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

Rui Pedro da Costa Domingos

Thesis to obtain the Master of Science Degree in

Electrical and Computer Engineering

Supervisors: Prof. Paulo Rogério Barreiros D'Almeida Pereira
Prof. Naércio Magaia

October 2022

I declare that this document is an original work of my own authorship and that it fulfills all the requirements of the Code of Conduct and Good Practices of the Universidade de Lisboa.

Abstract

Vehicular networks that include Vehicle-To-Vehicle (V2V) and Vehicle-To-Everything (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 Denial of Service (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.

Keywords

Vehicular Networks; Incentive Schemes; Game-Theory; Reputation; Black-hole; Gray-Hole

Resumo

As redes veiculares que incluem conectividade Vehicle-To-Vehicle (V2V) e Vehicle-To-Everything (V2X), têm um enorme potencial em termos de melhoria da segurança e da experiência de condução. Essas redes usam esquemas de incentivo que permitem que os nós aprendam com a sua vizinhança, avaliem a veracidade dos eventos relatados, bem como a honestidade dos veículos e, claro, incentivam a cooperação. Esquemas de incentivo convencionais têm suas próprias vulnerabilidades, que são criadas principalmente por ataques de nós com mau comportamento. Neste trabalho, projetamos uma nova estrutura de incentivo baseada na teoria dos jogos evolutivos, considerando reputações passadas para melhorar a cooperação entre veículos e reduzir o impacto de ataques de Denial of Service (DoS). Esta estrutura é testada para quatro protocolos de routing e dois tipos diferentes de ataques, ataques de buraco cinzento e ataques de buraco negro. Os resultados das simulações nestes diferentes cenários mostram um bom desempenho em redes mais densas, para protocolos com elevados overheads e para cenários onde nós mal comportados nunca cooperam com a rede, nomeadamente com ataques de buracos negros. Mais importante ainda, há uma melhoria significativa na sobrecarga e na latência.

Palavras Chave

Redes Veiculares; Sistemas de incentivo; Teoria do jogo; Reputação; Buraco negro; Buraco Cinzento

Contents

List of Figures

Acronyms

AI	Artificial Intelligence
CAR	Context Aware Routing
CVN	Cooperative Vehicular Networking
DD	Direct Delivery
DoS	Denial of Service
DTN	Delay-Tolerant Network
EGT	Evolutionary Game Theory
ESS	Evolutionary Stable Strategy
FC	First Contact
IoV	Internet of Vehicles
ITS	Intelligent Transportation Systems
MANET	Mobile Ad Hoc Networks
ONE	Opportunistic Network Environment
PRoPHET	Probabilistic Routing Protocol using History of Encounters and Transitivity
PGG	Public Good Games
RCAR	Reputation-based Context Aware Routing
RE	Replicator Equation
RSU	Roadside Unit
SnW	Spray-and-Wait

TFT	Tit-for-Tat
TTP	Trusted Third Parties
VANET	Vehicular Ad Hoc Network
VDTN	Vehicular Delay-Tolerant Network
V2I	Vehicle-To-Infrastructure
V2V	Vehicle-To-Vehicle
V2X	Vehicle-To-Everything

1

Introduction

Contents

This chapter provides an overview of the concepts of Internet of Vehicles (IoV) and the motivation for this thesis. Furthermore, the objectives of this thesis are explicitly stated and the structure of the document is outlined.

1.1 Motivation

Nowadays, the development of communication systems has opened a plethora of new and exciting possibilities for the world of Intelligent Transportation Systems (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 Vehicular Ad Hoc Network (VANET) has evolved into the much broader concept of IoV. We can longer consider only Vehicle-To-Vehicle (V2V) communication. Vehicle-To-Everything (V2X) communication used in IoV, includes another set of variants to account for, like Roadside Unit (RSU)s, 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 [2]. 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 Evolutionary Game Theory (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.

1.2 Objectives

The aim of this project is to design a novel evolutionary game theory-based incentive framework by considering past reputations to improve [2] cooperation among vehicles and reduce the impact of Denial of Service (DoS) attacks. This will lead to the following objectives:

- Implementation of multiple scenarios considering various circumstances.
- Creation and implementation of the evolutionary game theory-based incentive framework to improve cooperation among vehicles and reduce the impact of DoS attacks.
- Creation of metrics to evaluate the performance of the framework.

1.3 Outline

The report will be structured in the following way:

- Chapter 2: This chapter focuses on reviewing the state of the art for vehicular networks, game theory and subsequently EGT. Following, there is framing of social norm complexity and past reputation in cooperation. To finish, there is a brief description of various simulators and the reasons why the ONE simulator was chosen.
- Chapter 3: This chapter presents the approach to the design of the framework. It starts with a brief description of the approach to the problem, followed by in-depth of each of the parts of the framework and it ends with the description of the metrics used to evaluate the performance of the framework.
- Chapter 4: This chapter's primary objective is to examine the framework's robustness. It begins with the presentation of how and why the simulations scenario settings were chosen. This is followed by the analysis of results against two different types of DoS attacks (black-hole and gray-holes attacks). It finishes with a reflection on how the nodes' strategies in the framework evolved throughout the scenarios.
- Chapter 5: This chapter provides an overview of the accomplishments achieved in this work and makes recommendations for further work on this topic.

2

State of the Art

Contents

2.1 Vehicular Networks

VANET [4] is a subclass of Mobile Ad Hoc Networks (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 [5]. 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 [6]. IoV has two main technology directions: vehicles' neworking 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 [5].

2.2 Vehicular Delay-Tolerant Networks

Contrary to other networks, Disruption- or Delay-Tolerant Network (DTN) does not assume the existence of an end-to-end path between nor unlimited connectivity to the Internet for mobile and fixed devices. DTN proposes a Bundle protocol, where a request message encapsulates all needed information for an application request and address resolution. All of the answer's components are also bundled in a response message. Once the the message we want to deliver is prepared, the issue of how to transport it via the network with no end-to-end connectivity remains. A store-carry-and-forward strategy is employed for this. Nodes store their messages when a contact opportunity is unavailable. After establishing contact, the message is sent and a copy may be saved. The node that received the message will transport it around the network until a new contact is established, repeating the store-carry-and-forward procedure until the message reaches its destination. Messages have an expiration time, that when exceeded makes the message be deleted to prevent overloading the network. Interestingly this kind of network was developed for interplanetary communication where delay tolerance is in great need, but it also has plenty of potential on Earth's surface [7].

A Vehicular Delay-Tolerant Network (VDTN) is a DTN in which the nodes are vehicles. Due to the significant applications that may be achieved with these networks, they have arisen as a hot area of study.

VANETs employ vehicles as nodes, however unlike VDTNs, they presuppose end-to-end connectivity through some route, which might be hard in high mobility or sparse environments, such as rural or mountainous locations. This is where the capabilities of the DTN come into play, as in these areas the traffic density is often low, resulting in fewer possibilities for contact, and the vehicles drive at a high rate of speed, resulting in brief encounters. VDTNs augment VANETs with delay-tolerant characteristics to tolerate lengthy network connectivity interruptions [7].

2.3 Routing In Delay Tolerant Networks

Due to the lack of end-to-end connection in VDTNs, routing cannot be performed in the traditional Internet sense. Bundle Protocol is utilized for forwarding in VDTNs. However, the protocol does not explain how to configure routes between nodes, an issue that has been the focus of several research over the years. Numerous routing protocols have been created to solve this situation. Some of those protocols will be described next, they will be: Direct Delivery (DD), First Contact (FC), Epidemic, Spray-and-Wait (SnW), Probabilistic Routing Protocol using History of Encounters and Transitivity (PRoPHET) and MaxProp.

2.3.1 Routing Protocols

DD and FC are Single-Copy routing protocols. Single-Copy routing protocols retain a single copy of a message throughout the network, passing it from node to node until it is erased or the destination is reached.

Epidemic, SnW, PRoPHET and MaxProp are Multiple-Copy routing protocols. Many-Copy routing protocols can have multiple message copies flowing throughout an entire network. The number of copies may be limited or unlimited. Messages may be flooded at random or based on some sort of information, and it is important to make the appropriate difference. In random flooding protocols, like Epidemic and SnW, no previous network knowledge is required to pass bundles. In information-based protocols, like PRoPHET and MaxProp, information such as previous interactions and delivery predictability can be utilized to generate a weighted decision [8].

Direct Delivery

In DD [9], the source stores and transports each bundle message until its destination is reached. No network information is required for the forwarding decision to be made. Compared to other protocols, DD uses less buffer space, less bandwidth, and has the least amount of overhead, as just a single transfer is performed per bundle. In this context, overhead is the number of successfully sent messages in excess of the total number of messages delivered. DD is unsuitable for sparse networks due to the fact that if the source and destination never meet, the message will never be sent. This drastically reduces the deliver probability.

First Contact

In FC [10], each bundle is delivered from the source to the first random node discovered, after which it is erased from the source's buffer, resulting in a random search for the destination. Due to the protocol's random nature, no network information is required to determine forwarding decisions. Unfortunately, this property makes FC extremely susceptible to potential bundle drops owing to insufficient buffer capacity or even hostile node attacks. Due to the random first contact policy, a packet may take a lengthy random path before reaching its destination, resulting in high latency values and a high ratio of overhead in comparison to DD. In comparison to flooding methods, it has a lower dropping rate, smaller buffer space occupancy, and reduced overhead.

Spay-and-Wait

SnW has two forms, the normal mode and the binary mode. In both forms, there are two distinct phases: the spray phase and the wait phase. For Spray-and-Wait normal mode, the spray phase consists of spreading a predetermined number of message copies, say X messages, to the first X passing nodes for each message coming from a source node. During the wait phase, each node holding a message copy conducts DD if the destination was not discovered in the previous phase. In Spray-and-Wait binary mode, the spray phase still begins with the source node containing X copies, but only transmits half of the copies to the first node it meets, holding the remainder for itself. If the destination cannot be located, each node having more than one message copy will continue to transmit half of its copies to the first contact opportunity until only one message copy remains in the buffer. Once this condition is met, the node executes DD [8].

Epidemic

In the Epidemic protocol [11], at every contact opportunity, nodes attempt to change the bundles that they do not share, flooding the network with an endless number of copies of each message in order for it to reach its destination. In a hypothetical system with limitless message storage capacity and long enough connections between nodes, Epidemic would minimize delivery latency and maximize delivery ratio. Nonetheless, this is not achievable in a physical context. The protocol still has a higher delivery rate than non-flooding protocols, but it ends up with a high number of dropped bundles and overhead, as flooding wastes storage and bandwidth.

PRoPHET

When two nodes are in communication using the PRoPHET protocol [12], a new message copy is only transmitted if the other node has a greater delivery probability to the target. This probability is determined using the encounter history and transitivity. The transitive property is based on the observation that if node A regularly encounters node B and node B frequently encounters node C, then node B is likely a strong candidate for forwarding packets meant for node A.

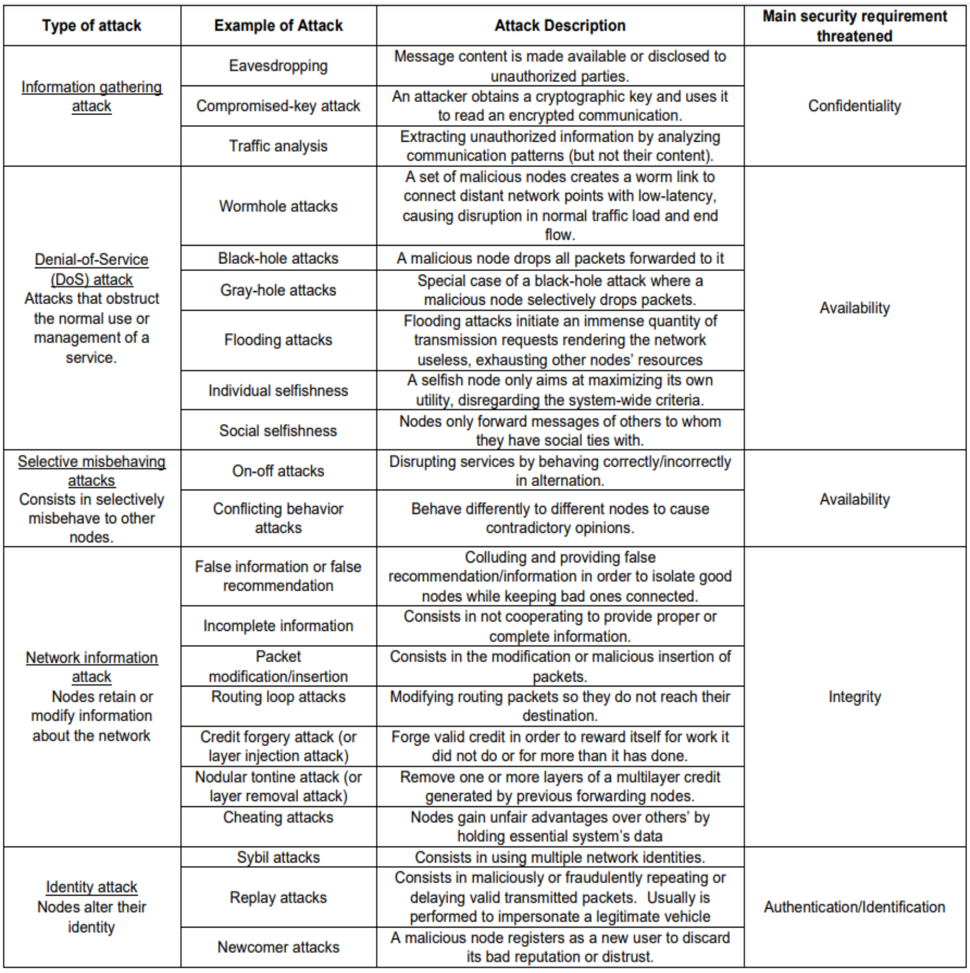
MaxProp

In Maxprop [13] when two nodes meet, they attempt to transmit any messages they do not already possess. Exchanging them depending on the number of message hops and the delivery probability calculated from previous encounters. Once the message reaches its destination, an acknowledgment is sent through broadcast to alert the encountered nodes to remove any existing copies of the delivered message, therefore reducing the burden of redundant message copies.

2.4 Types of attacks in Vehicular Delay-Tolerant Network

The protocols outlined in the preceding section illustrate that a degree of cooperation between nodes belonging to VDTNs is required for the delivery of messages. Unfortunately, nodes might misbehave because they are being controlled by a malicious user or by presenting selfish behavior. There is a big variety of attacks that can happen, and in most cases they make the network have a worse performance than it would otherwise.

In Figure 2.1 from [8] we can see a summary of this potential attacks. The classification of the attacks is based on the the Classification of Security Attacks in VANETs [14], as there is no fixed taxonomy for them. Confidentiality, integrity, authentication/identification, and availability are the assessed security requirements. Confidentiality addresses the problem of shielding data content from attackers, as only authorized parties should have access to network-transmitted data. Integrity is concerned with safeguarding the correctness of data, which means message contents should not be altered and information about the network sent between parties should be accurate. Authentication/identification is required for the appropriate identification of network participants. Availability serves to provide network resources to authorized entities.



Type of attack	Example of Attack	Attack Description	Main security requirement threatened
Information gathering attack	Eavesdropping	Message content is made available or disclosed to unauthorized parties.	Confidentiality
	Compromised-key attack	An attacker obtains a cryptographic key and uses it to read an encrypted communication.	
	Traffic analysis	Extracting unauthorized information by analyzing communication patterns (but not their content).	
Denial-of-Service (DoS) attack Attacks that obstruct the normal use or management of a service.	Wormhole attacks	A set of malicious nodes creates a worm link to connect distant network points with low-latency, causing disruption in normal traffic load and end flow.	Availability
	Black-hole attacks	A malicious node drops all packets forwarded to it	
	Gray-hole attacks	Special case of a black-hole attack where a malicious node selectively drops packets.	
	Flooding attacks	Flooding attacks initiate an immense quantity of transmission requests rendering the network useless, exhausting other nodes' resources	
	Individual selfishness	A selfish node only aims at maximizing its own utility, disregarding the system-wide criteria.	
	Social selfishness	Nodes only forward messages of others to whom they have social ties with.	
Selective misbehaving attacks Consists in selectively misbehave to other nodes.	On-off attacks	Disrupting services by behaving correctly/incorrectly in alternation.	Availability
	Conflicting behavior attacks	Behave differently to different nodes to cause contradictory opinions.	
Network information attack Nodes retain or modify information about the network	False information or false recommendation	Colluding and providing false recommendation/information in order to isolate good nodes while keeping bad ones connected.	Integrity
	Incomplete information	Consists in not cooperating to provide proper or complete information.	
	Packet modification/insertion	Consists in the modification or malicious insertion of packets.	
	Routing loop attacks	Modifying routing packets so they do not reach their destination.	
	Credit forgery attack (or layer injection attack)	Forge valid credit in order to reward itself for work it did not do or for more than it has done.	
	Nodular tontine attack (or layer removal attack)	Remove one or more layers of a multilayer credit generated by previous forwarding nodes.	
	Cheating attacks	Nodes gain unfair advantages over others' by holding essential system's data	
Identity attack Nodes alter their identity	Sybil attacks	Consists in using multiple network identities.	Authentication/Identification
	Replay attacks	Consists in maliciously or fraudulently repeating or delaying valid transmitted packets. Usually is performed to impersonate a legitimate vehicle	
	Newcomer attacks	A malicious node registers as a new user to discard its bad reputation or distrust.	

Figure 2.1: Summary of potential attacks to VDTNs

2.5 Cooperation in Vehicular Networking

Cooperative communication is an emerging technology, that refers to the processing of over-heard information at the surrounding nodes and retransmission towards the destination to create spatial diversity [15]. Cooperative communication technology plays an important role in improving the overall performance of vehicular networks.

With Cooperative Vehicular Networking (CVN) neighboring vehicles are able to cooperate through the sharing of information, giving them many transmission alternatives allowing for robust communication. Vehicles can cooperate with each other directly or any other elements in the network. An helper node or relay node, is a node that helps the node who sends the information by relaying said information. In CVN, nodes may not be willing to offer their services without incentive. The throughput of the network will be decreased if nodes misbehave by using attacks like the ones covered in the previous section.

Therefore, the development of effective incentive mechanisms for persuading the nodes to cooperate with each other, is crucial [16].

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 [17]. 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.

2.5.1 Reputation-based schemes

Reputation-based schemes are those in which the decision to interact depends on the reputation of other nodes.

The architecture of the reputation management system can be centralized, where a central authority collects information from the nodes, derives and provides the scores for all participant nodes; decentralized where node ratings storage is distributed; and hybrid which is a combination of the other two [1]. In a decentralized approach the evaluation of reputation is based on information from various subsets, which may cause inconsistencies when comparing with a more centralized approach. A distributed management system scales better, and is used in self-organized networks, contrary to the centralized one.

A reputation-based mechanism is composed of three phases (Figure 2.2) [1]:

- **Collection of evidence**: peer reputation is constructed based on the observation of the peer, experience with it, and/or recommendations from third parties. The collected information can be specific (evaluation of a specific functionality or aspect about a peer), general (evaluation of all functionalities about a peer), objective (or direct information, collected through personal interaction with a peer) and subjective (or indirect information, collected by listening to messages or negotiations between other nodes or by asking for their peers for opinions).

- **Cooperation decision**: based on the collected information, a node can make a decision on if it should cooperate with a peer based on that peer's reputation. There are many methods for computing the reputation of an entity like the Voting scheme (computing the sum of positive ratings minus the sum of negative ratings) or Average of ratings (computing the reputation score as the average of all ratings given by peers or users).

- **Cooperation evaluation**: a node must provide an evaluation of the degree of cooperation of the peer involved in the interaction. Peers behaving cooperatively, are rewarded by an increase in local reputation. A peer with a bad reputation will be excluded from the functionality provided by the entire group. Extra information about past interactions can be conveyed by the evaluation of current interactions.

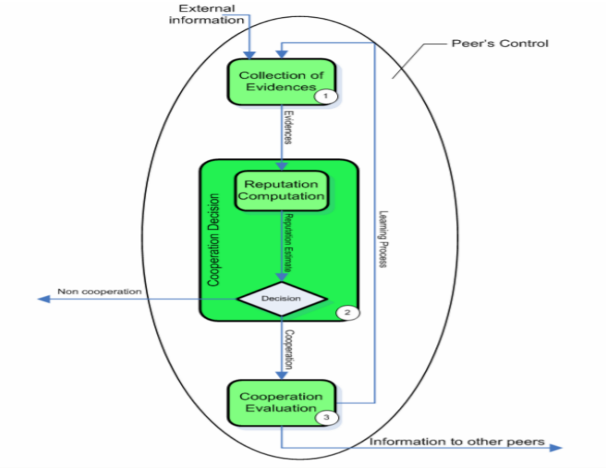


Figure 2.2: Reputation-based Mechanism [1]

An example of such scheme is Reputation-based Context Aware Routing (RCAR), proposed in [18]. It is an extension of Context Aware Routing (CAR) defined in [19]. RCAR is a decentralized reputation scheme where each node estimates the reputations of other based on its own network experience. Like CAR, RCAR is a probability based scheme. When forwarding a message, a node estimates the probability of selecting a node has a forwarding node, based on that node's reputation.

2.5.2 Remuneration-based schemes

Remuneration-based schemes are those in which well behaved nodes (nodes that cooperate) receive equivalent remuneration, and misbehaving nodes (nodes that do not cooperate) are punished with a penalty. This scheme requires Trusted Third Parties (TTP)(for example banks) to handle remuneration of good behaving peers.

A remuneration based mechanism has four main phases (Figure 2.3) [1]:

- **Negotiation**: the involved peers may negotiate the terms of the interaction. This negotiating gives more flexibility to this scheme. The negotiation can be performed either between the participating peers or between peers and a TTP.

- **Cooperation decision**: based on the outcome of the negotiation, the peer decides if wants or does not want to cooperate.

- **Cooperation evaluation**: cooperation is evaluated by the service requesting party, in terms of acceptability of the service to the request, and by the service providing party, in terms of proper remuneration. To guarantee the fairness of the evaluations, may require a TTP.

- **Remuneration**: the node that collaborated is remunerated. In this scheme, remuneration may consist of virtual currency units, real money or bartering units.

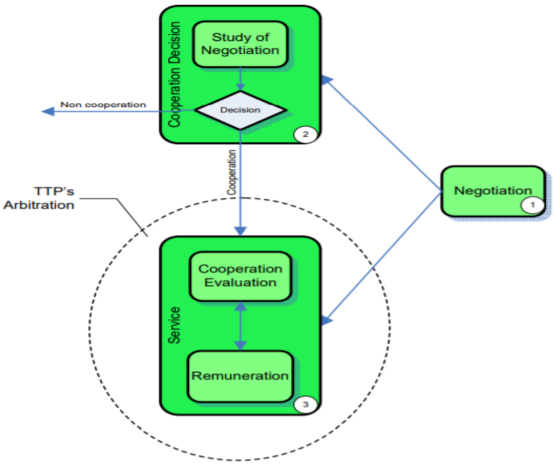


Figure 2.3: Remuneration-based Mechanism [1]

In remuneration systems, efficient cooperation incentive depends on creating a protocol that enforces a fair exchange of the remuneration. A fair exchange protocol provide ways to ensure that exchanges between nodes end in either all of them receiving what they were expecting or none of them receiving anything. The correctness of an exchange relies on the availability of a neutral TTP. These protocols can be either online, when the TTP mediates every exchange between the nodes, or offline, when the TTP mediates only if one of the nodes has doubts about the fairness of the interaction [1].

2.5.3 Game-based schemes

Game-based schemes are those in which forwarding decisions are modeled using game theory. More detailed in information on game theory can be found in section 2.6.

In these schemes, each node follows a strategy that tries to maximize the benefits and minimize resource consumption. For example, a node only decides to forward a message if the direct or indirect result of that action leads to the maximization of delivery probability [20].

An example o such schemes is a Tit-for-Tat (TFT) approach. In [21], the authors propose a pairwise TFT. TFT constraints state that the amount of traffic through a link, given pair of nodes, has to be equal in both directions. This specific approach uses generosity and contrition to deal with bootstrapping (when two nodes meet for the first time, basic TFT constraints will prevent relaying of packets, because no packets have ever been relayed with success by the other node). This routing protocol consists of the following elements:

- all nodes in the network change link state periodically.
- every source node computes forwarding paths based on link state, and uses source routing to send traffic.
- after receiving a packet each destination sends an ACK via flooding. The source node uses the ACK to update its TFT constraints for the next interval.

2.5.4 Comparison

Each scheme has its own advantages and drawbacks:

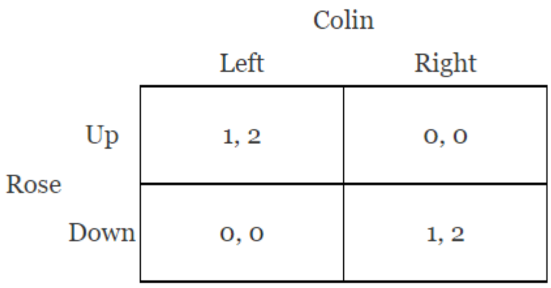
- **Reputation-based schemes**: Offers a trust management framework that improves cooperation but may require a TTP and nodes that misbehave can cause a big variety of attacks.
- **Remuneration-based schemes**: Presents a solid incentive for nodes to cooperate, but requires a TTP in order to function.
- **Game-based schemes**: Are highly efficient and offer great adaptability but are hard to evaluate when the number of nodes is very large.

2.6 Game Theory

Game theory, in its genesis, was a branch of mathematics used to study economics [22]. Nowadays and with rising interest in game-theory and its applications in computer science and Artificial Intelligence (AI), there are even discussions on if it actually works, the short answer being it depends [23].

Game theory's object of study are games which are well-defined mathematical objects. A game needs to have the following elements to be fully defined: the players, the information and actions available to each player, and the payoffs for each outcome. A game can either be cooperative or non-cooperative. A game is cooperative if the players can create externally enforce binding commitments. If players are unable to create alliances or all agreements must be self-enforcing, the game is non-cooperative. In classical game theory (non-cooperative game) there is a strong assumption that both players take rational decisions to maximize their gain reaching an ideal solution, the Nash Equilibrium. In game theory, each player has the same set of strategies. A strategy is a Nash Equilibrium if no player can do better than the other by unilaterally changing their strategy.

The normal game is represented by a matrix composed of the players, the strategies, and the payoffs. It can be represented in a broader sense by any function that relates a payout for each player with every possible action combination. In the example below (Figure 2.4), there are two players. One selects the row, while the other selects the column. Each player has two strategies, each of which is determined by the number of rows and columns. In this case Rose can chose between UP and Down strategies and Colin between Left and Right. The payoffs are given on the intersection of each column and row. The first number represents the compensation obtained by the row player (in this case, Rose), while the second represents the payoff received by the column player (in this case, Colin). For example, if Rose chooses Up as her strategy and Colin chooses Left as his strategy, Rose will get 1 as payoff and Colin will get 2 as payoff.



		Colin	
		Left	Right
Rose	Up	1, 2	0, 0
	Down	0, 0	1, 2

Figure 2.4: Payoff matrix of a 2-player, 2-strategy game

The developing of game-theory leads to its use in problems regarding computer science and engineering. The survey in [24] collects applications of game theory in wireless networking and presents them in a layered perspective, emphasizing on which fields game theory could be effectively applied.

2.6.1 Evolutionary Game Theory

EGT was originally used to study cooperative interactions in biological systems. It studies how the strategy adopted by a node evolves with interactions with other nodes. A strategy is considered a winning strategy when it shows its better performance in comparison to other strategies. EGT was inspired by evolution, for it was proposed to study populations with different strategies [25]. The three main components of EGT are the population, the game rules and the replicator dynamics, in biology terms, ”survival of the fittest”-the more apt specimens will spawn more offspring while weaker specimens will be eliminated. Two-key concepts of EGT are Evolutionary Stable Strategy (ESS) and Replicator Equation (RE).

ESS is a similar concept in evolutionary game theory to the Nash Equilibrium in Classical Game Theory. The strategy α employed by a certain population is evolutionary stable if the population that employs another strategy β always has less success. This means that if a population that employs α is invaded by part of a population that employs β, the population that employs strategy β will shrink to zero after some iterations as it is less successful in the environment compared with α [26].

REs are a common way to study the evolutionary dynamics. These show the growth rate of the proportion of individuals using a certain strategy and that rate is equal to the difference between the average payoff of that strategy and the average payoff of the population as a whole. RE, which translate the replicator dynamics of the population being studied, can show mathematically how the population evolves.

As we can see in Figure 2.5, EGT model has four phases:

- **Population X**: A population P(X) will have diversity among competing members. This diversity translates itself into a variety of different strategies to be used in the competition, which in this model is represented by a game.

- **Strategies and Game Rules**: The purpose of this game is to, under the rules of the game, test the strategies used by individuals. This different strategies lead to different payoffs, which translates into fitness (in nature, this would be the rate of reproduction).

- **Replicator dynamics**: Each member of the population undergoes replication or elimination based on the resulting fitness of each individual. This processes of replication or elimination are determined by REs. The end product of this replication dynamics will be a new population P(X+1), in which each individual has a new fitness determined by the results of the game.

- **Population X+1**: P(X+1) will return to the beginning of the cycle, and so on and so forth. This may eventually converge in a ESS.

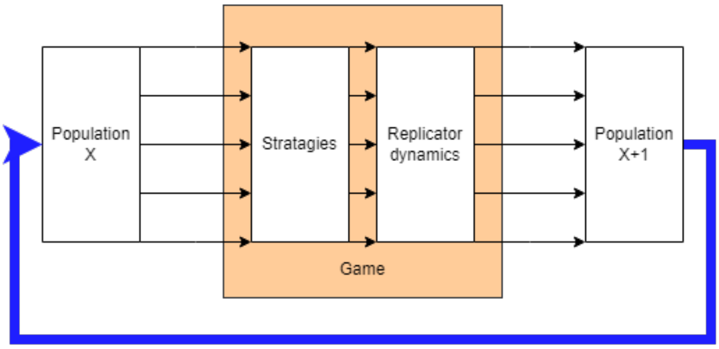


Figure 2.5: Evolutionary game theory scheme

2.6.2 Evolutionary Game Theory in Vehicular Networks

EGT is widely used in the study of wireless networks, for example, in network formation and dynamic routing. But more directed studies in the field of vehicular networks using EGT have started picking the interest of scholars, which led to some research results in the field.

[27] builds a VANET evolutionary game model and investigates the topic of VANET incentive co-operation. This work includes the formulation of packet forwarding games, in this case a Public Good Games (PGG). In a PGG, every node wants benefits and cooperating nodes contribute a certain cost to a public pool. The sum of all individual contributions is multiplied by a factor of r and equally divided among members of the group, independently of their contributions. When the value of r is increased, the benefits shared are also increased, which motivates nodes to cooperate. In this work, the game is formulated in the following way:

- At a given time instant, packets are generated by source nodes, which are initially considered cooperators. We assume that all nodes are interested in receiving the packets and a node may choose not to participate in forwarding, adopting a selfish behavior.
- The transmission of a packet in a single time slot is the unit cost, which is the maximum cost that a cooperator can contribute in a single time slot.
- Replicator dynamics of this game are achieved through the payoffs received in each round. At each instant of time nodes may adopt the strategies of a randomly chosen neighbor with higher payoff. This probability depends of the difference in payoffs between the two. The bigger the difference the higher the probability.

This model, combines a remuneration incentive scheme with EGT to achieve cooperation. It concludes that higher clustering of nodes favors cooperation due to difference in payoff being little and a reduced probability of nodes changing their strategies.

[17] looks into the VANET cooperative behavior and through simulation of the evolutionary game model. Similarly to [27], this work uses a PGG with the replicator dynamics of the game being that after each round of transmissions, the nodes have a probability to mimic the strategy of a neighbouring node, based on the difference in payoff between the two. The study shows that when adjusted properly, an EGT scheme can achieve a similar amount of information collection ratio as other schemes, while generating much less traffic.

[28] provides an evolutionary game model for the RSU bandwidth resource competition in Vehicle-To-Infrastructure (V2I) networks, and its evolutionary stable strategy is deemed to be the best answer for this competition. The EGT model in this work is constituted by:

- Player: the vehicle in the coverage area of RSU.

- Population: the vehicles that have the same strategy set.

- Strategy: the probability of a player i to access an RSU j.

- Payoff: the difference of the throughput of that strategy and the cost.

This work implements a game-theory scheme where each player has an amount of options equal to the number of available RSU plus the option of not accessing an RSU. The game is repeated, and in each iteration, each player observes the payoff of other players in the same network. Then, in the next iteration, the player adopts the observed strategy that gave the higher payoff.

[29] applies evolutionary game theory to model the evolution process of malicious users' attacking strategies in order to better evaluate reputation management schemes of IoV. The game works with 100 deceptive vehicles at all times, in a universe of 100000 vehicles. Deceptive vehicles are each given a strategy which is their deception intensity (average rate at which dishonest vehicles submit false traffic event messages to the central server). In the evolutionary game model used, natural selection (or replicator dynamics) is reflected in the fact that dishonest vehicles are removed from the network by the system as dishonest (when the system considers the deceptive intensity is too high). Reproduction is reflected in the fact that when the vehicle is removed from the network in the population, a new vehicle will join the population and enter the network. This reflects the reality that the owner of a dishonest vehicle may re-enter the network and continue to attack by replacing his account on the network. The strategy (deception intensity) of the new vehicle is given by an algorithm that chooses the most predominant strategy among attackers (deceptive vehicles). The work shows the evolution process of the attacking scenarios until a situation where most of the malicious vehicles finally get to their optimal attack strategy. It is concluded that the evaluation method can be used to quantify the effectiveness of a reputation management scheme of a vehicular network.

2.6.3 Social norm and past reputation

Considering indirect reciprocity, individuals expect a return from a third party and not from someone they helped directly. Helping, or not helping, the 'correct' individuals can enhance the likelihood of subsequently being helped by someone else. Social norms can be used to classify decisions. Each individual in a population uses the same social norm to assign reputations to other individuals. This reputation is attributed and disseminated by a bystander who witnesses interaction between two individuals.

To help the visualization of social norm, a donation game can be used, involving a 'donor' and a 'recipient'. In each iteration, the donor may cooperate, by helping the recipient at a cost to themselves while giving a benefit to the recipient (with the benefit being bigger than the cost), or defect (not providing help), in which case players do not get any costs or give any benefits. Reputation can be assigned to the players of the game based on certain criteria, thus forming a social norm (rules to assign reputation). Social norms can be divided into 4 orders:

- First-order norm: the assignment of a new reputation to the donor depends on their action towards the recipient
- Second-order norm: the assignment of a new reputation to the donor depends on their action towards the recipient and on the reputation of the recipient
- Third-order norm: the assignment of a new reputation to the donor depends on their action towards the recipient, on the reputation of the recipient and on the current reputation of the donor
- Fourth-order norm: the assignment of a new reputation to the donor depends on their action towards the recipient, on the reputation of the recipient, on the current reputation of the donor and on the previous reputation of the recipient.

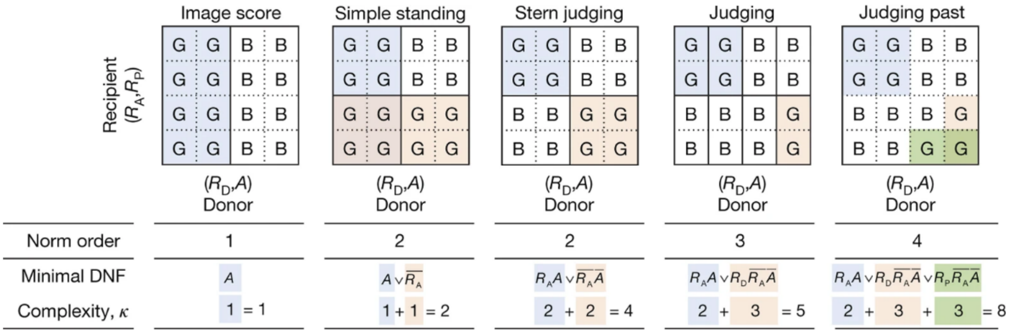


Image score				Simple standing				Stern judging				Judging				Judging past			
G	G	B	B	G	G	B	B	G	G	B	B	G	G	B	B	G	G	B	B
G	G	B	B	G	G	B	B	G	G	B	B	G	G	B	B	G	G	B	B
G	G	B	B	G	G	G	G	B	B	G	G	B	B	B	G	B	B	B	G
G	G	B	B	G	G	G	G	B	B	G	G	B	B	B	G	B	B	G	G

	Image score	Simple standing	Stern judging	Judging	Judging past
Norm order	1	2	2	3	4
Minimal DNF	A	$A \vee \overline{R_A}$	$R_A A \vee \overline{R_A}\,\overline{A}$	$R_A A \vee R_D \overline{R_A}\,\overline{A}$	$R_A A \vee R_D \overline{R_A}\,\overline{A} \vee R_P \overline{R_A}\,\overline{A}$
Complexity, κ	1 = 1	1 + 1 = 2	2 + 2 = 4	2 + 3 = 5	2 + 3 + 3 = 8

Figure 2.6: Reputation tables with various norm complexities [2]

A norm can be represented by a ‘reputation table’ like in Figure 2.6. Each entry in each table indicates the new reputation of the donor, which can be either good or bad (good, G; bad, B). This reputation can be based of the donor’s current reputation (R_D, which can be either good or bad), the donor’s action (A, which can be either cooperation C or defection D), and the current (R_A, which can be either good or bad) and past (R_P, which can be either good or bad) reputations of the recipient. Rows are ordered, from top to bottom, as $(R_A, R_P) = \{(G, G), (G, B), (B, B), (B, G)\}$ and columns are ordered, from left to right, as $(R_D, R_A) = \{(G, C), (B, C), (B, D), (G, D)\}$. The reputation in each table depends on what kind of norm order we use. For example, in the first table, there is the use of a first order norm where the only thing that matters is the action of the donor. Assuming cooperation gives a good reputation and defection gives you a bad one, we get the two first columns with a good reputation because the donor cooperates and the two last columns with with a bad reputation because the donor defects [2]. Above each table is the name of the norms used in said table:

- **Image score**: considers only A-if the donor cooperates it is good and if it defects it is bad.
- **Simple standing**:considers A and R_A-if the donor either cooperates or the recipient has good reputation, the donor is good.
- **Stern judging**:considers A and R_A-if a node cooperates and the recipient has a good reputation or defects and the recipient has a bad reputation, the donor is good.
- **Judging**: considers A, R_A and R_D-if the donor cooperates and the recipient has a good reputation or if the donor has a good reputation, defects and the recipient has a bad reputation, the donor is good.
- **Judging past**: considers A, R_A, R_D and R_P-if the donor cooperates and the recipient has a good reputation or if the donor has a good reputation, defects and the recipient has a bad reputation or the donor defects, the recipient has a bad reputation and a good past reputation, the donor is good.

In wireless networks the use of social norms can have huge impacts in cooperation among nodes. [30] presents a social norm based incentive scheme for mobile ad hoc networks. The incentive scheme in this work consists of a social strategy and a reputation scheme, where cooperative nodes are rewarded by an increase in their reputation which translates to better transmission quality in the future, and uncooperative nodes are punished by a decrease in their reputations which translates in the refusal of other nodes in forwarding their packets in the future. The reputation in this work is not binary like the example in the donation game of Figure 2.6. Instead, reputation has an attributed integer value that ranges from 0 to L. When an intermediate node performs well it adds 1 to their reputation until the maximum value of L. If it performs poorly its reputation will decrease by a certain value or decrease to 0, depending on the degree of deviation from a good behaviour.

2.7 Simulator

In order to design and implement a evolutionary game theory-based incentive framework for IoV, it is necessary to model this network and all the case studies necessary to develop it. There is yet to be a large enough infrastructure in place and readily available to test next-generation vehicular networks, due to the lack of devices equipped in vehicles and roads for this same purpose.

A simulator must be used. There are various simulators that can be used in the study of vehicular networks. [3] presents a review on the main simulators for Opportunistic Networks, as well as the advantages and disadvantages of those simulators. We can see an overview of this comparisons in Figure 2.7.

OMNet++ is a general simulator with several frameworks available and can be used to simulate many types of networks. According to [3], it also had a good performance and good documentation, as well as a very sophisticated user interface, with many possibilities for visualizing and inspecting various scenarios. This simulator does not offer the possibility to pass a simulation model to a real implementation. Mobility models are well supported and radio models adequately supported.

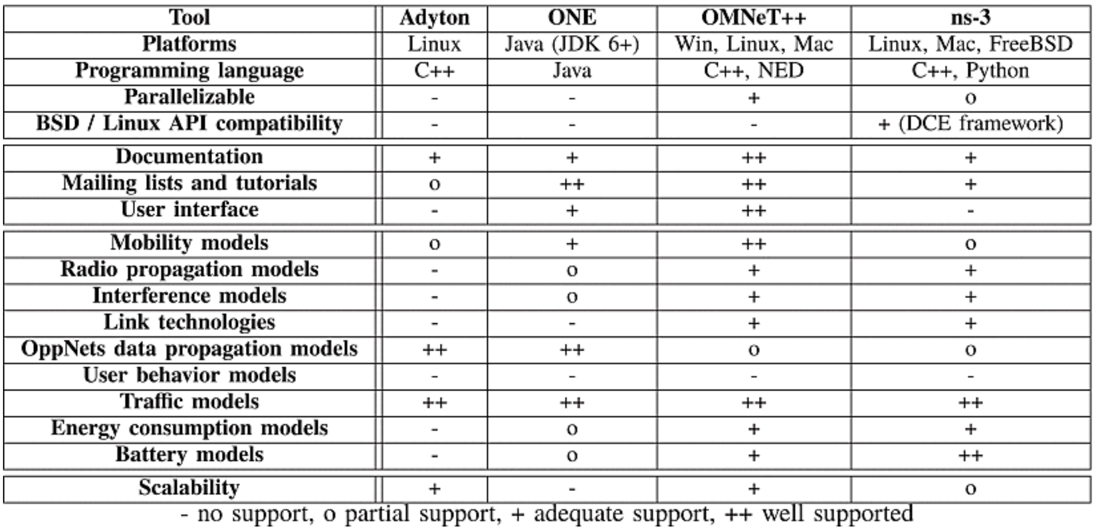
The ns-3 simulator is also a general simulator focused on the simulation of IP based networks. The main advantage of this simulator is that it is easy to move any simulation to the real world. The simulator is rather complex and lacks good user interface with debugging and visualization options, according to [3]. Mobility models are only partially supported and radio models adequately supported.

Opportunistic Network Environment (ONE) [31] is designed specifically for opportunistic networks. It has a graphical user interface, is easy to use and has a solid user community according to [3]. The ONE does not perform well for large simulations. Mobility models are adequately supported and radio models are only partially supported.

Adyton is also designed specifically for opportunistic networks, similar to ONE. It performs well in large simulations and has more available models. Adyton does not have a good graphical user interface, according to [3]. Mobility models are only partially supported and radio models are not supported.

The Veins [32] simulator is a vehicular network simulator based on OMNet++ and SUMO (a mobility traffic generator). Road traffic simulation is performed by SUMO, while network simulation is performed by OMNeT++. It offers a comprehensive suite of models to make vehicular network simulations as realistic as possible, without sacrificing speed. Veins has a graphical user interface, a solid and diverse user base and it contains a large range of simulation models.

Considering all the options above the chosen simulator was ONE [31]. It allows the creation of node motions using various models, the reproduction of message traffic and routing, cache management, and the viewing of both mobility and message passing via its strong graphical user interface. The ONE may also generate other data, including general message statistics and buffer occupancy reports. It is also as a small learning curve and is intuitive to use.



Tool	Adyton	ONE	OMNeT++	ns-3
Platforms	Linux	Java (JDK 6+)	Win, Linux, Mac	Linux, Mac, FreeBSD
Programming language	C++	Java	C++, NED	C++, Python
Parallelizable	-	-	+	o
BSD / Linux API compatibility	-	-	-	+ (DCE framework)
Documentation	+	+	++	+
Mailing lists and tutorials	o	++	++	+
User interface	-	+	++	-
Mobility models	o	+	++	o
Radio propagation models	-	o	+	+
Interference models	-	o	+	+
Link technologies	-	-	+	+
OppNets data propagation models	++	++	o	o
User behavior models	-	-	-	-
Traffic models	++	++	++	++
Energy consumption models	-	o	+	+
Battery models	-	o	+	++
Scalability	+	-	+	o

- no support, o partial support, + adequate support, ++ well supported

Figure 2.7: Comparison Between OppNets Simulators [3]

3

Design of the Reputation Framework

Contents

3.1 Approach to the problem

While creating an incentive framework, various factors need to be considered. The scheme of the framework and metrics need to be defined. First we need idealize the framework. We start with a version of the framework that works in theory and then tweak it as much as needed through testing and simulations. It also necessary to decide on which metrics to use in the evaluation of the performance of the network. After all the tests made and all the adjustments made, the framework is ready for further and more detailed testing which is present in Chapter 4.

3.2 Scheme for Framework

The scheme of the framework is the most important part of this project. It will define what we are trying to improve, what attack we are trying to stop or lessen the effects of, and what type of routing protocol we need to use.

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 [18]. It is also assumed that these lists of nodes cannot be forged or modified, by including the appropriate certificates.

First things first, we are trying to improve cooperation. To improve this cooperation we need an incentive. In this work this incentive will come from improvement in reputation, or in other words, the way nodes perceive each other. The better the evaluated node's reputation, the more the evaluator node will be willing to cooperate with that node. In this work, we choose to have a decentralized approached to the evaluation. This will be further explained in this section.

This framework will also try to prevent mainly DoS attacks from happening in the network. To do this the framework will need adaptability, and this adaptability will come from the game-theory elements introduced in it and further explained in this section.

To end this short summary of how the framework will work, the base routing protocol in which the framework works will need to be a multiple-hop routing protocol and the more overhead, the faster the framework will converge.

3.2.1 Overall Node Information

For the framework to work, nodes will have to account for certain data. This data will be covered in this section. The types of the data will be: the list of direct reputation, the list of indirect reputation, the number of received messages, the number of forwarded messages, the number of hops, the strategy and the fitness.

List of Direct Reputation

The list of direct reputation is a list of nodes that have their presence in the network acknowledged by this node. The information in this list is calculated by the node based on the observations made of other node's behaviours. Each node in the list has its unique identification, a reputation and a good boy value. The reputation is the value that translates how nodes evaluate each other. In the proposed framework this evaluation is not binary and can be between 0.5 and 1, where 1 is the best possible evaluation, meaning the node is considered good, and 0.5 the worst, meaning the node is considered misbehaving. The good boy value is a flag value that can be either 0, 1 or 2. The importance of this value will be explained in the next section.

List of Indirect Reputation

Much like the list of direct reputation, the list of indirect reputation is also a list of nodes that have their presence in the network acknowledged by this node. This list serves as a way to integrate other nodes reputation information that the node receives from other nodes. The differences between the lists are that the indirect list does not possess a good boy value and the reputation values are calculated in a different way that will be explained in the next section.

Number of Received Messages

The number of received messages is the total number of messages received by the node whose destination was that same node.

Number of Forwarded Messages

The number of forwarded messages is the total number of messages forwarded by the node, whose source is not said node.

Number of Hops

The number of hops, is the sum of the number of hops (how many nodes each message has passed through before getting to this node) of all the received messages received in the last hour, be it messages meant for that node or for another.

Strategy

The strategy is the reputation threshold at which a node considers another node as misbehaving. This will be further explained in the next section.

Work Ratio

The work ratio is given by the total number of received messages divided by the number of forwarded messages given by equation 3.1.

$$work\ ratio = \frac{received\ messages}{forwarded\ messages} \tag{3.1}$$

Average Work Ratio

The average work ratio is the average of the work ratios sent by other nodes. This will be further explained in the next section.

3.2.2 Reputation System

The reputation part of the scheme makes it possible to differentiate between misbehaving nodes and good nodes. There are two crucial parts to this system, how the direct reputation list is updated and how the indirect reputation list is updated.

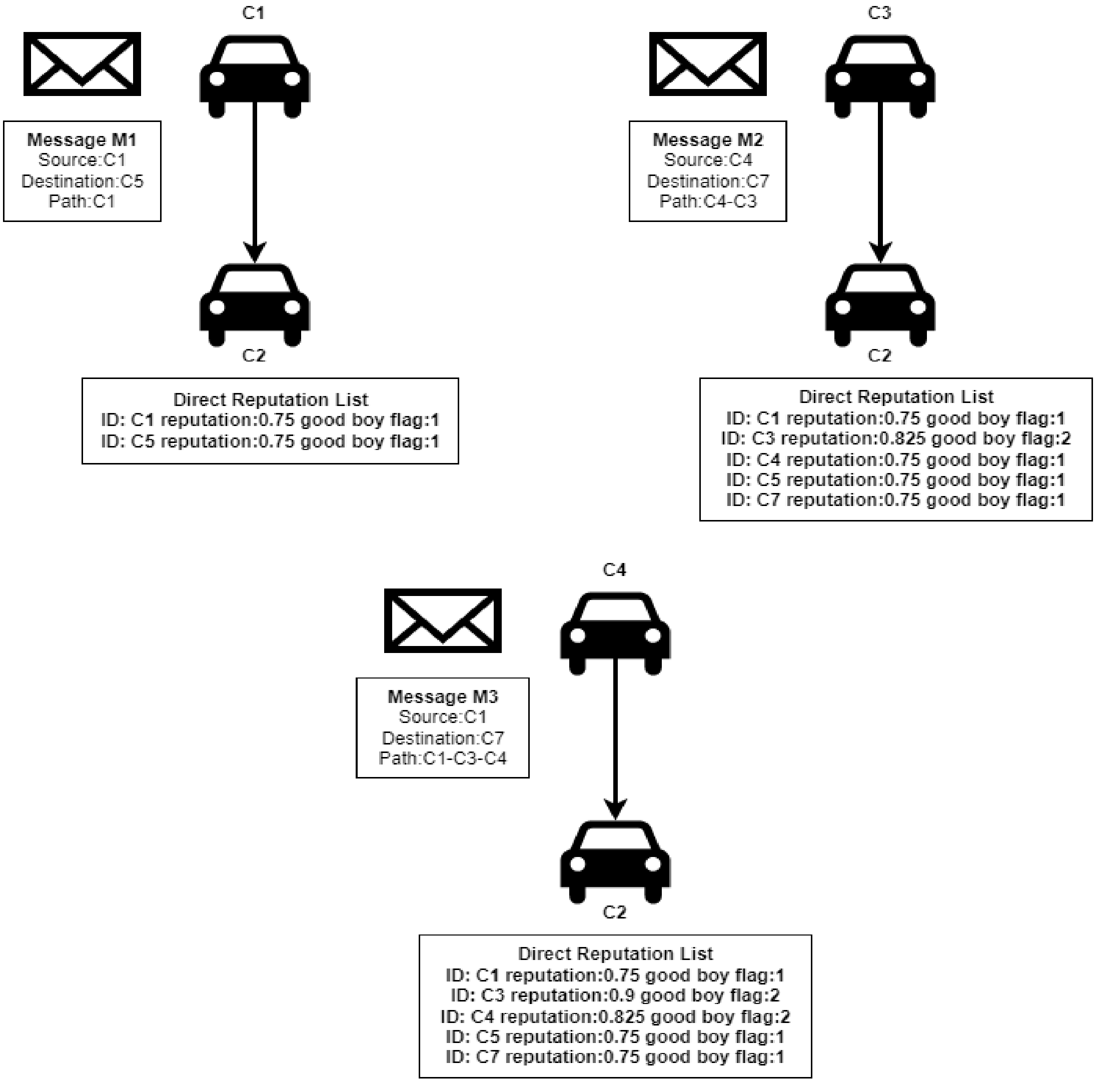
Direct Reputation List

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. This will be further explained in this section.

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.

Let us look at the example in fig. 3.1. In the case of message M1, admitting C2 had its direct reputation list empty, it will add the source node C1 and destination node C5 to the list with reputation 0.75 and good boy flag 1. In the case of message M2, C2 will add the source node C4 and destination node C7 to the list with reputation 0.75 and good boy flag 1. It will then add the path node C3 to the list with reputation 0.825 and good boy flag 2. In the case of message M3, C2 will make no changes to C1 and C7 as they are source and destination node and were already in the list. It will then add 0.075 to the reputation of both path nodes C3 and C4 as they were already in the list. It will also increment the good boy flag of C4 by one because it was smaller than 2 and it will not make any changes to the good boy flag of C3 because it was equal to 2.



Message M1
Source:C1
Destination:C5
Path:C1

Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C5 reputation:0.75 good boy flag:1

Message M2
Source:C4
Destination:C7
Path:C4-C3

Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C3 reputation:0.825 good boy flag:2
ID: C4 reputation:0.75 good boy flag:1
ID: C5 reputation:0.75 good boy flag:1
ID: C7 reputation:0.75 good boy flag:1

Message M3
Source:C1
Destination:C7
Path:C1-C3-C4

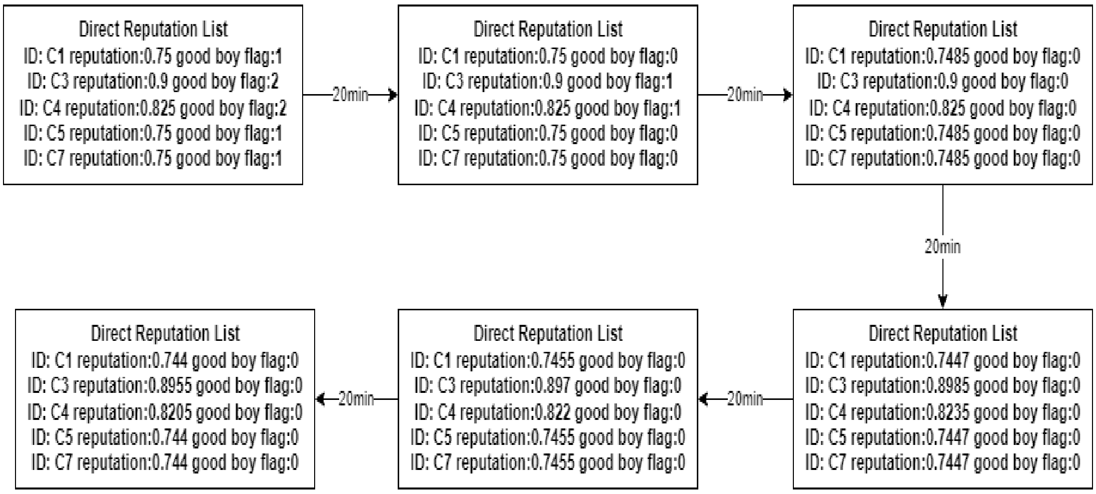
Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C3 reputation:0.9 good boy flag:2
ID: C4 reputation:0.825 good boy flag:2
ID: C5 reputation:0.75 good boy flag:1
ID: C7 reputation:0.75 good boy flag:1

Figure 3.1: Diagram exemplifying the increase in reputation in a direct reputation list

To make the direct reputation list the more secure against misbehaving nodes as possible, the reputation of nodes in the direct reputation list decreases with time. Every 20 minutes, the reputation of all nodes decreases if their good boy flag is equal to 0, otherwise their good boy flag decreases by 1. The good boy flag exists in order to reward nodes for forwarding messages and stopping their reputation from getting lower for not interacting with a specific node due to chance. The amount of reputation decrease is given by the equation 3.2, where ζ is the punishment value and the number of hops refers to the sum of the hops of all received messages in the last hour. After the decrease, if the reputation becomes smaller than 0.5 then reputation assumes the value of 0.5. For this work we chose 0.00015 for the punishment value ζ, as this was the value that worked the best during the testing of the framework.

$$decrease = -\zeta * number\ of\ hops \tag{3.2}$$

Let us assume that a node has 5 nodes on the direct reputation list, let us consider that the number of hops in all the received messages in the last hour is always 10, while the punishment value $zeta$ is 0.00015. Let also assume that the received messages do not alter any of the reputation values nor the good boy flags of the initial 5 nodes. The decrease in the 5 nodes reputations would be like what is represented in fig. 3.2. As we can see after the first 20 minutes, no decreases in reputation happen but every good boy flag is decreased by 1. After another 20 minutes only C1, C5 and C7 get a decrease in reputation because their good boy flags were equal to 0, while C3 and C4 get their good boy flag decreased by 1. In the all the subsequent intervals, all nodes get a decrease in reputation.



Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C3 reputation:0.9 good boy flag:2
ID: C4 reputation:0.825 good boy flag:2
ID: C5 reputation:0.75 good boy flag:1
ID: C7 reputation:0.75 good boy flag:1

Direct Reputation List
ID: C1 reputation:0.75 good boy flag:0
ID: C3 reputation:0.9 good boy flag:1
ID: C4 reputation:0.825 good boy flag:1
ID: C5 reputation:0.75 good boy flag:0
ID: C7 reputation:0.75 good boy flag:0

Direct Reputation List
ID: C1 reputation:0.7485 good boy flag:0
ID: C3 reputation:0.9 good boy flag:0
ID: C4 reputation:0.825 good boy flag:0
ID: C5 reputation:0.7485 good boy flag:0
ID: C7 reputation:0.7485 good boy flag:0

Direct Reputation List
ID: C1 reputation:0.7447 good boy flag:0
ID: C3 reputation:0.8985 good boy flag:0
ID: C4 reputation:0.8235 good boy flag:0
ID: C5 reputation:0.7447 good boy flag:0
ID: C7 reputation:0.7447 good boy flag:0

Direct Reputation List
ID: C1 reputation:0.7455 good boy flag:0
ID: C3 reputation:0.897 good boy flag:0
ID: C4 reputation:0.822 good boy flag:0
ID: C5 reputation:0.7455 good boy flag:0
ID: C7 reputation:0.7455 good boy flag:0

Direct Reputation List
ID: C1 reputation:0.744 good boy flag:0
ID: C3 reputation:0.8955 good boy flag:0
ID: C4 reputation:0.8205 good boy flag:0
ID: C5 reputation:0.744 good boy flag:0
ID: C7 reputation:0.744 good boy flag:0

Figure 3.2: Diagram exemplifying the decrease reputation of a direct reputation list over time

Indirect Reputation 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 is 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 checks 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 3.3, 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.

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 3.3 if it is.

$$new\ reputation = \alpha * sender\ new\ node\ reputation + (1-\alpha) * receiver\ current\ node\ reputation \tag{3.3}$$

Let us now consider the example in fig. 3.3. As we can see, the good boy flags of C3 and C5 in C2's direct reputation were incremented from 0 to 1 because their good boy flags in C1's direct reputation list were respectively 1 and 2. We can observe that C1's direct list has nodes C3, C4 and C5. In C2's updated indirect reputation list we can see that as C3 and C4 were already present in the list, so they are updated using equation 3.3, while C5 was added directly for it was not yet present. Now looking at C1's indirect reputation list we can that nodes C3, C6 and C7 are present. In C2's updated indirect reputation list we can see that as C7 was already present in the list, so it is updated using equation 3.3, while C6 was added directly for it was not yet present. As it was already updated through the C1's direct reputation list, C3's information remains the same.

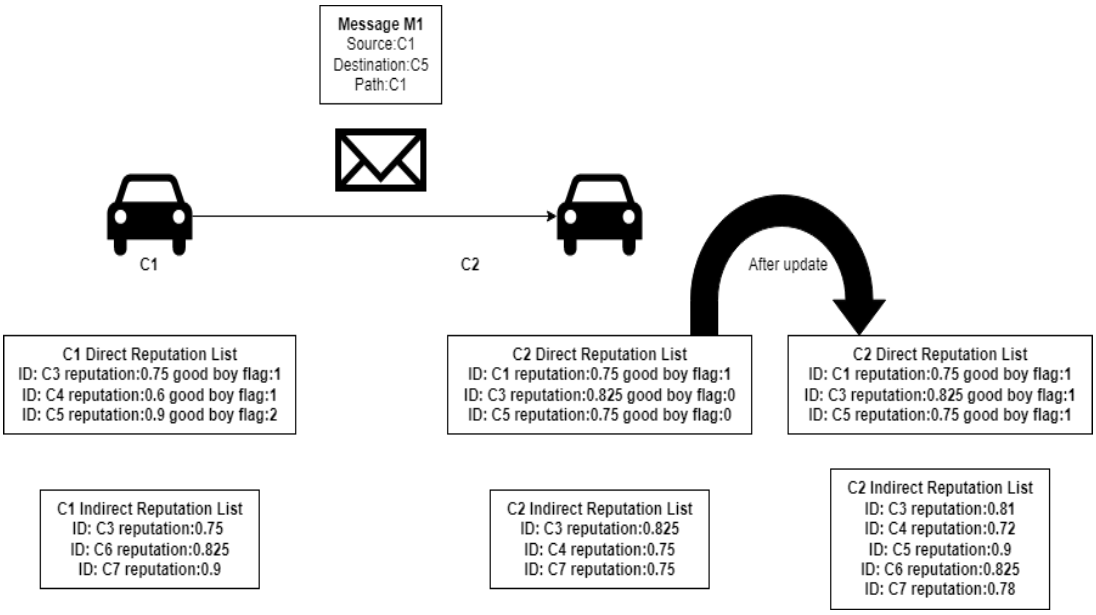

Message M1
Source:C1
Destination:C5
Path:C1
C1
C2
After update
C1 Direct Reputation List
ID: C3 reputation:0.75 good boy flag:1
ID: C4 reputation:0.6 good boy flag:1
ID: C5 reputation:0.9 good boy flag:2
C2 Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C3 reputation:0.825 good boy flag:0
ID: C5 reputation:0.75 good boy flag:0
C2 Direct Reputation List
ID: C1 reputation:0.75 good boy flag:1
ID: C3 reputation:0.825 good boy flag:1
ID: C5 reputation:0.75 good boy flag:1
C1 Indirect Reputation List
ID: C3 reputation:0.75
ID: C6 reputation:0.825
ID: C7 reputation:0.9
C2 Indirect Reputation List
ID: C3 reputation:0.825
ID: C4 reputation:0.75
ID: C7 reputation:0.75
C2 Indirect Reputation List
ID: C3 reputation:0.81
ID: C4 reputation:0.72
ID: C5 reputation:0.9
ID: C6 reputation:0.825
ID: C7 reputation:0.78

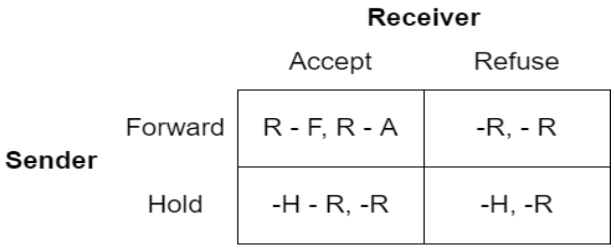
Figure 3.3: Diagram exemplifying the update of an indirect reputation list

3.2.3 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 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.

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 fig. 3.4.



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

Figure 3.4: Payoff matrix of the game

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 fig. 3.4, 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.

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 3.4, 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.

$$reputation = w_d * direct\ reputation + w_i * indirect\ reputation \tag{3.4}$$

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.

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 3.1.

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 3.5. The value chosen for α was 0.5.

$$average\ work\ ratio = \alpha * previous\ average\ work\ ratio + (1 - \alpha) * work\ ratio \tag{3.5}$$

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.

3.3 Reputation Framework Performance Metrics

First things first, performance metrics are needed to compare and analyse the framework results. Although some of the chosen metrics were explained in Chapter 2, it is important to clarify why they are used and how they are calculated in the simulator.

The chosen metrics are divided in two main groups, the routing protocol performance metrics and the classification of nodes metrics.

3.3.1 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.

Delivery Ratio

The delivery ratio represents the fraction of successfully delivered messages in comparison to the created messages. It is crucial in routing protocol assessment because the objective of said protocol is to maximize this metric. With an ideal scenario being the delivery of all 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 3.6. More than one copy of the messages can be created and therefore received in one of the various protocols. In this calculation, only the first copy to be created and the first copy to be delivered is considered.

$$delivery\ ratio = \frac{delivered\ messages}{created\ messages} \tag{3.6}$$

Good Nodes Delivery Ratio

In the presence of misbehaving nodes, the normal delivery ratio metric may not be an effective way to measure the effectiveness of the framework. 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 3.7. One of the goals of the framework is to increase this number in the presence of misbehaving nodes.

$$good\ nodes\ delivery\ ratio = \frac{delivered\ messages\ from\ good\ nodes\ to\ good\ nodes}{created\ messages\ from\ good\ nodes\ to\ good\ nodes} \tag{3.7}$$

Misbehaving Nodes Delivery Ratio

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 3.8. One of the goals of the framework is to decrease this number in the presence of misbehaving nodes.

$$misbehaving\ nodes\ delivery\ ratio = \frac{delivered\ messages\ from\ misbehaving\ nodes}{created\ messages\ from\ misbehaving\ nodes} \tag{3.8}$$

Average Latency

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 3.9. This metric helps quantify how much the framework and the presence of misbehaving nodes affect the time a message takes to be delivered.

$$average\ latency = \frac{\sum_{i=0}^{delivered\ messages}(delivery\ time_i - creation\ time_i)}{delivered\ messages} \tag{3.9}$$

Good Nodes Average Latency

Like the delivery ratio, the basic average latency does not tell us the whole story. To get a clearer picture of the effects of the framework on latencies, the good nodes average latency is calculated using only the messages from good nodes to good nodes as shown in equation 3.10. One of the goals of the framework is to decrease this number in the presence of misbehaving nodes.

$$good\ nodes\ average\ latency = \frac{\sum_{i=0}^{delivered\ messages\ from\ good\ nodes\ to\ good\ nodes}(delivery\ time_i - creation\ time_i)}{delivered\ messages\ from\ good\ nodes\ to\ good\ nodes} \tag{3.10}$$

Misbehaving Nodes Average Latency

The misbehaving nodes average latency does the same as the previous metric using only the messages from bad as shown in equation 3.11. One of the goals of the framework is to increase this number in the presence of misbehaving nodes.

$$misbehaving\ nodes\ average\ latency = \frac{\sum_{i=0}^{delivered\ messages\ from\ misbehaving\ nodes}(delivery\ time_i - creation\ time_i)}{delivered\ messages\ from\ misbehaving\ nodes} \tag{3.11}$$

Overhead Ratio

The overhead ratio indicates the exceeding number of transmitted messages in relation to the number of delivered messages as shown in equation 3.12. 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.

$$overheard\ ratio = \frac{transmitted\ messages - delivered\ messages}{delivered\ messages} \tag{3.12}$$

Good Nodes Overhead Ratio

The good nodes overhead ratio only considers messages from good nodes to good nodes as shown in equation 3.13.

$$good\ nodes\ overheard\ ratio = \frac{transmitted\ messages\ from\ good\ nodes\ to\ good\ nodes - delivered\ messages\ from\ good\ nodes\ to\ good\ nodes}{delivered\ messages\ from\ good\ nodes\ to\ good\ nodes} \tag{3.13}$$

Misbehaving Nodes Overhead Ratio

The misbehaving nodes overhead ratio only considers messages from bad nodes as shown in equation 3.14.

$$misbehaving\ nodes\ overheard\ ratio = \frac{transmitted\ messages\ from\ misbehaving\ nodes - delivered\ messages\ from\ misbehaving\ nodes}{delivered\ messages\ from\ misbehaving\ nodes} \tag{3.14}$$

3.3.2 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 reputation is the value that translates how nodes evaluate each other. In the proposed framework this evaluation is not binary and can be between 0.5 and 1, where 1 is the best possible evaluation, meaning the node is considered good, and 0.5 the worst, meaning the node is considered misbehaving.

Nodes possess a direct reputation list, where the information is more reliable because it is dependent on direct interaction of the node, and an indirect reputation list, where the information is less reliable because it depends on the opinions the other nodes have of each other.

Good Node Average Direct Reputation

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 3.15. One of the goals of the framework is to keep this value as close to 1 as possible.

$$good\ node\ average\ direct\ reputation = \frac{\sum_{i=0}^{G}\sum_{j=0}^{G}(node\ direct\ reputation_{ij})}{G^2} \tag{3.15}$$

Good Node Average Indirect Reputation

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 the same way as the previous metric but using values from the indirect reputation list as shown in equation 3.16. Once again the closest this value is to one the better.

$$good\ node\ average\ indirect\ reputation = \frac{\sum_{i=0}^{G}\sum_{j=0}^{G}(node\ indirect\ reputation_{ij})}{G^2} \tag{3.16}$$

Misbehaving Node Average Direct Reputation

The average direct reputation indicates the average reputation from the direct reputation list of misbehaving nodes when evaluated by good nodes. It is calculated by the sum of all the reputations given by good nodes to misbehaving nodes divided by the total number of good nodes G multiplied by the total number of bad nodes B as shown in equation 3.17. This value will depend on how badly the nodes are misbehaving.

$$misbehaving\ node\ average\ direct\ reputation = \frac{\sum_{i=0}^{G}\sum_{j=0}^{B}(node\ direct\ reputation_{ij})}{G * B} \tag{3.17}$$

Misbehaving Node Average Indirect Reputation

The average indirect reputation indicates the average reputation from the indirect reputation list of misbehaving nodes when evaluated by good nodes. It is calculated the same way as the previous metric but using values from the indirect reputation list as shown in equation 3.18.

$$misbehaving\ node\ average\ indirect\ reputation = \frac{\sum_{i=0}^{G}\sum_{j=0}^{B}(node\ indirect\ reputation_{ij})}{G * B} \tag{3.18}$$

False Good Node Direct Classification Ratio

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 3.19, where G is the total number of good nodes in the network.

$$false\ good\ node\ direct\ classification\ ratio = \frac{\sum_{i=0}^{G}(number\ of\ good\ nodes\ with\ direct\ reputation\ smaller\ than\ 0.75)_i}{G^2} \tag{3.19}$$

False Good Node Indirect Classification Ratio This metric is similar to the previous one, but it uses reputations from the indirect reputation list as shown in equation 3.20 where G is the total number of good nodes. According to this ratio, a node is considered as a misbehaving node in the indirect list if his reputation is below 0.75.

$$false\ good\ node\ indirect\ classification\ ratio = \frac{\sum_{i=0}^{G}(number\ of\ good\ nodes\ with\ indirect\ reputation\ smaller\ than\ 0.75)_i}{G^2} \tag{3.20}$$

False Misbehaving Node Direct Classification Ratio

This metric is the reverse of the false good node direct classification ratio, in the sense that it is the ratio of misbehaving nodes mistakenly evaluated as good nodes by good nodes in the direct reputation list in comparison to all the evaluations made of misbehaving nodes by good nodes. It is calculated using equation 3.21 where G is the total number of good nodes in the network and B is the total number of misbehaving nodes in the network.

$$false\ misbehaving\ node\ direct\ classification\ ratio = \frac{\sum_{i=0}^{G}(number\ of\ misbehaving\ nodes\ with\ direct\ reputation\ bigger\ than\ 0.75)_i}{G*B} \tag{3.21}$$

False Misbehaving Node Indirect Classification Ratio

This metric is similar to the previous one, but it uses reputations from the indirect reputation list as shown in equation 3.22 where G is the total number of good nodes and B is the total number of misbehaving nodes. This metric is the reverse of the false good node indirect classification ratio, in the sense that it is the ratio of misbehaving nodes mistakenly evaluated as good nodes by good nodes in the indirect reputation list in comparison to all the evaluations made of misbehaving nodes by good nodes.

$$false\ misbehaving\ node\ indirect\ classification\ ratio = \frac{\sum_{i=0}^{G}(number\ of\ misbehaving\ nodes\ with\ indirect\ reputation\ bigger\ than\ 0.75)_i}{G*B} \tag{3.22}$$

4

Robustness Results Analysis

Contents

4.1 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. The settings decided for the scenarios are summarized in Table 4.1.

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, PRoPHET and Maxprop. The results of this simulations will be discussed in the next sections.

Simulation time	7 days (168 hours)		
Movement Model	Shortest Path Map-Based Movement Model		
Node Buffer Size	5MB		
Node Speed	Pedestrians 1.8-5.4 km/h Cars 10-50 km/h Trams 25-36 km/h		
Node wait time	0-120 seconds		
Message creation interval	5-10 minutes		
Message Size	500kb-1Mb		
Message TTL (Time to live)	5 hours		
Interface data rate	250kBps=2Mbps		
Interface Transmission Range	10 meters		
Number of nodes	40	80	120
Network profile	17 pedestrians	37 pedestrians	57 pedestrians
	17 cars	37 cars	57 cars
	6 trams		

Table 4.1: Settings assigned to simulations scenarios

4.2 Black-hole node simulation

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.

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.

The reputation metrics translate how fast and how accurately the framework detects misbehaving nodes. To start, let us observe how fast the system differentiates between good and misbehaving nodes and how accurate this distinction is. The results are presented in fig. 4.1, fig. 4.2, fig. 4.3 and fig. 4.4, respectively for the SnW, Epidemic, PRoPHET and Maxprop routing protocols. On the left side of the images are the average reputation graphs, which represent the good nodes average direct reputation (green), the good nodes average indirect reputation(cyan), the misbehaving nodes average direct reputation (red) and the misbehaving nodes average indirect reputation (orange) over time. In reputation graphs, the objective of the framework is to make the misbehaving and good average reputation curves diverge, with good nodes average reputation curves tending to 1 and misbehaving nodes reputation curves tending to 0.5. On the right side of the images are the false ratio graphs, which represent the false bad node direct classification ratio (green), the false bad node indirect classification ratio(cyan), the false good node direct classification ratio (red) and the false good node direct classification ratio (orange) over time. In false ratio graphs, the objective of the framework make the curves of all false ratios be as close as possible to 0.

This system is not binary so nodes are neither good nor bad. They have a reputation, and this reputation can be either high or low, a high reputation meaning the node is considered good and a low reputation meaning the node is considered bad. To make it simpler to understand, we consider that a node with reputation smaller or equal to 0.75 is considered bad and a node with reputation bigger than 0.75 is considered good. We choose 0.75 as the threshold for this distinction because nodes are only punished by the system when their reputation is below this value, because the maximum strategy threshold is 0.75. The lower their reputation the more the node will be punished by the network. For example while a node with an average reputation of 0.74 will mostly only be punished by nodes with strategy thresholds of 0.75, a node with an average reputation of of 0.64 will be mostly be punished by nodes with strategy thresholds of 0.75, 0.70 and 0.65.

As we can see in the reputation graphs, the reputation framework clearly makes a good distinction between good and misbehaving nodes. With all the graphs showing a better average reputation for good nodes than for misbehaving nodes. Being the misbehaving nodes in this section attacking the network through black-hole attacks, they will not have an opportunity to improve their reputation, because they drop all the messages they receive. With all those messages dropped they will never have chance to gain reputation through forwarding messages. So we conclude that 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.

Looking at the graphs of the different protocols, we can clearly see a distinction between the SnW graphs and the graphs of the all other protocols. While all other reputation graphs from the other protocols show a tendency to diverge quickly with good nodes average reputations tending to high values between 0.9 and 1.0 and misbehaving nodes average reputation tending to 0.5, the SnW reputation graphs shows a very slow divergence or almost none at all. This is due to the amount of overhead in the network. All of the scenarios use the same message creation rate so this changes are clearly due to the difference in protocols. While all the protocols tested use message flooding, the SnW is the only one with a limited number of copies, which drastically reduces the overhead in relation to the other protocols. The fact that the attacks in question cause the dropping of messages which reduces the overhead even more does not help either. While the framework may not be geared towards low message rate situations and low overhead situations it still presented some results in terms of distinction between good and misbehaving nodes. Looking at fig. 4.1, we can clearly see the good node average reputation trend upwards while the misbehaving nodes average reputations trend downwards in the 40 nodes scenario, with this divergence being more accentuated the less misbehaving nodes the network has. For situations with 20% of misbehaving nodes in the network the number of dropped messages is less than the number of dropped messages with 80% of misbehaving nodes in the network, leading to more messages circulating and to a faster divergence. In the case of the scenarios with 120 nodes, the graphs barely diverge due to the limited overhead in the network caused by the protocol. Although the results with this protocol may not be the best, they show the indirect list functions perfectly. While in all false ratio graphs the false good node direct reputation ratio trends downward over time in scenarios with 40 nodes and stays between 0.3 and 0.5 throughout the simulation, the false good node indirect reputation quickly tends downwards staying between 0 and 0.04 in the 40 nodes scenario and between 0 and 0.2 in the 120 nodes scenario throughout the simulation.

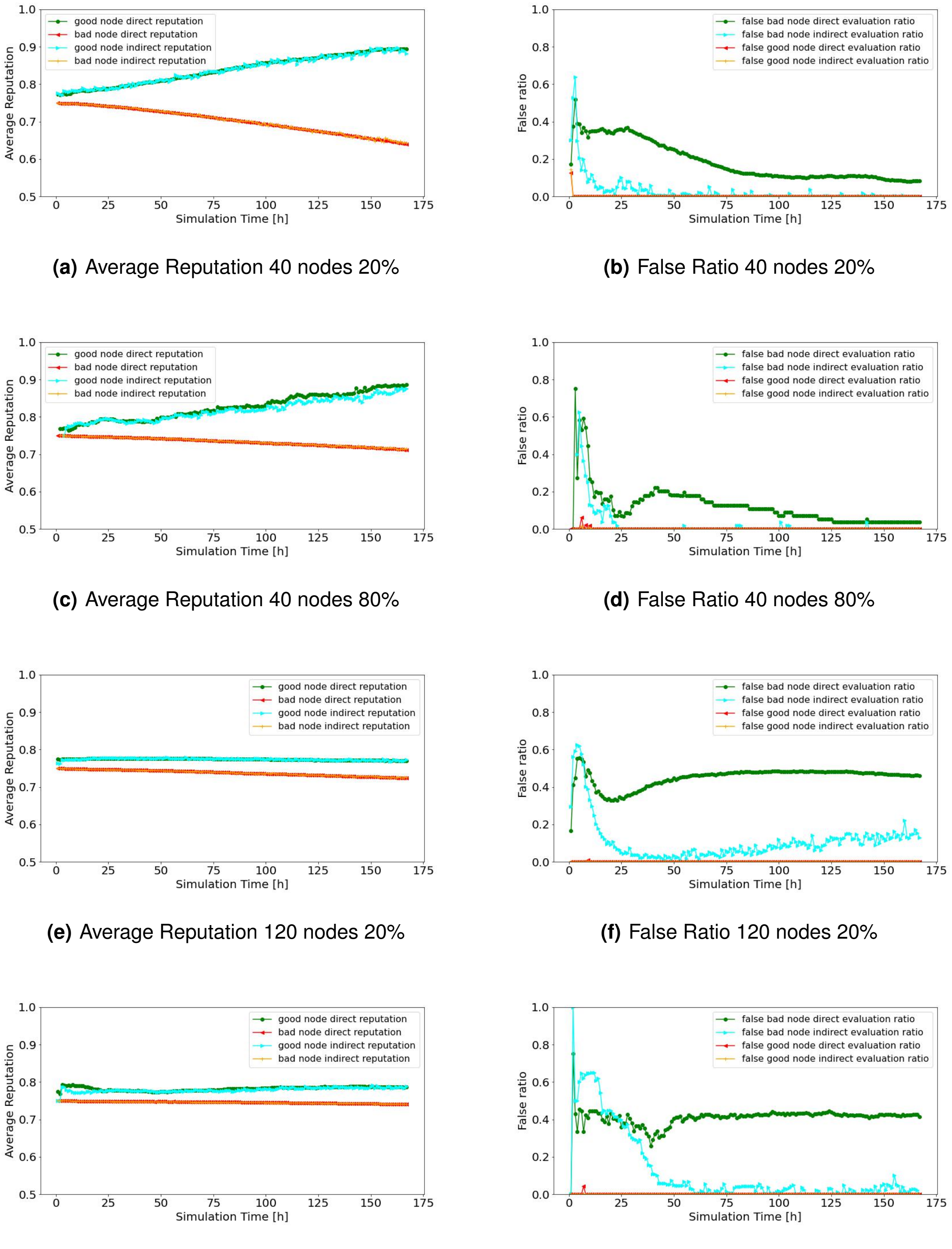
Looking at fig. 4.2, we can clearly see the divergence in reputation and some of the faster divergence times of all the scenarios. This is due to the very big overhead characteristic of the Epidemic routing protocol, caused by the unlimited flooding not based on any information. If we look at the false ratio graphs, the values of false good reputation ratios trend to values between 0 and 0.01 in less than 24 hours, except for the case of the 40 nodes scenario with 80% misbehaving nodes, where interactions are scarce and the number of dropped messages is very high. The best case scenario being 8 hours in the dense network of 120 nodes with 20% misbehaving nodes, where the overhead is the highest and the number of hops per message is also high due to the density of the network. In most cases this ratio never goes beyond 0.1 which is a very good sign. When looking at the average reputation graphs we can clearly observe that the complete divergence for the average reputation of misbehaving nodes to get to the minimum value of 0.5, takes a while longer with a time of about 50 hours to fully diverge in the cases with 20% misbehaving nodes. In the cases with 80% misbehaving nodes this divergence takes much longer due to the reduction in overhead caused by the dropping of messages. We can also clearly see that the 120 nodes scenario's average reputation of misbehaving nodes reaches the minimum value of 0.5 in about 100 hours, while the 40 nodes scenario's average reputation of misbehaving nodes trends in that way but clearly would take much longer.

Looking at fig. 4.3, we can see a very similar situation to the one observed in fig. 4.2 with the divergence time for the PRoPHET protocol about 50% slower than for the Epidemic routing protocol and with some higher false good reputation ratios. This is due to the nature of the PRoPHET routing protocol. Although it uses unlimited flooding causing the overhead to be fairly high, it uses information based flooding. This makes the overhead not as high as for the Epidemic routing protocol causing a slower divergence. Not only this, but the way the PRoPHET routing protocol information based flooding works, previously explained in section section 2.3.1 where a new message copy is only transmitted if the other node has a greater delivery probability to the target. This probability is using the encounter history and transitivity, which may lead to some good nodes being isolated and having an impact in their reputation. The high overhead of the protocol solves this problem for dense networks but as we can observe this in fig. 4.3, in the scenario of 40 nodes and 80% of those misbehaving, where the false good node reputation ratios first trend downwards, stabilizing between 0 and 0.2 throughout the simulation. Although this values are not very high considering most of the nodes in this scenario are misbehaving, it is something to take note of some lack of performance in sparse networks with this protocol.

Looking at fig. 4.4, we can see a very similar situation to the one observed in fig. 4.3. And although the reputation average graphs may be identical we can clearly notice the difference in the false ratio graphs. This is most noticeable in the false ratio graph of the scenario of 120 nodes with 20% of those misbehaving. This is, once again, due to the way the information based flooding works in this protocol. As previously explained in section section 2.3.1 this protocol is the only protocol that uses an acknowledgement once the message meets the destination, sent via broadcast to inform the encountered nodes to clear out the existing copies of the delivered message. This drastically reduces the overhead, especially in denser networks. This leads to what we see in the mentioned graph, where good false reputation ratios constantly are between 0.02 and 0.2. With the indirect ratio always clearly below the direct ratio, once again demonstrating the indirect lists function.

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 table 4.2.

Although the delivery ratios are not that affected by the framework, we can still see some clear improvements. In table 4.2, fig. 4.5 and fig. 4.6 there is a clear improvement in the Epidemic and PRoPHET routing protocol's good nodes delivery ratio, while for the other protocols it remains the same. There is a slight overall improvement, more noticeable in the denser scenarios of 120 nodes. This is especially true for the Epidemic routing protocol with very high percentages of misbehaving nodes, where the good nodes delivery ratio nearly doubles when 80% of the nodes are misbehaving. In the



(a) Average Reputation 40 nodes 20%

(b) False Ratio 40 nodes 20%

(c) Average Reputation 40 nodes 80%

(d) False Ratio 40 nodes 80%

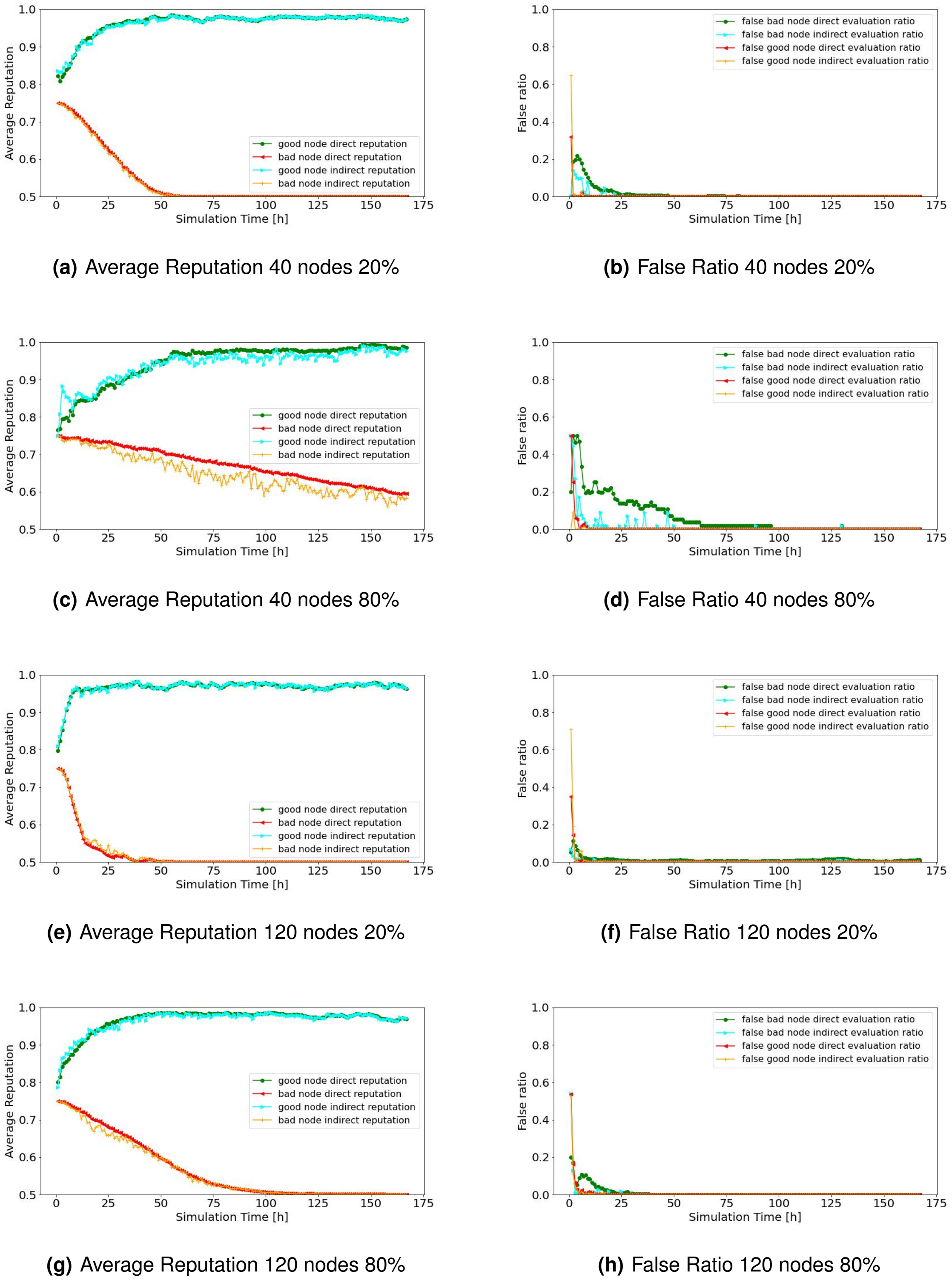
(e) Average Reputation 120 nodes 20%

(f) False Ratio 120 nodes 20%

(g) Average Reputation 120 nodes 80%

(h) False Ratio 120 nodes 80%

Figure 4.1: Comparison of Average Reputations and False Ratios for Spray and Wait, with 20% and 80% of misbehaving nodes to test robustness of scenarios with different number of nodes



(a) Average Reputation 40 nodes 20%

(b) False Ratio 40 nodes 20%

(c) Average Reputation 40 nodes 80%

(d) False Ratio 40 nodes 80%

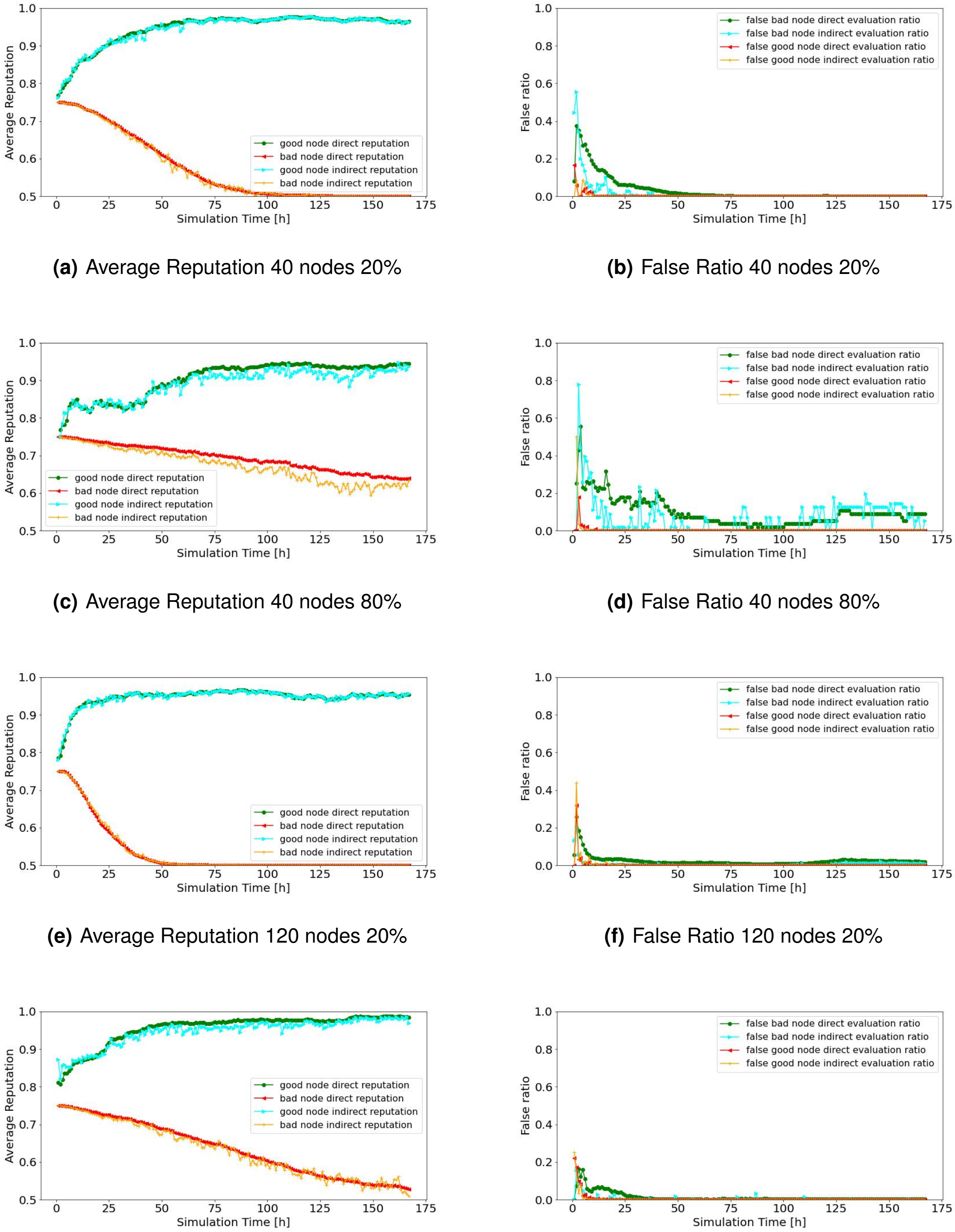
(e) Average Reputation 120 nodes 20%

(f) False Ratio 120 nodes 20%

(g) Average Reputation 120 nodes 80%

(h) False Ratio 120 nodes 80%

Figure 4.2: Comparison of Average Reputations and False Ratios for Epidemic, with 20% and 80% of misbehaving nodes to test robustness of scenarios with different number of nodes



(a) Average Reputation 40 nodes 20%

(b) False Ratio 40 nodes 20%

(c) Average Reputation 40 nodes 80%

(d) False Ratio 40 nodes 80%

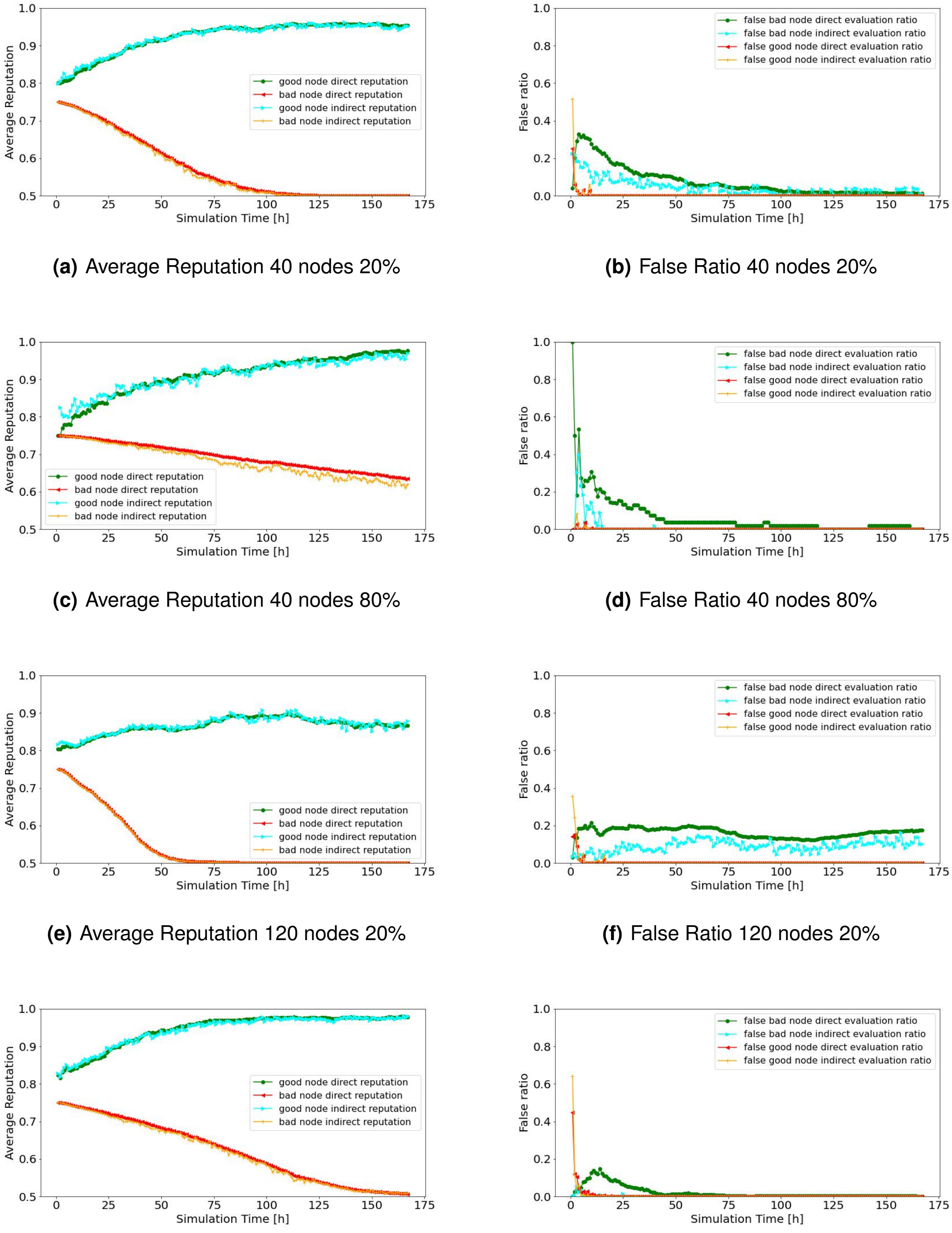
(e) Average Reputation 120 nodes 20%

(f) False Ratio 120 nodes 20%

(g) Average Reputation 120 nodes 80%

(h) False Ratio 120 nodes 80%

Figure 4.3: Comparison of Average Reputations and False Ratios for Prophet, with 20% and 80% of misbehaving nodes to test robustness of scenarios with different number of nodes



(a) Average Reputation 40 nodes 20%

(b) False Ratio 40 nodes 20%

(c) Average Reputation 40 nodes 80%

(d) False Ratio 40 nodes 80%

(e) Average Reputation 120 nodes 20%

(f) False Ratio 120 nodes 20%

(g) Average Reputation 120 nodes 80%

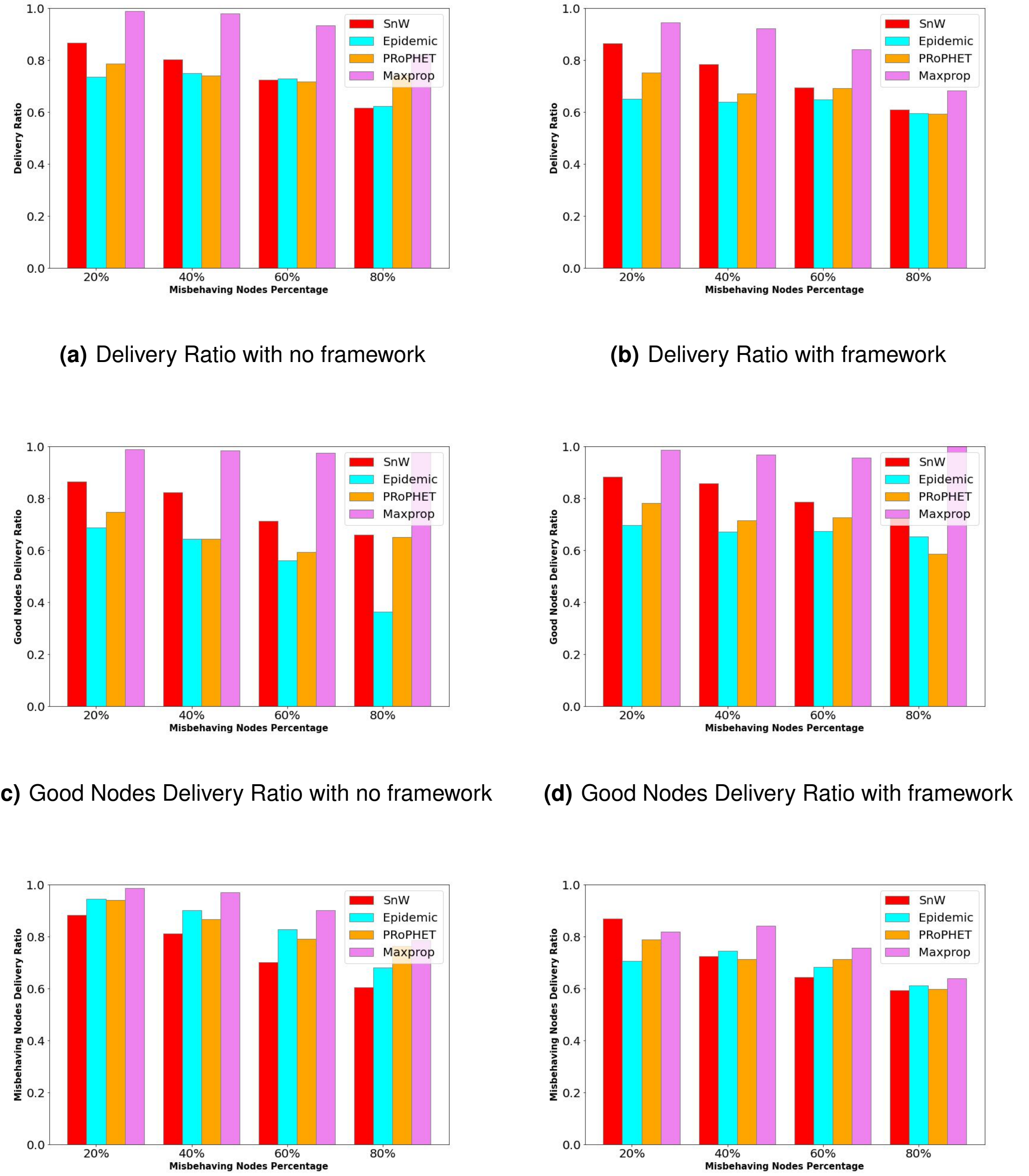
(h) False Ratio 120 nodes 80%

Figure 4.4: Comparison of Average Reputations and False Ratios for MaxProp, with 20% and 80% of misbehaving nodes to test robustness of scenarios with different number of nodes

case of the misbehaving nodes delivery ratio the improvement is much more noticeable, with all protocols except for the SnW performing better with overall smaller misbehaving nodes delivery ratios.

In the case of the overhead, with the exception of SnW and as we can see in table 4.2, there is a very big reduction of overhead overall when using the framework. The reduction in overhead was to be expected, because nodes become more selective in terms of which nodes they send messages copies to. For good nodes, the framework tries to minimize the messages sent to misbehaving nodes (who would drop them as soon as they receive them), which translates to smaller good nodes overhead ratio. This reduction is good because good nodes save energy by not sending messages that would in reality waste energy because they would be dropped. For misbehaving nodes, the framework tries to minimize the messages sent by them, which translates to smaller bad nodes overhead ratio. This reduction is good because it means the framework is punishing misbehaving nodes for their bad deeds. In figures fig. 4.7 and fig. 4.8 this reduction is more noticeable in the denser scenarios. In sparser scenarios with 40 nodes the reduction of good nodes overhead ratio is less accentuated than in the denser scenarios with 120 nodes. The good nodes overhead ratio also show a smaller decrease, percentage wise, the higher the misbehaving nodes percentage present in the network. In the case of the misbehaving nodes overhead ratio, the results show that, like the good nodes overhead ratio, in sparser scenarios with 40 nodes the reduction of bad nodes overhead ratio is less accentuated than in the denser scenarios with 120 nodes. The misbehaving nodes overhead ratio also show a smaller decrease, percentage wise, the higher the misbehaving nodes percentage present in the network. Percentage wise, the reduction of misbehaving nodes overhead ratio is much bigger than the reduction of good nodes overhead ratio. These are very good results because it shows that, although overhead was reduced overall (meaning nodes are not cooperating blindly with other nodes), the cooperation of the network with misbehaving nodes was hugely reduced by the framework.

Finally looking at the latency in table 4.2, fig. 4.9 and fig. 4.10 we can observe substantial changes in every protocol, except once again for SnW. When using the framework we can observe two different trends: the bigger the percentage of misbehaving nodes, the bigger the good nodes latency average increase in relation to the scenario not using the framework; the bigger the percentage of misbehaving nodes, the smaller the misbehaving nodes latency average increase in relation to the scenario not using the framework. What this means is that because good nodes end up only cooperating with each other due to the framework while not cooperating with misbehaving nodes, the more good nodes in the network, the smaller the good nodes average latency will be and the bigger the misbehaving nodes average latency will be. This means the framework is fulfilling its task of improving cooperation between good nodes while decreasing cooperation with misbehaving nodes.



(a) Delivery Ratio with no framework

(b) Delivery Ratio with framework

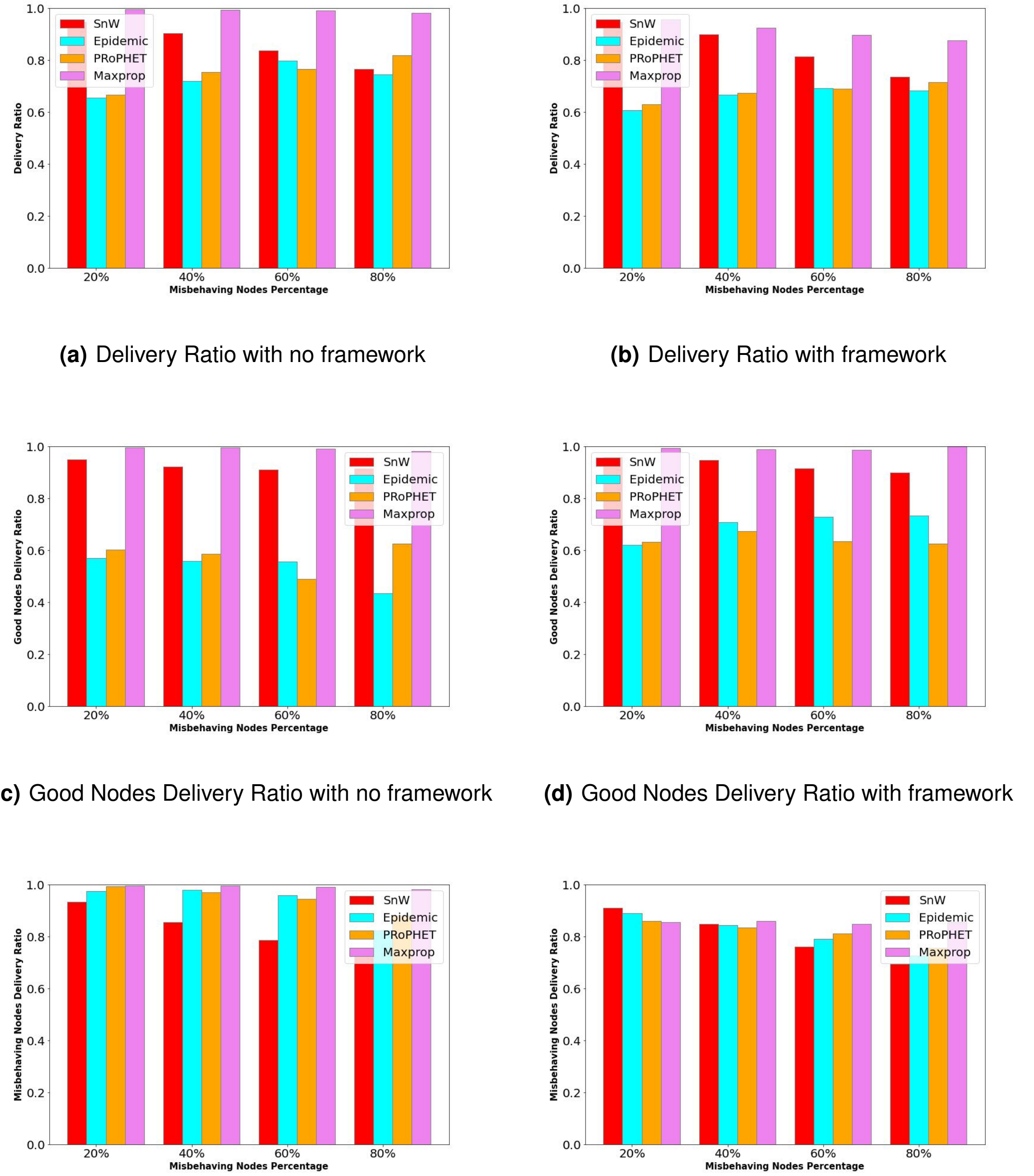
(c) Good Nodes Delivery Ratio with no framework

(d) Good Nodes Delivery Ratio with framework

(e) Misbehaving Nodes Delivery Ratio with no framework

(f) Misbehaving Nodes Delivery Ratio with framework

Figure 4.5: Comparison of Delivery Ratio, Good Nodes Delivery Ratio and Misbehaving Nodes Delivery Ratio for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes



(a) Delivery Ratio with no framework

(b) Delivery Ratio with framework

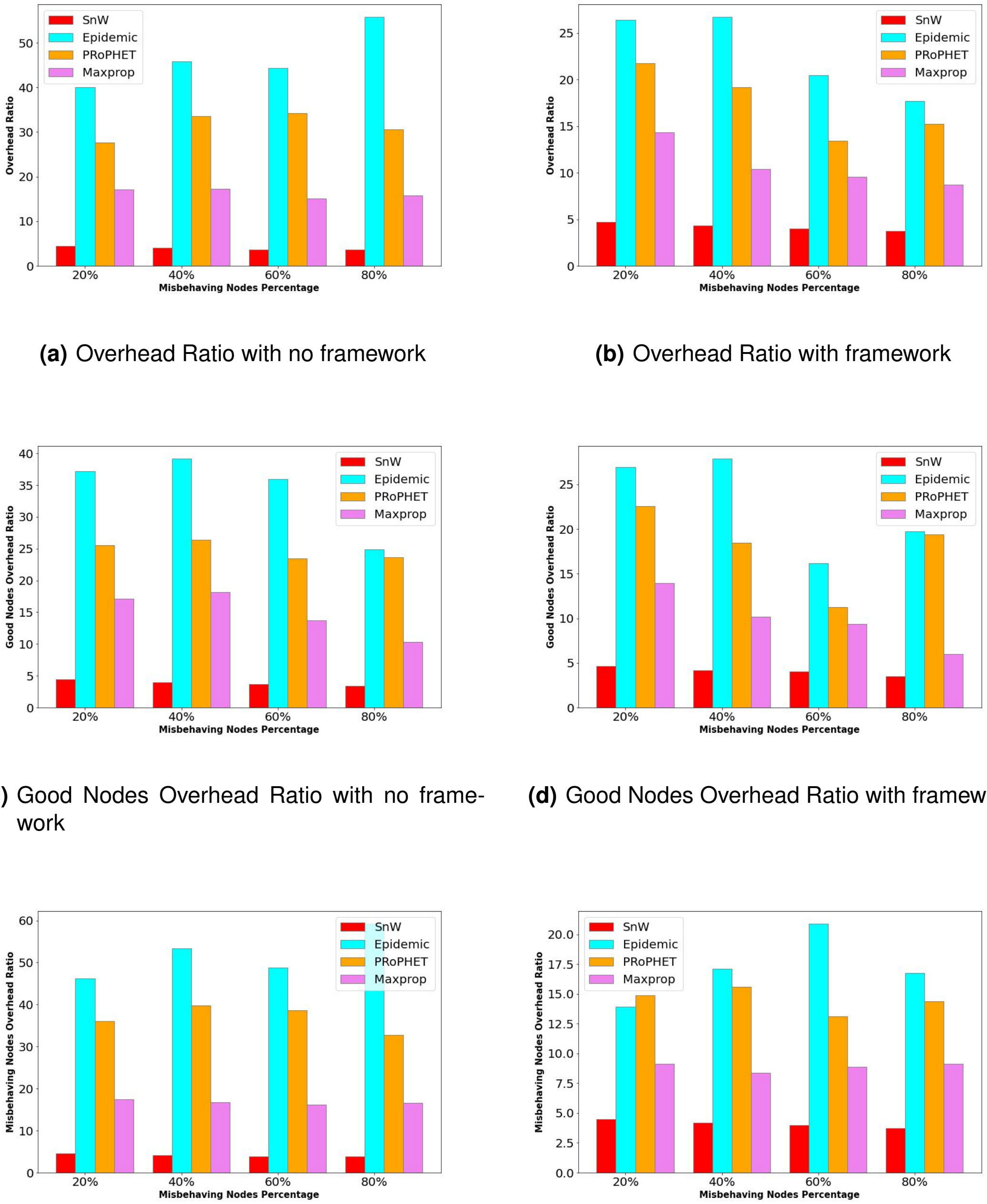
(c) Good Nodes Delivery Ratio with no framework

(d) Good Nodes Delivery Ratio with framework

(e) Misbehaving Nodes Delivery Ratio with no framework

(f) Misbehaving Nodes Delivery Ratio with framework

Figure 4.6: Comparison of Delivery Ratio, Good Nodes Delivery Ratio and Misbehaving Nodes Delivery Ratio for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes



(a) Overhead Ratio with no framework

(b) Overhead Ratio with framework

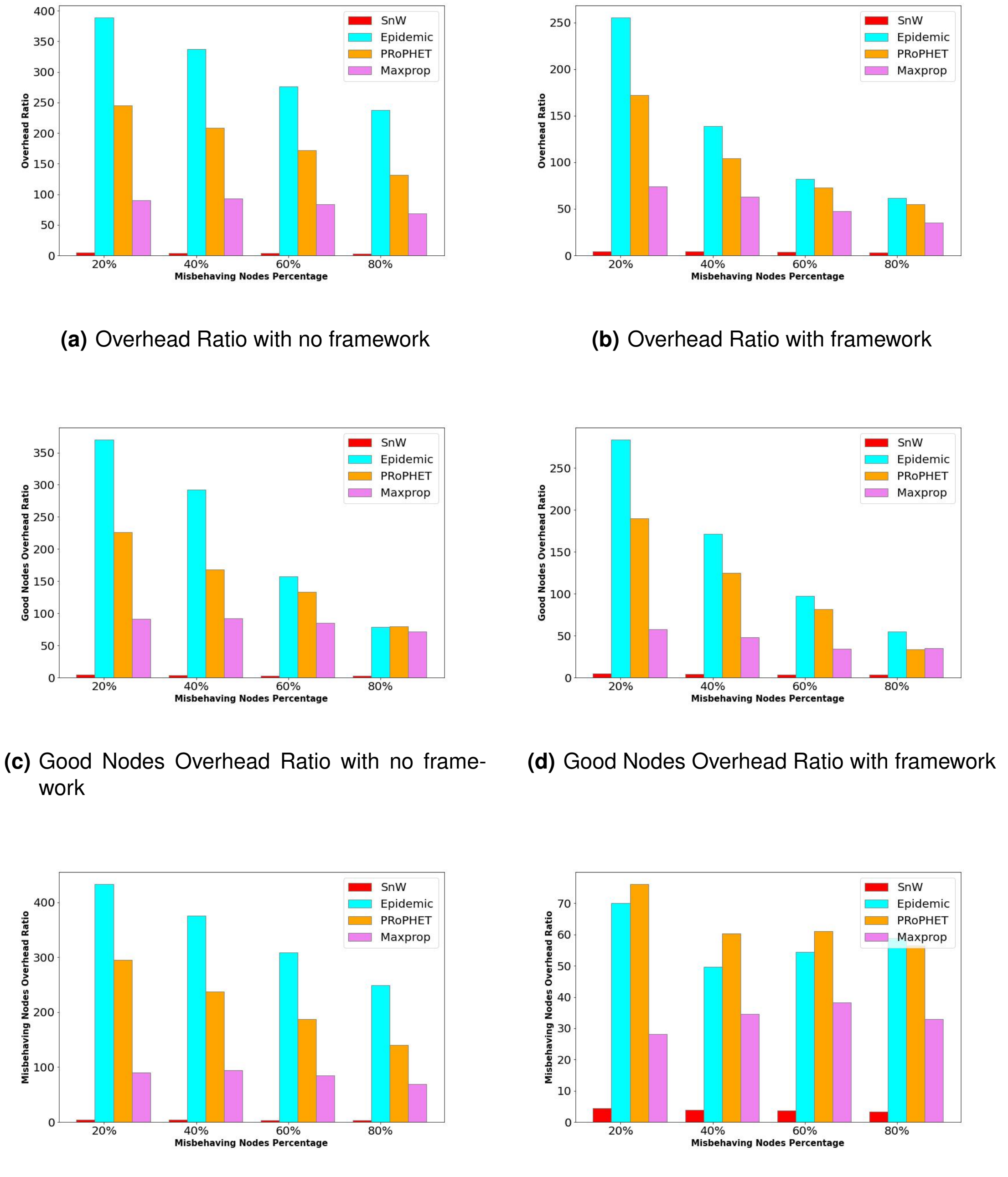
(c) Good Nodes Overhead Ratio with no framework

(d) Good Nodes Overhead Ratio with framework

(e) Misbehaving Nodes Overhead Ratio with no framework

(f) Misbehaving Nodes Overhead Ratio with framework

Figure 4.7: Comparison of Overhead Ratio, Good Nodes Overhead Ratio and Misbehaving Nodes Overhead Ratio for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes



(a) Overhead Ratio with no framework

(b) Overhead Ratio with framework

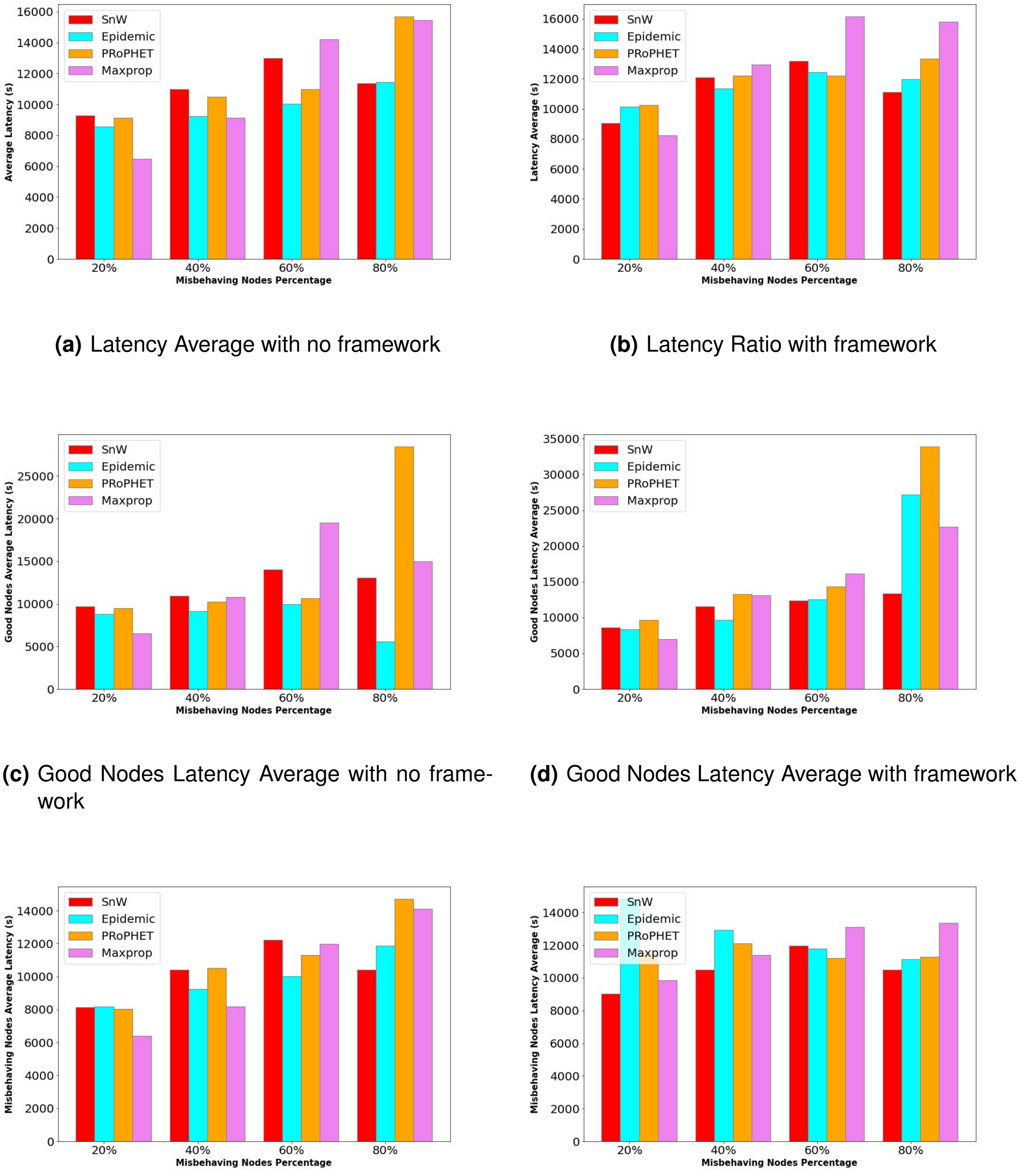
(c) Good Nodes Overhead Ratio with no framework

(d) Good Nodes Overhead Ratio with framework

(e) Misbehaving Nodes Overhead Ratio with no framework

(f) Misbehaving Nodes Overhead Ratio with framework

Figure 4.8: Comparison of Overhead Ratio, Good Nodes Overhead Ratio and Misbehaving Nodes Overhead Ratio for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes



(a) Latency Average with no framework

(b) Latency Ratio with framework

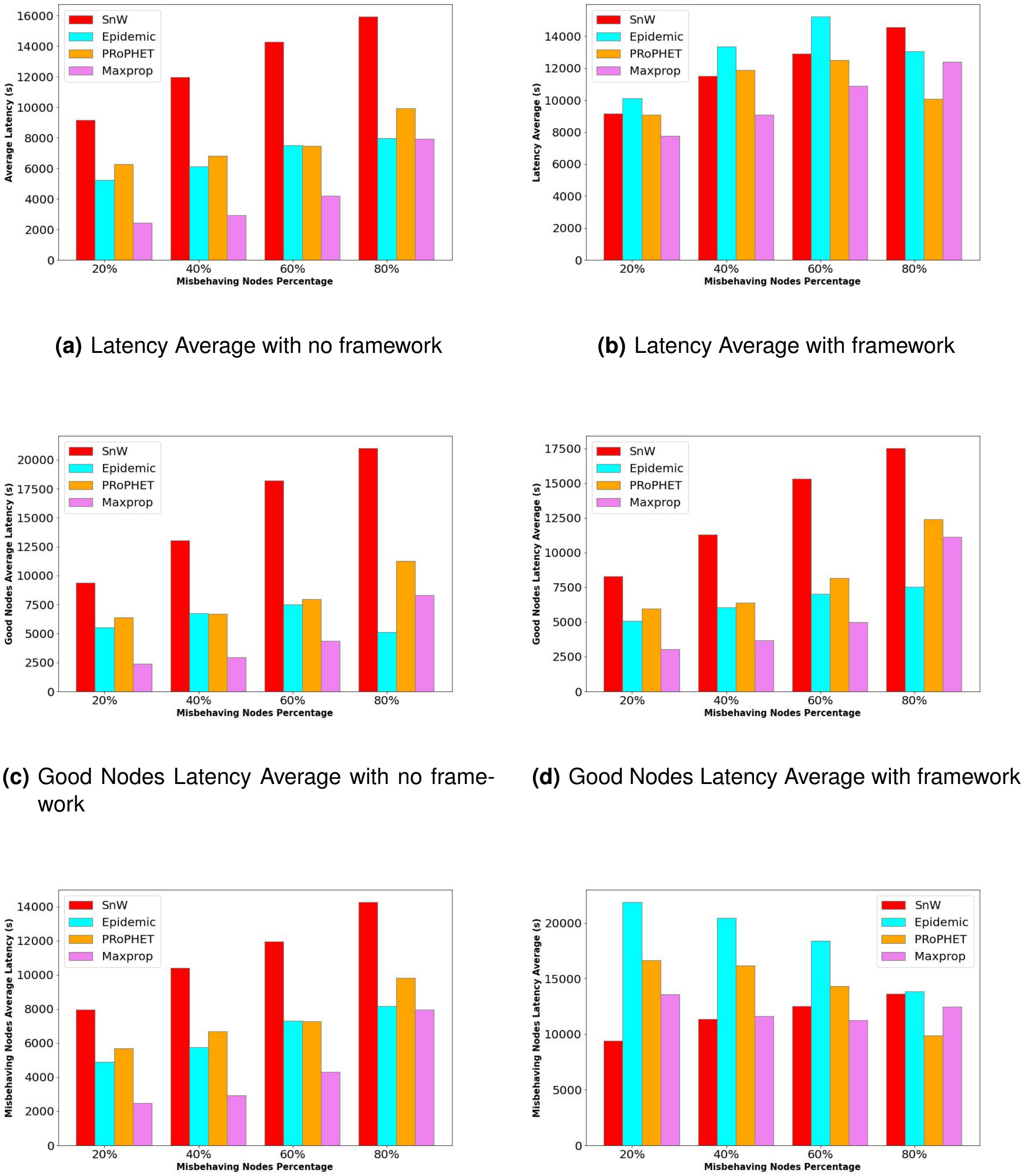
(c) Good Nodes Latency Average with no framework

(d) Good Nodes Latency Average with framework

(e) Misbehaving Nodes Latency Average with no framework

(f) Misbehaving Nodes Latency Average with framework

Figure 4.9: Comparison of Latency Average, Good Nodes Latency Average and Misbehaving Nodes Latency Average for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes



(a) Latency Average with no framework

(b) Latency Average with framework

(c) Good Nodes Latency Average with no framework

(d) Good Nodes Latency Average with framework

(e) Misbehaving Nodes Latency Average with no framework

(f) Misbehaving Nodes Latency Average with framework

Figure 4.10: Comparison of Latency Average, Good Nodes Latency Average and Misbehaving Nodes Latency Average for all protocols, with 20%, 40%, 60% and 80% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes

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 4.2: 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

4.3 Gray-hole node simulation

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.

The objective with this simulations is to see how well the system performs when it encounters nodes that are not fully misbehaving, and see until when the system remains relevant through the testing of various misbehaviour degrees. The objective of the framework in this simulations is to diverge in the values of good node and misbehaving node average reputation, while maintaining the max accuracy possible in terms of classifying a good node as a good node and a misbehaving node as a misbehaving node.

The graphs presented in images fig. 4.11, fig. 4.12, fig. 4.13 and fig. 4.14, respectively for the SnW, Epidemic, PRoPHET and MaxProp routing protocols, are presented in the same way as they were explained in the previous section. This time around the false good node ratio will be noticeable, because misbehaving nodes will also sometimes cooperate earning them reputation. In this simulation the average reputation graphs will prove much more important than the false ratio reputation because the objective of the framework is mainly to differentiate the good and misbehaving nodes by levels. The more misbehaved a node the bigger the divide between their average reputation and the good nodes average reputation.

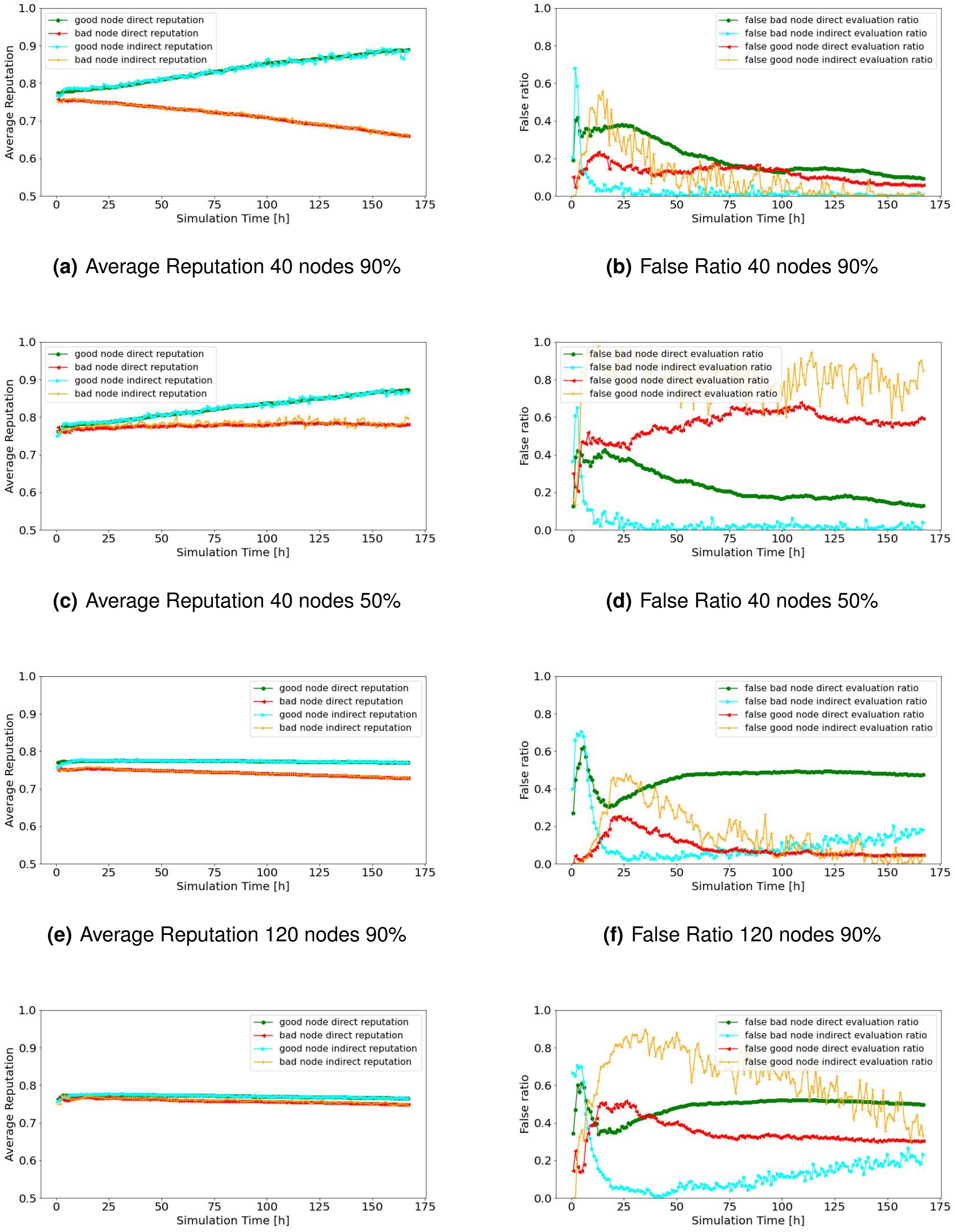
As expected, the results for SnW in fig. 4.11 are very similar to the ones in the black-hole attack scenarios in terms of the average reputation graphs. Due to the characteristics of the protocol previously explained the divergence is very slow, and the denser the network the slower it is. With a dropping probability of 90% the graphs are very similar to the the ones in the scenarios with the black-hole attacks, the only difference being that the false good reputation ratios trend downwards and stabilize to values between 0.05 and 0.06. With lower dropping probabilities, the results get progressively worse as the misbehaving nodes become harder to differentiate from the good nodes because they become progressively better behaved. When we get to 50% dropping probability the framework barely distinguishes good nodes from bad nodes. This result getting even worse when we look at the denser network of 120 nodes.

Looking at fig. 4.12 we can observe a very similar behaviour to the black-hole attack scenario in the 90% dropping probability scenario. The difference between them being, that at the beginning of the simulation the average reputation for misbehaving nodes gets very high only to then drop abruptly and get very close to the minimum value of reputation, and even reaching it in the denser network of 120 nodes. The very high overhead of the Epidemic routing protocol becomes a disadvantage for the framework when we are dealing with smaller dropping probabilities because it becomes harder for the nodes to distinguish good nodes from misbehaving nodes. This is due to the way the framework works. If a node forwards enough messages he will be able to maintain its reputation within the network making harder for the framework to differentiate between good and misbehaving. When we look at the 50% dropping probability scenario we can observe that although the framework differentiates good and misbehaving nodes in terms of reputation, misbehaving ends up mostly not being punished because their average reputation is above 0.75.

Looking at fig. 4.13 we can see that the framework behaves in a similar way to the one in fig. 4.12, the graphs even being identical in every scenario. As with the other protocols the framework works progressively worse as the dropping probability gets smaller.

Looking at fig. 4.14 we can observe that although it is very similar in behaviour to the last two, it performs much better than the other when in a denser network of 120 nodes. Looking at the scenario of 120 nodes with drop probability of 50% we can see that the framework clearly distinguishes the good and the misbehaving nodes, and it even classifies almost every misbehaving node as a misbehaving node, with the false good node ratios tending to values between 0.01 and 0.05. This is probably due to the discarding of messages and consequent reduction of overhead by the MaxProp routing protocol. Although this results do not prove the system works for low dropping probabilities it shows the potential the system has if we were to use for example different types of punishment, like for example dropping of messages sent by misbehaved nodes.

Once again, 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 table 4.3. The improvements in performance of the routing protocol are the same to the ones in the black-hole attacks scenarios. The only difference being that this improvements get progressively worse, the smaller the dropping probability of the misbehaving nodes. As we can see in table 4.3 and in images fig. 4.15, fig. 4.16, fig. 4.17, fig. 4.18, fig. 4.19 and fig. 4.20: for the SnW routing protocol the



(a) Average Reputation 40 nodes 90%

(b) False Ratio 40 nodes 90%

(c) Average Reputation 40 nodes 50%

(d) False Ratio 40 nodes 50%

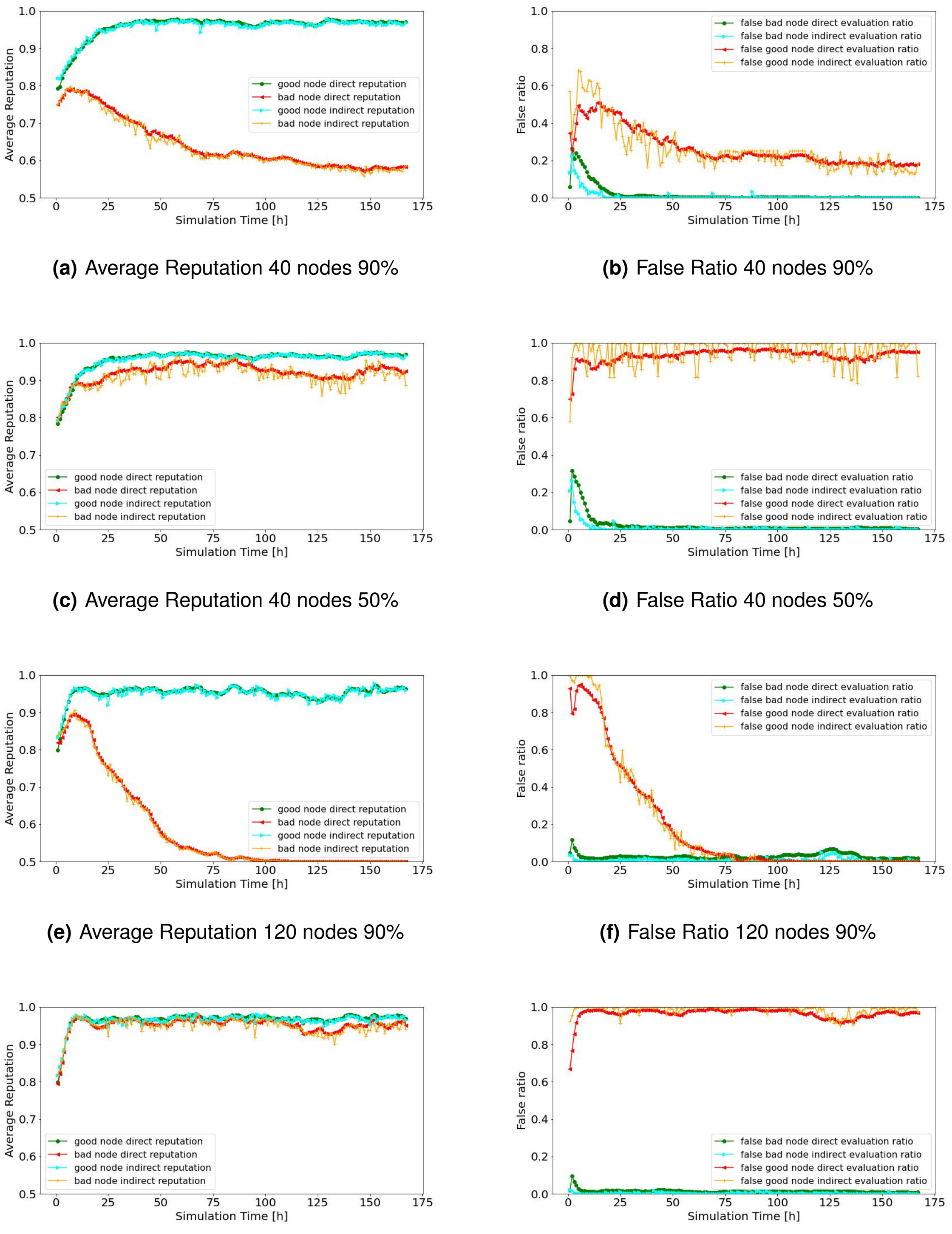
(e) Average Reputation 120 nodes 90%

(f) False Ratio 120 nodes 90%

(g) Average Reputation 120 nodes 50%

(h) False Ratio 120 nodes 50%

Figure 4.11: Comparison of Average Reputations and False Ratios for Spray and Wait, with 20% of misbehaving nodes with drop probabilities of 50% and 90% to test robustness of scenarios with different degrees of misbehaviour and different number of nodes



(a) Average Reputation 40 nodes 90%

(b) False Ratio 40 nodes 90%

(c) Average Reputation 40 nodes 50%

(d) False Ratio 40 nodes 50%

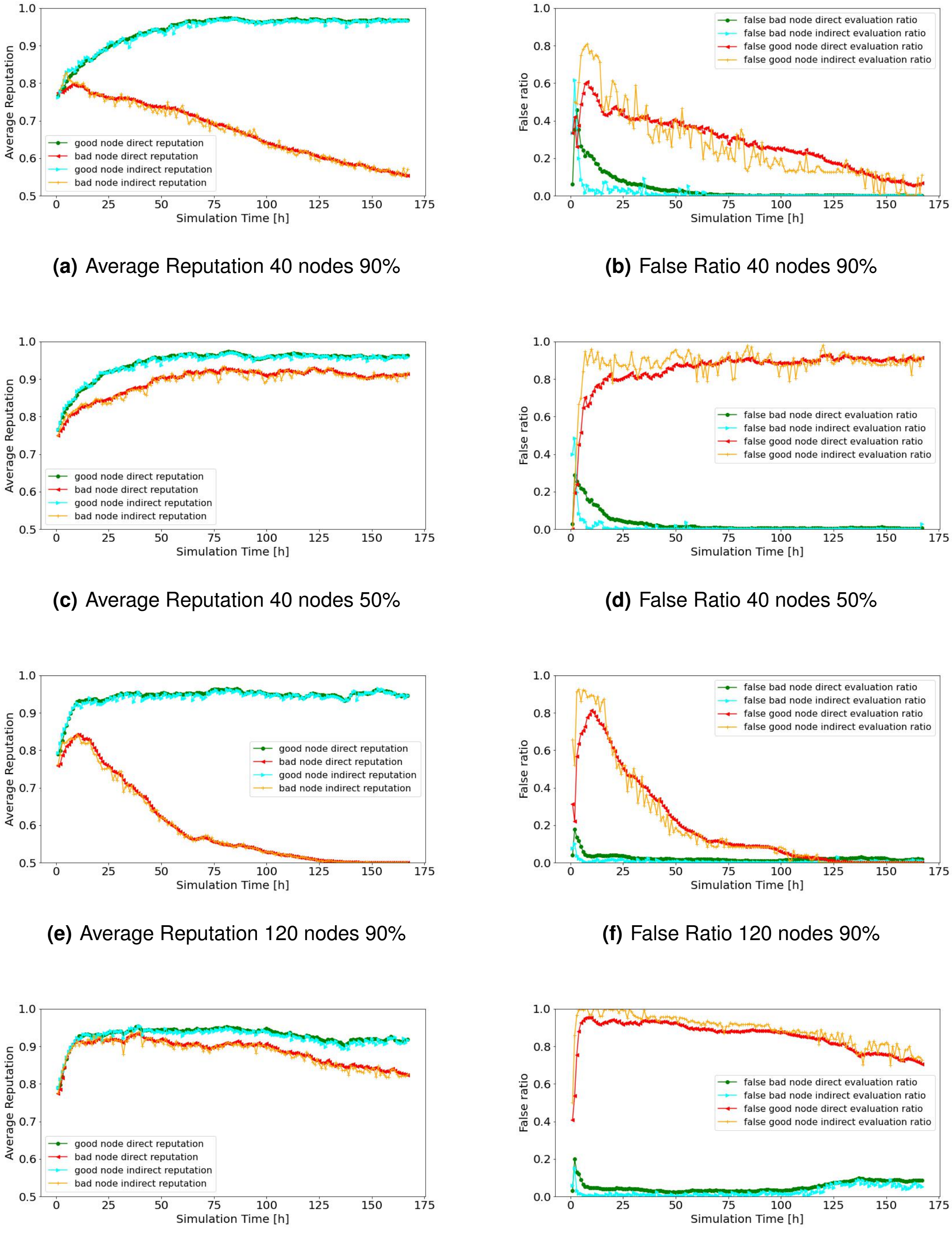
(e) Average Reputation 120 nodes 90%

(f) False Ratio 120 nodes 90%

(g) Average Reputation 120 nodes 50%

(h) False Ratio 120 nodes 50%

Figure 4.12: Comparison of Average Reputations and False Ratios for Epidemic, with 20% of misbehaving nodes with drop probabilities of 50% and 90% to test robustness of scenarios with different degrees of misbehaviour and different number of nodes



(a) Average Reputation 40 nodes 90%

(b) False Ratio 40 nodes 90%

(c) Average Reputation 40 nodes 50%

(d) False Ratio 40 nodes 50%

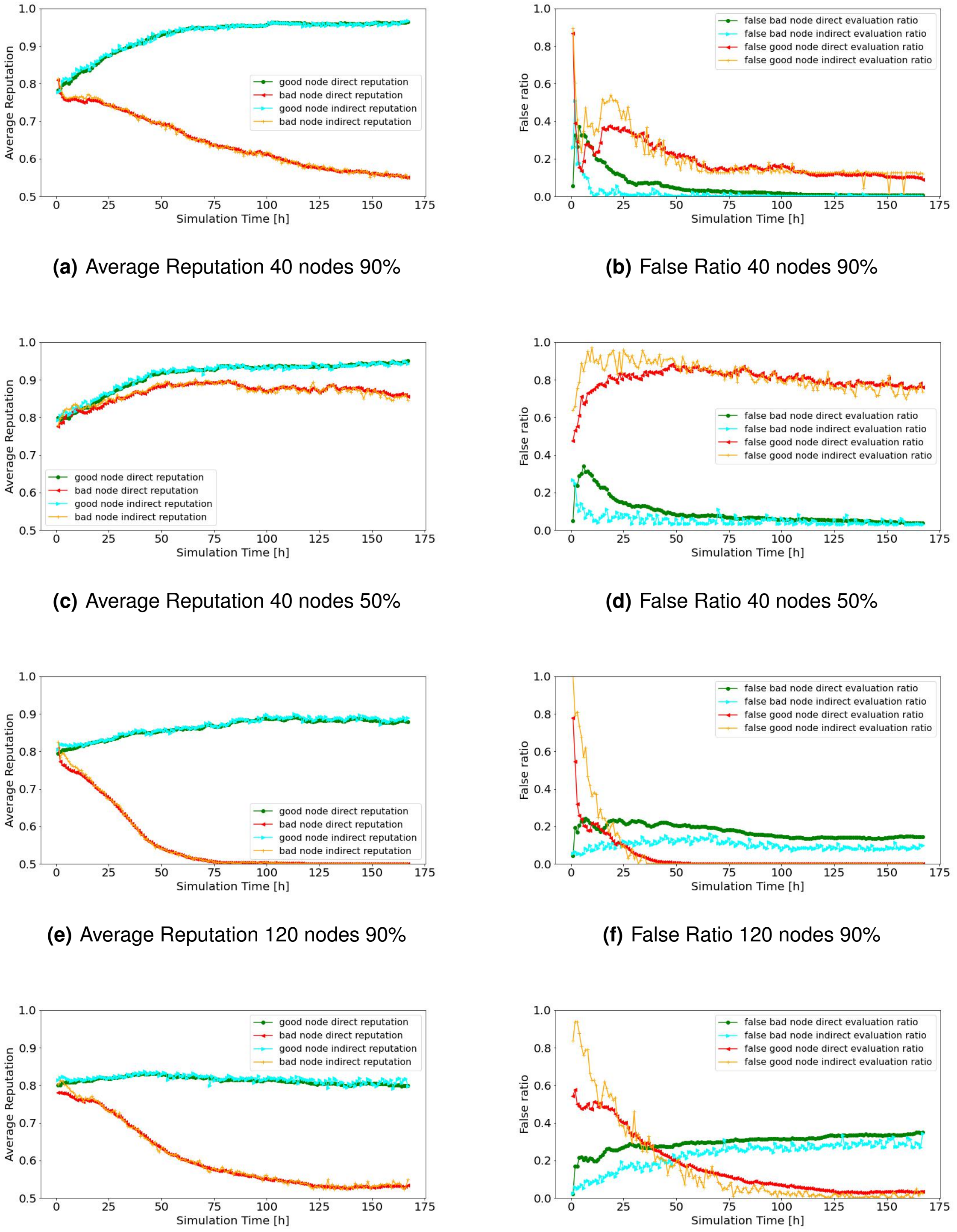
(e) Average Reputation 120 nodes 90%

(f) False Ratio 120 nodes 90%

(g) Average Reputation 120 nodes 50%

(h) False Ratio 120 nodes 50%

Figure 4.13: Comparison of Average Reputations and False Ratios for Prophet, with 20% of misbehaving nodes with drop probabilities of 50% and 90% to test robustness of scenarios with different degrees of misbehaviour and different number of nodes



(a) Average Reputation 40 nodes 90%

(b) False Ratio 40 nodes 90%

(c) Average Reputation 40 nodes 50%

(d) False Ratio 40 nodes 50%

(e) Average Reputation 120 nodes 90%

(f) False Ratio 120 nodes 90%

(g) Average Reputation 120 nodes 50%

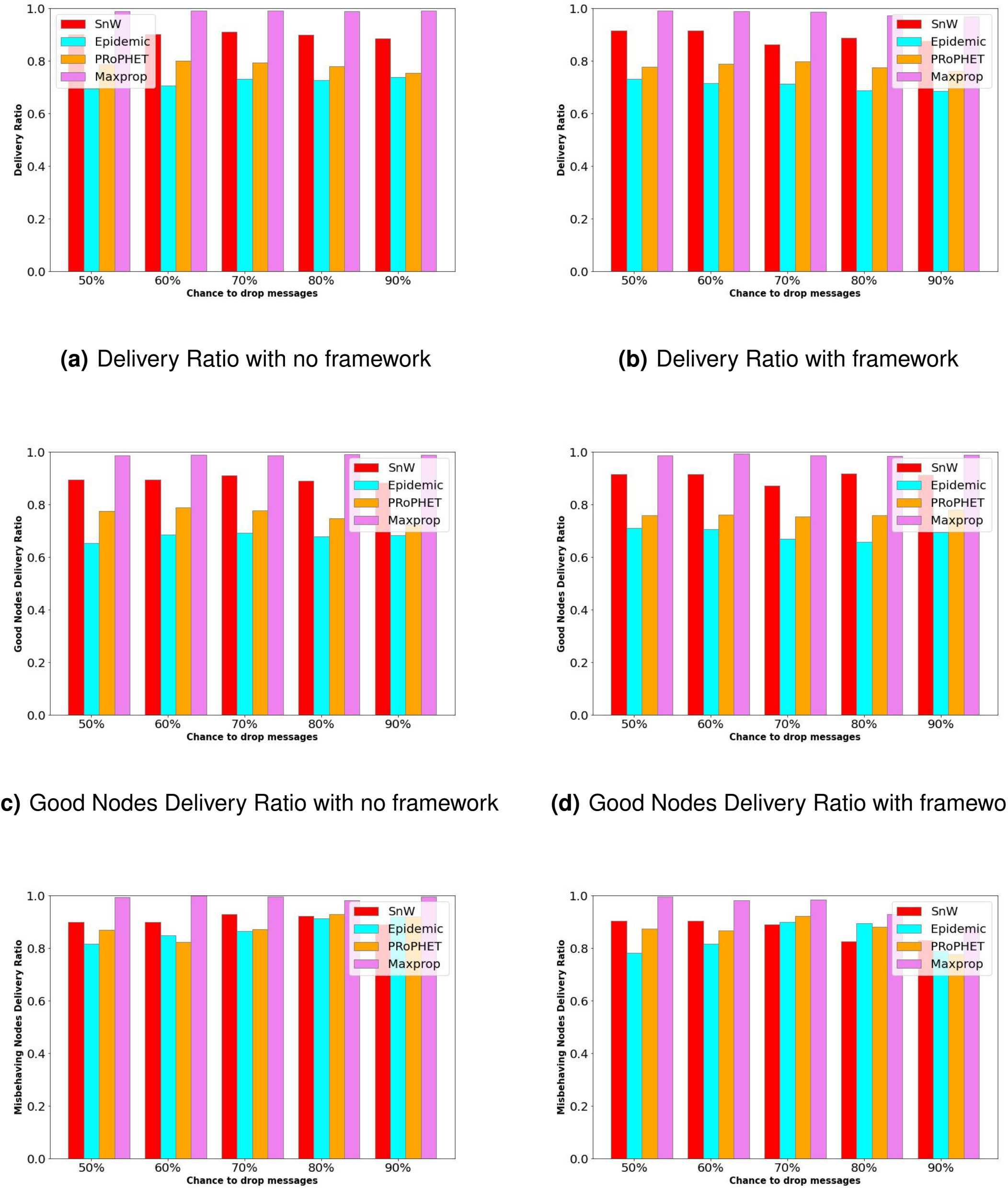
(h) False Ratio 120 nodes 50%

Figure 4.14: Comparison of Average Reputations and False Ratios for MaxProp, with 20% of misbehaving nodes with drop probabilities of 50% and 90% to test robustness of scenarios with different degrees of misbehaviour and different number of nodes

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 4.3: 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

improvements are barely noticeable; for the Epidemic routing protocol the improvements are noticeable until the 80% dropping probability; for the PRoPHET routing protocol the improvements are noticeable until the 80% dropping probability for sparser networks of 40 nodes and until the 70% dropping probability for denser networks of 120 nodes; for the MaxProp routing protocol the improvements are noticeable until the 80% dropping probability for sparser networks of 40 nodes and until the 50% dropping probability for denser networks of 120 nodes. The improvements diminish with the probability of a misbehaving node dropping messages. As we can see in table 4.3, although the 90% probability results are very similar to the ones observed before with the black-hole attacks, they are slightly worse. And when we reach the 50% drop probability the framework barely brings any change.



(a) Delivery Ratio with no framework

(b) Delivery Ratio with framework

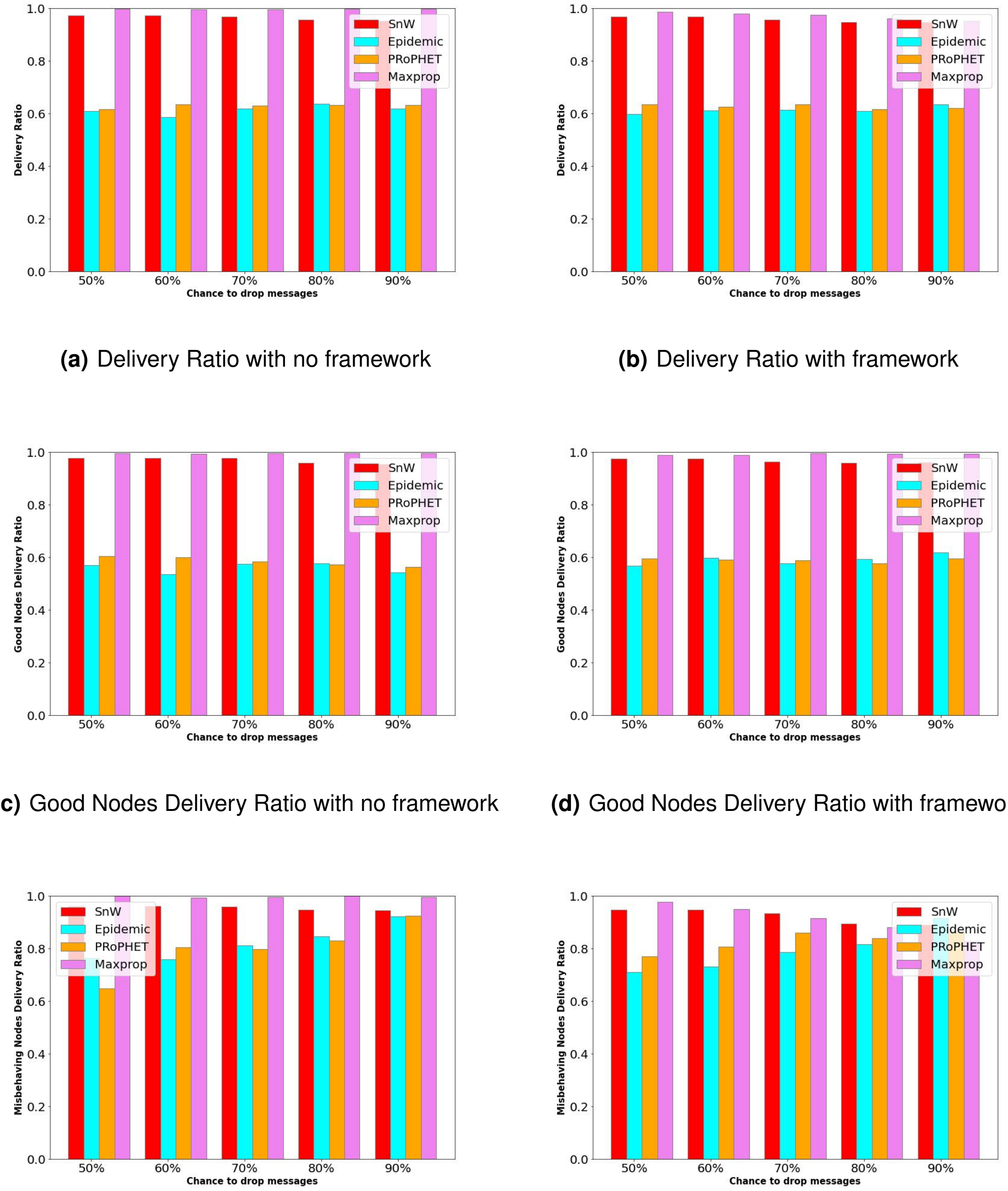
(c) Good Nodes Delivery Ratio with no framework

(d) Good Nodes Delivery Ratio with framework

(e) Misbehaving Nodes Delivery Ratio with no framework

(f) Misbehaving Nodes Delivery Ratio with framework

Figure 4.15: Comparison of Delivery Ratio, Good Nodes Delivery Ratio and Misbehaving Nodes Delivery Ratio for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes and different degrees of misbehaviour



(a) Delivery Ratio with no framework

(b) Delivery Ratio with framework

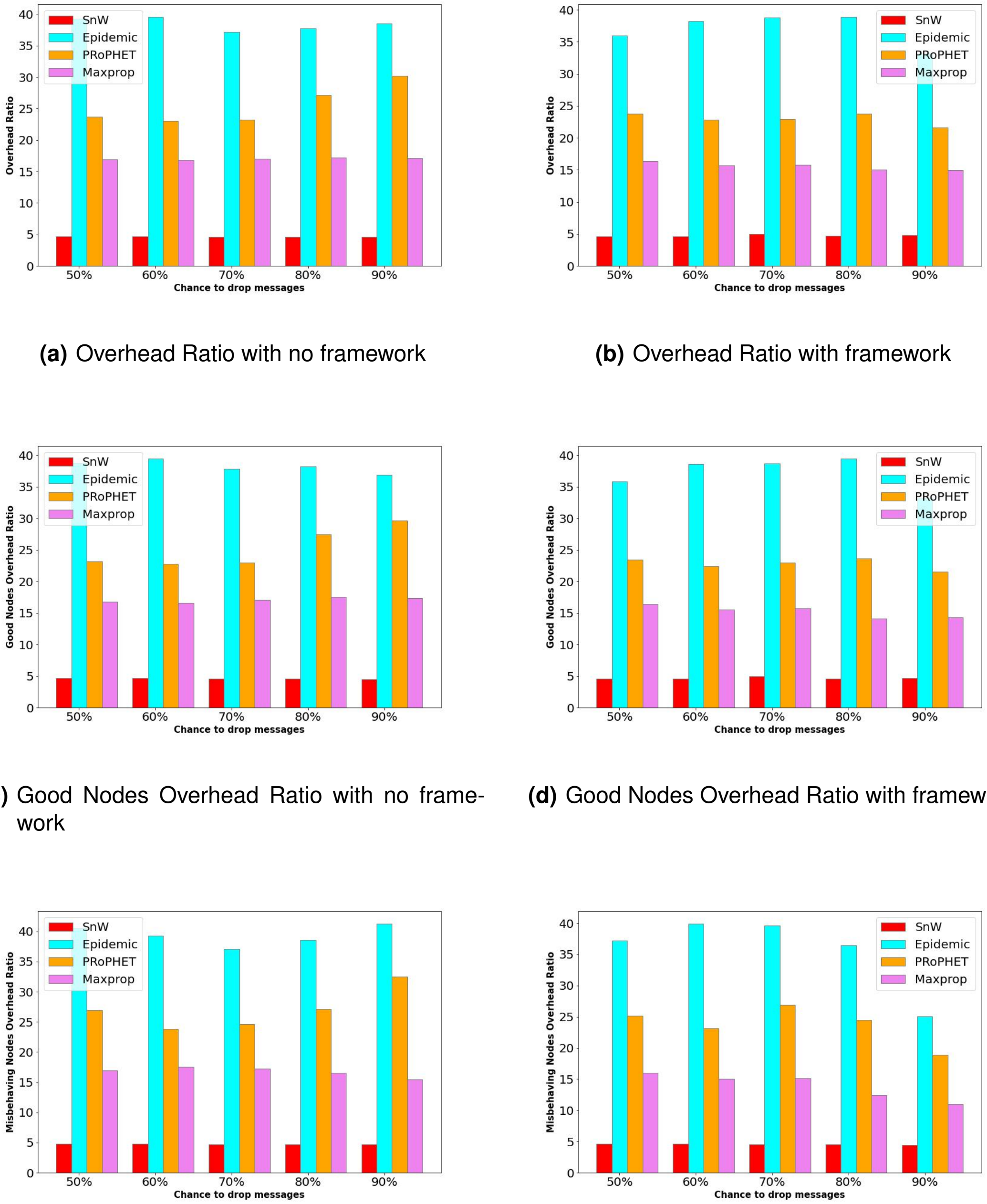
(c) Good Nodes Delivery Ratio with no framework

(d) Good Nodes Delivery Ratio with framework

(e) Misbehaving Nodes Delivery Ratio with no framework

(f) Misbehaving Nodes Delivery Ratio with framework

Figure 4.16: Comparison of Delivery Ratio, Good Nodes Delivery Ratio and Misbehaving Nodes Delivery Ratio for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes and different degrees of misbehaviour



(a) Overhead Ratio with no framework

(b) Overhead Ratio with framework

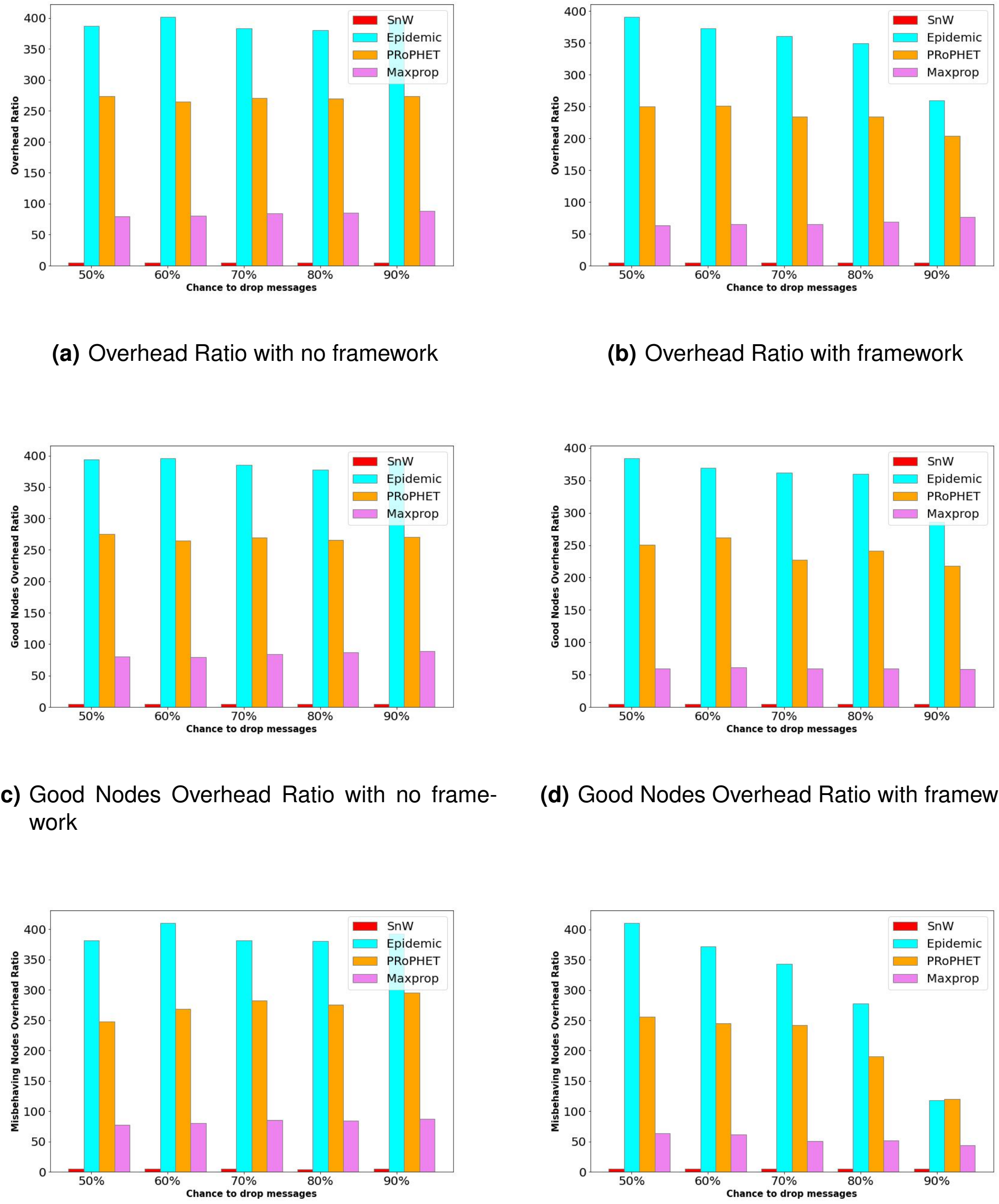
(c) Good Nodes Overhead Ratio with no framework

(d) Good Nodes Overhead Ratio with framework

(e) Misbehaving Nodes Overhead Ratio with no framework

(f) Misbehaving Nodes Overhead Ratio with framework

Figure 4.17: Comparison of Overhead Ratio, Good Nodes Overhead Ratio and Misbehaving Nodes Overhead Ratio for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes and different degrees of misbehaviour



(a) Overhead Ratio with no framework

(b) Overhead Ratio with framework

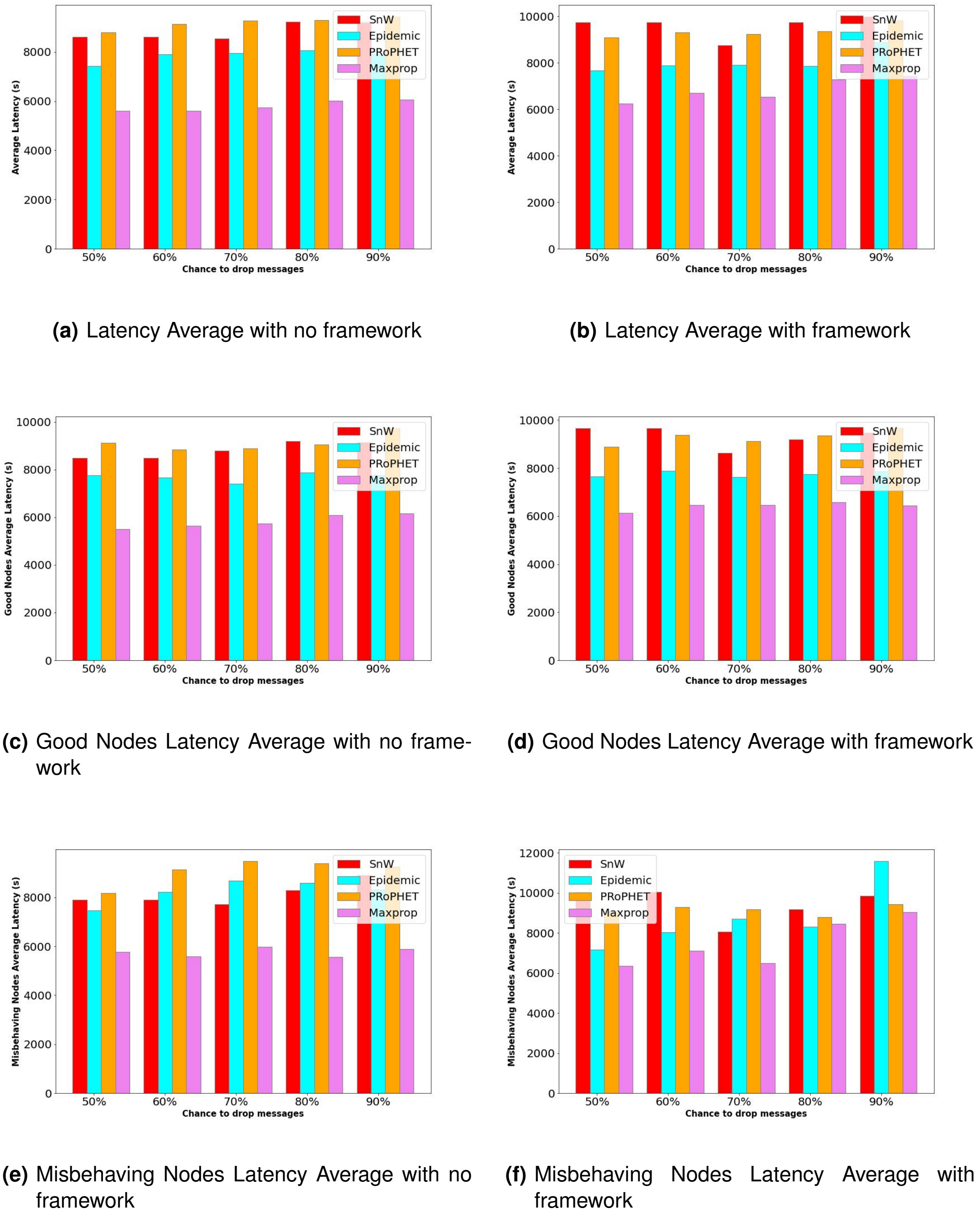
(c) Good Nodes Overhead Ratio with no framework

(d) Good Nodes Overhead Ratio with framework

(e) Misbehaving Nodes Overhead Ratio with no framework

(f) Misbehaving Nodes Overhead Ratio with framework

Figure 4.18: Comparison of Overhead Ratio, Good Nodes Overhead Ratio and Misbehaving Nodes Overhead Ratio for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes and different degrees of misbehaviour



(a) Latency Average with no framework

(b) Latency Average with framework

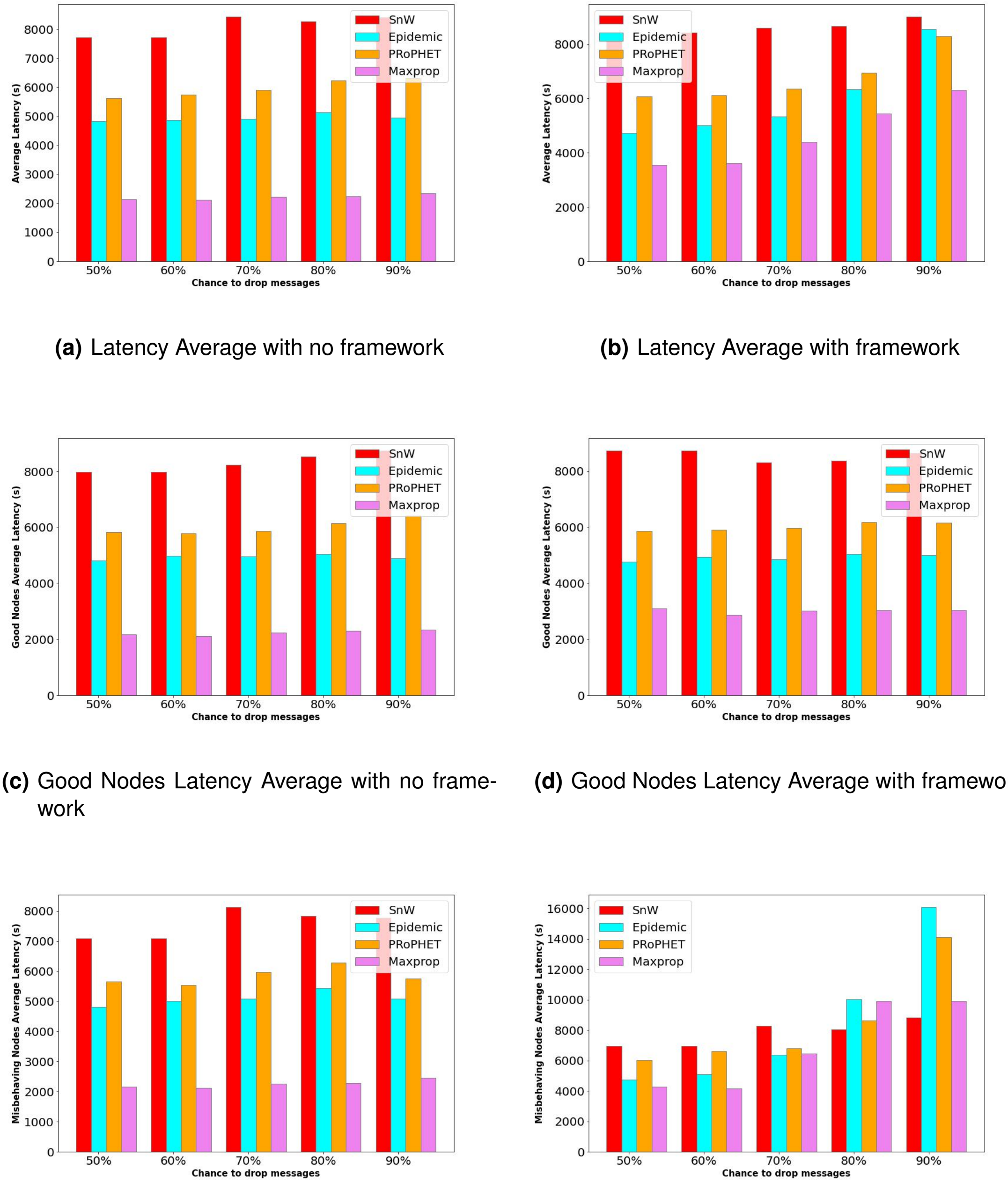
(c) Good Nodes Latency Average with no framework

(d) Good Nodes Latency Average with framework

(e) Misbehaving Nodes Latency Average with no framework

(f) Misbehaving Nodes Latency Average with framework

Figure 4.19: Comparison of Latency Average, Good Nodes Latency Average and Misbehaving Nodes Latency Average for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 40 nodes and different degrees of misbehaviour



(a) Latency Average with no framework

(b) Latency Average with framework

(c) Good Nodes Latency Average with no framework

(d) Good Nodes Latency Average with framework

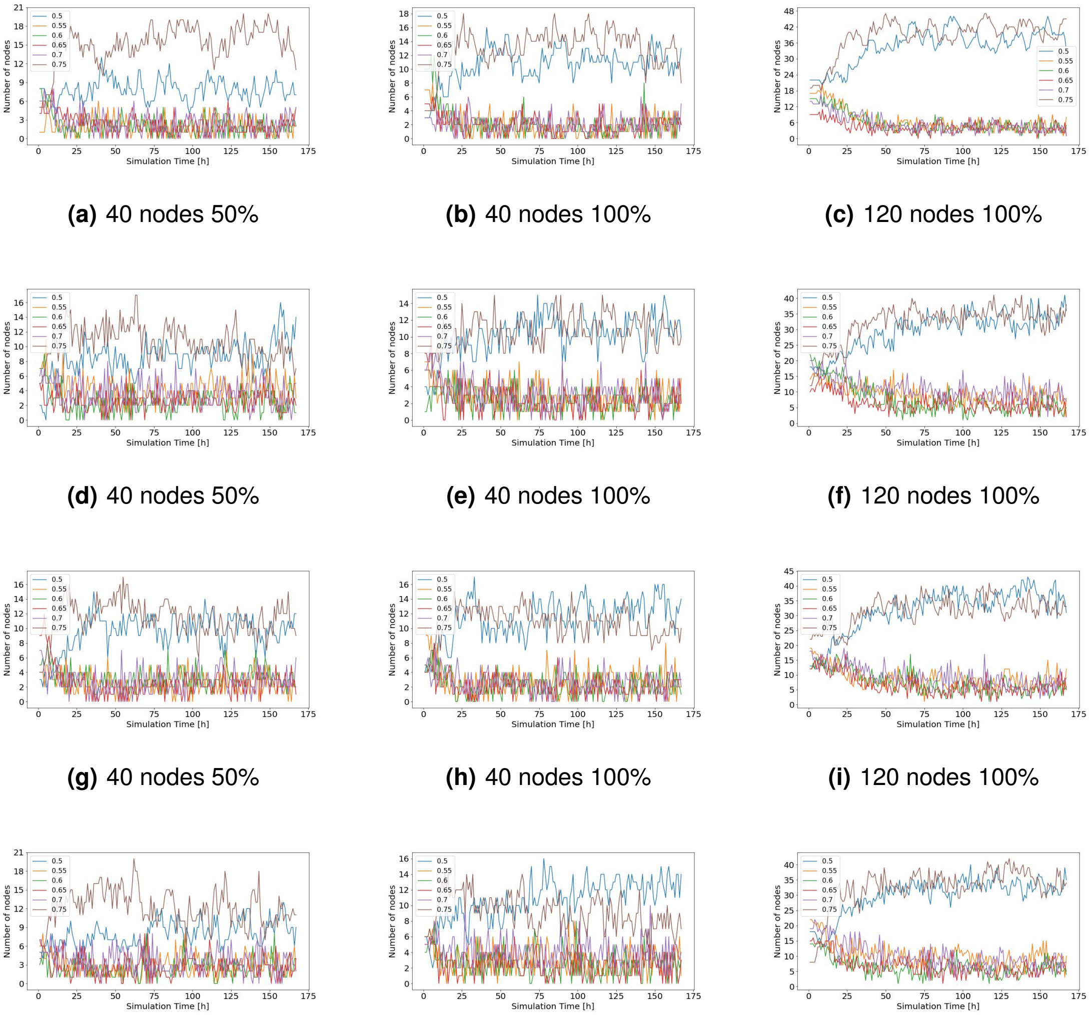
(e) Misbehaving Nodes Latency Average with no framework

(f) Misbehaving Nodes Latency Average with framework

Figure 4.20: Comparison of Latency Average, Good Nodes Latency Average and Misbehaving Nodes Latency Average for all protocols, with 20% of misbehaving nodes, with and without the framework, to test robustness of scenarios with 120 nodes and different degrees of misbehaviour

4.4 Strategy evolution

The graphs in fig. 4.21 represent how nodes change strategy thresholds over time. As we can see in the picked examples, nodes constantly change their strategy to adapt to the network. In the examples, the quantity of nodes, used protocol and percentage of dropped messages do not appear to have much of an effect on the behaviour of the nodes. In general, most nodes tend use the maximum threshold of 0.75 or the minimum threshold of 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. Although this may influence how the behaviour of the network turned out in this work, testing with a bigger spectrum of strategies and more varied situations may lead to different results and would be great in the development of this project.



(a) 40 nodes 50%

(b) 40 nodes 100%

(c) 120 nodes 100%

(d) 40 nodes 50%

(e) 40 nodes 100%

(f) 120 nodes 100%

(g) 40 nodes 50%

(h) 40 nodes 100%

(i) 120 nodes 100%

(j) 40 nodes 50%

(k) 40 nodes 100%

(l) 120 nodes 100%

Figure 4.21: Comparison of Strategy change over time in all previously tested protocols, with 20% of misbehaving nodes with drop probabilities of 50% (gray-hole nodes) and 100% (black-hole nodes) to test robustness of scenarios with different number of nodes

5

Conclusion

Contents

5.1 Achievements

This work is part of the SEEDS project (H2020-MSCA-RISE under number 101006411). The main purpose of this work is to develop an effective game theory-based incentive framework for the next-generation vehicular networks, adaptable to various routing protocols and scenarios that promotes cooperation while also diminishing the effects of DoS attacks. To achieve that goal, in chapter 2, the fundamental concepts of vehicular networks were summarized. This ranged from basic definitions of VANET and IoV to the most common routing protocols used, the most common attacks to said networks. Various types of incentive schemes were explained, with given examples for each of them, while highlighting advantages and disadvantages for each of them. Basic concepts for social norm, game theory and, more specifically, EGT were also summarized. Examples of their use in vehicular networks were also given. To finalize, this chapter presents various simulators and reasons as to why the ONE simulator was chosen to evaluate the framework in this work.

Chapter 3 focuses on the solution to the problem, by doing an in-depth explanation of the way the framework works. It starts of with the extra information each node needs to account for. It then explains how the framework distinguishes good from misbehaving nodes through a reputation system and how the node subsequently punishes them based on the distinction made by this same reputation system with a EGT based approach. To end this chapter, metrics to evaluate the framework are presented and carefully explained.

Chapter 4 presents the main results of this work with an analysis on said results. The conditions for the simulations scenarios are established. After this the results for two types of DoS attacks (gray-hole and black-hole attacks) are presented. From observing the results 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 it 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.

5.2 Future work

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.

Bibliography

[1] L. Strigini, N. F. Neves, M. Raynal, M. Harrison, M. Kaâniche, and F. v. Henke, *Resilience-Building Technologies: State of Knowledge–ReSIST NoE Deliverable D12*. Department of Informatics, University of Lisbon, 2007.

[2] 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.

[3] J. Dede, A. Förster, E. Hernández-Orallo, J. Herrera-Tapia, K. Kuladinithi, V. Kuppusamy, P. Manzoni, A. bin Muslim, A. Udugama, and Z. Vatandas, "Simulating opportunistic networks: Survey and future directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 2, pp. 1547–1573, 2017.

[4] S. Yousefi, M. S. Mousavi, and M. Fathy, "Vehicular ad hoc networks (vanets): challenges and perspectives," in *2006 6th international conference on ITS telecommunications*. IEEE, 2006, pp. 761–766.

[5] 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.

[6] 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.

[7] F. Warthman *et al.*, "Delay-and disruption-tolerant networks (dtns)," *A Tutorial. V.. 0, Interplanetary Internet Special Interest Group*, pp. 5–9, 2012.

[8] C. Nabais, P. R. Pereira, and N. Magaia, "Birep: A reputation scheme to mitigate the effects of black-hole nodes in delay-tolerant internet of vehicles," *Sensors*, vol. 21, no. 3, p. 835, 2021.

[9] T. Spyropoulos, K. Psounis, and C. S. Raghavendra, "Single-copy routing in intermittently connected mobile networks," in *2004 First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, 2004. IEEE SECON 2004.* IEEE, 2004, pp. 235–244.

[10] S. Jain, K. Fall, and R. Patra, “Routing in a delay tolerant network,” in *Proceedings of the 2004 conference on Applications, technologies, architectures, and protocols for computer communications*, 2004, pp. 145–158.

[11] A. Vahdat, D. Becker *et al.*, *Epidemic routing for partially connected ad hoc networks*. Technical Report CS-200006, Duke University, 2000.

[12] A. Lindgren *et al.*, “Probabilistic routing protocol using history of encounters and transitivity (prophet),” RFC 6693, IETF Document, Tech. Rep., 2012.

[13] 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.

[14] V. Nampally and M. R. Sharma, “A survey on security attacks for vanet,” *Security Challenges*, vol. 5, no. 10, 2017.

[15] E. Erkip, A. Sendonaris, A. Stefanov, and B. Aazhang, “Cooperative communication in wireless systems,” *DIMACS Series in Discrete Mathematics and Theoretical Computer Science*, vol. 66, pp. 303–320, 2004.

[16] E. Ahmed and H. Gharavi, “Cooperative vehicular networking: A survey,” *IEEE Transactions on Intelligent Transportation Systems*, vol. 19, no. 3, pp. 996–1014, 2018.

[17] 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)*. IEEE, 2014, pp. 124–129.

[18] 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.

[19] M. Musolesi and C. Mascolo, “Car: Context-aware adaptive routing for delay-tolerant mobile networks,” *IEEE Transactions on Mobile Computing*, vol. 8, no. 2, pp. 246–260, 2008.

[20] N. Magaia, P. Pereira, M. P. Correia, D. Rawat, J. Rodrigues, and I. Stojmenovic, “Security in delay-tolerant mobile cyber physical applications,” in *Cyber-Physical Systems: From Theory to Practice*. CRC Press, 2015, pp. 373–394.

[21] U. Shevade, H. H. Song, L. Qiu, and Y. Zhang, “Incentive-aware routing in dtns,” in *2008 IEEE international conference on network protocols*. IEEE, 2008, pp. 238–247.

[22] O. Morgenstern and J. Von Neumann, *Theory of games and economic behavior*. Princeton university press, 1953.

[23] M. Wooldridge, “Does game theory work?” *IEEE Intelligent Systems*, vol. 27, no. 6, pp. 76–80, 2012.

[24] D. E. Charilas and A. D. Panagopoulos, “A survey on game theory applications in wireless networks,” *Computer Networks*, vol. 54, no. 18, pp. 3421–3430, 2010.

[25] J. M. Smith and G. R. Price, “The logic of animal conflict,” *Nature*, vol. 246, no. 5427, pp. 15–18, 1973.

[26] J. M. Smith, *Evolution and the Theory of Games*. Cambridge university press, 1982.

[27] S. Shivshankar and A. Jamalipour, “An evolutionary game theory-based approach to cooperation in vanets under different network conditions,” *IEEE Transactions on Vehicular Technology*, vol. 64, no. 5, pp. 2015–2022, 2014.

[28] D. Wu, Y. Ling, H. Zhu, and J. Liang, “The rsu access problem based on evolutionary game theory for vanet,” *International Journal of Distributed Sensor Networks*, vol. 9, no. 7, p. 143024, 2013.

[29] Z. Tian, X. Gao, S. Su, J. Qiu, X. Du, and M. Guizani, “Evaluating reputation management schemes of internet of vehicles based on evolutionary game theory,” *IEEE Transactions on Vehicular Technology*, vol. 68, no. 6, pp. 5971–5980, 2019.

[30] C. Wu, M. Gerla, and M. van der Schaar, “Social norm incentives for network coding in manets,” *IEEE/ACM Transactions on Networking*, vol. 25, no. 3, pp. 1761–1774, 2017.

[31] 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*, 2009, pp. 1–10.

[32] C. Sommer, R. German, and F. Dressler, “Bidirectionally coupled network and road traffic simulation for improved ivc analysis,” *IEEE Transactions on mobile computing*, vol. 10, no. 1, pp. 3–15, 2010.