Journal of Computer Science 2012, 8 (11), 1914-1923
ISSN 1549-3636
© 2012 Science Publications
doi:10.3844/jcssp.2012.1914.1923 Published Online 8 (11) 2012 (http://www.thescipub.com/jcs.toc)

Random Dummy Packet Distribution Approach for Detection of Routing Misbehavior in Mobile Ad Hoc Network

[1]T. Sakthivel, [2]R.M. Chandrasekaran and [2]S. Vijay Bhanu

[1]Manonmaniam Sundaranar University, Thirunelveli, Tamilnadu, Pin-627012, India
[2]Department of CSE, Annamalai University, Chidambaram, Tamil Nadu, Pin-608 002, India

Received 2012-05-23, Revised 2012-09-19; Accepted 2012-11-06

ABSTRACT

A Mobile Ad hoc Network (MANET) works on the assumption that mobile nodes are cooperative. On the other hand, in practical, it may not be possible due to selfish, malicious and misbehaving nodes present in the network. These nodes are capable of interrupting the communication process and there is a potential for serious performance degradation. To improve the performance of the network, detection and elimination of node misbehavior is paramount. This study presents a new technique called Random Dummy packet Distribution (RDD) approach that can be used as an extension scheme in Dynamic Source Routing (DSR) protocol to recognize and exclude misbehaving nodes. The key objective of RDD approach is mobile nodes generate dummy packets randomly and forward to the destination node. It receives the acknowledgement only if there is no misbehaving node available in its root path. If it found any critical path, a trust factor evaluation used to prove its misbehavior. Finally, Information about node misbehavior has shared with all other participants in the network. To reduce the network traffic and routing overhead, RDD uses the random dummy packets for routing acknowledgement. The results obtained from simulation proved that, the RDD approach can lessen the impact of misbehaving nodes and hence, improve the performance of routing protocol. RDD is an add-on method for routing protocol in order to eliminate the misbehaving nodes and the performance analysis proves the effectiveness of RDD approach over other schemes.

Keywords: Mobile Ad Hoc Networks (MANET), Random Dummy Packet Distribution (RDD), Dynamic Source Routing (DSR), Node Misbehavior, Secure Routing, Trust Evaluation

1. INTRODUCTION

Mobile ad hoc Network (MANET) (Chlamtac *et al*., 2003) is a self-organized wireless network comprising a group of mobile nodes that can move anywhere in the network. MANET does not rely on a centralized administration. Node mobility causes dynamic change in the network topology and frequent unannounced disconnections. Mobile nodes communicate directly if they are within the communication range of each other. A mobile node cannot have single hop communication with the destination, due to factors such as the low transmission range and channel utilization. Therefore, MANET depends on intermediate nodes to forward messages to the destination. It relies only on multi-hop communication pattern. To maintain a cooperative MANET, all the participating nodes in the network should be friendly and willing to forward a message to other nodes without being selfish. In a wide range of applications, users with varying intentions allocate the resources that establish complete connectivity. MANET is vulnerable to attacks due to the open, cooperative and dynamic nature and needs a new method for secure communication (Zhou and Haas, 1999).

Corresponding Author: T. Sakthivel, Manonmaniam Sundaranar University, Thirunelveli, Tamilnadu, Pin-627012, India

1.1. Secure Routing

Nature of MANET brings new opportunities and challenges. It is vulnerable to a channel and physical attacks due to broadcasting nature. Attacks can be easily mounted due to vulnerability of the channel, a lack of centralized authority and dynamic topology change (Papadimitratos and Haas, 2002). Routing protocols designed for MANET is capable of managing dynamic network changes. Secure authentication and routing optimization technique is discussed in. Most of the routing protocols assumes that all nodes are co-operative in nature but, this assumption is not true in reality. Therefore, routing protocols should be robust against any cruel and misbehaving nodes. Numerous routing protocols are available for MANET to manage dynamic network conditions. These are vulnerable to security threats and fail to succeed against attacks. Malicious and misbehaving nodes are capable of generating various attacks and cannot be eliminated in the presence of strong cryptography techniques. It obeys all security primitives and protocol rules but misbehave at the time of packet transfer. Hence, mitigating routing misbehavior is a challenging task in the presence of malicious and selfish nodes. This work mainly deals with node misbehavior in routing.

1.2. Routing Misbehavior

MANET makes use of intermediate nodes to broadcast messages. Among the nodes in the network, node misbehaves by promising to forward the packet to other nodes but it fails to do so. Misbehavior of nodes can be in any one of forms such as overloading, selfishness, malicious activities and broken nodes. An overloaded node does not have sufficient Central Processing Unit (CPU) cycles, network resources in terms of bandwidth or buffer space to broadcast packets. A selfish node hesitates to forward the packet in order to maintain the battery life otherwise not willing to waste the available network resources. A malicious node drops the packets and thus results in DOS. A faulty node may occur due to software failure that stops it from forwarding the packets. To evaluate the impact of the node misbehavior in the network, classification of the misbehavior into three significant categories such as inactive nodes, selfish nodes and malicious nodes (Hollick *et al*., 2004). Based on the level of non cooperation of a node, the degree of misbehavior has decided. It seems that the selfish and malicious nodes result in much negative impact when compared to inactive nodes.

1.3. Inactive Nodes

The route discovery and packet forwarding are the two main steps in the routing process. The nodes in the network that neglect to participate in the network operation are inactive nodes. Inactive nodes include resource constrained nodes, lazy nodes and misconfigured nodes. Inactive nodes do not serve in the routing process. Inactive nodes cannot be assigned as either source or destination node. Therefore, node density gets decreases as inactive node increases. Results in network partition and the only subset of nodes take part in the routing process.

1.4. Malicious Nodes

Malicious nodes aim at reducing the network resources and delay the routing process. Malicious node may exist in various forms. A black hole attack is one of the malicious node behaviors that forward packets to the non existing destination node.

1.5. Selfish Node

A selfish nodes neglect to expose their presence in the network. Selfish nodes do not cooperate with the other nodes and discard the packets purposely without forwarding them. A selfish motive conserves battery energy, CPU cycles and bandwidth by not directly forwarding any packet passing through it.

1.6. Aim and Objectives

The major aim of this study is to identify any routing misbehavior by broadcasting random dummy packets into the network. The proposed technique effectively detects node misbehavior without imposing much routing overhead.

1.7. Scope of the Study

A node that participates in the routing process but does not forward packets on behalf of other nodes called selfish node. Selfish node uses the network resources only when it needs to communicate with the other nodes and it hesitate to forward the packets to save energy. Selfish nodes obey the routing process and network security mechanisms but, fail to cooperate in the routing process. This article proposes Random Dummy Packet Distribution (RDD) approach. In RDD approach, the source node generates the dummy packet and floods it to the destination. The destination node will send acknowledgment packet only for the dummy packet. If any critical node found in its route path, there is a chance for missing of acknowledgement. Then, the source node checks (trust factor calculation) to conform the critical node's misbehavior. Finally, the network eliminates misbehaving nodes and this information reach all the nodes in the network.

1.8. Related Works

Literature review reveals several approaches to detect the misbehaving nodes in MANETs. Marti *et al*. (2000), watchdog and path rater techniques detect misbehaving nodes. In watchdog technique, a buffer maintains most recently sent packets. The most recently sent packets checked with overheard packets for match. If they match with each other, the watchdog has no work to do. If a packet resides on the buffer beyond timeout period, watchdog increments a failure score for that node. When a score exceeds a certain level of threshold bandwidth, it concludes that the misbehavior of node and send the misbehavior alert message to the source node. In path rater, the pathrater provides ratings to nodes according to the following criteria. When pathrater detect a node from route discovery, then it considers it as a neutral node and provides a rating of 0.5. A node may rate itself with a rating of 1.0. At the time of path rate estimation, if all the nodes in the networks are neutral (except misbehaving nodes), then the pathrater selects the path that has a shortest length. When a node actively participates in the routing process, the rating of that node is incremented by 0.01 for every 200 ms. A Neutral node can earn a maximum rating of 0.8. If there is any route link failure during packet forwarding or the nodes move away from the transmission range, then the rating is decremented by 0.05.

The major idea behind a 2ACK scheme (Liu *et al*., 2007) is to transmit acknowledgment packets for every two hops in the opposite direction of the routing path. The acknowledgements only for a part of the received message in order to reduce overhead. It is the enhanced version of earlier work TWOACK (Balakrishnan *et al*., 2005). Another effective method for detecting misbehaving node is Cooperation Of Nodes: Fairness In Dynamic Adhoc NeTworks (CONFIDANT) algorithm (Buchegger and Boudec, 2002). Reputation and trust value calculated based on the observation and experience about behavior of other nodes. The monitor deploys "neighborhood watch" in which deviating nodes locally observed. The trust manager deals with alarm messages to warn the other nodes about the misbehaving nodes. After a node has gained knowledge about malicious nodes, it passes alarm messages to a set of nodes. The path manager component deals with the decision regarding path inclusion and deletion.

The sprite system proposed in (Zhong *et al*., 2003) is the credit-based techniques that provide incentives to ad hoc participant to achieve cooperation. This technique is on Credit Clearance Service (CCS). Whenever a node receives a packet, it maintains a receipt of the packet. The source node must pay for the intermediate nodes to forward the packet. The intermediate node updates the receipts to CCS and confirm received or broadcasted packets. The MARS (Zhao and Delgado-Frias, 2007) unite different features of MANET such as multi-hop routing, the end-to-end feedback mechanism and single path data transmission to defense against misbehaving nodes. A path without a misbehaving node can be discovered using this protocol. The source node transmits the data packet based on two different paths. One path is to transmit data packet and the other one is to exchange information. The feedback mechanism detects misbehaving node. Presentation in (Miranda and Rodrigues, 2003) explains the basic properties of protocols. It uses two key variables namely friends and foes. Friends are the set of nodes that eager to render service and foes are the nodes which are not eager to provide service. Sakthivel and Chandrasekaran (2012) proposed Path Tracing (PT) algorithm for detection and prevention of wormhole attack . It is an extension of DSR and PT calculates per hop distance based on the Round Trip Time (RTT) value and wormhole link using frequency appearance count. It detects the wormhole if per hop distance exceeds the maximum threshold range.

In a network that functions on a cooperation basis, a node may behave in a malicious manner. The property of cooperation of nodes helps to identify the malicious nodes. All nodes observe the other node's transmission in the network and then implement an Adaptive Quickest Detection (AQD) method to form a view that affects channel fading (Tomasin, 2011). View combined with a consensus algorithm at a fusion hub that blocks malicious nodes. The architecture of consensus algorithm considers the probability of malicious nodes purposely report false views in order to escape from detection.

2. MATERIALS AND METHODS

2.1. Overview of Random Dummy Packet Distribution (RDD) Approach

DSR is a reactive protocol primarily designed to provide a standard routing process in MANET

(Johnson and Maltz, 1996). DSR protocol is an on demand routing protocol and it is loop free. The basic operation of DSR is route discovery and route maintenance. All nodes maintain a routing table in its cache that gathers the route known to it. The route request is sent when a node does not have a route to the target node in its cache. When fresh routes are available, routing table updates new routes.

The intermediate node adds its own address in “route request” and then forwards towards the destination if it has a route. This is repeated until it reaches the destination. Once, “route request” reaches the destination, it sends “route reply” appending the details of intermediate nodes in the reverse path. This new route is updated to destination in the cache of the source node. After the route discovery, the route must be maintained till the packets reaches the destination as MANET has dynamic topology. The route maintenance checks whether a routing link has broken or not. The route is maintained based on parameters such as the number of hops, bandwidth utilized and end- to- end delay.

2.2. Overview of RDD

There are some nodes that do not obey the routing protocol and leads to misbehavior. To identify the misbehaving nodes in the routing protocol, we propose RDD approach. The main idea of RDD approach is that mobile nodes generate the dummy packets randomly and forward to the destination node in two different paths. It receives the acknowledgement only if there is no selfish or misbehaving node available in its root path. If it found any critical path, it evaluates the trust factor to conform its misbehavior. Finally, the confirmed node is isolated from the network. In order to reduce network traffic, only dummy packet transmissions are used for acknowledgement scheme.

Initially, a source node generates dummy packet in a random fashion and broadcasts it in a different path towards the destination on the basis of hop count. It waits for an acknowledgement from the destination node for dummy packets. If it does not receive any acknowledgement, then it switches to a critical path identification phase. In this phase, detection of the misbehaving link is more concentrated than that of a node. On generating and broadcasting the dummy packet, the source node waits until Time-to-Live (TTL) value gets expired. For a genuine path, it should receive acknowledgement within TTL value or else it is declared as a critical path. The critical path indicates that there may be misbehaving nodes in that path. Each node in the critical path is considered as a critical node and the link is the critical link. The acknowledgement scheme detects any critical path. To find the misbehaving node in the critical path, RDD calculates the trust factor for each node resides in the critical path.

The trust factor for each node in the critical path is calculated on the basis of a neighborhood monitoring system. The trust factor for a node is calculated using two parameters such as the number of packets sent and received. If both the parameters are equal, the corresponding node is a genuine one otherwise, it is a misbehaving or critical node. Therefore, by calculating the trust factor, a node can be conformed as a misbehaving node. The misbehaving node may hesitate to forward the received acknowledgement. The corresponding node is isolated from the network and will not be used in the future routing process. Since DSR has dynamic topology in nature, the nodes may walk away during the routing process. In such a case, the source node or sender generates “route error” message towards destination. The final stage of RDD scheme is that the critical node address is added in the blacklist of the sender node. Hence, the sender will not use the critical node for routing in the future.

Soon after the detection of misbehaving nodes it must be isolated from the network. This process must be completed within the penalization period. The source node intimates elimination of the misbehaving node to all other nodes by referring to Identity (ID) of a misbehaving node. With this reference, all other nodes in the network delete ID from their routing table. The proposed RDD scheme is then integrated to the DSR protocol to verify the effectiveness of the proposed scheme.

2.3. Proposed Scheme

There are many existing approaches for misbehavior detection in MANETs as in (Kargl *et al*., 2004). The proposed system mainly aims to reduce the routing overhead. In general, when a node transmits a packet to a certain destination, requester notifies its neighbors to find a path to a certain destination. The intermediate nodes decide whether to take part in the routing or not. The sender node selects an appropriate route among multiple discovered routes. The proposed RDD approach determines and excludes the colluding node based on the following steps.

2.4. Phases of RDD Scheme

2.4.1. Terms Used in RDD Approach

2.4.2. Dummy Packet

Packet does not contain any valid data.

2.5. Critical Node

Misbehaving node before the confirmation phase.

2.6. Misbehaving Node

Nodes does not cooperate with other nodes in packet forwarding.

2.7. Black List

List contains critical node IDs.

2.8. Observation Period

Time slot used to judge the node misbehavior

2.9. Dummy Packet Generation

In RDD approach, we use DSR algorithm for packet transmission. In the network, a dummy packet is generated by the source node when the network traffic is low, then it forwarded to two different shortest paths based on the hop count value. The destination node sends an acknowledgement to the originator only for a dummy packet, not all. If it does not get the Acknowledgement (ACK) or Route Error (RERR) message, then it enters into the next phase of RDD. The dummy packet generation reduces the additional communication overhead because dummy packet size is extremely small (because of no valid data). The dummy packet is generated in random manner, when network traffic is low. This dual transmission increases reliability of the routing protocol.

2.10. Identification of Critical Path

It is necessary to calculate the possibility of misbehaving routes. A route is set to be misbehaving when it has at least one misbehaving node. In this sub section, we investigate the detection of links which contains misbehaving nodes. According to RDD after sending the dummy packet the source node waits for a certain amount of time i.e., up to TTL expires. If it does not get ACK or RERR within TTL period, it considers that path as a critical path (there is a chance for the presence of misbehaving node). Each node in the critical path is considered as a critical node. Then, RDD calculates the success factor of each node along that path.

2.11. Trust Management

The critical link is detected using the transmission of dummy packet. This approach is effective only if we detect the critical nodes. To find out the critical node, we design a trust management system in which trust factor is calculated using trust parameters and each node stores it in its cache. The trust factor is calculated mainly based on each node's direct observation. Each node has a trust management system and it is responsible for collecting trust parameters and for calculating trust factor. The necessary trust parameters are feedback, the number of packet sent and received. The feedback is the service satisfaction provided by a node to its service requestors. The first function of trust management is to aggregate feedback from all other nodes and stores it in a feedback cell. Secondly, it calculates the trust factor of a corresponding node. Sakthivel *et al*. (2012) proposed a leader based reputation system. The elected leader is responsible for aggregation of trust value.

The **Fig. 1** shows the outline of a trust management system. The process of trust factor calculation is implemented in a distributed and dynamic manner at each node. There is no central server to maintain the trust factor of each node instead; a node attains other node's trust information from other nodes and calculates the trust factor of that node. This permits nodes to remain in an updated manner in the evaluation of the trust factor. Each node follows the same architecture to calculate the trust factor of other nodes. In **Fig. 1**, let us assume that node A as source and F as destination. It selects the path between them via B and C. The node A has to collect feedback about node B and C from the nodes D and E based on their monitoring and experience. Finally, the node A aggregates the feedback periodically in the feedback cell. For instance, D may contain the feedback of node D, E, B, C and A. The node must take the corresponding node's feedback using its ID. The feedback management system maintains a data structure of feedback informations along with the node's packet sent and received details.

2.12. Trust Factor Calculation

A node may misbehave by partially forwarding the packets to deceive the observer. This node misbehavior can be detected using the trust based system. In this study, the source node calculates the trust factor of the entire node along the critical path using the neighborhood monitoring system. The monitoring system continuously watches the activities of the neighbors. It maintains a trust table which records a packet sent and received entries.

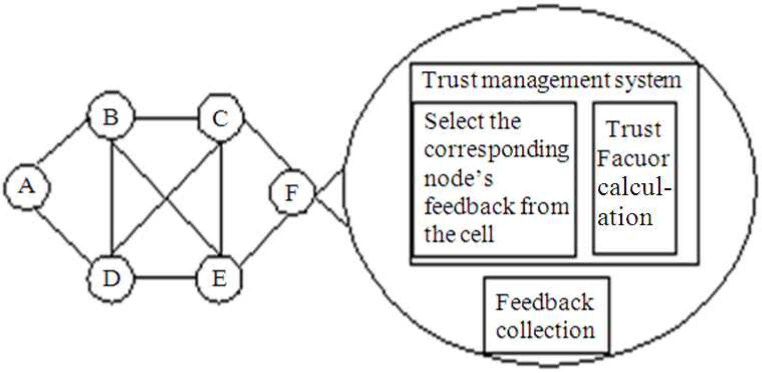
A
B
C
D
E
F
Trust management system
Select the corresponding node's feedback from the cell
Trust Facuor calcul-ation
Feedback collection

Fig. 1. Trust management architecture

The trust factor calculation is based on three parameters namely, feedback, the number of packet sent (NS_{ij}) and the number of packet received (NR_{ij}). Whenever a node collects feedback of a node from other nodes, it just compares them in the aspect binary credit of feedback. If the feedback provided is of high level satisfaction, it is awarded the credit of 1 otherwise 0. The mathematical expression (1) to calculate the trust factor for a node A is T (A) given as:

$$T(A) = \sum (NR_{ij}/ NS_{ij}) * F_d (n (A, i)) \quad (1)$$

The trust factor of any node can be calculated using the above expression. F_d (x) represents the service satisfaction level suggested by x. There is a predefined threshold value to determine whether it has a high or low trust factor. If a node has a trust factor above the threshold, it is given the credit of 1 or else it is given 0. Conditions for a node to be the critical are given below:

2.13. For node A, If

$NS_{ij} =$ NR_{ij} with high trust factor → node A is not a critical node

$NS_{ij} =$ NR_{ij} with low trust factor →node A is a critical node

$NS_{ij} \neq$ NR_{ij} with high trust factor →node A is not a critical node

$NS_{ij} \neq$ NR_{ij} with low trust factor →node A is a critical node

From the above conditions; it is clear that, for a node to be a genuine one, it must maintain an equal number of packets sent and received and must have high trust factor. Trust factor is calculated to detect the misbehaving nodes.

2.14. Misbehavior Confirmation

Trust factor calculated in the preceding section is used for validation of the misbehaving node. In the case of malicious and selfish nodes, once the node and its link chain is detected, it is informed to the sender node. Therefore, the detected misbehaving link is not advised to be chosen for data transmission. To confirm the behavior status of the node, we must examine the above conditions. A genuine node must satisfy two conditions. It must have an equal number of sent and received packets or high trust factor otherwise; it can be declared as a critical node and isolated from the network. Finally, the misbehaving node ID is added into the blacklist of the source node. It observes all the activities of the critical node for specified time slice called observation time. Once the observation period is over, the source node eliminates misbehaving node if continuing the same.

2.15. Elimination of Misbehaving Node

After the confirmation of the misbehavior, its unique ID added to the blacklist and the source node informs all other nodes. Normally, a single misbehaved node may initiate the packet drop attack. The identified node should be isolated from the network within the penalization period. The source node will announce this elimination with its ID to all other nodes in the network. All the nodes take part in routing will remove this ID from their route caches.

2.16. Integrating RDD with Routing Protocols

This sub section presents a approach to integrate the proposal with DSR. In DSR, a route can be discovered by receiving route replies or by observing the flow of packets over the network (Kargl *et al*., 2004). Each and every process in the routing are maintained in the route cache. When a node receives Route Request (RREQ) message, it should check whether it is the destination or not. If it is not the destination, it has the route to reach the destination using its cache. The dummy packet is sent using the DSR and received acknowledgement for the dummy packet. The sender sends the dummy packet only after the discovery of the route. The acknowledge packet is received for dummy packets sent. We generate only random dummy packets which do not increase the overhead much.

3. RESULTS AND DISCUSSION

3.1. Simulation of Non-Cooperative Nodes

To prove the negative impact of malicious and selfish nodes in the MANET, we have conducted several simulations using the NS-2 simulator The Network Simulator-ns-2, 2012. We have taken 50 nodes for the simulation that follows the random waypoint mobility model. The range of packet transmission of each node is about 150 m within the area of 1000×1000 m. The sending rate considered is 2 Mbps. The nodes in a given area move with the velocity of 0-20 m sec^{-1}. The pause time is 0 sec. that signifies that the mobility of the node is too slow. The percentage of misbehaving node is varied from 0-40% and the level of performance degradation is verified The simulator considers a set of nodes as a misbehaving, when it does not forward the packet as per agreement. It is assumed that the misbehaving nodes may drop partial or all the received packets.

3.2. Performance Evaluation

To analyze the performance of the MANET in the presence of the various percentage of misbehaving nodes. The following section discusses the performance analysis on various parameters.

3.3. Packet Delivery Ratio (PDR)

Figure 2 shows the packet delivery ratio of the proposed RDD approach with 2ACK scheme and original DSR. We modify the misbehaving ratio from 0-40 (40% of misbehaving nodes). From the figure, we can observe that all three schemes produce an appreciable packet delivery ratio in the absence of misbehaving nodes. When misbehaving node increases, PDR gradually decreases in 2ACK and RDD scheme. In case of the original DSR scheme packet delivery ratio decreases dramatically. The RDD scheme offers PDR of 56% even in the presence of 40% misbehaving node is in the network. Compared to the 2ACK scheme, RDD provides better PDR.

3.4. Packet Overhead

Figure 3 compares the packet overhead of RDD approaches with 2ACK, TWOACK scheme and the original DSR scheme. Except DSR, all the system causes overhead in the system because of ACK packet. Compare to TWOACK scheme 2ACK produces low overhead because it sends ACK only for selected packets not all. The proposed RDD scheme produces low overhead compared to 2ACK, because it generates few dummy packets. The 2ACK scheme produces 32% overhead, but RDD produces only 34% even in the presence of 40% of misbehaving nodes.

3.5. End-To-End Delay

End-to-End delay is the time taken to reach the destination from the source node. When the number of misbehaving node is 0 all the algorithms produces the same result. In the original DSR scheme delay is gradually decreasing because of low packet overhead and reduced network traffic. In case of 2ACK scheme, it is remarkably high compared to the proposed RDD scheme. The reason is 2ACK scheme produces extra overhead by sending the acknowledgement packet in two different paths. The end-to-end delay is reduced in RDD approach. The acknowledgement generated only for dummy packets. It causes considerably low packet overhead and traffic. **Figure 4** shows DSR produces only 0.01 m sec delay, 2ACK produces 0.049 m sec delay and RDD produces only 0.031 m sec delay even when the number of misbehaving node is 20.

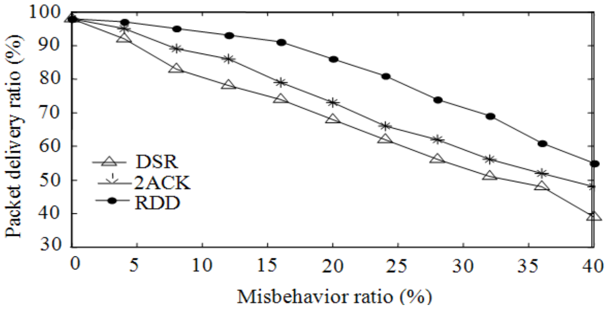

Fig. 2. Comparison of the packet delivery ratio

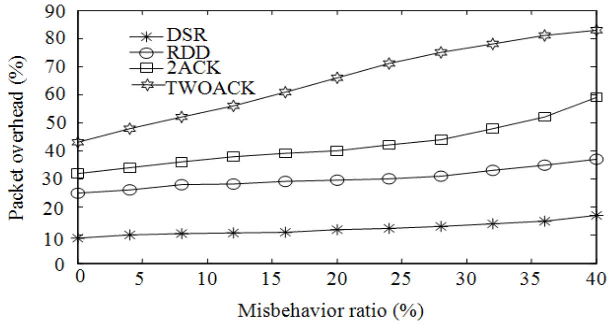

Fig. 3. Comparison of packet overhead

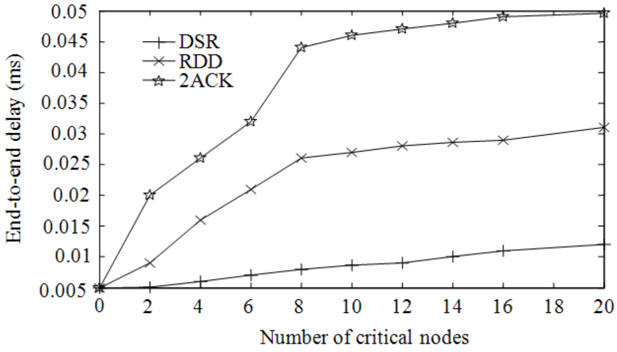

Fig. 4. End-to-End delay comparison of three different algorithms

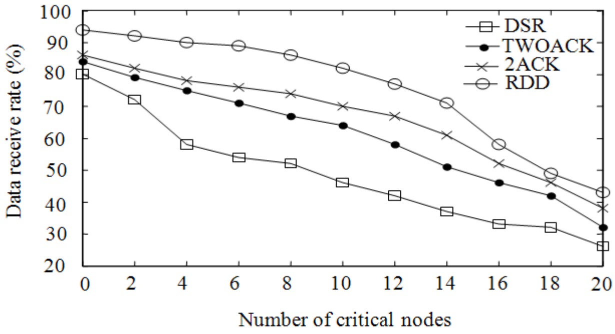

Fig. 5. Shows the data receiving rate of different schemes

3.6. Data Received Rate

The proportion of the number of data packets received to that of data packets sent. Because of a large number of RERR message due to the presence of misbehaving nodes, the DSR data receiving rate is extremely low. In case of TWOACK, 2ACK scheme data receiving rate increased up to 90% but, the proposed RDD scheme reaches 95% because of accurate detection of the selfish node with less overhead. It is evident from the curve that, compare to other schemes RDD produces better data receiving rate. **Figure 5** shows the data received rate of the proposed scheme and other existing misbehavior detection approaches.

3.7. Misbehavior Detection Rate

In **Fig. 6**, When the time is 70s the number of detection is only 12 in the CONFIDANT scheme but, RDD scheme detects 17 critical nodes in the same time. From the **Fig. 6**, RDD detected the misbehaving nodes as early as possible (i.e., 20s earlier) and excluded it from the network.

3.8. Throughput

In **Fig. 7**, it is clear that throughput decreases for an increase in number of critical nodes in both the schemes. The proposed RDD approach has higher throughput when compared to 2ACK approach. When the number of critical nodes goes beyond 30-40, then both the schemes fail as many routes are vulnerable.

3.9. Number of False Positives

From **Fig. 8**, we can observe that number of false alarm and TTL timeout are indirectly proportional to each other. If the TTL increases, the number of false positives decreases. All nodes move in a random manner and so there is a chance for large false positive values. The links are broken frequently in a high mobility network. The RDD approach may consider such nodes as critical.

3.10. Test Bed Evaluation

Buchegger *et al.* (2004) proposed test-bed environment for testing the performance of attack and misbehavior detection techniques. Most of the research projects on node misbehavior detection methods utilized network simulator tools to evaluate the performance. Simulator tool based research results may not reflect the actual performance metrics. The performance of the proposed techniques purely depends on the network environment and conditions. NS-2 simulator is an open source tool for wired as well as wireless network environment. NS-2 simulator supports simulation of realtime network environment and its performance evaluation is close to real-time environment. The real-time network set-up, configuration and its performance purely depend on the network environment factors. Hence, it is advisable to evaluate the proposed approaches in real-time test-bed environment to know the real impact and performance. We consider the test-bed evaluation as the future work of this research.

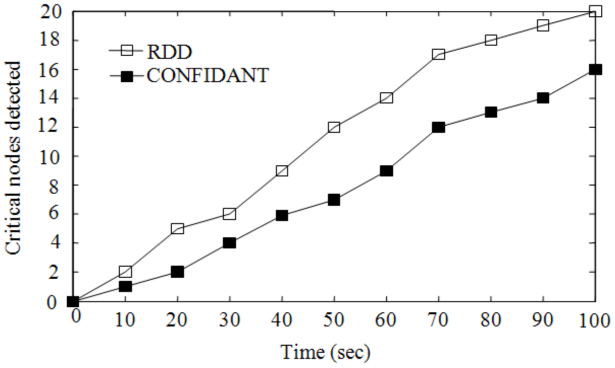

Fig. 6. Comparison of misbehavior detection rate of RDD and CONFIDANT

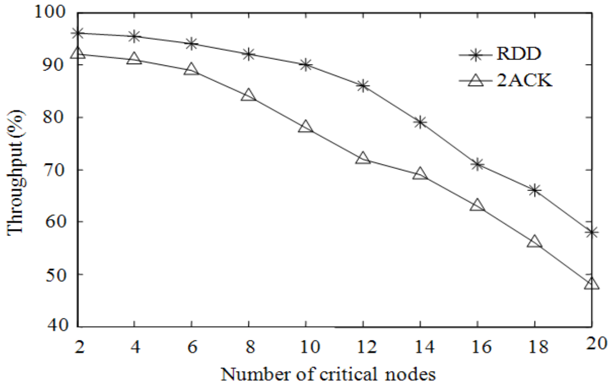

Fig. 7. Throughput comparison of RDD and 2ACK

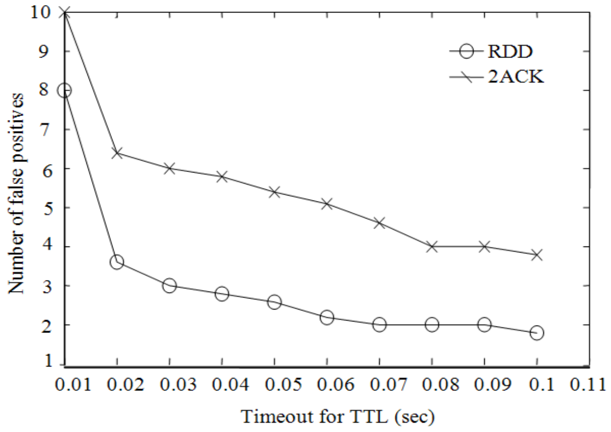

Fig. 8. Number of false positives in RDD and 2ACK

4. CONCLUSION

The misbehaving nodes have a negative impact on DSR protocol performance. In this study, we have presented a new method to detect misbehaving nodes in terms of malicious and selfish pursuit called Random Dummy packet Distribution strategy. The RDD approach is based on dummy packets, which are randomly generated and forwarded to the destination in two different shortest paths. If ACK or RERR does not return, it is considered as a critical path then the algorithm calculates the trust factor for each node in the critical path. We concluded that if a path contains at least one misbehaving node, the average performance decreases drastically. Therefore, the misbehaving node must be isolated from the network. The simulation results favor the proposed scheme and show its effectiveness in detecting the misbehaving nodes. The simulated results reveal the performance of the RDD approach over other schemes in the presence of malicious and selfish nodes.

5. REFERENCES

Balakrishnan, K., J. Deng and V.K. Varshney, 2005. TWOACK: Preventing selfishness in mobile ad hoc networks. Proceedings of the IEEE Conference on Wireless Communication and Networking, Mar. 13-17, IEEE Xplore Press, pp: 2137-2142. DOI: 10.1109/WCNC.2005.1424848

Buchegger, S. and J.Y.L. Boudec, 2002. Performance analysis of the CONFIDANT protocol. Proceedings of the 3rd ACM International Symposium on Mobile Ad Hoc Networking and Computing, Jun. 9-11, ACM Press, Lausanne, Switzerland, pp: 226-236. DOI: 10.1145/513800.513828

Buchegger, S., C. Tissieres and J.Y.L. Boudec, 2004. A test-bed for misbehavior detection in mobile ad-hoc networks-how much can watchdogs really do? Proceedings of the 6th IEEE Workshop on Mobile Computing System and Applications, Dec. 2-3, IEEE Xplore Press, pp: 102-111. DOI: 10.1109/MCSA.2004.5

Chlamtac, I., M. Conti, J.N. Jennifer and J.J.N. Liu, 2003. Mobile ad hoc networking: Imperatives and challenges. Ad Hoc Netw., 1: 13-64. DOI: 10.1016/S1570-8705(03)00013-1

Hollick, M., J. Schmitt, C. Seip and R. Steinmetz, 2004. On the effect of node misbehavior in ad hoc networks. Proceedings of the IEEE International Conference on Communications, Jun. 20-24, IEEE Xplore Press, pp: 3759-3763. DOI: 10.1109/ICC.2004.1313244

Johnson, D.B. and D.A. Maltz, 1996. Dynamic source routing in ad hoc wireless networks. Mob. Comput., 353: 153-181. DOI: 10.1007/978-0-585-29603-6_5

Kargl, F., A. Klenk, S. Schlott and M. Weber, 2004. Advanced detection of selfish or malicious nodes in ad hoc networks. Proceedings of the 1st European Workshop on Security in ad Hoc and Sensor Networks, Aug. 6-6, Springer Berlin Heidelberg, Heidelberg, Germany, pp: 152-165. DOI: 10.1007/978-3-540-30496-8_13

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. Mob. Comput., 6: 536-550. DOI: 10.1109/TMC.2007.1036

Marti, S., T.J. Giuli, K. Lai and M. Baker, 2000. Mitigating routing misbehavior in mobile ad hoc networks. Proceedings of the 6th International Conference on Mobile Computing and Networking, Aug. 06-11, ACM Press, Boston, MA, USA., pp: 225-265. DOI: 10.1145/345910.345955

Miranda, H. and L. Rodrigues, 2003. Friends and foes: Preventing selfishness in open mobile ad hoc networks. Proceedings of the 23rd International Conference on Distributed Computer System Workshop, May 19-22, IEEE Xplore Press, pp: 440-445. DOI: 10.1109/ICDCSW.2003.1203592

Papadimitratos, P. and Z.J. Haas, 2002. Secure routing for mobile ad hoc networks. Proceedings of the SCS Modeling and Simulation Conference on Communication Networks and Distributed Systems, (CNDS’ 02), KTH, San Antonio, TX, USA., pp: 193-204.

Sakthivel, T. and R.M. Chandrasekaran, 2012. Detection and prevention of wormhole attacks in MANETs using path tracing approach. Eur. J. Sci. Res., 76: 240-252.

Sakthivel, T., S. Vijaybhanu and R.M. Chandrasekaran, 2012. A novel leader based reputation approach for mobile ad hoc networks. Int. J. Comput. Appli., 47: 30-38. DOI: 10.5120/7258-0297

Tomasin, S., 2011. Consensus-based detection of malicious nodes in cooperative wireless networks. IEEE Commun. Lett., 15: 404-406. DOI: 10.1109/LCOMM.2011.022411.102050

Zhao, L. and J.G. Delgado-Frias, 2007. MARS: Misbehavior detection in ad hoc networks. Proceedings of the IEEE Global Communications Conference, Nov. 26-30, IEEE Xplore Press, Washington, DC., pp: 941-945. DOI: 10.1109/GLOCOM.2007.181

Zhong, S., J. Chen and Y.R. Yang, 2003. Sprite: A simple, cheat-proof, credit-based system for mobile ad-hoc networks. Proceedings of the 20th Annual Joint Conference of IEEE Computer and Communication, Mar. 30-Apr. 3, IEEE Xplore Press, pp: 1987-1997. DOI: 10.1109/INFCOM.2003.1209220

Zhou, L. and Z.J. Haas, 1999. Securing ad hoc networks. IEEE Netw., 13: 24-30. DOI: 10.1109/65.806983