Purdue University

Purdue e-Pubs

Department of Computer Science Technical Reports

Department of Computer Science

2002

Performance Study on Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks

Yi Lu

Weichao Wang

Yuhui Zhong

Bharat Bhargava
Purdue University, bb@cs.purdue.edu

Report Number:
02-016

PERFORMANCE STUDY ON TWO DISTANCE VECTOR ROUTING PROTOCOLS FOR MOBILE AD HOC NETWORKS

Yi Lu
Weichao Wang
Yuhui Zhong
Bharat Bhargava

Department of Computer Sciences
Purdue University
West Lafayette, IN 47907

Performance Study on Two Distance Vector Routing Protocols for Mobile Ad Hoc Networks

Yi Lu, Weichao Wang, Yuhui Zhong and Bharat Bhargava

Abstract—We investigate via simulation the performance issues of Destination-Sequenced Distance Vector (DSDV) and Ad-hoc On-demand Distance Vector (AODV) routing protocols for mobile ad hoc networks. Four performance metrics are measured by varying the maximum speed of mobile hosts, the number of connections, and the network size. Monte Carlo simulation method is used to estimate the expectation values of performance metrics. The correlation between network topology changes and mobility is investigated by using linear regression analysis. The simulation results indicate that AODV outperforms DSDV in less stressful situations, while DSDV is more scalable with respect to the network size and traffic load. It is observed that network congestion results in more than half of the dropped packets for both protocols. Our observation indicates that always sending packets through the shortest routes might cause congestion in sparse networks, but rarely in dense ones. The investigation demonstrates that reducing broadcast intervals from 15 seconds to 8 seconds improves the throughput of DSDV by about 10%, and shows that using longer packet queues does not help.

Index Terms—ad hoc networks, distance vector, routing protocol, performance, simulation

I. INTRODUCTION

The high mobility, low bandwidth, and limited computing capability characteristics of mobile hosts make the design of routing protocols challenging. The protocols must be able to keep up with the drastically and unpredictably changing network topology, with minimized message exchanges, in a computation efficient way.

The routing protocols may be categorized as *proactive*, *on-demand*, and *hybrid*, according to the way the mobile hosts exchange routing information. The proactive protocols include DSDV [1] and Source Tree Adaptive Routing (STAR) [2], which disseminate routing information among all the hosts in the network periodically, so that

every host has the up-to-date information for all possible routes irrespective of the need of any such route. On-demand routing protocols operate on a need basis, instead of assuming a uniform traffic distribution within the network and maintaining routes among all hosts at all times. An on-demand protocol discovers and maintains only active routes that are currently used for delivering data packets. If this is done intelligently, it utilizes network bandwidth more efficiently, at the cost of increased route discovery latency. AODV [3] and Dynamic Source Routing (DSR) [4] are representatives of on-demand protocols. Hybrid routing protocols, such as Zone Routing Protocol (ZRP) [5] and Core Extraction Distributed Ad Hoc Routing (CEDAR) [6], maintain a virtual routing infrastructure, apply proactive routing mechanisms in certain regions of a network and on-demand routing in the rest of the network. In addition to the tradeoff between the latency and the bandwidth usage, hybrid routing protocols are usually designed to support the security or Quality of Service (QoS).

Performance is critical in judging the merit of a routing protocol. Although there is little underlying theory that is actually of any use in formal performance analysis, we can give rules of thumb gained from simulations. An ad hoc routing protocol tends to be well-suited for some network contexts, yet less suited for the others [7]. A better understanding of the advantages and disadvantages of proactive and on-demand approaches in various network contexts will serve as a cornerstone for the development of new adaptive routing protocols. DSDV and AODV are investigated for this purpose. Both protocols utilize distance vector coupled with destination sequence number, and choose routes in the same matter. They are differentiated by the way in which they operate (i.e., proactive versus on-demand). Studying these two protocols gives insights into the differences between proactive and on-demand approaches. This analysis provides guidelines to improve these two specific protocols as well.

The rest of the paper is organized as follows. The related work on performance comparison of ad hoc routing protocols is briefly outlined in section II. Section III dis-

cusses in detail of the DSDV and AODV protocols. Section IV describes the simulation environment, including the mobility, traffic, and energy models. Section V introduces the design of experiments. The network context parameters, the performance metrics, and the performance evaluation methodology are discussed. The experiment results and analysis are presented in section VI. Section VII discusses the improvements of DSDV. Section VIII concludes this paper.

II. RELATED WORK

Several simulation-based performance comparisons have been done for ad hoc routing protocols in the recent years. Das et al. [8] evaluate performance of ad hoc routing protocols based on the number of conversations per mobile node using Maryland Routing Simulator (MaRS). The performance comparison of two on-demand routing protocols: DSR and AODV is presented in [9], using ns2 (network simulator) [10] for the simulation. The pause time and the offered traffic load are taken as parameters. In [11], GloMoSim [12] is used for the performance study of the STAR, AODV, and DSR routing protocols, taking the pause time as the parameter. The authors point out that simulating the same protocol in different simulators may produce differences in the results. The performance of two location-based routing protocols for ad hoc networks is investigated by using ns2 and the effect of average moving speed in different scenarios is presented in [13]. An adaptive distance vector routing algorithm is proposed in [14], and its performance, compared with AODV and DSR, is studied. The offered traffic load and the simulation time are the input parameters.

Compared with other research efforts, we study the performance of two distance vector routing protocols in a wide range of network contexts with varied network size, mobility, and traffic load. In addition to comparing the performance of protocols, we investigate the characteristics of proactive and on-demand approaches. To our knowledge, this paper is the first to take the power consumption as a performance metric.

III. DISTANCE VECTOR ROUTING PROTOCOLS FOR AD HOC NETWORKS

In a *distance vector* routing protocol, every host maintains a routing table containing the distances from itself to all possible destinations. Each routing table entry contains two parts: the preference outgoing host to use for the destination, and the distance to the destination. The distance metric might be the number of hops, the delay, the quality of links along the path, etc. The chosen next hops lead to the shortest path to the destination.

A. Destination-Sequenced Distance Vector (DSDV)

DSDV routing protocol is one of the first routing protocols designed specially for ad hoc networks. It extends the basic Bellman-Ford mechanism by attaching a sequence number, which is originated by the destination, to each distance. This destination sequence number is used to determine the "freshness" of a route. Routes with more recent sequence numbers are preferred for making packet forwarding decisions by a host, but not necessarily advertised to other hosts. For routes with the equal sequence number, the one with the smallest distance metric is chosen. Each time a host sends an update to its neighbors, its current sequence number is incremented and included in the update. The sequence number is disseminated throughout a network via update messages. The DSDV protocol requires each host to periodically advertise its own routing table to its neighbors. Updates are transmitted immediately when significant new routing information is available. Routes received in broadcasts are used to update the routing table. The receiver adds an increment to the metric of each received route before updating.

In DSDV, the broken link may be detected by the layer-2 protocol, or may be inferred if no broadcast has been received from a former neighbor for a while (e.g., three periodic update periods). A broken link is assigned a metric of ∞ (i.e., a value greater than the maximum allowed metric). When a broken link to a next hop is detected, the metric of any route through that next hop is immediately assigned ∞, and the sequence number associated with it is incremented. Such modified routes are immediately broadcast in a routing update packet. Handling broken links is the only situation when a sequence number is generated by a host other than the destination. To distinguish this situation, sequence numbers generated by the originating hosts are even numbers, while sequence numbers generated to indicate the ∞ metric are odd numbers. Any *real* sequence number will supersede an ∞ metric.

Two types of updates are defined in DSDV protocol. One, called "full dump", carries all the available routing information. The other, called "incremental", carries only information changed since the last full dump. Full dumps are generated relatively infrequently. If the size of an incremental approaches the size of a packet, a full dump can be scheduled so that the next incremental will be smaller.

Since all mobile hosts periodically advertise their routing information, a host can almost always locate every other host when it needs to send out a packet. Otherwise, the packet is queued until the routing information is available. DSDV guarantees loop-free paths to each destination [1].

B. Ad hoc On-demand Distance Vector (AODV)

AODV routing protocol is also based upon distance vector, and uses destination sequence numbers to determine the freshness of routes. It operates in the on-demand fashion, as opposed to the proactive way of the DSDV protocol. AODV requires hosts to maintain only active routes. An *active route* is a route used to forward at least one packet within the past *active timeout* period. When a host needs to reach a destination and does not have an active route, it broadcasts a Route Request (RREQ), which is flooded in the network. A route can be determined when RREQ is received either by the destination itself or by an intermediate host with an active route to that destination. A Route Replay (RREP) is unicast back to the originator of RREQ to establish the route. Each host that receives RREQ caches a route back to the originator of the request, so that RREP can be sent back. Every route expires after a predetermined period of time. Sending a packet via a route will reset the associated expiry time.

Every host monitors the link status of next hops in active routes by listening for "Hello" messages from its neighbors or for any suitable link layer notification (such as those provided by IEEE 802.11). When a link break in an active route is detected, a Route Error (RERR) is sent back along the path to the source. All hosts on that path notice the loss of the link. In order to report errors, every host maintains a *precursor list* for each route, containing the neighbors that are likely to forward packets on this route.

To prevent unnecessary network-wide dissemination of route request messages, the source may use an *expanding ring search* technique as an optimization. The search range is controlled by the time-to-live (TTL) field in the IP header of the RREQ packet. The search process is repeated with an incremented TTL (thus expanding the ring) until a route is discovered.

Another optimization is *local repair*. When a broken link in an active route is detected, instead of sending back RERR, the host first tries to locally repair the link by broadcasting RREQ for the destination. Although local repair is likely to increase the number of deliverable data packets, it may result in increased delay as well.

IV. SIMULATION ENVIRONMENT

We use ns2 (Network Simulator) for our simulation study. Each mobile host uses an omni-directional antenna having unity gain. The wireless interface works like the 914 MHz Lucent WaveLAN Direct-Sequence Spread-Spectrum (DSSS) radio interface [15]. WaveLAN is modeled as a shared-media radio with a nominal bit rate of 2 Mb/s, and a nominal radio range of 250m

[9]. The IEEE 802.11 Distributed Coordination Function (DCF) is used as the MAC layer protocol. A unicast data packet destined to a neighbor is sent out after handshaking with Request-To-Send/Clear-To-Send (RTS/CTS) exchanges and followed by an acknowledgement (ACK) packet. The broadcast packets are simply sent out without handshake and acknowledgement. The implementation uses Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA).

The DSDV protocol used in the simulation is an extension of CMU implementation with bugs fixed. The implementation closely matches the specifications [1]. An incremental update (or triggered update) is upgraded to a full update if one third of the routing entries are required to be advertised. The major constants are given in table I.

TABLE I
DSDV CONSTANTS

Parameter	Value
minimum time between triggered updates	1 sec
periodic update interval	15 sec
allowed update loss from a neighbor	3
packet queue length per routing entry	5

The AODV implementation is provided by ns2, which is according to the specifications [3]. This implementation enables expanding ring search and local repair. The primary constants are listed in table II.

TABLE II
AODV CONSTANTS

Parameter	Value
active route timeout	50 sec
reverse route lifetime	5 sec
initial TTL	5
increment TTL	2
hello message interval	1 sec
allowed hello loss	3
packet queue length per routing entry	64
maximum time to buffer a packet	30 sec

Mobility, Traffic, and Energy Models

Mobility: We use the *random waypoint* model [16] to generate movements of mobile hosts. At the beginning of a simulation, mobile hosts are randomly placed on a square field of 1000m x 1000m. Each host randomly chooses its destination in the field, and a moving speed

that ranges from 0 to the given maximum speed. All destinations and speeds are independent and identically distributed. Every host repeats the above step after it has reached the destination and waited a specified time (the pause time). According to this model, the speed and direction of the next movement have no relation to those of the previous movement. The mobility is represented by the maximum speed and the pause time in this model.

Traffic: The constant bit rate (CBR) traffic is used in the simulation. Each connection is specified as a Source-Destination (S-D) pair. For each S-D pair, the source is randomly chosen from all hosts, and the destination is randomly chosen from all hosts other than the source. The S-D pairs are mutually independent. The packet sizes are fixed at 512 bytes. The packet sending rate is 4 packets per second. Each connection starts at a time chosen randomly from 0-100 seconds, and ends when the simulation ends.

Identical mobility and traffic scenarios are used for both protocols.

Energy: Every host has an initial energy level value at the beginning of a simulation. For every transmission and reception of packets, the energy level is decremented by a specified value, which represents the energy usage for transmitting and receiving. When the energy level goes down to zero, no more packets can be received or transmitted by the host. According to the manufacturer specifications [15], the power requirements of the WaveLAN card are shown in table III, column 2. Column 3 shows the power requirements measured in [17], without any power management. In the simulations, we use the values in column 3. We let the initial energy of each host to be 4000 joules so that the energy level does not reach zero in the simulation period.

TABLE III

POWER REQUIREMENTS OF THE LUCENT WAVELAN PCMCIA WIRELESS ETHERNET CARD

State	Documented Requirements	Measured
suspended	0.00 W	0.00 W
receiving	1.48 W	1.52 W
transmiting	3.00 W	3.10 W

V. DESIGN OF EXPERIMENTS

A. Network Context Parameters

To comprehensively measure the performance of a protocol, various networking contexts must be considered. The following parameters are varied in our simulations:

- *Network Size* is measured as the number of mobile hosts. Since the simulation field is fixed, the network size also measures the density of mobile hosts. It affects network connectivity, which represents the average degree of a host (i.e., the average number of neighbors of a host).
- *Host Mobility* is determined by the maximum speed at which a host moves, and by the pause time between two movements.
- *Traffic Load* is the number of the CBR connections.

The network size, the mobility, and the number of connections are independent variables. Using many values for each could result in a voluminous set of combined values. Hence, we choose a *moderate value* for each parameter. When one parameter is being varied, the other two are assigned the moderate values, listed in table IV.

TABLE IV

MODERATE VALUES

Parameter	Value
Maximum Speed	4 m/s
Pause Time	10 seconds
Network Size	30 hosts
Number of Connections	number of hosts

B. Performance Metrics

The following four quantitative metrics, applicable to any routing protocol, are used to assess the performance:

1) *Packet Delivery Ratio:* The ratio of the data delivered to the destinations (i.e., throughput) to the data sent out by the sources. Packets may be dropped for three reasons: the packet buffer is full when the packet arrives, the link is broken when the packet is being transmitted, or the retransmission time exceeds the limit at MAC layer.

2) *Average End-2-end Delay:* The average time it takes for a packet to reach the destination. It includes all possible delays in the source and each intermediate host, caused by routing discovery, queueing at the interface queue, transmission at the MAC layer, etc.

3) *Normalized Protocol Load:* The routing load per unit data successfully delivered to the destination. The routing load is measured as the number of protocol messages transmitted hop-wise (i.e., the transmission on each hop is counted once). A unit data can be a byte or a packet. If data packets and routing packets have a similar size, the packet-based measurement reflects the real communications load

better than the byte-based one, because extra handshake and acknowledgement bytes are required for each packet transmitted. If the packet sizes vary over a wide range (e.g., from 10 bytes to 1400 bytes), the byte-based measurement is more accurate.

4) *Normalized Power Consumption*: The energy consumed by the communications per packet received at the destinations. We measure the power consumption because power is one of the precious commodities in mobile communications. Wireless devices may consume over 50% of total system power for current handheld computers, and up to 10% for high-end laptops [17]. This poses challenging demands on the design of power-efficient routing protocols.

C. Performance Evaluation Methodology

The performance of routing protocols is very sensitive to the movements of mobile hosts. Even the maximum speed and the pause time are fixed, two scenarios generated by the random waypoint model may differ significantly. Let Δt be a time unit small enough such that the speed ($\vec{v}$) of a host is constant in Δt. If the simulation time is m*Δt, the motion of a host i is uniquely determined by a random vector $\vec{X}_i = (\vec{v}_1, \vec{v}_2, \ldots, \vec{v}_m)$. The motions of n mobile hosts in the simulation study are also denoted as a random vector $\mathbf{X} = (\vec{X}_1, \vec{X}_2, \ldots, \vec{X}_n)$, which has a joint density function $f(\vec{x}_1, \vec{x}_2, \ldots, \vec{x}_n)$. Once all other parameters (such as traffic, the number of hosts, the simulation time, etc.) are fixed, the performance of a routing protocol is a n-dimensional function $g(\mathbf{X})$. We are interested in determining the expectation value of $g(\mathbf{X})$.

$$
\begin{aligned}
\theta &= E[g(\mathbf{X})] \\
&= \int \ldots \int g(\vec{x}_1, \ldots, \vec{x}_n) f(\vec{x}_1, \ldots, \vec{x}_n) d\vec{x}_1 \ldots d\vec{x}_n
\end{aligned}
$$

Because it is difficult to compute the preceding integral exactly and $g(\mathbf{X})$ is unknown, we approximate θ by means of simulation. We use the *Monte Carlo simulation* approach [18] to estimate $E[g(\mathbf{X})]$. We start by generating a random vector $\mathbf{X}^{(1)}$ with density f, and then compute $Y_1 = g(\mathbf{X}^{(1)})$. Then, we generate a second random vector $\mathbf{X}^{(2)}$, with density f and independent of the first random vector, and then compute $Y_2 = g(\mathbf{X}^{(2)})$. We repeat this r times, and get a series of random variables $Y_i = g(\mathbf{X}^{(i)})$, $i = 1, \ldots, r$. By the strong law of large numbers, we get

$$
\lim_{r \to \infty} \frac{\sum_{i=1}^r Y_i}{r} = E[g(\mathbf{X})] = \theta
$$

Thus, the average value of all the Y_i's can be treated as an estimate of θ.

In our simulation study, five scenarios are generated using the random waypoint model for each experiment, and the average value of the performance metric is used for analysis of the results. Please note that we use only five scenarios since the large memory requirement and long simulation time of ns2 prevented us from simulating more scenarios.

VI. RESULTS AND ANALYSIS

A. Correlation between Topology Changes and Mobility

Ad hoc routing protocols are designed to adapt to topology changes. The performance of a routing protocol is effected by the topological rate of change (i.e., the speed at which a network's topology is changing). It is difficult to control the topology change directly in simulations. If we are able to estimate the correlation between the topology changes and the mobility, the topology changes can be indirectly controlled with high accuracy by varying mobility. The topology changes can be represented as link changes or route changes. Our study demonstrates that:

- The link changes and route changes can be perfectly fitted into linear functions of the maximum speed when the pause time is fixed to 10 seconds.
- The link changes and route changes can be perfectly fitted into linear functions of the pause time when the maximum speed is fixed to 4 m/s.

As shown in figure 1, the maximum speed is treated as the predictor variable, and link changes and route changes as the response variables (with the pause time to be 10 seconds). The fitting curve is obtained by using linear regression with least squares [19].

$$
\begin{aligned}
\hat{Y} &= b_0 + b_1 X \\
b_1 &= \frac{\sum_{i=1}^n X_i Y_i - n\overline{XY}}{\sum_{i=1}^n X_i^2 - n\overline{X}^2} \\
b_0 &= \overline{Y} - b_1\overline{X}
\end{aligned}
$$

If we assume that the variations of the sample points about the line are normal, we can test the null hypothesis:

$$
H_0 : b_1 = 0
$$

using the *t-test* [19].

$$
\begin{aligned}
t &= \frac{b_1 \sqrt{\sum_{i=1}^n (X_i - \overline{X})^2}}{\hat{\sigma}} \\
\hat{\sigma}^2 &= \frac{\sum_{i=1}^n (Y_i - \hat{Y}_i)^2}{n - 2}
\end{aligned}
$$

For the link changes versus the maximum speed, $|t| = 24.1445$. For the route changes versus the maximum

speed, $|t| = 21.1927$. Both of them exceed the appropriate critical value of $t_{0.995}(10) = 3.169$ [1] (because 12 sample points are used for the linear regression, the degree of freedom is $10 = 12 - 2$). Thus the hypothesis H_0 that linear relationships between the link changes and the maximum speed, the route changes and the maximum speed does not exist is rejected with 99% confidence. The dotted lines in figure 1 indicate the confidence interval of 95%. In plain words, the values of the link changes and the route changes lie within the specified intervals, respectively, and the statement is made with 95% confidence.
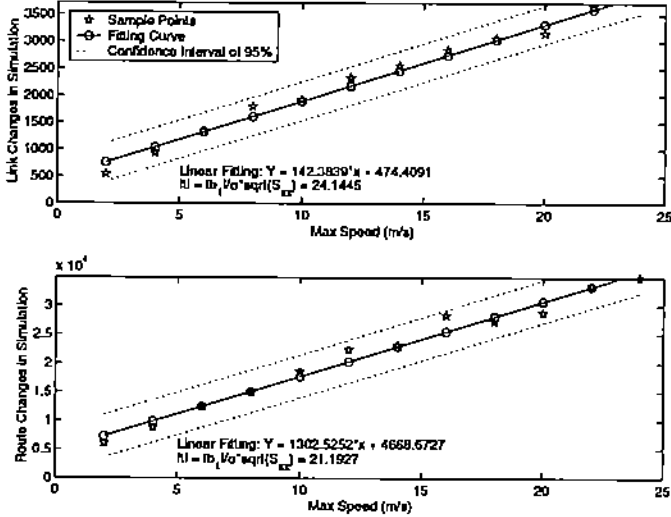

Fig. 1. Topology Changes vs. Maximum Speed

Figure 2 shows the linear regressions of the link changes versus the pause time and the route changes versus the pause time. H_0 hypothesis is also verified with t-test. Because only 6 sample points are used, the degree of freedom is 4. $t_{0.995}(4) = 4.604$, while the observed $|t|$ is 9.1826 and 8.0857 respectively. Thus H_0 is rejected with 99% confidence as well. The dotted lines in figure 2 show the confidence intervals of 95%.

B. Varying Maximum Speed

This set of experiments studies the impact of the maximum speed on the performance metrics. The number of mobile hosts and the number of connections are both 30. The maximum speed ranges over {4, 8, 12, 16, 20, 24}.

Delivery Ratio: As figure 3a shows, the packet delivery ratios for both protocols are less than 50%. When mobility is low (i.e., when the maximum speed is 4 m/s), AODV is able to deliver about 43% of total generated packets, and DSDV about 34%. As the mobility increases, the delivery

[1]The percentage points for the t-distribution are obtained from [19], using the two-tailed table.
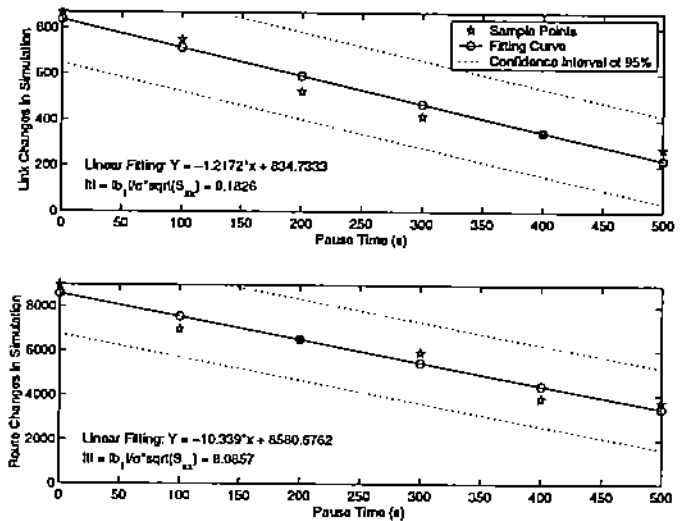

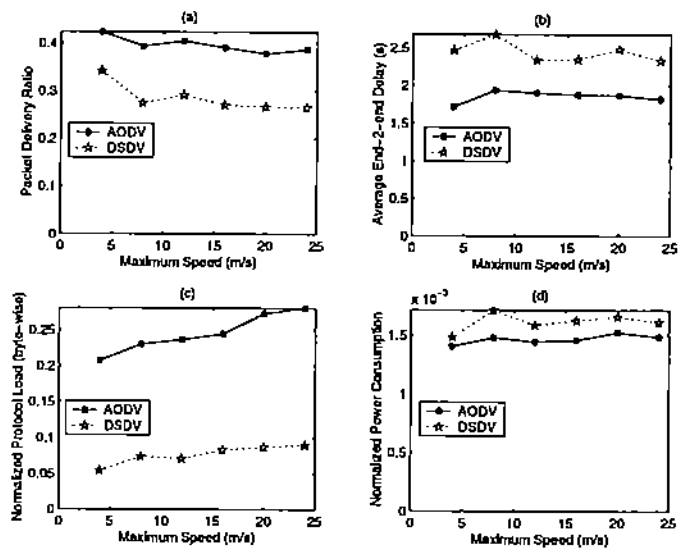Fig. 2. Topology Changes vs. Pause Time

Fig. 3. Performance vs. Maximum Speed

ratios of both protocols drop gradually, but DSDV has a little bigger drop. When the maximum speed reaches 24 m/s, the delivery ratios of AODV and DSDV are about 39% and 27% respectively.

Average End-2-end Delay: The maximum speed of 8 m/s is a turning point for both protocols, as shown in figure 3b. From 4 m/s to 8 m/s, the delay slightly increases. As the speed continues to increase, the delay tends to decrease very slowly. The delay of AODV is less effected by the maximum speed. It changes from about 1.7 seconds to 1.9 seconds, and then back to 1.8 seconds. For DSDV, the highest delay is about 2.7 seconds when the maximum speed is 8 m/s, the lowest delay is about 2.3 seconds when the maximum speed is 12, 16 and 24 m/s.

It is interesting that DSDV has a higher delay than

AODV in all cases, which seems to contradict to the obvious fact that it usually takes AODV more time to discover a route. This results from the implementations of the protocols. Although both implementations apply the drop-tail approach for packet queues, AODV poses a limit on the time a packet can be queued (table II), which currently is 30 seconds. DSDV keeps packets in queues no matter how long they have stayed, while AODV drops those have stayed for more than 30 seconds.

Normalized Protocol Load: Our simulation study (not shown in figure 3) indicates that AODV has a high packet-wise normalized protocol load, with about 2.5 to 3 protocol packets sent or forwarded per received data packet. For DSDV, the corresponding number is 0.25 to 0.3. The reason is that a DSDV protocol packet contains many routes, while an AODV protocol packet contains at most one route (e.g., RREQ).

We compare only the byte-wise protocol load in the rest of this paper. DSDV introduces a significantly (3-4 times) lower protocol load than AODV (Figure 3c). Both protocols generate slightly more routing load as the maximum speed increases.

Normalized Power Consumption: As illustrated in figure 3d, the normalized power consumptions for both protocols are rather stable. DSDV consumes more power per packet, even though DSDV introduces a much lower protocol overhead. Actually, three kinds of packets contribute to the total power consumption as shown below.

$$P_{total} = P_{protocol} + P_{delivered} + P_{dropped}$$

AODV "wins" due to the way it handles link breaks. When a broken link of a route is detected, a RERR packet is sent to the source. Every host along the path notices the broken link immediately, and drops or queues packets locally. Even though DSDV treats a broken link as a significant routing information and triggers a routing update, there is a minimum time interval between two triggered updates (currently 1 seconds as specified in table I). On average, information about a broken link is delayed 0.5 second at each host. In the meantime, those hosts that have not received information keep sending packets, which will be dropped eventually, to their next hops. Sending these packets unnecessarily consumes a remarkable amount of power.

Summary of Impact of Mobility on Performance: With moderate network size and traffic load, the performance of DSDV and AODV is relatively stable as the mobility increases. AODV outperforms DSDV in all performance metrics except for the normalized protocol load. DSDV and AODV have similar performance in power consumption regardless of mobility.

C. Varying Number of Connections

The next set of experiments demonstrates the effect of the traffic load, which is measured as the number of connections. We choose the moderate network size and mobility (i.e., 30 mobile hosts, 4 m/s maximum speed, and 10-second pause time). The number of connections varies from 10 to 80, increasing 10 each time. We use CBR with-
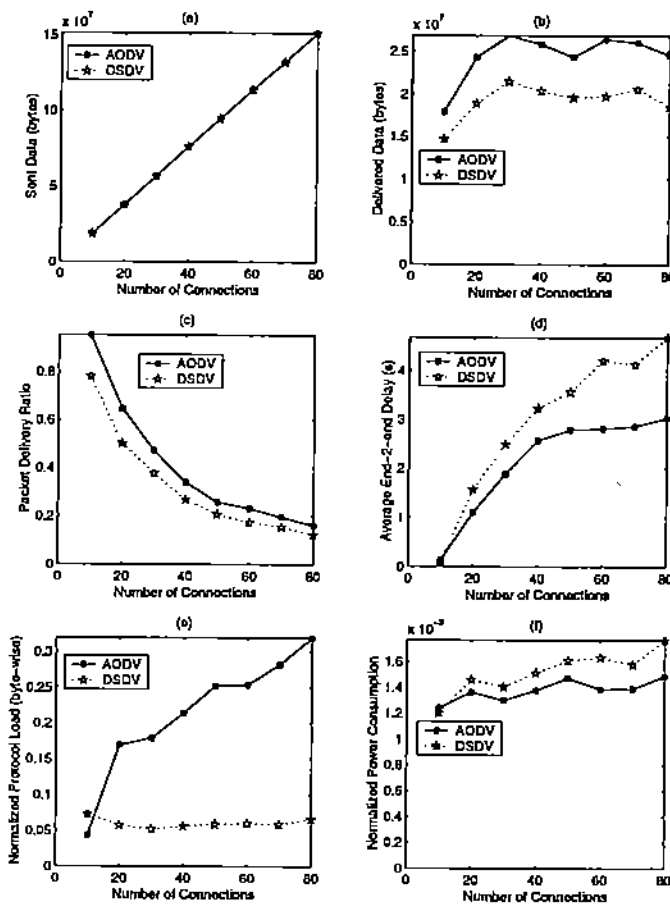

Fig. 4. Performance vs. Number of Connections

out retransmission, so the traffic load is simply the combined sending rate of all connections. Since all connections use the same rate to send data packets, the load linearly increases with the number of connections as shown in figure 4a.

Throughput and Delivery Ratio: DSDV's throughput starts saturating at 30 connections (figure 4b). Although AODV's throughput is higher than that of DSDV's, it also starts to saturating at 30 connections. The delivery ratio of AODV (figure 4c) drops dramatically from more than 90% to about 28% when the number of connections increases from 10 to 50, while that of DSDV drops from about 80% to about 20%. For more than 50 connections, the ratios of both DSDV and AODV drop more gradually because the network has already been fully loaded.

Average End-2-end Delay: As figure 4d shows, for 10 connections, DSDV and AODV have similar delay, which is about 0.1 second. We observe that the delays for both protocols increase rapidly with the number of connections (from about 0.1 second to 3 and 2.5 seconds for 80 connections, respectively). This is due to the high level of traffic congestion at certain regions of the ad hoc networks. The network congestion occurs because:

- The ad hoc network has a dynamic topology so that any mobile host may become a bottleneck.
- The CBR traffic is unresponsive to congestion.
- Both DSDV and AODV take the hop count as the metric of a route, and neither of them has any mechanism for choosing routes in such a way that the data traffic can be more evenly distributed in the network.

After the number of connections reaches 40, an average delay of AODV grows more slowly, while that of DSDV grows as fast as before. One possible reason is that AODV utilizes priority queues, in which higher priority is given to protocol packets. Thus, a protocol packet is always handled before any data packet even if it arrives later than data packets. DSDV does not distinguish protocol packets and data packets at the queue level so that packets are handled in the order in which they arrive. When the network is heavily loaded, it may take more time for DSDV to build a route.

Normalized Protocol Load: As figure 4e shows, for DSDV, the number of protocol packets is determined mostly by the network size and mobility. The effect of the number of connections can be almost ignored. It explains why the normalized protocol load stays fairly stable at 0.06 with an increasing number of connections. The normalized protocol load even drops a little bit from 10 connections to 30 connections because of the increase of throughput. The normalized protocol load of AODV increases sharply as the number of connections increases. AODV performs better than DSDV at 10 connections, which agrees with the original intention of the design of on-demand routing protocols. At 80 connections, the protocol load for AODV is about 4 times higher than for DSDV. The bad performance of AODV results from the following factors:

- As the number of connections gets larger, every host should become a source of one or more connections, if sources are independent and identically distributed among all hosts. Since each host discovers routes individually, more RREQ packets are broadcast.
- Unicasting RREP to the origination of the RREQ prevents valuable routing information from being propagated to other hosts.
- AODV treats network topology as a directed graph.

It might need to discover two different directions for the same path twice due to a short reverse route lifetime.

Normalized Power Consumption: As shown in figure 4f, DSDV consumes on average more power than AODV except for 10 connections. The normalized power consumptions for both protocols increases gradually from 10 connections to 80 connections (the increase is about 50% for DSDV, and about 25% for AODV).

Summary of Impact of Number of Connections on Performance: As shown in figure 4, AODV performs very well when the traffic load is low. It is at least as good as DSDV for all measured performance metrics at 10 connections. The performance worsens significantly for both protocols as the traffic load increases. The normalized protocol load of DSDV is more stable than that of AODV for the growing number of connections. A stable normalized protocol load is a desirable property for the scalability of a routing protocol, because it indicates that the actual protocol load linearly increases with the effective traffic (i.e., with successfully delivered data).
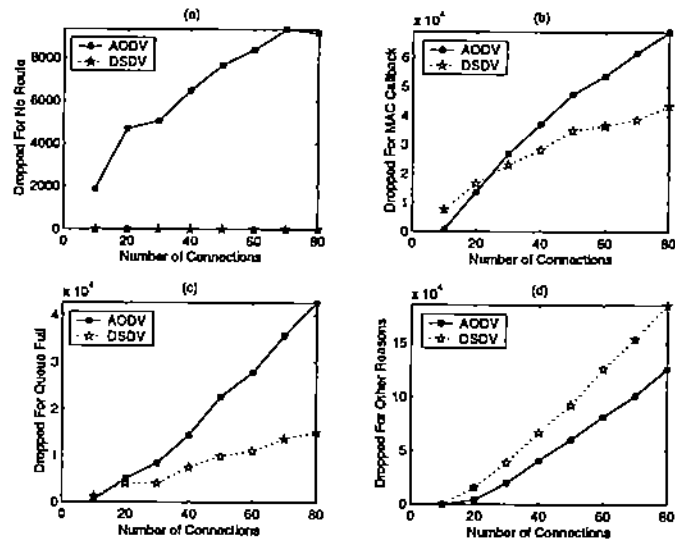

Fig. 5. Dropped Packets vs. Number of Connections

Dropped Packets: Since the delivery ratio drops dramatically with an increase in traffic load, we are interested in investigating the reasons for packets being dropped. We check this by studying the ns2 trace files.

Figure 5 shows the number of packets dropped for four reasons. The "Other Reason" includes the following that are not specified in the ns2 trace file:

1) MAC is in the idle mode or, the host is sleeping when a packet arrives.
2) Energy reaches zero.
3) Retransmission time exceeds the limit in CSMA.

Because the simple energy model is used in the simulations and the initial energy level is high enough, the cases 1 and 2 did not happen. The case 3 is mainly caused by congestion at the MAC layer. When a collision is detected by CSMA, if the retransmission time is within the limit, CSMA does a exponential backoff. Otherwise, the packet is dropped with the reason set to "other reason".

For DSDV, no packet is dropped due to "no route" to the destination. It is guaranteed by design and by the implementation of the protocol. For AODV, the number of packets dropped due to "no route" increases from 2000 to 10000, as shown in figure 5a. These packets are dropped when they have stayed in queues longer than the maximum time to buffer a packet, which is specified in table II. It testifies to the statement on the average delay made in section VI-B from another perspective. If there was no limit on the time a packet can be buffered, the expectation value of delay could be approximated as follows,

$$E^{new}[delay] = \frac{\alpha * E[delay] + \beta * T}{\alpha}$$

where α is the number of delivered packets, β is the number of dropped packets for no route, T is the maximum time a packet can be buffered, $E[delay]$ is the average delay obtained from the simulation, and $E^{new}[delay]$ is the approximate average delay without time limit. We name $E^{new}[delay]$ as *compensated delay*. Figure 6 shows that the compensated delay for AODV is higher than the average delay of DSDV, same values for DSDV are shown in figure 4d.
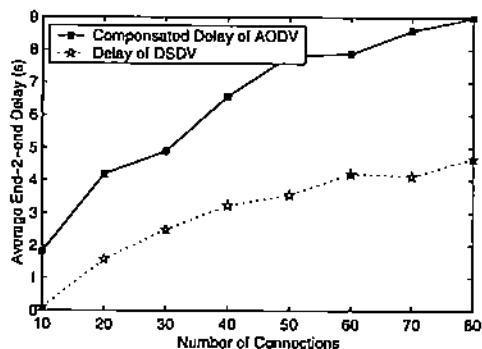

Fig. 6. Delay Compensation of AODV

As figure 5b and 5c show, for 10 connections, AODV almost does not drop packets due to a MAC callback (i.e., the next hop is not a neighbor now), or queue being full. However, the number of packets dropped for AODV increases with the number of connections at a rate higher than DSDV. DSDV drops fewer packets than AODV for above two reasons in most cases except for a low traffic load.

From figure 5, we can calculate that more than half of the dropped packets result from "other reasons" (which includes only congestion). DSDV performs better for the first three reasons, but worse than AODV for avoiding congestion. Although both DSDV and AODV do not utilize any congestion control or avoidance mechanism to balance traffic load, AODV in fact distributes the data traffic more evenly in the network. AODV tries to build the shortest route when it originates a request, but it keeps the built route as long as the route does not break, even if a shorter route is available at a later time. In contrast, DSDV tends to always send packets via the shortest routes. Forwarding packets through the shortest routes will likely push traffic to several heavily burdened hosts and congest the network.

D. Varying Number of Mobile Hosts

The last set of experiments investigates the effect of the number of mobile hosts. All hosts move randomly at the maximum speed of 4 m/s. The pause time between two movements is 10 seconds. The number of mobile hosts increases from 20 to 70 by 10s. The number of connections is equal to the number of hosts. Because the simulation area is fixed to 1000m x 1000m, the experiments also demonstrate the impact on performance of the density of hosts.

Performance: The offered traffic load increases linearly with the number of hosts (as shown in figure 7a), which is equal to the number of connections (the DSDV line is covered by the AODV line). Figure 7b shows that the throughput for DSDV is rather stable regardless of the number of hosts, which implies that delivery ratio for DSDV decreases nearly linearly with the number of hosts. The throughput for AODV drops with the number of hosts, about 35% from 20 hosts to 70 hosts. Compared with DSDV, AODV has a better performance for a sparser network (fewer than 40 hosts), and worse performance for a denser one (more than 40 hosts).

As in other experiments, as shown in figure 7d, AODV outperforms DSDV in terms of an average delay, because of the time limit on buffered packets.

DSDV and AODV have similar protocol loads for 20 mobile hosts. Both of them introduce more overhead as the number of hosts increases, with the load for AODV growing faster than for DSDV (figure 7e).

Both DSDV and AODV have similar normalized power consumption for a sparse network (figure 7f). For DSDV, the increase of power consumption is nearly linear with the host number. The power consumption for AODV increases faster than for DSDV. For 70 hosts, AODV con-
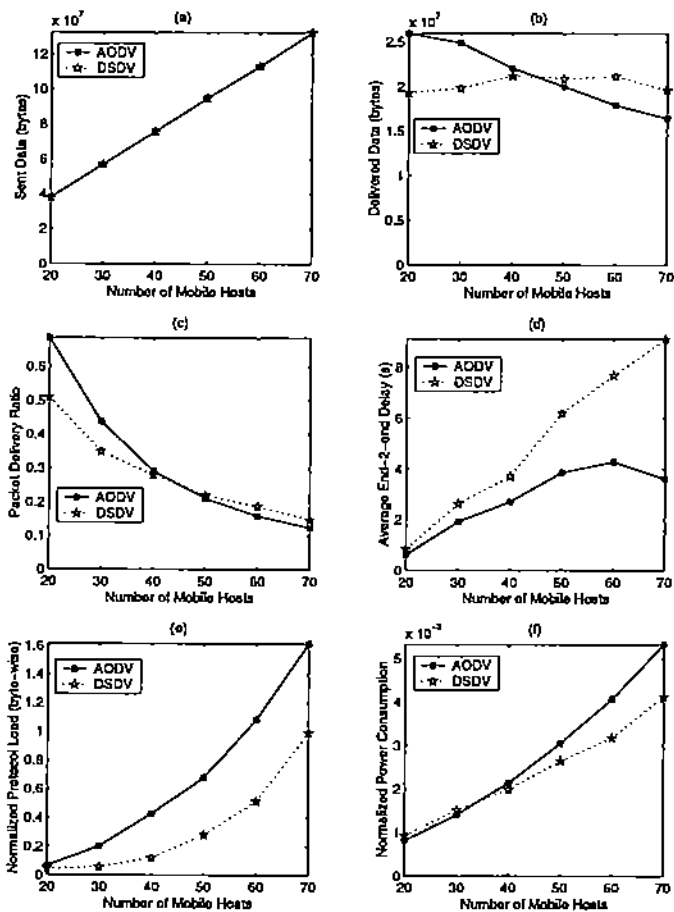
Fig. 7. Performance vs. Number of Mobile Hosts

Fig. 8. Dropped Packets vs. Number of Mobile Hosts
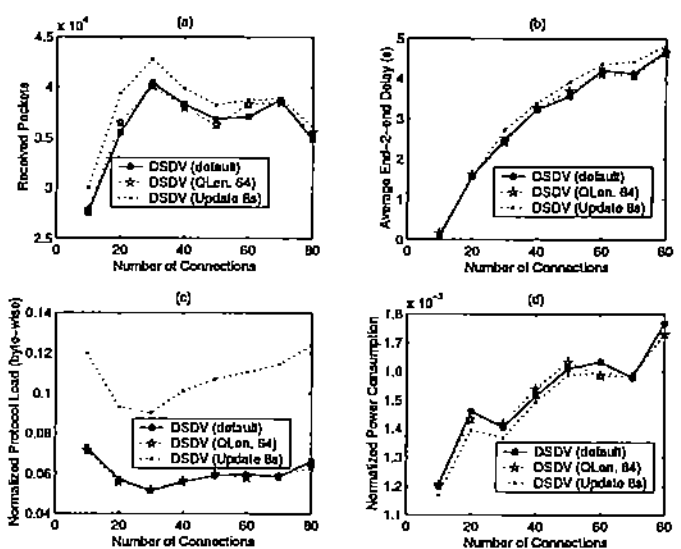
Fig. 9. Performance Comparison of Different DSDV Implementations

sumes 33% more energy than DSDV per 1k-byte data delivered.

From the results provided in figure 7, we can tell that DSDV is more scalable with respect to the number of hosts. It seems that 40 hosts per square kilometer is the turning point. For more than 40 hosts, DSDV equals or outperforms AODV for all metrics (the average delay is an exception that should not be considered).

Dropped Packets: Figure 8 shows how many packets have been dropped for each of the four reasons. Congestion is the dominant reason. DSDV still drops more packets due to congestion ("other reason"), but the gap between DSDV and AODV does not increase with the number of hosts. Because the number of links in the network is on the order of $O(n^2)$, where n is the number of hosts, the probability that shortest routes are converging to a few hosts is very low when n is big enough.

Figure 8c illustrates that fewer packets are dropped by DSDV due to a full queue as the number of hosts increases.
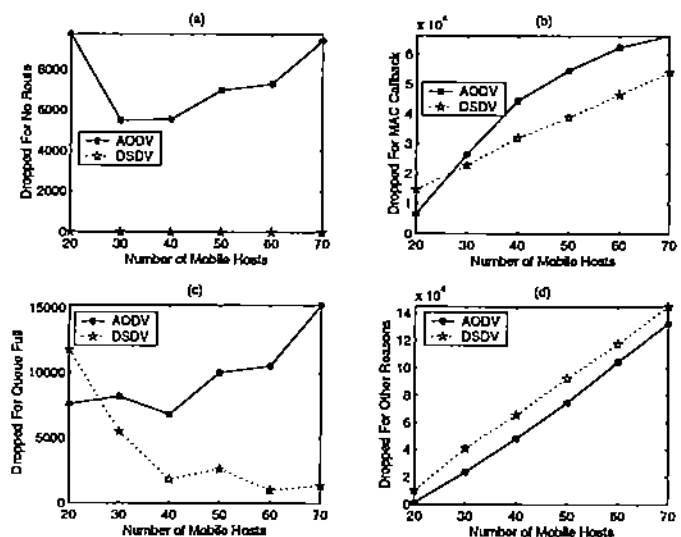
VII. FURTHER DISCUSSION ABOUT DSDV

A. Reduce Broadcast Interval of DSDV

The time interval between broadcasting routing information is one of the most important parameters of DSDV [1]. As shown in figure 5, in total about $5.5 * 10^4$ packets are dropped for 80 connections due to either a MAC callback or a full queue, which means that the outgoing links are broken or the routes are not establish timely. Some of these situations could be avoided by broadcasting routing information more frequently, at the cost of a higher protocol overhead. The question are: How much improvement of performance can be obtained? How much will it cost?

We reduced the broadcast time interval from 15 seconds to 8 seconds, and rerun the set of experiments described

in section VI-C, using the same settings, parameters, scenarios, and connections.

Figure 9a (the "Update 8s" curve) shows that the throughput increases about 10% for less stressful cases (i.e., for fewer than 70 connection). The average delay is almost the same (figure 9b). The normalized protocol load doubles as we expect (figure 9c). The power consumption slightly decreases, because packets are dropped earlier as we explain in section VI-B.

B. Increase the Queue Length of DSDV

Figure 5c shows that about $1.5*10^4$ packets are dropped due to a full queue. Since the queue length for DSDV is only 5, much smaller than that for AODV, it is natural to ask this question: Will a longer queue increase the throughput of DSDV?

We set the queue length to 64 and rerun the set of experiments again. The results are shown in figure 9 (the "QLen. 64" curve). The performance metrics are almost the same as those measured for the original DSDV implementation. Thus, the longer queue does not help in improving performance of DSDV.

VIII. CONCLUSIONS

Conclusion 1: For the movements of mobile hosts generated by the random waypoint model, as shown in figures 1 and 2, the link changes and route changes are, with a very high probability, linear functions of the maximum speed (with the pause time to be 10 seconds), and linear functions of the pause time (with the maximum speed to be 4 m/s), respectively. The maximum speed does not affect much the performance of DSDV and AODV at the range of 4 m/s to 24 m/s (figure 3).

Conclusion 2: In less stressful situations, AODV outperforms DSDV for all metrics except for normalized protocol load. AODV reaches its performance peak in the series of simulations with 10 connections in a 30-host network, as illustrated in figure 4. DSDV outperforms AODV in denser networks with a higher traffic load. It is more scalable than AODV with respect to the density of the mobile hosts and the number of connections.

Conclusion 3: Congestion at MAC layer is the dominant reason for more than half of packets being dropped for both protocols (figure 5). Sending packets always via the shortest routes might cause congestion in sparse networks. This phenomenon is less visible in dense networks where the shortest routes are not likely to share a few heavily burdened hosts.

Conclusion 4: For DSDV, reducing the time interval between broadcasts improves the throughput by about 10% for low traffic load, at the cost of doubled protocol load. Longer queue does not help DSDV in achieving higher throughput, as illustrated in figure 9.

In general, we can state: (1) The protocol load for the proactive routing protocols (such as DSDV) grows as the number of hosts increases, while that of the on-demand routing protocols (such as AODV) increases with the number of Source-Destination (S-D) pairs. The proactive approach performs better when the number of S-D pairs is close to the number of hosts. (2) The on-demand approach consumes less power, because it propagates the link break information faster, thus it avoids sending packets that are dropped eventually. (3) Network congestion is the dominant reason for dropping packets for both proactive and on-demand approaches. It is affected by route selection.

ACKNOWLEDGMENTS

REFERENCES

[1] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers," in *Proceedings of the SIGCOMM 94 Conference on Communications Architectures, Protocols and Applications*, August 1994, pp. 234–244.

[2] J. J. Garcia-Luna-Aceves and M. Spohn, "Source-tree routing in wireless networks," in *Proceedings of 7th International Conference on Network Protocols*, 1999.

[3] C. E. Perkins, E. M. Belding-Royer, and S. R. Das, "Ad hoc on-demand distance vector (AODV) routing," *IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manet-aodv-11.txt*, June 2002.

[4] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, pp. 153–181, 1996.

[5] Z. J. Haas, M. R. Pearlman, and P. Samar, "The zone routing protocol (ZRP) for ad hoc networks," *IETF Internet Draft. http://www.ietf.org/internet-drafts/draft-ietf-manet-zone-zrp-02.txt*, July 2002.

[6] R. Sivakumar, P. Sinha, and V. Bharghavan, "CEDAR: A core-extraction distributed ad hoc routing algorithm," *IEEE Journal on Selected Areas in Communications*, vol. 17, pp. 1454–1465, August 1999.

[7] S. Corson and J. Macker, "Mobile ad hoc networking (MANET): Routing protocol performance issues and evaluation considerations," *RFC 2002*, January 1999.

[8] S. R. Das, R. Castaeda, and J. Yan, "Simulation-based performance evaluation of routing protocols for mobile ad hoc networks," *Mobile Networks and Applications*, vol. 5, pp. 179–189, 2000.

[9] C. E. Perkins, E. M. Royer, S. R. Das, and M. K. Marine, "Performance comparison of two on-demand routing protocols for ad hoc networks," *IEEE Personal Communications*, vol. 8, no. 1, pp. 16–28, Febuary 2001.

[10] "The network simulator - ns-2," *http://www.isi.edu/nsnam/ns/*.

[11] H. Jiang and J. J. Garcia-Luna-Aceves, "Performance comparison of three routing protocols for ad hoc networks," in *Proceedings of IEEE ICCCN 2001*, 2001, pp. 547–554.

[12] X. Zeng, R. Bagrodia, and M. Geria, "GloMoSim: A library for parallel simulation of large-scale wireless networks," in *Proceedings of the 12th Workshop on Parallel and Distributed Simulations - PADS'98*, May 1998.

[13] T. Camp, J. Boleng, B. Williams, L. Wilcox, and W. Navidi, "Performance comparison of two location based routing protocols for ad hoc netowkrs," in *Proceedings of IEEE Infocom 2002*, New York, June 2002.

[14] R. V. Boppana and S. P Konduru, "An adaptive distance vector routing algorithm for mobile, ad hoc networks," in *Proceedings of IEEE Infocom 2001*, Anchorage, Alaska, April 2001, IEEE, vol. 3, pp. 1753–1762.

[15] Lucent Technologies, "WaveLAN/PCMCIA card user's guide," October 1996.

[16] J. Broch, D. A. Maltz, D. B. Johnson, Y. Hu, and J. Jetcheva, "A performance comparison of multi-hop wireless ad hoc network routing protocols," in *Proceedings of the fourth annual ACM/IEEE international conference on Mobile computing and networking*. 1998, pp. 85–97, ACM Press.

[17] R. Kravets and P. Krishnan, "Application-driven power management for mobile communication," *Wireless Networks*, vol. 6, no. 4, pp. 263–277, 2000.

[18] S. M. Ross, *Probability Models for Computer Science*, Academic Press, 2002.

[19] N. R. Draper and H. Smith, *Applied Regression Analysis (Third Edition)*, John Wiley & Sons, Inc., 1998.