Asian Journal of Information Technology 15 (3): 376-383, 2016
ISSN: 1682-3915


# A Novel Routing Algorithm for Selfish Node Detection for Optimal Data Transmission in Mobile Ad Hoc Network

[1]S. Senthilkumar and [2]J. William
[1]Department of CSE, University College of Engineering, Pattukottai, Tamil Nadu, India
[2]Department of IT, M.A.M. College of Engineering, Tiruchirappalli, Tamil Nadu, India

**Abstract:** A Mobile Ad hoc Network (MANET) is an infrastructure less but capable of self configuring network that contains wide range of wireless devices that can move freely in any direction and each device can act as a router or end host. The Mobile Ad Hoc network is too big and the communicating nodes can be out of range. So, each node in MANET can act as router and forward the data to neighbour and the data reach the destination. The relay traffic consumes the resources and power. But, the power and the resources are very limited, so each node needs to conserve them for its own purpose. The presence of selfish nodes may affect Quality of Service (QoS) especially reliability and data transmission. A demand arises to detect the selfish nodes and neglect them to improve the quality of service of a MANET. In this study, an improved AODV algorithm is proposed to detect the selfish nodes and a new routing mechanism is proposed for traffic relay. The proposed algorithms are simulated and they showed better quality of service by enhancing packet delivery ratio and reducing packet delay.

**Key words:** MANET, selfish nodes, reliability, QoS, AODV, packet delivery ratio, packet delay

## INTRODUCTION

Mobile Ad-Hoc Network (MANET) is a Multi-Hop network that is composed of multiple mobile nodes connected through wireless links but truly without fixed or any existing infrastructure. Despite of many advantages of MANET, the limitations are also there like minimum memory, limited battery power and bandwidth and cooperative behaviour of neighbouring nodes.

Cooperative behaviour is the biggest concern in MANET as the mobile nodes tend to save their battery power, bandwidth and memory for its own purpose. The battery power is the one of the most wanted resource, so each node tries to conserve it as much as possible and atleast for its own purpose. The success of MANET depends on how other nodes cooperate to each other to transmit or forward the data to destination. The data transmission consumes more battery power and some of the nodes deliberately do not want to participate in data transmission to save its power. These nodes are termed as misbehaving nodes and they cause network partitioning. These misbehaving nodes are termed as selfish nodes. The performance of Mobile Ad-Hoc network can be hugely affected by some of misbehaving nodes and selfish nodes.

The success of MANET relies on improving the performance of mobile nodes. The performance can be improved by identifying selfish nodes and isolating them. The selfish node normally doesn't reply to any probe messages to hide its existence and make it very difficult to identify it. Sometimes, they involve in changing TTL time to minimum value and causes the transaction doesn't occur.

The motivation of the proposed algorithm is to find selfish nodes in the route to destination. The trust value (Pirzada *et al*., 2006) represents the level of cooperativeness of each node in the route. The node's trust value is calculated by the number of transactions in which the node has involved. The node, whose trust value is maximum, is selected as trust manager.

The trust manager is a mobile node which has the highest trust value and the trust manager is given full freedom to select the route for the communication and isolate the selfish node if any. The communication starts with the request that is sent by the sender to trust manager.

The trust manager reads the message that contains source ID, destination ID, message content, its trust value. The trust manager finds the paths to reach destination. Each node's trust value is calculated by

**Corresponding Author:** S. Senthilkumar, Department of CSE, University College of Engineering, Pattukottai, Tamil Nadu, India

summing up the values of battery power, bandwidth and mobility. The trust value of each node in the route is summed up to calculate trustiness of the route. The most trusted route is selected for transmission and blocked for certain time. The trust manager sends message to source node to start communication using the blocked route.

The trust manager sends the probe message to each node in the route and waits for reply message. Sometimes, the node wouldn't participate by not sending the reply message or the reply message may be lost due to congestion. In these circumstances, the trust manager asks for trust value of that node from (Biradar and Manvi, 2012) its neighbour. The trust value of that node reveals about its cooperativeness or selfishness. The trust manager considers the node if the trust value is good, i.e., cooperative. Otherwise, the node is given the message to change its attitude to cooperative. The message is sent for thrice and the node is not considered if it doesn't change its attitude.

**Literature review:** The following are some of the proposed techniques for misbehaving nodes detection in MANET found in the literature. Marti *et al.* (2000) proposed the Watchdog and Pathrater techniques for detecting and mitigating routing behaviour. Watchdog mechanism is responsible for detecting misbehaviour node in the network by promiscuously listening to its next hop's transmission. Thus by listening to its neighbours, a node can detect whether packets sent to its neighbour for forwarding have been successfully forwarded by its neighbour or not. If a neighbour repeats any misbehaviour more times than a predefined threshold value, it is considered as misbehaviour node in the MANETs. In this case, the pathrater cooperates with the routing protocols (Song and Zhang, 2010) to avoid the misbehaviour nodes in future transmission. Even though, watchdog is capable of detecting malicious nodes rather than links, it suffers from ambiguous collisions, receiver collisions, limited transmission power, false misbehaviour report, collusion, partial dropping and listening mode adopted by the watchdog is not appropriate for a MANET because the watchdog can work only when links are bidirectional. In practical, many unidirectional links may exist in MANETs due to the topology control.

CONFIDANT (Cooperation of Nodes: Fairness in Dynamic Ad Hoc Networks) is reputation-based solution, presented by Buchegger and Boudec (2002). It aims to detect and isolating a misbehaving node. Each mobile node consists of four components namely monitor, trust manager, reputation system and path manager. Monitor component watches not only transmission of neighbouring nodes but also observes routing behaviour. Trust manager make use of information from the monitor technique and evaluates the trust level of each mobile node. Maintaining a local rating list and a black list, the reputation system point out misbehaviour nodes to be avoided for routing. Finally, the path manager ranks paths according to the reputation of nodes along the path and deletes paths containing malicious mobile nodes. However, the monitor cannot exactly tell misbehaviour from mere coincidence such as collision and the scalability is also a problem.

Michiardi and Molva (2002) proposed CORE (Collaborative Reputation Mechanism), a mechanism based on reputation to detect selfish nodes (Buchegger and Boudce, 2002) and enforce cooperation among them. The limitation with CORE is that the most reputed nodes may become congested as most of the routes are likely to pass through them. Also, the limitations of the monitoring system in networks with limited transmission power and directional antennas have not been addressed in CORE.

The major problem of the reputation systems of both CORE and CONFIDANT is that they always (regardless of the behaviour of nodes) require periodic packet exchanges, resulting in an important overhead.

He *et al.* (2004) proposed Secure and Objective Reputation-based Incentive (SORI) approach for encouragement of packet forwarding and discipline selfish behaviour using reputation based punishment mechanism. Reputation value of a node based on packet forwarding ratio of nodes. It has three modules for neighbour monitoring, reputation propagation and punishment. Neighbour (Fiore *et al.*, 2013) monitoring system is responsible for to collect information about packet forwarding behaviour of neighbours that is node N keeps count of number of packets sent by node N to the node X for forwarding, called RFN(X) (request for forwarding) and number of packets actually forwarded by node X for node N, called HFN(X). Reputation of a node is computed using these values. Trust value of a node is directly proportional to number of packets forwarded through the node (RFN(X)). Trust value is used to give high priority to the reputation value received from the node.

Reputation propagation system is responsible for communicating reputation of nodes among neighbours when there are significant changes in reputation of some node's. One way hash chain is used for authentication of reputation information messages and data packets.

Punishment system is responsible for deciding the probability of dropping packets of a misbehaving node in proportion of its selfish. The merits of the scheme are computationally efficient as compared to other methods and it reduces the communication overhead. It fails to differentiate between malicious and selfish nodes. It also has poor performance in the case of cooperation node.

The existing techniques suffer from several bottlenecks such as ambiguous collisions, receiver collisions, unidirectional links, partial dropping and limited transmission power. The main issue is that the event of successful packet reception can only be accurately determined at the receiver of the next-hop link but the watchdog technique only monitors the transmission from the sender of the next-hop link. To overcome this problem in TWOACK scheme (Balakrishnan *et al*., 2005) focuses on the problem of detecting misbehaving links instead of misbehaving nodes. TWOACK scheme detects misbehaving link and then seeks to alleviate the problem of routing misbehaviour (Liu *et al*., 2007) by notifying the routing protocol to avoid them in future routes. It is done by sending back a TWOACK packet on successful reception of every data packet which is assigned a fixed route of two hops in the direction opposite to that of data packets. Basic drawback of this scheme includes it cannot distinguish exactly which particular node is selfish node. So, normal nodes became part of misbehaving link and therefore cannot be further used the network and it will cause the traffic congestion on the network.

## MATERIALS AND METHODS

**Trust manager election:** MANET contains number of mobile nodes and each node needs the support of other nodes to transmit the information to destination. But, some of the nodes may misbehave by not cooperating with source node's transaction to save its resources. So, there is a high demand of a mechanism to find the intermediate node is cooperating or not. There are number of research works carried out to detect the selfish node (Roy and Chaki *et al*., 2011) such as intermediate node responds to probe message. But, those mechanisms fail when any one of the intermediate node is misinterpreting the acknowledgement. So, we need some special way to avoid the security issue and to detect the selfish node. We propose Trust Manager Node (TMN) is a node which has the highest trust value and is responsible for successful transaction (Wang *et al*., 2011).

In MANET, a new algorithm is proposed to elect the trust manager node. The node whose trust value is maximum will be elected as trust manager Node (Liu *et al*., 2011). The trust value represents number of transactions in which the node shows its cooperativeness. Initially, the trust value of any node is zero and the value is incremented by one whenever the node involves in data transaction.

Sometimes, the elected trust manager may be down or the link may be failure. When source node tries to send the data to destination, it sends the request to trust manager. If trust manager is active, it sends wait message to source. Source will wait for the route information from trust manager.

If the trust manger is down, it will not send wait message to source. The source remains inactive until TTL and retransmits the request again. When the source doesn't get any message then it decides that the trust manager is down and initiates the Trust Manager Election algorithm. The source sends the elect probe message to all the nodes. On receiving electrode message, each node sends its own id, trust value. Once, the source gets the reply from all nodes, it checks for the node whose trust value is maximum. The maximum trust value holder will be elected as trust manger. The source will send the intimation to newly elected trust manager and broadcast the newly elected trust manager id to all the nodes.

**Selfish node detection:** Whenever, the source wants to carryout any transaction, it sends the request to trust manager. The trust manager sends behaviour probe message to the nodes in the route to destination. Such as probe message each node checks the availability of power, mobility and other resources. The node sends the reply message. The reply message contains the node id, availability of resources and number of transaction involved by it.

The trust manager compares the availability of resource with threshold value. If the node satisfies the criteria, the node is said to be cooperative. If the node fails to meet the criteria, the node is said to be selfish.

The trust manager checks all the intermediate nodes in the route are cooperative or selfish. When the node acts as selfish, the trust manager sends the warning message for thrice to change its behaviour. Even then, the node remains in its misbehaviour, the node will be sent to block listed nodes and those nodes will never be considered for any future transaction.

**Algorithm 1: Selfish node detection**

Input: Set of mobile nodes n, Trust Manager TM, data.
Start
STEP 1: Sender sends REQ message to Trust manager.

STEP 2: If TM is alive, it sends "WAIT" message immediately to Sender.
Else Sender waits until TTL and retransmits REQ.
STEP 3: Sender waits as it gets "WAIT" else initiates Trust Manager Election GOTO Election
STEP 4: Trust Manager finds the routes to destination
STEP 5: Trust Manager sends Behaviourprobe message to all the intermediate nodes in the route.
STEP 6: If the intermediate node is interested to take part in transmission, it sends the Reply.
Else it won't reply or defer reply GOTO Warning.
STEP 7: Trust Manager analyses each node's reply that contains its resources level.
STEP 8: Trust Manager calculates node's ability to forward the data by summing up node's power, Memory, Processor utilization, mobility, bandwidth and interference.
STEP 9: Trust Manager sums up the trustiness of all the nodes in the route and selects the most trustful and cooperating route.
STEP 10: TM blocks the route for the transaction.
STEP 11: Trust Manager sends "READY" message to Sender.
STEP 12: Sender sends the data to first node in the route.
STEP 13: Each node increments its trust value by 1 after forwards the data to next neighbour and the data reach the destination.
STEP 14: The Receiver sends the "Complete" message to Trust Manager on completion of data transmission.
STEP 15: Trust Manager unblocks the route for future transactions.
Stop.

**Algorithm 2: Election**

STEP 1: Sender initiates the election.
STEP 2: Sender send "Elect Probe" message to all the nodes.
STEP 3: Each node sends its id and trust value to sender.
STEP 4: Sender confirms that it gets the reply from all.
STEP 5: Sender elects the Trust Manager whose Trust value is maximum among all the nodes.
STEP 6: Sender broadcasts Trust Manager id.

**Route information and data delivery:** The trust manager confirms all the intermediate nodes in the route are cooperative. The trust manager sends route information to the source and the source will send the data to the first intermediate node in the route.

When the trust manager finds that any one of the intermediate node is selfish in the route, the trust manager tries another route to reach destination where all the intermediate nodes are cooperative.

If two or more routes are available, the trust manager calculates the cooperativeness of the route by summing up the cooperativeness of each node in the respective route. The highest cooperativeness route is selected for the transmission.

In Fig. 1, trust manager is the node who is taking decision how to reach the destination effectively. When the sender needs to transmit any data to any node, it cannot simply send the data to the neighbouring node. The sender initiates the transmission with REQ message to trust manager. The trust manager checks the behaviour of all the intermediate nodes. The trust the behavioural manager collects

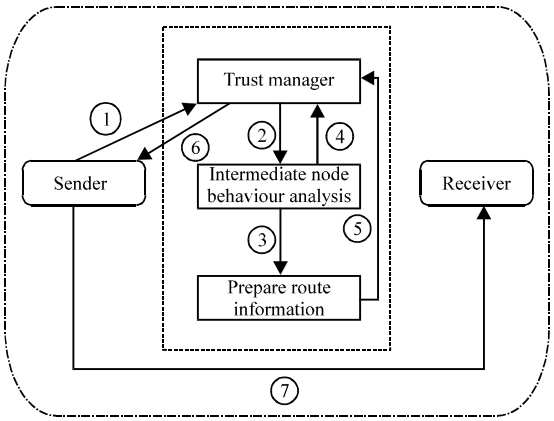


Fig. 1: Architectural diagram

report from all the intermediate nodes and prepares the best route and blocks the route for the transaction.

Trust manager sends the route info to the sender. The sender now sends the data first neighbouring node in the route. On receiving the data, the receiver sends END message to trust manager. Trust manager unblocks the route for future transactions.

**Implementation:** The cooperativeness of a node is calculated by analysing the current energy level, distance and mobility status, interference, memory and processing power of the node.

**Energy level:** Consider a node n with Full battery Energy (FE) and the Available battery Energy at any time is represented by (AE).

At time t = 0, no transaction is taken place, the node doesn't involve in any operation and maintains full battery energy. Consider, a packet is transmitted between time t and t+1. The energy that is used to transmit a bit is (BE) and N is the number of bits transmitted between the time slots t and t+1. The node also performs certain operations to transmit the bits. Let us take the energy that is required for those operations are OE. The available energy at time t+1 is calculated by:

$$AE_{t+1} = AE_t - ((BE \times N) - OE) \quad (1)$$

The available energy is calculated by subtracting the energy needed for transmitting number of bits and other operations from current energy. AE can be insufficient or sufficient. AE can be insufficient if the value is less than the threshold. AE can be sufficient and take part in transmission if the value is greater than threshold value.

**The distance:** The nodes m and n are positioned at (a1, b1) and (a2, b2), respectively. The nodes m and n have different mobility with respect to their speed and direction.

Let us consider the nodes m and n move to x1 and y1 distances and reach new positions (a1', b1') and (a2', b2'). At time t = 0, the distance between the nodes m and n is calculated as:

$$R_{mn} = SQRT ((a1-a2)^2 + (b1-b2)^2) \quad (2)$$

where, $R_{mn}$ represent the remoteness between m and n nodes. The node m takes t time and s1 speed to move to x1 place and the node n takes t time and s2 speed to reach y1 place. The distance moved by them can be calculated by:

$$R1 = s1 \times t \quad (3)$$

Where, R1 is the remoteness of node m:

$$R2 = s2 \times t \quad (4)$$

Where, R2 is the remoteness of node n. At time t, node m becomes R1 remoteness where a1' and b1' can be calculated by:

$$a1' = a1 + R1 \quad (5)$$

$$b1' = b1 + R1 \quad (6)$$

The remoteness of node n is calculated by:

$$a2' = a2 + R2 \quad (7)$$

$$b2' = b2 + R2 \quad (8)$$

At time t, the distance between the nodes m and n is calculated as:

$$R_{mn} = SQRT ((a1'-a2')^2 + (b1'-b2')^2) \quad (9)$$

where, $R_{mn}$ represent the remoteness between m and n nodes. The available memory size and processing power are calculated and the ability to carry the data is also analysed.

**Selection of route:** The cooperativeness of a node in the route can be calculated by summing up the values of different parameters discussed as above. C(N) defines cooperativeness of a node N, $R_{mn}$ represents mobility, AE is available energy, AM represents available memory and AP states available processing power. Then, C(N) is calculated as follows:

Table 1: Simulation parameters

| Simulation parameters | Values |
|---|---|
| Simulator | NS-2 (v.2.34) |
| Simulation area | 1000×1000 |
| Number of nodes (x) | 50 |
| Transmitter range | 250 |
| Bandwidth | 2 MB |
| Packet size | 1000 bytes |
| Buffer length ($MS_s$) | 50 packets |
| Traffic type | CBR |
| Simulation time | 100 sec |
| Receiver energy ($R_x$) | 0.01 |
| Transmitter energy ($T_x$) | 0.02 |
| Initial energy ($E_i$) | 100j |
| Routing protocol | Improved AODV |
| Propagation model | Two way ground |
| Antenna type | Omni directional |
| MAC type | 802_11 |
| Mobility model | Random way point model |

$$C(N) = R_{mn} + AE + AM + AP \quad (10)$$

The cooperativeness of a route can be calculated by summing up the cooperativeness of each node in the route. C(R1) is the cooperativeness of a route R1 and C(N1) is the cooperativeness a node N1:

$$C(R1) = C(N1) + C(N2) + \ldots \quad (11)$$

The highest cooperativeness route is elected for transmission.

**Performance evaluation:** The proposed approach was evaluated using Network Simulator 2 (NS2) version 2.34. The random waypoint mobility model is used to determine the mobility of nodes. The parameters used in the simulated are enlisted in Table 1. The proposed scheme is measured and compared the performances with AODV measurements.

## RESULTS AND DISCUSSION

**Packet Delivery Ratio (PDR):** It defines the ratio of the number of packets received by the destination node to the number of packets actually sent by the sender node. It defines the ratio of the number of packets received by the destination node to the number of packets actually sent by the sender node. We evaluate the packet delivery ratio performance with varying parameters like Nodes, Interval, Packet size, Pause time and Speed. Figure 1 shows the effects of the PDR on the Improved AODV. The simulation graph insists that Improved AODV shows better packet delivery ratio than conventional AODV algorithm. The simulation study clearly reveals that even the parameters such as nodes, packet size, interval time,

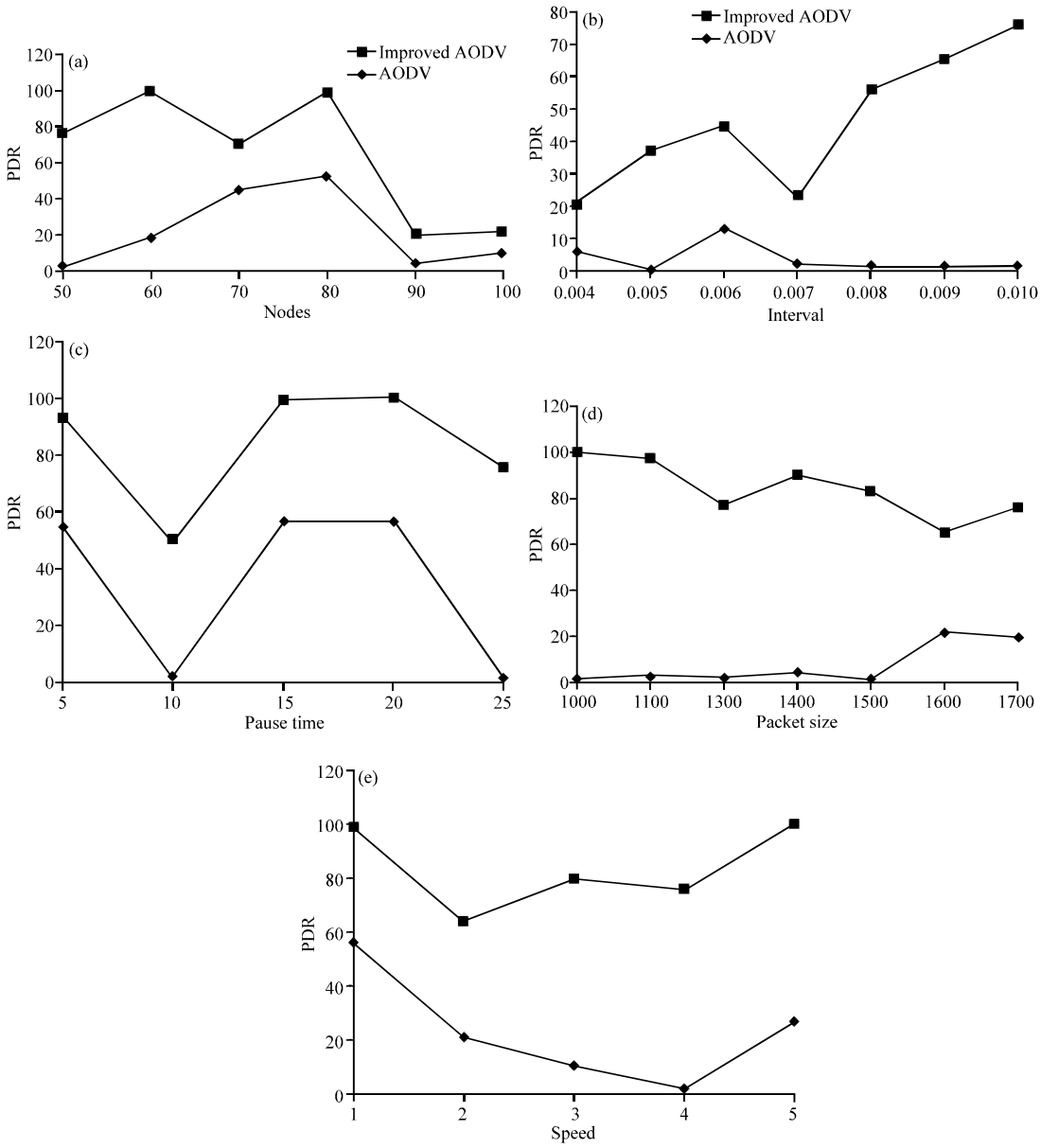


Fig. 2: Comparison results between the AODV and the Improved AODV: a) Nodes; b) Interval; c) Pause time; d) Packet size and e) Speed

speed and pause time are increases with the Improved AODV delivers more number of packets than existing AODV.

**Packet delay:** It is defined (Biradar and Manvi, 2002) as the average time taken to transmit the predefined number of packets from source to multicast destinations for various group sizes.

We examine the packet delay performance with different parameters like nodes, interval, packet size, pause time and speed. Figure 2 and 3 shows the effects of the packet delay on the improved AODV. The simulation graph insists that improved AODV reduces the packet delay than conventional AODV algorithm. The simulation study clearly reveals that even the parameters such as nodes, packet size, interval time, speed and pause time are increases with the improved AODV taken less time to transmit the number of packets to destination than existing AODV.

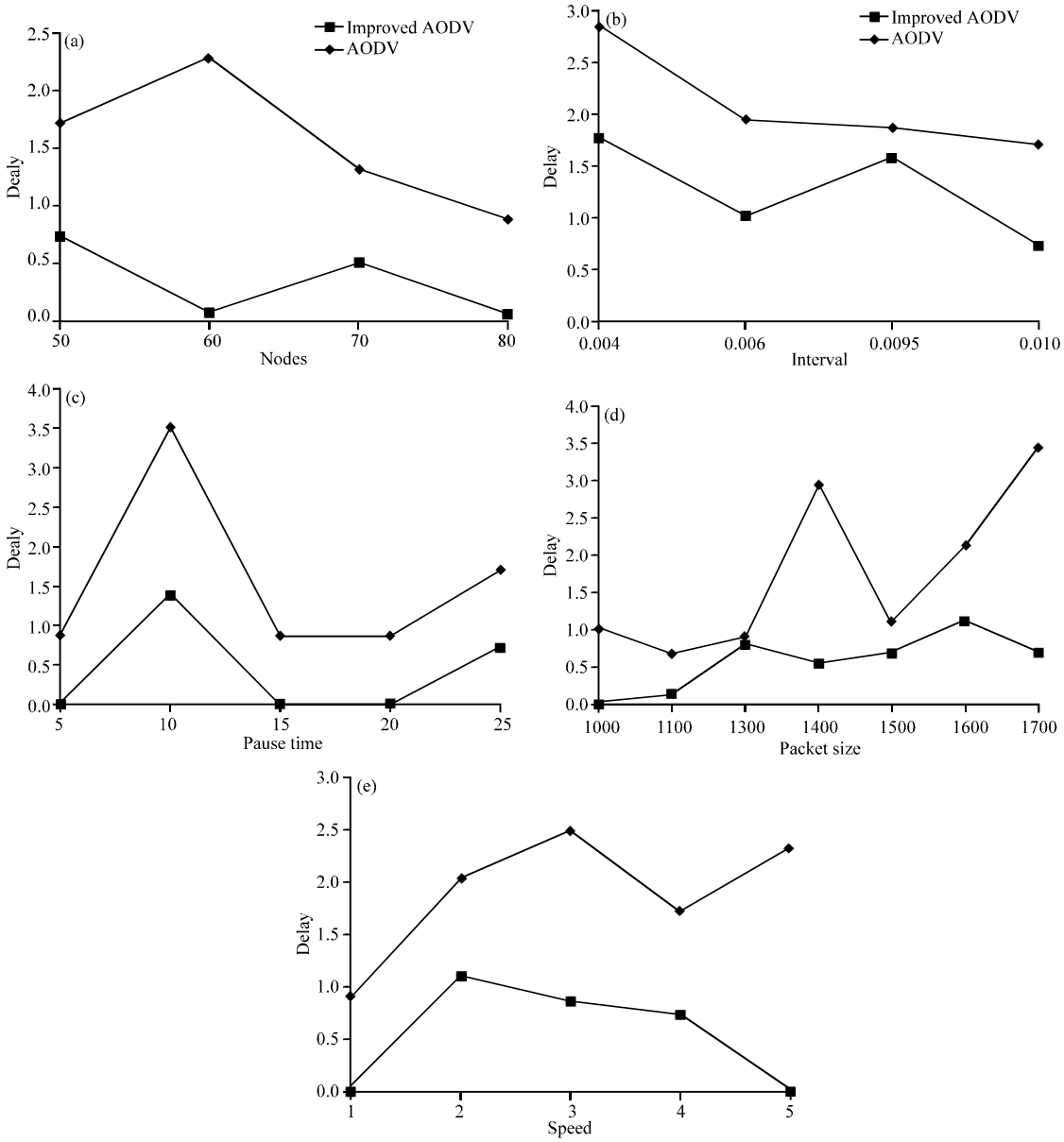


Fig. 3: Comparison packet delay results between the AODV and the Improved AODV: a) Nodes; b) Interval; c) Packet size, d) Pause time and e) Speed

## CONCLUSION

The success of MANET heavily depends on how the nodes are cooperating to each other to forward the data to the destination. The success of MANET may be degraded by the presence of selfish nodes. In this study, we proposed an innovative approach to find selfish nodes. The parameters like battery power, memory, mobility, interference and, etc. are used to find the selfishness of a node. A new algorithm is proposed to find the route to destination. The punishment mechanism is carried out by the trust manager node to change the selfishness attitude of a node. To avoid the chaos when the trust manager is failure, the election algorithm is also proposed. The simulation is carried out by comparing conventional AODV and new algorithm and results are compared. The results show that new algorithm gives better results than conventional AODV in terms of packet delivery ratio and packet delay etc.

## REFERENCES


Balakrishnan, K., J. Deng and P.K. Varhney, 2005. TWOACK: Preventing selfishness in mobile ad hoc networks. Proceedings of the IEEE Wireless Communications and Networking Conference, Volume 4, March 13-17, 2005, New Orleans, LA., USA., pp: 2137-2142.

Biradar, R.C. and S.S. Manvi, 2012. Neighbor supported reliable multipath multicast routing in MANETs. J. Network Comput. Appli., 35: 1074-1085.

Buchegger, S. and J.Y. Le Boudec, 2002. Performance analysis of the CONFIDANT protocol. Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, June 9-11, 2002, Lausanne, Switzerland, pp: 226-236.

Fiore, M., C.E. Casetti, C.F. Chiasserini and P. Papadimitratos, 2013. Discovery and verification of neighbor positions in mobile ad hoc networks. Mobile Comput. IEEE. Trans., 12: 289-303.

He, Q., D.P. Wu and P. Khosla, 2004. SORI: A secure and objective reputation-based incentive scheme for ad-hoc networks. Proceedings of IEEE Wireless Communications and Networking Conference, Mar. 21-25, Atlanta, GA, United States, pp: 825-830.

Liu, K., J. Deng, P.K. Varshney and K. Balakrishnan, 2007. An acknowledgment-based approach for the detection of routing misbehavior in MANETs. IEEE Trans. Mobile Comput., 6: 536-550.

Liu, Y., K. Li, Y. Jin, Y. Zhang and W. Qu, 2011. A novel reputation computation model based on subjective logic for mobile ad hoc networks. Future Gener. Comput. Syst., 27: 547-554.

Marti, S., T.J. Giuli, K. Lai and M. Baker, 2000. Mitigating routing misbehavior in mobile ad hoc networks. Proceedings of the 6th Annual ACM/IEEE International Conference on Mobile Computing and Networking, August 6-11, 2000, Boston, MA., USA., pp: 255-265.

Michiardi, P. and R. Molva, 2002. CORE: A collaborative reputation mechanism to enforce node cooperation in mobile ad hoc networks. Proceedings of the IFIP TC6/TC11 6th Joint Working Conference on Communications and Multimedia Security, September 26-27, 2002, Portoroz, Slovenia, pp: 107-121.

Pirzada, A.A., C. McDonald and A. Datta, 2006. Performance comparison of trust-based reactive routing protocols. IEEE Trans. Mobile Comput., 5: 695-710.

Roy, D.B. and R. Chaki, 2011. MADSN: Mobile agent based detection of selfish node in MANET. Int. J. Wireless Mobile Netw., 3: 225-235.

Song, C. and Q. Zhang, 2010. Protocols for stimulating packet forwarding in wireless ad hoc networks security and privacy in emerging wireless networks. Wireless Commun. IEEE., 17: 50-55.

Wang, J., Y. Liu and Y. Jiao, 2011. Building a trusted route in a mobile ad hoc network considering communication reliability and path length. J. Netw. Comput. Appl., 34: 1138-1149.