Analysis of the Dynamic Source Routing Protocol for Ad Hoc Networks

Rajendra V. Boppana Anket Mathur
CS Department, UT San Antonio, San Antonio, TX 78249. USA
boppana@cs.utsa.edu amathur@cs.utsa.edu

Abstract

The Dynamic Source Routing (DSR) protocol is a simple and robust routing protocol designed for use in multi-hop wireless ad-hoc networks of mobile nodes. Several of the optimizations proposed in the protocol tend to hurt the performance, especially in the case of high node mobility and low traffic load. This issue has been studied extensively, and DSR is shown to perform better with certain optimizations turned off. In this paper, we show that DSR's performance is unsatisfactory even with these modifications. We suggest three simple and intuitive changes to the routing protocol. Using simulations, we show that the new techniques provide significant performance improvements for various network densities and traffic loads. To illustrate the relative significance of the proposed changes, traffic load and network density on the overall performance, we present 2^k factorial analyses of the simulation data. Based on the statistical analysis, we show that limiting replies by destination is the most beneficial change to the routing protocol and that network density has significant impact on performance in uncongested networks.

1. Introduction

A mobile ad-hoc network (MANET) is a multi-hop wireless network formed by a group of mobile nodes that have wireless capabilities and are in proximity of each other. MANETs facilitate communication among mobile users in situations—military or civil emergency––where fixed infrastructure is infeasible. Most MANETs are based on IEEE 802.11 or WiFi medium access control (MAC) standard [14]. Owing to external noise and interference from competing transmissions and node mobility, the routes in a MANET break frequently. The *Dynamic Source Routing* (DSR) [1,4] is one of the widely used routing protocols for MANETs. Several previous studies indicate that some of the route gathering techniques and optimizations proposed in the original protocol actually hurt the performance in many situations and make DSR underperform another commonly used routing protocol—ad hoc on demand distance vector (AODV) [12]. Because of source routing, however, DSR is considered to be desirable from security aspect [3,4]. Several previous studies indicate the benefit of turning off some of the "optimization" features of DSR to improve its performance [5,8,9].

In this paper, we show that even with these modifications, DSR's performance is unsatisfactory (nearly 40% of the injected packets are dropped), especially at low traffic loads. (Unlike several previous studies, we investigate the protocol performance at *low* traffic loads.) We propose three simple and intuitive changes to the routing protocol: (a) limiting the replies sent by destinations in response to route requests from sources, (b) sorting the routes based on freshness rather than hop count, and (c) limiting the number of routes kept per destination to one. Using simulations, we show that these features improve DSR's performance. To illustrate the relative significance of various factors—the three proposed changes to routing protocol, traffic load and network density, we present 2^k factorial analyses of the simulation data. Based on the statistical analysis, we show that limiting replies by destination is the most beneficial change to the routing protocol and that network density has significant impact on performance in uncongested networks.

2. Dynamic Source Routing (DSR)

Basic Operation

Each node in the network maintains a *route cache* in which it caches the routes it has learned. To send data to another node, if a route is found in its route cache, the sender puts this route (a list of all intermediate nodes) in the packet header and transmits it to the next hop in the path. Each intermediate node examines the header and retransmits it to the node indicated after its id in the packet route. If no route is found, the sender buffers the packet and obtains a route using the route discovery process described below.

Route Discovery and Maintenance

To find a route to its destination, a source broadcasts a *route request* packet to all nodes within its radio transmission range. In addition to the addresses of the source and the destination nodes, a route request packet contains a *route record*, which is an accumulated record of nodes visited by the route request packet. When a node receives a route request, it does the following.

- If the destination address of the request matches its own address, then it is the *destination*. The route record in the packet contains the route by which the request reached this node from the source. This route is sent back to the source in a *route reply* packet by following the same route in reverse order. (We assume bidirectional links. The alternative reply mechanism for unidirectional links is not considered here.)
- Otherwise, it is an *intermediate* node. If the node has not seen this request before and has a route to the destination in its cache table, it creates a route reply packet with the route from its cache, and sends it back to the source. Such replies are called Intermediate-Node replies; if it does not have a route, it appends its own address to the route record, increments hop count by one, and rebroadcasts the request.

When the source receives a route reply, it adds this route to its cache and sends any pending data packets. If any link on a source route is broken (detected by the MAC layer of the transmitting node), a *route error* packet is generated. The route error is unicasted back to the source using the part of the route traversed so far, erasing all entries that contain the broken link in the route caches along the way.

Optimizations

By virtue of source routing, nodes have access to a large amount of routing information. For instance, the route indicated in a route request/reply or data packet can be used to learn routes to every other node on the route. DSR makes use of route caching aggressively. For example, a destination replies to every route request that it receives, and the source keeps the excess replies as alternate routes to the destination. Several optimizations to this basic protocol have been proposed and have been evaluated to be very effective by the authors of the protocol [1]. Some of them are:

- Data Salvaging: If an intermediate node encounters a broken link and has an alternate route to the destination in its cache, it can try to salvage the packet by sending it via the route from its cache.
- Gratuitous Replies: When a node overhears a packet addressed to another node, it checks to see if the packet could be routed via itself to gain a shorter route. If so, the node sends a *gratuitous reply* to the source of the route with this new, better route.
- Route Snooping: A node that overhears a data packet and does not have the packet route in its own cache, adds the new route to its cache for future use.

Security and Performance Issues

Certain features of DSR hurt its performance or make it vulnerable to security attacks [3-9].

No Expiration of Routes: Without an effective mechanism to remove excessively old (*stale*) entries, route caches may contain broken or non-minimum hop routes. Using stale routes causes loss of data packets (low delivery rate) and wastes network bandwidth. Route replies from intermediate nodes and snooping data packets exacerbate this problem by polluting caches with stale routes [5-7].

Intermediate-Node (IN) Replies: Intermediate-node replies make the route learning process faster because all route requests do not need to travel all the way to the destination. Without route freshness indication, however, it results in polluting caches with stale routes when node mobility is high and data transmissions are infrequent [5,6].

When a source receives the bad route reply, it tries to send the waiting data packet along the route. Upon failure of one of the links along the route, a route error packet is propagated back to the source, which then issues a new route request, starting the process all over again.

Data Salvaging: Data Salvage can be useful in relatively static networks, in which routes remain stable for relatively long periods of time. However, in a MANET, it is likely that the route in the intermediate node's cache was older, and hence, also invalid. Trying to salvage a data packet by using another bad route would result in a waste of time and bandwidth. Also, a malicious node may misroute data packets without risking its detection under the guise of data salvaging.

Gratuitous Replies: Like data salvaging, gratuitous replies can be of limited benefit when the routes are fresh and nodes are not malicious. Otherwise, this feature degrades performance, security, or both.

Evaluation of DSR

We analyzed the performance of the original DSR and the impact of turning off some the optimizations discussed above. To turn off intermediate node replies, we modified the DSR code so that when an intermediate node hears a new route request, it simply rebroadcasts it, even if it has a route to the destination. To turn of data salvage, we modified the code so that a data packet that cannot be transmitted to the next hop specified in the source is dropped and a route error message is sent to the source. Gratuitous replies are turned off by not sending route shortening messages to packet sources. We also modified the route replies and request packets to carry timestamps so that we can keep track of route creation time and ages of routes used. We give a quantitative measure of the staleness of routes that has been so widely reported but not measured in literature.

Simulation environment: All simulations were run on the Glomosim network simulator [10]. The modifications were made to the implementation of DSR written for Glomosim. A 100 node network in a field size of 1000m x 1000m was used. The mobility model used was random waypoint [11] in a square/rectangular field. In random waypoint, each node starts its journey from its current location to a random location within the field. The speed is randomly chosen to be between 1-19 m/sec. Once the destination is reached, another random destination is targeted after a specified pause. We used 0-second pause time, which results in continuous node mobility in our simulations.

Twenty-five CBR (constant bit-rate) over UDP connections (distinct sources and destinations) were used to generate traffic by injecting 512-byte packets with average interpacket time varied according to the load rate desired. For each configuration, the network is simulated for 600 seconds.

We used delivery rate, the percentage of injected packets that are delivered to destinations, and average age of routes used to analyze the performance. At low loads, the delivery rate gives a measure of route correctness rather than load balancing or other issues of the protocol. We indicate route ages since it is frequently mentioned in literature without quantitative evaluation. In addition to the original DSR, we simulated three variations: intermediate nodes replies off (denoted as 'IN off' in the graphs below), data salvaging off (DS off), or both off. The gratuitous replies option was turned off in all modified versions of DSR since it seems to provide little performance benefit and is a significant risk for blackhole and other security attacks.

Figure 1 gives the delivery rates and Figure 2 the ages of routes used. The route age is extremely high for the original DSR. Turning off IN replies, improves the route age and throughput. Turning off data salvage improves reduces route age significantly indicating that major contribution to the average route age is from stale routes used by intermediate routes trying to overcome stale source routes in data packets. Data salvage alone does not impact performance. In conjunction with IN replies off, however, data salvage provides marginal performance benefit. Given that malicious node detection becomes harder with data salvage, turning it off is preferable.

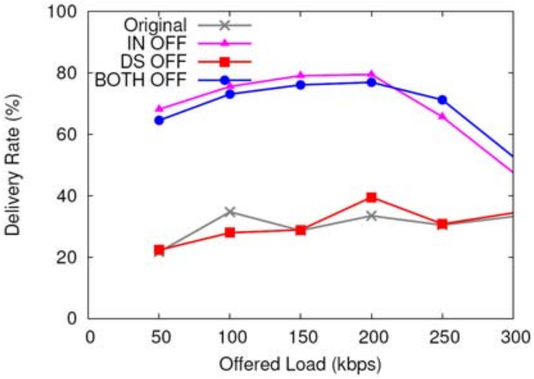

Figure 1. Performance of DSR with some of the features turned off.

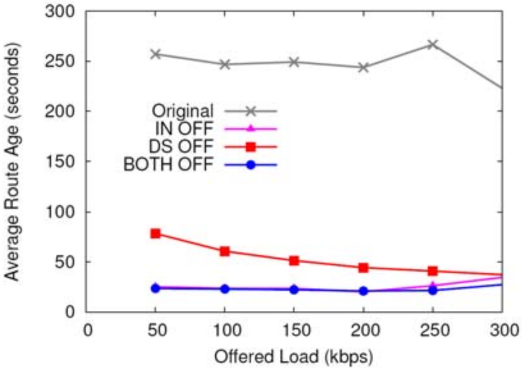

Figure 2. Average age of routes used.

We use the original DSR with gratuitous replies, IN replies and data salvage turned off (the last two are commonly recommended in literature to improve performance and/or security) as the 'Base DSR' for the remainder of the paper.

3. New Techniques to Improve DSR Performance

The throughput achieved by Base DSR at a load of 50 Kbps for the example network configuration is about 32 Kbps (64% delivery rate), whereas, the throughput achieved by AODV for this network is close to 40 Kbps (80% delivery rate). So, to improve the performance of

DSR further, we evaluate three simple, intuitive routing modifications based on our observations of other protocols.

Limiting Replies from Destination

In the original implementation of DSR, a destination node replies to every route request packet it receives. This, however, results in a lot of unnecessary route replies when the same route request is heard by a destination multiple times. This can also result in ‘bad’ routes being added to the route cache of the source. For instance, consider 2 route requests that take the same number of hops, but different paths to reach the destination at different times. The request that reaches the destination late possibly took a path that was more congested. Instead of being discarded, this request is also replied to, and because it had the same hop count as the previous request, it is added to the top of the route cache of the source. Hence, when a data packet is to be sent, a congested route is tried before the route that was not congested.

We modified DSR such that destination nodes will reply to a route request only if (a) the last route request from that source was older than the current one or (b) the last route request has the same timestamp (the same route request took different routes to the destination) but the current request took fewer hops. This ensures that replies are sent only for fresh request packets and multiple replies are sent only if they improve route hop count. This feature can be easily implemented using request and reply timestamps in route request and reply packets.

Giving Preference to Fresher Routes

The original DSR keeps multiple routes to a destination ordered by hop count. This ensures that routes used are minimum hop count routes, but also ensures that a stale 1-hop route overrides a fresher 2-hop route to the same destination.

We modified the route cache such that it maintains routes to a particular destination in the following order:

(a) A route with a later request time is given preference over a route with an earlier request time;

(b) If the request times of two routes are the same, then a route with shorter hop count is given preference over a longer route;

(c) If both the request time and the hop count of two routes are the same, then a route with a later reply time is given preference over a route with an earlier reply time.

Keeping Only One Route per Destination

If routes are ordered by freshness, and the first route fails, it is very likely that the older routes stored in the cache will also fail. By trying all the routes in the cache before sending a new route request, a lot of time and bandwidth is wasted. In this technique, only one route determined to be the main route by freshness or hop count is kept in the cache. The current trend is to keep multiple routes and switch to a new one as soon as one fails. Keeping multiple routes improves throughput and reduces overhead when the network is congested and alternate routes are fresh. Since our interest in uncongested networks, and AODV, which keeps only one route per destination, performs well at low traffic, it will be interesting to see if DSR can benefit from this feature.

Performance Analysis of Proposed techniques

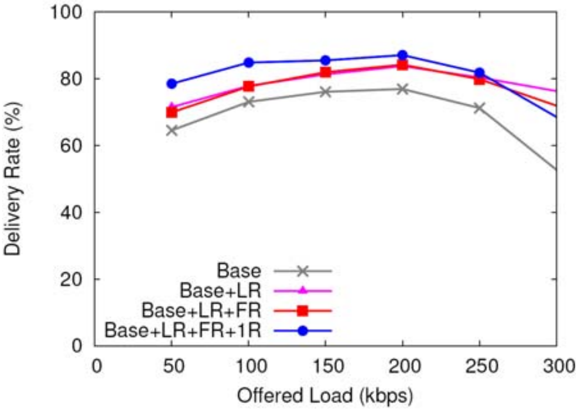

Figure 3. Delivery rates of base DSR and proposed modifications.

Figure 3 gives delivery rates of ‘Base DSR’ (original DSR with IN replies, DS and gratuitous replies turned off) and combinations of the three proposed techniques applied to the base DSR. (LR indicates limited replies, FR routes sorted by timestamps, and 1R one route per destination.)

Applying all three proposed techniques, denoted ‘Base + LR + FR + 1R’ in the graph, achieves the best performance until the network starts to saturate for high loads (>250 Kbps). At these high loads, most routes are congested. In such a scenario, congested links could be wrongly identified as ‘broken’, resulting in route errors and route requests propagating throughout the network. Keeping only 1 route increases the routing overhead (Figure 4), and hurts the performance for high loads. At lower loads, there is enough network bandwidth to absorb the additional control traffic caused by 1R option. It is noteworthy that the average ages of routes used (Figure 5) with the proposed techniques are also reduced significantly compared to the base DSR. Compared to the original DSR, the combination of base DSR with LR, FR and 1R options reduces route age by a factor of 15 and improves delivery rates by a factor of 2 to 4.

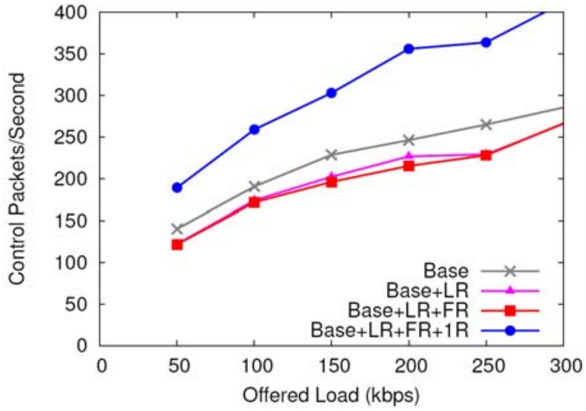

Figure 4. Control overhead for various forms of DSR.

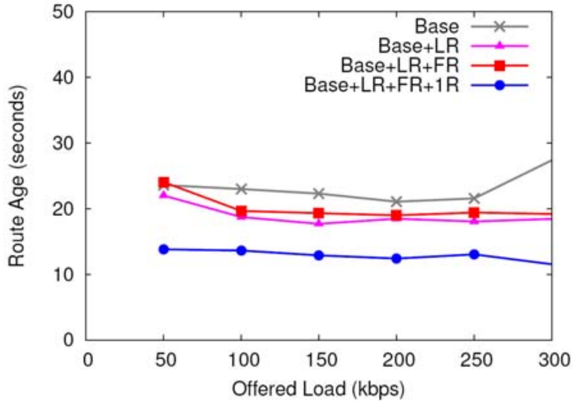

Figure 5. Route ages for various forms of DSR.

4. Factorial Analysis

While graphs such as those presented in the previous section give a general overview of the performance changes caused by the different techniques, a more rigorous analysis is needed to determine the relative impacts of the techniques and interactions of 2 or more of them on performance. We used 2^k factorial design [13], which facilitates this type of statistical analysis for k factors. Each factor used in the analysis is varied between two values (denoted as levels -1 and +1). A total of 2^k simulations need to be conducted varying the factors of interest systematically, keeping all other input parameters fixed, for the analysis. The factors considered and their values at each level are given below.

Factor	Level -1	Level +1
A. Limited replies	Off	On
B. Cache route order	Hop count	Timestamp
C. # Routes in cache	Many	One

Table 1 shows the factorial analysis of the proposed routing protocol modifications with load kept constant at 100 Kbps (low traffic). Delivery rate (column y in the table) is used as the performance metric.

There are eight rows corresponding to each combination of factor levels; the achieved delivery rate for each case is indicated. Each column labeled with a combination of A, B, and C indicates a factor combination. The effect of each factor combination on the performance is calculated as

Dot Product{Column for the factor, Column y} / 8.

For factor A, it is $(-1)(73.2) + (1)(78) + (-1)(71) + (1)(77.8) + (-1)(82.7) + (1)(84) + (-1)(82.8) + (1)(85) = 1.89$.

The value calculated for column 'I' is the average delivery rate, denoted y_{av}. For the data in Table 1, it is 79.3.

The total variation of y or sum of squares total (SST) is calculated as

$$\text{Total variation of } y = SST = \Sigma(y_i - y_{av})^2.$$

The variations of various combinations of factors are calculated. For example,

Sum of squares due to A (SSA)

$$= (\text{Effect of } A)^2 \times 2^3 = 28.5$$

Now fraction of total variation due to each factor (or combination of factors) can be calculated as the ratio of sum of squares value for that combination and SST. For example,

Fraction of variance due to A = SSA/SST

A quick glance at the table indicates that the number routes kept per destination is the most significant factor at this traffic load. A similar analysis for traffic load of 200 Kbps is given in Table 2. (To save space, we present the data differently: the sign matrix is not indicated; the delivery rates for various combinations of factor levels are indicated in a row; the calculations for effects and allocation of variations due to factors are indicated as before.) Now, 'limited replies' is the most significant factor. For both traffic loads, the order in which routes are maintained is not significant though preferring fresh routes provides marginal improvement in throughput. Between limited replies and one route per destination techniques, the former is more significant as shown in Table 3, which uses these two techniques and traffic load for 3-way factorial analysis. These tables also indicate the impact of combinations of these factors. For example, Table 3 shows that the combination of limited replies and load has slightly positive impact on performance (factor effect of 2.25 and variation due to the combination of 0.13), but the combination of keeping single route and load has slightly adverse impact at high loads (factor effect of -2.49 and variation of 0.16). Using all three modifications gives the best overall performance for uncongested networks. For loads that lead to saturation, traffic load will be the most dominant factor.

Though there are many studies on DSR, it is hard to compare results in one paper to those in another owing to different field sizes and other parameters used. One factor we are very much interested in is the network density—the number of nodes within a radio transmission area. It is calculated as follows.

$$(N \cdot \pi r^2)/(lw) \text{ nodes/radio range,}$$

where N is number of nodes, r radio transmission range, l length of the field and w breadth of the field. For the example network, N=100, r=250 m, l=1000 m, w=1000 m, and the network density is 19.6 nodes/radio range. Two commonly used field sizes in literature are 1500x300 m^2 and 2200x600 m^2. With everything else kept the same, the corresponding network densities are 43.6 and 14.8, respectively. We simulated these networks and analyzed the significance of the three routing techniques. The results are the same. Number routes per destination is the most significant factor at 100 Kbps and limiting route replies by destination is the most significant factor at 200 Kbps. The order of routes used has negligible impact on performance.

To see the impact of network density and traffic load, we conducted a 2^3 factorial analysis with the following factors.

Factor	Level -1	Level +1
A. Routing protocol	Base DSR	Base DSR+ LR+ FR+ 1R
B. Traffic load	100 Kbps	200 Kbps
C. Network density	Low (2200x600)	High (1500x300)

The analysis is given in Table 4. Base DSR enhanced with the three proposed techniques has the most impact on overall performance; more than 20% of the variation in performance is due to network density. This indicates that some of the discrepancies in performance claims by different researchers are due to the densities of networks used.

5. Related Work

Routing features that hurt DSR's performance has been extensively studied [5-9,12,15]. The most commonly recommended remedy is turning off intermediate node replies. In addition, several complicated caching strategies and different types of caches have been investigated to improve performance [5-7,15]. Expiration of unused routes and broadcasting route errors and route replies for wider dissemination of routing information have been investigated and found to provide some benefit [5,15]. One study [15] investigated sending only one reply by destination (a restrictive form of our proposed limited replies, which allows more than one reply by destination). It is noteworthy that in all of these studies the throughputs or delivery rates are low, often about 64-72%, at traffic loads 50-100 Kbps (Figures 6-8 in [15]). We achieve better performance without complicated caching strategies and simpler routing modifications. We have not come across any factorial analysis of the routing features or other important factors such as network density.

6. Conclusions

DSR is a widely used routing protocol for mobile ad hoc networks, but has very low delivery rates and poor performance in lightly loaded networks with high node mobility. Several of the modifications proposed in the literature such as turning off intermediate node replies improves the performance somewhat.

This paper presents three simple (and used in other routing protocols) techniques—limiting replies sent by destination, keeping only one route per destination, and preferring fresher routes over shorter ones—to further improve the performance of DSR. Factorial analysis indicates that both limited replies and one route per destination improve performance significantly and the third feature does not impact performance. While multiple routes may benefit at higher traffic loads, keeping only one route per destination helps sender nodes gather routes when the topology changes. Without using any complicated strategies, our proposed techniques perform significantly better than previously proposed modifications at very low traffic loads (50-100 Kbps) and about the same at higher traffic loads.

Additional factorial analysis indicates that, besides routing protocol features, network density impacts the overall performance measurably.

In future we intend to expand the statistical analysis to evaluate the significance of mobility and traffic patterns.

Acknowledgments

This research has been partially supported by NSF grant EIA-0117255 and AIA grant F30602-02-1-0001.

Table 1 Factorial analysis of three routing techniques at 100 Kbps load for medium network density.

Field Size: 1000x1000 sq. m (Medium Density); Load: 100 Kbps									
		Limited Replies	Route Order	# Routes in Cache					Del. Rate
	I	A	B	C	AB	AC	BC	ABC	y
	1	-1	-1	-1	1	1	1	-1	73.20
	1	1	-1	-1	-1	-1	1	1	78.00
	1	-1	1	-1	-1	1	-1	1	71.00
	1	1	1	-1	1	-1	-1	-1	77.80
	1	-1	-1	1	1	-1	-1	1	82.70
	1	1	-1	1	-1	1	-1	-1	84.00
	1	-1	1	1	-1	-1	1	-1	82.80
	1	1	1	1	1	1	1	1	85.00
Factor effect	79.31	1.89	-0.16	4.31	0.36	-1.01	0.44	-0.14	
Sum of squares	1.88E+02	2.85E+01	2.11E-01	1.49E+02	1.05E+00	8.20E+00	1.53E+00	1.51E-01	
Variation due to factor	1.00	0.15	0.00	0.79	0.01	0.04	0.01	0.00	

Table 2 Factorial analysis of three routing techniques at 200 Kbps load.

Field Size: 1000x1000 sq. m (Medium Density); Load: 200 Kbps A. limited replies, B: route order, C: number of routes per dest.								
Factor levels	-1, -1, -1	1, -1, -1	-1, 1, -1	1, 1, -1	-1, -1, 1	1, -1, 1	-1, 1, 1	1, 1, 1
Delivery rate (y)	76.90	83.75	74.20	84.20	70.30	86.90	70.45	87.05
	I	A	B	C	AB	AC	BC	ABC
Factor effect	79.22	6.26	-0.24	-0.54	0.39	2.04	0.32	-0.39
Sum of squares	3.53E+02	3.13E+02	4.75E-01	2.37E+00	1.24E+00	3.34E+01	8.13E-01	1.24E+00
Variation due to factor	1.00	0.89	0.00	0.01	0.00	0.09	0.00	0.00

Table 3 Factorial analysis of limited replies and number of routes per destination.

Field Size: 1000 x 1000 sq. m (Medium Density). Route Order: By freshness Factors: A, limited replies: off (-1) or on (+1) B, number of routes in cache: many (-1) or one (+1) C, traffic load: 100 Kbps (-1) or 200 Kbps (+1)								
Factor levels	-1, -1, -1	1, -1, -1	-1, 1, -1	1, 1, -1	-1, -1, 1	1, -1, 1	-1, 1, 1	1, 1, 1
Delivery rate (y)	71.00	77.80	82.80	85.00	74.20	84.20	70.45	87.05
	I	A	B	C	AB	AC	BC	ABC
Factor effect	79.06	4.45	2.26	-0.09	0.25	2.20	-2.49	1.40
Sum of squares	3.04E+02	1.58E+02	4.10E+01	6.13E-02	5.00E-01	3.87E+01	4.95E+01	1.57E+01
Variation due to factor	1.00	0.52	0.13	0.00	0.00	0.13	0.16	0.05

Table 4 Factorial analysis of routing techniques, traffic load and network density.

Factors: A: base DSR (-1), enhanced DSR (+1), B: 100 Kbps (-1), 200 Kbps (+1), C: low density (-1), high density									
		DSR	Load	Density					Del. Rate
	I	A	B	C	AB	AC	BC	ABC	y
	1	-1	-1	-1	1	1	1	-1	64.50
	1	1	-1	-1	-1	-1	1	1	71.60
	1	-1	1	-1	-1	1	-1	1	60.55
	1	1	1	-1	1	-1	-1	-1	73.00
	1	-1	-1	1	1	-1	-1	1	74.70
	1	1	-1	1	-1	1	-1	-1	86.90
	1	-1	1	1	-1	-1	1	-1	58.95
	1	1	1	1	1	1	1	1	87.55
Factor effect	72.22	7.54	-2.21	4.81	2.72	2.66	-1.57	1.38	
Sum of squares	8.30E+02	4.55E+02	3.89E+01	1.85E+02	5.91E+01	5.64E+01	1.97E+01	1.53E+01	
Variation due to factor	1.00	0.55	0.05	0.22	0.07	0.07	0.02	0.02	

References

[1] D. Johnson, D. Maltz and Y. Hu. The dynamic source routing protocol for mobile ad hoc networks. IETF MANET Working Group, Internet Draft 2003.

[2] T. Jiang, Q. Li, and Y. Ruan. Secure Dynamic Source Routing Protocol. In Proceedings of the Forth International Conference on Computer and Information Technology (CIT), 2004.

[3] R. N. Mir and A. M. Wani. Security Analysis of Two On-Demand Routing Protocols in Ad Hoc Networks. In Proceedings of ACM MOBIHOC 2001.

[4] D. Johnson and D. Maltz. Dynamic Source Routing in Ad Hoc Wireless Networks. In Mobile Computing, edited by Tomasz Emilienski and Hank Korth, Kluwer Academic Publishers, 1996.

[5] M. K. Marina and S. R. Das. Performance of Route Caching Strategies in Dynamic Source Routing. In Proceedings of Int'l Workshop on Wireless Networks and Mobile Computing (WNMC), 2001.

[6] Y. Hu and D. Johnson. Caching Strategies in On-Demand Routing Protocols for Wireless Ad Hoc Networks. In Proceedings of the Sixth Annual International Conference on Mobile Computing and Networking (Mobicom), 2000.

[7] W. Lou and Y. Fang. Predictive Caching Strategy for On-Demand Routing Protocols in Wireless Ad Hoc Networks, Wireless Networks, vol. 8, no. 6, 2002.

[8] T. Goff et al., Preemptive Routing in Ad Hoc Networks. In Proceedings of the 7th Annual Int'l Conference on Mobile Computing and Networking (Mobicom), 2001.

[9] D. De Couto, D. Aguayo, J. Bicket, and R. Morris. A High-Throughput Path Metric for Multi-Hop Wireless Routing. In Proceedings of MobiCom 2003.

[10] X. Zeng, R. Bagrodia, and M. Gerla, "Glomosim: A library for parallel simulation of large-scale wireless networks," in *Workshop on Parallel and Distributed Simulation*, 1998, pp. 154–161.

[11] T. Camp, J. Boleng, V. Davies. A survey of mobility models for Ad Hoc Network Research. In Wireless Communication and Mobile Computing (WCMC): Special Issue on Mobile Ad Hoc Networking: Research, Trends and Applications. vol. 2, no. 5, 2002.

[12] C. E. Perkins, E. M. Royer, S. R. Das, M. K. Marina, Performance Comparison of Two On-demand Routing Protocols for Ad Hoc Networks, IEEE Personal Communications 8 (1) (2001) 16–28.

[13] Raj Jain. The Art of Computer Systems Performance Analysis: Techniques for Experimental Design, Measurement, Simulation, and Modeling. John Wiley & Sons, 1991.

[14] IEEE Computer Society LAN/MAN Standards Committee, "Part 11: Wireless LAN, medium access control (MAC) and physical layer (PHY) specifications," IEEE, Inc., Standard ANSI/IEEE 802.11, 1999.

[15] M.K. Marina and S. R. Das, Impact of Caching and MAC Overheads on Routing Performance in Ad Hoc Networks, Computer Communications, 2003.