IBIMA Publishing
Communications of the IBIMA
http://www.ibimapublishing.com/journals/CIBIMA/cibima.html
Vol. 2011 (2011), Article ID 413976, 11 pages
DOI: 10.5171/2011.413976

Improving the Performance of Wireless Ad-hoc Networks: Accounting for the Behavior of Selfish Nodes

Houssein Hallani and Seyed Shahrestani

School of Computing and Mathematics, University of Western Sydney, Penrith South DC, Australia

Abstract

Modern Wireless Local Area Networks (WLANs) with relatively high data rates have become an attractive technology for providing Internet connectivity for mobile users. Ad-hoc networks are a collection of mobile nodes that can be deployed without the need for any centralized management infrastructure. In such a set-up, to establish the required communication paths, each node must be willing to act as a potential router. In practice though, some nodes may act selfishly and refuse to forward packets. In Ad-hoc networks, a node may be considered as misbehaving for different reasons, for instance when it acts selfishly, refusing to forward packets. In some circumstances, the node can be overloaded, or they simply want to save their resources by not forwarding packets unless they are of direct interest to the node itself. Conversely, these nodes may still be expecting others to forward packets on their behalf. In this paper, we report the experimental results obtained from a typical Ad-hoc networks that contain selfish nodes. We also analyze the behavior of the nodes, to establish some quantifiable measure of their reliability. Such measures, based on the behavior history of the nodes, are then utilized to improve the performance and reliability of the widely used Ad-hoc On Demand Distance Vector routing protocol. We also report the results of simulations of large Ad-hoc networks in the presence of malicious or selfish nodes. These results clearly indicate the capabilities of the proposed approach in discovering communication paths with a minimal number of malicious or selfish nodes.

Keywords: Ad-hoc networks, Behavior analysis, Routing protocols, Selfish node, Wireless LAN.

Introduction

Professional deployment of WLANs requires the capability to broaden the coverage, which in turn may require the deployment of costly infrastructures. Ad-hoc networks consist of wireless nodes communicating without the need for a centralized administration. Perkins (2000) provides a fundamental explanation of Ad-hoc networks. In different sessions, data traffic flows over one or more paths between succeeding nodes to reach their destinations. In other words, as discussed in Shen (2008), the reliability of the communication process depends on all of the nodes potentially contributing to the routing of packets.

As we have discussed in our previous works, for instance in Hallani and Shahrestani (2005), wireless Ad-hoc network provides a number of advantages. The first of these, relates to ease and simplicity. A node, which is capable of reaching one or more available neighboring nodes, can be added easily to the network. The second is that wireless Ad-hoc networks allow the users to overcome the geographical and location limitations. This is because all nodes in the network can provide network connectivity for their neighboring nodes. This significantly improves on the connectivity provided by a single access point in an infrastructure mode wireless network. Another key advantage of this type of network is that

 Contact author: Seyed Shahrestani, e–mail: seyed@computer.org

they do not have a single point of failure. Scalability is also an advantage as Ad-hoc networks are robust and can be easily scaled up. Finally, wireless Ad-hoc networks offer significant cost savings as the existing environment does not have to be modified drastically to accommodate the addition of nodes to the existing and evolving network.

On the other hand, the technology and implementation of such networks present some serious concerns. They include consideration of the signal strength, the number of hops between the two communicating nodes, and the inherit lack of security, as discussed in Patwardhan, et al (2005). Such a node may be overloaded, or it may merely want to save its resources, as it does not see any advantages for itself through routing of packets. Marti, et al (2000) discuss how such selfish behavior, can result the functioning of the whole network to be drastically distressed.

Computer simulation has become one of the primary tools for evaluating the performance of wireless Ad-hoc networks. In our previous works Hallani and Shahrestani (2005), we have reported on simulation results of different scenarios, for which physical experiments have also been carried out. One of our motivations for those works was to validate the simulation process and results for Ad-hoc networks using OPNET. In this study, we expand those works to analyze the behavior of nodes, to establish some quantifiable measure of their reliability. The results of simulation studies, using OPtimised NETwork (OPNET) software simulator, under the same settings as the experimental networks are also reported. Correlation between the two sets of results is found to be satisfactory enough to validate the simulation technique and process. With this validation, using similar simulation methods, the examination of a rather large Ad-hoc network is then carried out**.** Using the results of such analysis, we propose an approach that is based on taking into account the behavior history of the nodes, while establishing the packet routing paths. The reported approach is an extension of the widely used Ad-hoc On Demand Distance Vector routing protocol. The results of simulation studies clearly show that by applying the proposed approach, significant improvements in the reliability of the Ad-hoc networks are accomplished.

Routing in Ad-hoc Networks

In the last few years, different routing protocols for Ad-hoc networks have been proposed. But most of them tend to ignore the fact that all the nodes in the network will not necessarily fully cooperate in routing the packets from source to destination. As discussed by Ning and Sun (2005), in general, many Ad-hoc devices operate on battery power. Consequently, power consumption for each transmission has a certain cost and significance. So, in reality, the assumption that all nodes perform the task of forwarding data, from which they do not directly benefit, while consuming their own battery power, is not always achievable. There is little reason to assume that some nodes will not try to achieve the benefits of participating in the network and avoid the disadvantages it involves. This could mean that some nodes may refuse to forward packets as expected and thereby decrease the efficiency of the network. Due to the dynamic nature of Ad-hoc networks, identifying nodes that express such malicious behaviour is a difficult task. The node originating the transmission might be out of range for detecting the malicious act.

In general, routing protocols can be classified as either proactive or reactive. Proactive protocols attempt to maintain up to date routing information for sending packets from each node to every other node at all times. The routing information is usually kept in a number of different tables, which are periodically updated. The proactive protocols include protocols like Destination-Sequenced Distance Vector (DSDV), Wireless Routing Protocol (WRP), and Optimised Link State Routing (OLSR). The main differences between these protocols, relate to the way the routing information is updated, and the type of information kept at each routing table. In addition, different number of tables may be maintained by each routing protocol. On the other hand, reactive protocols are designed

to reduce the overheads of proactive protocols, by maintaining information for active routes only. A number of different reactive routing protocols have been proposed. Ad-hoc On-demand Distance Vector (AODV), Dynamic Source Routing (DSR), and Temporally Ordered Routing Algorithm (TORA) are among them. AODV is the protocol that is used in this study.

In our work, the main focus surrounds on-demand routing protocols, where the route is discovered only when a node wants to send data to another node. The routing protocol used in this study is the AODV protocol. A detailed description of this protocol can be found in Perkins and Royer (1999). Given that AODV is the routing protocol used in this study, using the results of the studies reported in Zapata (2004) and Komathy (2008), the following attacks are of concern. An attacker can invade a route by generating a fake Route Request (RREQ) message. Also the attacker may create a Route Reply (RREP) message to disrupt an existing route between two communicating nodes. Further, an inside attacker can form a loop in the network to consume resources of the nodes in the loop by generating faked RREP. Finally, the attacker may send fake Route Error (RERR) messages to disrupt routes.

Trust and the Impact of Selfish Nodes

Compared to conventional wired networks, wireless networks are more vulnerable to attacks. Unlike wired networks where an attacker must first gain access to the media, in Ad-hoc networks access to communication media is already available. Many attacks on Ad-hoc networks can be launched from inside as well as from outside the network. In this work, only internal attacks caused by malicious nodes or the effect of nodes acting selfishly are studied. Such nodes may try to broadcast traffic to all nodes in the network or simply drop packets. An inside attacker can generate fake routing messages causing a break down between the source and the destination, eventually leading to an invaded route or isolated node.

In human relationships, trust is often expressed linguistically rather than numerically. As discussed in Shahrestani (2008), trust plays an important role in the cooperation and interaction between real world entities. In Ad-hoc networks, a measure of the trust level can be established by analysing the past behaviour of the nodes. A node that in the past demonstrated dependability and responsiveness will gain increasing trust. On the other hand, the unwillingness of a node to cooperate with other nodes will affect its trust level. In the proposed evaluation model, the cooperative nature of a node, or conversely its selfishness, is determined by the ratio of packet it has dropped. Its malicious behaviour, or conversely its trustworthiness level, is based on the ratio of the packets it has forwarded to the wrong destination, the number of replay attacks generated by the node, and the number of false routing messages it has produced.

Significant work has been done to improve routing in wireless Ad-hoc networks. Some of them apply a reputation analysis to tackle the problems associated with malicious and selfish nodes. Others, such as the work reported in Hadjichristofi, et al (2005), make use of the public and symmetric key infrastructure by designing secure routing solutions. This is an ongoing and active area of research. Many important problems and challenges still need to be addressed. These include the absence of a fixed infrastructure and centralized administration, as key management becomes a complicated problem and in turn making it difficult to provide proper security solutions.

An extension of AODV to secure the protocol has also been proposed by Zapata (2004). In this approach, it is claimed that the hop count information is the only mutual field in AODV and so used hash chains to secure this field. This approach also works under the assumption that an efficient key management system that distributes public keys to all nodes of the network is present. This is a serious drawback for its application in Ad-hoc networks in most practical situations. A reputation-based scheme to identify malicious nodes has also been studied by Komathy (2008). If a node fails to route the packet, it gets a low reputation mark that over time can result in expulsion of the node from the network.

However, this approach has the serious drawback of requiring acknowledgment to be sent by the destination to achieve higher reputation for the routing nodes that behave properly.

Trust Dependent Routing

In our proposed Behaviour-based AODV approach the source node attempts to find a route to the destination node that is free of malicious and selfish nodes. This is somehow different from the traditional AODV protocol, trying to choose the shortest route. To achieve this, a new parameter is added to AODV protocol relating to the behaviour history of the nodes. For each node, this parameter is a function of the packets relayed by that node, including control and data packets. In the initial stage, this parameter is the same for all nodes. Every time a node forwards a packet the parameter is incremented. Conversely, when a node fails to transmit a packet that it is supposed to relay, the parameter is decremented.

When a source node S desires to transmit a data packet to a destination node D, S must acquire the next hop node along the path to D. If this information is not readily available then route discovery is performed on demand. In a typical Ad-hoc situation, there are R1, . . . , Rn, totally n possible routes from the source S to the destination D. In each route there exist an x number of relay nodes n_1, . . . , n_j , . . . , n_x to help in forwarding the packets from S to D.

When S wants to send a message to D, and does not already have a valid route to that destination, it initiates a path discovery process to locate other nodes. The source node S propagates a RREQ to its neighbors. The RREQ packet includes: The IP address of D, the sequence number of D, trust level (the minimum trust level of all nodes in the current found route), hop count, and lifetime.

The destination sequence number field in the RREQ message is the last known destination sequence number for this destination and is copied from the destination sequence number field in the routing table. If no sequence number is known, the unknown sequence number flag must be set. The trust level field is equal to the source node's trust level. The hop count field is set to zero. When a neighbor node receives the RREQ packet, it will be forwarded if it matches some conditions.

When an intermediate node receives the RREQ from its neighbor, it first increases the hop count value in the RREQ by one. This is to account for the new hop through the intermediate node if the packet is not going to be discarded. The originator sequence number contained in the RREQ must be compared to the corresponding destination sequence number in the routing table. If the originator sequence number of the RREQ is greater than the existing value, the intermediate node compares the trust level contained in the RREQ to its current trust level to get the minimum. The intermediate node then updates the trust level of RREQ with the minimum. At this stage, the updated trust level of the RREQ is the trust level of the route. If the originator sequence number contained in the RREQ is greater than the existing value in its routing table, the relay node creates a new entry with the sequence number of the RREQ. If the originator sequence number contained in the RREQ is equal to the existing value in its routing table, the trust level of the RREQ must be compared to the corresponding trust level in the routing table. In the case that the trust level contained in the RREQ is greater than the trust level in the routing table, the relay node updates the entry with the information contained in the RREQ.

During the process of forwarding the RREQ, intermediate nodes record the addresses of neighbors from which the first copy of the broadcast packet was received in their routing tables. This in turn establishes a reserve path. If additional copies of the same RREQ are received later, these packets will be discarded. Once the RREQ reaches the destination D or an intermediate node with a valid route to D, the destination or intermediate node generates a Route Reply (RREP) packet and unicasts it back to the neighbor from which it received the RREQ. In the case where the generating node is the destination itself, it must update its own sequence number to the maximum of its

current sequence number and the destination sequence number in the RREQ packet originating the RREP. The destination node places its sequence number into the destination sequence number field of the RREP and enters the value zero in the hop count field of the RREP. When generating a RREP message, a node copies the destination IP address, the originator sequence number and the trust level from the RREQ message into the RREP message.

When an intermediate node receives the RREP from its neighbor, it first increases the hop count value in the RREP by one. As the RREP is forwarded back along the reverse path, the hop count field is increased by one at each hop. Thus, when the RREP reaches the source, the hop count represents the distance, in hops, of the destination node D from the source node S. The originator sequence number contained in the RREP must be compared to the corresponding destination sequence number in the routing table entry. If the originator sequence number of the RREP is greater than the existing value, the node compares the trust level contained in RREP to its current trust level to get the minimum, and then updates the trust level of RREP with that minimum. This minimum value represents the trust level of the route.

As with AODV, when a source node S has a packet destined for a destination node D, the routing module of the source node broadcasts a route request for a route from node S to node D. All the neighbours of node S receive the route request and check their local routing tables for a path to D. If any of them has a route to D, it sends a route reply back to node S. If multiple neighbours have routes to node D, they all reply back to node S. When multiple paths exist, using BAODV, node S chooses the route from the neighbour with the highest value of the parameter that indicates the trustworthiness.

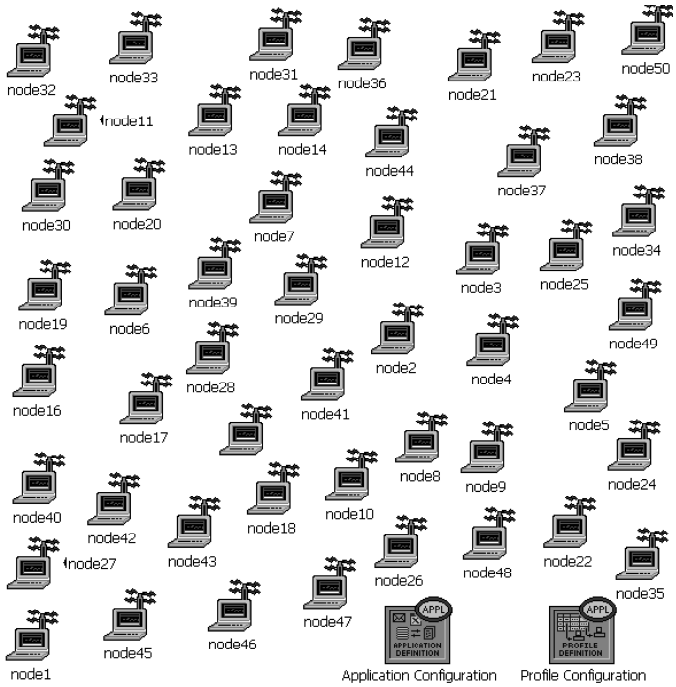

Figure 1. A Snapshot of the OPNET Simulation Setup

It should be noted here that there is a possibility that an intermediate node forwards the packet to a third node that is not a part of the route in order to deceive the originator node. This is solved by checking the acknowledgment sent back from the destination node to the source node. When an intermediate node receives

the acknowledgment packet, it retrieves the record corresponding to the IP address of the packet. The record contains the previous-hop and the next-hop nodes of the IP address. If the information matches, it forwards the acknowledgment to the previous-hop. In addition, it deletes the entry for the IP address from the routing table. If the information does not match, the intermediate node will decrement the behaviour parameter of the node that delivered the acknowledgment and aborts the packet.

Simulation Studies and Implementation Considerations

The simulation studies are carried out using OPNET Modeler V11.5. Each simulation scenario consists of fifty nodes, an Application Configuration, and a Profile Configuration. Fig. 1 shows a snapshot of the simulation setup. The Application and Profile Configuration are used to define the type of traffic sent between the nodes. The channel speed of the wireless LAN is set to 11 Mbps. The routing protocol used in the simulation is the AODV protocol. The MAC layer model is the OPNET implementation of IEEE 802.11 WLAN model.

To study the effects of the presence of malicious nodes in Ad-hoc networks, three performance metrics will be measured for a number of scenarios and situations. These are the throughput, the round-trip delay, and the packet loss rate. In order to facilitate the comparisons between the different approaches, all performance parameters are combined into one indicative index. The Overall Performance Index (OPI) is calculated as the weighted sum of the three performance metrics that have been considered so far. The simulation studies consist of a number of scenarios replicating practical situations. Each scenario runs in five different situations. In the first situation, none of the fifty nodes of the Ad-hoc network acts maliciously. In the second situation, five nodes chosen randomly out of the fifty nodes are acting maliciously. In the third situation, ten malicious nodes are present. In the fourth situation, fifteen nodes act as malicious nodes. In the fifth situation, twenty out of the fifty nodes are malicious node.

Table 1. Description of the Scenarios Used

	Description
Baseline Scenario	only two nodes involved in the communication, node2 is sending TCP traffic to node4
First Scenario	node2 and node3 are communicating simultaneously with node4 sending TCP traffic
Second Scenario	node 4 is receiving TCP traffic generated and sent at the same time from node2, node3, and node5
Third Scenario	node2 is sending TCP traffic to node5 (node2 is not within the range of node5 so node2 uses other nodes as relay nodes)
Fourth Scenario	only two nodes involved in the communication, node2 is sending UDP traffic to node4
Fifth Scenario	node2 and node3 are communicating simultaneously with node4 sending UDP traffic
Sixth Scenario	node 4 is receiving UDP traffic generated and sent at the same time from node2, node3, and node5
Seventh Scenario	node2 is sending UDP traffic to node5 (node2 is not within the range of node5 so node2 uses other nodes as relay nodes)
Eighth Scenario	node1 is sending TCP traffic to node50 (all nodes are motionless)
Ninth Scenario	node1 is sending TCP traffic to node50 (all nodes are mobile at a speed of 10m/s following a defined trajectory)
Tenth Scenario	node1 is sending UDP traffic to node50 (all nodes are motionless)
Eleventh Scenario	node1 is sending UDP traffic to node50 (all nodes are mobile at a speed of 10m/s following a defined trajectory)

The malicious nodes are implemented in four different ways. Some malicious nodes drop packets based on the simulation time (for example dropping all packets when the simulation time is between 50 and 100 sec). Other malicious nodes forward some of the packets to the wrong destinations. Some other malicious nodes fabricate and broadcast false routing messages. Other malicious nodes launch replay attacks.

Also, to study the effect of nodes mobility on the performance of Ad-hoc networks, all nodes move randomly 60 sec after the start of each simulation with a speed of 10 m/s. The rationale behind waiting for 60 seconds before the nodes start to move is to give them a reasonable time to establish their routing tables. Nodes move for 20 sec, pause at their destination for 60 sec and move back to their original locations.

In the baseline scenario, only node2 and node4 are involved in the communication. TCP traffic is sent from node2 to node4 and the throughput and the packet loss rate are measured at node2. In the first scenario node2 and node3 are set up to send TCP traffic to node4. While in the second scenario node5, node3, and node2 are communicating simultaneously with node4. In the third scenario node2 is sending TCP traffic to node5 through other nodes acting as relay nodes between the source and the destination. To check the effect of the transport protocol used between the communicating nodes on the performance of the Ad-hoc network, the same scenarios are repeated when the communicating nodes are sending UDP data traffic. All simulations run for five minutes. Table1 is a summary of the simulation scenarios. Due to space limitations, the results of only a few of the simulations are presented here.

Table 2. Packet Loss Comparison for the Baseline, First, Second and Third Scenarios

	Malicious TCP Traffic	Malicious UDP Traffic
Baseline Scenario	13.8%	9.82%
First Scenario	27.74%	22.68%
Second Scenario	28.5%	23.12%
Third Scenario	28.9%	22.27%

Table 2 shows the packet loss percentage variation for the baseline, first, second and third scenarios while the destination node is receiving TCP traffic. Also these values represent both situations where the malicious nodes are sending UDP and TCP traffic. It is also clear from these values that the packet loss rate is affected by the presence of the malicious nodes in the network. This table also shows that this performance metric is also weighed down by the transport protocol that the malicious nodes are using. This might be attributed to the fact that malicious nodes are trying to retransmit their traffic when using TCP. This process at node2 cannot distinguish between normal and malicious traffic causing higher packet loss rate compared to when malicious nodes are using UDP.

This part discusses the results of the fourth, fifth, sixth, and seventh scenarios when the communicating nodes are sending UDP data traffic. As stated before, this has been done to check the effect of the transport protocol on the performance of Ad-hoc networks. The values in Table 4 show the throughput variation for these scenarios. The measurement is made at the sending node (node2) and the table shows both situations where the malicious nodes are sending UDP and TCP traffic. It is noticeable from this table that the malicious nodes have affected the throughput between the communicating

nodes for all scenarios. These values also indicate that the impact on the throughput is less when the malicious nodes are using UDP traffic. This can be attributed to the use of the window mechanism to control the flow of data in TCP. When a TCP connection is established each end of the connection allocates a buffer to hold incoming data. If the receiving application can read data as quickly as it arrives, the receiver will send a positive window advertisement with each acknowledgement. However, it is well known that if the sender is faster than the receiver, incoming data will eventually fill the receiver's buffer. As data and malicious traffic arrive at node2, node2 sends acknowledgements to each node causing delay and full buffer at node2. In this situation node2 advertises a zero window.

A sender that receives a zero window advertisement must stop sending until it receives a positive window, causing delays at node2.

Table 4 shows the packet loss percentage variation for the fourth, fifth, sixth, and seventh scenarios. Also this table shows both situations where the malicious nodes are sending UDP and TCP traffic. It is also clear from these values that the packet loss rate is affected by the presence of the malicious nodes in the network. These values also show that this impact differs based on what transport protocol the malicious nodes are using. For example in the fifth scenario, the packet loss rate when 40% of the nodes are acting maliciously has raised from 0 to around 8% when malicious nodes are using UDP protocol compared to 19% when malicious nodes are sending TCP background traffic. By comparing the values in these four tables, it is noticeable that the throughput and packet loss rate when nodes are communicating using TCP protocol is higher compared to when they are using UDP protocol which might be due to the use of the windows mechanism in the connection oriented TCP .

Table 3. Throughput Comparison for the Fourth, Fifth, Sixth and Seventh Scenarios Measured in Mbps

	Malicious TCP Traffic (Measured in Mbps)	Malicious UDP Traffic (Measured in Mbps)
Fourth Scenario	4.91	5.15
Fifth Scenario	2.38	2.71
Sixth Scenario	1.51	1.72
Seventh Scenario	2.11	2.23

Table 4. Packet Loss Comparison for the Fourth, Fifth, Sixth and Seventh Scenarios

	Malicious TCP Traffic	Malicious UDP Traffic
Fourth Scenario	8.05%	3.37%
Fifth Scenario	18.6%	7.14%
Sixth Scenario	19.1%	7.59%
Seventh Scenario	12.25%	9.96%

Here, we discuss the results of the second part of the simulations. As stated before, in this part node1 is communicating with node50 via other nodes as shown in Fig. 1, which act as relay nodes between the source and the destination. Several scenarios and simulations were performed before and after applying the proposed BAODV approach in order to study the effect of the use of the behaviour history of the nodes on the overall performance.

In the eighth scenario the communicating nodes are sending TCP data traffic while the throughput comparison measured at node50. The results for this case clearly show that the throughput increases when BAODV is used. This is due to the fact that node1 is now sending the packets to node50 through a route which has a reduced number of malicious nodes, compared to using AODV alone.

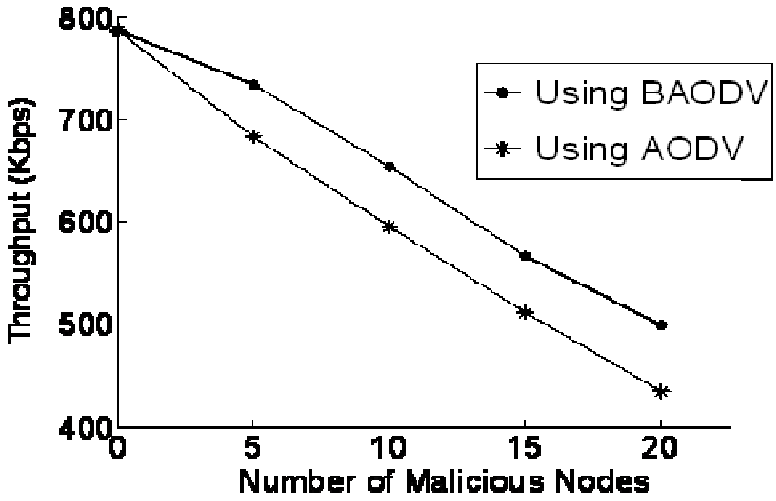

Fig. 2- Throughput Comparison for the Tenth Scenario Measured at Node50. In this Scenario Node1 is Sending UDP Traffic

The graphs in Fig. 2 show the throughput comparison for the ninth scenario. In this scenario nodes are moving according to the defined trajectory given earlier in the paper. Studying these graphs, it is noticeable that the throughput has also increased when applying the proposed approach. It is also clear that the throughput is higher when the nodes are motionless. This is due to the fact that when moving, the node can drop the connection with its neighbors causing the routing protocol to reinitiate the route between source and destination.

Table 5. Packet Loss Comparisons for the Eighth, Ninth, Tenth, and Eleventh Scenarios

	Without the Proposed Approach	With the Proposed Approach
Eighth Scenario	51%	45%
Ninth Scenario	57%	49%
Tenth Scenario	44%	36%
Eleventh Scenario	52%	44%

Table 5 shows the packet loss percentage values for the eighth, ninth, tenth and eleventh scenarios when 40% of the nodes are acting maliciously. This table shows both situations before and after applying the proposed approach. It is noticeable here that the packet loss rate has decreased with the proposed approach for all scenarios. The decrease in the packet loss can also be credited to the fact that the new route between source and destination has none, or less, malicious nodes. It can also be noted that the packet loss is lower when the nodes are motionless. This can be attributed to the fact that packets are dropped when losing the connection between the moving nodes.

Concluding Remarks

In this paper, the effect of malicious and selfish nodes on the performance of Ad-hoc networks is presented. With the lack of central infrastructure in these networks, evaluating and establishing trust and dependability between their comprising nodes is not an easy task. To overcome this difficulty, a new approach based on utilization of past behaviour of nodes is proposed. The approach referred to as BAODV, is an extension of the AODV protocol. This approach is based on the behaviour history of all member nodes of Ad-hoc networks. The results of a number of simulation studies based on using conventional routing techniques with and without implementing the proposed approach are also reported. The results corresponding to cases where the proposed approach has been implemented show significant improvements in the performance and reliability of the wireless Ad-hoc networks in the presence of malicious or selfish nodes. For instance, with 40% of the nodes of the Ad-hoc network acting maliciously, and nodes being either stationary or mobile, increases in the throughput of 11% and 13% respectively, can be achieved.

Acknowledgement

We would like to thank OPNET for their kindness in providing us with Modeler software license, which has greatly assisted in finalizing this paper. We also would like to thank Cisco for the generous scholarship which has permitted us to reach forward with our studies.

References

Hadjichristofi, G., Adams, W. & Davis, N. (2005). "A Framework for Key Management in Mobile Ad-Hoc Networks," Proceedings International Conf. on Information Technology: Coding and Computing, (ITCC 2005). 568-573.

Hallani, H. & Shahrestani, S. (2005). "Performance Evaluation and Simulation Verification for Wireless Ad-hoc Networks," *WSEAS Transactions on Communications*, 4, 355-362.

Komathy, K. & Narayanasamy, P. (2008). "Trust-Based Evolutionary Game Model Assisting AODV Routing against Selfishness," *Journal of Network and Computer Applications*, 31 (4). 446-471.

Marti, A. S., Giuli, A., Lai, A. & Baker, A. (2000). "Mitigating Routing Misbehavior in Mobile Ad-Hoc Networks," Proceedings 6th Int. Conf. on Mobile computing and networking, Boston, Massachusetts, United States, 2000, 255-265.

Ning, P. & Sun, K. (2005). "How to Misuse AODV: A Case Study of Insider Attacks against Mobile Ad-Hoc Routing Protocols," Ad-hoc Networks, 3, 795-819. OPNET Modeller, http://www.opnet.com.

Patwardhan, A., Parker, J., Joshi, A., Iorga, M. & Karygiannis, T. (2005). "Secure Routing and Intrusion Detection in Ad-hoc Networks," Proceedings 3rd IEEE International Conf. on Pervasive Computing and Communications, (PerCom 2005). 191-199.

Perkins, C. (2008). Ad-hoc Networking, *Addison-Wesley*.

Perkins, C. & Royer, E. (1999). "Ad-hoc on-demand Distance Vector Routing," *Mobile Computing Systems and Applications*, 90-100.

Shahrestani, S. (2008). "Utilization of Soft Computing to Improve Cooperative Management Efficiency," *WSEAS Transactions on Circuits and Systems*, 7 (7). 620-629.

Shen, Z. & Thomas, J. (2008). "Security and QoS Self-Optimization in Mobile Ad-hoc Networks," *IEEE Transactions on Mobile Computing*, 7 (9). 1138-1151.

Zapata, M. G. (2004). 'Secure Ad Hoc on-Demand Distance Vector (Saodv) Routing,' Internet Engineering Task Force (IETF) Draft.