low and high speeds. Our simulation parameters accomplished this goal.

As discussed in [2], a square simulation area allows MNs to move more freely than a rectangular simulation area; however, a square simulation area results in a smaller average number of hops between the senders and receivers than a rectangular simulation area with the same area (assuming the MNs have the same transmission range). We, therefore, chose to use a rectangular simulation area.

Table IV shows that our simulation area and transmission range are smaller than those used in [1] and [2]. However, if MNs are placed uniformly in the simulation area, and if edge effects are considered (i.e., fewer neighbors exist for those MNs near an edge), then an MN in [1] has an average of 11.7 neighbors and an MN in the random scenarios of [2] has an average of 6.3 neighbors. Our simulation parameters give us an average of 7.7 neighbors.

As mentioned, MNs move according to the random waypoint model [1]. With this mobility model, there is a complex relationship between node speed and pause time. For example, a scenario with fast MNs and long pause times actually produces a more stable network than a scenario with slower MNs and shorter pause times. Figure 1 illustrates that long pause times (i.e., over 20 seconds) produce a stable network (i.e., few link changes per MN) even at high speeds [10]. In other words, even though our simulations run for 1000 seconds, the figure indicates that the network is pretty stable for all pause times over 20 seconds. Thus, we chose to keep the pause times short and to vary speed along the x-axis in all of our simulations.

In our simulations, the speed of an MN between the MN's current location and its next destination is chosen from a uniform distribution between $avg \pm 10\%$ meters per second (m/s), where avg is set to 0, 1, 5, 10, 15, and 20. For example, when our speed is set to 20 m/s, all nodes have speeds between 18 and 22 m/s. In [1], when the speed is set to 20 m/s, the average speed is only 10 m/s. Our narrow range of speeds prevents the creation of a stable "backbone" consisting of a few slowly moving MNs.

Figure 2 illustrates the average MN neighbor percentage for MNs using the random waypoint model (speed is 1 m/s and pause time is zero) as time progresses. The average MN neighbor percentage is the percentage of total MNs that are a given MN's neighbor. As Figure 2 illustrates, there is high variability during the first 600 seconds of simulation time as MNs moving with the random waypoint model initially move to (or through) the center of the simulation area. We remove this variability in our simulation results by having the MNs move for 1000 seconds of simulation time before sending any data packets. Thus, when data begins transmitting in the two reactive protocols at simulation time 1000, there is no routing state in any of the MNs. As a result, initial route request packets are flooded in the entire network for both DSR and LAR. Since DREAM is a proactive protocol, MNs using DREAM begin sending control packets at simulation time 950 seconds; thus, location information used in DREAM is propagated in the network before data packets begin transmitting. Data packets begin transmitting at 1000 seconds simulation time. Our simulations then execute for another 1000 seconds (until the simulation clock is at 2000 seconds).

Our communication model is similar to the communication model used in [1] and [2]. Specifically, we have 20 CBR (con-

TABLE IV
Simulation parameters

	in [1]	*in* [2]	*herein*
Simulator	NS2	NS2	NS2
Simulation time	900s	250s	1000s
Simulation area	1500x300m	1000x1000m	300x600m
Number of MNs	50	50	50
Transmission range	250m	250m	100m
Average neighbors	11.72	6.32	7.76
Movement model	random waypoint	random waypoint	random waypoint
Maximum speed	1 and 20 m/s	0-20 m/s	0-22 m/s
Average speed	1 and 10 m/s	not specified	0-20 m/s
Pause time	0, 30, 60, 120, 300, 600, 900 s	1 s	10 s ± 10%
CBR sources	10, 20, or 30	15	20
Data payload	64 bytes	64 bytes	64 bytes
Packet rate	4 packets/s	5 packets/s	4 packets/s
Traffic pattern	peer-to-peer	random	peer-to-peer

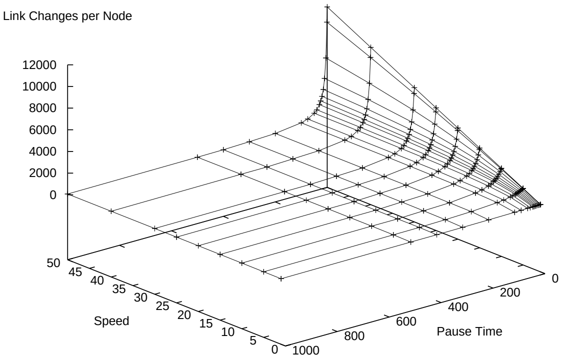

Fig. 1. Link breakage vs. speed vs. pause time.

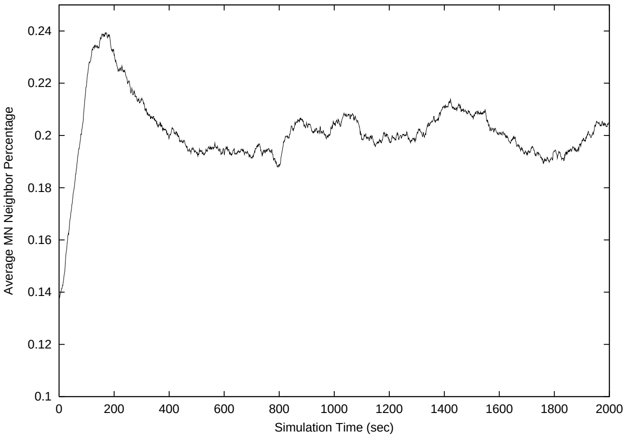

Fig. 2. Average neighbor percentage vs. time.

stant bit rate) sources sending 64 byte packets at a rate of 4 packets per second to 20 receivers. One difference between the communication models is that [2] randomly spreads the traffic among all MNs, while [1] and our simulations create peer-to-peer traffic patterns. Peer-to-peer traffic stresses the network protocols since traffic is concentrated in specific areas of the network. We avoid unnecessary contention in the transmission of packets; we offset the transmission of a data packet by 0.0001 seconds for each of the 20 peer-to-peer communication pairs.

We performed 10 simulation trials for each of six speeds. We used our generation program (*mobgen* instead of *setdest* from [8]) to generate 60 different mobility scenarios. The generation program from [8] begins the simulation with each MN stationary for pause time seconds [1]; in other words, during the first pause time seconds of the simulation, the average number of neighbors remains constant. Our generation program begins the simulation with each MN randomly selecting whether or not it is stationary or moving toward its first randomly chosen destination. In addition, speed and pause times in *mobgen* are chosen from a uniform distribution. The same 60 mobility scenarios are used to compare the different routing protocols. At zero speed, we use network configurations that occur after the MNs have moved for 1000 seconds. In other words, we first allow the static MNs to distribute in a fashion that is typical of the random waypoint model. In addition, at zero speed, we use network configurations that are not partitioned between the sources and destinations since all protocols fail when the network is partitioned.

IV. SIMULATION RESULTS

In our comparison of DSR-P (promiscuous mode), DSR-NP (non-promiscuous mode), LAR-NP Box, LAR-NP Step, and DREAM (which is, by definition, NP), we consider the following performance metrics: protocol overhead, network-wide data load, end-to-end delay, and data packet delivery ratio[1]. The data packet delivery ratio is the ratio of the number of

---

[1] In our discussions below, DSR refers to both DSR-P and DSR-NP and LAR refers to both LAR Box and LAR Step.

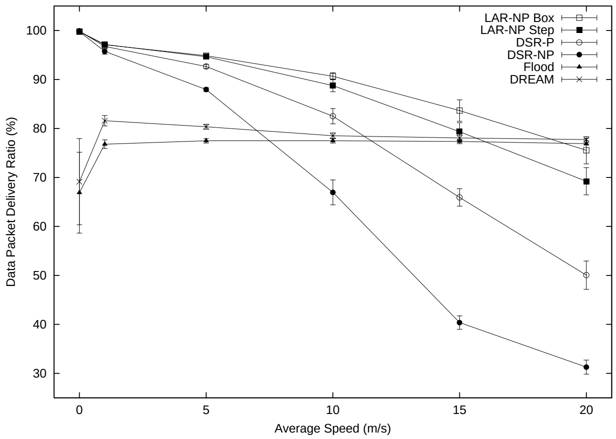

Fig. 3. Data packet delivery ratio vs. speed.

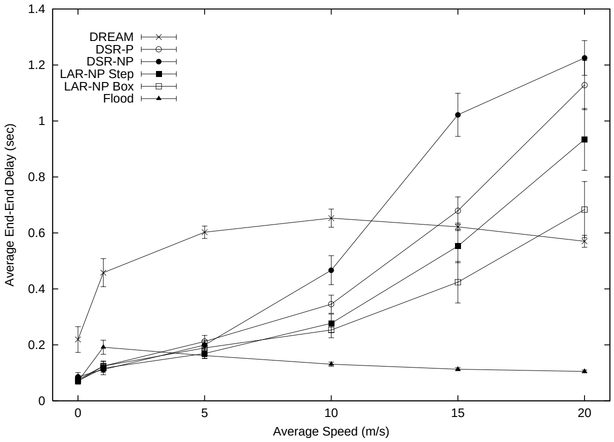

Fig. 4. End-to-end delay vs. speed.

data packets delivered to the destination nodes divided by the number of data packets transmitted by the source nodes. We compare the performance results of the five protocols with flooding every data packet in the ad hoc network (Flood), which allows us to determine quantitatively how well the five routing protocols do against a baseline case.

In our simulations, Flood and DREAM protocols have the highest average hop count: approximately 4.0 across all speeds. (The average hop count for Flood and DREAM is calculated from the first data packet to arrive at the destination.) DREAM resorts to its flooding recovery procedure often (see discussion of Figure 4); thus, the average hop count of DREAM and Flood are similar. LAR Box and LAR Step find routes in a similar fashion; thus, the average hop counts of these two protocols are nearly the same: approximately 3.5 across all speeds. Since the LAR protocols deliver a higher percentage of data packets than DSR (see Figure 3), the extra packets delivered by LAR are traveling along longer routes. As a result, the two DSR protocols have the lowest average hop count. DSR-P and DSR-NP have an average hop count of 3.0 at low speeds; at higher speeds, the average hop count for DSR-P drops under 2.8 and the average hop count for DSR-NP drops under 2.3. Which indicates that both DSR protocols have difficulty maintaining long routes at high speeds.

All the performance results presented are an average of 10 different simulation trials. We calculate a 95% confidence interval for the unknown mean, and we plot these confidence intervals on the figures. Since most of the confidence intervals are quite small (in fact, some of the intervals are smaller than the symbol used to represent the mean on our plots), we are convinced that our simulation results precisely represent the unknown mean.

A. Performance

Figure 3 illustrates the data packet delivery ratio versus speed. When speed is zero, the data packet delivery ratios for the DSR and LAR protocols are 100% and the data packet delivery ratios of the DREAM and Flood protocols are approximately 68%. 100% delivery ratio is not achieved by the DREAM and Flood protocols due to the limited buffer size and the contention and congestion in the network caused by the flooding nature of these two protocols. (See Figure 4 for a discussion on how often DREAM floods the entire ad hoc network.)

Contention and congestion also contribute to the constant data packet delivery ratio for DREAM and Flood as speed increases from 1 m/s to 20 m/s. In other words, contention and congestion, due to the flooding behavior of these two protocols, override the effect of speed.

In Figure 3 for low (or no) speed, the data packet delivery ratios of the DSR and LAR protocols are almost equivalent. As speed increases, however, the data packet delivery ratios of the two LAR protocols are higher than the data packet delivery ratios of the two DSR protocols. When a route is broken from a source to a destination in LAR, the source is able to use location information on the destination to find a new route to the destination more efficiently than DSR's route discovery method.

The data packet delivery rate decreases, as speed increases, for the DSR and LAR protocols. As speed increases, it is much more difficult to find a usable route to a destination. Figure 3 does illustrates that the use of promiscuous mode in DSR significantly aids MNs in learning useful routes.

Figure 4 illustrates the average end-to-end delay of a data packet as speed increases. (The average end-to-end delay of Flood and DREAM is calculated from the first data packet to arrive at the destination.) Since no partitions are included in the zero speed results, the DSR and LAR protocols only need to do route discovery once at zero speed. DREAM also has a good chance of sending data packets without the recovery procedure at zero speed. Thus, compared to higher speeds, all five routing protocols have a smaller end-to-end delay.

As shown in Figure 4, DREAM has the highest average end-to-end delay of all six protocols at speeds less than or equal to 10 m/s. At zero speed, location information on the MNs in DREAM is accurate; however, due to contention and congestion in the network, there is a good chance that a data packet (or an ACK packet for a data packet) does not reach its intended destination. Specifically, the DREAM recovery procedure (i.e., flooding) is used approximately 40% of the

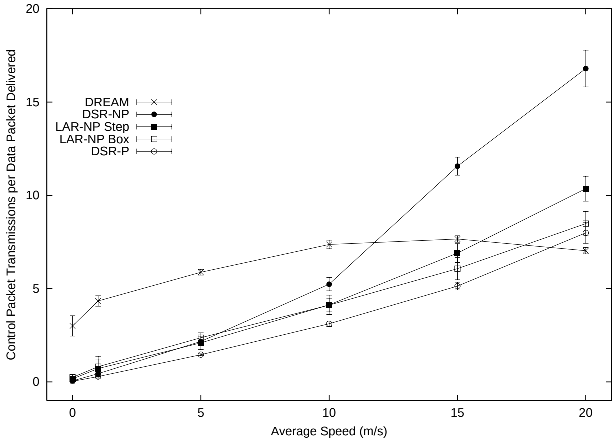

Fig. 5. Control packet overhead vs. speed.

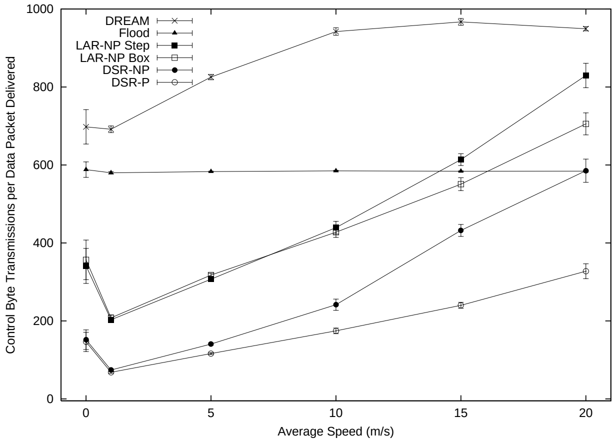

Fig. 6. Control byte overhead vs. speed.

time at zero speed. Since a source has a timeout for receiving an ACK of 500 ms in the DREAM protocol (see Table III), the end-to-end delay for DREAM at zero speed is approximately 0.2 seconds. At 1 m/s, the DREAM recovery procedure is used approximately 90% of the time and at 5 m/s and higher, the DREAM recovery procedure is used for almost every data packet transmitted.

As speed increases, more route requests are needed in DSR and LAR; thus, end-to-end delay increases with speed in both protocols. The end-to-end delays of DSR are slightly higher than the end-to-end delays of LAR since LAR is (sometimes) able to use location information to focus its search for a route to a destination. At some speeds, DSR-NP has a higher end-to-end delay than DSR-P. A route request in DSR-NP takes longer than a route request in DSR-P, since an intermediate MN in DSR-P may respond to the route request instead of the destination MN.

As shown in Figure 4, Flood has (almost) the lowest average end-to-end delay of all six protocols. At low speeds, the average end-to-end delay of Flood is equal to or higher than the average end-to-end delay of DSR and LAR. DSR and LAR spend little time on route discovery at low speeds. At high speeds, however, both DSR and LAR spend time on route discovery; thus, at high speeds, the average end-to-end delay for Flood is lower than the average end-to-end delay of DSR and LAR.

B. Overhead/Load

Figure 5 shows the number of control packet transmissions for each data packet delivered as speed increases, which helps capture the power overhead requirements of each protocol. DREAM transmits many small control packets in its exchange of location information. Since DREAM is the only protocol with a proactive element, and the only protocol that returns an ACK for each data packet that is delivered from the forwarding zone, DREAM has the highest control packet overhead at low speeds.

LAR control packet overheads are either equal to or higher than DSR-P control packet overheads. In DSR-P, an intermediate MN responds to a route request if a route is available. In LAR, based on the discussion in Section II-B.2, the route request is forwarded all the way to the destination before a response occurs. Thus, LAR has the potential of transmitting more control packets than DSR-P.

DSR-NP has higher packet overhead than DSR-P at speeds greater than 5 m/s. An MN using promiscuous mode learns new routes (which sometimes prove to be useful at a later time) from packets not addressed to it. Promiscuous mode operation is more beneficial at higher speeds; thus, the difference in packet overhead between DSR-NP and DSR-P is more pronounced at higher speeds. Higher overhead can be acceptable if the performance (e.g., the data packet delivery rate) is also higher. Figure 3 illustrates that this is not the case for DSR-NP.

The control packet overheads of the DSR and LAR protocols increase substantially as speed increases, since more route error and route request packets are transmitted at higher speeds. In DREAM, an ACK is returned by D for each copy of each data packet it receives from the forwarding zone (i.e., not from the recovery procedure). As discussed in Figure 4, the recovery procedure is not used often at low speeds; at high speeds, however, the recovery procedure is used often. Thus, the control packet overhead of DREAM increases at low speeds and decreases at high speeds (i.e., fewer ACKs are transmitted at higher speeds).

Figure 6 illustrates the number of control byte transmissions (in both control packets and data packets) for each data packet delivered as speed increases, which helps capture the bandwidth overhead requirements of each protocol. Both DREAM and Flood have high control byte overhead due to the large number of data packets both these protocols send (see Figure 7). Flood has lower control byte overhead than DREAM since Flood does not transmit any control packets; DREAM, on the other hand, transmits many (small) control packets containing location information (see Figure 5).

The control byte overheads for the two LAR protocols are higher than the control byte overheads for the two DSR protocols. In addition to transmitting as many (or more) control packets, LAR packets (both control and data) are each 36 bytes larger than DSR packets due to the location and speed

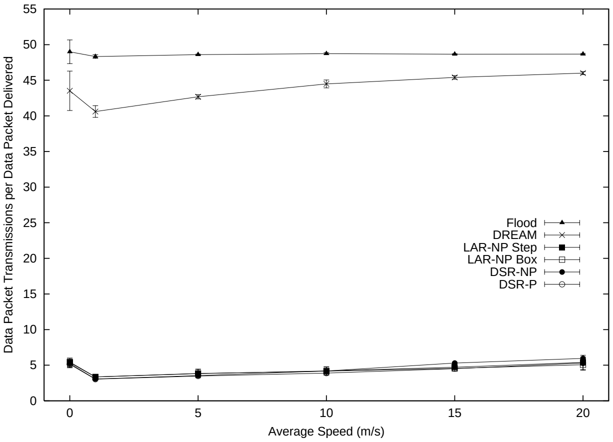

Fig. 7. Data packet load vs. speed.

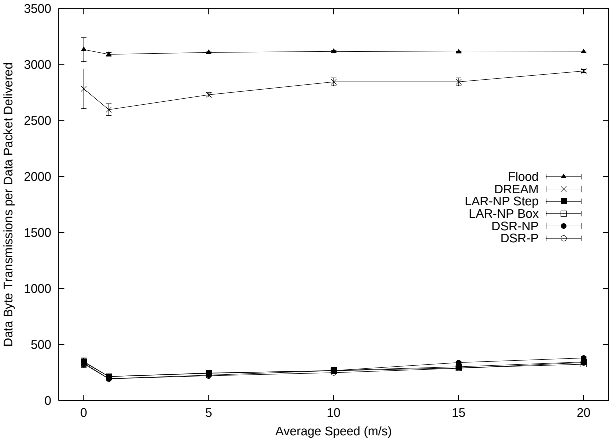

Fig. 8. Data byte load vs. speed.

information included in each packet. Furthermore, since the average number of hops is larger for the two LAR protocols, the source route included in each LAR packet is larger than the source route included in each DSR packet.

At all non-zero speeds, DSR-NP has higher control byte overhead than DSR-P due to the higher number of control packet transmissions (see Figure 5). As speed increases, the control byte overheads of both LAR and DSR increase substantially; in both cases, more route error and route request packets are transmitted as speed increases.

Figures 7 and 8 illustrate the data load of the six protocols studies as a function of node speed. In the DREAM protocol, data packets are first flooded in the forwarding zone and then possibly flooded in the entire network. In other words, the DREAM protocol never unicasts a data packet. In our investigation, the DREAM recovery procedure is called between 40-100% of the time (see discussion of Figure 4). Thus, both Flood and DREAM have extremely high data load, which is shown in Figure 7, the number of data packet transmissions per data packet delivered, and Figure 8, the number of data byte transmissions per data packet delivered. As speed increases, the data load for both Flood and DREAM remains constant due to the flooding behavior that occurs in each protocol.

Since both LAR and DSR unicast data packets, both LAR and DSR have similar data loads (see Figures 7 and 8). Lastly, since fewer data packets are delivered at higher speeds (see Figure 3), both LAR and DSR have slightly higher data load for each data packet delivered at higher speeds.

Figures 9 and 10 illustrate the total number of packets and bytes transmitted by the six protocols. These two figures capture the total network load (overhead and data) that occurs. Although Flood has zero control overhead, these two figures illustrate that it is an inefficient protocol to transmit data in terms of network bandwidth utilization and node energy usage.

V. RELATED WORK

Previous simulation results have been presented for the protocols evaluated in this paper. However, this is the first paper to provide a detailed, quantitative evaluation comparing their relative performance. In addition, none of the results previously reported on LAR and DREAM used a simulation of a complete physical layer and MAC. We implemented both LAR and DREAM in NS-2 to provide our performance investigation with a complete physical layer and MAC.

A. *Prior Results on DSR*

The results presented in [1] on DSR-P are quite different from the results presented herein. For example, all the data packet delivery ratios presented in [1] for DSR are over 95%. Their results are not comparable to ours because of the differences in our simulation environments. First, the average number of neighbors in [1] is much larger than our average number of neighbors (see Table IV). Second, the maximum average speed considered in [1] is only 10 m/s; our maximum average speed is 20 m/s. Third, the transmission range is 250m in a 1500x300m simulation area. Thus, the percentage of the simulation area that is covered by the transmission range is 43.6%. In our simulations, the percentage of the simulation area that is covered by the transmission range is only 17.4%. Lastly, the metric used for the x-axes in [1] is pause time, rather than speed. As discussed in Section III, speed has a much greater impact than pause time on link breakage rates [10].

The results presented in [2] on DSR-NP are also quite different from ours. The results in [2] evaluate pause times equivalent to 1 s; we evaluate much longer pause times uniformly chosen from 10 s ± 10%. In addition, only 15 CBR sources transmit data in [2] while 20 CBR sources transmit data herein. Lastly, results presented in [2] are taken from only 250 seconds of simulation time. As shown in Figure 2, there is high variability in the average number of neighbors during the initial seconds of simulation time for MNs using the random waypoint model. Since the authors of [2] do not present confidence intervals for the unknown mean in the random scenarios, the precision of their estimates can not be determined.

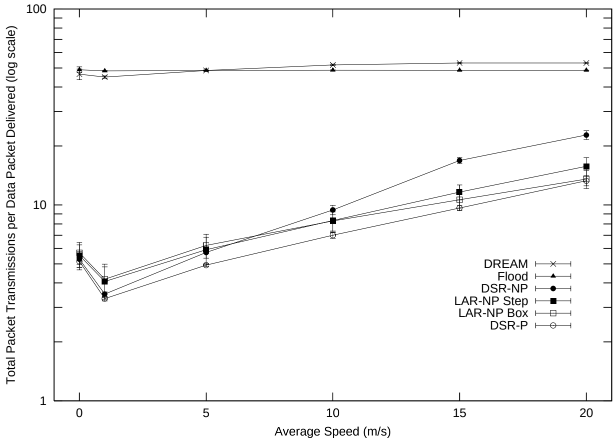

Fig. 9. Total packets transmitted vs. speed.

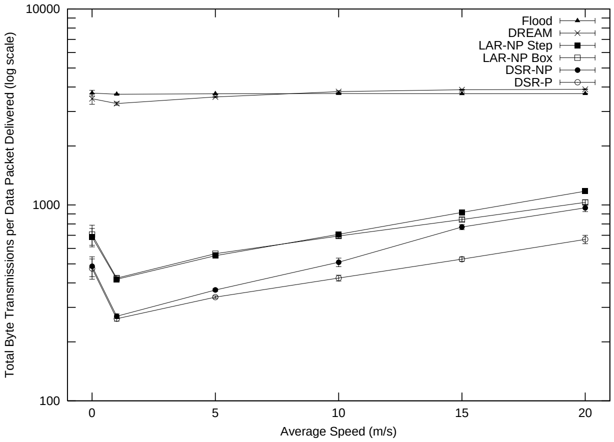

Fig. 10. Total bytes transmitted vs. speed.

B. Prior Results on LAR

Many of the performance results presented in [3] have an x-axis for the transmission range, number of MNs, or the error factor in GPS location information (which is no longer relevant). Of the results in [3] that have an x-axis of speed, only one figure has a y-axis representing a statistic that we calculate. Although the x-axis in [3] defines speed as units per second instead of meters per second, and although a complete physical layer and MAC are not simulated in [3], the results presented in Figure 6(a) in [3] are comparable to the results presented in Figure 5 herein. Specifically, as speed increases, the number of control packet transmissions per data packet delivered increases for both LAR protocols and increases more substantially for DSR-NP's method of flooding route request packets.

C. Prior Results on DREAM

The results presented in [4] are substantially different from ours. Specifically, the data packet delivery ratios presented for DREAM are all over 80% and the end-to-end delay presented for DREAM is smaller than the end-to-end delay presented for DSR. Since the simulation environment given in [4] is in clock ticks and units, it is difficult to compare their simulation environment with our simulation environment (which is in meters and seconds). There are, however, a few major differences in the simulation environments studied that are certain.

First, the results presented in [4] were obtained from the Maisie simulation package which does not offer a complete physical layer and MAC. Second, the mobility model used in [4] is a Brownian motion mobility model which creates a more stable network than the random waypoint model. Third, in [4], the transmission range is 40 units over a 100x100 unit simulation area. Thus, the percentage of the simulation area that is covered by the transmission range is 50.2%; in other words, little routing occurs in their simulation results. (As mentioned, the percentage of the simulation area that is covered by the transmission range in our simulations is only 17.4%.) Lastly, although we do not know what a clock tick is compared to a second, the 30 MNs in [4] only transmit (approximately) 1.5 to 12 packets per 300 clock ticks which we believe is a much smaller data load than the data load studied herein.

D. Prior Results on DREAM and LAR

The only prior comparison (of which we are aware) of DREAM and LAR exists in [11]. Although only three figures comparing DREAM and LAR are given in [11], only one of the three figures exist herein. The authors of [11] found, as we have (see Figure 3), that DREAM is more robust to mobility than LAR. They attribute this fact to the partial flooding of data packets that occurs in the (cone) forwarding zone; we suspect, however, that this fact is due to the flooding of data packets that occurs from the recovery procedure. The packet delivery ratios presented in [11] for LAR are similar to the packet delivery ratios presented in Figure 3. The packet delivery ratios presented in [11] for DREAM, however, are much larger than the packet delivery ratios presented herein. We attribute this difference to not having contention and congestion fully modeled in [11].

VI. CONCLUSIONS

Conclusion 1: The added protocol complexity of DREAM does not appear to provide benefits over Flood. In the DREAM protocol, each data packet is first flooded in the forwarding zone and then (possibly) flooded in the entire network through the recovery procedure. As discussed in Figure 4, the DREAM recovery procedure is used almost all the time if the MNs move; thus, the end-to-end delay of DREAM is much higher than the end-to-end delay of Flood. As illustrated in Figure 3, compared to Flood, the DREAM protocol has equivalent data packet delivery ratio for all speeds. In addition, Figures 5 through 8 illustrate that the packet and byte network load of DREAM is comparable to the packet and byte network load of Flood. Thus, there appears to be no reason to include the additional protocol complexity of DREAM over the simple protocol of Flood.

Conclusion 2: Location information improves DSR, especially at high speeds. Figure 3 illustrates that using location information improves the data packet delivery ratio of DSR significantly. In fact, since LAR-NP offers higher data packet delivery ratio than DSR-P, *the use of location information is more beneficial than the use of promiscuous mode operation.* There is a cost for this increase in data packet delivery ratio. Specifically, at 20 m/s, the two LAR protocols improve the data packet delivery ratio of DSR-P by approximately 40% and the end-to-end delay by approximately 20%. The cost for this improvement is a 15% increase in the number of packet (control and data) transmissions for each data packet delivered and a 70% increase in the number of byte (control and data) transmissions for each data packet delivered.

The performance benefits are more substantial for DSR-NP. At 20 m/s, the two LAR protocols improve the data packet delivery ratio of DSR-NP by approximately 130% and the end-to-end delay by approximately 35%. In addition, the two LAR protocols decrease the number of packet (control and data) transmissions for each data packet delivered in DSR-NP by 30%. The cost for this improvement is a 15% increase in the number of byte (control and data) transmissions for each data packet delivered. Since the cost of transmitting packets in a wireless network is much more severe than the cost of transmitting bytes, the increase in data packet delivery ratio is worth the extra overhead to include location information in DSR.

Conclusion 3: Promiscuous mode operation improves the performance of DSR significantly. As shown in Figure 3, the data packet delivery ratio for DSR-P at speeds greater than 1 m/s is significantly higher than that of DSR-NP. As shown in Figures 5 and 6, the control overhead for DSR-NP at speeds greater than 1 m/s is significantly higher than the control overhead for DSR-P. Thus, promiscuous mode operation improves the performance of DSR significantly. As discussed in Section II-B.2, in order to ensure the sources are updated with recent location information on the destinations, an intermediate MN does not respond to a route request in LAR. Thus, the benefits of promiscuous mode operation are reduced in LAR. However, LAR may still benefit from promiscuous mode operation via the propagation of location and speed information; further study on the use of promiscuous mode operation in LAR is needed in order to evaluate potential benefits.

Conclusion 4: Our implementation of DREAM provides a simple location service. Recently, a few of the location based routing protocols proposed (e.g., [5], [6]) have assumed the availability of some location service (e.g., Grid's Location Service [12]) to translate an MN's address to the MN's geographical location. The authors of DREAM proposed that an MN transmits location information adaptively based on when the MN has moved a specified distance from its last update location. Details on this location service, however, are not provided in [4]. In Section II-C.2, we propose a solution for the transmission of location information adaptively. In related work we have developed a suite of location services, including the proposal derived from DREAM, and have compared their performance and accuracy [13].

Conclusion 5: There is a tradeoff between average end-to-end delay and data packet delivery ratio. We were able to achieve (almost) 100% data packet delivery ratio for LAR at high speeds when an infinite queue of data packets is allowed. In other words, data packets are stored in a queue until a route to the destination is found. Once a route becomes available, all packets in the queue for the destination are immediately transmitted. (Although we did not test it, an infinite queue in DSR should perform in a similar manner.) If data packet delivery ratio was the only important performance metric, we would set the time to hold packets awaiting routes to infinity. In this situation, the data packet delivery ratio would be maximized at the cost of average end-to-end delay.

References

[1] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, "Multi-hop wireless ad hoc network routing protocols," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp. 85–97.

[2] P. Johansson, T. Larsson, N. Hedman, B. Mielczarek, and M. Degermark, "Routing protocols for mobile ad-hoc networks - a comparative performance analysis," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1999, pp. 195–206.

[3] Y. Ko and N.H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp. 66–75.

[4] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A distance routing effect algorithm for mobility (DREAM)," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 1998, pp. 76–84.

[5] B. Karp and H. T. Kung, "GPSR: Greedy perimeter stateless routing for wireless networks," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 2000, pp. 243–254.

[6] R. Jain, A. Puri, and R. Sengupta, "Geographical routing using partial information for wireless ad hoc networks," *IEEE Personal Communications*, pp. 48–57, Feb. 2001.

[7] D. Johnson and D. Maltz, "Dynamic source routing in ad hoc wireless networks," in *Mobile Computing*, T. Imelinsky and H. Korth, Eds., pp. 153–181. Kluwer Academic Publishers, 1996.

[8] The CMU Monarch Project, "The CMU monarch extensions to the ns simulator," URL: http://www.monarch.cs.cmu.edu/. Page accessed on January 5th, 2001.

[9] S. Basagni, "Personal communication," Nov. 2000.

[10] J. Boleng, "Normalizing mobility characteristics and enabling adaptive protocols for ad hoc networks," in *Proceedings of the 11th Local and Metropolitan Area Networks Workshop*, March 2001.

[11] M. Gerla, G. Pei, and S.-J. Lee, "Wireless, mobile ad-hoc network routing," in *Proceedings of the IEEE/ACM FOCUS'99*, 1999.

[12] J. Li, J. Jannotti, D. De Couto, D. Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proceedings of the ACM/IEEE International Conference on Mobile Computing and Networking (Mobicom)*, 2000, pp. 120–130.

[13] T. Camp, J. Boleng, and L. Wilcox, "Location information services in mobile ad hoc networks," in *International Communications Conference (ICC)*, 2002, To appear.