DSDV Extension to Enhance the Performance of Ad Hoc Networks in High Diverse-Velocity Environments

Mada' Abdel Jawad, Saeed Salah (✉), Raid Zaghal
Al-Quds University, Jerusalem, Palestine
sasalah@staff.alquds.edu

Abstract—Mobile Ad-Hoc Networks (MANETs) are characterized as decentralized control networks. The mobile nodes route and forward data based on their routing information without the need for routing devices. In this type of networks, nodes move in an unstructured environment where some nodes are still fixed, others are moving in a constant velocity, and others move with diverse velocities; and thus, they need special protocols to keep track of network changes and velocity changes among the nodes. Destination Sequenced Distance-Vector (DSDV) routing protocol is one of the most popular proactive routing protocols for wireless networks. This protocol has a good performance in general, but with high speed nodes and congested networks its performance degrades quickly.

In this paper we propose an extension to the DSDV (we call it Diverse-Velocity DSDV) to address this problem. The main idea is to modify the protocol to include node speed, determine update intervals and the duration of settling time. To evaluate the performance of the new protocol, we have carried a number of simulation scenarios using the Network Simulator tool (NS-3) and measured relevant parameters such as: packet delivery ratio, throughput, end-to-end delay, and routing overhead. We have compared our results with the original DSDV and some of its new variants. The new protocol has demonstrated a noticeable improvement of performance in all scenarios, and the measured performance metrics outperform the others except the average delay where the performance of the new protocol was modest.

Keywords—MANET, DSDV, I-DSDV; E-DSDV, O-DSDV, Simulation Network Performance, NS3;

1   Introduction

Mobile Ad-hoc Networks (MANETs) are special types of Wireless Sensor Networks (WSNs) where the nodes (computers, laptops, smart phones, etc.) are connected without a predetermined infrastructure. The topology and routing information keep changing frequently. This means that there is no need to use dedicated routers or access points to connect any two nodes; a node can act as a router to forward traffic to other nodes. All nodes are working collaboratively in a distributed fashion to keep the network alive. Normally, nodes belonging to this type of network have a small

memory and limited battery capacity. Furthermore, links between nodes are wireless making them subjected to path disconnections and frequent packet losses. The mobility nature of the nodes makes it difficult to build routing information in a coherent way. MANETs are currently being deployed in many disciplines. For example, in the military arena they can be used to exchange information between the vehicles, soldiers and headquarters. On a more personal level, MANETs can simplify the interconnection between lightweight devices such as laptops, smartphones, and tablets. They can also be helpful in disaster situations when the network infrastructure is subjected to sudden and catastrophic damages that might affect the stability and lifetime of the network. MANETs face several challenges such as: transmission range is very limited; security issues due to the wireless environments; the power and processing resources are very limited. Furthermore, the mobility nature of a MANET makes its topology highly dynamic which leads to an increase the packets' loss and route changes.

Many networking protocols were proposed to deal with these challenges. Routing protocols are special algorithms and mechanisms used by wireless devices to find the best path (route) for forwarding data between two (or more) nodes. Many of them were proposed in the literature, and enhanced versions continue to appear. Efficient routing protocols usually try to optimize several performance metrics such as: control overhead, bandwidth consumption), throughput, end-to-end delay, security, fault tolerance and energy efficiency. The proposed routing protocols usually focus on some of these performance aspects and ignore the others based on their target applications and usage purposes. There are different classifications for routing protocols suggested by the research community; one of these main classifications is the work done by [3]. In this work, the authors classified existing routing protocols into three many categories:

1. The proactive or table-driven routing protocols
2. The reactive or on-demand routing protocols
3. The hybrid protocols which merge concepts from both the proactive and reactive methods

In the proactive routing method, every node maintains one or more routing tables, these tables contain routing information about all the nodes within the network. These protocols update the table(s) periodically which might lead to increasing network overhead, bandwidth and power consumptions, but it minimizes the end-to-end delay and keeps the routing information up to date. Examples of such category of protocols are Optimized Link State Routing Protocol (OLSR) [17], Destination Sequenced Distance Vector (DSDV [4], and Wireless Routing Protocol (WRP) [19]. In the reactive routing method, the node's routing information will not be maintained frequently or periodically. If a node needs to communicate with another node, it will first send a request (broadcast) to the whole network, every node will pass the request until it reaches the destination node. Next, the destination node will reply using the same path which was used for the request message. In this case, the source node will not have the routing information until it receives the reply message. This process increases the end-to-end delay, but it saves both thee bandwidth and power consumption. Examples

of this category are: AODV Ad Hoc On-demand Distance Vector (AODV) [5] and DSR Dynamic Source Routing (DSR) [7], among others.

Finally, the hybrid routing protocols were built to use concepts from methods to get the maximum performance. These protocols will maintain a routing table for nodes within some ranges and will send routing requests for the nodes located outside this specified range. Examples of this routing method are: ZRP Zone Routing Protocol (ZRP) [29] and Order One Routing Protocol (TORA) [26], among others. We implemented several simulation scenarios using NS3 simulator and compared the results of the new protocol protocols with four other routing protocols which have provided an improvement to the DSDV protocol and relevant to our protocol. The names of these protocols are: DSDV, I-DSDV, E-DSDV and the Optimized DSDV routing protocols (O-DSDV). The comparative study was done based on four performance metrics: packets delivery ratio, throughput, End-to-End delay and routing overheard.

2 Related Work

In this section, we present a brief description of several protocols that have been proposed by the research community, and mainly we focus on the DSDV and its variants since these are relevant to the work suggested in this paper. They are: the generic DSDV, I-DSDV, E-DSDV and the Optimized DSDV. In the following subsections we give a brief overview of these protocols.

2.1 Destination sequenced distance vector

DSDV routing protocol [4] is an Ad-Hoc network proactive loop free distance routing protocol. In this protocol, the nodes periodically broadcast packets with routing information about themselves. These packets contain fields such as: destination, hop-count, and sequence number, among others. The neighbors receive these messages and update their routing tables accordingly. In this case, the updates will be added to the routing table in certain situations, e.g., when there is no routing information about the destination in the routing table, or the updated message has a higher sequence number, or it has the same sequence number but with a shorter path (smaller hop count). New updates in the routing table will be immediately broadcasted after increasing the metric parameter in the records by one. The routing information in the routing table will periodically be rebroadcasted as well. Fig 1 illustrates the broadcasting mechanism used by this protocol.
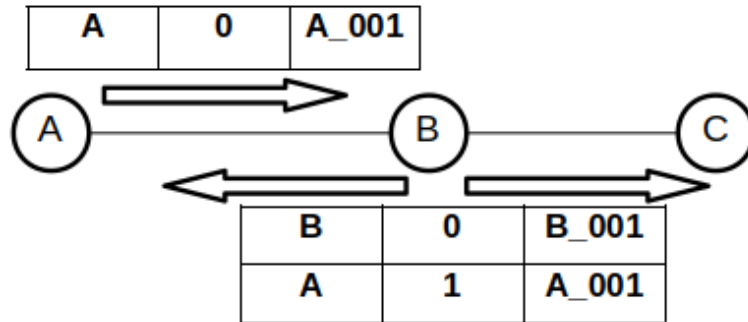
Fig. 1. An example of a routing message exchange used by the DSDV routing protocol

The DSDV routing table mainly contains the following fields:

- **Next hop:** It mainly represents the first node (the closest node's neighbor) to be visited to reach the destination.
- **Metric:** It is a measured value that represents the number of nodes which the data packet will visit before reaching the destination.
- **Sequence number:** It is a sequential number to distinguish old updates from new ones.
- **Settling time:** This metric represents the time that a node will wait before broadcasting the incoming updates (any update for the record will reset the timer).
- **Hold down time:** It is the time that a node will wait before considering the record is expired (i.e., broken connection).

2.2 Efficient DSDV routing protocol E-DSDV

E-DSDV [11] was presented as an extension to the DSDV. The main purpose of this protocol is to reduce network congestion. DSDV routing protocol is designed to choose the shortest path to reach any node within the network, but this principle does not apply to the E-DSDV protocol, which was mainly proposed to deliver routing data updates as quickly as possible, even if this is done by choosing non-optimal paths. The purpose of this mechanism is to avoid network bottlenecks by eliminating settling time that was used in the DSDV protocol to reduce oscillation and replacing it with a new mechanism. The nodes in this protocol always consider the updates only when they have higher sequence numbers. In case a node receives a route advertisement message with higher sequence number, it will advertise the message immediately. The node will discard the new route advertisement messages having the same sequence number or smaller hop-count.

This protocol does not give importance to the length of the path; yet is concentrates on paths that do not contain network congestion. The use of this protocol increases the end-to-end delay; i.e., passing data over longer paths increases the time required to move from source to destination. Improved DSDV Routing Protocol (I-DSDV) I-

DSDV [22] improves the packets' delivery ratio of the DSDV in high mobility scenarios. A new field was added to the DSDV routing message. In some cases, routing tables contain broken links and these links are usually fixed when the data updates arrive from the source. This process takes time to be able to reuse the path. I-DSDV proposed a new approach to repair broken links without having to wait for the source. The new message has four fields: destination, type, hop-count and sequence number. Nodes with stale routes will send a routing message and will set the message type as INVALID. The neighbors of the sender will deal with this message as a route request. If these neighbors have a VALID route information with a better or an equal sequence number and smaller or equal hop-count, they will broadcast the VALID routing information. On the other hand, if the neighbors do not have a valid routing information, they will rebroadcast the received INVALID routing message in case they need this route. Otherwise, they will remove the route from their table only.

The repairing process of broken links in this protocol requires an effective link with the same length as the broken link or less. The existence of this condition reduced the possibility of using the available paths only because it is longer. In our point of view, it is better to use any available path – even if it is a bit longer than the optimal path – as a temporary solution until the next updates arrive.

2.3 Optimized DSDV routing protocol

O-DSDV [28] is a modified version of the DSDV, which was mainly presented as a new approach to reduce end-to-end delay, increase throughput, and maximize packet delivery ratio in VANETS' applications. The authors analyzed the performance of the DSDV protocol and noticed that its performance is low, especially in high-traffic environments, and thus they made several modifications to suit these conditions; The O-DSDV suggests two modes, high velocity mode and low velocity mode. While the protocol will act like the original DSDV in low velocity modes, it reduces its update interval and sets its settling time equal to zero for the incoming updates in high velocity scenarios. When the speed of the node exceeds 25 m/s, the protocol gradually reduces the update period until the speed reaches 40 m/s. The update period is settled at a predefined low value. For settling time, it has two values one for high speeds and the other for low speeds. In this way, the received updates will be sent immediately, and the interval between nodes' updates will be shorter in high mobility environments.

This approach enhanced the DSDV performance, but it does not behave well in highly diverse velocity environments. The neighbors of the high-speed node will not be informed about this node's speed, and consequently, they will deal with the updates from such nodes as any other node. This will lead to delays when the updates advertise the high-speed nodes. Processing updates coming from high-speed nodes quickly is necessary because such updates lose their value quickly.

3 Proposed DSDV Extension

This section describes the new settings and changes that we have made on the DSDV protocol. We also include a discussion on the justification of making these modifications, our methodology to implement them and the assumptions that we have considered.

3.1 Assumptions

We assume that all nodes in the network are using this new protocol and are equipped with location and velocity determination devices like Global Positioning System (GPS). As the information received from these devices will be used for controlling several parameters in the DSDV such as periodic update time, hold-down time, and settling time.

3.2 Methodology

As previously mentioned, several techniques have been proposed to enhance DSDV's stability. However, these mechanisms sometimes actually degraded the performance of the protocol, especially in the highly-diverse velocity environments. Therefore, in this work, we suggest a new approach that is specially designed to increase the performance the DSDV algorithm in the high mobility environments. We followed the scientific methodology in which we propose the theoretical part first, and then validate it using simulations. Our methodology consists of five procedures:

- The nodes that receive routing information updates about their neighbors within the scope or two hops away, should advertise the received updates immediately without any delay.
- High speed nodes should gradually reduce the update intervals and settling time according to the speed.
- High speed nodes should inform other nodes about their speed mode to help them calculate "Settling" and "Hold Down" times.
- The nodes that detect a broken connection should try to recover it by sending a route request to the neighbors only.
- The nodes will prefer the routes through the lowest velocity nodes when they receive several route suggestions to a specific destination with the same hops-count.

In the DSDV internal operation, the goal of the "settling time" is to ensure that the node has the shortest path to every other node in the network. We found that the "settling time" should not be applied to all types of routes. For example, as illustrated in Fig 2, when node (A) receives a routing information about one of its neighbors, it will wait for the specified "settling time" before rebroadcasting it to the other neighbors. Although node (A) will wait for the specified "settling time", it will not get a shorter path to these neighbors because it has the best path already. The shortest path in this

case is only between the node and itself. This is how we applied the first step in our methodology.
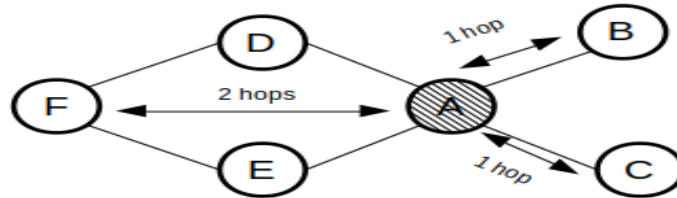

Fig. 2. Applying settling time to one-hop or two-hops neighbors will not guarantee a shorter route.

The second and third steps in our methodology will affect nodes with high velocity. The update period in the DSDV is fixed, and in most applications its value is fixed to 15 seconds. This time is considered very long in high diverse velocity situations. For example, Fig 3 illustrates a scenario where node M is moving in high velocity and the other nodes are either stationary or moving very slowly. An intermittent line represents M's signal range. At some points, node M will send its update to all nodes through its neighbors (E and F, in this case), then it will leave the area and its neighbors will change few seconds later. The nodes in the network will lose the route to node M quickly, and node M will have to resend a new update after the update period expires. In our approach, we will take care of this problem by:

1. Forcing the update time to be variable
2. By making it proportional to the node's velocity. The position of a high velocity node changes rapidly, and thus it sends its updates more frequently to its new neighbors, and remain connected with the rest of the network.
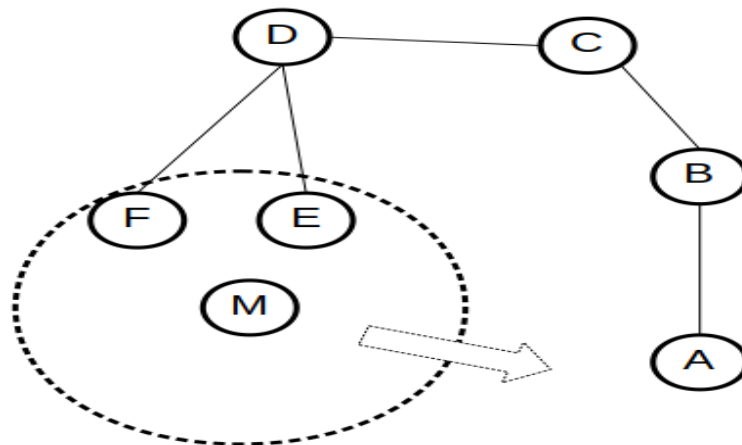

Fig. 3. Example of high mobility scenario.

To consider the node's velocity in the DSDV calculations, we have modified the update time by including another parameter, i.e., the speed of the node. The new suggested formula of the update time is as follows:

$$Updatetime = T_d - \frac{V_n}{V_{max}} \times (T_d - T_{min})$$

(1)

where:

- Td represents the default update time.
- Vn represents the actual node velocity.
- Vmax is the maximum node velocity.
- Tmin is the least update time period.

In case the node's velocity is very low, or if the node is stationary, then Vn = 0, and this means that the value of the update time equals Td which is the default value in this case. Furthermore, when the velocity of the node is increased, the value of the update time will be decreased accordingly. To keep the update time above zero the lower limit to the Tmin value has been added.

Back to Fig 3, when nodes E and F receive routing information update about node M, they will wait for the expiry of the "settling time" before resending this update to the other nodes. For every node which receives this update, it will apply the same policy which will lead to a considerable delay of these updates by the remote nodes. This type of delay can significantly affect the performance of DSDV protocol, especially when nodes move in high velocity. In Fig 4, the routing information of node M will reach node A after node M leaves the area, and its neighbor nodes are changed. Thus, we suggest that every node should inform every other node about its velocity. The nodes that receive the routing updates should set their own "settling" and "hold down" times based on the velocity of the source node. We suggest the following formula to calculate the adjusted "settling time":

$$SettlingTime = T_s - \frac{V_s \times T_s}{V_{max}}$$

(2)

where:

- Ts represents the default settling time
- Vs represents the velocity of the sender.
- Vmax is the maximum velocity.

The above equation (Eq. 2) estimates the settling time as inversely proportional to the velocity of the source node. In this way, we can improve the availability of the routing information about high velocity nodes through the network for a shorter time. Furthermore, the velocity of the source nodes can be used for tuning the hold down time interval. We note that the routing information about the fast points will expire

early. The following equation is suggested to set the holding down timer for the received routing records:

$$HoldsTime = \frac{T_d \times V_n}{V_{max}} \times (\alpha - \gamma)$$

(3)

Where γ is the minimum limit and α is a constant value which has the default value of 3.

We can also use the added node's velocity to optimally select the most stable paths. For example, as illustrated in Fig 5, node A has two paths to node D, and the two paths have the same hop-count. In this case, node A will use only one path to send data to node D. When node A receives routing information about node D through node C first, it will use this path to send data to node D. In this case, we argue that this node (which has two paths to the same destination and they both have the same hop-count) will select the neighbor with the slower velocity since it has more updated and correct entries.
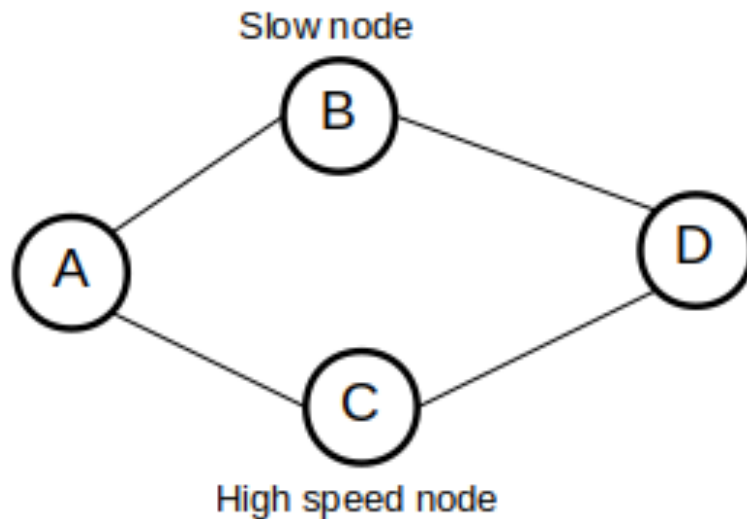

Fig. 4. An example of selecting a stable path among several ones.

The last enhancement to be considered in this study is the ability to recover the broken connections. When a connection between two nodes is broken, this connection will not be recovered until receiving a new routing information, even though it is possible to recover this connection in a shorter time. For example, as illustrated in Fig 5 the connection between nodes A and D is broken because of node's A movement, node A has two other paths to node D, but it will not know about these paths until node D sends its next periodic update. To overcome this situation, we suggest a new mechanism to send a recovery request to other neighbors to keep the link between the two nodes alive if possible. Thus, in Fig 6 node A will send requests to both nodes B

and C. In case node B or C has a connection to node D it will reply to the request otherwise it will do nothing.
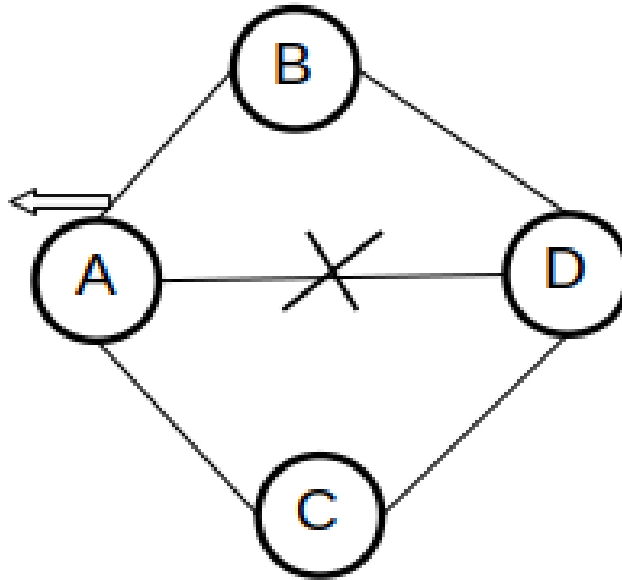

Fig. 5. An example of recovering a broken connection.

To consider all these modifications, we have modified the generic DSDV message header format by including two new parameters: node speed and message type. The node's speed parameter will register the node's current velocity, and the message type parameter which will identify the message type (normal update or recover connection request). Both parameters will be added by the source node in every message update.

It is worth noting (as illustrated in Fig 6) that the size of the header remains constant. After studying the generic DSDV header, we found that 4 Bytes (32 bits) have been allocated to the hop-count parameter which is considered extremely big when compared to the hop-count size of other routing protocols like the OLSR and AODV, where the hop-count size is 1 Byte in both of them. Therefore, we found a good opportunity to re-use half of the space allocated to the hop-count parameter in DSDV (16 bits) to serve us to achieve the goals of this work. In the new modification, we argue that 16 bits, which can represent more than 65000 hops should be more than enough for the hop-count counter. We use the remaining 16 bits to store the new parameters (speed and type) and thus we can expand the functionality and add new features without increasing the header size. The new header layout is shown in Fig 6.
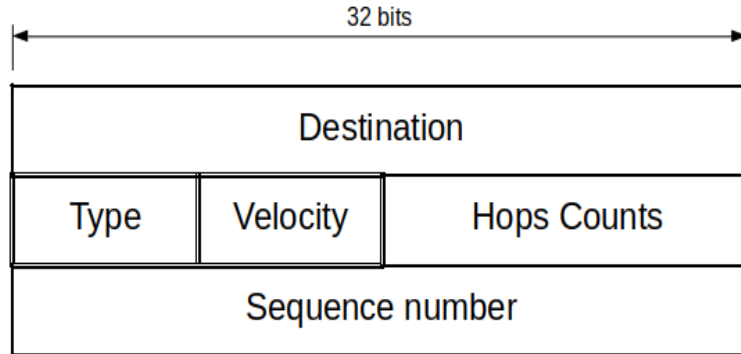
Fig. 6. The new proposed structure of the DSDV header.

As shown in Fig 6, the size of the new update message is still the same (12 Bytes) with two new parameters: type and speed. The type parameter can have two states: UPDATE or REQUEST. The same message will be used for sending updates or sending a route request. The second parameter (speed) will be used to inform other nodes about the current speed of the source node.

4 Performance Evaluation

In this section, we provide a description of the quantitative metrics that will be used to evaluate the performance of the new modifications. We mainly use the following four metrics:

- Packet Delivery Ratio (PDR): It is defined as the ratio of data packets delivered to the destinations to those generated by the Constant Bit Rate (CBR) sources.
- Throughput: It is the total number of bits that are successfully delivered to the destination in a given time period.
- End-to-End Delay: It is defined as the average time delay for data packets to reach from the source node to the destination node. This includes all possible delays that were caused by buffering, interface queuing and data retransmission.
- Normalized Routing Overhead: It is defined as the total number/size of routing packets transmitted per data packets.

We evaluated the extended version based on the Network Simulator tool (NS3) which is extensively used by the research community to evaluate routing protocols and their variants. NS3 is an enhanced version of NS2; It is an open source software and a discrete-event network simulator. It is implemented using C++ and its scripts also can be implemented using C++. It has also better performance and enhanced memory management when compared to NS2. The simulator also includes many tracing models which makes it easy to measure the modified protocol's performance.

5   Simulation Results

We have built three different simulation scenarios to measure the performance of the modified version of the DSDV protocol and compare the results with four routing protocols considered in this study. The first scenario was built to measure the performance in diverse load environments. We repeated every experiment twelve times using different number of nodes. The second scenario was built to measure the performance in diverse velocity environments where every node has a different speed. The scenario was repeated multiple times with different pause time. The last scenario was carried out to compare the performance among the routing protocols based on different speed environments. All the nodes in this scenario have the same speed. The experiment was repeated several times with a new speed value for all nodes in each iteration. The simulation area in the three scenarios is fixed to 750m x 750m. The transmission range of each node is set to 250m. The duration of each scenario is fixed to 100s. Nodes mobility was based on random waypoint model. Table 1 summarizes these parameters along with their values.

Table 1.   Simulation Parameters and their values.

Parameter	Value
Area	750mx750m
Transmission range	250m
Duration	100s
Traffic sources	CBR (512 bytes)
Mobility Model	Random waypoint
Transport protocol	UDP

The results of our study were split into three scenarios as follows:

5.1   Dense vs. Sparse mobile ad hoc networks

In this scenario, the number of nodes varied from 10 to 60 with an increment of 10 nodes per each experiment run. The pause time is initially set to zero and node's speed is fixed at 30 m/s). The rest of the parameters remained unchanged and given values as described in Table 1. The simulation results of this experiment are shown in Fig 7 where Part (a) illustrates the effect of increasing the number of nodes on the Packet Delivery Ratio (PDR); Part (b) shows the effect of increasing the number of nodes on the network Throughput; Part (c) displays the relationship between increasing number of nodes and the average End- to-End delay; And finally, part (d) presents the relationship between increasing the number of nodes and the routing overhead.

Based on Fig 7 we conclude the following findings:

1. The new DSDV version has the same behavior as the other protocols, but it showed a noticeable improvement in terms of PDR

2. The overall performance of the new modification is better in sparse networks compared to dense networks, whereas the difference in packets delivery ratio was about 10% when the number of nodes number was fixed to 30

3. In terms of throughput, the new protocol outperforms the others in all cases, but the results were close to the others when there are very few or very high number of nodes

4. The average End-to-End delay in the new protocol was not improved since the average delay was close to other protocols, but when the number of nodes was increased, the average delay was also increased significantly compared to the others

5. The routing overhead analysis showed that the new protocol outperforms the others except for the protocol E-DSDV which showed a low level of routing overload.

In summary, the performance of all the protocols was better in the sparse networks than in the dense ones. The new protocol also showed a remarkable improvement in all the performance metrics except for the average delay which was close to the other protocols.
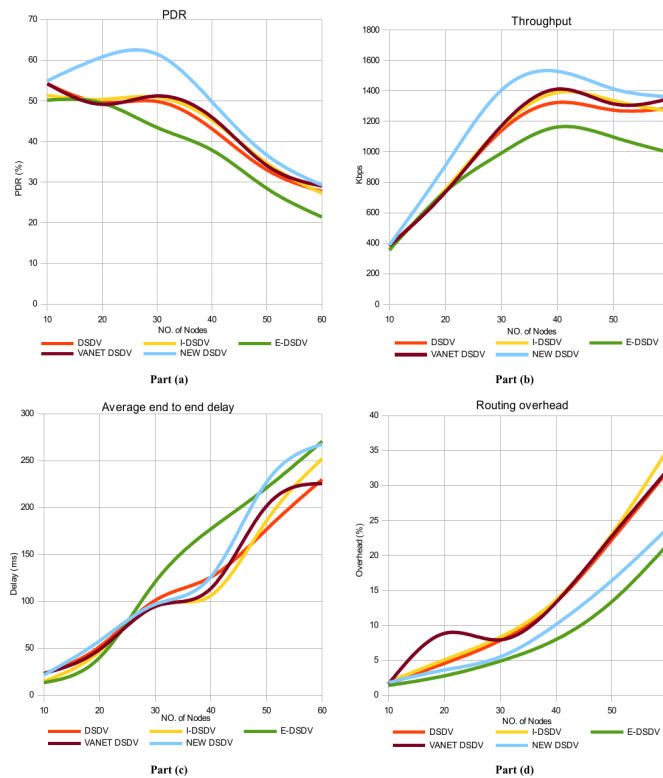

Fig. 7. Performance comparative charts, Part (a) Packet delivery ratio vs. number of nodes, Part (b) Throughput vs. number of nodes, Part(c) End to end delay vs. number of nodes, and Part (d) Routing overhead vs. number of nodes.

5.2   Continuous vs. Discrete mobility mobile ad hoc networks

In this scenario the pause time was varied from 0 to 100 seconds with an increment of 20 seconds by each run. All the nodes in this scenario are moving in a fixed speed (30 m/s). The nodes stop moving at several random locations for the specified pause time. When the pause time is set to zero, the nodes move without stopping. This scenario has a duration of 100s, which means that when the pause time was set to 100s, the nodes will stay in fixed places. The rest of the parameters are unchanged. The results are shown in Fig 8, where Part (a) illustrates the effect of increasing pause time on the (PDR); Part (b) shows the effect of increasing pause time on the network Throughput; Part (c) displays the relationship between increasing pause time and the average End-to-End delay; And finally, part (d) presents the relationship between increasing pause and the routing overhead.
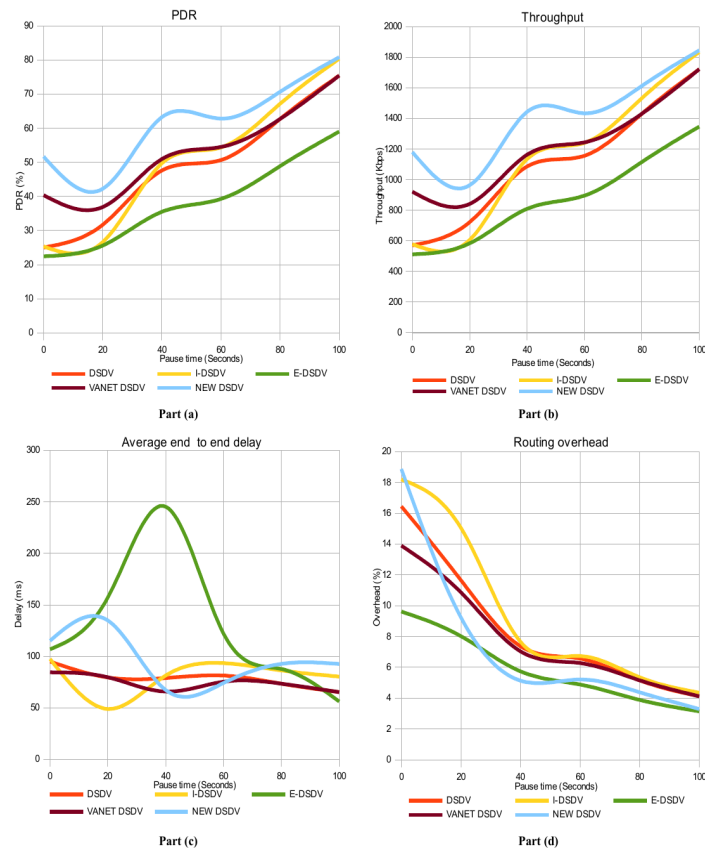

Fig. 8. Performance comparative charts, Part (a) Packet delivery ratio vs. pause time, Part (b) Throughput vs. pause time, Part(c) End to end delay vs. pause time, & Part (d) Routing overhead vs. pause time.

In this scenario, the nodes move in a continuous movement intercepted with random stops. Their duration is shown in the graph. Increased paused time intervals means greater stability of data transmission paths, which improves the performance of the routing protocols. Furthermore, the performance of all the protocols is improved with increasing the pause time's period. The charts also show an increasing in the PDR, especially in the new modified version, where performance outperforms the others in all stages. The same applies to throughput as its value is directly proportional to the pause time interval, but it is noted that the performance of the protocol I-DSDV is approaching the performance of the new protocol with the increase of the pause time interval.

For the average End-to-End delay, it was almost constant regardless of using various paused time intervals for all protocols except for the E-DSDV protocol. For the new modified version, a high average delay was observed when the stopping periods were short. The performance is similar to the other protocols after increasing pause time interval. Routing overload is inversely proportional with increasing pause time, because data the lifetime of transmission paths become longer. Therefore, the routing overhead has decreased for all protocols with increasing the pause time interval. The new protocol showed a considerable improvement with increased pause time and was better than the others.

5.3 Low vs. High Mobility Networks

In this scenario the speed of nodes varied from 5 to 30 with an increment of 5 m/s. The scenario has 30 nodes in all cases. The pause time is zero for all nodes. The rest of the parameters are described in the Table 1. After applying this scenario, we got the results shown in Fig 9. The same analysis is also applied here, but in this case, we study the effect of varying node's velocity on the four measured parameters.

As in the previous scenarios, the new protocol showed a significant improvement in the PDR compared to the others, and this enhancement clearly appears as the node's speed gets higher. It is also noted from Fig 9 that the performance of the protocols in general was less than the new modified version when the movement included different pause time periods. For the delay rate, all protocols showed a noticeable increase, especially in continuous movement environments. The new protocol showed a higher average delay compared to the others. This can be explained by the fact that the new protocol fixes the broken links with other nodes even if they present longer paths, which increases the required data arrival time. For the routing overhead, it increased for all protocols. The new protocol showed better performance than the others except for E-DSDV protocol.
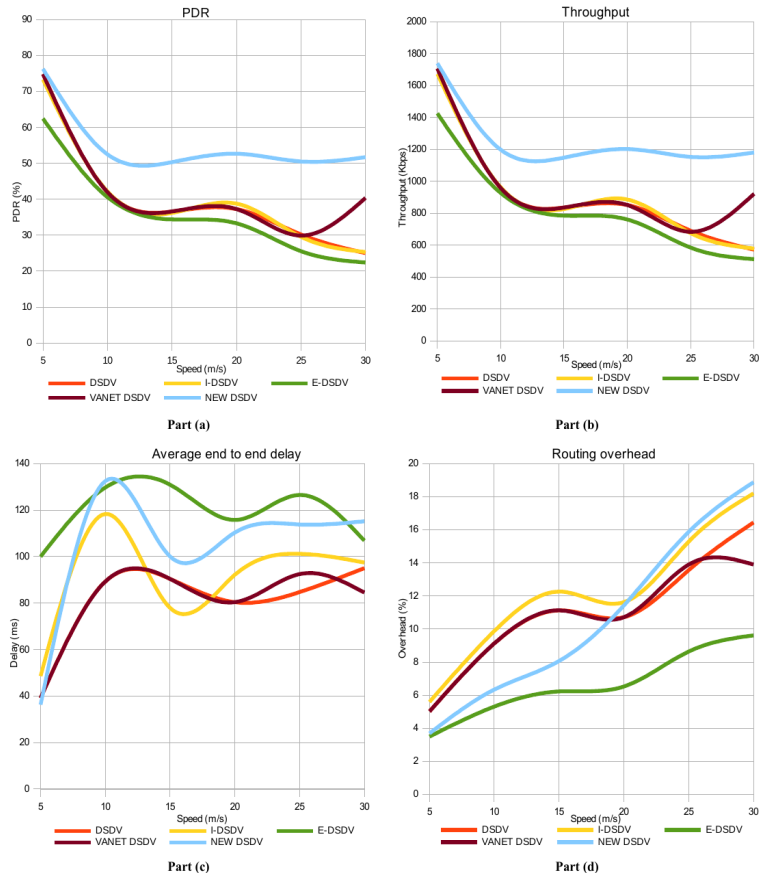
Fig. 9. Performance comparative charts, Part (a) Packet delivery ratio vs. speed, Part (b) Throughput vs. speed, Part(c) End to end delay vs. speed, Part (d) Routing overhead vs. speed.

6   Conclusion and Future Work

ODSDV protocol is one of the most popular routing protocols in Ad hoc networks. It is extensively used by the research community with many variants proposed in the past and still appearing. In this paper, we propose a novel extend version of this protocol by modifying the messages header formats and its internal operations. One of these modifications is to add two new parameters to the message header, namely: node speed and message type; these two parameters had enabled us to fine-tune the measured update and settling time intervals which are considered crucial to the internal operations of this protocol. To study the effect of these modifications, we have implemented several scenarios: using variable number of nodes (Dense vs. Sparse network), using different pause time intervals (Continuous vs. Discrete Mobility Pat-

terns), and finally, testing the protocols using different values of nodes' speeds (Low vs. High Mobility Networks).

The simulation results had shown that the new protocol can provide significant improvements in all scenarios. In terms of packet delivery ratio, the new protocol was the best in all cases. The new protocol outperforms the others regarding the throughput. However, it did not show any improvement in the average End-to-End delay due to the broken link repair mechanisms. Nevertheless, we think this issue is already compensated by the fact that the protocol can fix broken links (even with non-optimal paths) to keep the nodes connected and we believe this is a good compromise. Finally, as all proactive routing protocols considered in this study have demonstrated high routing overhead relative to the actual data that was transmitted, our protocol has actually reduced the overhead compared to these protocols.

As a future work, we intend to expand the proposed protocol by adding the node's coordinates and its direction of movement. Obviously, this will increase the header size and thus increase the overhead of routing updates, but we believe that having this new information at hand will enable each node to take smarter decisions which can impact other performance metrics.

7   Acknowledgement

8   References

[1]   Abd Rahman, A., Ahmad Zukarnain, Z. (2009). Performance Comparison of AODV, DSDV and I-DSDV Routing Protocols in Mobile Ad Hoc Networks. European Journal of Scientific Research. 31(4), 566-576. https://doi.org/10.31142/ijtsrd158

[2]   Kumar, A., Yadav S., Maurya, K. (2013). An Overview of Mobile Ad-Hoc Networks: Architecture, Routing and Challenges. International Journal for Advance Research in Engineering and Technology. l (1), 47-51.

[3]   Maqbool, B., Peer, M. (2010) Classification of Current Routing Protocols for Ad Hoc Networks - A Review. International Journal of Computer Applications, 7(8).

[4]   Perkins, C., Bhagwat, P. (1994). Highly dynamic Destination sequenced distance vector routing (DSDV) for mobile computers. Proceeding of ACM SIGCOMM. 234-244. https://doi.org/10.1145/190809.190336

[5]   Perkins, C., Belding-Royer, E., Das., S. (2003). Ad hoc On-Demand Distance Vector (AODV) Routing. IETF. RFC 3561. https://doi.org/10.17487/rfc3561

[6]   Bhatia, D., Sharma, D. (2016). A Comparative Analysis of Proactive, Reactive and Hybrid Routing Protocols over open Source Network Simulator in Mobile Ad Hoc Network. International Journal of Applied Engineering Research. 11(6), 3885-3896.

[7]   Johnson, D., Maltz, D. (1996). Dynamic Source Routing in Ad Hoc Wireless Networks. Mobile Computing Journal. 5, 153–181. https://doi.org/10.1007/978-0-585-29603-6_5

[8] Ahmeda, G., Barskarb, R., Barskarc, N. (2012). An Improved DSDV Routing Protocol for Wireless Ad Hoc Networks. International Conference on Communication, Computing & Security (ICCCS-2012). 6, 822 – 831. https://doi.org/10.1016/j.protcy.2012.10.100

[9] Agarkhed, J. (2017). A Survey on DSDV Routing Protocol in Ad hoc Network. International Journal of Emerging Trends & Technology in Computer Science. 6(5), 114-117.

[10] Loo, J., Mauri, J., Ortiz, J. (2012) Mobile Ad Hoc Networks Current Status and Future Trends. CRC Press. https://doi.org/10.1201/b11447

[11] Khan, K., Reddy, A., Zaman, R. (2008). An Efficient DSDV Routing Protocol for Wireless Mobile Ad Hoc Networks and its Performance Comparison. IEEE. 506-511. https://doi.org/10.1109/ems.2008.11

[12] Khan, K., Zaman, R., and Reddy, A. (2009) The Performance of the Extended DSDV (eDSDV) MANET Routing Protocol and its Suitability in Integrated Internet-MANET. International Journal of Simulation Systems, Science & Technology. 10(2), 38-46.

[13] Gorantala, K. (2006). Routing Protocols in Mobile Ad-hoc Networks. MSc. Dissertation, Dept. Of Computer Science, Umeå University, SE-901 87 UMEÅ SWEDEN.

[14] Naseem, M., Kumar, C. (2013). EDSDV: Efficient DSDV Routing Protocol for MANET. International Conference on Computational Intelligence and Computing Research, IEEE. 1-4. https://doi.org/10.1109/iccic.2013.6724265

[15] Elias, M., Campista, M., Rubinstein G. (2014). Advanced Routing Protocols for Wireless Networks. ISTE Ltd and John Wiley. https://doi.org/10.1002/9781118984949.ch6

[16] Gulati, M., Kumar, K. (2014). Performance Comparison of Mobile Ad hoc Network Routing Protocols. International Journal of Computer Networks & Communications. 6(2). https://doi.org/10.5121/ijcnc.2014.6211

[17] Jacquet, P., Muhlethaler, P., Clausen, T., Laouiti, A., Qayyum, A., Viennot, L. (2001). Optimized link state routing protocol for ad hoc networks. in Proceedings of the 5th IEEE Multi Topic Conference, INMIC 200I. 62-68. https://doi.org/10.1109/inmic.2001.995315

[18] Rehman, S., Khan, M., Zia, T., Zheng, L. (2013). Vehicular Ad-Hoc Networks (VANETs) - An Overview and Challenges. Journal of Wireless Networking and Communications. 3, 29-38.

[19] Murthy, S., and Garcia-Luna-Aceves, J. (1996). An Efficient Routing Protocol for Wireless Networks. ACM Mobile Networks and App., Special Issue on Routing in Mobile Communication Networks. 183–97. https://doi.org/10.1007/bf01193336

[20] Shelja, S., Suresh, K., K., R. (2011). Study of Route Reconstruction Mechanism in DSDV Based Routing Protocols. International Journal of Computer Engineering Science. 1(2), 34-46.

[21] Ahn, S., Shankar, A. (2001). Adapting to route-demand and mobility in ad hoc network routing. Proceedings Ninth International Conference on Network Protocols, IEEE. 44 – 52. https://doi.org/10.1109/icnp.2001.992759

[22] Liu, T., Liu, K. (2007). Improvements on DSDV in Mobile Ad Hoc Networks. International Conference on Wireless Communications, Networking and Mobile Computing, IEEE. 1637 – 1640. https://doi.org/10.1109/wicom.2007.412

[23] Larsson, T., Hedman, N. (1998). Routing Protocols in Wireless Ad-hoc Networks - A Simulation Study. MSc. Dissertation, Dept. Of Computer Science, Luleå University of Technology, Stockholm, Sweden.

[24] Prasan, U., Murugappan, S. (2016). An Analysis on Vehicular Ad-Hoc Networks: Research Issues, Challenges and Applications. International Journal of Applied Engineering Research. 11(6), 4569-4575.

[25] Lee, U., Midkiff, S. Park, J. (2005). A proactive routing protocol for multi-channel wireless ad-hoc networks (DSDV-MC). International Conference on Information Technology: Coding and Computing (ITCC'05). 2,710 – 715. https://doi.org/10.1109/itcc.2005.26

[26] Park, V., Corson, M. (1997). A highly adaptive distributed routing algorithm for mobile wireless networks INFOCOM '97. Sixteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Driving the Information Revolution, Proceedings IEEE. 3, 405 – 1413. https://doi.org/10.1109/infcom.1997.631180

[27] Kai, W., Yang-yang, G., Jian-feng, G., Ya-juan, Q. (2011). MDSDV: a modified DSDV routing mechanism for wireless mesh networks. The Journal of China Universities of Posts and Telecommunications, Elsevier. 18(2), 34-39. https://doi.org/10.1016/s1005-8885(10)60142-2

[28] Yang, X., Sun, Z., Miao, Y., Kang, N. Yang, Y. (2015). Performance Optimization for DSDV in VANETs. 17th UKSim-AMSS International Conference on Modelling and Simulation (UKSim), IEEE. 514-519. https://doi.org/10.1109/uksim.2015.29

[29] Haas, Z. (1997). A new routing protocol for the reconfigurable wireless networks. Proceedings of ICUPC 97 - 6th International Conference on Universal Personal Communications, San Diego, CA. 2, 562 – 566. https://doi.org/10.1109/icupc.1997.627227

9   Authors

Mada' Abdel Jawad is a vocational teacher in the Computer Workshop at Salfit Industrial School in Palestine. He is currently a Master student in Computer Science at Al-Quds university in Jerusalem. He is working on his Master's Thesis on Routing Protocols in MANETs. He received his B.Sc. in Computer Engineering at Philadelphia University in Jordan. He is also interested in Open Source Software development and Operating Systems.

Saeed Salah is an Assistant Professor at the Department of Computer Science at Al-Quds University in Jerusalem. He received his B.Sc. in Electrical Engineering from Al-Najah National University in Palestine in 2003, his M.Sc. degree in Computer Science from Al-Quds University in 2009, and his Ph.D. from the Department of Signal Theory, Telematics and Communications of the University of Granada, Spain. His research interests are mainly focused on Network Management, Routing Protocols, Information and Network Security, Machine Learning, knowledge-based Systems, and Blockchain. Dr. Salah published many peer-reviewed research papers in recognized international Journals and Conferences. Moreover, he acts as a reviewer for a number of international journals and conferences.

Raid Zaghal is an Assistant Professor at the Department of Computer Science at Al-Quds University in Jerusalem. Dr. Zaghal received his Ph.D. in Computer Science from Kent State University in Ohio, USA in 2005, and his Master in Computer Science from the American University in Washington DC in 1996. Currently he is an Assistant Professor, lecturer and researcher at the Computer Science Dept at Al-Quds

University (since 1996). His research interests are in Network Theories and Protocols' Design, Routing Protocols, MANETs, Cloud-Computing, and Mobile Applications. In the past 13years, he has advised more than 20 Master students on a number of research projects in these fields and published many articles in International Conferences and Journals.