World Applied Sciences Journal 25 (1): 01-08, 2013
ISSN 1818-4952

DOI: 10.5829/idosi.wasj.2013.25.01.915

Dynamic Path-Switching: A Multiple Constrained QoS Routing Algorithm for Manets

[1]Albara W. Awajan, [1]Omar S. Al-Dabbas, [2]Firas A. Albalas and [2]Radwan S. Abujassar

[1]Department of Computer Engineering, Faculty of Engineering,
Al-Balqa Applied University Al-Salt P.O. Box: 19117, Jordan
[2]Faculty of Science and Advanced Technology,
Yarmouk University, Irbid P.O. Box 21110, Jordan

Submitted: Aug 13, 2013; **Accepted:** Sep 20, 2013; **Published:** Sep 24, 2013

Abstract: The rapid development of communication systems in the last few decades has increased the deployment of multimedia and real-time applications within it. These applications require the support of Quality of Service routing algorithms that support multiple constraints to provide a service the meets the user expectations. Applying QoS on mobile ad-hoc networks (MANET) raises the challenge because of MANET's dynamic nature. In this paper, a novel QoS routing algorithm is proposed, called Dynamic Path-Switching algorithm (DPS). DPS expands the normal QoS routing algorithms by trying to prevent path breakage instead of trying to maintain them. First, DPS discovers all paths that satisfy the QoS constraints for a particular service and chooses the highest stability. After sending data on the optimal path, DPS re-generates a path discovery request. When a new path is found, DPS starts sending data on it and discards the old path. This mechanism reduces path breakage because it changes transmission to new path before the old path breaks. DPS is simulated again AODV and MAODV in NS-2 simulator where simulation results showed that DPS performs better than both AODV and AOMDV algorithms. Simulation results also show that DPS is scalable and performs well in high mobile environments.

Key words: Dynamic-Path Switching · Mobile Ad-hoc Networks · Multiple Constraints · Quality of Service · Routing Algorithm · Stability

INTRODUCTION

A mobile ad hoc network (MANET) is a group of wireless mobile nodes that can be formed without the need of a fixed infrastructure or centralized administration. MANETS can be installed quickly and have the ability to operate in a stand-alone manner which makes it favorable in many situations such as military communications and disaster rescue.

Finding a path between source and destination is one of the common challenges in communication networks. Routing protocols and algorithms for MANETs are classified to three main types, proactive, reactive and hybrid. Proactive routing, where a routing table is maintained at every node, such as OLSR (Optimized Link State Routing Protocol) [1]and (DSDV) Destination-Sequenced Distance-Vector[2]. The drawback of this type of algorithms is that the table needs to be updated frequently which consumes a lot of memory bandwidth and energy. Reactive routing does not require the maintenance of routing tables. When a source node is ready to send traffic it initiates a route discovery process to find a suitable path. Examples of such algorithms are DSR (Dynamic Source Routing) [3] and AODV (Ad-hoc On- Demand Distance Vector) [4]. The reactive routing protocols decrease the routing overhead and energy consumption in the network. There are some MANET routing algorithms and protocols that use both methods, these methods are known as the Hybrid methods such as the ZRP (Zone Routing Protocol) [5].

The increased use and enhanced capabilities in mobile devices has resulted in the development of many multimedia and real-time applications that use MANETs to connect mobile devices together, whether they are mobile devices in conference rooms or mobile devices in rescue operations. In order to maintain the Quality of

Corresponding Author: Albara W. Awajan, Department of Computer Engineering, Faculty of Engineering, Al-Balqa Applied University Al-Salt P.O. Box: 19117, Jordan, Tel: +96253491111, Fax: +96253491111.

Service for these Applications, there is a need for QoS routing protocols and algorithms that establishes a communication path between the source host and destination host based on multiple constraints such as delay, loss and bandwidth, usually referred to, as QoS parameters. QoS routing using multiple constraints is a very challenging task as it is considered to be an NP-complete problem [6]. In addition, the dynamic nature of the topology for MANETs increases the challenge to find a path that satisfies the QoS constraints for real-time services [7, 8].

In this paper a novel QoS routing algorithm is proposed, called Dynamic Path-Switching algorithm (DPS), which finds the optimal path between the source node and the destination node that satisfies the QoS requirements for a particular service. The Algorithm takes into account the end-to-end stability state of the paths when selecting the optimal path to reduce link breakage in an established connection during transmitting of data. DPS starts sending data to the source node using the optimal path found, after a period of time and while data is still being sent, DPS generates a new path-discovery request. DPS switches sending packets from the old path to the new path found and this is repeated till all data is sent. This process is called path-switching technique which aims to prevent link breakage in paths before it happens, where most algorithms concentrate on maintaining paths after link breakage.

Related Work: The challenging objective of any QoS routing protocol is to find the optimal path from the source to the destination that meets the QoS requirements of the service requested. In the recent period many studies have been introduced for solving the QoS routing problem in MANETs. Many of the studies in this area have considered only one constraint such as delay [9] or bandwidth [10, 11]. Other studies find the path between the source and destination based on two constraints such as [12, 13]. Unfortunately, using one or two metrics is mostly not enough to insure the quality of service required for stringent and high demanding real-time services such as video conferencing. Therefore, multiple constraints are best to insure a high level of quality.

With regards to QoS routing with multiple constraints (more than two constraints) in MANETs, there is not much of literature available in this matter. In [14], a QoS multicast routing protocol is proposed (QoS_MAODV) where it finds the optimal path based on a two-level model. The metrics included in the path selection is divided into network layer metrics (hop-count, power level and stability and buffer level) and into application layer metrics (delay, throughput and cost). The route discovery process uses the network metrics to find the paths and uses the application metrics in the path selection. All network metrics are mapped to a class code that indicates the level and type of QoS, where each class is associated with a network metric (hop-count, delay or throughput). The QoS_MAODV algorithm proposed in [14] is a very interesting approach but in reality the algorithm only optimizes a single metric when choosing the optimal path. The metrics used in the route discovery process do not guarantee a path the meets the requirements of real-time and multimedia applications and is also targeted for multicasting networks. Another multi-constrained algorithm is proposed in [15] where a cost function is defined on many metrics (deadline, bandwidth, loss rate, neighbor number, hop count and the route quality) to result in one single metric. Then the path is found by minimizing the cost value. There are no results found in [15], as it still part of an ongoing work but in general the proposed algorithm does not guarantee an optimal path because all metrics are converted to one single metric which does not find the optimal path. In [16], Ad-hoc on Demand Multiple-Path Distance Vector (AOMDV) is proposed which is an extension to AODV algorithm that finds multiple disjoint paths from the source node to destination node. AOMDV simulation results show a high improvement in the performance with regards to end-to-end delay, overhead and packet loss.

In conclusion, most literature is concerned in optimizing the QoS constraints but does not consider the dynamic environment and movements of paths after a path connection is established. The remaining literature tries to choose paths based on stability such as [17-20], but does not solve the path breakage after a path connections is established. In this paper, DPS extends what is done in the literature and tries to prevent path breakage before it happens. DPS algorithm finds all paths from the source node to the destination node using multiple constraints, where the np-complete problem is reduced by filtering out any path that violates the QoS constraints. Then the path with the highest stability is selected as the optimal path. DPS starts sending data to the source node using the optimal path found, after a period of time and while data is still being sent, DPS generates a new path-discovery request. DPS switches sending packets from the old path to the new path found and this is repeated till all data is sent. This process aims to prevent link breakage in paths because DPS keeps looking for a better path before the path connections breaks.

In this paper the DPS will be compared with AODV [4, 21] and AOMDV [16, 22] using the NS-2 simulator to see the overall performance.

QoS Constraints: Many network constraints have been used by routing algorithms to determine the best path; most common used metrics are hop count, cost, path-length, bandwidth, delay, jitter (delay variation), security and packet loss (unreliability) [23]. Some constraints can be relevant to one service but not to another. Each service has a set of QoS constraints that have to be met in order to deliver that service; thus, it is important to define these constraints and the minimum/maximum value for each constraint to be used in the routing process.

In [6], constraints are classified based on the composition rules. Accordingly, if C(n1, n2) is the constraint of a link connecting the two nodes n1, n2 then, for any path P= (n1, n2,…, ni), constraints could be classified in to three types:

Additive Constraints: Where the calculation of the constraint for the whole path, is the sum of the constraint's value for each individual link as follows:

$$m(p) = m(n1, n2) + \ldots + m(ni-1, ni) \quad (1)$$

Delay, delay jitter and cost are considered to be additive metrics.

Multiplicative Metrics: Where the calculation of the constraint for the whole path, is the product of the constraint's value for each individual link as follows:

$$m(p) = m(n1, n2) \times \ldots \times m(ni-1, ni) \quad (2)$$

Loss Probability can be indirectly considered a multiplicative metric, because it has to be transformed to an equivalent metric that follows the composition rule. So if, 2 represents the success ratio over the link(n1, n2), then the loss probability over the whole path can be given as:

$$m(p) = 1-[(1-m(n1, n2)) \times \ldots \times (1-(m(ni-1, ni))] \quad (3)$$

Concave Metrics: Where the calculation of the metric for the whole path is calculated as in the following:

$$m(p) = \min[m(n1, n2), \ldots m(ni-1, ni)] \quad (4)$$

- Bandwidth is a concave constraint.

Stability is a constraint that is used to give an indication for the probability of a link to persist for a certain life span. Many studies have proposed various models to estimate and calculate the stability constraint for a link and the full path [17-20]. In this study, a random value between 0 and 1 is given to indicate the stability constraint based on normal distribution. The stability model is considered a concave constraint.

In this study, QoS constraints are divided into two types.

Essential QoS Constraints: The constraints that must be satisfied when finding the optimal path in order to deliver a particular service. The most essential constraints used when delivering real-time and multimedia applications are throughput, delay and packet loss. Therefore the essential QoS constraints are used when establishing connection between the source host and the destination host.

Supportive QoS Constraints: These constraints are not essential for delivering a particular service, but they are important to increase the life-time of path which the service is being delivered on. MANETs have dynamic topologies resulting from the mobility nature of their nodes. This dynamic characteristic cause's link breakage in communication paths before delivery of the service is finished. Supportive QoS constraints such as stability and battery life are used to find a path that is more reliable which decreases the risk of link breakage. Hence, increasing the overhead and time of repairing or re-allocating a new path for delivering the service.

It should be clear that the essential QoS constraints are more important than supportive QoS constraints because if any of the essential QoS constraints are not satisfied in the established path, the service will not be delivered appropriately. Whereas, the supportive QoS constraints if not met will not stop the service but will increase the overhead of the network which can be tolerated.

Algorithm Procedure

Path Discovery: The path discovery process finds all the paths that satisfy the essential QoS constraints for a particular service. When the source node needs to send data to a destination node it floods a route request message (RREQ) to the destination node. The main fields in a RREQ message are the sequence number, source address, destination, route list, bandwidth, delay jitter and stability. When a RREQ message reaches the neighbor node it updates the QoS parameters in the RREQ message.

If the QoS parameters violate the QoS application constraints the RREQ message is discarded, if it does not violate the QoS Constraints it is forwarded again to the neighbors of that node. The stability parameter is calculated but not used in the path discovery process; it is used later in the path selection process. This process is repeated until the RREQ message reaches the destination node or discarded if it violates the QoS constraints.

The destination node will receive all RREQ messages that satisfy the QoS application constraints. Each RREQ message represents a distinct path from the source to the destination with the end-to-end QoS parameter values. The RREQ message also stores the list of nodes it passed while reaching the destination to be used by the Route Reply message (RREP).

Path Selection: In the path discovery process the RREQ messages represent the paths that satisfy the application QoS constraints. Therefore, if no RREQ messages reach the destination node that means that there is no path that satisfies the QoS constraints, if more the one RREQ message reaches the destination node, which means many paths satisfy the QoS constraints and that is usually the general case. When more than one path is available the algorithm chooses the path with the highest stability. The destination node sends a route reply (RREP) message to the source node using the reverse path stored in the RREQ message for the chosen path to start sending data. The stability parameter is not used in the path discover process, it is only used in the path selection process to pick the most stable path from all available paths. The stability constraint is chosen as a second stage because it can be tolerated because selecting a path randomly from the available paths will also be able to deliver data within the quality requested because all of them satisfy the QoS constraints, but the life time of the path chosen randomly would not be the same as the path chosen with the highest stability.

Path Maintenance: The algorithm proposed is dynamic and jumps from one path to another in time intervals to insure that the QoS constraints are always up to date and the end-to-end stability of the path is the best available. The algorithm does not wait for the path to break in order to maintain or find another path. While the source node is transmitting data to the destination node, it looks for a new path that satisfies the QoS constraint then starts delivering of data packets over the new path. The process is repeated until all data is sent from source to destination.

Table 1:NS2 Simulation parameters

Parameters	Value
Number of nodes	10-50
Simulations time	500 seconds
Environment size	1000m x 1000m
Traffic size	CBR (continuous bit) rate
Packet size	512byte
Queue length	50
Mobility model	Random waypoint
Antenna type	Omni directional
Radio transmission range	250m
Mobility speed	2-10 m/s
Throughput capacity channel	2Mbps
Pause time	100-600 s

This is done because while data is sent the nodes are in continuous movement and the risk of path breakage increases every second.

Another important advantage for path-switching is that when a path connection breaks while data is transmitted from source node to destination node, there is no need to generate a new path request or to maintain the broken path. And that is because DPS in that instance of time is already looking for a new path so it can switch data transmission to it.

Simulation and Results: The DPS algorithm is implemented and evaluated using the NS-2 simulator by extending the AODV routing protocol. The RREQ messages of the AODV protocol where extended to include the QoS metric. The DPS calculates and updates the values of the QoS metrics in the RREQ messages on each hop of the path. The RREQ messages are dropped by a node and are not forwarded if the QoS metrics in the RREQ data packet violate the QoS requirements. DPS algorithm is examined against AODV and AOMDV algorithms using the NS2 simulator.

To evaluate the DPS, a simulation was performed in NS-2 simulator where the simulation environment contained different scenarios that contained from 10 to 50 randomly distributed mobile ad-hoc nodes moving in a 1000mx1000m area for 500s. The mobility model used in the simulation was the random waypoint model, where the node moves from random start point to a destination point, after reaching the destination it pauses for a while before it starts moving to a new destination. The radio transmission range for each node was set to 250m. The speed of nodes varies from 2m/s to 10m/s. data source is CBR (Continues Bit Rate). Data packet size is 512byte where each flow generated 10 data packets per second. The throughput capacity channel is 2Mbps. And the pause time varies from 100s to 600s. The simulation parameters used in the simulation are summarized in Table 1.

The performance measures used to compare the proposed algorithm with AODV and AOMDV are 1) packet delivery ratio which is the ratio between the number of packets received to the number of packets sent. 2) Average overhead packets which are the number of control messages that are used to manage the network. 3) end-to-end delay which is the total delay that the packet needs to travel from the source to destination. 4) Throughput which is the rate that the packets flow through the network per second. The simulation scenarios in this paper test two important features of ad-hoc algorithms. The first feature is scalability, where the algorithm should be able to perform well with the increase of the size of the network. The second feature tested is mobility where the nodes pause is decreased when changing the direction of the node.

Scalability Simulation Results: In this section the effect of increasing the network size on the performance measures will be evaluated for DPS, AODV and AOMDV. This test gives an indication if the algorithm is scalable or not where scalability is the ability of the algorithm to handle the growth of the network in terms of number of nodes. In this scenario all parameters will be as stated earlier and the number of nodes will be increased with each simulations run.

Figure1 shows the effect of increasing the number of nodes on the packet delivery ratio (PDR). In general the PDR decreases with the increase of the number of nodes, this is due to the increase of the intermediate mobile nodes, therefore, increasing the probability of path breakage and the loss of data packets. Figure 1 shows that both DPS and AOMDV perform better than AODV when increasing the number of nodes. DPS also performs better then AOMDV, this is because DPS considers the stability metric which insures less path breakage, where AOMDV sends packets on multiple paths but does not have a backup path and does not consider the stability of the paths available.

Figure 2 shows the effect of increasing the number of nodes in the network on the end-to-end delay for DPS, AODV and AOMDV. It is clear from Figure2 that the delay for all three algorithms increases with the increase of number of nodes because of the raise in the number of links in a path and also results in an increase in link breakage in the paths. AOMDV performs better than DPS and AODV in small size networks. When network size is more than 20 nodes DPS performs better than AODV and AOMDV because DPS considers the delay and stability metrics when choosing the optimal path where AODV and

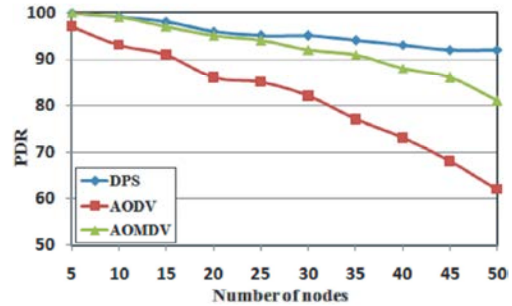

Fig. 1: Number of nodes VS. PDR

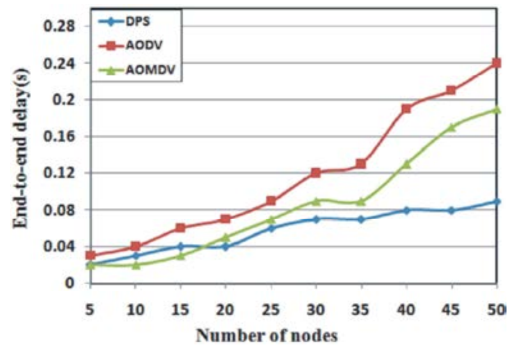

Fig. 2: Number of nodes VS. End-to-End delay

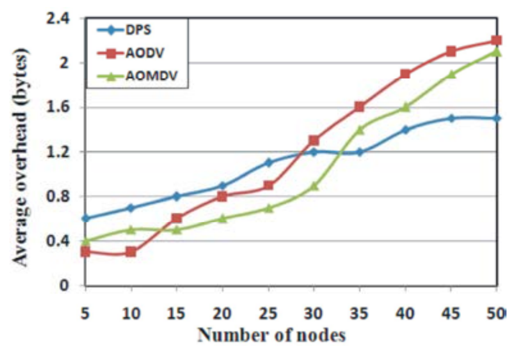

Fig. 3: Number of nodes VS average overhead

AOMDV do not. In addition, DPS path-switching technique decreases link breakage before it occurs, which decreases the end-to-end delay.

Figure 3 shows that the increase in the number of nodes in the network results in an increase in the overhead messages in the network. This is because of the increase in link breakage between nodes therefore the need to re-establish a new connection.

AOMDV performs better in small sized networks then AODV and DPS. But with the growth of network size the total overhead in AODV and AOMDV starts to increase while the total overhead for DPS increases less than AODV and AOMDV. DPS has less overhead because the paths chosen are more stable and path breakage possibility is much less than the other two algorithms.

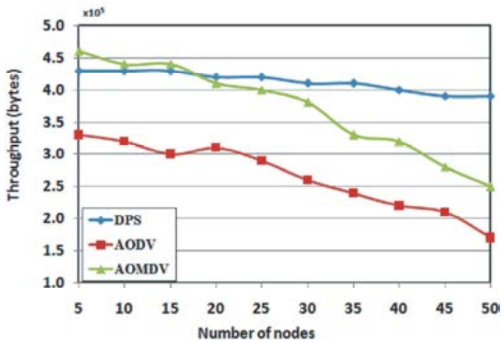

Fig. 4: Number of nodes VS. throughput

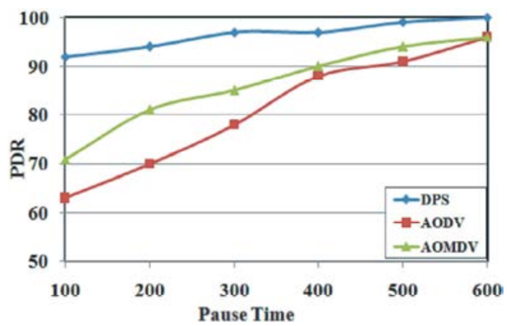

Fig. 5: Pause time VS. PDR

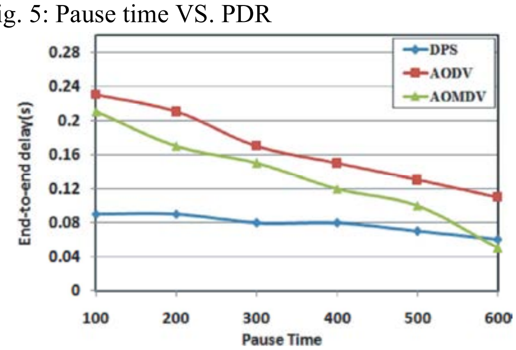

Fig. 6: Pause time VS. End-to-End delay

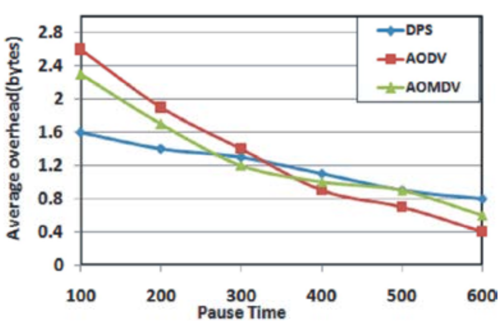

Fig. 7: Pause time average overhead

In Figure 4, it is very clear that DPS performs better than both AODV and AOMDV with regards to throughput. Both AODV and AOMDV do not support multiple constraints whereas DPS optimizes bandwidth, delay and Jitter. .

Mobility Simulation Results: In this section the effect of increasing the mobility on the performance measures will be evaluated for DPS, AODV and AOMDV. These simulation scenarios give an indication of how good the algorithms can handle the mobility in the network nodes. The pause time parameter is used to represent the mobility in the network where pause time is the time the node stops at the destination point before changing direction and going to another destination. In this scenario all parameters will be as stated earlier and the pause time will be increased with each simulations run. The results are shown and explained as follows:

Figure 5 shows that increasing the pause time for the nodes in the network increasing the PDR for all algorithms and that is because increasing the pause time decreases the mobility in the network and as a result, link breakage becomes less. It is clear from Figure5 that DPS algorithm performs better than AODV and AOMDV when the pause time is decreased. This is because DPS selects the optimal path based on its stability metric. The improvement over AODV is around 30% and over AOMDV is around 20% when pause time is 600s.

Figure 6 shows that with all algorithms, the increase of pause time the end-to-end delay decreases because the mobility is less with high pause times. It is also noticeable in Figure6 that DPS manages to perform well when decreasing the pause time because it considers the stability metric in the path discovery and selection process. DPS outperforms AODV and AOMDV .

Figure 7 shows the effect of pause time on average overhead for DPS, AODV and AOMDV. The increase of pause time for the mobile nodes decreases the overhead because the links become more stable. DPS in general has high overhead even if the pause time is high because of its continuous search for alternative paths. Also, AOMDV has high overhead because it searches multiple paths to send data on. Thus, AODV has lower overhead when the mobile nodes have high pause times. But when the pause time decreases and mobility becomes higher, AODV's overhead messages exceed the average overhead for DPS and AOMDV because link breakage increases with low pause time. DPS out performs both AODV and AOMDV in low pause times because it considers the stability of nodes when choosing the optimal path.

In Figure 8, it is noticeable that DPS performance is much higher than AODV and AOMDV with regards to Throughput. This is because DPS optimizes throughput when searching for the optimal path. It is also clear that DPS can perform well with the increase of mobility, whereas AODV and AOMDV performance drops down.

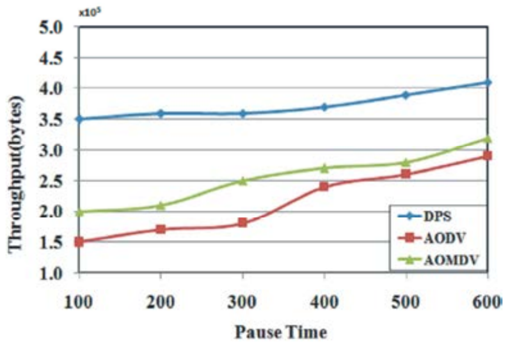

Fig. 8: Pause time v throughput

Conclusion and Future Work: In this paper a QoS routing algorithm called Dynamic path-switching is proposed. DPS uses multiple constraints to find the optimal path and takes into account the end-to-end stability of a path. DPS also decreases path breakage by using the path-switching mechanism that periodically switches data transmission from the current path to a new generated path. Path-switching aims to find a new path before old path breaks because of continuous node movement. DPS algorithm was compared to AODV and AOMDV routing algorithms using NS2 simulator. The results demonstrate that DPS has high performance when compared to AODV and AOMDV. DPS performs very well when increasing the mobility of nodes. DPS also shows high scalability when increasing the number of nodes in the network.

One of the points that would give more strength to this work if it were to be done in the future is the ability of DPS algorithm to negotiate the QoS required depending on the network resources. In addition, comparing DPS to other available QoS routing algorithms would be an advantage to get a more accurate test.

REFERENCES

1. Clausen, T. and P. Jacquet, 2003. RFC 3626-Optimized Link State Routing Protocol (OLSR). IETF RFC3626,
2. Perkins, C.E. and P. Bhagwat, 1994. Highly dynamic destination-sequenced distance-vector routing (DSDV) for mobile computers. ACM SIGCOMM Computer Communication Review, 24(4): 234-244.
3. Johnson, D.B., D.A. Maltz and J. Broch, 2001. DSR: The dynamic source routing protocol for multi-hop wireless ad hoc networks. Ad hoc networking, 5: 139-172.
4. Perkins, C.E. and E.M. Royer. Ad-hoc on-demand distance vector routing. 1999. IEEE.
5. Pearlman, M.R. and Z.J. Haas, 1999. Determining the optimal configuration for the zone routing protocol. Selected Areas in Communications, IEEE Journal on, 17(8): 1395-1414.
6. Wang, Z. and J. Crowcroft, 1996. Quality-of-service routing for supporting multimedia applications. Selected Areas in Communications, IEEE Journal on, 14(7): 1228-1234.
7. Corson, M.S., 1997. Issues in supporting quality of service in mobile ad hoc networks. in IFIP International Workshop on Quality of Service (IWQoS).
8. Chakrabarti, S. and A. Mishra, 2001. QoS issues in ad hoc wireless networks. Communications Magazine, IEEE, 39(2): 142-148.
9. Birmiwal, S., *et al.*, 2012. Delay minimization in multihop wireless networks: Static scheduling does it. IEEE.
10. Leng, H., *et al.*, 2009. Routing on Shortest Pair of Disjoint Paths with Bandwidth Guaranteed. IEEE.
11. Sivakumar, R., P. Sinha and V. Bharghavan, 1999. CEDAR: a core-extraction distributed ad hoc routing algorithm. Selected Areas in Communications, IEEE Journal On, 17(8): 1454-1465.
12. Badis, H. and K. Al Agha, 2005. QOLSR, QoS routing for ad hoc wireless networks using OLSR. European Transactions on Telecommunications, 16(5): 427-442.
13. Sarma, N., S. Nandi and R. Tripathi, 2008. Enhancing Route Recovery for QAODV Routing in Mobile Ad Hoc Networks. IEEE.
14. Latha, P. and R. Ramachandran, 2009. QoS Constrained Multicast Routing For Mobile Ad Hoc Networks. IJCSNS, 9(7): 66.
15. Babri, M., O. Souleymane and G. Anoh, 2011. A multi critria QoS Routing Protocol. International Journal, pp: 3.
16. Marina, M.K. and S.R. Das, 2006. Ad hoc on-demand multipath distance vector routing. Wireless Communications and Mobile Computing, 6(7): 969-988.
17. Srinivasan, P. and P. Kamalakkannan, 2012. REAQ-AODV: Route stability and energy aware QoS routing in mobile Ad Hoc networks. in Advanced Computing (ICoAC), 2012 Fourth International Conference on. IEEE.
18. Song, Q., *et al.*, Link Stability Estimation Based on 1Link Connectivity Changes in Mobile Ad-hoc Networks. Journal of Network and Computer Applications, 2012.

19. Zhang, Z., Z. Jia and H. Xia, 2012. Link Stability Evaluation and Stability Based Multicast Routing Protocol in Mobile Ad Hoc Networks. in Trust, Security and Privacy in Computing and Communications (TrustCom), 2012 IEEE 11th International Conference on. IEEE.
20. Sridhar, S. and R. Baskaran, 2012. Conviction Scheme for Classifying Misbehaving Nodes in Mobile Ad Hoc Networks. Advances in Computer Science and Information Technology. Computer Science and Engineering, pp: 586-593.
21. Perkins, C., E. Belding-Royer and S. Das, 2003. Ad hoc on-demand distance vector (AODV) Routing.
22. Marina, M.K. and S.R. Das, 2002. Ad hoc on-demand multipath distance vector routing. ACM SIGMOBILE Mobile Computing and Communications Review, 6(3): 92-93.
23. Kaufman, C., R. Perlman and M. Speciner, 2002. Network security: private communication in a public world Prentice Hall Press.