

RESEARCH

Open Access



Load balance technique with adaptive position updates (LAPU) for geographic routing in MANETs

Santhi Venkatraman*  and Sai Kiran Sarvepalli

Abstract

In geographic routing, position updates and load distribution are crucial to achieve a good performance. There is a concern with load distribution when it comes to dense network where there are large number of connections and huge traffic. Load balanced, Adaptive Position Update (LAPU) routing protocol solves the problem of load distribution among paths in a network with reduced position updates occurring adaptively and next hop decisions based on node's congestion status in geographic routing. In our proposed work, to achieve load balancing, a node selects two best nodes available to the destination based on node mobility and queue length and it divides the load among them, i.e., it transfers the packets in both paths. Parameters such as end-to-end delay of packet transmission, energy consumption, throughput, and routing overhead are considered to compare the performance of the proposed work.

Keywords: Adaptive position updates, Algorithm design, Beacon updating, Load balancing, Routing, On demand learning

1 Introduction

Nodes in mobile ad hoc networks (MANETs) are dynamic in nature maintaining links based on the current situation around them. Unlike in the centralized network, nodes in MANET are independent.

With the increase in interest, managing a MANET has become efficient with many new routing protocols being proposed. Most of the routing protocols proposed Ad hoc On-Demand Distance Vector Routing (AODV) [1], Dynamic Source Routing (DSR) [2], and Greedy Perimeter Stateless Routing (GPSR) [3] previously used the periodic transmission of beacons to keep the neighbors updated. A beacon signal contains the information to identify the node's presence and information related to node's transmission capabilities. Transmitting beacon signals continuously consumes the energy of a node. It increases the traffic in the network. With the increase in traffic, delay in delivering the packet to a destination, packet dropping ratio, and average energy consumption also increase.

To improve the performance, there is a need to design a methodology to transmit the beacon signals adaptively based on a metric. The condition on which the situation is defined should consider the node movement and also neighboring nodes. The decision on the next hop must be taken efficiently considering the current topology and predicted future status of a node.

One such scheme is distance-based approach [4] where a beacon is transmitted whenever a node has moved a given distance called beacon distance. This distance-based approach poses a problem when the speed of nodes is different. If each node travels with different speeds, slow nodes transmit infrequently and fast nodes transmit frequently. The faster node may soon disappear. The faster nodes may not detect the slower nodes, thus reducing the connectivity. Another such scheme is speed-based approach [5]. In speed-based approach, the beacon interval and neighbor time-out interval are dependent on the speed of node. However, each node calculates its neighbor time-out interval as a multiple k of the beacon interval. Nodes send their calculated values of time-out interval in their beacon transmissions. A receiving node then determines the time-out for this neighbor which equals the minimum of the

* Correspondence: santhi@yahoo.com

Department of Computer Science and Engineering, PSG College of Technology, Coimbatore, Tamil Nadu, India

neighbors' time-out interval as indicated in the beacon and its own time-out interval calculated from its current speed. This reduces the problem in distance-based approach. Another approach in adaptive position update mechanism [6] was reactive beaconing updates. In reactive beaconing, the beacon generation is triggered by data packet transmissions. When a node has a packet to transmit, the node first broadcasts a beacon request packet. The neighbors over-hearing the request packet respond with beacons. However, in reactive beaconing, node requests a beacon transmission whenever required. It can cause redundant excessive beacon requests and transmissions in a large network. Signal stability-based adaptive routing [7] for mobile ad hoc networks is also an adaptive position update mechanism, and it depends on the signal stability to trigger a beacon update to a node's neighbors. It depends on the physical characteristics namely signal stability which varies on the environment surrounding a node.

Adaptive Position Update (APU) protocol [8] was proposed considering the limitations in previous protocols. It is mainly concentrated on position updates adaptively based on the current network topology using mobility (velocity) of nodes to trigger position updates in geographic routing. It follows techniques to transmit the beacon signals only when required based on the situation of nodes in the network. Mobility Prediction (MP) rule in APU helps in determining the mobility of a node. This routing scheme has the advantage of reducing the total overhead of beacon transmission, delay, and energy consumption. When the current situation is considered, not only the node's position but also the node's congestion state should be taken into account because the node may not be able to transfer further data if it gets congested.

Most of the routing protocols proposed previously on MANETs [9–11] included load balancing techniques based on periodic transmission techniques of beacons. Here, we proposed the load balancing for Adaptive Position Update (LAPU) routing protocol based on adaptive transmission techniques of beacons.

The primary objective of load balanced routing protocol is to distribute the load among nodes in a network to reduce congestion. To reduce the effect of higher load along a path or at a node, along with the node's congestion state, the mobility of node can be taken into account. To check the congestion of a node, we can use the queue space available at each node and the mobility of node can be measured with the help of velocity.

LAPU is a load balancing technique using both mobility of node and its congestion state, in which a source node identifies the nodes available with a queue length less than the threshold value from the replied paths. During the procedure of identifying the route from source to destination, Mobility Prediction rule identifies

the velocity of nodes and triggers adaptive beacons based on predicted position and actual position coordinates of a node. This rule (discussed in next sections) helps the nodes to maintain neighbor lists with least velocity. A least velocity, i.e., slower node, can be considered more reliable than the faster node as it stays in the vicinity of a node for more time. After identifying the routes, each node selects two nodes as next hop and divides the load among them. Our proposed work (LAPU) thus concentrates on load balancing technique for geographical routing in MANETs considering the congestion state and mobility of a node and provides a solution to overcome the problems in load balancing phenomena.

The rest of the paper is organized as follows. Section 2 briefly explains APU algorithm. Section 3 throws light upon the Ad hoc On-demand Multipath Distance Vector routing (AOMDV). This protocol has become popular because it is based on providing a multipath solution to avoid congestion or link failures in AODV routing protocol. Thus, AOMDV has been considered to compare the performances of APU and LAPU. Sections 4 and 5 provide the information about LAPU. The performance analyses of the three algorithms are given in the Section 6. Finally, the conclusion and the future work are discussed in Section 7.

2 Adaptive Position Update (APU)

The APU transmits beacons based on a simple prediction formula called Mobility Prediction (MP) rule and adaptive learning of routes using another mechanism called On Demand Learning (ODL) rule.

Instead of periodic beaconing, APU adapts the beacon update intervals to the mobility of the nodes and the amount of data being forwarded in the neighborhood of the nodes. APU employs two mutually exclusive beacon triggering rules, which are discussed in the following subsections.

2.1 Mobility Prediction (MP) rule

This rule is to express the beacon update process. The goal of this rule is to send the next beacon update from the node when the error occurred between the predicted locations. A simple location prediction scheme is used.

2.2 On Demand Learning (ODL) rule

ODL rule allows active nodes that are involved in data forwarding. This rule aims to achieve a node broadcasts beacons on-demand that is in response to data forwarding activities that occur in the area of that node. According to this rule, whenever a node overhears a data transmission from a new neighbor, it broadcasts a beacon as a response.

3 Ad Hoc On-demand Multipath Distance Vector routing

Ad Hoc On-demand Distance Vector routing (AODV) is a popular routing protocol in mobile ad hoc networks (MANETs) getting a route on only when demanded by the source node. Ad Hoc On-demand Multipath Distance Vector routing (AOMDV) [12] is a multipath version of AODV routing protocol. Having a single route can result into congestion and has significance effect on the nodes present in the single path carrying the load from source to destination. If the options are more, in choosing a route, it would reduce the effect on a single path. So, AOMDV routing protocol discovers different paths without loops based on sequence numbers while processing a route request from source to destination.

As the number of links are more, if the existing link is broken, we can choose the alternate route to avoid any delay in packet loss. The other reason is a MANET is significantly large in size. So, finding a new route if the existing link is broken increases the routing overhead and also latency in packet transmission. So, the AOMDV routing protocol in turn balances the load by selecting an alternative path if the existing link is broken.

So, the approach of balancing load in AOMDV is done when the link is broken due to link failures or congestion at nodes by keeping an alternate path as a standby. However, if the links are many, there is an option to choose alternate path. But, a congestion control mechanism is required to balance the load. This protocol has become popular because it is based on providing a multipath solution to avoid congestion or link failures in AODV routing protocol. Thus, AOMDV has been considered to compare the performances of APU and LAPU.

4 LAPU model and methodology

Load balancing approach to a MANET depicts the idea of distributing the load among different paths in the ad hoc network with node congestion as the primary parameter whereas the velocity of nodes is checked during the position updates. This section provides the example of working and complete architecture of LAPU. Here, some basic assumptions are taken: (1) all nodes know about their own position and velocity, (2) all connections are bi-directional, (3) the beacon update processes consist of the current location and velocity of the nodes, and (4) data packets can piggyback the position and velocity updates and all one-hop neighbors work in the promiscuous mode (i.e., all neighbors listen to the data to determine any control information included in the data packet itself).

In Fig. 1, the nodes S and D are the source and destination. Before the initiation of route request (RREQ) and transmission of data packets, neighbor list updates are

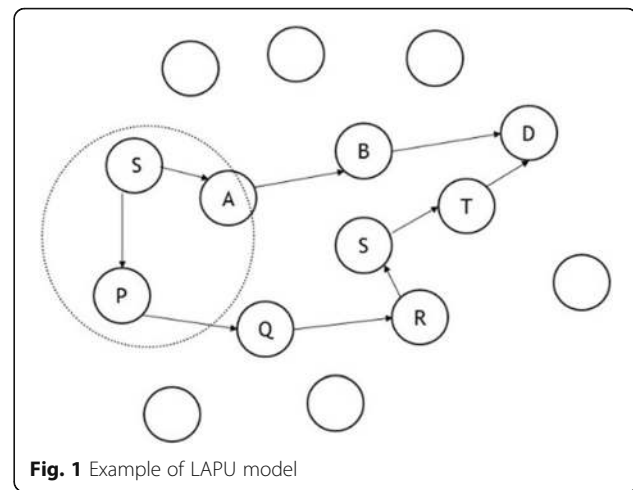
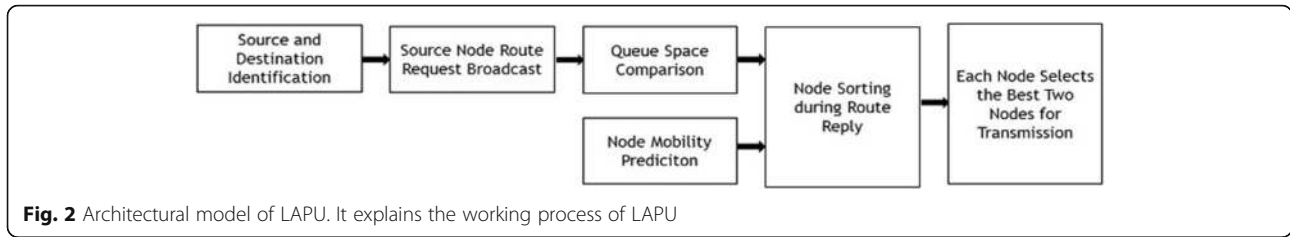


Fig. 1 Example of LAPU model

carried by each node. In the example, let us consider the nodes move with high mobility and some other nodes with low mobility. If a node is moving faster, its neighbor previously would have calculated the predicted position and determines whether it can be in its vicinity or not after a particular time instance. If the node is moving in a wrong direction, the deviation increases and threshold value is crossed. This causes triggering of beacons. The resulting routing table does not contain node with faster mobility or node deviated from its original course. This deviation parameter is calculated using the Mobility Prediction rule [8] allowing only the nodes with least velocity as the node's neighbors. Along with this rule, ODL rule [8] also contributed in making load balancing effective by maintaining a rich neighbor list. These rules are explained in following section using mathematical formulae. The nodes A and P are the neighboring nodes of S. When the source node requires a transmission to destination node D, it initiates a RREQ. It is broadcasted from source node to all of its neighbors. In this case, the route request is broadcasted to A and P which are the neighbors of S. Before the nodes are included in the routing table, source node checks the queue space of nodes, i.e., A and P in this case. Based on the queue space, algorithm decides the status of node. If the space is less than the threshold, it is congested. At the same point, neighboring nodes are also checked for their congestion status. If all the neighbors are congested, a priority queue is used for further transmission. While the routing process is going on, due to the Mobility Prediction scheme, nodes with least velocity alone are included in the neighbor lists. Once the routes are available, each node selects the two best nodes from the routing table and divides the load among them. Thus, the load on a single path is divided among two paths causing load balancing.

Figure 2 shows the model overall architecture. It shows initial identification of source and destination.



Then, node positions and neighbors are identified. After movement of nodes, each node should maintain track of its neighboring nodes and trigger an update if required using MP rule. This maintains the neighbor list with the nodes having least velocity. Simultaneously, the queue space comparison is done with the transmission of route request. Combining both the velocity of node and queue space, each node’s routing table is filled with neighbors with least velocity and least queue space. Thus, a next hop decision would be the best node in terms of velocity and congestion status. The two challenges involved in this algorithm were (1) detecting the mobility of node and selecting nodes with least mobility and (2) detecting the congestion status of nodes and their neighbors and take the decisions of next hop and usage of priority queue. The challenges were achieved using algorithms provided in the next subsections.

4.1 Mechanism of LAPU

4.1.1 Mobility Prediction (MP) rule

The goal of MP rule is sending the next beacon update when there is a change in the predicted location of a node in its neighbor’s list. Deviation factor is calculated based on current and future predicted positions based on velocity and initial position of a node. If the predicted error in the position estimate is greater than a certain threshold D_{min} , then the next beacon is broadcasted immediately. This is maintained as the deviation parameter D_{devi}^i determined by the node’s current position and velocity and future estimated position and velocity given in (1).

Table 1 contains notations required for (1). Figure 3 shows the process of updating routing table of each node frequently based on mobility.

Table 1 Notations used for mobility prediction

Variables	Definition
(X_i^j, Y_i^j)	Node i position coordinates at time T_i
V_x^i	Node i velocity at x axis
V_y^i	Node i velocity at y axis
(X_p^i, Y_p^i)	Predicted position coordinates of node i
(X_a^i, Y_a^i)	Actual position coordinates of node i

$$D_{devi}^i = \sqrt{(X_a^i - Y_p^i)^2 + (Y_a^i - Y_p^i)^2} \quad (1)$$

where T_c and T_l are current time and last beacon broadcast time and

$$X_p^i = X_l^i + (T_c - T_l) \times V_x^i \quad Y_p^i = Y_l^i + (T_c - T_l) \times V_y^i$$

Algorithm for node mobility follows as:

if $\{D_{devi}^i$ is greater than $D_{min}\}$

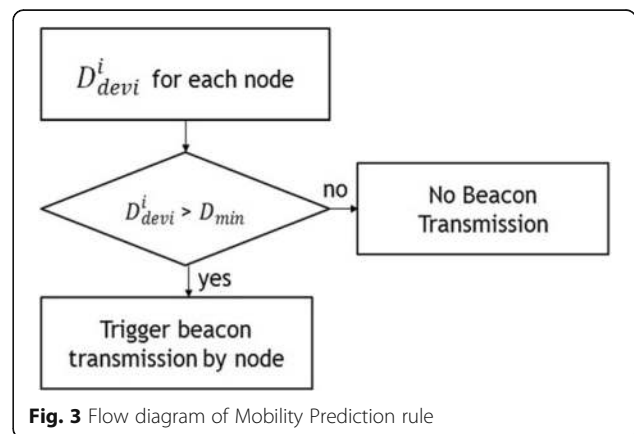
Trigger the beacon updates

Nodes with faster velocity are removed from routing table due to the action based beacon updates.

The “first challenge” of maintaining nodes only with a less velocity is thus achieved by MP rule updating routing table by removing faster nodes soon before they were considered for routing as they would be out of vicinity in less time.

4.1.2 On Demand Learning (ODL) rule

Taken alone, the MP rule may not be sufficient for maintaining accurate local topology. According to ODL rule, a new beacon is broadcasted as a response by a node when it listens to the packet transmission by any source; as a result, it is included in the neighbor list of its surrounding nodes. Along with MP rule, ODL rule gives a decent topology with nodes that can be trusted with a satisfying velocity and a rich neighbor list. Having



these advantages can provide a base for determining the load balancing situation.

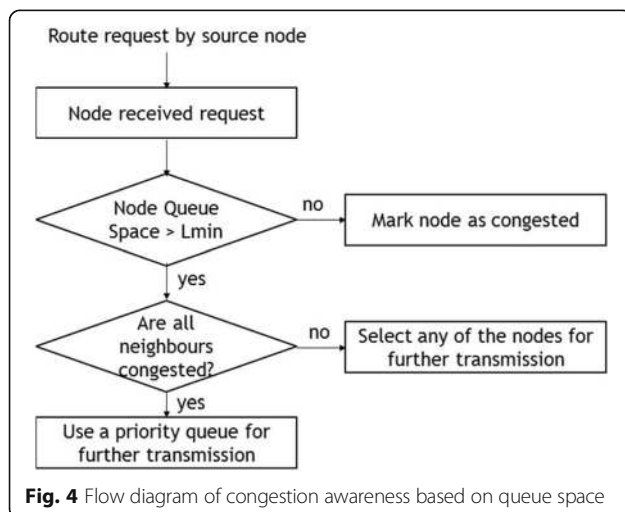
4.1.3 Node sorting based on queue space

The flow diagram given in Fig. 4 shows the process of congestion management. The broadcasted route request from source node is received by all the nodes that are near it. Any intermediate node that has the route to destination replies with a route reply to the source node. In LAPU, we use a queue for each node and also the queue space. Queue space is the remaining queue length after data packets occupying their positions in the queue.

$$Q_{\text{space}} = Q_{\text{capacity}} - Q_{\text{length}} \quad (2)$$

Each node maintains this space parameter calculated from (2) in order to determine their congestion state. Once the route request is received by any node, the routing algorithm checks for the node's queue space. If it is less than the threshold value L_{min} , it is marked as congested. It implies that the node cannot hold any more of the data. So, alternate route is searched to reach the destination. If the node is free, then its neighbors are checked for congestion state. If all the neighbors are congested, node has no alternate way to forward the route request RREQ. So, a priority queue is maintained and the RREQ is transmitted using that priority queue as there are no neighbors for forwarding the request. This is the extreme rare case in a MANET. So, our algorithm checks for least possible cases of routing in a MANET. In this manner, the route request reaches the destination or any intermediate node replies with a route reply.

Algorithm for node sorting based on queue space:
if $Q_{\text{space}} > L_{\text{min}}$.



if { RREQ is a retransmitted request AND no route to the destination }

Use priority queue for broadcast of RREQ
else

Mark the node as congested and broadcast after a random amount of time.

Route reply by a node invokes sorting of routing table of each node in the path.

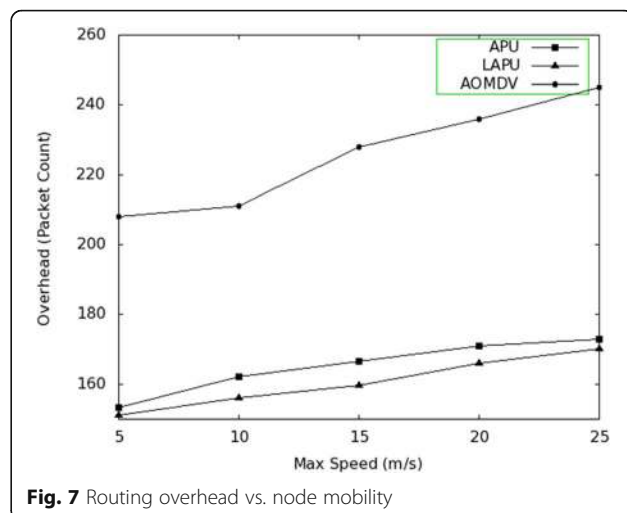
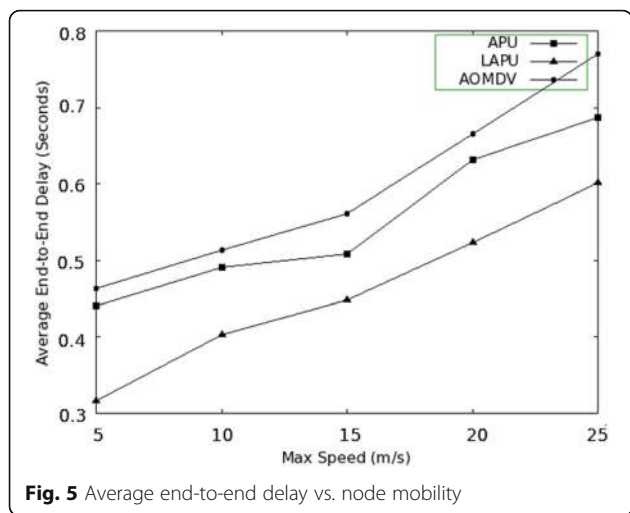
In case of a route reply, all the neighbors are sorted based on the queue space by giving preference to the nodes with least occupied queue. This provides us a congestion-free state in further transmission. Thus, the "second challenge" of identifying congested nodes is achieved using queue length. When transferring data, each node should select next hop destination to reach the final destination. A next hop is selected from the routing table of each node. But, it is already sorted based on queue space. Thus, only best node at the top is selected. Load balancing is also approached on dividing the load between two paths. So, there are two next hop decisions selected by each node. Thus, the load is transferred in two paths reducing the load on a single path. So, LAPU mechanism detects velocity using the MP rule and congestion state of a node using the algorithm based on queue space and finally achieves the load balancing phenomena.

5 Analysis of LAPU mechanism

Performance of LAPU mechanism is compared with the help of average end-to-end delay, throughput, routing overhead, and energy consumption. To analyze these factors, we consider three scenarios that can occur in a MANET using a load balanced approach. The first scenario is a source node that is having more neighbors but some are congested. In this case, the algorithm works simple, sorting the nodes based on Q_{space} . As the nodes are sorted based on the queue space, mobility is checked. Two best nodes are selected at each node as next hop decision, and load is divided among the paths. In second scenario, source node's all neighbors are congested. If all the neighbors are congested, we use a priority queue to make the data transmission. In the third scenario, the data rates are varied to check whether the algorithm can deliver the packets with ease. This case helps us to prove that data with higher transmission rate can be transferred at a less cost checking for congestion at the same time to achieve load balancing.

6 Simulation results and discussion

The simulation is carried out using NS-2 simulator tool on a Linux system. The area used for simulation is 1000×1000 . The number of nodes and queue length varies on the scenario considered. The packet size is 2000 bytes. The results are compared with the APU

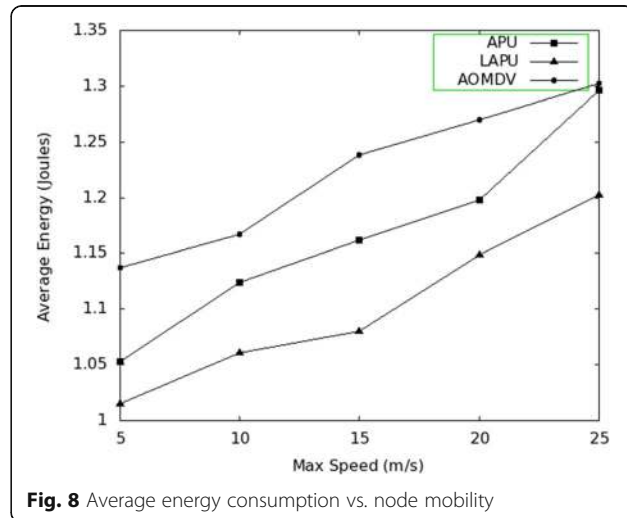
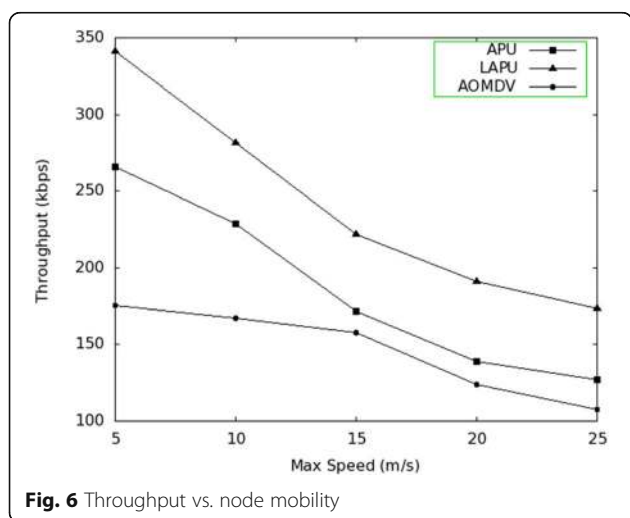


[8] and AOMDV [12] schemes. The data rate is considered to be default CBR rate of 448 kbps. Initial energy for the participating nodes is 100 J. The simulation experiments are carried out based on the three scenarios mentioned in the previous section. According to the scenario, the maximum speed, numbers of sources, and data rates are varied. The impact of algorithm on network is found out by considering parameters namely routing overhead, average end-to-end delay, throughput, and average energy consumption of the nodes. Delay includes the packet transmission delay, queuing delay, route discovery buffering, and retransmission delay at MAC layer. Load balancing improves the packet transferring method, and thus, delay is reduced. Throughput defines the efficiency of network in delivering the packets. Routing overhead defines the number of beacon packets (control packets) per data packet transmission. It should be reduced in network.

6.1 Node mobility impact

In this scenario, the simulation environment includes a total of 50 nodes and the number of sources data are 20 transmitting data to a single destination. Impact of node mobility on the performance of LAPU is compared with APU and AOMDV. Figure 5 shows the comparison of average end-to-end delay among APU, AOMDV, and LAPU. If the maximum speed is taken as 15 m/s and single node alone is considered, then the end-to-end delay is 4.09 s in the case of LAPU, 5.2 s in the case of APU, and 5.8 s in the case of AOMDV. As the node’s speed is increased, the delay is increased as shown in Fig. 5 because increase in mobility allows the node to get out from a node’s vicinity sooner. This causes frequent retransmission of packets, thus increasing the delay.

Throughput comparison shown in Fig. 6 implies the improvement of LAPU over APU and AOMDV. When a maximum speed of 15 m/s and a single node alone is considered, the link provided a throughput of 150 kbps



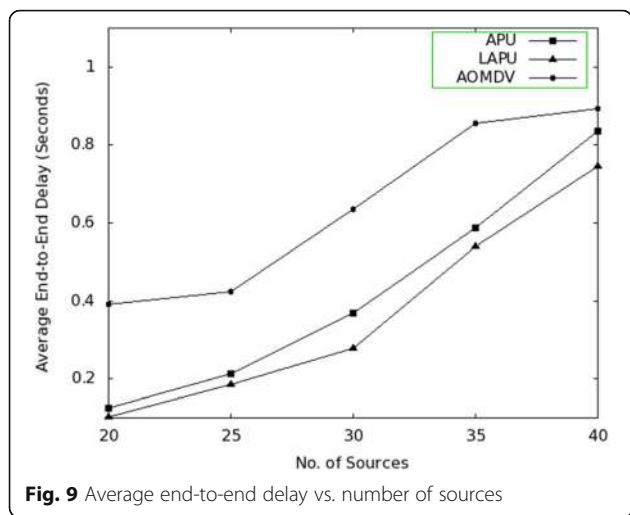


Fig. 9 Average end-to-end delay vs. number of sources

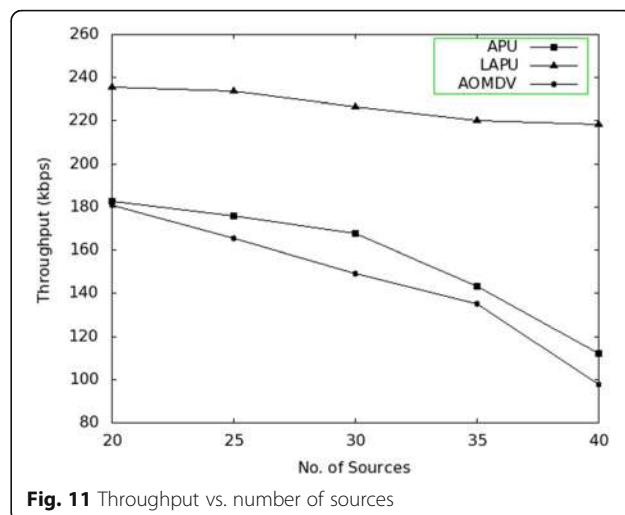


Fig. 11 Throughput vs. number of sources

in the case of AOMDV, 160 kbps in the case of APU, and 250 kbps in the case of LAPU. The performance of LAPU is better than that of the other two schemes due to it maintains accurate topology along the data forwarding path and also it shares the load in two different paths. Routing overhead shown in Fig. 7 implies that LAPU has provided a significant improvement over APU with a 15% change and 31% change over AOMDV when the maximum speed is 15 m/s. The maximum node mobility is varied from 5 to 25 m/s. Figure 7 shows that LAPU and APU can decrease the number of beacons exchanged without comprising on throughput. But AOMDV employs the periodic scheme; the routing overhead is independent of the node mobility. The average energy consumption is less when the speed is varied as shown in Fig. 8. In the case of a speed of 15 m/s, there is a 10% improvement in LAPU compared to APU and 11% when compared to AOMDV.

6.2 Impact of number of sources

The next scenario is the impact of increase in number of sources in a network. In a MANET, the traffic flow is high in real time and there are large numbers of resources competing for the resources. To achieve that, we considered a scenario with 100 nodes in which the numbers of sources are varied from 20 to 40 with an interface queue length of 10 packets and Q_{space} value of 5 to a single destination. With the increase in number of sources, the chance of congestion increases and we considered a worst case scenario with all sources pointing to a single destination. Even in this case, the LAPU scheme is proved to be working better than APU and AOMDV. The results are compared in case of previously mentioned metrics viz. average end-to-end delay, routing overhead, throughput, and average energy consumption. The average end-to-end delay of packets in case of varying number of sources is as shown in the Fig. 9. It shows

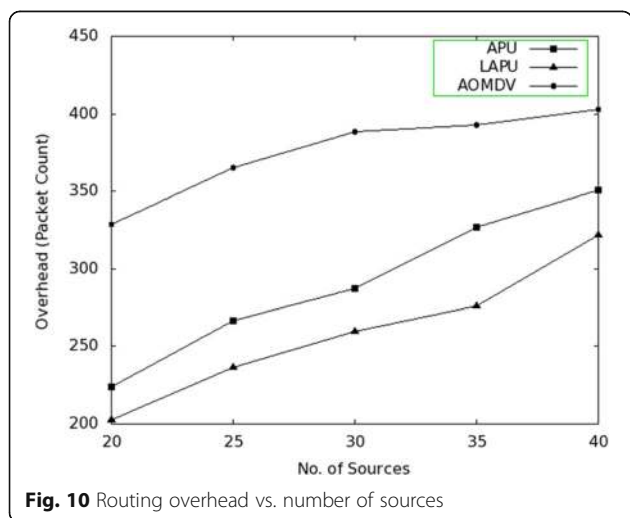


Fig. 10 Routing overhead vs. number of sources

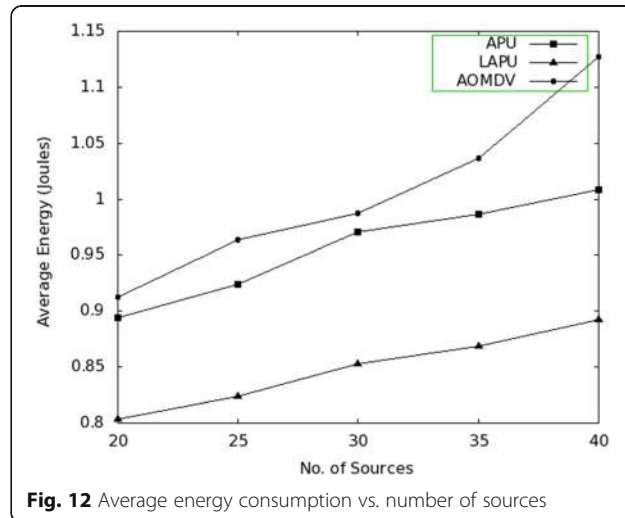


Fig. 12 Average energy consumption vs. number of sources

Table 2 Average end-to-end delay vs. data rate

Packets/s	Average end-to-end delay (s)		
	AOMDV	APU	LAPU
10	20.1215	10.0749	8.29059
15	20.2658	14.38418	10.5736
20	20.589	19.6312	16.4302

a significant improvement in the case of LAPU. When 20 sources are considered, the average end-to-end delay is 0.4 s in the case of AOMDV, 0.12 s in the case of APU, and 0.08 s in the case of LAPU. LAPU maintains accurate topology for data forwarding path, and it also divides the load. Figure 10 shows the routing overhead compared between APU and LAPU when sources are varied. As shown in the figure, LAPU has less routing overhead leading to less effect on performance of network. In this case, in case of 20 sources, LAPU is 11% better than APU and 38% better than AOMDV. Figure 11 shows the throughput when varied with the number of sources in the network. The destination is considered to be a single node in all the cases. When 20 sources are considered, APU and AOMDV are provided a throughput of 185 kbps and for the same situation, LAPU is provided a throughput of 235 kbps. LAPU throughput is better than APU since the reach ability of data to the destination is faster due to two best paths available between the source and destination. Average energy consumption shows the amount of energy used by nodes during the routing procedure in MANET. Figure 12 shows the average energy consumption in case of impact due to number of sources. LAPU is 12.5% better to APU and 13% better to AOMDV in case of 20 sources.

6.3 Impact of data rates

This scenario considers the case of varying data rates. When the data rates are low, the packet delivery takes a decent time because of less flow through a link. But if the data rates are higher, there is a high amount of load transferring through the link. Managing such load is a difficult task. LAPU considers load balancing and helps in achieving good performance. Tables 2, 3, 4, and 5 show the results compared with APU and AOMDV

Table 3 Throughput vs. data rate

Packets/s	Throughput (kbps)		
	AOMDV	APU	LAPU
10	255.5	277.666	309.951
15	286.232	355.073	407.581
20	386.85	462.392	567.185

Table 4 Routing overhead vs. data rates

Packets/s	Routing overhead (packet count)		
	AOMDV	APU	LAPU
10	428	224	202
15	459	250	219
20	477	252	229

when the data rates are changed from 10 packets/s to 20 packets/s. In Table 2, average end-to-end delay of AOMDV is increased little bit only. This is because the data forwarding has taken place with same route only. Thus, delays become insensitive to increase in data rates slowly. But in the case of LAPU, it has exact topology for data forwarding like APU and also divides the load based on queue space. If the data rates are increased, the load flowing through a link increases. This reduces its throughput as there might be packet loss. To avoid that, load on a single path should be reduced. So, LAPU divides the load among two paths. This reduces the effect on a single path. The algorithm works well with a 20% change in terms of average end-to-end delay, 11.5% improvement in throughput with very much reduced routing overhead, and average energy consumption when the data rates are considered.

7 Conclusion

The LAPU scheme uses two parameters: velocity and congestion state to balance the load in a wireless network. Velocity checking is achieved through MP rule, and congestion is checked using queue length at node. The algorithm provides a decent solution for load balancing with adaptively updated positions, making the proposed work more effective than many existing protocols. The results were compared and analyzed with APU using NS2 simulator to show that the proposed method works better in different scenarios. In terms of average end-to-end delay, LAPU produced results 25% better than APU and 30% better than AOMDV. Throughput of LAPU was 11% better when compared with APU and 15% better when compared with AOMDV. Routing overhead and average energy consumption were better with APU in all the cases showing the effectiveness of LAPU algorithm.

Table 5 Average energy consumption vs. data rates

Packets/s	Average energy consumption (J)		
	AOMDV	APU	LAPU
10	0.913659	0.894134	0.843833
15	0.923686	0.907376	0.851004
20	0.948593	0.937427	0.902013

Acknowledgements

Not applicable

Funding

Not applicable

Availability of data and materials

Not applicable

Authors' contributions

SV participated in selecting research topic, carried out the MANET studies. SKS carried out the experiment in NS2 environment and done further studies along with SV. SV and SKS participated in the creating initial manuscript. SV participated in the sequence alignment and drafted manuscript as per reviewer requirements. All authors read and approved the final manuscript.

Competing interests

The authors declare that they have no competing interests.

Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Received: 24 July 2016 Accepted: 5 March 2018

Published online: 02 April 2018

References

1. Das SR, Belding-Royer EM, Perkins CE, Ad hoc On-Demand Distance Vector (AODV) Routing, (2003)
2. D Johnson, Y Hu, D Maltz, *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. IETF RFC4728 **15**, pp. 1–107 (2007)
3. B Karp, HT Kung, in *ACM MobiCom. GPSR: greedy perimeter stateless routing for wireless networks*, pp. 243–254 (2000)
4. S Basagni, I Chlamtac, V Syrotiuk, B Woodward, in *ACM/IEEE MobiCom '98. A distance routing effect algorithm for mobility (DREAM)*, pp. 76–84 (1998)
5. M Heissenbuttel, T Braun, M Walchli, T Bernoulli, Evaluating the limitations of and alternatives in beaconing. *Ad Hoc Net* **5**, 558–578 (2007)
6. Q Chen, SS Kanhere, M Hassan, KC Lan, in *Proc. Int'l Conf. Comm. (ICC '06)*. Adaptive position update in geographic routing (2006), pp. 4046–4051
7. Rogit Dube, Cynthia D Rais, Kuang-Yeh Wang, Satish K, Signal stability-based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Pers. Commun.* **4**, 36–45 (1997)
8. Q Chen, SS Kanhere, M Hassan, Adaptive position update for geographic routing in mobile ad hoc networks. *IEEE T Mobile Comput* **12**, 489–501 (2013)
9. H Hassanein, A Zhou, *Routing with Load Balancing in Wireless Ad-Hoc Networks, 4th ACM International Workshop on Modeling, Analysis and Simulation of Wireless and Mobile Systems* (2001), pp. 89–96
10. Y Lee, G Riley, A workload-based adaptive load-balancing technique for mobile ad hoc networks, wireless communications and networking conference. *IEEE* **4**, 2002–2007 (2005)
11. AS Varghese, SC Jebakumari, in *International Conference on Communications and Signal Processing (ICCSP)*. Dynamic Beacon Based and Load Balanced Geo Routing in MANETS, pp. 682–686 (2014)
12. MK Marina, SR Das, Ad hoc on-demand multipath distance vector routing. *Wirel Commun Mobile Comput* **6**, 969–988 (2006)

Submit your manuscript to a SpringerOpen[®] journal and benefit from:

- Convenient online submission
- Rigorous peer review
- Open access: articles freely available online
- High visibility within the field
- Retaining the copyright to your article

Submit your next manuscript at ► springeropen.com
