ASDSR: An Adaptive Stable DSR Routing Protocol for Mobile Ad-hoc Networks

https://doi.org/10.3991/ijim.v14i08.11692

Hutaf S. Natoureah (✉)
Hashemite University, Zarqa, Jordan
hutaf@hu.edu.jo

Huda Karajeh
The University of Jordan, Amman, Jordan

Alaa' Abu Serhan
Hashemite University, Zarqa, Jordan

Abstract—The process of finding a route between the transmitter and the receiver node in the Mobile Adhoc Networks (MANets) is a renewed issue that is becoming more and more interesting to the researchers as this type of networks grow and expand. The dynamic nature of MANET and the limited capabilities of wireless nodes in terms of memory size and battery charge are the most important obstacles to the routing (path-finding) process between nodes. In this research, we introduced a new protocol based on the well-known DSR protocol. The proposed routing protocol adds a mechanism to control the RREQ Flooding process; this aims to reach more stable routes while reducing the overhead of routing process caused by link breakage between nodes, and reduces the overhead of network flooding with RREQ messages with each attempt to find a path. In this proposed mechanism, RREQ messages are sent to subset of the devices that are adjacent to the transmitter node, this selection depends on a stability evaluation criterion calculated based on three weighted factors: the speed of the node, the out-degree value (the number of adjacent nodes), and the number of tracks stored in the nodes' memory. The proportion of devices selected is automatically changed adaptively to ensure the achievement of the expected throughput for this network. The proposed protocol was tested using simulation where results showed that ASDSR proved an enhancement in route stability about (0.13), and a decrease in the number of deleted routes by (9%), while maintaining the expected packet delivery ratio of the original DSR by about (0.86).

Keywords—MANET, Routing, Route stability, DSR

1 Introduction

Mobile Ad hoc network, a term considering networks in which nodes are free to move from location to another while there is no infrastructure to connect and manage transaction between them [16]. Due to its lack of underlining infrastructure, nodes connect with each other in a decentralized manner, where each node expected to participate in the routing process of transmitting messages from source node to the destination node and they are assumed to work collaboratively while routing messages in order to accomplish an accepted level of service [3].

Finding an efficient route between two requires a routing process that depends on a specific routing algorithm. The routing algorithms is a set of rules that are responsible for finding the more appropriate route between two nodes in a network according to specific efficiency metric. Different proactive and reactive routing algorithms are introduced in the literature, where routes between two hosts may consist of many hops through other hosts in the network. Proactive protocols [7, 13, 18, 22] suggests storing routing table in each node's memory resulting in minimum transmission time while sending packets since the route already known, but it consumes mobiles storage capacity. On the other hand, Reactive protocols [12] [27] find routes on demand; if the node wants to send packet, it will search for the route and store it in its memory. Reactive protocols give better storage utilization while there is a delay in the process of searching for suitable route.

Due to the nature of MANET, link breakage between nodes occurs when two adjacent nodes go far from the range of each other because of nodes mobility and frequent changes in their locations. This is considered one of the most expected problems in mobile ad hoc networks [25]. Hence, a stable, efficient, and less overhead routing protocol is needed to facilitate finding stable routes from source to destination node.

1.1 Dynamic source routing protocol (DSR)

DSR is one of the reactive routing protocols in MANET, DSR relies on searching for the route once it is required [27]. This process involves two stages: route discovery; which includes flooding of a packet called Route Request (RREQ) message that passes through the network looking for a route to the destination node. The second stage is route maintenance, which begins when the communication between two nodes is interrupted, and then a Route Error (RERR) packet is sent by nodes surrounding the failed link to inform other nodes about this change. Many challenges lie in the process of routing in MANET, such as, the limited nodes storage and battery, the burden of overflow of messages during the process of establishing the route, and the continues movement of network nodes causes frequent interconnection of nodes. In DSR, if node ni want to send a packet to node nj the method is illustrated as follows:

1. Node ni will search for a route in its memory which leads to node nj. If there is a route, the message will be sent directly through the saved route, otherwise, step2 will be performed.
2. ni will try to find the path that fulfill its demand. The process of finding route starts by generating a special message called Route Request (RREQ), this message contains information about the source ni and destination nj of the transmission and

additional information about the intermediary nodes passed-through by this RREQ in order to keep track of the found path.

3. The RREQ message will be broadcasted to all surrounding nodes. If node ng received the RREQ, and it is not the desired destination, it will re-broadcast the RREQ again to its neighbors. The process continued until RREQ received by the destination node nj
4. If nj received the RREQ, then nj will generate another message called Route Reply (RREP) message. RREP will contain the same information about ni and nj from RREQ just received.
5. RREP will be sent back through the found route from one node to another until reaching the sender node ni.
6. The initiator node of RREQ and any node within the route will save the new discovered route to nj in its cache memory.

Link breakage between nodes occurs when two adjacent nodes go far from the range of each other because of nodes mobility and frequent changes in their locations. This is considered one of the most expected problems in mobile ad hoc networks [25]. In DSR, if link breakage occurs, each node realized the failure will generate a Route Error (RERR) message and broadcast it over all nodes to delete any route depends on this failed link.

Fig1. shows an example of using DSR in the process of broadcasting RREQ message starting from the sender node (N1) searching for a route to a destination node (N21). All neighbors of (N1) will receive the RREQ message (N2,N3,N4,N5,N6), unfortunately, none of them is the desired recipient, as a result, each of the neighbors have to participate in the routing protocol and re-broadcast the message to its neighbors. If a node received the same RREQ another time it will not re-broadcast it. This process continues until reaching (N21). You many notice that there are many paths guide to the destination node (N21). The first RREQ received by (N21) will be considered as the desired route and then node (N21) in its role, will generate Route Reply Message (RREP) and send it back through the found path until reaching the sender N1. During this process, each node inside this route will save the route to (N21) in its memory. Here, (N1) will save the route (N3, N10, N22, N21), (N3) will save a route to node (N21) too, which is (N10, N22, N21), and (N10) will also save the route (N22, N21).

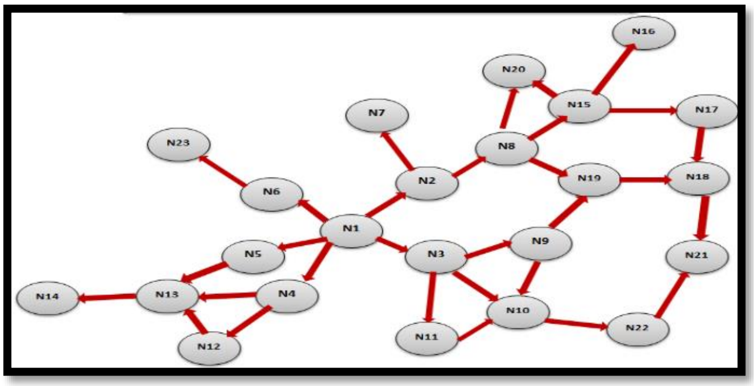

Fig. 1. DSR broadcasting process.

1.2 Related work

In this section we will describe techniques and directions used by other researchers in the area of routing protocols in MANET. Our proposed method is actually based on DSR routing protocol; therefore, we will explore some of the existing DSR enhancement techniques observed from previous researches.

Since the battery limitation is one of the most challenges in Ad-Hoc networks, [19] proposed a route discovery mechanism taking into account nodes' energy consumption in order to increase node life time. When a neighboring node received the RREQ message through a route discovery process, it will add its node ID in the request packet and update its total energy field in the RREQ and then rebroadcast the RREQ message. The simulation result showed that their Modified DSR protocol achieved less energy consumption when compared to the original DSR protocol.

[5] Proposed an enhancement to the route discovery process considering nodes energy consumption for transmission and reception of single packet rather than considering minimum number of hops in the network. When a destination node receives the RREQ message, it will send the RREP message via only nodes having less energy consumption. The authors simulate their ideas and prove its effectiveness regarding the average throughput and energy consumption compared with DSR and other routing protocols. DSR comparisons showed an enhancement about 0.48 achieved in nodes' energy consumption, and about 0.75 enhancements in network throughput for 50 nodes test.

In [9], a multipath energy aware routing for wireless ad hoc network was presented. All possible paths between a source-destination are stored, the destination replies to all route requests (RREQ)s that arrive, and the source stores all the paths of received route reply (RREP)s. The paths are ordered by an energy-based metric. An enhancement about 0.10 in achieved in packet delivery ratio showed by the simulation, while network overhead incremented nearly by 0.01.

[6] Proposed an enhancement on a DSR protocol in order to increase network throughput by increasing the number of maximum retransmission requests size. The authors performed this by setting the value of the maximum retransmission parameter to 32. From simulation, throughput results are analyzed showing that the initial value of throughput is increased by 8.08% of updated DSR and final value is increased by 4.43%. The sample mean value is increased by 5.85%.

In [28] an algorithm was suggested to reduce the flooding of RREQ packets in the network. The basic idea states that while route discovering process, the node receives a RREQ packet will check the following parameters: its own residual battery, received signal strength, and speed. If the defined thresholds for the node for these mentioned parameters satisfy, then the RREQ packet will be forwarded in the network, otherwise, the RREQ packet will be discarded. Evaluation method using Qualnet simulator showed that it results in an increased throughput and average residual battery of the node and decreased the average end to end delay and jitter.

In [8], the authors developed a caching strategy to reduce the flooding overhead by avoiding route discovery as much as possible. In their strategy, the available path is preferred rather than the optimal one; nodes update their cache quickly to minimize end-to-end delay for short-lived traffic by using active packets. The correct information in caches allows to speed up the route discovery and even to avoid it. The packet delivery ratio of the proposed strategy is 99%, which indicates that their proposed approach successfully updated nodes caches and significantly reduced the chance of packet loss due to a route failure.

[2] Proposed a modified DSR termed as Mobile internetwork broadcast infrastructure technique (MIKBIT). In this approach, broadcasting aims to reduce the high overhead involved in flooding while route discovery. Simulation showed an increase in throughput by 130%, and the average end to end delay is less by 90%, jitter is reduced and packet delivery ratio is improved than that of DSR protocol.

In [26] authors applied the Genetic algorithm and Ant Colony swarm intelligence method on DSR protocol to find the optimal path taking into account the misbehaving nodes in the network since the selfish and incapable nodes are some of the causes of network low performance. In their proposed algorithm, any path from the source node to the destination node is a feasible solution; the optimal solution is the shortest one. Another research by [14] also proposed an ant-based hierarchical routing protocol that uses Ant colony optimization algorithm to meet the application-specific Quality of service requirements of heterogeneous traffic generated by the source nodes. Simulation results approved that their protocol outperforms other protocols in terms of energy efficiency, end to end delay, and packet delivery ratio.

[11] Proposed an optimized DSR routing protocol with ant algorithm. Evaluation and Analysis for the proposed approach was done with various scenarios. DSR ant has 48% smaller delay, 1.37 times smaller hop count, and throughput up to 3.6 times larger than the standard DSR, but DSR-ant has routing overhead 58% larger than standard DSR.

[1] Used clustering approach to select only a subset of nodes that need to forward the packet to (RREQ flooding) through route discovery process. This approach aimed to reduce network overhead by eliminating the unnecessary flood. Simulation results

showed that their optimized DSR algorithm enhances the packet delivery ratio by minimum of 20% and reduces the RREQ flooding by a minimum of 30% by integrating passive clustering with DSR, the number of unnecessary flooding of RREQ packet reduced by minimum 30% in mobile ad hoc networks. The number of Route Errors (RERR) was reduced by 17 %. The packet delivery ratio was improved by a minimum of 20% as well.

[15] Proposed an improved DSR protocol which adopts adaptive routing shortening method through introducing time parameters; routing quality is decided by routing survival time. Adaptive automatic routing shortening can produce routes with small amount of hops and guarantee remaining survival time of new route. The performance of this method compared with the DSR protocol using simulation, and enhancement is noticed in the throughput metric and the delivered packets ratio, while there is a bit increase in the delay metric.

[17] Proposed an algorithm for modifying DSR protocol, which is referred to as Modified DSR (MDSR). A memory management algorithm also proposed; in which packets are transmitted with minimum required energy. Results reveal that this algorithm achieved 47% less number of control packets in the networks of low mobility and 4% less number of control packets for the network of highly mobility nodes.

[4] Proposed a QoS-based (Quality of Service based) protocol to improve the performance of the DSR protocol and enhance the reliability of the network. This protocol considered the QoS of the paths in route discovery process in which the path with highest QoS value is selected. The QoS function depends on the factors: the available bandwidth of a node, connectivity level, distance ratio and velocity ratio. Using simulation, nearly 0.40 enhancements was achieved in packet delivery ratio metric.

In our proposed ASDSR, the aim is to find the more stable route while decreasing routing process overhead by introducing an enhancement to the DSR algorithm. During route discovery process, flooding is done by selecting a subset of neighbor nodes to send RREQ message to them rather than broadcasting it to all network nodes. This selection is performed according to a computed stability value depending on three weighted factors: node speed, number of its out-going links, and the number of stored routes in its memory, weight values depends on network requirements. Routes found and stored in nodes' caches will be stable as much time as possible due to taking nodes stability into account during route construction. The remaining of the paper is organized as follows: in the next section a discussion of the proposed ASDSR and its algorithm is presented, then we described the stability function and the evaluation method of our enhancement followed by results discussion and analysis. At the end of the paper a conclusion is presented.

2 The Proposed Adaptive Stable DSR Protocol (ASDSR)

In DSR protocol, when an initiator node starts route discovery stage, it will send the control packet RREQ to each neighbor node. The process of flooding RREQ through the network and rebroadcasting it will generate routing overhead through the network which will affect nodes energy-consumption and the overall performance of the network. Unfortunately, this process will be repeated for each time a new route is needed. Thus, frequent link breakage makes this type of flooding is a main concern.

Our proposed ASDSR protocol aims to find a more stable route while decreasing route discovery overhead on the network. The following steps illustrate how the proposed ASDSR protocol behaves when node n_i wants to send a packet to node n_j that are depicted in figure 2:

Step 1: Checking cache memory: Node ni will search for the route in its cache memory. If a stored route found, the message will be sent, otherwise, step2 will be performed.

Step 2: Generating RREQ message: Node ni will try to find the path that fulfills its demand. The process of discovering a route starts by generating a special message called Route Request (RREQ). This message contains information about the source ni and destination nj of the transmission and additional information about all intermediary nodes that received and re-transmit the RREQ during the routing process in order to keep track of the found path to the destination node.

Step 3: Flooding RREQ based on stability: Node ni reads the stability value for each neighbor node. Assuming each node in a network calculates the stability value periodically and make available to other nodes. The RREQ message will be send to a subset of neighbor nodes that have greater stability values. If node ng received the RREQ, since ng is not the required destination, it will re-send the RREQ again to a subset of its neighbors according to their stability values. The process continued until RREQ received by the destination node nj.

Step 4: Reach the destination: If nj received the RREQ, then nj will generate another message called Route Reply (RREP) message that contains the same information from RREQ just received.

Step 5: Sending back the RREP: RREP will be sent back through the found route from one node to another until reaching the sender node ni.

Step 6: Saving a rout: The initiator node of RREQ and any node within the route will save the new discovered route to nj in its cache memory.

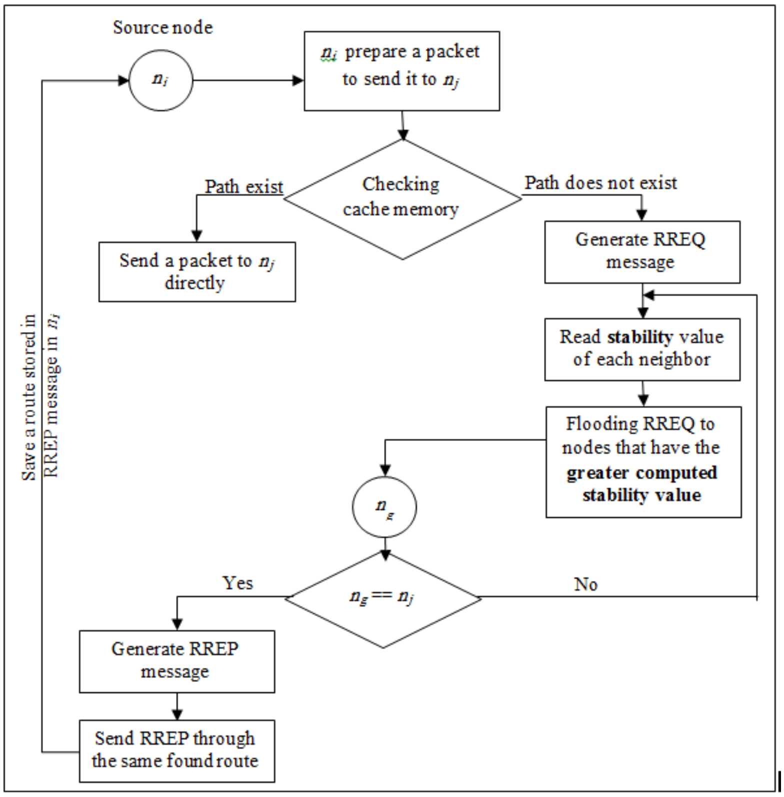
Source node
n_i
n_i prepare a packet to send it to n_j
Checking cache memory
Path exist
Path does not exist
Send a packet to n_j directly
Generate RREQ message
Read **stability** value of each neighbor
Flooding RREQ to nodes that have the **greater computed stability value**
n_g
$n_g == n_j$
Yes
No
Generate RREP message
Send RREP through the same found route
Save a route stored in RREP message in n_i

Fig. 2. ASDSR routing process

Note that sending RREQ to just a set of neighbors will shorten the process and reduce RREQ flooding (*s* in the figure denotes the computed stability value for a node). The selection of these neighbors related to a computed stability value; for example, node (N1) sent the RREQ to node (N2) and node (N6) and node (N4) because they have the largest stability values among all N1 neighbors. The route found may differ from that found by the Original DSR (refer to example in section 1.2). According to Figure 3; the route found is not the shortest path (N2, N8, N19, N18, N21). This path has a length of (5 hubs) compared with (4 hubs) path found before, In this protocol, we can achieve our goal of finding routes that stay for longer duration while decreasing the routing process overhead, the shortest path is not our goal here.

Figure 3show the process of searching for a route from the sender node (N1) to the recipient node (N21).

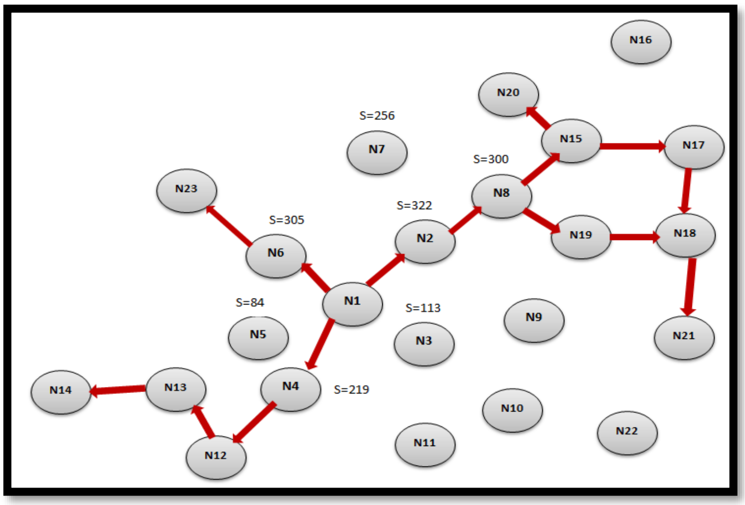

Fig. 3. ASDSR RREQ message flooding based on stability

In the proposed ASDSR a dynamic percent of neighbor nodes will be selected each time to send RREQ message to them; this percent is adaptive with a defined threshold for the accepted average for packet-dropping percent. If the percent of the dropped packets exceeded this threshold, the percent of neighbors that will receive the RREQ message will increase by 0.10 until the average for packet-dropping rate decreased and ensure maintaining an accepted network throughput. To determine this threshold value, we had studied the performance of the original DSR, we noted that the percent of dropped packets wasn't exceed (0.14), therefore, we selected a threshold value which is (0.14) as an accepted value for the average packet-dropping rate, such that the total performance of the network should not exceed this value. Figure 4 shows the changes in the dynamically determined neighbors' percent value during a simulation of 50 time slots.

As noticed from the figure, neighbors selection percent start from (0.50), at time slot (15), the percent start to increase slightly to indicate that the number of dropped packets start exceeds the defined threshold value, it reaches (1) at time slot (20) and returns to decrease at time slot (22), and thus stays increasing and decreasing to ensure an acceptable network throughput.

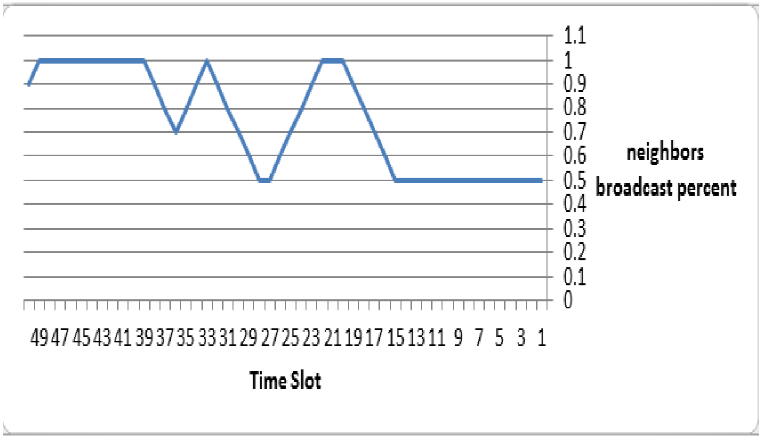

Fig. 4. Neighbors Selection Percent in ASDSR

3 Stability Function

Node stability value is an indicator to how much this node is trusted to establish a long-life route rout passing through it. In our proposed approach we used three main factors as indicators to nodes' stability; node speed, out-degree value, and the number of stored routes.

Node speed, the first factor, is the most critical factor which affects route stability, when node moves in high mobility speed it will not be effective decision to create a route passing through it, since the probability of link breakage will be very high. The second factor is the out-degree value; this factor indicates the number of out-going edges (number of neighbors) for this node. Neighbor node with higher number of out-degree value is more dependable compared with isolated nodes; a node with low-speed and few number of surrounding nodes will not be a good choice for route selection. The third factor is the number of stored routes in nodes' cache memory; the node with more routes stored in its cache memory is considered as an active node and has a strong social network with other nodes, then it will be a good choice for route construction. Stability values will be computed according to Equation (1):

$$S_{ni} = \alpha\,(M - SP_{ni}) + \beta\,(O_{ni}) + \gamma\,(R_{ni}) \tag{1}$$

where S_{ni} is the stability factor value computed for node n_i. M is the maximum speed value defined for nodes movement in the simulator**.** $\mathbf{SP_{ni}}$ is the speed of node n_i. O_{ni} is the number of outgoing links for node n_i. R_{ni} is the number of routes stored in node n_i memory. α is the weight for the node speed factor, β is the weight for the out-degree factor, and γ is the weight for stored routes factor, where the summation of all of these weights equals 1 as shown in Equation (2):

$$\alpha+ \beta+ \gamma=1 \tag{2}$$

The values chosen for the weights α, β, and γ can be determined depending on several network features, such as nodes transition speed (high, low), mobility model, nodes distribution, nodes battery and memory capacities, and many others indicators variables for network state. For the purpose of analysis, we tested our algorithm using several combinations of values of α, β, and γ. Each of these combinations is biased towards one of the three factors as will be mentioned in section 4.

4 The Proposed ASDSR Protocol Modeling

Assume that N represents the set of network nodes $N=\{ni,nj,\ldots.,nk\}$ and L represents the set of links between nodes, for example $L=\{<ni,nj>,<nj,nk>\}$ if there is a direct link between *n*i and *n*j and a direct link between *nj* and *nk*. The direct link indicates that the two nodes are in the transmission range of each other and thus they can communicate directly and send data without requiring any hubs over intermediary nodes. $\forall n \in N$, it has properties such as its position, speed, transmission range, movement direction, and stability value. $\forall \boldsymbol{l} \in L$, ***l*** connects two nodes (ex, *n*i and *n*j) if the distance between *n*i and *n*j less than or equals nodes' transmission range. We call *n*i and *n*j adjacent nodes and the link ***l*** incident to the nodes *n*i and *n*j.

In MANET, sending data between nodes occurs all the time. If node *n*i want to send data to node *n*j, then *ni* is called the initiating (transmitter) node and node *n*j is called the terminating (receiver, recipient) node The sequence of edges (links) connecting intermediary nodes from *ni* until reaching *nj* is called the "path". In a path, the number of included edges represents the "path length" between *n*i and *nj*, the path relation between *n*i and *nj* is defined as: $p(ni,nj)$, if there exist a path from *n*i to *n*j. Minimal path length between *ni* and *nj* is the path which involves the least number of edges, in this research we are not interested with finding the shortest path; instead we are looking for the more stable path.

Links between nodes are transitive, such that, If $p(ni,nj)$ exists, and the distance $d(ni,nj)!=0$, and $p(nj,nk)$ exists and $d(nj,nk)!=0$, this infers that there exist a path between *n*i and *nk* $p(n1,n3)$ where $d(ni,nk) != 0$. If $\forall ni,nj \in N$, $\exists p(ni,nj)$, then G is a connected network. In real mobile networks we cannot ensure fully connectedness between nodes all the time due to nodes mobility behavior.

In our model the broadcasting graph $G^\backslash$ is actually a sub-graph from the actual network graph G, such that $G^\backslash= (N^\backslash, L^\backslash)$ where $N^\backslash \subset N$ and $L^\backslash \subset L$. $\forall \boldsymbol{l} =<ni, nj>$, $\boldsymbol{l} \in L^\backslash$ if and only if the stability value for node *nj* is within the highest values compared with all *ni* neighbors. For a graph G= (N, L), where $N= \{ni,nj,\ldots\ldots nk\}$, we represent the direct connections between nodes by using the n * n adjacency matrix adj whose elements *aij* is given by the definition:

$$aij = \begin{cases} 1: & \text{if } < ni, nj > \in L \\ 0: & \text{otherwise} \end{cases} \qquad (3)$$

4.1 Simulation and analysis

MANET in its nature is continues system, where the state variables that describe the system (such as mobile nodes locations, speeds, links between nodes…) changes during the time, link failure occurs when nodes get outside from the coverage area of another node. In order to study the system as it evolves over time and collect statistical variables that evaluate our proposed ASDSR, simulation experiments were performed.

According to the definitions in section *B*, we built a continuous stochastic simulator for Mobile Ad-hoc Network using Random Walk (RW) mobility model. Modeling starts by dispersing nodes through a predetermined rectangular area with dimension (D), nodes move continuously, each time slot, a node may change its speed or direction according to a pre-defined probability. The new speed for node (i) at time slot (t) is chosen by uniform distribution between speed minimum and maximum values [S_{min}, S_{max}], we use the notation ($SP_{ni}(t)$) to denote speed of node n_i at time slot *t*. The direction is chosen randomly by selecting a random angle (Ø) from the interval (0, 2*π*). During the interval, a node moves with a velocity vector ($SP_{ni}(t)$ cos Ø, SP_{ni} (t) sin Ø). According to this model we built a simulator using (C++) programming language.

We assumed that all nodes wishing to communicate with others, and within the ad hoc network all nodes are willing to participate fully in the protocols of the network. In particular, each node participating in the network should also welcome to forward packets for other nodes in the network. To represent our model by a simulator we assumed that (k) is the number of nodes in the network, where (*n1*, *n2*, *n3*, …., k) are the nodes. The new location for a moving node at time slot (t+1) is computed according to the following equation:

$$n_{ix}(t+1)= n_{ix}(t)+SP_{ni}(t)*\cos Ø_{ni}(t) \quad (4)$$

$$n_{iy}(t+1)=n_{iy}(t)+SP_{ni}(t)*\sin Ø_{ni}(t) \quad (5)$$

where, $n_{ix}(t)$: x location of node (i) at time slot (t), $n_{iy}(t)$: y location of node (i) at time slot (t), $SP_{ni}(t)$: speed of node (i) at time slot (t), $Ø_{ni}(t)$: angle of movement for node (i) at time slot (t)

In order to determine if two nodes are adjacent to each other or not, we should determine if the distance between them not go beyond their transmission range, this is done by computing the distance between these two nodes (assume, n_i and n_j) according to the following equation:

$$d_{ij} = \sqrt[2]{\left(n_{jx} - n_{ix}\right)^2 - \left(n_{jy} - n_{iy}\right)^2} \quad (6)$$

When a route deleted from nodes' cache memory due to link breakage, we can compute the average routes stability according to Eq.7:

$$AvgS = \sum_{i=1}^{NR} (\mathrm{RR(i)} - \mathrm{RS(i)})/\mathrm{NR} \quad (7)$$

where: NR is number of routes created during simulation time. RS is the time slot at which the route is created, and RR is the time slot at which the route is removed from the cache because of link failure.

Simulation Parameters values are shown in Table 1; The number of nodes used in the simulation scenarios is 100 nodes scattered uniformly in a rectangular area of dimension Ten meters. Nodes have an equal radio propagation range equal two meters. At each time slot, 10 sessions started between nodes to exchange messages. Nodes move in speed between minimum and maximum values of 0.20 m and 1.50 m/time slot respectively. The simulation runs for fifty time slot every try. Results analyzed from collecting statistical information of a sample of 10 runs.

Table 1. Simulation Parameters

Simulation Parameters Values

Parameter	***Value***
Simulation Time	50 time slot
Area size	10 x 10 m^2
Radio Propagation	2 meters
Minimum speed	0.20 m/time slot
Max speed	1.5 m/time slot
Number of Nodes	100
Number of Sessions each time slot	10

5 Experimental Results Analysis

In this section we will discuss the results of applying the proposed ASDSR protocol compared with the performance of the original DSR algorithm. The proposed ASDSR protocol use three weighted factors as indicators of node stability; node speed, out-degree value, and the number of stored routes. Based on the computed stability value, the ASDSR protocol choses a set of neighbor nodes that will be included the flooding RREQ process. Therefore, we will use four different combinations of the weights (α, β, γ) to these factors to use them in computing the stability values as mentioned in Eq. 1. These combinations were used in the stage of evaluating the performance of the proposed ASDSR protocol as a test values:

1. (OF) Out-degree Factor [20 60 20]: the values for the weights (α, β, γ) are chosen so that the out-degree of neighbor node will have the highest weight; assuming that the node with more neighbors is more likely to result in a stable path passing through it, denoted by Out-degree factor (OF).
2. (SF) Speed Factor [60 20 20]: in this combination means that the major dominant factor in computing nodes' stability value is the node speed; neighbor nodes of lower speed are more trusted to broadcast RREQ message through them, denoted by Speed Factor (SF).
3. (SRF) Stored Rout Factor [20 20 60]: in this scenario, stability value is biased towards node that has more stored routes in its memory since it is the highly active

node and thus it is a trusted node to get stable routes passing through this node, denoted by stored routes factor (SRF)

4. (EDW) Equally distributed Weights [30 40 30]: by giving each factor a nearly equal weight without biasing towards any of them, assuming that all factors are important and affects the process of selecting a stable route.

Simulation of the proposed ASDSR algorithm has been performed on a mobile simulated environment, where nodes move all the time according to Random Walk (RW) mobility model. Simulation results are analyzed and compared with the original DSR protocol in three main dimensions: average route stability, number of deleted routes, and packet delivery ratio.

5.1 Routes stability

We define route stability value as the period of time between two time slots; the time slot at which the route was created and the time slot when it was deleted. Within simulation of fifty time slots, routes stayed for average (14.1) time slots in the original DSR, While they were exist for average (16) time slots in the ASDSR for all weights, an enhancement about (0.125) time slots was achieved by ASDSR. Fig.5 shows the average route stability over 10 runs. Because of the unified values for network state variables, the different combinations of weights resulted in a close stability values, the EDW slightly outperforms the other combinations of weights.

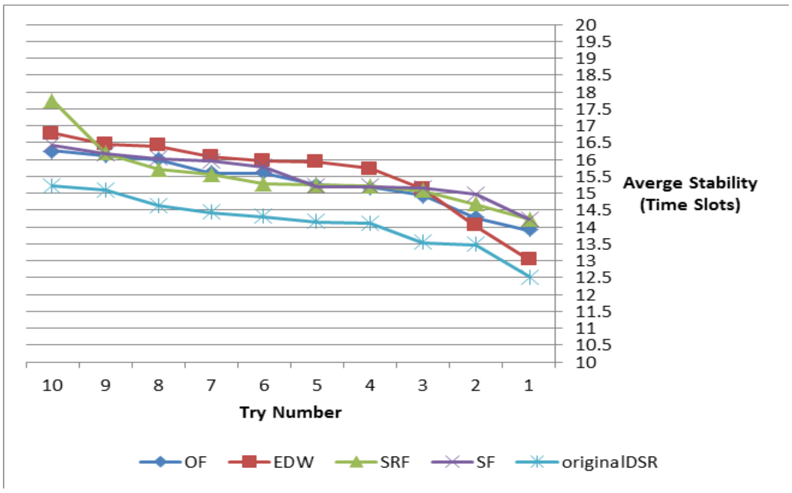

Fig. 5. Average Route Stability Metric

5.2 Number of deleted routes

The results of stability are supported with gaining less overhead caused by the routing algorithm, this could be noticed from the number of the deleted routes, where the number of the deleted routes were decreased by nearly by 1400 routes in our ASDSR protocol as shown in Fig.6. This enhancement gained although the network environment and mobility model applied including the average link breakage was the same in all simulation experiments and node mobility was random and uniformly distributed.

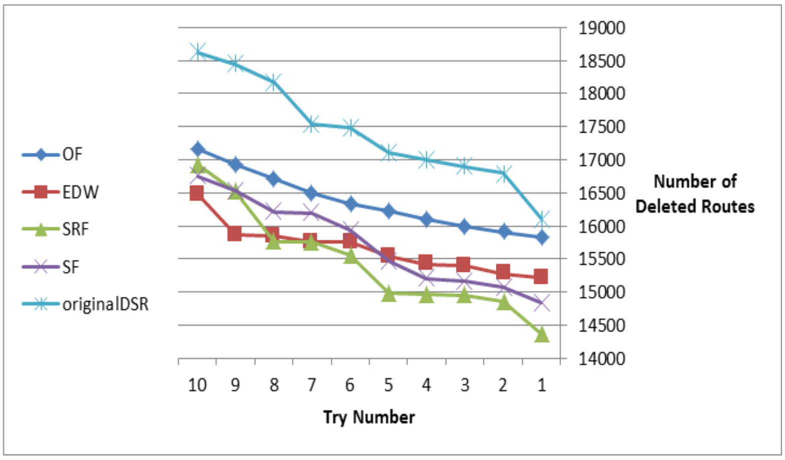

Fig. 6. The Number of Deleted Routes

From a deeper look to the curve, we notice that the number of stored routes in nodes' memory had a significant role in decreasing the number of deleted routes; therefore, establishing routes passing through nodes that have more routes stored in their memories will decrease the routing process overhead since the number of deleted routes will be minimized and there will be no need to search for a new route every time.

5.3 Packet delivery ratio

Packet delivery ratio could be defined as the number of packets successfully sent during a specified period of time relative to all sending attempts. Results are very close between the original DSR and our proposed ASDSR, this indicates that we can achieve the same throughput while decreasing network overhead and getting more stable routes using our proposed enhancement. These results are shown in Fig.7 below. The closest value was achieved by OF; outgoing links factor, where the number of neighbors for a node give it higher priority to be selected.

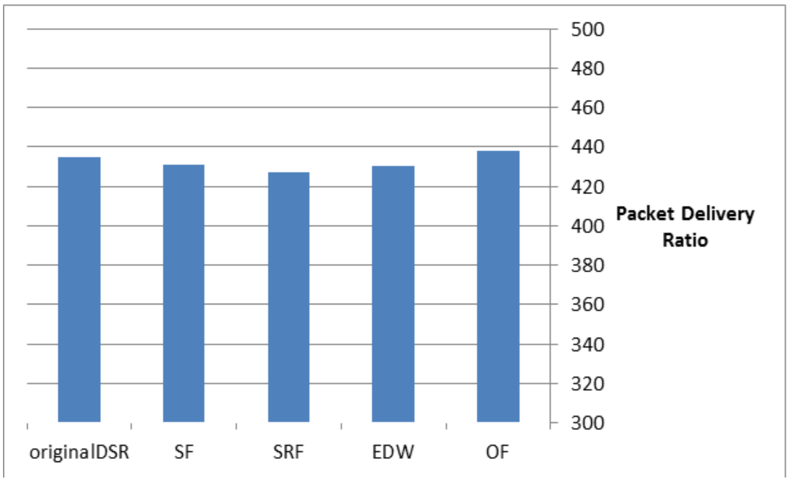

Fig. 7. Packet Delivery Ratio Metric

6 Conclusion

In this paper, we proposed ASDSR protocol as an enhancement on the DSR protocol, where the most stable route is found by restricting flooding Route Request Message (RREQ) to a subset of each node's neighbors according to a computed stability value based on three weighted factors; node speed, out-going degree, and the number of stored routes in nodes' cache memory. A dynamic and adaptive process was applied for selecting the percent of the neighbor nodes based on monitoring the packet delivery ratio and adjust the percent of neighbors included in the RREQ flooding process in a way that achieve an accepted packet delivery ratio. Our proposed ASDSR was evaluated using simulation; results were collected and compared with the original DSR protocol showed an enhancement in the overall routes stability about (0.13), and a decrease in the number of deleted routes by (9%), while maintaining an accepted value for the packet delivery ratio (0.86).

In the next study, we will make an exploratory study about network features that affect the selection process of weight values for each of the three factors and what are the suggested values for these weights at various scenarios.

7 References

[1] A. Barve, A. Kini, O. Ekbote and J. Abraham, (2016), "Optimization of DSR routing protocol in MANET using passive clustering", 2nd International Conference on Communication Control and Intelligent Systems (CCIS), Mathura, pp. 23-27. https://doi.org/10.1109/ccintels.2016.7878193

[2] A. Gupta, R. Upadhyay and U. R. Bhatt. (2014),"MIKBIT- Modified DSR for MANET", International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, pp. 268-271, 2014. https://doi.org/10.1109/icicict.2014.6781291

[3] Cheng R. H., Wu T.-K., Yu C. W. (2010), "A highly topology adaptable ad hoc routing protocol with complementary preemptive link breaking avoidance and path shortening mechanisms". Wireless Networks 16(5): 1289–1311, 2010. https://doi.org/10.1007/s11276-009-0203-0

[4] D. Al-Terri, H. Otrok, H. Barada, M. Al-Qutayri, R. M. Shubair and Y. Al-Hammadi. (2015), "Q-DSR protocol in vehicular ad-hoc networks",11th International Conference on Innovations in Information Technology (IIT), Dubai, pp. 162-165, 2015. https://doi.org/10.1109/innovations.2015.7381533

[5] D. Badal and R. S. Kushwah. (2015), "Nodes energy aware modified DSR protocol for energy efficiency in MANET," 2015 Annual IEEE India Conference (INDICON), New Delhi, pp.1-5., 2015. https://doi.org/10.1109/indicon.2015.7443740

[6] D. N. Vadhwani, M. Singh and D. Kulhare. (2013), "Enhanced DSR with optimized throughput using opnet simulator", 2013 Nirma University International Conference on Engineering (NUiCONE), Ahmedabad, pp. 1-4,2013. https://doi.org/10.1109/nuicone.2013.6780100

[7] David B. Johnson and David A. Maltz. (1996), "Dynamic Source Routing in Ad Hoc Wireless Networks", in Mobile Computing, edited by Tomasz Imielinski and Hank Korth, Chapter 5, pp. 153-181, Kluwer Academic Publishers, 1996. https://doi.org/10.1007/978-0-585-29603-6_5

[8] Dimitri Marandin. (2007), "Improvement of Link Cache Performance in Dynamic Source Routing (DSR) Protocol by Using Active Packets" Next Generation Teletraffic and Wired Wireless Advanced Networking, Springer, vol. 4712, pp. 367–378, 2007 https://doi.org/10.1007/978-3-540-74833-5_31

[9] F. De Rango, P. Lonetti and S. Marano. (2008), "Energy-aware metrics impact on Multipath DSR in MANETs environment", International Symposium on Performance Evaluation of Computer and Telecommunication Systems, Edinburgh, pp. 130-137, 2008.https://doi.org/10.1007/978-0-387-09490-8_19

[10] H. Yang and Z. Li. (2017), "A routing optimization technology based on neural networks in MANET", 8th International Conference on Computing, Communication and Networking Technologies (ICCCNT), Delhi, pp. 1-6, 2017. https://doi.org/10.1109/icccnt.2017.8203920

[11] Istikmal. (2013), "Analysis and evaluation optimization dynamic source routing (DSR) protocol in Mobile Adhoc network based on an algorithm", International Conference of Information and Communication Technology (ICoICT), Bandung, pp. 400-404, 2013. https://doi.org/10.1109/icoict.2013.6574609

[12] Jahangir Khan, Korangi Creek. (2010), “MANET Reactive Protocols-Tutorial Review”, International Journal of Computer Applications (0975 – 8887) Volume 12– No.4, December 2010. https://doi.org/10.5120/1669-2250

[13] Kapang Lego. (2011), "Comparative study of Ad-hoc routing protocol", Indian journal of Computer Science and Engineering; vol. I, No. 4 364-371 28, July 2011.

[14] K. Dilip, Kaur. Tarunpreet. (2017), "AntMQoS: An Ant-based Multi-Constrained QoS Routing Protocol for Wireless Sensor Networks", Proceedings of the World Congress on Engineering and Computer Science, San Francisco, USA, Vol I WCECS 2017, October 25-27, 2017.

[15] L. Pan. (2014), "An optimization strategy for DSR protocol", The 7th IEEE/International Conference on Advanced Infocomm Technology, Fuzhou, pp. 143-147, 2014.https://doi.org/10.1109/icait.2014.7019545

[16] Long, Z., He, Z. (2007), "Optimization and implementation of DSR route protocol based on ad hoc network", International Conference on Wireless Communications, Networking

and Mobile Computing,IEEE, pp. 1508–1511 (2007). https://doi.org/10.1109/wicom.2007.380

[17] M. Tamilarasi, S. Chandramathi and T. G. Palanivelu. (2008), "Overhead reduction and energy management in DSR for MANETs", 3rd International Conference on Communication Systems Software and Middleware and Workshops (COMSWARE '08), Bangalore, pp. 762-766, 2008. https://doi.org/10.1109/comswa.2008.4554513

[18] Muhammad Saleem. (2010), "Research article on performance modeling of Ad-hoc Routing Protocol," EURASIP Journal on wireless communication and networking; vol.l Article lD 373759, 10, february,2010.

[19] N. M. Upadhyay, K. Gaurav and A. Kumar. (2014), "Modified DSR, an energy conserving approach to DSR protocol in MANET," 2014 International Conference on Communication and Signal Processing, Melmaruvathur, pp. 1146-1149, 2014https://doi.org/10.1109/iccsp.2014.6950032

[20] N. Meghanathan. (2009), "A beaconless node velocity-based stable path routing protocol for mobile ad hoc networks", IEEE Sarnoff Symposium, Princeton, NJ, pp. 1-5, 2009. https://doi.org/10.1109/sarnof.2009.4850273

[21] N. Wang and J. Chen. (2006), "A Stable On-Demand Routing Protocol for Mobile Ad Hoc Networks with Weight-Based Strategy", Seventh International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), Taipei, pp. 166-169, 2006. https://doi.org/10.1109/pdcat.2006.20

[22] O.H. Hussien, T.N. Saadawi, M.l. Lee. (2005), "Probability Routing Algorithm for Mobile Ad Hoc Networks' Resources Management", IEEE Journal on Selected Areas in Communications, vol. 23, 12 Dec, 2005. https://doi.org/10.1109/jsac.2005.857205

[23] Pramita Mitra, Christian Poellabauer, Shivajit Mohapatra. (2007), "Stability aware routing: Exploiting transient route availability in manets", HPCC, pp. 508-520, 2007. https://doi.org/10.1007/978-3-540-75444-2_49

[24] S. Shukla, S. Sharma and S. Kumar. (2013), "Sequence-number aided timer based DSR protocol in MANET: TDSR", 4th International Conference on Computer and Communication Technology (ICCCT), Allahabad, pp. 151-156, 2013. https://doi.org/10.1109/iccct.2013.6749619

[25] T. G. Griffin, F. B. Bruce, and G. Wilfong. (2002), "The Stable Paths Problem and Interdomain Routing," IEEE/ACM Transactions on Networking, Vol. 10, No. 2, pp. 232-243, April 2002. https://doi.org/10.1109/90.993304

[26] T. Varshney, A. Katiyar and P. Sharma. (2014), "Performance improvement of MANET under DSR protocol using swarm optimization", International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, pp. 58-63, 2014. https://doi.org/10.1109/icicict.2014.6781253

[27] Tony Larsson and Nicklas Hedman, (2010), "Routing Protocols in Wireless Ad-hoc Networks" International conference on Information & Network Technology- A Simulation Study Stockholm,2010.

[28] U. R. Bhatt, N. Nema and R. Upadhyay. (2014), "Enhanced DSR: An efficient routing protocol for MANET", 2014 International Conference on Issues and Challenges in Intelligent Computing Techniques (ICICT), Ghaziabad, pp. 215-219, 2014. https://doi.org/10.1109/icicict.2014.6781282

8 Authors

Hutaf S. Natoureah is a lecturer in the Hashemite University of Jordan specialized in computer science, exactly in computer Ad hoc wireless networks and Artificial Intelligence algorithms Email: hutaf@hu.edu.jo

Huda Karajeh is an Assistant Professor of Computer Information Systems; The University of JordanVerified email at ju.edu.jo. Email: h.karajeh@ju.edu.jo

Alaa' Abu Serhanworks at Hashemite University of Jordan Email: alaaa@hu.edu.jo

Article submitted 2019-09-17. Resubmitted 2020-01-31. Final acceptance 2020-01-31. Final version published as submitted by the authors.