THE ANNALS OF "DUNĂREA DE JOS" UNIVERSITY OF GALATI
FASCICLE III, 2021, VOL. 44, NO. 1, ISSN 2344-4738, ISSN-l 1221-454X
ELECTROTECHNICS, ELECTRONICS, AUTOMATIC CONTROL, INFORMATICS
Article DOI: https://doi.org/10.35219/eeaci.2021.1.03

Set Backup Routes on Sensor Network

[1]Saad S. Naif, [2]Shian S. Sulaiman, [3]Mohammed Hussein shukur

[1]*College of Medicine, University of Duhok, Duhok, Kurdistan region, Iraq, (e-mail: saad.naif@uod.ac).*
[2]*College of Medicine, University of Duhok, Duhok , Kurdistan region, Iraq, (e-mail: shian.sulaiman@uod.ac).*
[3]*Department of Computer Science, College of Science, Cihan University-Erbil, Erbil, Kurdistan region, Iraq, (e-mail: Mohammed.shukur@cihanuniversity.edu.iq)}*

Abstract: There are many techniques of using routing protocols on Sensor Network's mobile nodes, yet failures can occur in Wireless Networks path. High mobility leads to slipping nodes in the routing table, causing packet loss; current protocols suggest using rediscover paths. Nevertheless, this process increases delay. This paper proposes a design of two different methods to select the shortest multipath and select the longest multipath to get different scenarios to Set Backup Routes on Sensor Network "SBSN" to avoid using rediscover routes by providing different paths and improve the performance and efficiency of a network depending on the method of packets processing declared in the Proposed System. The proposed technique merges An Ad Hoc On-Demand Distance Vector "AODV" and Multipath-Ad Hoc On-Demand Distance Vector "M-AODV". Comparison of a delay in transmitting rate, loss of packets, and packet delivery ratio to ensure a network's transmitting performance and efficiency Achieves low broadcast redundancy. It avoids the challenges of establishing paths from low to high-mobility of nodes between a single source and destination.

Keywords: MANET, AODV Multi-path, AODV, Disjoint, SBSN, Backup Route.

1. INTRODUCTION

The heterogeneous mobile ad hoc network MANET connection can happen through a wireless mesh network. [1] In dynamic topology, the mobility nodes are responsible for much control and delays in the routing of wide network congestion [2]–[4] and large loading [5]–[8]; there are many Routing Protocols, such as M-AODV, which is used for selecting multipath between the source and destination node. [2], [9], [10] However, some of these protocols are inefficient and have a short life span on changing topology networks "mobile network nodes" [10], [11] such as a Sensor Network by lowering network efficiency since this protocol generates more control packets by the nodes on the network [3], [12], [13]. When a node receives a control packet from a neighbor RREQ Route Request, RREP Replay Request, it rebroadcasts the packet after updating the routing table, if it specifies a new path between source and destination based on the first hop in RREQ and RREP. [13]–[15] As a result, all of these control packets will increase the likelihood of packet congestion on the node channel [4], [10], raising routing overhead [4], [16], routing discovery latency [3], [16], and these parameters will affect the risk of packet failure [3], [10], [14], decreasing packet distribution ratio [3], [4], [17], and increasing routing discovery frequency [15]. A major problem with mobile multi-hop Networking is how to effectively deliver packets between highly efficient nodes the multi-routing protocol generates sets of Routes. Every sensor operates in the network, and network topology changes spread between the nodes. Due to packet loss, large overhead, and network delay, Sensor network congestion can be triggered by the routing discovery and the change in the topology for the wireless network. [6], [12], [18]. Therefore, this study aims to reduce packet loss, increase packet delivery ratio, and avoid challenges to improve a network.

2. RELATED WORK

The Multipath Routing Protocol allows the discovery of multiple paths between nodes for communication with overhead and traffic control packets like M-AODV used to compute Multiple loop-free and link-distinguished paths for each node along with a list of the next loops, to track multiple routes to routing entries [6], [7], [9], [18], [19]. The number of hops are used to submit destination path advertisements. An intermediate node accepts RREQ only with a different end hop. Each node route advertising specifies an alternative route to a destination so that several reverse routes are established and identified in their route list at intermediate nodes to the same destination. [6], [9]. A node supports loop freedom only[6]. Therefore, the advertised hop count does not adjust with the same hop number since the maximum hop count is used. All such copies will, however, lead to routing load if they are accepted to create routes. The first Hop field is given for each RREQ to indicate the first source neighbor. Each node searches a first list for each RREQ to track the source neighbors list receiving a copy of the RREQ. The destination will only respond to an RREQ through a single neighbor to ensure connection disjoint in the first hop of the RREP [7], [12], [15]. Multiple routes between nodes that wish to contact can be efficiently found through the Multi-Routing Protocol MRP [5], [19], [20]. This algorithm is based on AODV Multi-path [12]. Since nodes mobility in a network, this protocol can prevent traffic congestion and frequent communication interruptions. Disjoint protocol uses route accumulation as an essential component of the protocol to escape loops [5], [19], [20]as the topology of the sensor network varies dynamically and there are few nodes in the network. The Disjoint Algorithm is a separate routing detection method from AODV Multi-path, and the source routing algorithm is used. The Disjoint protocol can classify network traffic into different priorities and implement priority planning and queuing management mechanisms to ensure QoS by seeking the shortest trajectory with lower loads. [12], [21]. Since it minimizes routing overhead by using the next-hop in any separate RREQ and RREP to choose multipath between source and destination, this article proposes a Disjoint Multi-Routing Protocol DMRP system as a normal approach in achieving and supplying QoS in ad hoc network. We made the source node as the destination node for RREP to allow the intended node to choose competing positions. Moreover, in the node domain, the next hop for the reverse routing continually transmits RREP and ensures the first disjoint node of the RREP packet route transmitted is reciprocated. The possibility of the node to find backup routes SBSN in this way.

3. PROPOSED SYSTEM

The node will start the route discovery procedure with a SBSN routing protocol with a local service feature to establish new RREQ Route Request packet route diffusion. The source routing and broadcast node table in RREQ waits for an RREP Route Reply packet, the reverse routing loop can be avoided by providing several separate paths to the destination node as much as possible depending on the next-hop inside the control packet. The vast network size is possible to use the SBSN. It connects many nodes through the next hop to hold the intermediate node ID, RREQ and RREP control packets. Often minimize overheads due to the use of the following hop field instead of the route accumulation field, since the routing table size at each node decreases, and storage sizes are small enough to store several routes.

3.1. Pseudocode of SBSN

3.1.1. Shot path:

1. SBSN::recvRequest(Packet *p)
2. Scr checks if routes to Dst are exist. Else broadcast RREQ.
3. Drop RREQ if Scr = Index.
4. Drop RREQ if ((saddr = Index or Seq_Num_ RREQ >= Seq_Num_RT) & (hop_count_RT) > (hop_count_ RREQ)) .
5. Else previous_path_delete & Reverse_path_insert (saddr, hop_count_RREQ+1).
6. forward(SBSN_rt_entry*)
7. Else if ((forward_path && (reverse_path_lookup (reverse_ nexthop, reverse_ Scr) == NULL))))
8. Else if ((rq->rq_dst == index) && (new_ SBSN.path(saddr(),reverse_ Scr))) { Reverse_path_insert (saddr, hop_count_RREQ+1) }
9. sendReply (RREQ_src,RREQ_dst,Seq_Num_ RT,saddr)}
10. SBSN:: recvReply (Packet *p)
11. If ((saddr(),!= index) & (Seq_Num_ RREP >= Seq_Num_RT) & ((hop_count_RT) > (hop_count_ RREP))) {
12. previous_path_delete & forward_path_insert (saddr, hop_count_RREQ+1)}
13. forward((SBSN_rt_entry*)

3.1.2. Long path:

1. SBSN::recvRequest(Packet *p)
2. Scr checks if routes to Dst are exist. Else broadcast RREQ.
3. Drop RREQ if Scr = Index.
4. Drop RREQ if ((saddr = Index or Seq_Num_ RREQ >= Seq_Num_RT) & (hop_count_RT) < (hop_count_ RREQ)) .
5. Else previous_path_delete & Reverse_path_insert (saddr, hop_count_RREQ+1).
14. forward(SBSN_rt_entry*)
15. Else if ((forward_path && (reverse_path_lookup (reverse_ nexthop, reverse_ Scr) == NULL))))
16. Else if ((rq->rq_dst == index) && (new_ SBSN.path(saddr(),reverse_ Scr))) { Reverse_path_insert (saddr, hop_count_RREQ+1) }
17. sendReply (RREQ_src,RREQ_dst,Seq_Num_ RT,saddr)}
18. SBSN:: recvReply (Packet *p)
19. If ((saddr(),!= index) & (Seq_Num_ RREP >= Seq_Num_RT) & ((hop_count_RT) < (hop_count_ RREP))) {
20. previous_path_delete & forward_path_insert (saddr, hop_count_RREQ+1)}
21. forward ((SBSN_rt_entry*)

3.2. Multipath Routing Discovery

The "Scr" source node initially checks whether the source node sends data packets to "Dst" the destination node of the routing information, If routes are exist, or the source node will begin to scan paths by sending process requests to nodes broadcasting RREQ control packets and using "saddr" the next hop of intermediate nodes to get multi node-disjoin If routes are not exist. In case of turn upside-down at the ending RREQ packets. The intermediate node excludes RREQ if it has already been received until the next hop equals the index or RREQ has a large number of hops, and it accepts the receiving of RREQ. When "Seq Num_ RREQ" Sequence number in RREQ is greater than "Seq Num_RT" Sequence number in routing table, or when Sequence number in RREQ equals Sequence number in routing table, and "hop count RT" in routing table is greater than "hop count_ RREQ" hop count in RREQ, the shortest route is identified. The node then deletes the previous reverse path and records a new reverse path that includes the neighbor node's IP address from the next hop in RREQ to the routing table and appends its IP address to the same next-hop, then rebroadcasts RREQ to the neighbor. If the routing table has the forward route to the destination node, the intermediate node returns the RREP to the source node. Here, the process will reduce the delay of discovering the route; otherwise, it broadcasts the RREQ to the neighbors. This method will continue until the RREQ reaches the destination node, at which point, a sequence of routes will be calculated by matching the current next hop in the obtained RREQ with all possible next hops in a set of reverse paths in the routing table. Then, in the same reverse route as receiving RREQ, The destination send RREP to the source node; when an intermediate node receives RREP, it deletes the previous forward path from the routing table and it applies the IP of the neighbor node from the next hop of RREP to the forward path, then it records the IP to next-hop and forwards RREP to a single reverse-path when the RREP Sequence number is greater than the routing table Sequence, or RREP Sequence Number is equal to the routing table sequence number. However, the hop count RREP is less than the Hop count in the routing table, so this process will contain at any hop after another hop until the path discovery source node is completed. using another method, we modify this routing protocol to select the longest path. Therefore, intermediate nodes and destination nodes support RREQ when the RREQ count is larger than the hop count on the routing table and they accept RREP if the RREP count is larger than the routing table's hop count.

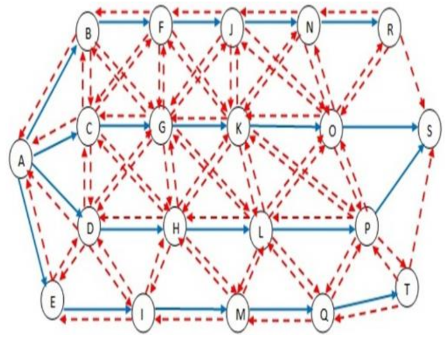

Fig. 1. Select 2 paths between node A and S. reject RREQ by intermediate or destination node because of large numbers hop count or RREQ already be received. accept RREQ because of short hop count. Nodes B, C, D, and E accept request from source node because the first hop is equal to 1, node A rejects request from B, C, D, and E because source id in RREQ is the index, node B rejects RREQ from node C because the hop count in RREQ is greater than hop count in the routing table, node G accepts RREQ from node C and rejects RREQ from node B for some reason, such as the “delay” of processing many RREQ on node B received from other node .

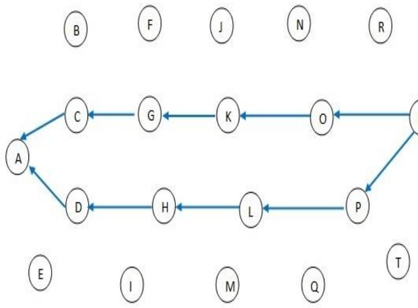

Fig. 2. Select the 2 shortest paths between node A and S. accept RREP by intermediate node and the source code.

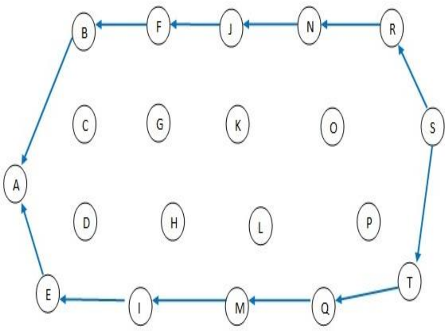

Fig. 3. Select the 2 longest paths between node A and S. accept RREP by intermediate node and the source code.

3.3. Path Maintenance

A route error message contains the previous Hop and the next-hopaddress while managing the routing process. When the link from a nearby node is blocked into the source along the reverse track, the error message will be created when an error message from a nearby node is received. A routing loop is an issue for different node network types. They happen if a routing algorithm processes errors and the path to a specific destination forms a loop in a node group. If a node has already been visited by the packet limit of having just one path per destination, a new randomly selected path would be generated, but this issue has been avoided in RREQ and RREP by relying on the next hop. When it computes several loop-free paths at a destination node, there are two problems. Which node should announce to others in many routers? Since each path has different hop numbers, an arbitrary selection can lead to loops, so the next hop field in RREQ can be used to avoid this case by holed node id in every process. The second node should consider the advertised routes. Again, naively accepting all tracks can lead to looping. SBSN Routing computing has three main features that help it achieve a low redundancy and prevent a broadcast flood in Sensor Network. Next hop in RREQ is decreasing broadcast packets in multipath routing “Shortest path and the longest path previously discovered during PREQ broadcasting are two different cases between intermediate nodes on the network” to selecting backup routes.

4. IMPLEMENTATION

We use NS-2 network simulator with 50 nodes moving in 500 m 500 m of rectangular space and the stay time of 0. The source node sends a broadcast message RREQ and RREP, each one is 512 bytes in size, and the period of the simulations 500 second. The two max. movement speeds of "10 m/s, 20 m/s, 30 m/s, 40 m/s, 50 m/s " were simulated for each node. A different pattern of rectangular area nodes comprises the simulation environment. In the area, the nodes are located randomly, and each has a radio range. As a model of mobility node, the probabilistic graphical model is selected.

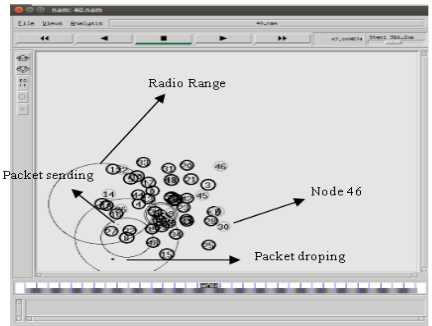

Fig. 4. Network Animator

5. RESULTS

The following parameters evaluate the output of two SBSN protocol cases:

5.1. The End-to-End Delay

All possible receiving packet delays as we show in fig 5, the short path method is better than the long path method for the time of discovering path because we have greater number of nodes at long paths and the neighbor nodes move out of the radio range so that the processing will be the longest.

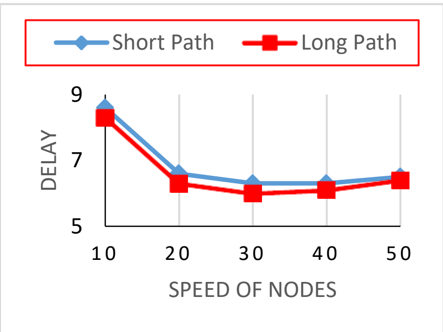

Fig. 5. The End-to-End Delay

5.2. Average of Packet Delivery Ratio

Does the target node obtain the packet ratio over the packet number sent by the source? It gives an indicator of the routing protocol's end-to-end distribution functionality. The Average Packet Delivery Ratio in the long path method is less than the short path method because of the expiring of time to live at every packet reason of many node numbers and more processing times will cause packet loss.

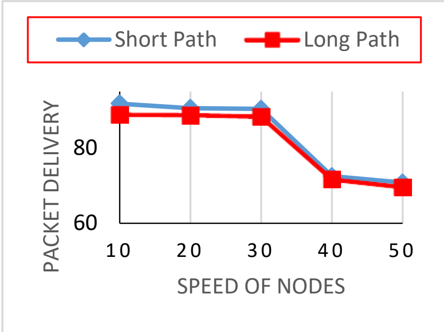

Fig. 6. Average of Packet Delivery Ratio

5.3. Average Packet loss Ratio

The major cause for the loss of packets is congestion and the end of time to live. If a crash occurs between two or more network-wide packets between intermediate nodes, the packet is lost.

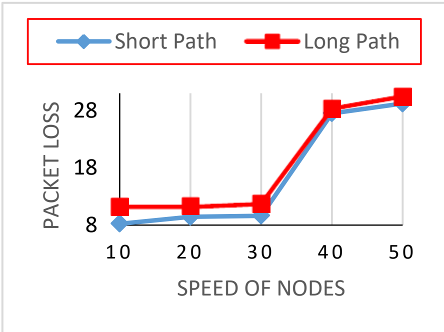

Fig. 7. Average of Packet loss Ratio

5.4. Average Packet throughput

The high ratio of throughput in the short path means there are fewer nodes in paths, so there is a low probability of packet loss compared to discovering long path needing extra processing time for any number of packets to re-found the route to the destination node.

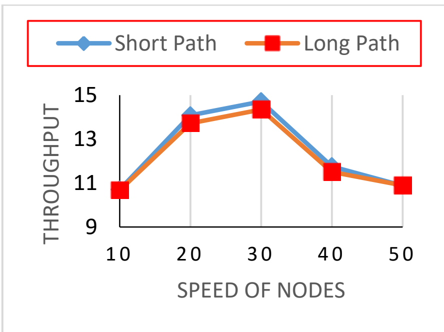

Fig. 8. Average Packet Throughput

6. CONCLUSIONS

By recording the shorter hop count in the routing table entry, SBSN has provided high Quality of Service and it has reached to several Backup Routes. The simulation's effects are obvious: the shortest path in SBSN outperforms the case of the longest path because the shortest path has a higher packet delivery ratio, lower packet loss rate, low end-to-end delay, and high Average Packet throughput. These characteristics make the protocol suitable for the highest performance for multipath between the source nodes and destination nodes. The routing rate of renewal is less than that of the longest route within the complex network topology, adjusting in a way that higher than the cost of routing the shorter path protocol, thus decreasing the amount of network packet routing reducing the likelihood of packet crashes and increasing network efficiency. In a future work, we intend to select nodes with low mobility speed in two different cases of short paths and long paths then evaluate additional parameters.

REFERENCES

[1] A. T. Sasongko, G. Jati, and W. Jatmiko,(2018), Multipath Routing Protocols in Vehicular Ad hoc Networks (VANETs): Performance Evaluation in Various Conditions, *ISSIMM 2018 - 3rd Int. Semin. Sensors, Instrumentation, Meas. Metrol. Proceeding*, pp. 34–38, doi: 10.1109/ISSIMM.2018.8727733.

[2] R. Park, W. Kim, and S. Jang, (2013), Improved AOMDV to increase path stability by utilizing the node mobility information in MANET, *Int. Conf. Inf. Netw.*, pp. 417–422, doi: 10.1109/ICOIN.2013.6496415.

[3] F. Yang, S. Ling, H. Xu, and B. Sun, (2012), Network coding-based AOMDV routing in MANET, *Proc. 2012 IEEE Int. Conf. Inf. Sci. Technol. ICIST 2012*, pp. 337–340, doi: 10.1109/ICIST.2012.6221663.

[4] A. K. Yogi and J. Surana, (2017), An implementation of Modified AOMDV routing protocol in different wireless networks, *Proc. 2016 Int. Conf. ICT Business, Ind. Gov. ICTBIG 2016*, pp. 9–13, doi: 10.1109/ICTBIG.2016.7892677.

[5] A. Vidwans, A. K. Shrivastava, and M. Manoria, (*2014*), QoS enhancement of AOMDV routing protocol using queue length improvement, *Proc. - 2014 4th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2014*, pp. 275–278, doi: 10.1109/CSNT.2014.60.

[6] U. S. Kushwaha and P. K. Gupta, (*2014*), AOMDV routing algorithm for Wireless Mesh Networks with local repair (AOMDV-LR), *Int. Conf. Commun. Signal Process. ICCSP 2014 - Proc.*, pp. 818–822, doi: 10.1109/ICCSP.2014.6949957.

[7] V. Lalitha and R. S. Rajesh, (*2014*), Power aware and topology aware ad-hoc on-demand multipath distance vector routing for MANET, *2014 IEEE 8th Int. Conf. Intell. Syst. Control Green Challenges Smart Solut. ISCO 2014 - Proc.*, pp. 115–119, doi: 10.1109/ISCO.2014.7103929.

[8] M. Bheemalingaiah, M. M. Naidu, D. Sreenivasa Rao, and P. Vishvapathi, (2017), Performance analysis of power-Aware node-disjoint multipath source routing in mobile ad hoc networks, *Proc. - 7th IEEE Int. Adv. Comput. Conf. IACC 2017*, pp. 361–371, doi: 10.1109/IACC.2017.0084.

[9] D. N. Patel, S. B. Patel, H. R. Kothadiya, P. D. Jethwa, and R. H. Jhaveri, (2015), A survey of reactive routing protocols in MANET, *2014 Int. Conf. Inf. Commun. Embed. Syst. ICICES 2014*, no. 978, doi: 10.1109/ICICES.2014.7033833.

[10] B. Sharma, S. Chugh, and V. Jain, (2014), Energy efficient load balancing approach to improve AOMDV routing in MANET, *Proc. - 2014 4th Int. Conf. Commun. Syst. Netw. Technol. CSNT 2014*, pp. 187–192, doi: 10.1109/CSNT.2014.247.

[11] S. Stancescu and A. A. Luca, (2013), Optimal routing protocols in Wireless Sensor Networks, *2013 11th Int. Conf. Telecommun. Mod. Satell. Cable Broadcast. Serv. TELSIKS 2013*, vol. 1, pp. 149–154, doi: 10.1109/TELSKS.2013.6704911.

[12] Z. Qu, W. Ren, and Q. Wang, (2010), A new node-disjoint multi-path routing algorithm of Wireless Mesh Network, *2010 Int. Conf. Comput. Mechatronics, Control Electron. Eng. C. 2010*, vol. 4, pp. 1–3, doi: 10.1109/CMCE.2010.5609590.

[13] C. W. Ahn, S. H. Chung, T. H. Kim, and S. Y. Kang, (2010), A node-disjoint multipath routing protocol based on AODV in mobile adhoc networks, *ITNG2010 - 7th Int. Conf. Inf. Technol. New Gener.*, pp. 828–833, doi: 10.1109/ITNG.2010.158.

[14] H. Zhong and T. T. Zhou, (2019), Research and Implementation of AOMDV Multipath Routing Protocol, *Proc. 2018 Chinese Autom. Congr. CAC 2018*, pp. 611–616, doi: 10.1109/CAC.2018.8623785.

[15] P. K. Manohari and N. Ray, (2016), Multipath routing protocols in MANETs: A study, *2016 1st Int. Conf. Innov. Challenges Cyber Secur. ICICCS 2016*, no. ICICCS, pp. 91–96, doi: 10.1109/ICICCS.2016.7542318.

[16] T. Yelemou, (2013), New approach to improve the robustness of AOMDV protocol, *2013 10th Int. Multi-Conference Syst. Signals Devices, SSD 2013*, pp. 1–6, doi: 10.1109/SSD.2013.6564012.

[17] M. Thaalbi, N. Tabbane, T. Bejaoui, and A. Meddahi, (2013), A weighted QoS aware multipath routing process in Mobile Ad hoc networks, *Proc. - Int. Symp. Comput. Commun.*, pp. 336–341, doi: 10.1109/ISCC.2013.6754969.

[18] R. Mythreyi and A. Julian, (2015), QOD enabled AOMDV protocol for hybrid wireless networks, *ICIIECS 2015 - 2015 IEEE Int. Conf. Innov. Information, Embed. Commun. Syst.*, pp. 0–5, doi: 10.1109/ICIIECS.2015.7192919.

[19] V. Arya and C. Gandhi, (2014), NDj - AODV: Node disjoint multipath routing in Mobile Ad Hoc Networks based on AODV protocol, *2014 7th Int. Conf. Contemp. Comput. IC3 2014*, pp. 601–606, doi: 10.1109/IC3.2014.6897241.

[20] C. Lal, V. Laxmi, and M. S. Gaur, (2012), A node-disjoint multipath routing method based on AODV protocol for MANETs, *Proc. - Int. Conf. Adv. Inf. Netw. Appl. AINA*, pp. 399–405, doi: 10.1109/AINA.2012.49.

[21] S. Ding and L. Liu, (2010), A node-disjoint multipath routing protocol based on AODV, *Proc. - 9th Int. Symp. Distrib. Comput. Appl. to Business, Eng. Sci. DCABES 2010*, pp. 312–316, doi: 10.1109/DCABES.2010.70.