

A Lightweight Threshold-Based Improvement on DSDV

Rahem Abri and Sevil Sen

Department of Computer Engineering,
Hacettepe University, Ankara, Turkey
{rahemabri, sevilсен}@hacettepe.edu.tr

Abstract. One of the main research challenges in mobile ad hoc networks (MANETs) is designing adaptive, scalable and low-cost routing protocols for these highly dynamic environments. In this paper, we propose a new improvement on DSDV, which is one of the most popular proactive routing protocols in MANETs. We introduce a new metric called *hop change metric* in order to represent the changes in the network topology due to mobility. We determine a threshold value based on this metric in order to decide the full update time dynamically and cost effectively. The proposed approach (La-DSDV) is compared with the original DSDV and ns-DSDV. Simulation results show that our threshold-based approach improves the packet delivery ratio and the packet drop rate significantly with a reasonable increase in the overhead and the end-to-end delay.

Keywords: DSDV, proactive routing protocols, adaptive routing protocols, mobile ad hoc networks, mobility metric, hop change metric, update time.

1 Introduction

Mobile ad-hoc networks (MANETs) have become very popular for military applications, disaster recovery operations in which the fixed network infrastructure might not be available due to wars, natural disasters, and the like. MANETs combine wireless communication with high degree node mobility. In these dynamic networks, nodes that are within each other's transmission range can communicate directly, while those that are far apart must rely on intermediate nodes to relay messages. A routing protocol in such a network is responsible for finding/updating routes and providing communication between end points through cooperative intermediate nodes. A variety of routing protocols have been proposed for mobile ad hoc networks in the literature, but no proposed protocol has been accepted as standard yet.

There are three kinds of routing protocols on MANETs: proactive, reactive and hybrid protocols. Proactive routing protocols use periodic exchange of control messages between nodes to build up a routing table. In proactive routing protocols, routes are ready before they are needed. So, two nodes who want to communicate with each other can start communication immediately by using the route available in their

routing table. However, there is a considerable overhead due to the high number of control messages in order to keep routing table up-to-date. On the other hand, reactive protocols start route discovery when a route is needed. The biggest disadvantage of these protocols is the latency caused by the route discovery process which is need to be carried out before the communication between the end nodes starts. There are also some hybrid protocols which combine proactive and reactive routing protocols and take advantage of both.

One of the main research challenges in mobile ad hoc networks is designing adaptive, scalable and low-cost routing protocols for these highly dynamic environments. Proactive routing protocols mainly use static update period time for keeping routes up-to-date, which is against the dynamic nature of MANETs. This might cause low packet delivery ratio and high packet drop rate under high mobility. Updating routing table adaptively is the main focus here. In this research, we use DSDV as an exemplar routing protocol which is one of the most popular proactive routing protocols on MANETs. In this research, we aim to improve packet delivery ratio, packet drop rate, overhead and end-to-end delay by changing the update period time dynamically instead of using the static update period time as in the original DSDV. We introduce a new metric called *hop change metric* in order to achieve that. The proposed approach is low cost in terms of computation and communication. There is no message exchanging between nodes in order to decide the update period time. Every node decides upon updating locally. The simulation results show that the proposed approach (La-DSDV) improves the packet delivery ratio and the packet dropping rate with a reasonable increase in the overhead and the end-to-end delay. Moreover, the hop change metric could reflect different mobility patterns dynamically and cost effectively in proactive routing protocols.

The remainder of this paper is organized as follows. Section 2 describes the original DSDV protocol and explains how DSDV selects routes and responds to changes in the network topology. Section 3 summarizes some improvements proposed for DSDV in the literature. Section 4 introduces our method and the simulation environment. Section 5 presents the simulation results on networks with varying mobility patterns. Section 6 concludes and presents the future work.

2 DSDV

Destination-Sequenced Distance Vector routing protocol (DSDV) [1] [2] is a proactive routing protocol based on the Distributed Bellman-Ford algorithm. In DSDV, each route has a sequence number originated by the destination node which indicates how old the route is. Each destination node originates a sequence number every time it builds a new route. Sequence numbers are proposed to prevent routing loops to occur in the network.

In DSDV, each node maintains routing information (destination node, next hop, hop count, destination sequence number) for all nodes in his routing table and this

information is updated periodically. So, each node must be in active mode at all the time. Even if there is no change in the network topology, the periodic update occurs. Hence these updates result in high traffic overhead in DSDV.

During the route selection process, route updates are carried out when the destination sequence number in a control message is bigger than the sequence number in the routing table. This ensures to use always the newest information from the destination. If the sequence numbers are equal, the shortest route is selected.

There are two types of response to topology change in DSDV: full and incremental update. Immediate advertisement broadcasts information when a new route is added, or when a link is broken, and the like. In these situations, necessary update messages are propagated immediately to the neighbor nodes. Where all routing information is sent in the full update, only entries that have changed are sent to other nodes in the incremental mode.

3 Related Work

There are some approaches to address issues in routing protocols caused by the dynamic topology of MANETs. One of the solutions is using hybrid routing protocols which combines the best properties of both proactive and reactive protocols by changing the routing protocols adaptively. This change depends on the current configuration of the network and the behavior of the nodes. ZRP [9], ZHLS [10] and HARP [11] are some examples of hybrid protocols. ZRP reduces the control overhead of proactive routing protocols and decreases the latency of routing discovery in reactive routing protocols. While ZHLS reduces the communication overhead, HARP reduces delays happening during early path maintenance. Another approach is to change some parameters of routing protocols dynamically based on some criteria such as mobility, power, and traffic.

Several improvements have been proposed for DSDV in the literature. One of these approaches is ARM-DSDV introduced in [7]. ARM-DSDV is a control mechanism which dynamically adapts the routing protocol with the following two metrics: mobility metric and route demand metric. The mobility metric indicates the changes in the number of neighbors; the route demand metric indicates which destinations are currently involved in data forwarding. ARM-DSDV dynamically adjusts the update period time and the content of the control messages based on these metrics. It is believed that updates should be more frequent under high mobility. That's why the mobility metric is calculated by considering the changes in the number of neighbors (one hop away nodes). Each node evaluates the mobility metric and sends it to his neighbor nodes in order to obtain aggregate mobility metric which is the average mobility metric in the neighborhood. This might cause high communication overhead. Different nodes can send their control messages at different times in this technique. This is a different approach than the original DSDV sending all updates together. The update period control and the update content control are carried out locally in each

node with some overhead. This approach can be applied to any proactive protocol as stated in [7].

One of the most important problems in the original DSDV is the diagnosis of invalid routes. This is called the stale route problem and many improvements on DSDV focus on this problem. The stale route problem occurs when a route is broken. Since there is no alternative route maintained in the routing table, the next periodic update has to be waited for re-building the route. One of the approaches work on this issue is Imp-DSDV [4]. In the original DSDV, a node who has observed a broken link assigns infinity to the hop metric for this link in his routing table and waits for the next update period. However the nodes maintain an alternative route in their routing tables in Imp-DSDV. In other words, when a link is broken, the alternative route is used for communication immediately. They introduce a new field called *type* which holds the validity of the route in the routing table.

Another improvement on DSDV protocol is Eff-DSDV [3] which aims to use an alternative route again when a broken link is detected. In the original DSDV, when there is a high number of broken links, the stale route problem causes low packet delivery ratio [6]. In Eff-DSDV, when a node detects a broken link, the node uses a temporary link from his neighbors which have a valid route to destination. An alternative link is created by sending two one-hop messages; ROUTE-REQUEST and ROUTE-ACK. An additional field for route update time to the routing table is introduced. The update time is embedded into ROUTE-ACK message and it is used for selecting a temporary route. If a node receives multiple ROUTE-ACK messages from different neighbors, it will choose the route which is updated recently.

Another approach proposed to solve the stale route problem is I-DSDV [5]. This algorithm has improved the packet delivery ratio without any message exchange and any overhead. This approach also reduces the end-to-end delay and the number of dropped data packets. It shows a better performance than both the original DSDV and Imp-DSDV. In I-DSDV, each node keeps two routing tables. They are called the main routing table and the secondary routing table respectively. As a result of these two routing tables, every node has two routes for each destination. The routes in the secondary routing table can be valid or invalid. The valid routes in this table should have the same hop metric and the same destination sequence number as in the main routing table. However the next hop to the destination is different. At the beginning, all routes in the secondary routing table are invalid. When a node receives a route update with the same metric and the same sequence number as in his main routing table, but with the different next hop, the secondary route is updated. When a route in the main routing table is broken, it is replaced with the respective route in the secondary routing table.

Another approach proposed for DSDV is the optimization in ns-2 [8]. ns-2 is a simulation tool developed at the University of California, Berkeley. In this optimization, a metric which represents the number of changes in the routing table is introduced. When the change multiplied by 3 is bigger than the routing table size, a full update by all nodes has started. The algorithm increases the packet delivery ratio

considerably. However it also increases the end-to-end delay and the overhead of control messages. The algorithm sends a lot of full updates under high mobility. It forces the nodes to send the full update almost every second (where the periodic update time in the original DSDV is 15 seconds) under high mobility.

4 The Method

The full dumps of the nodes can be transmitted relatively infrequently when little movement of mobile nodes is occurring [1]. On the other hand, the periodic update needs to be more frequent under high mobility due to the high number of changes in the network topology. This is the basic assumption in this study. We introduce a new metric called *hop change metric* which represents the changes in the number of hops in the routing table. We define the Equation 1 to calculate this metric. This metric is calculated at each node when an update packet is arrived. In the Equation 1, $HopCount_{New}^i$ is the new hop count between the current node and the node i . $HopCount_{Previous}^i$ is the previous hop count at the previous update. $T_{NewUpdate}$ and $T_{PreviousUpdate}$ are the last and the previous update period times respectively.

It is believed that the change in the hop count is a good representative of the mobility. The high number of change in the hop count can be a sign of high mobility. Furthermore, this change affects every node in the same way. Hence we can determine a periodic update time without exchanging any information between nodes. This is a simple and low cost approach in terms of computation and communication. It does not introduce any communication cost to determine the update time as in the mobility metric given in [7]. This mobility metric based on the neighborhood change is well accepted and shown to be a more accurate measurement than the mobile speed [12]. The hop change metric defined here is more cost effective than the mobility metric [7]. Furthermore, the hop change metric is believed to give a broader view of mobility than the neighborhood change, which might differ considerably from one node to another, in proactive routing protocols.

$$\frac{\sum_{i=1}^{i=Number\ of\ Nodes} |HopCount_{New}^i - HopCount_{Previous}^i|}{T_{NewUpdate} - T_{PreviousUpdate}} \quad (1)$$

Number of Nodes

Our approach shows a similarity with the optimization done in ns-2. While we only consider changes in the number of hops, ns-DSDV takes into account any change to the routing table such as adding a new node, updating sequence number/hop count. Some of these changes do not have a direct relation with the topology change due to mobility. Furthermore, the threshold value for the hop change metric is defined empirically here. Our purpose here is to update routes more frequently when needed under high mobility. As it is seen in the simulation results, our method outperforms ns-DSDV.

In this approach, we determine a threshold value for the hop change metric empirically. First of all, we determine the range of the hop change metric and, then evaluate the packet delivery ratio, end-to-end delay, network overhead and drop rate at different values in this range. A threshold is selected based on these performance metrics. As mentioned before, we calculate the hop change metric when an update packet is received. If the calculated hop change metric is bigger than the defined threshold, a full update is carried out.

We train our system to define the threshold by using a network under medium mobility (pause time = 10s). We evaluate the performance of our method by using networks with varying mobility levels from low to high. Lastly, we compare our protocol (La-DSDV) with the original DSDV protocol and the ns-DSDV. The following performance metrics are used in the comparison: the packet delivery ratio (PDF), the overhead, the end-to-end delay and the packet drop rate.

4.1 Simulation Environment

We evaluate our method on networks simulated by using the ns-2 simulator [8]. Our simulation parameters are presented in the Table 1. The parameters not given here are the default parameters of the simulator.

Table 1. The simulation parameters.

Parameter	Value
Network Dimensions	1000*1000
Number of Nodes	100
Packet Traffic	CBR with 60 connections
Speed	0-20 m/s
Pause Time	0, 5, 10, 15, 20, 25 s
Transmission Range	250 m
Simulation Time	500 s
Mobility Model	Random waypoint

5 Simulation Results

First of all, we analyze the hop change metric and define different threshold values. Table 2 shows the performance of our method at different threshold values. We use a network under medium mobility (pause time=10) for training and, determining a threshold value for the hop change metric.

Table 2. The performance at different threshold values.

Hop Change Metric	PDF	Overhead	End-to-End Delay	Drop Rate
0.08	0.6658	3.546	1109.76	0.3277
0.09	0.6706	3.217	820.125	0.3237
0.1	0.6859	2.794	398.057	0.3097
0.11	0.6647	2.788	290.819	0.3324
0.12	0.6682	2.763	455.821	0.3288
0.13	0.6616	2.736	303.004	0.3559

According to the results given in the Table 2, the threshold value is set to 0.1 in this study. Even this value does not show the best result, it shows a good result in each performance metric. Figure 1 shows the packet delivery ratio at different threshold values. In terms of PDF, it performs the best at 0.1.

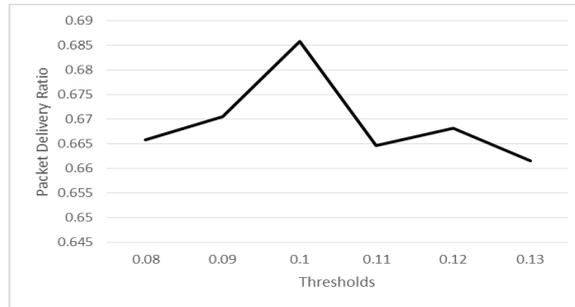


Fig. 1. Packet delivery ratio at different threshold values

For testing, various networks are created by running simulations at different pause times (0, 5, 15, 20, and 25 seconds). We run DSDV, ns-DSDV and our approach on these networks with varying mobility patterns and compare their performance metrics (packet delivery ratio, network overhead, end-to-end delay and packet drop rate). Figure 2 demonstrates the packet delivery ratio for La-DSDV, DSDV and ns-DSDV protocols on various networks. The figure shows that La-DSDV achieves better results in low, medium and high mobility than the original DSDV. It can be concluded that La-DSDV is preferable than the original DSDV from the packet delivery ratio point of view. It outperforms ns-DSDV considerably under medium mobility. Since we employ a network under medium mobility for training, the best results are generally obtained for the networks simulated at pause time of 10 seconds in testing. The results could be improved by determining different thresholds for different networks with varying mobility patterns.

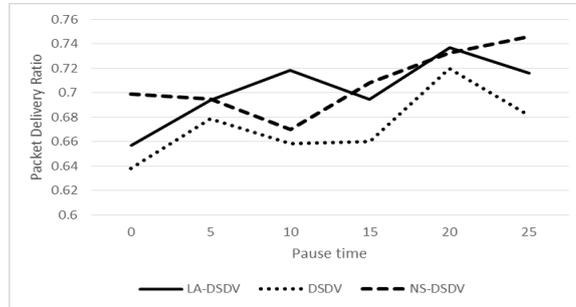


Fig. 2. Packet delivery ratio of La-DSDV, DSDV and ns-DSDV

Figure 3 depicts the end-to-end delay for networks with varying mobility patterns. As it is seen, La-DSDV has a lower end-to-end delay than ns-DSDV. The update time which is decreased down to one second in ns-DSDV might cause this delay. Our approach also does not increase the delay much when compared with the original DSDV.

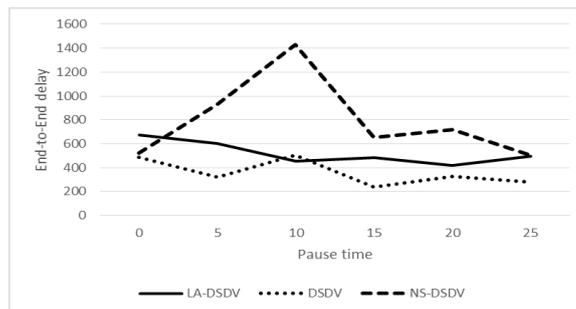


Fig. 3. End-to-end delay of La-DSDV, DSDV and ns-DSDV

Figure 4 demonstrates the overhead caused by the routing control packets in each protocol. La-DSDV protocol adds a little overhead to the original DSDV; however it improves other performance metrics such as the packet delivery ratio and the packet drop rate. The increase in the overhead is caused by the additional updates when there is a big change in the network topology. It is a trade-off to make between the packet delivery ratio and the overhead. The La-DSDV protocol has much lower overhead than the ns-DSDV protocol.

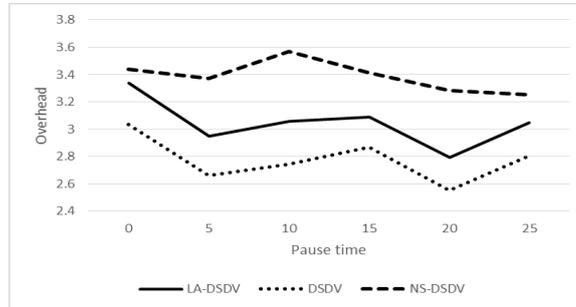


Fig. 4. Network overhead of La-DSDV, DSDV and ns-DSDV

Finally, the packet dropping rate is demonstrated in Figure 5. La-DSDV protocol has less packet drop rate than the original DSDV protocol. The details of the simulation results are presented in Table 3. Even though mobility is one of the biggest factors on the results, other factors such as network topology, traffic patterns also play a part. These factors could be investigated by running more number of simulations in the future.

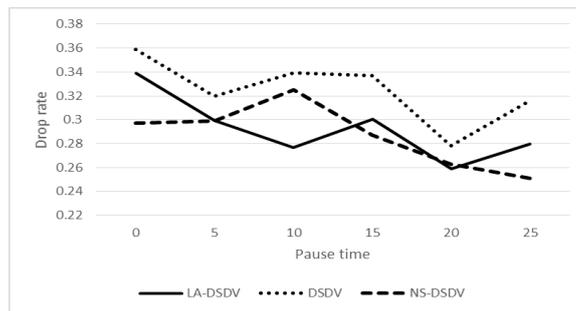


Fig. 5. Packet drop rate of La-DSDV, DSDV and ns-DSDV

Table 3. The performance of La-DSDV on networks with varying mobility levels

Pause time	PDF	Overhead	End-to-End Delay	Drop Rate
0	0.6569	3.336	675.888	0.3282
5	0.6941	2.95	603.186	0.2855
10	0.7185	3.058	454.939	0.2769
15	0.6946	3.087	482.507	0.2973
20	0.7369	2.794	421.667	0.2504
25	0.7162	3.045	497.444	0.293

We have also created five different networks under medium mobility and evaluated all routing protocols (La-DSDV, DSDV and ns-DSDV) on these networks. The

average results can be seen in Table 4. This table also shows that La-DSDV protocol achieves a better packet delivery ratio and packet drop rate than the original DSDV. The increase in the end-to-end delay and the overhead is reasonable. The average values of overhead and end-to-end delay are between the original DSDV and ns-DSDV (much closer to the original DSDV).

Table 4. The average performance of the protocols on five networks under medium mobility.

Protocol	PDF	Overhead	End-to-End Delay	Drop Rate
La-DSDV	0.73634	2.6864	366.191	0.25968
DSDV	0.71784	2.4902	286.701	0.27698
ns-DSDV	0.73428	3.2086	619.602	0.26026

To sum up, La-DSDV improves the packet delivery ratio and the packet drop rate with a reasonable increase in the overhead and the end-to-end delay. It also outperforms ns-DSDV. Our proposed approach shows a good performance on networks with varying mobility patterns from low to high mobility. The hop change metric represents the topology changes due to mobility well and it can be used for determining the full update time adaptively. Moreover, it does not introduce any communication cost to determine the update time as in the mobility metric based on the neighborhood change given in [7]. In proactive routing protocols, the hop change metric is also believed to give a broader view of mobility than the neighbourhood change which might differ considerably from one node to another. To conclude, we introduce a new metric which reflects different mobility patterns dynamically and cost effectively in proactive routing protocols here. This new metric can be used in various applications on MANETs.

6 Conclusion

In this research, a new lightweight threshold-based scheme is proposed in order to improve the low packet delivery ratio of the original DSDV under high mobility. We define a new metric called *hop change metric* which shows the changes in the number of hops in a routing table. It is believed that this metric can be a good representative of the changes in the network topology due to mobility. The results support this belief and show that our approach based on this metric improves the packet delivery ratio and the packet drop rate with a reasonable increase in the overhead and the end-to-end delay. This is a simple method which decides upon the update time without communicating with other nodes in the network. Since the communication between nodes is the main cause of battery depletion, it is an important attribute for the nodes that usually run on battery power in MANETs. This new metric can be used in various applications on MANETs. In the future, we would like to work on an adaptive system which changes the periodic update time dynamically by taking into account other criteria such as traffic, power as well. Moreover, the metric which represents the mobility, the changes in the topology the best is aimed to be explored.

References

1. C. E. Perkins, P. Bhagwat,: Highly Dynamic Destination-Sequenced Distance Vector Routing (DSDV) for Mobile Computers, In Proceedings of the Conference on Communications, Architectures, Protocols and Applications (SIGCOMM), pp. 234-244, (1994).
2. G. He,: Destination-Sequenced Distance Vector (DSDV) Protocol, Networking Laboratory, Helsinki University of Technology, (2002).
3. K. Ur Rahman Khan, A. Venugopal Reddy, R. U Zaman,: An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and Its Performance Comparison, In Proceedings of the Second UKSIM European Symposium on Computer Modeling and Simulation, pp. 506 – 511, (2008).
4. J. Lu, B. Zhang, G. Han, J. Wang, W. Dou,: A New Improvement on DSDV, In proceeding of the 7th International Conference on Wireless Communications, Networking and Mobile Computing (WiCOM), pp. 1-4, (2011).
5. T. Liu, K. Liu,: Improvements on DSDV in Mobile Ad Hoc Networks, In Proceedings of the International Conference on Wireless Communications, Networking and Mobile Computing, pp. 1637-1640, (2007).
6. A. H. Abd Rahman, Z. A. Zukarnain,: Performance Comparison of AODV, DSDV, and I-DSDV Routing Protocols in Mobile Ad Hoc Networks, In Proceedings of the European Journal of Scientific Research vol. 31, pp. 556-576, (2009).
7. S. Ahn and A. Udaya Shankar: Adapting to Route-demand and Mobility (ARM) in Ad hoc Network Routing, In Proceedings of the 9th International Conference on Network Protocols, pp. 745–764, (2001).
8. ns-2, <http://www.isi.edu/nsnam/ns/> (accessed in July, 2013).
9. Zygmunt J. Haas and R. Pearlman,: The Performance of Query Control Schemes for the Zone Routing Protocol, In Proceedings of the Journal IEE/ACM Transactions on Networking, vol. 9, pp. 427-438, (2001).
10. M. Joa-Ng and I-Tai Lu,: Peer-to-peer Zone-based Two-level Link State Routing for Mobile Ad Hoc Networks, In Proceedings of the IEEE Journal on Selected Areas in Communications vol.17 , pp. 1415-1425, (1999).
11. N. Nikaein, W. Shiyi, and C. Bonnet,: HARP: Hybrid Ad hoc Routing Protocol, In Proceedings of the In International Symposium on Telecommunications (IST), (2001).
12. B. Sun, K. Wu, Y. Xiao, and R. Wang, Integration of mobility and intrusion detection for wireless Ad Hoc networks, International Journal of Communication Systems vol. 20 no. 6, pp. 695-721, (2007).