

A game theory-based incentive framework for the next-generation vehicular networks

Rui Pedro da Costa Domingos
rui.domingos@tecnico.ulisboa.pt

Instituto Superior Técnico, Universidade de Lisboa, Lisboa, Portugal

Abstract—Vehicular networks that include V2V and V2X connectivity, have huge potential in terms of improving the safety and driving experience. These networks use incentive schemes that allow nodes to learn from their surroundings, assess the veracity of reported events as well as the honesty of vehicles and, of course, encourage cooperation. Conventional incentive schemes have their own vulnerabilities, which are created mostly from attacks from misbehaving nodes. In this work we design a novel evolutionary game theory-based incentive framework by considering past reputations to improve cooperation among vehicles reduce the impact of DoS attacks. This framework is tested for four routing protocols and two different types of attacks, gray-hole attacks and black-hole attacks. The results of the simulations in these different scenarios show a good performance in denser networks, for protocols with high overheads and for scenarios where misbehaving nodes never cooperate with the network, namely black-hole attacks. Most importantly, there is a significant improvement in overhead and latency.

Index Terms—Vehicular Networks, Incentive Schemes, Game-Theory, Reputation, Black-hole, Gray-Hole

I. INTRODUCTION

Nowadays, the development of communication systems has opened a plethora of new and exciting possibilities for the world of ITS. These systems can prove to be incredibly useful in the improvement of passenger safety and giving said passenger an enjoyable trip by providing entertainment and information. There have been significant advancements in electronics aiding car drivers, along with communications outside the vehicle. The concept of VANET has evolved into the much broader concept of IoV. We can longer consider only V2V communication. V2X communication used in IoV, includes another set of variants to account for, like RSUs, pedestrians, buildings and sensors placed along roads. The information from all these different sources can be disseminated in the network ensure efficient emergency warning notifications, improve management of traffic levels in cities, among many other possibilities.

However, security is a key challenge in ITS for IoV since, nowadays, there exists a plethora of security attacks that can negatively impact its reliability. When investigating individual behaviors, and by considering that each vehicle or group of vehicles is controlled by rational entities, vehicles might misbehave maliciously or not. This problem is further exacerbated by the absence or impracticality of centralized control in vehicular environments. Conventional incentive schemes allow nodes to learn from their surroundings. For example, from

neighboring vehicles, and then assess the veracity of reported events and the honesty of vehicles. However, these schemes have their own vulnerabilities, such as false accusations and praise, which can be exploited to manipulate the entire network for the benefit of malicious vehicles/entities.

The projected number of vehicles in vehicular networks and the amount of information crossing them will be enormous. This may lead, for example, to congestion in the network. To avoid this problem, cooperation between vehicles is key. As cooperation is not always the norm in vehicular network, nodes need incentives to cooperate. This can be achieved through incentive schemes that can be game theory-based, reputation-based or remuneration-based.

As individuals we judge others based on their a past actions [1]. This concept when applied to networks can aid in the creation of an innovative system of reputation. The survival of the fittest is the rule in nature, and when applied to game-theory, we get EGT. Furthermore, by combining a reputation system based on past reputation and EGT, we can create an innovative incentive framework to improve cooperation among vehicles. Which is the purpose of this work.

II. BACKGROUND AND RELATED WORK

A. Vehicular networks

VANET [2] is a subclass of MANET. A set of mobile devices that communicate among themselves wirelessly is a MANET. In VANETs specifically, networks have no fixed infrastructure, so they use vehicles to provide network functionality. However, due to mobility constraints and driver behavior, VANETs exhibit characteristics that are very different from generic MANETs. VANET turns every participating vehicle into a wireless router, allowing vehicles to connect to each other, creating a wide range network. As vehicles leave signal range and drop out of the network, other vehicles can join said network, connecting vehicles to each other and creating a mobile Internet.

VANET only covers a very small mobile network, subject to mobility constraints and the number of connected vehicles. Traffic jams, bad driver behaviors, tall buildings and complex road networks, difficult the use of VANETs [3]. Even with all this "roadblocks", VANETs are well suited for short term applications and small scale services, like for example collision prevention or road hazard control notifications.

IoV on the other hand, focuses on the integration of elements such as humans, vehicles, things, networks, and environments to create an intelligent network based on computing and communication capabilities that supports services for large cities or even a whole country [4]. IoV has two main technology directions: vehicles' networking and vehicles' intelligence. Vehicles' networking consists of VANET, Vehicle Telematics (also called connected vehicles) and Mobile Internet. Vehicles' intelligence refers to the integration of driver and vehicle being more intelligent by using network technologies (swarm computing, deep learning, cognitive computing etc). The concept of VANET has steadily been evolving into IoV, making VANET a subset of IoV [3].

B. Incentive schemes

Incentive schemes/mechanisms can be used to manage and coordinate decentralized and self-managed systems, making up for the lack of a central or dedicated entity. A cooperative behavior may result in an increase of the nodes' resource consumption, for example forwarding nodes may spend additional energy and bandwidth during packet transactions. Even so, it was demonstrated that cooperation can succeed over defection [5]. The objective of incentive schemes is to guarantee that a cooperation brings overall more benefits than a passive or malicious uncooperative defectiveness. It may be desirable that these schemes differentiate defection due to valid reasons from malicious uncooperative ones. Cooperation schemes can be categorized as reputation-based, remuneration based and game-based schemes. Although some schemes may use elements from different categories.

III. FRAMEWORK INNER WORKINGS

A. Assumptions

Some assumptions are made in regard to the VDTN. It is expected that each network node has a unique identification that allows other network nodes to differentiate between them. This identifications cannot be forged or modified. In addition, messages include a list of the identifiers of the nodes they have passed by. Each node adds its identity, beginning with the source, when it receives a message that is not intended for itself, in a similar fashion to [6]. It is also assumed that these lists of nodes cannot be forged or modified, by including the appropriate certificates.

B. Reputation System

1) *Direct List*: Let us start by the direct reputation list. The objective of this list is to be a more reliable source of information, because it is not influenced by data provided by other nodes. When a node receives (receiver) a message, the receiver will check all of the other nodes the message passed through, including the source, and what is the destination node of the message. The receiver will never add itself to its own direct reputation list. Each forwarded message by a node will increase that node's reputation by reward value β . Forwarded messages are a sign that the node cooperated therefore giving it a reward in the form of an increase in reputation. For this

work we chose 0.075 for the reward value β , as this was the value that worked the best during the testing of the framework.

Source and destination nodes that are not in the list, are added to the list with a reputation of 0.75, which is the intermediate value of the reputation, and their good boy flag starts with value 1. We initiate the reputation of each node with 0.75 so the node can prove itself through its actions and does not start with any disadvantage. If the source or destination node was already in the list, no changes are made to either their reputation or good boy flag. Source and destination nodes are not doing a good deed in the eyes of the other nodes, therefore they do not have any improvement in reputation value, either when they are inserted in the list or when they are already in it. Their good boy flag starts as 1, to give them time to improve their reputation before it starts decreasing over time. The reputation only decreases when the good boy flag is equal to 0. We can see this in 1.

Let us assume the reward value β is equal to 0.075. Any node in the path of the message that is not a source node (path node) that is not in the list is added to the list with a reputation of 0.825, which is the the sum of the intermediate value of the reputation with the reward value β , and their good boy flag starts with value 2. If a path node was already in the list, the reward value β is added to their reputation and if their good boy flag is smaller than 2, the flag is incremented by 1, otherwise it remains the same. The path nodes, are nodes that have done a good deed (forwarding a message) in the eyes of other nodes, therefore they receive an improvement in reputation value. Their good boy flag is incremented by 1, when they are already in the list, to further reward them for their good deed. Their good boy flag starts, when they are not already in the list, to give them time to improve their reputation before it starts decreasing over time and reward them for their good deed.

2) *Indirect List*: To make sure the system is as fair as possible to all nodes, the scheme also has an indirect reputation list. When a node sends a message, it also sends its direct and indirect reputation lists. The node that receives the lists, updates its own indirect reputation list and the good boy flag of its direct reputation list.

First of all, like with the direct reputation list, the node never adds itself to the indirect reputation list. A receiver node will always check the direct reputation list of the sender first. It will run through this direct list and check for every node on this list that is not present on its own indirect list. For nodes not present in the receiver's indirect reputation list, the receiver will add the node with the corresponding reputation given by the sender. For nodes already present in the receiver's indirect list, the receiver will calculate the new value of reputation using an Exponentially Weighted Moving Average given by equation 1, where the value decided for α was 0.2. This way of updating the indirect reputation guarantees that past reputation is accounted for, and a node is not put at a disadvantage because of not participating in the network recently. This situation may happen, not because of the node is misbehaving, but because the network is very sparse.

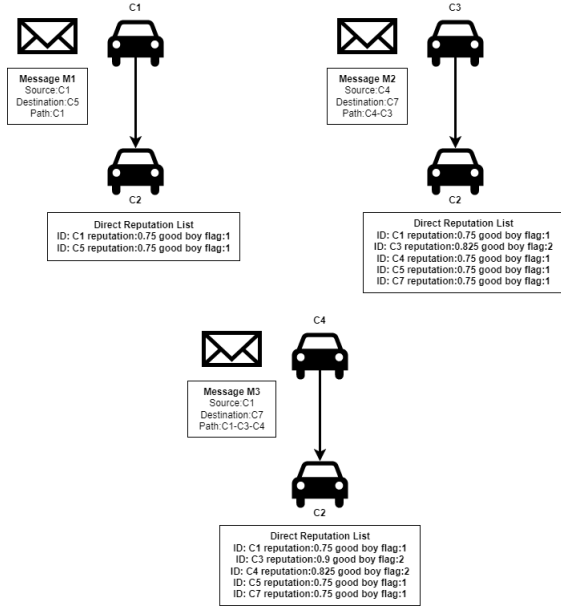


Fig. 1. Diagram exemplifying the increase in reputation in a direct reputation list

While checking the direct list of the sender, the receiver will also check if the nodes of the sender are in their own direct list. If this turns out to be true, the node will then check that node's good boy flag in its own reputation list and compare it to the good boy flag of that same node in the direct list of the sender. If the good boy flag equals 0 in the receiver's direct list and the good boy reputation flag is bigger than 0 in the sender's direct list, the receiver will increment that node's flag by 1. By doing this the nodes activities in the network are aware of the good deeds of more nodes, making the scheme more fair and making the system converge faster in terms of identifying misbehaving nodes.

Finally the node will check the sender's indirect list and check for nodes that were not already updated through the sender's direct list. If found, these nodes will be added in the same way the ones from the direct list were added: adding the node with the corresponding reputation if the node is not present, and using equation 1, where S is sender's new node reputation and R is the receiver's current node reputation, if it is. We can see this in 2.

$$\text{new reputation} = \alpha * \text{sender new node reputation} + (1 - \alpha) * \text{receiver current node reputation} \quad (1)$$

C. Evolutionary Game Theory System

If the reputation part of the system differentiates good nodes from bad nodes, the game theory part of the scheme is what makes the nodes act on their opinion of other nodes (in this case, a decision based on their reputation). This part of the system is based on EGT. It can be divided in the game played

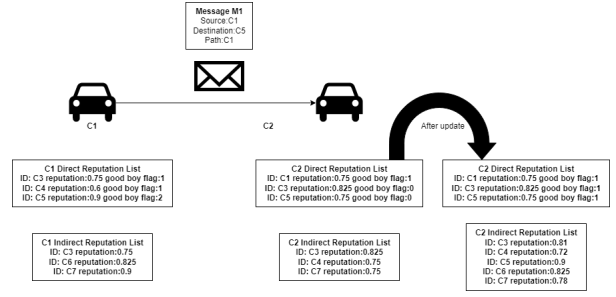


Fig. 2. Diagram exemplifying the update of an indirect reputation list

		Receiver	
		Accept	Refuse
Sender	Forward	R - F, R - A	-R, -R
	Hold	-H - R, -R	-H, -R

Fig. 3. Payoff matrix of the game

by the nodes (one-on-one interactions between nodes), the strategies used by the nodes and finally how this reputations change based on fitness of each individual. Nodes will divide themselves in populations and be constantly adapting to the network. A population is considered a group of nodes using the same strategy.

1) *The game:* When nodes have a choice when they send and receive messages from each other, and that is what makes this game theory approach possible. Each time nodes establish a connection between each other, they have two possible moves each. The sender node can either forward the message or hold on to it. The receiver node can either accept or refuse to receive the message. The game can be translated to the payoff matrix represented in 3.

In the matrix, R represents possible gains in reputation. As was explained in the previous section, the reputation of a node gets an increase in other nodes lists only when the node appears in the path of the message without being the source (or in other words when he forwards a message where he is not the source). This gain in reputation is **possible** and not guaranteed, because there is no way to guarantee the receiver is not a misbehaving node that will instantly drop the message after receiving it. H represents the cost of holding a message in the buffer, A represents the cost of accepting to store a message in the buffer and F represents the cost of forwarding a message. H, A and F use the energy and resources of the node, and that is why they are considered as negative payoffs in the matrix.

As we can see in 3, when the sender decides to forward a message he will have two possible outcomes: if the receiver accepts he will have the positive payoff of possible gains in reputations minus the cost of forwarding a message; if the receiver refuses it will have a negative payoff in possible gains

in reputation. On the other end, when the sender decides to hold onto a message he will always have the cost of keeping a message in its buffer, which consumes its resources, minus a negative payoff in possible gains in reputation.

In the case of the receiver when he decides to accept a message he will have two possible outcomes: if the sender forwards the message, the receiver will have the positive payoff of possible gains in reputations for forwarding the received message later minus the cost of accepting a new message in the buffer, which consumes resources; if the sender holds on to the message, the receiver will have a negative payoff in possible gains in reputation for forwarding the received message later. On the other end, when the receiver decides to refuse a message he will always have negative payoff in possible gains in reputation for forwarding the received message later.

2) *Strategies*: The strategies are the way nodes will decide how to choose between the options they have in the game. In the framework, these strategies are translated into the way nodes evaluate each other. In other words, strategies are the threshold values that nodes use to differentiate good nodes from misbehaving nodes.

The reputation varies between 0.5 and 1, while the strategies vary between 0.5 and 0.75 in intervals of 0.05. This means that nodes can use as a strategy the values: 0.5, 0.55, 0.6, 0.65, 0.7 and 0.75. If the reputation of the evaluated node is above the threshold it is considered a good node and if the reputation is below the threshold or equal to, it is considered a misbehaving node. When a node enters the network it a random strategy. The strategies only go until the intermediate value of reputation (0.75) to prevent unfairness, allowing nodes to build their reputation up, and keep it above most thresholds even when they do not participate in the network due to reasons out of their control. The variety of strategies keeps nodes on their toes, for if they slack off and do not cooperate with the network, their reputation may fall below thresholds of other nodes that will in turn stop cooperating with them. The process of decision is made in two steps.

The first step is the calculation of the reputation. This calculation is made by the evaluator node using reputation values from the direct and indirect lists.

When the evaluated node is present in both lists, reputation is calculated using equation 2, where w_d is the weight of the direct reputation and w_i is the weight of the indirect reputation. The chosen values for w_d and w_i , were 0.6 and 0.4 respectively. The direct weight is slightly larger than the indirect weight to prioritize direct information, which is more reliable because it is not as dependent on other nodes as indirect reputation.

When the evaluated node is present in only the direct list, the value on that list is used. When the evaluated node is present in only the indirect list, the value on that list is used.

When not present in any of the lists, the evaluated node is considered a good node. This is done in order maintain fairness and give all nodes the benefit of doubt when no data about them is available.

$$reputation = w_d * direct\ reputation + w_i * indirect\ reputation \quad (2)$$

The second step is the comparison between the threshold and the reputation of the evaluated node, and the decision based on this comparison. In the case of the sender, if the receiver's reputation is smaller or equal to the threshold, it will refuse to receive the message, otherwise it will accept it. In the case of the receiver, if the sender's reputation is smaller or equal to the threshold, it will hold on to the message, otherwise it will forward it.

3) *Average Work Ratio*: Average work ratio is the way nodes know if they need to change their strategy to adapt to the network, in other words it is how nodes evaluate how fit they are in relation to the network. In this case the average work ratio is an average of the work ratios of the various nodes in the network, given by the total numbers of received messages which destination was the node itself divided by the total number of forwarded messages given by equation 3.

$$work\ ratio = \frac{received\ messages}{forwarded\ messages} \quad (3)$$

The average work ratio, is calculated through consensus of the network in a decentralized manner, which means nodes share their own work ratio's with each other. This is made using the same method used for the indirect list. When a node sends a message it also sends its work ratio. When a node receives this ratio, if the average work ratio did not have a value yet, it becomes equal to the received ratio, otherwise it is calculated using an Exponentially Weighted Moving Average given by equation 4, where previous average work ratio and work ratio. The value chosen for α was 0.5.

$$average\ work\ ratio = \alpha * previous\ average\ work\ ratio + (1 - \alpha) * work\ ratio \quad (4)$$

Each node does a self evaluation every 60 minutes. It compares its own work ratio, with the average work ratio. If the value of its work ratio is bigger than the average work ratio, the strategy value is decreased by 0.05, unless the strategy value was already at the minimum. If the value of its work ratio is smaller than the average work ratio, the strategy value is incremented by 0.05, unless the strategy value was already at the maximum. If the value of the work ratio is equal to the average work ratio, the strategy value is maintained. This process allows the node to adapt itself to the network, by comparing how much work it is doing in relation to the other nodes in the network. If the node is doing more work, the strategy threshold gets bigger so the node forwards less messages. If the node is doing less work, the strategy threshold gets smaller so the node forwards less messages.

IV. METRICS

First things first, performance metrics are needed to compare and analyse the framework results. It is important to clarify

why they are used and how they are calculated in the ONE simulator [7]. The chosen metrics are divided in two main groups, the routing protocol performance metrics and the classification of nodes metrics.

A. Routing Protocol Performance Metrics

The Routing Protocol Performance Metrics metrics have the goal of evaluating how well the protocols perform when confronted with different percentages of misbehaving nodes. The delivery ratio represents the fraction of successfully delivered messages in comparison to the created messages. In the simulator, this value is calculated by the division of the total number of delivered messages by the total number of created messages, as shown in equation 5.

$$\text{delivery ratio} = \frac{\text{delivered messages}}{\text{created messages}} \quad (5)$$

To represent this effectiveness the metric good nodes delivery ratio is calculated. It is similar to the delivery ratio metric, but it uses the number of the delivered and created messages from good nodes to good nodes as shown in equation 6.

$$\text{good nodes delivery ratio} = \frac{\text{delivered messages from good nodes to good nodes}}{\text{created messages from good nodes to good nodes}} \quad (6)$$

In a similar fashion to the previous metric, the misbehaving nodes delivery ratio calculates the delivery ratio of a certain type of messages, this time the ones created by misbehaving nodes. This metric is calculated by dividing the number of delivered messages from misbehaving nodes by the number of created messages from misbehaving nodes as shown in equation 7.

$$\text{misbehaving nodes delivery ratio} = \frac{\text{delivered messages from misbehaving nodes}}{\text{created messages from misbehaving nodes}} \quad (7)$$

The average latency is the average time delay between the creation and delivery of messages. It is calculated by the division of the total sum of all the latencies (a latency is time delay between the creation and delivery of a message) by the total number of delivered messages as shown in equation 8. The same logic used for good and misbehaving nodes' delivery ratios applied in equations 6 and 7 respectively, is used for latency as represented in equations 9 and 10.

$$\text{average latency} = \frac{\sum_{i=0}^{\text{delivered messages}} (\text{delivery time}_i - \text{creation time}_i)}{\text{delivered messages}} \quad (8)$$

$$\text{good nodes average latency} = \frac{\sum_{i=0}^{\text{delivered messages from good nodes to good nodes}} (\text{delivery time}_i - \text{creation time}_i)}{\text{delivered messages from good nodes to good nodes}} \quad (9)$$

$$\text{misbehaving nodes average latency} = \frac{\sum_{i=0}^{\text{delivered messages from misbehaving nodes}} (\text{delivery time}_i - \text{creation time}_i)}{\text{delivered messages from misbehaving nodes}} \quad (10)$$

The overhead ratio indicates the exceeding number of transmitted messages in relation to the number of delivered messages as shown in equation 11. The bigger this value, the more excess messages are being transmitted, which is not always a bad thing. For example when reputation is being passed through messages, a high overheard ratio equates to more information available for all nodes. The same logic used for good and misbehaving nodes' delivery ratios applied in equations 6 and 7 respectively, is used for overhead as represented in equations 12 and 13.

$$\text{overheard ratio} = \frac{\text{transmitted messages} - \text{delivered messages}}{\text{delivered messages}} \quad (11)$$

$$\text{good nodes overheard ratio} = \frac{\text{transmitted messages from good nodes to good nodes} - \text{delivered messages from good nodes to good nodes}}{\text{delivered messages from good nodes to good nodes}} \quad (12)$$

$$\text{misbehaving nodes overheard ratio} = \frac{\text{transmitted messages from misbehaving nodes} - \text{delivered messages from misbehaving nodes}}{\text{delivered messages from misbehaving nodes}} \quad (13)$$

B. Node's Reputation Metrics

The Node's Reputation metrics have the goal of evaluating how well good nodes classify other nodes when confronted with different percentages of misbehaving nodes. The good node average direct reputation indicates the average reputation from the direct reputation list of good nodes when evaluated by good nodes. It is calculated by the sum of all the reputations given by good nodes to good nodes divided by the total number good nodes squared as shown in equation 14. The good node average indirect reputation indicates the average reputation from the indirect reputation list of good nodes when evaluated by good nodes. It is calculated by the sum of all the reputations given by good nodes to good nodes divided by the total number good nodes squared as shown in equation 15.

$$\text{good node average direct reputation} = \frac{\sum_{i=0}^G \sum_{j=0}^G (\text{node direct reputation}_{ij})}{G^2} \quad (14)$$

$$\text{good node average indirect reputation} = \frac{\sum_{i=0}^G \sum_{j=0}^G (\text{node indirect reputation}_{ij})}{G^2} \quad (15)$$

The misbehaving node average direct and indirect reputation indicates the average reputation from the direct and indirect reputation lists of misbehaving nodes when evaluated by good nodes. They are calculated in a similar way to the good nodes average reputations but are calculated by the sum of all the

reputations given by good nodes to misbehaving nodes divided by the total number good nodes multiplied by the total number of misbehaving nodes as shown in equations 16 and 17.

$$\text{misbehaving node average direct reputation} = \frac{\sum_{i=0}^G \sum_{j=0}^B (\text{node direct reputation}_{ij})}{G * B} \quad (16)$$

$$\text{misbehaving node average indirect reputation} = \frac{\sum_{i=0}^G \sum_{j=0}^B (\text{node indirect reputation}_{ij})}{G * B} \quad (17)$$

The false good node direct classification ratio is the ratio of good nodes mistakenly evaluated as misbehaving nodes by good nodes in the direct reputation list in comparison to all the evaluations made of good nodes by good nodes. This metric is important because if bad behaviour is being punished, it measures the number of good nodes that are being punished unfairly. According to this ratio, a node is considered as a misbehaving node when its reputation reaches 0.75 in the direct list. 0.75 being the threshold below which nodes start to be punished. As the reputation is in constant change, a node may be wrongly evaluated as a misbehaving node in one moment and due to its actions be evaluated as good in another. This metric is calculated in the simulator using equation 18, where G is the total number of good nodes in the network. The same logic is applied to the indirect list in 19.

$$\text{false good node direct classification ratio} = \frac{\sum_{i=0}^G (\text{number of good nodes with direct reputation smaller than } 0.75)_i}{G^2} \quad (18)$$

$$\text{false good node indirect classification ratio} = \frac{\sum_{i=0}^G (\text{number of good nodes with indirect reputation smaller than } 0.75)_i}{G^2} \quad (19)$$

The reverse logic is applied to the ratio of misbehaving nodes mistakenly evaluated as good nodes by good nodes is applied to create the false misbehaving node direct and indirect classification ratios, given respectively by 20 and 21, where G is the total number of good nodes in the network and B is the total number of misbehaving nodes in the network.

$$\text{false misbehaving node direct classification ratio} = \frac{\sum_{i=0}^G (\text{number of misbehaving nodes with direct reputation bigger than } 0.75)_i}{G * B} \quad (20)$$

$$\text{false misbehaving node indirect classification ratio} = \frac{\sum_{i=0}^G (\text{number of misbehaving nodes with indirect reputation bigger than } 0.75)_i}{G * B} \quad (21)$$

V. SIMULATION SCENARIOS

After idealizing, testing and tuning the framework, choosing the simulation scenario in which to run various tests is the logical step ahead.

The ONE simulator offers many options in the regard of simulation scenarios. The main reasons for the choices made ahead in terms of the scenario were for ease of comparison with other works and to test how robust the framework is.

The number of nodes in a network has a significant impact on the number of connections made between nodes. As a result, the amount of transferred messages increases with the number of nodes. On a map of the same size, fewer nodes may result in a more sparse network, whereas more nodes may result in a more dense network. Therefore, there are fewer possibilities to exchange messages in a network with fewer contacts. But with fewer messages, the buffers of the node are not as full, allowing them to hold messages for longer without dropping them. The exact opposite occurs in dense networks. More contact chances, more messages shared, hence, more messages are dropped due to congestion increase.

Considering the map of Helsinki used in the simulations, for a sparser network it was decided that 40 nodes would be adequate. And 120 nodes for a denser network. An intermediate value of 80 nodes was also used to observe how the framework behaves in a network that is neither sparse nor dense.

As with the number of nodes in the network, the profile of the network has an effect on the transmission of messages. The amount of cars, pedestrians, and trams in every simulation has an effect on the various metrics, due to the fact that each group moves at a different rate and is restricted to certain locations. It was decided to incorporate all three kinds of nodes to more accurately simulate the diversity that a next generation vehicular network may have. All simulations have the same number of trams, while the rest of the nodes are equally divided between cars and pedestrians. It was also decided that a random node would generate a message every 5 to 10 minutes. This low rate of message creation was chosen in order to test the robustness of the framework when faced with smaller numbers of messages and minimize the occupation of the buffers. The movement model chosen for this work was the Shortest Path Map-Based Movement model. In this model nodes follow the shortest path between the current location and the destination.

With the simulation scenario set, it is necessary to test the effectiveness of the framework when it encounters situations it was made to minimize. In this work we simulated two different types of DoS attacks: black-hole attacks and gray-hole attacks. The routing protocols tested in this simulations were SnW in binary mode, Epidemic [8], PRoPHET [9] and Maxprop [10].

VI. ROBUSTNESS RESULTS ANALYSIS

For the black-hole node attacks we tested each scenario described previously, with 4 different routing protocols (SnW in binary mode, Epidemic, PRoPHET and Maxprop), and with different percentages of misbehaving nodes (in this case, black-hole nodes). The simulations were made with 4 different

percentages of misbehaving nodes: 20%, 40%, 60% and 80%. The simulations were made with the reputation framework implemented and with it not implemented, to see how better the protocol's performance is with the reputation framework implemented.

For the gray-hole node attacks, we tested each scenario described previously, with 4 different routing protocols (SnW in binary mode, Epidemic, PRoPHET and Maxprop), always with 20% of the total nodes in the network being misbehaving nodes (in this case, gray-hole nodes). As gray-hole nodes do drop a fixed percentage of messages, the simulations were made with 2 different percentages of misbehaving node dropping probability: 50% and 90%. The simulations were made with the framework implemented and with it not implemented, to see how better the protocol's performance is with the framework implemented.

Before comparing the impact of the framework in the protocols performance, we analyse the reputation metrics and the behaviour of nodes when using the framework. This behaviour is translated in how they change strategies over time. All of the analysis done in terms of reputation and behaviour only considered the good nodes reputation lists and strategies because although misbehaving nodes still have their own reputations lists and strategies, they do not use them in any way to make decisions. We need to look at how fast the reputations of the good and misbehaving nodes diverge and how big false bad nodes ratio (good nodes evaluated as bad) is, because the false good nodes ratio (misbehaving nodes evaluated as good) will always be reduced to 0 very quickly. As all misbehaving nodes start with a reputation of 0.75, the moment the reduction of reputation with time kicks in, they will be considered misbehaving nodes. The divergence of the reputations is directly proportional to the number of messages generated (copies included) and to the number of hops each message makes in route to its destination. The more messages and the more hops per message, the faster the divergence in reputation.

In terms of distinction between good and misbehaving nodes the framework benefits from more nodes in the network and a higher number of messages in circulation. When looking at the average reputations over time, the higher the number of messages in circulation and nodes, the faster the framework diverges in terms of good and misbehaving nodes' reputations. This explains the slow divergence time with SnW routing protocol, where the overhead is very limited, so the number of message copies circulating through the network is also very low. This leads to the nodes not sharing information fast enough for the system to diverge quickly and also leads to a less accurate classification of said nodes. We can observe the exact opposite with the Epidemic routing protocol, where the overhead is very high, which leads to a faster and more accurate classification of the nodes.

The systems distinction of nodes gets progressively worse the less a misbehaving node misbehaves. When looking at the average reputations and false ratios, these get worse the smaller the probability of misbehaving nodes dropping

messages get. The best results are observed when nodes drop 100% of messages in black-hole attacks, and the worst results are observed when nodes drop 50% of messages in gray-hole attacks.

The results in the distinction of good and misbehaving nodes reflect directly on the routing protocol performance. The framework punishes misbehaving nodes by making other nodes refuse to accept messages misbehaving nodes want to send to a certain destination. This will not have a great impact in the delivery ratios, because nodes eventually find their destination and deliver the message directly, without resorting to the cooperation of other nodes. The framework most noticeable changes are for the overhead and latency. The variations in delivery ratio, overhead ratio and latency average for good and misbehaving nodes with the framework in comparison to the same scenarios without framework are represented in I and II.

We can see very little change in the delivery ratio, as expected. In the case of black-hole attacks, there is a clear reduction in the Epidemic and PRoPHET routing protocol's misbehaving nodes delivery ratio while for the other protocols it remains the very similar to when the framework is not used.

The overhead gets an overall reduction when using the framework. This reduction was to be expected. But the results prove that cooperation among good nodes improves and cooperation with misbehaving gets worse. With the exception of SnW, where the changes are barely noticeable, all protocols see the good nodes overhead decreasing slightly. Meaning copies of messages that would go to misbehaving nodes are not being created because good nodes are refusing interact with misbehaving nodes. In the case of misbehaving nodes overhead, it gets a very big reduction, much bigger percentage wise than the reduction for the good nodes overhead. Meaning good nodes are refusing to interact with misbehaving nodes. This improvements in overhead get less noticeable, the less misbehaviour, misbehaving nodes presents, which leads to better results for black-hole attacks, and worse results for gray-hole attacks with less message drop probability.

In the case of latency, the good nodes latency, with the exception of the SnW routing protocol stays the similar or gets slightly less when the framework is at its best. The big difference observed comes with the misbehaving nodes latency. This latency gets much bigger the better the framework is working and when the majority of nodes in the network are good nodes. This is due to the fact that by refusing to cooperate with misbehaving nodes, the misbehaving nodes have to deliver their messages directly without resorting to hops between other nodes, making the time necessary to deliver said message, much higher. This is also happens to good nodes when misbehaving nodes are the majority. Although in a lesser degree good nodes latency gets bigger because they refuse to interact with misbehaving nodes, which make up most of the network.

Finally, we look at the strategies each node uses in the EGT-based part of the framework. In general, most nodes tend use the maximum threshold of 0.75 or the minimum threshold of

Protocol	Number of Nodes	Variation in Good Nodes Overhead Ratio [20%/80%]	Variation in Misbehaving Nodes Overhead Ratio [20%/80%]	Variation in Good Nodes Latency Average [20%/80%]	Variation in Misbehaving Nodes Latency Average [20%/80%]	Variation in Good Nodes Delivery Ratio [20%/80%]	Variation in Misbehaving Nodes Delivery Ratio [20%/80%]
Spray and Wait	40 nodes	+4.4%/+3.3%	-1.1%/-2.1%	-11.4%/+2.6%	+11%/+4.4%	-2%/+9.8	-1.6%/-1.7%
	120 nodes	+5.2%/+18.4%	+2.1%/+5.2%	-11.5%/-16.6%	+18%/+1.8%	+1.1%/+1.6%	-2.6%/-4.6%
Epidemic	40 nodes	-27.6%/-20.9%	-70%/-71.7%	-5.6%/+384.8%	+81.3%/+6.2%	+1.4%/+79.6%	-25.3%/-10%
	120 nodes	-23.3%/-30.3%	-85.8%/+76.4%	-7.6%/+47.5%	+347.3%/+69.6%	+9.1%/+69%	-8.7%/-11.9%
PRoPHET	40 nodes	-11.8%/-18.1%	-58.6%/-56%	+1.4%/+19%	+43.1%/+33.3%	+4.7%/+9.9%	-16.1%/-21.7%
	120nodes	-16.1%/-57.7%	-74.2%/-59.7%	-6.8%/+10.4%	+55.4%/+0.5%	+5%/+0%	-13.4%/-14.3%
MaxProp	40nodes	-19%/-41.7%	-47.6%/+45.3%	+7.4%/+46.3%	+54%/+5.2%	-0.4%/+2.2%	-16.9%/-18.9%
	120 nodes	-36.6%/-51.2%	-68.5%/-52.3%	+26.6%/+33.2%	+451.9%/+56.8%	-0.2%/+1.7%	-14.1%/-12.9%

TABLE I

PERFORMANCE SUMMARY FOR DIFFERENT PROTOCOLS IN SCENARIOS WITH DIFFERENT NUMBERS OF NODES WITH 20% AND 80% MISBEHAVING NODES, TO CHECK ROBUSTNESS OF THE REPUTATION FRAMEWORK

Protocol	Number of Nodes	Variation in Good Nodes Overhead Ratio [50%/90%]	Variation in Misbehaving Nodes Overhead Ratio [50%/90%]	Variation in Good Nodes Latency Average [50%/90%]	Variation in Misbehaving Nodes Latency Average [50%/90%]	Variation in Good Nodes Delivery Ratio [50%/90%]	Variation in Misbehaving Nodes Delivery Ratio [50%/90%]
Spray and Wait	40 nodes	+2.1%/+4.1%	-0.8%/-5.3%	+6.7%/+3.7%	+15.18%/+10.6%	-4.2%/+3.2	-1.5%/-6.7%
	120 nodes	-0.1%/+3.2%	-0.5%/+2.3%	+21.6%/+1.1%	+8.33%/+13.6%	-1.4%/+0.7%	-2.3%/-5.9%
Epidemic	40 nodes	-7.5%/-10%	-8.4%/-39.3%	-1.3%/+1.3%	-3.9%/+39.5%	+8.7%/+1.6%	-4.4%/-14.4%
	120 nodes	-2.5%/-27.1%	+7.8%/-70%	-1%/+1.8%	-1%/+216.9%	-0.5%/+14.4%	-7%/-0.9%
PRoPHET	40 nodes	+1.2%/-37.3%	-6.3%/-42%	-2.6%/+0.8%	+11%/+1.8%	-2%/+7.9%	+0.5%/-15.4%
	120nodes	-8.7%/-19.5%	+3.3%/-59.3%	+0.7%/+4.1%	+6.2%/+145.7%	-1.5%/+5.8%	+18.9%/-6.5%
MaxProp	40nodes	-2.5%/-17.9%	-5.2%/-28.5%	+11.7%/+4.6%	+10.1%/+53.5%	+0.2%/+0.1%	+0.3%/-11.8%
	120 nodes	-26.1%/-34.4%	-18.1%/-50.3%	+43.2%/+41.1%	+98.3%/+334.1%	-0.9%/+0.2%	-2.3%/-17%

TABLE II

PERFORMANCE SUMMARY FOR DIFFERENT PROTOCOLS IN SCENARIOS WITH DIFFERENT NUMBERS OF NODES WITH 20% MISBEHAVING NODES WITH 50% AND 90% DROPPING PROBABILITY TO CHECK ROBUSTNESS OF THE REPUTATION FRAMEWORK

0.5. This is due to how the framework is implemented, nodes keep their threshold in the maximum until their work ratio is smaller than the network's work ratio and nodes keep their threshold in the minimum until their work ratio is bigger than the network's work ratio. This constant change of strategy thresholds stops the network from achieving an ESS, which was to be expected due to the ever changing conditions of the network.

VII. CONCLUSION AND FUTURE WORK

We can conclude that the use of indirect information in the framework makes the distinction between good and misbehaving nodes much faster and accurate. This is further enforced by the fact that the framework clearly benefits from protocols that use more hops, which makes the information of the various nodes be shared further and quickly. Following this, logic the framework benefits from denser scenarios with more nodes, because it allows for hops between nodes. We can also deduce that the framework works better when more messages are in circulation. This means that protocols with high levels of overhead are ideal for the framework and from scenarios with high message creation. Furthermore, although it behaves very well in the case of black-hole attacks, when confronted with gray-hole attacks the more well behaved the misbehaving nodes (the smaller the percentage of dropped messages), the harder it is for the framework to distinguish them from good nodes. When gray-hole nodes do not discard a very high number of messages, they can be confused as network congestion.

The framework does not have much of an impact on the delivery ratio of the nodes, either good or bad. This is due to the punishment for a node considered as misbehaving being the refusal of the message said node wants to forward.

Although this leads to improvements in the overhead and latency in the context of cooperation.

The best results obtained can be observed in the denser scenarios of 120 nodes, when 20% of those nodes are black-hole nodes, with the Epidemic protocol. The good nodes overhead ratio diminishes by 23.3% when using the framework, which equates to not forwarding messages to misbehaving nodes that would drop the messages. The good nodes overhead ratio diminishes by 85.8% when using the framework, which means good nodes are not cooperating with misbehaving nodes. The good nodes average latency when using the framework, has retains a similar value to the same scenario without the framework. The misbehaving nodes average latency when using the framework gets 3 times bigger than it was without the framework. This indicates good nodes are not cooperating with misbehaving nodes, which makes them have to directly deliver their messages increasing latency time by a lot. The worst results obtained can be observed when the framework work with SnW routing protocol. The low overhead of this framework leads to slower and less accurate distinction between good and misbehaving nodes, which produces bad results in terms of routing protocol performance.

Unfortunately, not all goals of this work were achieved and because of that and to develop this work, proposals for future work are suggested in this section. To improve on this particular framework there are some changes and tweaks that can be made.

On the reputation side of things, the framework can account for the size of the messages when calculating the reward value β . The bigger the message, the bigger the reward should be for forwarding it. This alteration would also cause the need for other changes in the reputation system, one of which would be to include the message size as a metric for the reduction

of reputation as well. This will make the framework more fair and add yet another dimension to it.

On the game theory side of things, testing a bigger spectrum of strategies may lead to different results that can be more valuable to the network. To add to this various punishments for misbehaving nodes may be applied like for example, dropping messages of misbehaving nodes or clear buffers of messages from misbehaving nodes when they are considered as such. This change in punishment may have lot of implications in the framework. This changes focus more on the action side of things and not so much on the detection part, which would clearly add another dimension to this work.

Lastly, on the testing part of things, testing the framework with a bigger number of nodes, testing how much more energy efficient the framework makes the good nodes in the network and different types of different model would be great. The ONE simulator although easy use, has poor scalability which makes hard to make simulations with a big number of nodes. These simulations would be great to understand the behaviour of the framework in very large scales. Testing the framework while using a movement model closer to real life would be great to test the real usability of the framework. A Real Trace Movement Model, where the nodes in the network follow movement directly taken from real life roads, would be a very interesting model to subject the framework to.

The great conclusion of this work being, that although not perfect yet, with more work and development this framework has real potential to be used to further develop vehicular networks in the future.

REFERENCES

- [1] F. P. Santos, F. C. Santos, and J. M. Pacheco, "Social norm complexity and past reputations in the evolution of cooperation," *Nature*, vol. 555, no. 7695, pp. 242–245, 2018.
- [2] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): challenges and perspectives," in *2006 6th international conference on ITS telecommunications*, pp. 761–766, IEEE, 2006.
- [3] F. Yang, S. Wang, J. Li, Z. Liu, and Q. Sun, "An overview of internet of vehicles," *China communications*, vol. 11, no. 10, pp. 1–15, 2014.
- [4] J. Contreras-Castillo, S. Zeadally, and J. A. Guerrero-Ibañez, "Internet of vehicles: architecture, protocols, and security," *IEEE internet of things Journal*, vol. 5, no. 5, pp. 3701–3709, 2017.
- [5] J. Zhang, V. Gauthier, H. Labiod, A. Banerjee, and H. Afifi, "Information dissemination in vehicular networks via evolutionary game theory," in *2014 IEEE International Conference on Communications (ICC)*, pp. 124–129, IEEE, 2014.
- [6] G. Dini and A. L. Duca, "Towards a reputation-based routing protocol to contrast blackholes in a delay tolerant network," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1167–1178, 2012.

- [7] A. Keränen, J. Ott, and T. Kärkkäinen, "The one simulator for dtn protocol evaluation," in *Proceedings of the 2nd international conference on simulation tools and techniques*, pp. 1–10, 2009.
- [8] A. Vahdat, D. Becker, *et al.*, *Epidemic routing for partially connected ad hoc networks*. Technical Report CS-200006, Duke University, 2000.
- [9] A. Lindgren *et al.*, "Probabilistic routing protocol using history of encounters and transitivity (prophet)," tech. rep., RFC 6693, IETF Document, 2012.
- [10] J. Burgess, B. Gallagher, D. D. Jensen, B. N. Levine, *et al.*, "Maxprop: Routing for vehicle-based disruption-tolerant networks.," in *Infocom*, vol. 6, Barcelona, Spain, 2006.