Engineering and Technology Journal e-ISSN: 2456-3358

Volume 07 Issue 01 January-2022, Page No.-1139-1148

DOI: 10.47191/etj/v7i1.05, I.F. – 6.39

© 2022, ETJ

Developing Secure Multipath Routing (AODV-SMR) Protocol Framework Based Multipath Network Routing Protocol

Marwah Yahya Albeladi[1], Omaima Bamasag[2], Ghadah Aldabbagh[3]

[1]Department of Computer and Information Science, Applied Collage, Taibahu University, Al-Madinah Al-Munawara 20012, Saudi Arabia

[2]Department of Computer Science Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

[3]Department of Computer Science Faculty of Computing and Information Technology, King Abdulaziz University, Jeddah, Saudi Arabia

ABSTRACT: Mobile ad hoc networks (MANETs) are wireless networks, which consist of mobile nodes that communicate over wireless media. MANETs are typically characterized by high mobility and frequent link failures. Since routing protocols are crucial for the success of MANETs, multipath routing can be employed to reduce link failure so that alternate paths will be available. This paper aims to develop and evaluate the performance of a new multipath routing protocol based on the ad hoc on-demand distance vector (AODV) routing protocol. It is an on-demand multipath routing protocol named ad hoc on-demand distance vector-secure multipath routing (AODV-SMR). The on-demand multipath routing protocol was evaluated against three performance metrics: packet delivery ratio, average end-to-end delay and normalized routing load. The simulation results, obtained using an OMNET++ simulator, showed that the proposed protocol exhibits significant improvement in the packet delivery ratio. It also outperformed the ad hoc on-demand multipath distance vector (AOMDV) protocol in the average end-to-end delay, and outperformed AODV in the normalized routing load metric.

KEYWORDS: multipath routing protocol; AODV; AOMDV; MANET; Packet Delivery Ratio; Average End-to-End Delay.

I. INTRODUCTION

Mobile ad hoc networks (MANETs) have many useful applications such as in military battlefields, conferences, personal area networking, vehicular ad hoc networks, emergency disaster relief, research, and education in remote areas with poor infrastructure[1-3]. A MANET is a dynamic network consisting of a collection of wireless mobile nodes that communicate with each other without the use of any centralized authority [4]. These nodes, which are autonomous and self-organized, each possess a wireless interface to communicate with other nodes [5, 6]. A MANET is more vulnerable than a wired network due to its ad-hoc nature. Some inherent features of MANET, as listed in [7, 8], increase its vulnerabilities.

The ad hoc on-demand distance vector (AODV) protocol [9-11] is intended for use by mobile nodes in an ad-hoc network. The network is dynamic, which means it does not depend on infrastructure. Instead, it relies exclusively on node composition, navigating through nodes and transferring data from one node to another until the desired node is reached. Since the protocol is not immutably linked to a specific node or path, it is efficient and flexible. Also, since MANET is reactive and on-demand, large cost reductions can be achieved, and in the event of a compromise, redundancy can also aid in maintaining network connectivity without being affected by the disabled part. However, its advantages can also be considered drawbacks which present dependency issues. For example, MANET's reliance on node composition to transfer messages is the biggest drawback since MANETs are characterized by dynamic topology, high-node mobility and limited battery power. Network outages or disconnections and message delivery failure can occur if as few as one node is out of the network range.

The current research developed a new protocol multipath protocol called ad hoc on-demand distance vector-secure multipath routing (AODV-SMR) to work on an uncrossed/non-intersection multipath, as opposed to singular path transmission. Sending data packets through more than one path leads to a decrease in data packet loss. The evaluation of the proposed protocol shows significant improvement in packet delivery ratio and outperforms the ad hoc on-demand distance vector (AOMDV) protocol in average end-to-end delay and outperforms AODV in the normalized routing load metric.

The rest of the paper is organized as follows. In Section II, related work is reviewed. Section III presents the proposed

AODV-SMR protocol framework. Section IV discusses performance analysis of the proposed protocol. Section V presents a case study. Section VI displays the performance results. Finally, Section VII concludes the paper and Section VIII suggests directions for future work.

II. RELATED WORK

In this section, author review different studies on multipath routing protocols including protocols and simulation analysis.

The node-disjoint multipath routing protocol (NDM_AODV), presented in[12, 13], aims to overcome the limitations of on-demand, unipath routing protocols such as AODV and DSR [14]. The shortest reverse routing hops and loop-free paths techniques are incorporated in the NDM_AODV protocol. This protocol reduces the routing overhead and ensures multiple path disjointedness and sufficiency. It also monitors the residuary energy of nodes. Simulation results of the NDM_AODV protocol showed that it improved the packet delivery ratio and reduced the normalized routing load and average end-to-end delay.

In [15], the authors propose a novel optimized QoS multipath routing protocol (O-QMRP) for MANETs to overcome the limitations of other MANET protocols. It considers the interaction of multiple factors across different layers in order to identify node disjoint paths with the least delays. The authors used expected path delay as a metric to select routes that considered the following points: 1. the signal-to-noise ratio at the physical layer, 2. the maximum data rate at the MAC layer, 3. the node's queuing delay, which reflects link quality and wireless medium utilization around the node (i.e., channel awareness, queue size, and load). As opposed to both AODV and AOMDV, where criteria for path selection is based on minimum hop-count, their proposed protocol takes into consideration the delay encountered at each node during path selection. As such, their simulation results showed that their proposed protocol, O-QMRP, had less average end-to-end delay compared to AODV and AOMDV.

Besides [15], the authors propose multipath extensions to AODV. Their resulting protocol is referred to as ad hoc on-demand multipath distance vector (AOMDV). Their protocol is characterized by its low inter-nodal coordination overheads, the ability to find and discover multiple disjoint paths without using source routing and minimal additional overhead (compared to AODV) when obtaining alternate paths. The authors observed that AOMDV, in comparison with AODV, reduces packet loss by up to 40% and offers a significant reduction in delay. It also improves routing overhead, by approximately 30%, by reducing the frequency of route discovery operations.

In [16], the authors propose a node-disjoint routing protocol (NDJ-AODV) that discovers multiple node-disjoint paths to the destination. This protocol can be used for load balancing, QoS-based routing and, more specifically, for performing energy-efficient routing. This protocol does not employ the path accumulation concept used by most node-disjoint routing protocols, but rather, it applies the concept of overhearing in a wireless medium to identify two node-disjointed paths. The routes discovered by the route discovery procedure are loop free. The performance results confirmed that the NDJ-AODV is particularly efficient in discovering multiple node disjointed paths with minimal routing overhead. The protocol also results in a high percentage of data packet delivery ratio and minimizes average end-to-end delays.

Several studies have produced multipath routing protocols not in MANET. Examples of these protocols include a "jamming-resilient multipath routing protocol" (JarmRout), which means that intended interruption or/and jamming along with local and remote failures tend not to obstruct the general network function of FANETs [17]. The design of the JarmRout protocol combines three main structures, namely "traffic load scheme", "link quality scheme" and "spatial distance scheme" [ibid]. The purpose of these schemes is to identify spatial node-disjoint multiple paths and combine them with high link quality and light traffic load in order to deliver data packets from source nodes to destination or target ones. Accordingly, the authors of the research will use OMNeT++ to develop a modified, disconnected and stimulus-based framework to evaluate its functionality by means of intensive simulation experiments. The latter are selected or planned in terms of packet delivery ratio, packet delivery latency, energy usage, and "end-to-end communication outage" ratio. The results will help to indicate whether the JarmRout protocol has the capacity to improve the latency and ratio of packet delivery on one hand and decrease the communication outage ratio on the other.

Another multipath routing protocol is known as L-CROP. Here, the resultant routes can lead to a limited stage of interference to one another, due to the distributed algorithm built on the overhearing of "path reply packets". This protocol signals the capacity to cope with issues related to subsurface communication. Therefore, the authors aim to simulate the proposed resolution in underwater networking situations to emphasize its ability to achieve better results of packet delivery ratio and fewer cases of disruption-based packet loss. This takes place with respect to general multipath routing approaches, in cases where the latter are placed on the top of MAC protocols known for their disruption or interference-avoiding trait [18, 19].

III. PROPOSED AD HOC ON-DEMAND DISTANCE VECTOR - SECURE MULTIPATH ROUTING (AODV-SMR) PROTOCOL FRAMEWORK

After studying the AODV system, it was found that it needed the following improvements to operate more efficiently: 1) Create a number of paths between the source and the destination. This helps the continuity of communication; even if the connection fails in one of the paths, the connection may still work on an alternate path. 2) Create a system that helps in avoiding crossing paths between the source and the destination. 3) Amend the data transmission system to make the source node use multiple paths instead of a single path. This reduces congestion and delays. 4) Amend the transmitter system to repeat the packets sent through the paths. This increases the effectiveness of the network to reduce data loss, increase confidence and secure data transfer. 5) Identify lost packets or amendments in the message header which show if there is an intruder or malicious packet in the system.

Converting AODV to multipath resulted in five main obstacles. The rest of this section will discuss each obstacle and present the solution used to overcome it.

A. *Modification to the problem of nodes replying to first request only*

The first problem that was faced was that all nodes (i.e., source, intermediate, destination) only saved the first request or reply in the routing table. They update or ignore requests that occur after saving the first request.

In AODV, a node receives a set of path requests from the same node or different nodes.The protocol only records the first request and then compares it with the following requests. If it is the latest or shortest, it updates the registered request in the routing table. However, if it is anything else, it ignores it. Therefore, to modify this protocol to allow for multiple paths, the procedure for receiving messages needs to be modified.

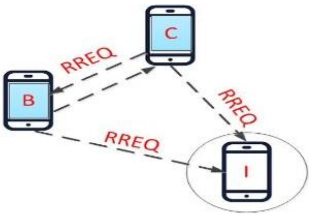


Fig. 1 (a): Node I received two requests from both node B and node C

If the request was sent from a node which did not send the previous request, it adds the request. For example, in Fig. 1 (a), since node I received two requests from both node B and node C, it is required to save both.

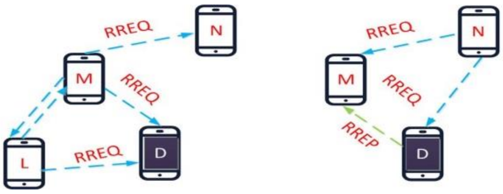


Fig. 1 (b): Destination node receives requests from a group of nodes

The destination node also presents the same problem, because when the destination node receives requests from a group of nodes, it only adds the request from the first node. As shown in Fig. 1 (b), first, node D received two path requests from nodes M and L, then it received a new request from node N. However, even though the destination node is receiving the requests, it will only record the first request and ignore the rest.

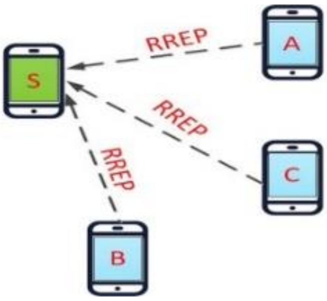


Fig. 1 (c): Source node must keep all replies to create different paths

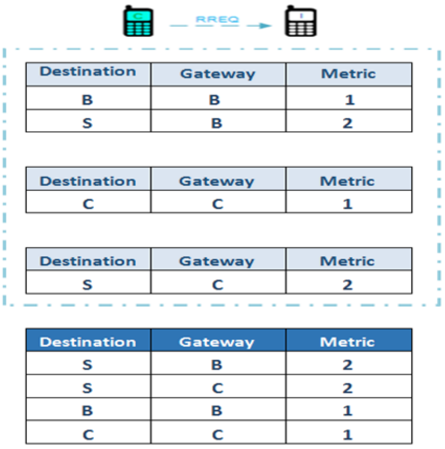
As shown in Fig. 1 (c), the same problem also appears later, in both the source and intermediate nodes, when replying to path requests. These nodes record only the first reply and ignore the rest. While the intermediate node is only required to save one unique reply (which is not repeated with the corresponding node), the source node must save all replies in order to provide alternate paths to the destination node.

To overcome the above-mentioned problems, modifications are done to the intermediate, destination and source nodes.

1) *Intermediate Nodes*

A set of conditions and rules are applied to the set of path requests. These include the non-recurrence of recording the request more than once as well as not storing an old request or a long path. An incoming request is compared with what is saved in the routing table. If the sequence number of the incoming request is greater, it means that the message is the latest, and updates its data. However, if it is identical to or smaller than the message already received or older, it ignores the message. If the sequence number is identical, but the number of hops is less, the path is shorter. If the message sequence number in the table is unknown, then it processes the update. The modifications will be performed on the function that is receiving requests (which occurs after the node receives the request).

These rules ensure that the intermediate nodes are able to overcome the first problem, that the system is efficient with no empty circles and that the requests are not only updated, but also that the shortest path is chosen. However, the data does not update when the following scenario occurs: if the node that sent the recorded message in the table (from the previous hop) is different from the node sending the new message. When this occurs, then a new path is added ,and the data is not updated.



Destination	Gateway	Metric
B	B	1
S	B	2

Destination	Gateway	Metric
C	C	1

Destination	Gateway	Metric
S	C	2

Destination	Gateway	Metric
S	B	2
S	C	2
B	B	1
C	C	1

Fig. 1 (d): Create two paths, from node C and B, respectively.

The result of these modifications is shown in the routing table in Fig. 1 (d). As can be seen from the figure, Node I, which already had a request from node B, received a new request from node C. Node I added the request from node C. Now it has two paths: one from node C and another from node B. Each one requiring two hops.

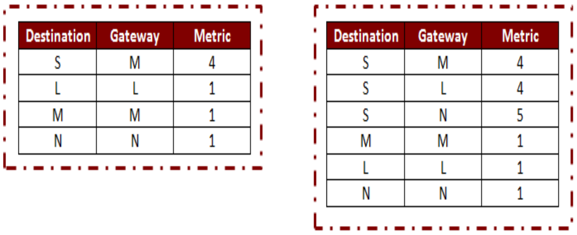
2) *Destination Nodes*

The rules that are followed for the destination node differ from those for the intermediate node. The modification focuses on the following two points: message expiration and adding new requests.

• Message Expiration: Due to the age of the message, the message is deleted before it reaches the destination node, especially if the number of request messages that reach the destination is more than two. To resolve this problem, the following exception is added: if the current node is the destination node, then there is no deletion (even if expired).

• Adding the new request: An exception to the rules was added; that if the current node is the destination node, it will add the new request.

Fig. 1 (e) shows the routing table of the destination node before and after the amendment is performed. In the first table, the destination node D saves one path for the node S through node M, which needs four hops. However, after the amendment, shown in the second table, the node holds three paths through nodes M,L and N. Nodes M and L need four hops while node N requires five.



Destination	Gateway	Metric
S	M	4
L	L	1
M	M	1
N	N	1

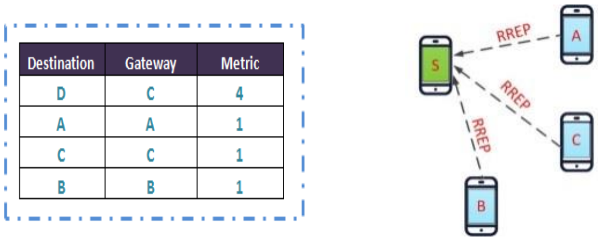
Destination	Gateway	Metric
S	M	4
S	L	4
S	N	5
M	M	1
L	L	1
N	N	1

Fig. 1 (e): The destination node routing table before and after the amendment

3) *Source Nodes*

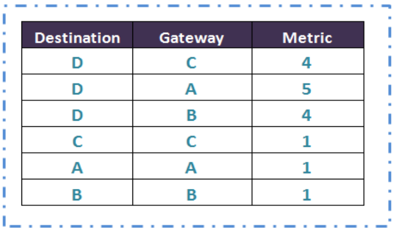
The source node is different from the previous two cases since it does not receive path requests. Instead, it receives replies to path requests which it must store. Therefore, we amend the part that receives the reply and not the part that receives the request. Also, an exception to add replies is also added. The exception preserves the conditions needed for the node to not receive duplicate replies, update new ones and to keep the replies that come from a different node.

This is shown clearly in the routing table in Fig. 1 (f). In the first diagram of the routing table, we found that the source node saved one reply from node C to reach node D in four hops.



Destination	Gateway	Metric
D	C	4
A	A	1
C	C	1
B	B	1

Fig. 1 (f): Source node saves one reply from node C to reach to node D in four hops



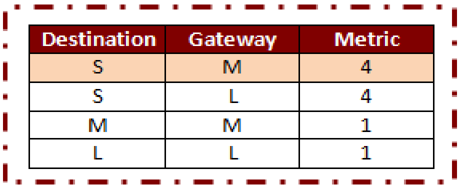
Destination	Gateway	Metric
D	C	4
D	A	5
D	B	4
C	C	1
A	A	1
B	B	1

Fig. 1 (g): Source node saves three paths from node C and B

However, after the amendment, shown in Fig. 1 (g), the source node ends up saving three paths to the destination from nodes A, B and C. While nodes B and C need four hops, node A requires a longer path of five hops.

B. Modification to the problem of the destination node replying to first node only

The second problem that was faced was that when a request to the destination node is sent, the destination node always replies to the first node in the table. When the destination node replies to a request, it searches for requests which carry the source node address and sends its reply. However, when more than one request is listed, it replies only to the first request every time, as shown in Fig. 2 (a). This problem appeared after the first problem was solved. The destination node’s reply follows the following condition: that the request must have the source node path.

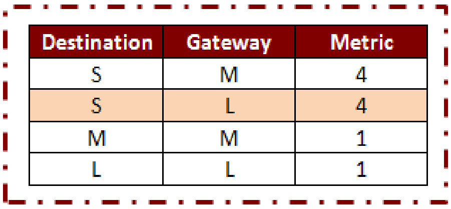


Destination	Gateway	Metric
S	M	4
S	L	4
M	M	1
L	L	1

Fig.2 (a): The reply is only sent to the first node that sent the first request in the routing table

In the function responsible for sending replies, the request contains the source node path. It saves the next hop in the variable (nextHop), which it uses to send the replies. The problem is that this variable is constant in all the replies. It is also the first record in the table with the applicable condition that prevents the creation of new paths.

In order to solve this problem, the code was amended so that the node replies using the last request (i.e., the latest request that was sent). Since the reply is immediately sent after receiving the request, we made a cycle (a loop) on all of the elements in the routing table where the condition applies. Then, the last request to put the path inside the variable (nextHop) is chosen, as shown in Fig. 2 (b). This amendment solves problem two.



Destination	Gateway	Metric
S	M	4
S	L	4
M	M	1
L	L	1

Fig.2 (b): The destination node responds to all requests and creates different paths

C. Modification to the problem of intersection of intermediate node replies

The third problem was regarding the intersection of intermediate node replies; whereby intermediate nodes choose the same node to reply to. This problem is similar to the previous one, but with intermediate nodes instead of the destination node. The intermediate node acts like the destination node; it searches for the first element that the condition is applied on. This creates the problem, because the requests are sent to the nodes synchronously, so the order of request registration is the same. Therefore, the reply is the same, which means that all the nodes reply to the node that sent the first request. While this problem is similar to the one with the destination node, the solution applied there cannot be used here. That is because if all the intermediate nodes use the last request, then the problem is still not solved, and all nodes will still send a reply to one node.

To fix this problem, we created an alert message that is sent by the intermediate node after it receives the reply it sent to the other nodes informing them that it has chosen a certain node to send to. Then, each of the other nodes deletes this node from its routing table. When the reply is sent to one of the other nodes, it will choose another unused node with no intersection. It will then alert its neighboring nodes to its choice. As can be seen in Fig. 3 (a), node L alerted its neighboring nodes that it chose node I to reply to. Node M then deletes this node from its table.

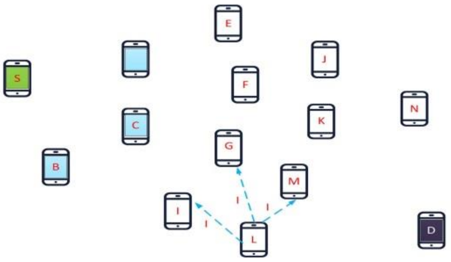


Fig. 3 (a): Node L alerts neighboring nodes that it chose node I to reply to

D. Modification to the problem of the appearance of a fault before deleting an intersecting path

The fourth problem was that a fault appears if the node receives a reply before deleting the intersecting path. It is possible that the deleted node is the one which is prepared to reply. Here an error message appears because the deleted node cannot be found after deletion. Therefore, the response/reply must be delayed until the node deletes the intersecting paths and then receives a reply.

The events occur as shown in the example in Fig. 4 (a). First, node L sends a path request to node D, and node D sends a reply. Then node M sends a path request to node D, and node D sends a reply. After that, node L sends a message to node M asking it to delete node I. Then, node M deletes node I, and node L sends the reply to node I. When it is node M’s turn, a fault occurs because it was prepared to reply before deletion, and it relied on the existence of the deleted node.

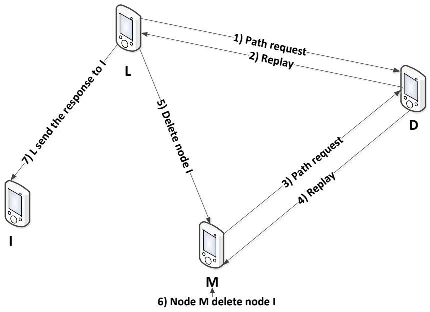


Fig.4 (a): Node M receives a reply before it deletes the intersection path.

To solve this problem, the events were reordered so that the deletion occurs before replying. The events will occur in the following order: First, a general variable, called (N_rrep), is added to distinguish the first reply from the rest of the replies to the source node. It is initialized to zero '0'. Second, a delay is allowed for sending other replies. N_rrep will test if a reply is the first since first replies are passed without delay.

However, subsequent replies are given a delay. After solving these issues, the system will be able to create a group of non-intersecting paths. The system will send data across these paths, ensuring delay reduction, increase in productivity and reduction of data loss.

E. Modification to the problem of ending path discovery after the first reply

The final problem that was faced was that the source node ends the search and discovery for paths when it receives the first reply and begins data transmission. It does not wait for the rest of the replies to return and save their paths.

This is the final problem that is faced when coverting AODV from one path to multipath. After the source node sends a path request which reaches the destination node, the destination node then replies to each path request with the reverse path. The problem is that when the source node receives the first reply, it launches the function to indicate the end of the discovery of a path (completeRouteDiscovery).

To solve this problem, a variable to calculate the number of paths received at the source node was added. The variable is compared to a request-calculating variable. When there is a match or when the waiting time finishes, the function begins. The source node waits for the arrival of the last reply (or the end of the wait time) and then calls the (complete Route Discovery) function.

IV. PERFORMANCE ANALYSIS

In the previous section, we explained concepts regarding the AODV protocol and the mechanisms by which data is transmitted. We also described the ways in which we developed the system to work more efficiently. To confirm the validity of the research, we chose the AOMDV protocol [20] to act as a benchmark. AOMDV was chosen since it has a high reputation in most systems and protocols, and since it was the most widely used benchmark for comparison in previous studies. The comparison will be based on the AODV protocol since both the AOMDV and our new AODV-SMR protocol are built on it.

We will present the results of these comparisons through group trials that were conducted using mobility and pause time. We will compare the properties of the three protocols as well as the quantitative performance results according to the following three metrics, packet delivery ratio, average end-to-end delay and normalized routing load.

F. Shared Protocol Operations

The following sub-sections describe the four shared operations, as shown in Fig. 5, carried out by the three protocols (AODV, AOMDV and AODV-SMR): route discovery procedures, creation of routing paths, road maintenance, and conditions for updates and addition/creation.

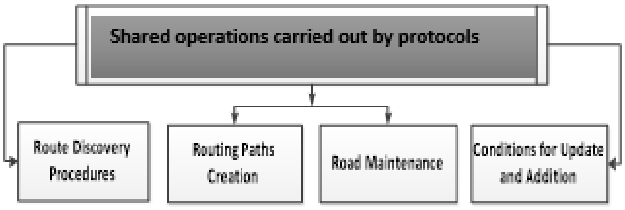

Shared operations carried out by protocols
Route Discovery Procedures
Routing Paths Creation
Road Maintenance
Conditions for Update and Addition

Fig.5: Shared operations carried out by protocols

1) Route Discovery Procedures

Route discovery procedures begin when a particular path is needed. The source node sends a flood of path requests in the form of RREQ messages and awaits replies. After the destination node and neighboring nodes (intermediate nodes) receive the message, they record the reverse path to the source node and increment the number of hops (i.e., hops+1). Then, they insert the data (e.g., sequence number, reverse path, number of hops) into the routing table.

The message is then rebroadcasted. The sequence number ensures that the message is not resent to the same nodes, while the hop number accelerates the data update process by finding the shortest path available. Moreover, the discovery process continues until the message reaches the destination node.

2) Creation of Routing Paths

The destination node replies to a request message with an RREP message. This reply uses the reverse path to reach the source node (of the applicant). Routing paths between the source node and the destination node can then be created from reverse path groups.

3) Road/Path Maintenance

Path stability is maintained by sending an RERR (error message) in the event of contact with a node is lost. This type of message is sent to the source nodes to instigate the re-discovery process, and in turn, shift data transmission to an alternative path. Also, AODV utilizes self-cleaning mechanisms such as inspecting the validity in messages, continuously updating paths and deleting old or unused paths.

4) Conditions for updates and addition/creation

There are conditions that prevent receiving repetitive messages and loop configuration within the routing path. They also regulate the continuous update of messages; giving preference to the shortest and most recent messages. These conditions prevent nodes from adding or updating any message unless it is new, newer or shorter in hop numbers. This process is completed by reviewing the sequence number, the number of hops and the validity date. These conditions dictate that the sequence number of the new message needs to be greater than the message number in the routing table. The number of hops also needs to be less. If these conditions are met, the update occurs. If these conditions are not met, then the message is ignored, and no update occurs. This is the most basic code in AODV, which has been developed by both protocols.

As previously described, the protocol automatically compares the incoming request with what is recorded in the routing table. Based on the sequence number, one of the following scenarios occurs:

1) A sequence number greater than the one recorded indicates that the message is recent and updates the data accordingly.
2) If it is identical or smaller, the message may have already been received or older, in both cases it is ignored. If the sequence number is identical, but the number of hops is less, then the path is shorter; the shortest path is always chosen.
3) If the message in the table has an unknown sequence number, then it will automatically update. This is due to the amendment made to the receiving request function, a function that operates after the node receives the request.

AODV-SMR protocol is significantly different from the AODV and AOMDV protocols. However, it is similar to AODV since the protocol does not contain any amendments to the routing table nor does it add a variable. AODV-SMR, however, has increased flexibility. It allows the node to retain more than one message if the message is from a different node. Therefore, while the routing table does not differ in structure, the difference is in the number of messages permitted.

V. CASE STUDY

The following is a case study that elaborates the difference between how the three protocols (AODV, AOMDV and AODV-SMR) create paths.

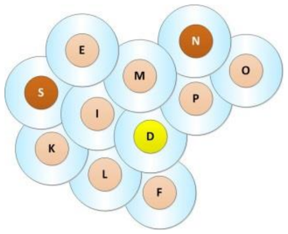


Fig.6 (a): Case architecture

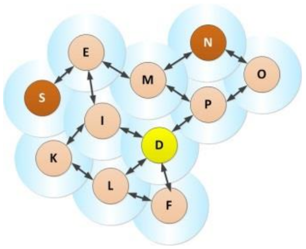


Fig.6 (b): The nodes continue broadcasting the message to each other after deployment

As can be seen in Fig. 6 (a), the source node, S, is connected to three nodes. The destination node, D, is connected to four nodes. Node I (an intermedite node) is connected to six nodes. Since the nodes are overlaping, this offers a variety of paths that can be configured and utilized.

We will assume that a path request RREQ message is deployed. While AODV halts the discovery process as soon as the source node receives a reply RREP, AODV-SMR delays halting discovery until a complete number of reply messages are received by the source node. We also assume that the network is flooded by messages and that nodes continue broadcasting to each other. The deployment appears as follows in Fig. 6 (b) for the three protocols.

The difference between the protocols is in the way each protocol deals with these messages, the amount of feedback kept by each intermediate node through the procedures and by the conditions set by the protocol to save messages in the routing table. The biggest difference, however, lies in the reply mechanism and how feedback is used to form paths from the source node to the destination node for data and packet transmission. The objective of this example is to explore the last concept further. Each protocol treats feedback uniquely and paths are constructed in a multitude of ways.

AODV: It is a basic protocol that produces a single path from the source node to the destination node. The chosen path is normally the latest and shortest path, as shown in Fig. 6 (c).

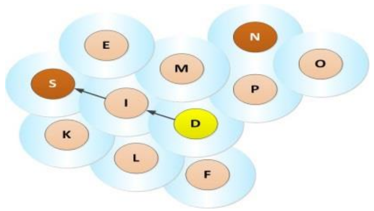


Fig.6 (c): Single latest and shortest path produced by AODV

AOMDV: It acquires a high number of paths because it is regulated by simple conditions. It allows nodes to participate in more than one path and comparisons between paths based on length are not required. In this example, the number of connected paths reached 32. Fig. 6 (d) only displays the direct paths, without displaying the rest of the paths, including those which contain many loops Also, this protocol is only link-disjoint or node-disjoint path [21]which appears is node I that participates with more than nine paths..

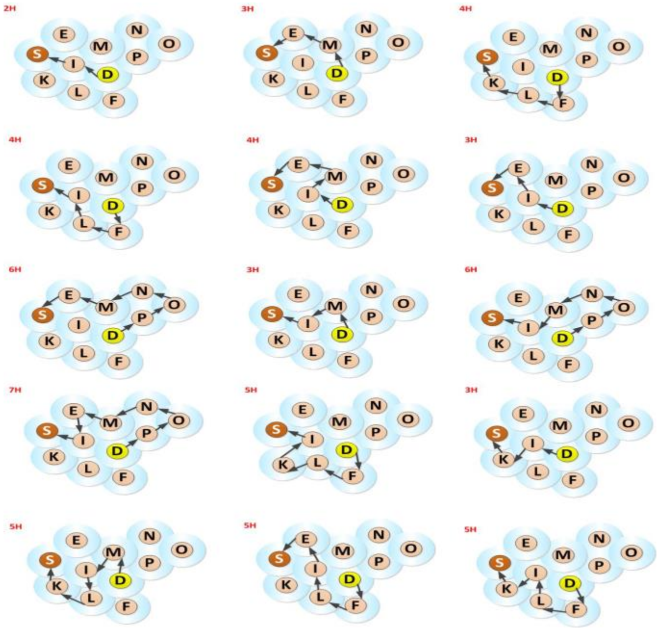


Fig. 6 (d): AOMDV producing a high number of intersected paths

AODV-SMR: It is the best of the three protocols for the following reasons. The protocol constructs a set of paths rather than a singular path. It maintains the non-intersection of paths and avoids creating any loops. Importantly, there is no delay in AODV-SMR caused by extensive path discovery. Also, congested pathways are avoided in this protocol since nodes do not participate in more than one path. AODV-SMR has two useful characteristics: First, the simplicity of its implementation; the lack of complexity in the modifications to the structure of the routing table as well as to the terms of updates and addition. Second, the power and accuracy of use; the processes that govern the identification of a limited number of paths and selection of the most appropriate, as shown in Fig. 6 (e).

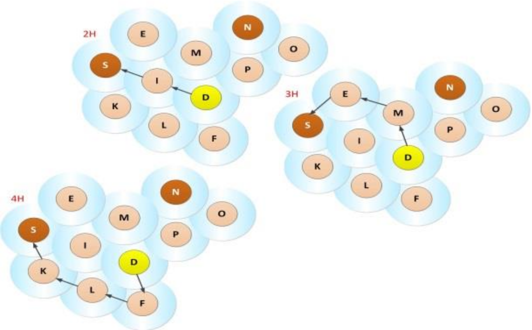


Fig. 6 (e): The paths produced by SRM -AODV protocol / proposed protocol

VI. FINDINGS - PERFORMANCE RESULTS

The performance of the AODV-SMR protocol was evaluated using OMNET++. It assesses the effectiveness of the AODV-SMR, AODV and AOMDV protocols through a set of performance metrics. There are two scales for each metric. The first scale (mobility) works using 6 various speeds: 0 (stability) 5, 10, 15, 20 and 25 meters per second, and the results are tested at a constant time duration of 100 seconds. The second scale (pause time) uses a constant speed of 5 m/sec, and the results are tested for different periods.

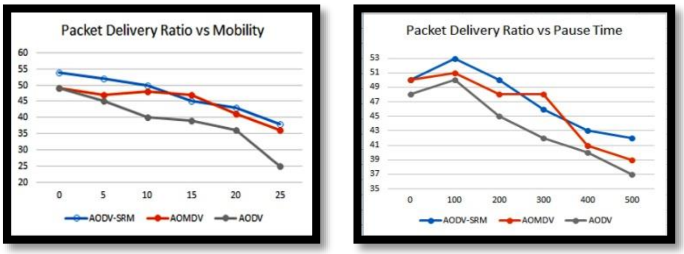
The autors used standard IEEE 802.11 as a layer of MAC and the beneficiary of feedback mechanism. The network parameter was 1000 × 1000 sq. and the number of nodes was 50.

Table 1: Simulation Parameters

Parameters	Value
Simulation Time	700 sec
Scenario Dimension	1000*1000 sq.
Channel Type	Channel/Wireless Channel
Antenna Model	Omni Directional Antenna
Radio Propagation Model	Two Ray Ground
MAC Layer Protocol	AODV-SMR
Number of Nodes	50 nodes
Transport Protocol	UDP
Routing Protocol	AODV-SMR, AODV, AOMDV
Mobility Model	Random Way Point
Maximum Node Speeds	0, 5, 10, 15, 20, 25 m/sec
Pause Times	0, 100, 200, 300, 400, 500 sec
Traffic Type	Constant Bit Rate (CBR)
Packet Size	512 bytes

The following subsections provide further information on the performance metrics that were used: packet delivery ratio, average end-to-end delay, and normalized routing load. A comprehensive comparison will be undertaken between the three protocols incorporating percentage of differences, and also the benefits of each system will be discussed. Comparison of the protocls for each metric will be based on mobility or pause time. For mobililty, the simulation is held at a constant duration of 100 seconds for 6 varying test speeds: 0 (stability), then 5, 10, 15, 20, and 25 meters per second. For pause time, the speed is maintained at 5 m/sec, and data is collected once the pause time reaches 0, 100, 200, 300, 400, and 500 seconds.

A) Packet delivery ratio: This is the ratio of packets received by the destination node to the packets generated by the source node.



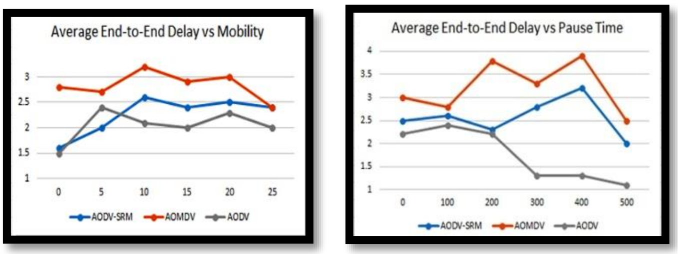
Mobility(a) **Pause time(b)**

Fig. (7): Packet delivery ratio vs. mobility and pause time

The X axis in Figs 7 (a) and 7 (b) denotes mobility (m/sec) and pause time (s), respectively. The Y axis, in both Figures, denotes the data packet delivery ratio (%). As can be seen in Figure 7, as speed increases, mobility increases. There is also a positive correlation between message loss probability and speed. Pause time measures protocol performance at a given duration in the presence of different traffic. Whenever the traffic increases, the number of messages received decreases.

The AODV-SMR protocol achieved a higher packet delivery ratio compared to the AODV. That is because packets are distributed across more than one path, thereby reducing the proportion of lost packets. AODV-SMR also displays the following improvements over AOMDV: it avoids the intersection of paths, sharing of nodes and interminable alternative path discovery. Therefore AODV-SMR is better than AOMDV in more than one feature.

B) Average end-to-end delay: It measures the time elapsed from when packets are created in the source node until they are received by the destination node. The delays in all sent packets are summed, and the result is divided by the number of packets received. This average is called the average delay.

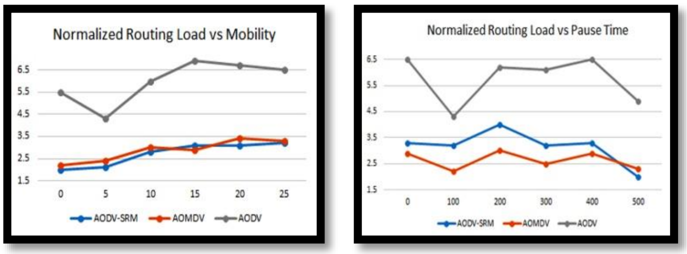


Mobility(a) **Pause time(b)**

Fig. (8): Average end-to-end delays vs. mobility and pause time

The X axis in Figs 8 (a) and 8 (b) denotes mobility (m/sec) and pause time (s), respectively. The Y axis, in both figures, denotes the average end-to-end delay (s). As can be seen from Fig. 8, AODV has better results and fewer delays because it halts the discovery process and begins the data transfer after securing a single path. However, as mobility increases, the multiple paths feature appears, in which case, the protocol does not initiate the re-search process, but relies on an alternative path. AODV remains a superior protocol even as mobility increases. When comparing AOMDV with AODV-SMR, AODV-SMR is better because AOMDV is considered the protocol with the highest delay in packet delivery. As such, this shows a discrimination of AODV-SMR in another side of the performance.

C) Normalized routing load: Normalized routing load per packet is the number of control packets required by each data packet. It is also called the control packet overhead. The normalized routing load reflects the efficiency of the routing protocol, and indirectly reflects the stability of the paths in the dynamic environment. The smaller the overhead, the more stable the path. It is computed by dividing the general cost by the number of packets received from the source to the destination. It provides an estimate of the cost of loading each message. This average is called the average delay.



Mobility(a) **Pause time(b)**

Fig. (9): Normalized routing load vs. mobility and pause time

The X axis in Figs 9 (a) and 8 (b) denotes mobility (m/sec) and pause time (s), respectively. The Y axis, in both figures, denotes the normalized routing load. As shown in Fig. 9, AODV-SMR competes with AOMDV regarding least cost. AODV bears the most cost due to the number of packets lost and the number of re-discovery processes required, which increase the cost shared between the packets.

VII. CONCLUSION

In this paper, we proposed a new node and link-disjoint multipath routing protocol based on the AODV protocol. The new protocol overcomes the limitations of on-demand single path routing protocols such as AODV, and multipath routing protocols such as AOMDV. Multipath routing protocols can limit the impact of mobility-related link failures in MANETs. Also, link failures related to energy depletion can be successful over multipath routing protocols. The proposed AODV-SMR is a highly efficient and effective protocol. It provides more reliability in a mobile ad-hoc network.

The authors evaluated the performance of AODV-SMR using OMNET++ simulation. Our results show that AODV-SMR outperforms the most prominent multipath routing protocol, AOMDV, particularly in the packet delivery ratio metric. An additional merit of AODV-SMR is that it utilizes alternative non-intersecting paths and can actively repair communication failures caused by node movements in and out of range.The protocol can successfully reduce data loss, improve delay durations, and reduce the pace of path discovery.

VIII. FUTURE WORK

As described previously, the AODV-SMR is the on-demand multipath routing protocol. The authors will develop this protocol to enhance passive acknowledgment (for security). Also, AODV-SMR will adequately defend itself against a large class of attacks and reduce the effect of many other attacks. AODV-SMR will provide more resistance against different attack classes. This comes with a slight increase in routing overhead. We will extend AODV-SMR to defend against other attack categories, such as Sybil attacks.

REFERENCES

1. Toh, C.K., *Ad hoc mobile wireless networks: protocols and systems*. 2001: Pearson Education.
2. Liu, J., et al., *End-to-end delay modeling in buffer-limited MANETs: A general theoretical framework.* 2015. 15(1): p. 498-511.
3. Khanna, N. and M.J.I.J.o.C.S. Sachdeva, *Study of trust-based mechanism and its component model in MANET: Current research state, issues, and future recommendation.* 2019. 32(12): p. e4012.
4. Persis, D.J. and T.P.J.I.N. Robert, *Review of ad-hoc on-demand distance vector protocol and its swarm intelligent variants for Mobile Ad-hoc NETwork.* 2017. 6(5): p. 87-93.
5. Komai, Y., et al., *K nearest neighbor search for location-dependent sensor data in MANETs.* 2015. 3: p. 942-954.
6. Oleshchenko, L. and K. Movchan. *AODV Protocol Optimization Software Method of Ad Hoc Network Routing*. in *International Conference on Computer*

Science, Engineering and Education Applications. 2021. Springer.

7. Goyal, P., et al., *Manet: vulnerabilities, challenges, attacks, application.* 2011. 11(2011): p. 32-37.
8. Khudayer, B.H., et al., *Efficient route discovery and link failure detection mechanisms for source routing protocol in mobile ad-hoc networks.* 2020. 8: p. 24019-24032.
9. Nemade, C.H. and U.J.R.G.-G.I.E.T. Pujeri, *Execution Evaluation of AODV Protocol Using NS2 Simulator for Emergency Automobile.* 2021. 11(4): p. 1792-1801.
10. Das, S., C. Perkins, and E.J.I.R. Royer, July, *Ad hoc on demand distance vector (AODV) routing.* 2003. 10.
11. El-Semary, A.M. and H.J.I.A. Diab, *BP-AODV: Blackhole protected AODV routing protocol for MANETs based on chaotic map.* 2019. 7: p. 95197-95211.
12. Ding, S. and L. Liu. *A node-disjoint multipath routing protocol based on AODV*. in *2010 Ninth International Symposium on Distributed Computing and Applications to Business, Engineering and Science*. 2010. IEEE.
13. Goyal, M., et al., *Modified Dragon-Aodv for efficient secure routing*, in *Advances in Computing and Intelligent Systems*. 2020, Springer. p. 539-546.
14. Thakker, V.M., et al. *Choosing optimal routing protocol by comparing different multipath routing protocols in mobile Adhoc networks*. in *2018 2nd International Conference on Inventive Systems and Control (ICISC)*. 2018. IEEE.
15. Obaidat, M.a., et al. *A novel multipath routing protocol for MANETs*. in *2011 7th International Conference on Wireless Communications, Networking and Mobile Computing*. 2011. IEEE.
16. Marina, M.K., S.R.J.W.c. Das, and m. computing, *Ad hoc on-demand multipath distance vector routing.* 2006. 6(7): p. 969-988.
17. Pu, C.J.I.A., *Jamming-resilient multipath routing protocol for flying ad hoc networks.* 2018. 6: p. 68472-68486.
18. Azad, S., P. Casari, and M.J.I.w.c.l. Zorzi, *Multipath routing with limited cross-path interference in underwater networks.* 2014. 3(5): p. 465-468.
19. Jiang, S.J.W.N.P.F.T.t.U.A., *Routing in UWANs.* 2018: p. 287-313.
20. Taha, A., et al., *Energy efficient multipath routing protocol for mobile ad-hoc network using the fitness function.* 2017. 5: p. 10369-10381.
21. Khan, K., W.J.I.J.o.I.S. Goodridge, and Applications, *Energy aware Ad-Hoc on demand multipath distance vector routing.* 2015. 7(7): p. 50-56.