Routing in Ad Hoc Networks to Discourage Selfish Nodes

COMP 4203 - Project Proposal

James Tyler (100976157), Bradley Koon (100972222), Imdad Ali (101099711)

March 7th, 2018

**I. Introduction**

        For our project, we plan to modify the SA-DSR routing protocol proposed in [2] using selfish node detection techniques described in [1] in order address the shortcomings of SA-DSR and help to discourage selfish behaviour of nodes in MANETs. In section II we summarize our first article, [1], in section III we summarize our second article, [2], in section IV we discuss our proposed work, and section V concludes our proposal.

**II. Article #1 Summary - Selfish Node Detection in Ad Hoc Networks**

In 2017, Mangesh M Ghonge and Dr. P.M Jawadhiya published an article on selfish attack detection in mobile ad hoc networks. This paper addresses the issues with selfish nodes in MANETS, discusses various methods of detecting selfish nodes, and proposes some new ideas for developments of new methods in the future. In a MANET, mobile nodes have a limited battery capacity and become “selfish” in order to conserve their energy. Route discovery and packets forwarding are said to use a lot of energy, bandwidth, CPU and memory. Sometimes, selfish nodes may instead refuse forwarding packets for other nodes and focus on delivering their own data.

This article also described how in situations such as sensor networks and conference meetings, there are some nodes that do not share a common objective with other nodes and do not feel as if they “belong” to the network. Regardless of battery level, nodes in these situations will become selfish to preserve their own battery life. This can disrupt the communication of the networks that they are in. Since MANETS are wireless, security against malicious nodes has also been brought up as a concern. Malicious nodes can join a network and become selfish in order to disrupt the operation of the network for its own advantage. Thus, having measures against selfish nodes could improve the communication of the network and improve security. This paper describes five existing methods for selfish node detection. These methods were classified as follows:

*Audit based system:*Isolates both continuous and selective packet droppers. This is done by keeping a record of the behaviour of such nodes in the network.

*Credit based system:*This method does not aim to identify misbehaving nodes in the network. It instead tries to encourage other nodes in the network to help forward by giving them “credit”. The more nodes forward, the more “credit” they accumulate.

*Reputation based system:*Uses a rating system to evaluate the trustworthiness of nodes in forwarding traffic. The Node’s behaviour is taken into consideration. If a node is consistently co-operating with other nodes and forwarding traffic, they are given a higher rating. If not, they are given a lower rating.

*Acknowledgement based system:*The destination sends an acknowledgement to the source to inform whether or not it has successfully received the message.

*Collaborative based system:*When a node in the network suspects that another node in the network is misbehaving, it lets other nodes know of the misbehaving nodes. This way other nodes know of the misbehaving node and will be sure to not forward any traffic to them.

The design of MANET routing protocols has been centered around efficiency and convenience but none of them have addressed the issue of selfishness before. Mobile nodes have limited battery lives and spend a great deal of energy during packet forwarding and route discovery. When a mobile node becomes low enough on energy, it can refuse to help forward packets for other nodes. Selfish awareness is especially crucial for the functioning of the network since nodes always rely on other nodes to forward their messages. Some nodes in the network use more energy than others, in particular, intermediate nodes. The drawbacks of the mentioned selfish detection solutions, are that none of them consider the node’s willingness to forward data packets. This paper predicts that taking into account these measures is important during routing, route discovery and route maintenance to better identify selfish nodes so that they aren’t chosen to collaborate. The assignment based system was proposed to hypothetically address the issues with existing selfish node detection methods.

The authors discuss their new idea for a selfish detection scheme, with the following goals:

1. Develop and test the assignment based selfish node detection system that takes into account the node’s willingness to forward packets.
2. Develop an algorithm to maintain social selfishness of nodes in the mobile network.
3. Develop an algorithm to maintain the willingness of the nodes.
4. Evaluate the trustworthiness of the nodes.
5. Evaluate the reputation management system.
6. Provide a system (assignment based system) to satisfy user demands for selfishness.

**III. Article #2 Summary - SA-DSR Routing Protocol**

        In 2016, Dimitra G. Kampitaki and Anastasios A. Economides published a academic paper titled, “Novel routing protocol for Mobile Ad Hoc Networks with selfish and altruistic nodes”, which addresses the issues pertaining to selfish nodes in Ad Hoc networks and proposes a solution in the form of a new routing protocol. The authors begin by suggesting that in an open Mobile Ad Hoc Network (MANET), mobile nodes will inevitably become selfish as a means to preserve their own resources. It is also suggested that there is a difference between selfish nodes and malicious nodes since selfish node behaviour can be “...predicted and addressed without excluding them from the network.”.

        Since the cooperation of other nodes in an open MANET cannot be assumed, the authors propose a new routing protocol, Selfish Aware Dynamic Source Routing (SA-DSR) which is based on the existing protocol, Dynamic Source Routing (DSR), but takes into account the selfishness of other nodes when choosing a route.

        DSR was one of the first reactive routing protocol for MANETs, consisting of two phases; Route Discovery Phase, and Route Maintenance Phase. In the discovery phase the source node floods the network with RREQ (Route Request) packets in order to find a route to the destination. All nodes that receive this pass it along until it is received by the intended destination node. The destination node replies with a RREP (Route Reply) packet, which traverses the same route in reverse to the source node, saving a path for future communication. The Route Maintenance Phase is intended to keep known routes alive, assuming they can still exist.

        SA-DSR follows the same general methodology as DSR, but makes a few key changes to the Route Discovery Phase, and the RREP packets. Firstly, all nodes must keep track of a few new values. Each node maintains a “Selfishness Type” which can correspond to any of the following four values: Always Altruistic (AA), Sometimes Selfish (SS), Often Selfish (OS), and Always Selfish (AS). The Selfishness Type is determined based on the battery level of the device; devices with high power levels will tend to be Altruistic, whereas devices with low battery levels will tend to be more selfish. Nodes also maintain a “Altruism Coefficient (AC)” which represents the nodes satisfaction level with the network so far. Each nodes’ “Forwarding Probability (FP)” is a value combining the Selfishness Type and the AC. As the name suggests, the FP is the probability that the node will forward the RREQ packets to other nodes.

        When a source node wants to send a message to a destination node, it first checks its routing table for known paths to that destination. If there are two or more such paths, the source node chooses the path with the lower SDP value, if there are less than 2 paths, the source node floods the network with RREQ packets. For every retransmission of a RREQ packet, the source node increases its AC value, if there are no retransmissions of the RREQ packet, the source node decreases its AC value. When a relay node receives a RREQ packet, it makes a decision as to whether to forward the packet, or drop the packet. If the relay node forwards the packet, the sources AC will be increased, and the source node will be more likely to help out the relay node in the future. If the relay node drops the packet, the source nodes’ AC might be decreased if other relay nodes do the same, making the source node less likely to contribute to routing for other nodes later on. When a RREQ packet is received by the destination node, a RREP packet is created and returned via the same route the RREQ took. The FP value of each node is added to the RREP packet as it is passed back to the source node so that the source can calculate the SDP value for each obtained path.

        The researchers simulated the following scenarios in their research; normal DSR without any selfish nodes in the network, normal DSR with selfish nodes in the network, and SA-DSR with selfish nodes in the network. They collected data using three metrics; Packet Delivery Ratio (PDR), Average End-to-End Delay (AEED), and Normalized Routing Overhead (NRO). The results of the simulation showed that when using DSR without selfish nodes, the PDR is the best, but as soon as selfish nodes begin to appear, it quickly becomes the worst performing protocol. SA-DSR in comparison has very similar results to DSR with only altruistic nodes, despite having selfish nodes in its network.

**IV. Proposed Work**

Discussion of Problem & Project Goal

As mentioned in both articles, selfish node behaviour negatively impacts the performance of a MANET, and is an issue that must be addressed. In [1], various methods of detecting selfish nodes are discussed, and in [2] a routing protocol that takes advantage of selfish behaviour in an effort to discourage it is presented. Before moving forward, it is important to clarify that there are two types of non-malicious selfish nodes. Pure Selfish Nodes are nodes that display the behaviour discussed in [2], where they take no part in the Route Discovery or Route Maintenance phases. Partial Selfish Nodes, mentioned in [1], are nodes who take part in the Route Discovery and Maintenance phases, but that drop data packets when they are assigned the position of a relay node.

        The primary reason that a node would want to become selfish is in order to preserve its own resources, specifically its energy. In most routing protocols for MANETs there is no regulation of the energy consumption distribution across the network. It is entirely possible, even likely that certain nodes will be contributing more of their resources to the benefit of the network than others. The SA-DSR routing protocol proposed in [2] aids in regulating the energy consumption across the network by using selfish behaviour as part of the protocol itself (see above summary for details). SA-DSR uses selfish node behaviour in a controlled manor, removing the need for nodes to become selfish by their own means. On the surface it may seem that SA-DSR is a well-rounded solution to selfish nodes, but there is one core issue; SA-DSR assumes that nodes will only exhibit Pure Selfish behaviour, and only within the constraints of the protocols rules. The protocol does not expect, or address the possibility of nodes in the network exhibiting Partial Selfish Behaviour. This is the problem that we will be addressing in our project.

        Our goal is to modify the proposed SA-DSR protocol to use Selfish Node Detection techniques proposed in [1] to avoid routing packets through Partial Selfish Nodes. The resulting protocol will be one that uses Pure Selfish behaviour in a controlled manner, reducing the need to become a selfish node, and that is reduces the negative impact that Partial Selfish Nodes impart on the network.

Proposed Modifications to SA-DSR Protocol

        We plan to modify the SA-DSR protocol to use some of the selfish detection techniques from [1] in order to reduce the negative impact of partial-selfish nodes on the network. SA-DSR, as proposed in [2] assumes that all selfish nodes are pure selfish nodes, and thus has no way of managing the misbehaviour of partial selfish nodes. By modifying SA-DSR to use a combination of Acknowledgment Based and Reputation Based selfish node detection, we believe we can minimize the negative impact these nodes have on the network.

        As in many reputation based schemes, each node will keep a table of its first hop neighbours, and track their behaviour using a modified version of TWOACK. The TWOACK scheme dictates that whenever a node receives a data packet, it sends an acknowledgement back two hops in the communication chain so that the first node knows whether or not the second node indeed transmitted the data. The first node can then update its behaviour table for its neighbour accordingly, and use this information in the Route Discovery Phase to influence the Forwarding Probability value in the RREP packet being sent back from the destination to the source. This way nodes with good behaviour are warning each other about the misbehaviour of others.

        One issue with the TWOACK scheme however is that it relies on the potentially selfish node to retransmit the TWOACK packet to the originating node. A selfish node could just as well spoof that packet as well, making it even more difficult for its neighbours to track its misbehaviour. As a means to prevent this, we would like to explore a new idea of sending the TWOACK packet through a separate intermediate node not involved in the communication chain. This would mean that the selfish node cannot hide its misbehaviour from its neighbours nearly as effectively. Although we believe this method will be effective at evaluating the behaviour of nodes, it is likely that the overhead for doing so will have a significant impact on the efficiency of the networks communications. Nonetheless we would like to explore the idea if time permits, and continue to use our TWOACK-Reputation hybrid method for our simulations.

Simulation Scenarios

        We plan to develop a simulation environment in which we can simulate a MANET, and implement various routing protocols. We will implement the routing protocols discussed in [2]; DSR and SA-DSR, as well as our modified version of SA-DSR, so that we can compare the performances of each in various scenarios.

The first scenario we will test is the performance of the DSR protocol. We will generate test data including nodes with various positions and movements and initial battery levels, and a random set of desired communications to make. We will run the scenario, with the nodes using the DSR protocol to communicate, with 4 different selfish node configurations. Firstly, with no selfish nodes, then with pure selfish nodes present, then with partial selfish nodes present, and finally with a mixture of different type of selfish nodes present. This test will help us to establish how the two different kinds of selfish behaviour affect the performance of the protocol.

The second scenario will be to run the simulation with the same test data, only this time using the SA-DSR protocol as defined in [2]. This test will help us to evaluate how the SA-DSR improves, or fails to improve the performance of the network with different selfish node configurations. We expect SA-DSR to be an improvement over DSR when there are Pure Selfish nodes in the network, but not to have much effect when there are Partial Selfish nodes in the network.

The third scenario will be the same as the second, only this time using our modified version of SA-DSR. This test will help us determine whether or not our modifications improved the performance of the network over normal SA-DSR in each type of selfish node configuration. We expect to see improved performance in Partial Selfish node configurations, and little change in the other configurations.

For each simulation run we will collect data corresponding to several metrics. We will use the same three metrics that were used in the simulations described in [2], as well as an additional metric of our own, which we believe to be the most indicative of the success or failure of SA-DSR. Packet Delivery Ratio (PDR), Average End-to-End Delay (AEED), and Normalized Routing Overhead (NRO), were all used as metrics in the simulations described in [2]. PDR is the percentage of successfully delivered packets, AEED is the average time a data packet takes to get delivered, and NRO is the total control packets divided by the total packets received in the network. Collecting these metrics will help us to compare the performances of the protocols and will reveal any performance shortcomings of SA-DSR with and without our modifications. The new metric that we are adding we will call Battery Depletion Deviation (BDD). BDD is calculated as the standard deviation of the changes in battery levels of nodes across the network. A lower BDD indicates that the energy consumption across the network is fair, whereas a high BDD indicates that the energy consumption is poorly balanced, and certain nodes in the network are working significantly harder than others. We hold this metric as the most significant because it is the best indicator as to whether or not a given protocol is doing a good job of removing the need for nodes to exhibit selfish behaviour to survive comfortably in the network. If SA-DSR with or without our modifications doesn’t have improved BDD over DSR in the same scenarios, then their usefulness is questionable to say the least.

Program Implementation

Our program will consist of a back-end which will include routing protocol implementations, data collection, logging, exporting, and a front-end which will provide a graphical user interface responsible for visualizing ongoing simulation and creating real time updates graphs of the properties of the nodes in the network. The user will be able to choose the run parameters including protocol of choice (DSR, SA-DSR, and SA-DSR with modifications), number of nodes, and starting battery levels of nodes.

Our frontend interface will be sectioned. It will contain four (potentially more) sections on the same page that will allow us to easily see statistics about the nodes and any exportable data from the experiment. The frontend will allow users to modify the nodes’ properties such as, battery level, mobility, and simulation speed, before starting the simulation. The input data for the properties will remain the same from the start for each test using each protocol in order to conduct a fair test. Most of all, we’ll be using the sections of graphs to analyse the data of how fairly distributed the power consumption is across the network using DSR and SA-DSR with and without modifications.

For each simulation run, we will generate input data including number of nodes, position of nodes, starting battery levels, and mobility of nodes. The same set of data will be used in simulations for all protocols in order to ensure our results are useful for comparison. As mentioned previously our simulations will be run with different node selfishness configurations as well. We also plan to have pre-set scenarios that exemplify the shortcomings of certain protocols, for example, having two groups of nodes separated by a single intermediate node who will likely end up being used for most communications, and thus be expending the most resources.

As the simulation runs, the displayed graphs will update in real time based on the properties of the nodes in the network. This data will be collected by the backend, and logged to files so that they can be exported as a \*.csv files that can be later imported into data analysation software such as a spreadsheet processor. The data collected will correspond to the 4-metrics described in the previous section, and will be overlaid so that the performances of the different protocols can be easily compared.

The metric which for which we hold the highest significance is BDD since the purpose of the SA-DSR protocol is primarily to discourage the need to selfish behaviour. As such, we will be tracking the battery levels which will be decreasing by a constant factor every few seconds, as they do in the real world. When a node transmits a packet and/or processes a packet partially or completely, then the node will lose a certain (constant) amount of energy. We will run the simulations with all nodes starting at the same battery level, and with all nodes starting at different battery levels.

We hypothesize that for the DSR simulations, all scenarios involving selfish nodes will have a negative impact on the performance of the protocol, but when no selfish nodes are present, we expect DSR to be the best performing protocol with all metrics except BDD, for which we expect poor results. When running the same simulation only with SA-DSR, we expect slightly worse (but very close) performance for PDR, AEED, and NRO (when pure selfish nodes are present), but we expect significantly improved performance with respect to BDD when using no selfish nodes, or pure selfish nodes are present. When simulating SA-DSR with partial selfish nodes, we expect significantly worse results with respect to PDR, AEED, and NRO since the protocol has no means of dealing with partial selfish nodes. When running the simulations against our modified SA-DSR protocol, we expect similar results to SA-DSR with respect to all metrics when running with pure selfish nodes, and we expect a boost in performance to PDR, AEED, and NRO in comparison to normal SA-DSR when there are partial selfish nodes in the network. We also expect better BDD results since less nodes will need to retransmit data packets due to partial nodes dropping them.

**V. Conclusion**

In summary, we propose the addition of partial selfish node detection techniques to the existing SA-DSR protocol in order to address its shortcomings, and we propose a new metric to evaluate the effectiveness of SA-DSR at discouraging selfish behaviour in MANETs. Our simulation software will provide a real-time visualization of simulation runs, and will collect and export data for comparison and analysis.

**References**

[1] D. G. Kampitaki and A. A. Economides, "Novel routing protocol for Mobile Ad Hoc Networks with selfish and altruistic nodes," *2016 International Conference on Telecommunications and Multimedia (TEMU)*, Heraklion, 2016, pp. 1-5.

doi: 10.1109/TEMU.2016.7551935

[2] M. M. Ghonge, P. M. Jawandhiya and V. M. Thakare, "Selfish attack detection in mobile Ad hoc networks," *2017 International Conference on Innovations in Information, Embedded and Communication Systems (ICIIECS)*, Coimbatore, 2017, pp. 1-4.

doi: 10.1109/ICIIECS.2017.8276136