



**HAL**  
open science

# A consensus-based approach to reputational routing in multi-hop networks

Edward Staddon, Valeria Loscri, Nathalie Mitton

► **To cite this version:**

Edward Staddon, Valeria Loscri, Nathalie Mitton. A consensus-based approach to reputational routing in multi-hop networks. ITU Journal on Future and Evolving Technologies, 2023. hal-03969218

**HAL Id: hal-03969218**

**<https://inria.hal.science/hal-03969218>**

Submitted on 2 Feb 2023

**HAL** is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

# A CONSENSUS-BASED APPROACH TO REPUTATIONAL ROUTING IN MULTI-HOP NETWORKS

Edward Staddon<sup>1</sup>, Valeria Loscri<sup>1</sup>, Nathalie Mitton<sup>1</sup>

<sup>1</sup>Inria, Lille, France

NOTE: Corresponding author: Edward Staddon, edward.staddon@inria.fr

---

**Abstract** – When it comes to the security of the Internet of Things (IoT), securing their communications is paramount. In multi-hop networks, nodes relay information amongst themselves, opening the data up to tampering by an intermediate device. To detect and avoid such malicious entities, we grant nodes the ability to analyse their neighbours behaviour. Through the use of consensus-based validation, based upon blockchain’s miners, all nodes can agree on the trustworthiness of all devices in the network. By expressing this through a node’s reputation, it is possible to identify malicious devices and isolate them from network activities. By incorporating this metric into a multi-hop routing protocol such as AODV, we can influence the path selection process. Instead of defining the best route based upon overall length, we can chose the most reputable path available, thus traversing trustworthy devices. By performing extensive analyses through multiple simulated scenarios, we can identify a decrease in packet drop rates compared to AODV by  $\approx 48\%$  and  $\approx 38\%$  when subjected to black-hole attacks with 30 and 100 node networks respectively. Furthermore, by subjecting our system to varying degrees of grey-holes, we can confirm its adaptability to different types of threats.

**Keywords** – Consensus, Cyber Security, IoT, Reputation, Routing

## 1. INTRODUCTION

The Internet of Things (IoT) has become part of our everyday lives, providing services in multiple areas. From "Smart" equipment to wearable healthcare devices, the IoT processes a lot of important and sensitive data. Furthermore, as is the case with wearable healthcare devices such as a pacemaker, by allowing a connection with the open Internet, we also open the corresponding attack surface to new threats [1]. This can result in the loss of sensitive data and can even go as far as cause significant health risks to the patient. In some use cases such as smart agriculture, IoT devices must operate in hostile environments where a direct connection with a base station or access point is not always available. To maintain communications, these devices employ the multi-hop paradigm, allowing intermediate nodes to transmit and relay passing packets to their destination. However, in doing so, we also increase the chance of attack, as any node in our network could compromise our routing activities [2].

One means to provide an extra layer of security is allowing nodes to only converse with neighbours that they trust. The notion of trust is deeply embedded in the human psyche and is a main contributor to how we form relationships. The parameters of how trust is defined varies from person to person, however, a fundamental element is the notion of reputation, where the higher the reputation, the more likely we are to trust said person or entity. Indeed, although the reputation influences the trust value, the opposite is also the case, where breaking someones trust severely impacts that person’s

reputation. By rendering the reputation of someone or something common knowledge, any change will be perceived by everyone, meaning that any impact will have inevitable repercussions. This system can be applied to the digital networking world where nodes possess a known reputation value, allowing their neighbours to determine if they can be trusted. As a result, in a similar fashion to human interactions, if a node acts badly in the network, their reputation will decrease, allowing easy separation between malicious entities and good trustworthy nodes.

In multi-hop IoT networks, nodes are generally left to their own devices, operating as configured and routing data when needed. This means, there is no shared memory between devices, meaning that data must be actively provided to each node for them to know it. This is important since as we said previously, the reputation values are known by all nodes in the network. A well known method for sharing data in a distributed manner whilst maintaining data integrity is through the use of the blockchain [3]. Made popular through its uses in many different cryptocurrencies, such as the infamous Bitcoin [4], the blockchain brings many elements to the table which can be of use. The blockchain employs devices known as "miners" which are responsible for the creation, validation and addition of new data in the form of blocks, into the chain itself. These miners employ a *Proof of Work (PoW)* technique for block validation, ensuring that only valid blocks get input into the blockchain, reducing the risk of incorrect data injection. To allow data to traverse multi-hop networks, many

63 routing protocols exist, each with their own advantages. 119  
64 By incorporating the newly acquired knowledge of node 120  
65 reputation thanks to the blockchain, intermediate nodes 121  
66 are now capable of not only determining the trustwor- 122  
67 thiness of their neighbours, but also influencing their 123  
68 routing abilities. Many routing protocols use various 124  
69 metrics to determine the best route to take towards the 125  
70 destination which could be influenced by a nodes rep-  
71 utation. This is the case of the Ad hoc On-Demand  
72 Distance Vector (AODV) routing protocol, where the  
73 route with the lowest hop count is preferred [5]. Being  
74 a reactive protocol, route discovery is only performed  
75 when needed, meaning accurate up-to-date reputational  
76 values can be used. During route discovery, the source  
77 node broadcasts a Route Request (RREQ) packet, ask-  
78 ing for a route towards the destination. This packet is  
79 relayed by each node it encounters, each one increasing  
80 the hop count by one, until the requested destination is  
81 reached. The destination then responds back via unicast  
82 towards the source with a Route Reply (RREP) using  
83 the shortest route available. By analysing the trust-  
84 worthiness of each node, we can influence the *hop-count*  
85 to increase the corresponding "length" the more mali-  
86 cious nodes are present. As a result, AODV would nat-  
87 urally select the shortest route, only here this doesn't  
88 correspond to the least number of hops, but the highest  
89 trustworthiness overall.

90 In this paper, we propose a consensus-based module for  
91 routing protocols using reputation metrics to determine  
92 the most trustworthy route in the network. The main  
93 contributions are as follows:

- 94 • Firstly, we perform an analysis of previous works  
95 in the literature around the notion of "reputation"  
96 as well as different uses of blockchain, in partic-  
97 ular their applications to wireless routing activi-  
98 ties. We also explore the different security improve-  
99 ments which have been proposed for AODV in re-  
100 cent years.
- 101 • Next we define and propose updated metrics based  
102 on previous works for the computation of nodes rep-  
103 utation, as well as the addition of a *Reputation De-*  
104 *cay* system, allowing nodes to be reintegrated into  
105 the network after a certain period of inactivity. We  
106 also explain how a consensus-based configuration  
107 inspired from blockchain's miners which allows us  
108 to grant the network the ability to adapt and deter-  
109 mine these values without prior knowledge, before  
110 sharing the results throughout the network thanks  
111 to blockchain technology.
- 112 • We also present how our system can be incorpo-  
113 rated into a reactive routing protocol, in this case  
114 AODV as well as a few updates to the existing pro-  
115 tocol, allowing our system to function at peak effi-  
116 ciency.
- 117 • Finally, we analyse the performance of this new pro-  
118 tocol, called *AODV-Miner*, by comparing it to basic

AODV functionality in extensive simulations with  
networks of 30 and 100 nodes with varying net-  
work topologies. By pitching both protocols against  
black and grey hole with varying degrees of mali-  
cious presence and intentions, we demonstrate a re-  
duction in packet drop rates by  $\approx 48\%$  and  $\approx 38\%$   
with 30 and 100 nodes respectively.

126 The rest of this paper is organised as follows: Section  
127 2 analyses previous work in the areas of reputation,  
128 blockchain and AODV security and presents the differ-  
129 ences with our module. Section 3 defines our system  
130 model, before presenting our module and *AODV-Miner*  
131 in Section 4. Then, Section 5 explains our implemen-  
132 tation and simulation parameters before analysing the  
133 results in Section 6. Finally, we discuss these results  
134 and future endeavours in Section 7 before concluding  
135 this paper in Section 8

## 136 2. RELATED WORKS

137 Our system is based around two distinct elements: Rep-  
138 utation and Blockchain; and also uses a third in our  
139 analysis: AODV. Each of these notions are not new and  
140 have been extensively evaluated in the scientific litera-  
141 ture. Furthermore, AODV has seen many new proposi-  
142 tions to upgrade its functionality and security since its  
143 elaboration. However, as far as we are aware, none use  
144 a dynamically elected consensus-based reputation sys-  
145 tem, derived from blockchain's miners. In this section  
146 we present these three elements as well as an analysis of  
147 some of the improvements they have received and their  
148 uses in routing activities before defining our system and  
149 its differences.

### 150 2.1 Behavioural Reputation

151 Inspired from the human psyche, the notion of repu-  
152 tation can be applied to an IoT network, where here  
153 nodes will chose a higher, more reputable neighbour  
154 over others. This is the case of [6] where the authors  
155 use trust-based methods to identify nodes in the net-  
156 work, based on their previous activities. By evaluating  
157 multiple types of activities based on node social inter-  
158 actions and QoS, the resulting trust profiles are evalu-  
159 ated by other nodes before being adopted. In a sim-  
160 ilar fashion, [7] integrates this functionality into their  
161 routing protocol for Wireless Sensor Networks, where  
162 they compute a trust value per node, based upon their  
163 previous activities. By analysing their sincerity in for-  
164 warding data, acknowledging previous packets as well as  
165 the nodes energy consumption, this value is then used  
166 to determine the most trustworthy candidate to relay  
167 the data throughout the network. However, reputation  
168 and trust metrics can be expressed in multiple fashions.  
169 For example, the authors of [8] evaluates neighbouring  
170 behavioural patterns using inter-node cooperation. On  
171 the other hand, the authors of [9] use a signature based  
172 methodology, validating data integrity and confirming

173 if data has reached the intended sink.

## 174 2.2 Blockchain-Based Sharing

175 The main advantage of the blockchain is its immutabil-  
176 ity [10], which has led it to being used in many other  
177 areas, such as that of IoT security [11]. However, they  
178 possess many challenges related to the specific context  
179 of the IoT, such as resource limitations and data man-  
180 agement where power hungry *PoW* and block storage  
181 become a problem. That being said, the blockchain has  
182 seen its fair share of attention in the area of security,  
183 such as providing authentication and trust services to  
184 the IoT [12] and increasing data integrity and authentic-  
185 ity [13]. Since our interests revolve around routing, we  
186 concern ourselves with the different methods employed  
187 to increase routing security [14].

188 An example is the work performed by the authors of  
189 [15]. Here the blockchain stores information related to  
190 the data transmission, allowing all nodes to participate  
191 in determining the "legality" of the exchanges. In [16],  
192 the authors use the blockchain to store and share the sta-  
193 tus of the network in real-time to enhance the routing  
194 process. By checking the list of transactions, nodes can  
195 determine the most efficient route, thus avoiding con-  
196 gested areas and nodes. This technology has also been  
197 used in Unmanned Aircraft Systems as in [17], improv-  
198 ing both routing activities and authentication. Here,  
199 a lightweight blockchain deployment is used, providing  
200 each drone with identification and authentication infor-  
201 mation. The authors of [18] propose a novel routing  
202 protocol based on blockchain contractual methodology.  
203 By using the ledger to store smart contract addresses  
204 indicating when routing is needed, routes can be offered  
205 and determined when needed.

## 206 2.3 AODV Routing Protocol

207 AODV related security has been an interest in the litera-  
208 ture for some time since its original conception. Indeed,  
209 AODV is susceptible to multiple types of attacks [19]  
210 targeting packet control fields, such as source and desti-  
211 nation IP or sequence numbers, as well as hop-count  
212 forging. As a result, the authors of [19] propose an  
213 Intrusion Detection System, capable of detecting and  
214 countering these vulnerabilities by comparing the net-  
215 works activities to predefined specifications where any  
216 deviation is considered malicious. The authors of [20]  
217 take a different standing point, directly targeting cer-  
218 tain vulnerabilities in an effort to enhance the overall  
219 security. Their Intrusion Detection Model allows the  
220 detection of multiple attacks, such as Denial of Service,  
221 impersonation or a compromised node, which is then iso-  
222 lated from network activities by the Intrusion Response  
223 Models. In all, their approach is capable of increasing  
224 the routing efficiency, rendering AODV more robust, as  
225 the slight cost of a higher overhead. In [21], the authors  
226 use advanced numerical analysis to increase the secu-

227 rity of AODV during routing. By using methods such  
228 as cryptography or numerical sequences, they are able  
229 to increase the overall performance when subjected to  
230 black-hole attacks.

231 Reputation-based metrics and blockchain have also been  
232 used in line with AODV. Indeed, in [22], the authors  
233 extend the AODV-UU protocol to incorporate reputa-  
234 tion based metrics, identifying malicious and trustwor-  
235 thy nodes. By integrating the reputation value directly  
236 into the discovery process, it is possible to identify paths  
237 passing through malicious nodes, allowing them to be  
238 avoided. Regarding the blockchain, the authors of [23]  
239 propose the protocol BAODV, using blockchain's hash  
240 chaining to authenticate nodes and confirm data in-  
241 tegrity. By incorporating the IP address of malicious  
242 nodes in the discovery messages, BAODV can in cir-  
243 cumnavigate the malicious entities. Another approach  
244 used in [24] is the construction of a blockchain network,  
245 allowing the identification of routes towards the des-  
246 tination. Each path node is added to the blockchain  
247 network, avoiding malicious entities and identifying the  
248 most optimal route to take. In [25], the authors unite  
249 both elements, using reputation-based metrics to influ-  
250 ence routing activities and the blockchain to distribute  
251 the reputation throughout the network. Their approach  
252 includes an extension to the reputation metric where the  
253 length of a route is manipulated dependant on the nodes  
254 reputation, lengthening it if they possess malicious ten-  
255 dencies. In regards to blockchain dissemination, the au-  
256 thors also define specific network grids in which miners  
257 are identified and are responsible for the computation  
258 of the reputation and blockchain distribution. This ap-  
259 proach allows the type of node to be exploited, privi-  
260 leging powerful nodes for this role over weaker counter-  
261 parts. However, once nodes have been defined as miners  
262 they cannot partake in routing activities, which reduces  
263 the number of potential relays in the network.

## 264 2.4 Our Contribution

265 To define our system, we take inspiration from multi-  
266 ple approaches, in particular [25]. However, one major  
267 difference is that our module is not directly integrated  
268 into a specific routing protocol, but can be adapted  
269 to fit others, influencing and exploiting the route dis-  
270 covery and upkeep functionalities. By doing so, we  
271 allow the ability to dynamically build a route profile,  
272 meaning no prior knowledge of the network or nodes is  
273 needed. Furthermore, by updating the previously anal-  
274 ysed reputation-based approaches to use this dynamic  
275 route profile, we allow nodes to identify activities which  
276 distinctly deviate from expected, the main advantage of  
277 which is no need for any advanced or heavy techniques.  
278 We also define a lightweight version of blockchain, simi-  
279 lar to [17], significantly reducing its role to that of a  
280 dissemination tool with lower weight and complexity.  
281 We also repurpose its miners to perform behavioural  
282 validation responsibilities, similar to [25], however, we

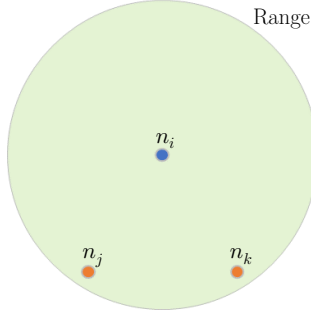


Fig. 1 – Communication range of node  $n_i$

283 include the addition of dynamic role selection, allow-  
 284 ing nodes to take on the role of miners or routers at  
 285 will. By not defining specific roles at the start, the net-  
 286 work can, therefore, adapt to fluctuating typologies and  
 287 also take advantage of new nodes with no user inter-  
 288 vention needed. This paradigm also redefines the re-  
 289 source intensive *PoW* process, into a consensus-based  
 290 validation system, allowing nodes to select the best re-  
 291 sults to be shared throughout the network. As a result,  
 292 our new *Validation Miners* differ significantly from their  
 293 blockchain counterparts, all the while holding key posi-  
 294 tions in the network.

### 3. SYSTEM MODEL & THREAT TAXONOMY

295  
 296  
 297 Our system is based around specific models and threat  
 298 information. In this section we explore both our net-  
 299 work and validations models, before taking a look at  
 300 our threat taxonomy.

#### 3.1 Network Model

301  
 302 We consider an interconnected wireless network scenario  
 303 with  $N$  static nodes, each possessing a fixed transmis-  
 304 sion range. Each node has at least one other in com-  
 305 munications range, called a neighbour, forming a par-  
 306 tial mesh topology, an example is shown in Figure 1.  
 307 We can see that node  $n_i$  possesses a fixed transmission  
 308 range, encompassing two other nodes, its neighbours.  
 309 These interconnections allow any one node to contact  
 310 all others in the network, resulting in both stable con-  
 311 nections and durable routes. As we can see in the figure,  
 312 multiple nodes can be in range of multiple others. By  
 313 using the wireless medium, we accept that it is possible  
 314 for inevitable transmission overlaps to occur, result-  
 315 ing in areas of collision. Our choice of using AODV as a  
 316 base for our system means that the nodes already take  
 317 on certain characteristics which are useful to our system.  
 318 For a reactive protocol to function correctly, all partici-  
 319 pating devices must be capable of receiving any routing  
 320 related traffic at any given time. As a result, we consider  
 321 that all nodes remain in an active listening state, con-  
 322 stantly analysing all passing packets waiting for a poten-  
 323 tial AODV discovery message. Our nodes also possess  
 324 the ability to decide on their own role per participated

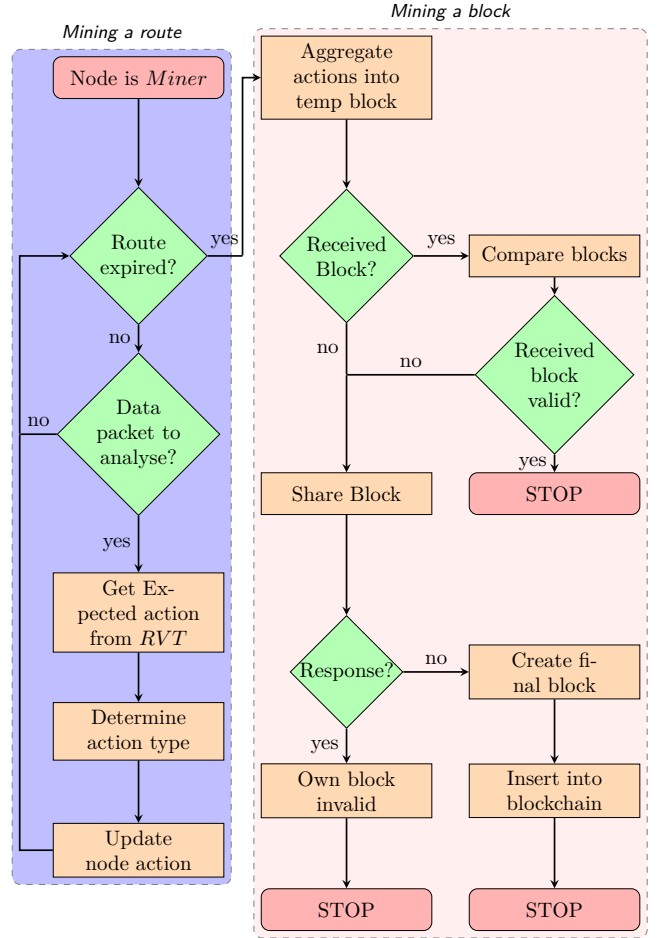


Fig. 2 – Validation flowchart

325 route, making them either a routing node (**forwarding**  
 326 **information along the corresponding route**), or a valida-  
 327 tion miner (**observing and confirming the routing activi-**  
 328 **ties of neighbouring routing nodes for the same route**).  
 329 **Both roles are mutually exclusive for each route, mean-**  
 330 **ing a miner cannot participate in routing activities, as**  
 331 **this would be a conflict of interest**. With the additional  
 332 ability of being able to participate in multiple routes si-  
 333 multaneously, the nodes can, therefore, take on multiple  
 334 roles.

#### 3.2 Validation Model

335  
 336 As stated previously, each and every node has the ability  
 337 to become a validation miner and, as a result, partici-  
 338 pate in validation activities. The role of these miners is  
 339 twofold, illustrated in Figure 2:

- 340 1. They are responsible for validating routing behav-  
 341 iour between their neighbours, which we define  
 342 as "*mining a route*".
- 343 2. They confirm and distribute the resulting behav-  
 344 ioural analysis throughout the network in  
 345 blockchain form, which we define as "*mining a*  
 346 *block*".

347 To reach their first objective, *mining a route*, the min-

348 ers must possess the ability to validate the behaviour  
 349 of their neighbours. This is achieved by allowing all  
 350 nodes to overhear and analyse passing RREP packets,  
 351 from which each miner can extract the expected for-  
 352 wards ( $src \rightarrow dst$ ) and reverse ( $dst \rightarrow src$ ) hops. These  
 353 are then added to their respective *Route Validation Ta-*  
 354 *bles (RVT)*, allowing the miners to verify all passing  
 355 data packets along the corresponding route, thus imme-  
 356 diately detecting when a deviation occurs. Upon over-  
 357 hearing a network transaction, the miner classifies the  
 358 resulting communication as either *Good* or *Bad*, depend-  
 359 ing if the activity was expected or not. A more in-depth  
 360 distinction between the two activities is presented be-  
 361 low. Figure 1 depicts this process where since nodes  $n_j$   
 362 and  $n_k$  are in  $n_i$ 's transmission range,  $n_i$  is in a posi-  
 363 tion to overhear all of their messages. All activities are  
 364 accumulated and stored for each neighbouring node of  
 365 the mined route. As stated previously, with wireless  
 366 transmissions comes the possibility of collisions or jam-  
 367 ming attacks. As a result, it is possible that miners end  
 368 up in the overlapping transmission zones, meaning they  
 369 cannot correctly perform their activities. Since this is a  
 370 general wireless issue, we address this problem for the  
 371 miners to the best of our ability, through the possibility  
 372 of multiple miners per route. This means that multiple  
 373 miners can overhear and validate the same nodes, de-  
 374 creasing the chance of all being jammed, increasing the  
 375 efficiency and resiliency of our system.

376 Once the route expires from the routing tables, the min-  
 377 ers transition into their second activity: confirmation  
 378 and dissemination, visible on the right of Figure 2. To  
 379 begin, each miner aggregates all results for each node  
 380 in communications range for that route into a tempo-  
 381 rary block. These blocks are shared amongst surround-  
 382 ing miners which all partake in the confirmation pro-  
 383 cess. As a result, only blocks confirmed by consensus  
 384 are deemed valid and disseminated throughout the net-  
 385 work via the blockchain. We use the blockchain here  
 386 as it provides a secure means for both confirming and  
 387 sharing the different blocks. However, our lightweight  
 388 version, although following the basic blockchain princi-  
 389 pal, differs in certain aspects. The main difference is the  
 390 adaptation of the *Proof of Work* for block confirmation,  
 391 where here miners simply compare the received block  
 392 with their own, only responding if a difference has been  
 393 detected. This approach keeps the notion of consensus,  
 394 where the most common block will be kept, all the while  
 395 reducing network traffic between miners. As a result, a  
 396 miner having transmitted their block and not received a  
 397 response deems their own valid, incorporating it into the  
 398 blockchain and disseminating throughout the network.  
 399 The resulting blocks permit all nodes to updating the  
 400 reputation for all participating nodes. **It is, however,**  
 401 **important to note that our current model omits possible**  
 402 **threats towards the validation process itself. This choice**  
 403 **was motivated by our desire to demonstrate the feasibil-**  
 404 **ity of our security module, before further analysing and**  
 405 **proposing advanced security protocols to prop up this**

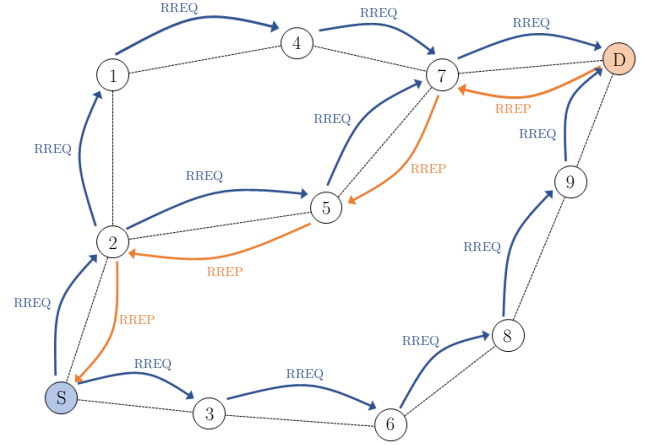


Fig. 3 – AODV discovery process

406 vulnerability.

### 407 3.3 Threat Taxonomy

408 Threat detection in our system is reduced to a binary  
 409 operation, since all miners possess the knowledge of the  
 410 expected route. Explicitly, if a routing node transmits  
 411 a valid data packet towards the correct next hop for its  
 412 destination, then it has performed a *Good* action. Any  
 413 other action is considered *Bad* and, therefore, identified  
 414 as a malicious activity. As such, our system is capable  
 415 of detecting multiple types of active threats, simply by  
 416 their actions during forwarding. Table 1 presents a brief  
 417 taxonomy of threats which can be fully, or partially de-  
 418 tected. It is important to note that some threats also  
 419 possess passive variants. Contrary to their active cousin,  
 420 these threats hide in the background and do not impact  
 421 day-to-day operations and are generally considered to be  
 422 reconnaissance related, such as packet sniffing or eaves-  
 423 dropping [26]. Since these are impossible to detect in  
 424 our context, only active threats are considered.

#### 425 3.3.1 Routing Threats

426 Possibly the most important action in a multi-hop net-  
 427 work is the act of routing itself. As a result, it is im-  
 428 portant to reduce and eliminate any threat which seeks  
 429 to impact network performance. By not transmitting  
 430 towards the expected next hop, a malicious node can  
 431 either transmit to the wrong next hop, or not trans-  
 432 mit it at all. For example in Figure 3, node 7 can use  
 433 Packet Redirect (*RTE07*) to deviate a packet from node  
 434 5 packet to node 4 instead of the destination. In the  
 435 same idea, by destroying all packets with Black-hole at-  
 436 tack (*RTE03*) or only some with a Grey-hole type attack  
 437 (*RTE04*, *RTE01* & *RTE06*), data will never reach the  
 438 destination. In either case, any deviation from the next  
 439 expected hop will result in immediate detection by the  
 440 miners. This also functions with other attacks, such  
 441 as Sinkhole (*RTE02*) or Wormhole (*RTE05*), which can

**Table 1** – System Active Threat Taxonomy

Threat Type	Threat ID	Threat	Description
Routing	RTE01	On-Off Attack	Random activation, dropping all or selectively dropping packets then randomly deactivate, causing periods of no attack where all packets are transmitted
	RTE02	Sinkhole	Trick other nodes to route traffic to a central point, allowing modification, dropping or forwarding at will to original destination or external device
	RTE03	Black-hole	All messages passing through a black-hole device are dropped, no exceptions
	RTE04	Grey-hole	Some messages passing through a grey-hole device are dropped, either randomly or by specific criteria
	RTE05	Wormhole	All messages passing through a wormhole device are captured and forwarded to another location inside/outside the network
	RTE06	Selective Forwarding	Similar to grey-holes, packets are forwarded or dropped based on specific criteria, or simply at random
	RTE07	Packet Redirect	Redirect passing traffic to wrong destination, or wrong next hop
Data	DTA01	Message Modification	Changing the content of passing messages, either at random or corresponding to specific criteria, changing the end result of the transmitted data
	DTA02	Replay	Capture a passing packet and replay it with or without modification at a later date
Node	NDE01	Byzantine	Multiple nodes are compromised and behave in an arbitrary manner causing network disruption
	NDE02	Node Capture	A node is compromised, granting ability to impact and control the network
	NDE03	Malicious Node	A node is compromised, transmitting false information to the network
	NDE04	False Node	A new node is added to the network, potentially replacing existing node, injecting false data as well as disrupting routing or spreading malicious code to other nodes, taking over them or destroying them from the inside

442 use another medium to reroute data, such as nodes 7  
443 and 3 being connected using a cellular connection, thus  
444 elongating the route taken. In any case, since no cor-  
445 responding transmission is detected by the miners, this  
446 activity is considered malicious. It is important to note,  
447 however, that some of these attacks can impact multi-  
448 ple aspects of the network. For example, a Sinkhole at-  
449 tack manipulates routing tables to force traffic to transit  
450 through it, allowing it free access to the data. Although  
451 our system is capable of detecting deviations in expected  
452 routing, it is not currently specialised in detecting ma-  
453 nipulations of AODV route discovery itself.

### 454 3.3.2 Data Threats

455 When sharing data, especially using the wireless  
456 medium, data integrity and privacy become an issue.  
457 Our taxonomy presents two data based threats which

458 can be detected. The first concerns Message Modifi-  
459 cation (*DTA01*) which directly impacts data integrity  
460 by modifying the packets payload or even header. The  
461 second concerns the re-transmission of previously send  
462 messages, known as Replay (*DTA02*). To counter these  
463 threats, miners keep records of passing messages, allow-  
464 ing them to detect sudden changes to data integrity and  
465 resurfacing of previously encountered packets. Further-  
466 more, since miners can only function when a route is  
467 present, if a packet is re-transmitted after the route has  
468 expired and no other is active, it is immediately dis-  
469 carded and considered malicious.

### 470 3.3.3 Node-Based Threats

471 When nodes are left to their own devices without reg-  
472 ular maintenance or surveillance, tampering becomes a  
473 threat. In many cases, gaining access to existing de-



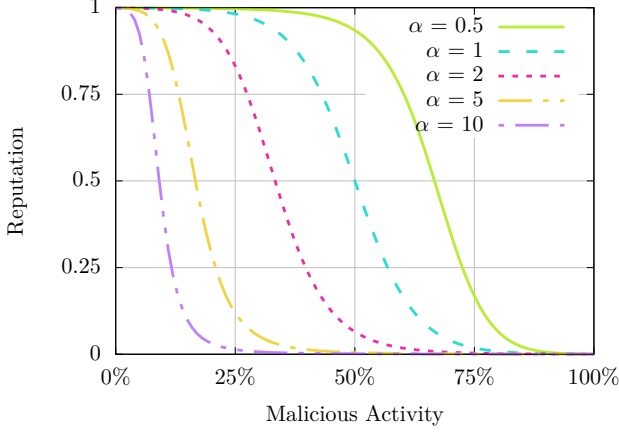


Fig. 4 – Reputation Evolution

vices, or injecting a new node (*NDE04*) into a network provides surveillance capabilities to the malicious party. Although these threats are not detectable in our context, four active node-based threats are, however, they are only detectable in certain conditions. For example, if Node 7 in Figure 3 aims to impact routing efficiency, then all deviations will be detected by the miners, which is the case of Byzantine attacks (*NDE01*). Captured, malicious or even new nodes (*NDE02*, *NDE03* & *NDE04*), can also be detected when acting upon the routing process or through modifying messages. However, if their goal is to legitimately inject invalid data into the network, then these threats are not detected.

## 4. CONSENSUS-BASED ROUTING

In this section, we present a consensus-based routing module using reputation metrics, implemented on top of the AODV protocol called *AODV-Miner*.

### 4.1 Behavioural Analysis

To be able to accurately identify the activities of a routing node, their behaviour must be analysed. As explained previously, the miners possess the knowledge of the expected neighbouring hops for a specific route. By extracting and analysing the overheard transmissions, the miners are capable of detecting different threats. If a threat is detected, the transmission is labelled as malicious, thus impacting the reputation of the transmitting node.

#### 4.1.1 Node Reputation

The reputation of a node represents their trustworthiness in the network. As a result, it is calculated for the list of *good* and *bad* actions. These binary actions, are determined from the behavioural analysis, differentiating expected and non modified transmissions as *good* and anything else as *bad*. As a result, the mode actions there are in either category, the more the reputation will tend towards the corresponding value. In short,

the greater the amount of *good* actions, the higher the reputation, and vice-versa.

$$S_{good_n} = \sum_{i=1}^{W_n} good\ actions_{n_i} \quad (1)$$

$$S_{bad_n} = \sum_{i=1}^{W_n} bad\ actions_{n_i} \quad (2)$$

We define  $S_{good_n}$  and  $S_{bad_n}$  as the sum of *good* and *bad* actions respectively for node  $n$ , as computed in (1) and (2). We also define  $W_n$  as the size of the action window time frame, corresponding to the number of previous actions taken into account during the calculation. By increasing or decreasing this value, we can influence the precision of the calculation. This allows the miner to take into account only the actions of the last exchange, or all actions during the last  $W_n$  exchanges. With this, we can open up the nodes history, allowing the network to have a longer or shorter memory when it comes to nodes actions.

Armed with the quantity of *good* and *bad* actions during the time frame, we can calculate the nodes reputation. The reputation  $R_n \in [0, 1]$ , is expressed as a sigmoid function, where the exponent  $\delta_n \in [-1, 1]$  represents the weighted value of the relation between  $S_{good_n}$  and  $S_{bad_n}$ , calculated in (1) and (2).

$$R_n = \frac{1}{1 + e^{-\delta_n}} \quad (3)$$

$$\delta_n = \beta \times \frac{S_{good_n} - \alpha \times S_{bad_n}}{S_{good_n} + \alpha \times S_{bad_n}} \quad (4)$$

We define two variables for the calculation of  $\delta_n$ , the first of which is  $\beta = 8$  which corresponds to the sensitivity factor influencing the sigmoid function, as presented in [25]. The second,  $\alpha$ , is the weight of malicious actions upon the reputation. By changing this value, we can increase or decrease the impact of *bad* actions in relation to *good* actions. As a result, it is possible to increase or decrease the consequences of misbehaving nodes, making the network more or less tolerant. Figure 4, presents the evolution of a node's reputation based upon the value of  $\alpha$ . As we can see, the higher the value, the higher the impact on the overall reputation and the more unforgiving the network becomes. **This illustrates the impact of a node becoming malicious, where the more malicious actions are performed, the more the reputation will decrease. Furthermore, thanks to  $\alpha$ , we can specify the impact of these actions, allowing the reputation to respond quickly to variations and changes in the nodes behaviour.**

#### 4.1.2 Reputation Decay

As presented in Section 3.3, certain threats can pertain to malicious access or corruption of legitimate nodes.



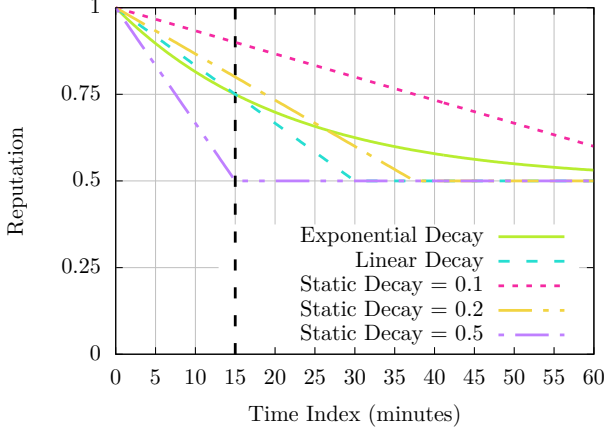


Fig. 5 – Reputation Decay

553 Once their activities detected, a bad reputation is inevitable, resulting in the node no longer being used during routing. However, once a node has been isolated from the network, the attacker no longer has any use for it. In many cases, the malicious party will move to a better position to continue their attack, leaving the compromised node alone. Since a nodes reputation only evolves when they participate in routing activities, there is no way to re-integrate this node back into the network. To counter this issue and permit reintegration, we propose a new metric called *Reputation Decay*. Overtime when the node does not participate in routing activities, their reputation will slowly decay towards the neutral value of 0.5. This will increase the chances of a node being used once more for routing, allowing it to clear its name. However, this decay does not change the number of *good* and *bad* actions performed by the node, but serving simply as a means for granting it a second chance. It also allows nodes which possess a very good reputation and have not been used for a while, to decrease back towards the neutral 0.5 as well.

574 We define  $Rd_{n_t}$  as the reputation decay of node  $n$  at time  $t$ ,  $\lambda$  as the decay factor,  $t_{\frac{1}{2}R}$  as the half-life of the reputation and  $R_{n_t}$  as the resulting decayed reputation of node  $n$  at time  $t$ .

$$Rd_{n_t} = (t - t_{R_n}) \times \left(\frac{\lambda}{t_{\frac{1}{2}R}}\right) \quad (5)$$

$$R_{n_t} = R_n - Rd_{n_t} \quad (6)$$

578 By varying the value or the function of  $\lambda$ , we can influence the rate of decay, allowing the convergence towards 0.5 to occur sooner or later. Figure 5 shows the evolution of the decay rate from a base value of 1 towards the neutral 0.5, with a half-life of  $t_{\frac{1}{2}R} = 15 \text{ min}$  with various decay methods. For the rest of our analysis, we kept a half-life of 15 minutes and decided on a linear decay function with a decay value of  $\lambda = 0.25$ . As a result, a nodes reputation will return to neutral from

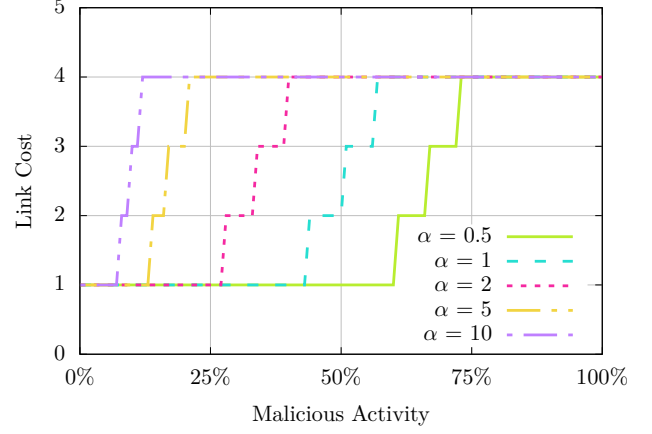


Fig. 6 – Link Cost Evolution

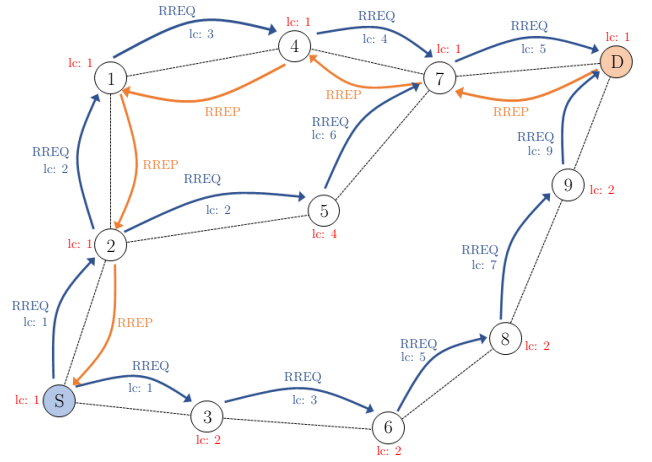


Fig. 7 – AODV-Miner discovery process

588 either extreme of 1 or 0, after  $2 \times t_{\frac{1}{2}R}$ , corresponding here to 30 minutes.

## 590 4.2 Protocol Integration

591 With the ability to calculate the reputation of a node based upon its actions, it is necessary for it to be integrated into the AODV routing protocol. Being a reactive routing protocol, discovery is performed only when needed, meaning it can take advantage of the existing reputations. However, for the reputation to influence the choice of route, modifications to the existing AODV packet structure is necessary. Furthermore, with new additions to the discovery process, we can provide the necessary information for the miners to accurately and reliably perform their activities.

### 602 4.2.1 Link Cost

603 As explained previously, AODV determines the best route based on the number of hops thanks to the RREQ *hop-count* field, thus discarding longer routes and keeping only the most direct possible. However, in our context it is necessary to exchange the length of the route

and instead use its reliability factor. As performed in [25], we replace the *hop-count* field with a metric called *link-cost*. This allows the nodes to calculate the "cost" of using a certain neighbour, based upon that neighbour's current reputation. With this metric, we can differentiate and separate *good* nodes from *bad* ones by simply increasing the *link-cost* the lower the nodes reputation. Upon receiving an RREQ or RREP packet, the node calculates the sender's reputation, along with its potential decay. It then determines the *link-cost* corresponding to the final reputation, increasing the value of the *link-cost* field accordingly. By updating this field, no modifications are brought to the overall functionality of AODV, where the route with the lowest *hop-count* is selected, only here the value corresponds to the most reliable route. This allows the route to contain as few malicious nodes as possible, all the while facing a trade off of longer routes for increased route integrity.

$$C_n = \lfloor (1 - R_{n_t}) \times (C_{max} - (C_{min} - 1)) + C_{min} \rfloor \quad (7)$$

We define  $C_n$  as the *link-cost* between the current node and the node  $n$ , with  $R_{n_t}$  corresponding to the reputation of said node at time  $t$ . As  $R_{n_t}$  is normalised between 0 and 1, it is necessary to expand the adapt the resulting *link-cost*. We, therefore, define  $C_{min}$  and  $C_{max}$  as the minimum and maximum values possible for this cost. By setting  $C_{min} = 1$ , we assure that even with an excellent reputation, the *link-cost* field will always be incremented by one, thus removing the risk of infinite cost calculation loops. Finally, the resulting value is then decreased to the nearest natural number, less than or equal to the calculated value. Since AODV's *hop-count* field is only one byte in width, the value of the *link-cost* must be adjusted accordingly. With an overall maximum potential network cost of 255, we can calculate the maximum possible *link-cost*  $C_{max}$  based upon the number of potential nodes in the network.

$$C_{max} = \frac{255}{L_{max}} - 1 + C_{min} \quad (8)$$

With  $L_{max}$  corresponding to the maximum possible route length (i.e., number of nodes traversed), we can adjust the precision of the *link-cost* metric. For example, with  $L_{max} = 32$ , we could accommodate a maximum value of 8, whereas  $L_{max} = 64$  would only allow for 4 individual values. By proposing an adaptable scaling function, we can increase or decrease the precision of the *link-cost* metric in relation to the number of nodes. Also, by tying this value into AODV itself with the NET\_DIAMETER parameter, we can provide a seamless integration between the two. However, although AODV allows each node to customise the value of NET\_DIAMETER accordingly, our method needs the value of  $L_{max}$  to remain constant throughout the network, or risk a route being dropped for cost overflow. For the rest of our analysis, we decided on  $L_{max} = 64$ ,

which corresponds to the maximum TTL value widely used in networking, resulting in our routes containing at most 64 nodes. Figure 6 shows the calculated *link-cost* values for the different reputational values previously presented in Figure 4. Figure 7 illustrates the discovery process of *AODV-Miner*. By comparing with Figure 3, we can see the differences where node 5 exhibits malicious tendencies. Since AODV selects the shortest route possible in terms of hops, the RREPs will always transit via node 5 for a maximum of 4 hops compared to 5 hops via the other routes, putting the data at the mercy of our bad guy. By adding the *link-cost* into the equation, we can influence the route selection process, thus avoiding the malicious entity. This is visible in Figure 7 where each node possesses a *link-cost* ( $lc$ ). Since node 5 is malicious, we assume it has received a low reputation, resulting in a high *link-cost* of 4. This high value causes an increase of the total route cost, bringing it up to 6 from the source node to node 7. In this case, the top route is the winner, with a total cost of 5 from source to destination, making it the most efficient and trustworthy route.

Thanks to the quick reactions of the reputation metric, the *link-cost* can also adapt in a timely manner, immediately influencing the selection of the next route. Indeed, since the validation process takes place after a route has expired, the updated reputations only enter into play the next time the node is needed. This means that as long as the route remains active, the malicious node can impact the routing activities, however, the more actions it performed the more severe the consequences. It is also important to note that by artificially lengthening the route used dependant on each node's reputation, we do not explicitly isolate nodes from routing. Our method simply encourages the protocol to seek another route towards the destination avoiding the malicious entities as much as possible. However, in some cases, no alternative routes exist, and the malicious node is utilised, thus impacting the network security. Further study into these two points can help reinforce the network security, and is also one of our current directions.

#### 4.2.2 RREP 2-Hop

So that the miners can achieve their goals of route validation, they must know to whom the packets must be sent. By overhearing passing RREPs, miners can construct their view of the expected route towards the destination, but also back towards the source, adding the hops to the corresponding *RVTs*. Unfortunately, although overhearing RREP packets allows the miners to construct parts of the route, they are missing some elements of the big picture. Indeed, since RREPs only serve to inform node  $n - 1$  to transmit towards  $n$ , the miners are only aware of the expected exchange between these nodes. This information is insufficient, as in many cases node  $n$  is not the destination and will, therefore, need to transmit its data on-wards. However, it its

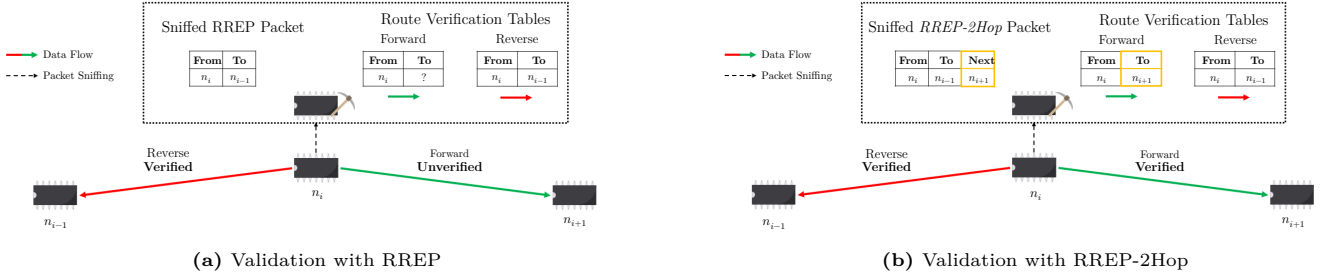


Fig. 8 – Illustration of the need for RREP-2Hop

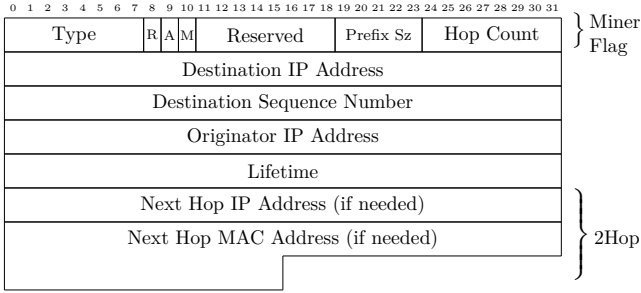


Fig. 9 – RREP-2Hop packet structure

748 activities, the miners themselves must be selected and  
 749 differentiated from the routes routing nodes.

### 750 4.3.1 Miner Selection

715 current state, the miners are incapable of prediction to  
 716 whom this packet will go, meaning they are incapable  
 717 of validating the behaviour. This problem is illustrated  
 718 in Figure 8a, where we can see that our miner can only  
 719 overhear the communications coming from node  $n_i$ . As  
 720 a result, the RREP packet only informs on the reverse  
 721 route back to the source through  $n_{i-1}$ , and not the for-  
 722 wards route towards  $n_{i+1}$ .

723 To remedy this, we propose an amelioration to the  
 724 RREP packet format, allowing us to include the infor-  
 725 mation for the next hop. This new packet format, called  
 726 *RREP-2Hop* is presented in Figure 9. We can see the  
 727 addition of the new *2Hop* section, containing the IP and  
 728 MAC addresses of the nodes next hop. By providing the  
 729 MAC addresses of the next hop, the miners can com-  
 730 plete their *RVTs* and achieve their goals. By also taking  
 731 advantage and incorporating the corresponding IP ad-  
 732 dress, each node can also construct *2Hop Routes* in their  
 733 routing tables if they so desire. As we can see in Fig-  
 734 ure 8b, this new addition allows the miner to determine  
 735 the forwards route from  $n_i$  towards  $n_i + 1$ , allowing full  
 736 validation to take place. So as to allow our solution to  
 737 be adapted to existing AODV routing, we also incorpo-  
 738 rated a *Miner Flag* into the packet header. This allows  
 739 the system to differentiate and identify the RREP pack-  
 740 ets, allowing the choice to function with or without our  
 741 addition.

## 742 4.3 Behavioural Validation

743 To be able to determine the reputation and influence  
 744 the route selection, there are a few steps which need  
 745 to be performed. In this section, we present the min-  
 746 ers themselves, taking a look at how they perform their  
 747 different roles. However, before they can perform their

751 As stated previously, we provide the ability for all net-  
 752 work nodes to determine their own role per route. How-  
 753 ever, nodes cannot take on both roles of miner and  
 754 router at the same time for the same route as this could  
 755 result in a conflict of interest. This is because a routing  
 756 node cannot objectively analyse their own behaviour,  
 757 or that of the node which has transmitted the infor-  
 758 mation to them. Furthermore, by separating the roles  
 759 between multiple nodes, we reduce the probability that  
 760 the potential malicious node could also impact the val-  
 761 idation phase, subsequently corrupting the reputation  
 762 table. The selection process is performed during the  
 763 AODV route phase, allowing all miners to be identified  
 764 and possess all routing information needed to perform  
 765 route validation once the route becomes active. As pre-  
 766 sented previously with the definition of *RREP-2Hop*,  
 767 miners use RREP packets to gather the necessary hop  
 768 information. Upon receiving an RREP packet, the node  
 769 first analyses the destination address. If the RREP is  
 770 destined for them, then they identify themselves as part  
 771 of the route, processing the packet information as nor-  
 772 mal and constructing the different routes in its routing  
 773 table, using the *2Hop* address if desired. On the other  
 774 hand, if the RREP is not destined for them, then the  
 775 node enters an internal validation phase. They first  
 776 check if they are not already a router for the route,  
 777 in which case the RREP is immediately dropped with-  
 778 out further analysis. If not, then the destination link-  
 779 layer address is extracted from the packet header and  
 780 the *2Hop MAC* address from the *RREP-2Hop* payload.  
 781 Both addresses are then used to construct the reverse  
 782 and forwards *Routing Validation Table* entries for the  
 783 node which transmitted the RREP.

### 4.3.2 Routing Analysis & Validation

---

**Algorithm 1** Miner route validation run at miner  $m$  upon reception of  $\text{pkt}(llsrc, lldst, src, dst)$

---

```

1: if New packet detected then
2:   Create new  $buf_{pkt}$  entry with  $hash_{pkt}$ 
3:   set  $buf_{pkt}$  as valid
4: else Previous malicious activity detected ; Exit ;
5: end if
6:  $RTE =$  Get route entry for  $[src \rightarrow dst]$ 
7:  $RVT =$  get validation tables from  $RTE$  for  $llsrc$ 
8: if  $RTE$  &  $RVT$  both empty then
9:    $\triangleright$  No route validation table, Malicious
    behaviour
10:  Increment  $bad_{llsrc}$ ; Set  $buf_{pkt}$  as invalid
11: else
12:   $nextHop_{pkt} =$  get the next hop from  $RVT$ 
13:  if  $nextHop_{pkt} \neq lldst$  then  $\triangleright$   $lldst$  is not the
    next expected hop - Malicious behaviour
14:    Increment  $bad_{llsrc}$ ; Set  $buf_{pkt}$  as invalid
15:  else  $\triangleright$  Valid behaviour
16:    Increment  $good_{llsrc}$ 
17:  end if
18: end if

```

---

Once the route discovery has completed, the route can begin transmitting data. The selected miners then begin to "mine their route" by observing and analysing all data traffic originating from neighbour nodes. To accurately analyse the data exchange, the miners utilise their forwards and reverse  $RVTs$ . Each table contains the ordered list of expected hops in transmission range of the miner. These tables, visible in Figure 8b, allow the miner to verify all packets follow the same hop ordering. This allows us to detect any redirecting attacks where the destination is not conform to the table entry, or packet destruction where the hop list is not traversed completely. However, it is important to note that as presented previously, we are only able to validate data originating from the routes source towards the routes destination and not intermediate exchanges taking advantage of the routing table entries.

For each packet received, the miners process the data to determine its authenticity, as presented in Algorithm 1. During the analysis, the miners verify the packets destination as well as its integrity, allowing it to identify if the transmitting node has malicious tendencies. The verification phase stays active as long as the route itself is in use. Upon expiration, the miners first check their passing packet buffer, identifying packets currently in transit. If the buffer contains data, then the last associated node is considered to have not transmitted the data onwards and, therefore, increasing the number of  $bad$  actions. Once all actions have been totted up, the miners all drop their  $RVTs$  for the route and enter their final phase of block confirmation.

### 4.3.3 Block Confirmation & Dissemination

To allow consensus based confirmation, the miners must first create their own block containing the number of  $good$  and  $bad$  actions for each and all routing nodes which it has mined. The block is then broadcast up to a maximum distance of 2 hops, allowing it to reach only nodes in proximity which are potentially miners for the route. Upon receipt of such a block, the miner proceeds with two calculations. Firstly, they analyse the number of  $good$  and  $bad$  actions contained in the block, calculating the number differences with their own block. If this value is too high, the block is considered to be invalid and the miner transmits their own block as a response. However, if no differences are detected, the miner then performs an efficiency evaluation to determine if the block is more efficient than its own. This is achieved by calculating the percentage of nodes in common in the received block,  $P_B$  versus the miners own block  $P_M$ , with  $B$  corresponding to the list of nodes in the received block and  $M$  those in the mined block.

$$P_B = \frac{|M \cup B|}{|B|} \quad (9) \quad P_M = \frac{|M \cup B|}{|M|} \quad (10)$$

The miner only transmits its own block in this case if it is deemed more efficient, in other words if  $P_B < P_M$  where  $P_M$  is considered to possess more nodes overall and a higher percentage of shared nodes. Since miners can corrupt the results of this exchange, the process relies on a consensus where responses from miners overrule previously transmitted blocks. To stop validation loops, miners can only transmit their own block once, allowing the last block to correspond to the majority. If the received block is considered more efficient, the miner then identifies all common nodes as "overridden", meaning they have been confirmed by another more efficient block. This allows miners to detect if they possess a node which has not been validated by other miners, allowing them to re-transmit their own block containing only the missing nodes for validation. As a result, the last blocks to be received and not overruled are considered both valid and more efficient since they possess the largest quantities of nodes possible, without overlapping with other blocks. The only task left is purely blockchain related, where the miners hash the contents of their blocks, inserting the hash of the last received blockchain block, then inserting it into the blockchain by broadcasting it throughout the network. This allows all network nodes to extract the list of  $good$  and  $bad$  actions for each node, knowing that the block is valid.

## 5. IMPLEMENTATION & SIMULATION

As stated in the previous section, each node contains two  $RVTs$ , storing the ordered list of forward hops, towards the destination, and reverse hops, back towards the source. The nodes also possess a *Packet Buffer*, containing a list of packet hashes as calculated by miners

**Table 2** – Simulation Parameters

Parameter	Setting
Area	<i>Varying</i>
Number of nodes ( $N$ )	<i>Varying</i>
Malicious Activity	<i>Varying</i>
Malicious Weight ( $\alpha$ )	<i>Varying</i>
Distribution	Random uniform
Transmission Range	50m
Max Length ( $L_{max}$ )	64
Window Size ( $W_n$ )	5
Reputation Decay	Linear
Initial Reputation	0.5
Number of Simulations	100
Simulation Duration	15 min.
Messages per Transmission	5
Transmission Interval	1 min.

870 along with their next expected hop. This allows the  
 871 miners to detect modifications to the packets, as well as  
 872 serving as a reminder as to which hop is next expected  
 873 for this packet. The nodes also own a *Node Reputation*  
 874 *Table*, which contains the list of *good* and *bad* actions  
 875 for each node as extracted from the blockchain. These  
 876 actions are input into Eq. (1) - (4) to calculate the  
 877 nodes current reputation. The number of actions stored  
 878 in this table is influenced by the size of the Reputation  
 879 Window  $W_n$  as shown in Eq. (1) and (2).

880 Since our implementation revolves around a light-weight  
 881 version of the blockchain, its functionalities are emu-  
 882 lated. This means that the chain itself is not stored on  
 883 the nodes, but only disseminated and analysed by the  
 884 network. By not storing the received blocks, we save  
 885 on node memory, which we can put to other uses such  
 886 as reputation values or the behavioural validation itself.  
 887 Upon receipt of a new block from the blockchain, each  
 888 node calculates the blocks hash, allowing them to verify  
 889 the integrity of each subsequent block. When a route  
 890 discovery is triggered, each node accesses the *Node Rep-*  
 891 *utation Table* entry for the RREQ or *RREP-2Hop* sender  
 892 and calculates the corresponding reputation. The node  
 893 then determines the time since the last use of the corre-  
 894 sponding node and applies the reputation decay function  
 895 (5) as needed. The resulting reputation is then fed to  
 896 the *link-cost* function (7), providing the corresponding  
 897 cost for using said node. By comparing the *link-cost*  
 898 field of received RREQs, we can make sure to propagate  
 899 only the lowest values onwards, thus eliminating poten-  
 900 tially malicious routes as the discovery process advances.  
 901 However, with the addition of this metric, it is possible  
 902 that on occasion the calculated *link-cost* is lower than  
 903 the previous. This is due to a field overflow after a sig-  
 904 nificant number of hops and as a result the correspond-  
 905 ing RREQ can be discarded as it can be considered too  
 906 malicious. By only propagating RREQs with low *link-*  
 907 *cost* values, we can assure that the destination only re-  
 908 ceives the most reliable routes possible. Furthermore,  
 909 contrary to the approach in [25], here the destination  
 910 node does not wait for the most reliable route before  
 911 responding towards the source, thus providing all possi-  
 912 ble routes for the source source itself to choose the  
 913 best possible. In our implementation, upon receipt of  
 914 an RREQ, the destination waits for a small period of  
 915 time before transmitting the RREP back towards the  
 916 source. If any subsequent better RREQs are received,  
 917 the destination waits once more before transmitting the  
 918 corresponding RREP. Once the RREPs return to the  
 919 source node, the node also waits for a slightly longer  
 920 time period for potential other RREPs to arrive, before  
 921 transmitting along the most efficient route. Any sub-  
 922 sequent RREPs update the route as transmissions are  
 923 occurring, without impacting network operations.

924 **5.1 Simulation Settings**

925 For our analysis, *AODV-Miner* was implemented us-  
 926 ing the Contiki-NG [27] operating system and subse-  
 927 quently simulated using their Cooja simulator. Table  
 928 2 presents the general parameters used throughout our  
 929 simulations. The simulated Cooja nodes possess a wire-  
 930 less interface using the IPv6 net-stack running a 6LoW-  
 931 PAN network layer and a non-beacon-enabled always  
 932 on CSMA radio. Although CSMA allows to reduce  
 933 the probability of collisions, it does not remove it en-  
 934 tirely, especially concerning nodes which are list listen-  
 935 ing and overhearing transmissions. Since this problem  
 936 can impact AODV and data transmissions as much as  
 937 our Miners, we rely on the underlying network proto-  
 938 cols as well as our multi-miner validation approach to  
 939 reduce the possible consequences. Similarly, the always-  
 940 on radio permits the nodes to remain in the necessary  
 941 active state, needed for both AODV and the valida-  
 942 tion miners. Their on-board systems are initialised us-  
 943 ing individually generated seeds, allowing each node to  
 944 possess a different random generator, all the while pro-  
 945 viding precise calibration of parameters. The different  
 946 malicious nodes are distributed throughout the network  
 947 using a random distribution function, only impacting  
 948 data traffic whilst leaving AODV related communica-  
 949 tions unscathed for the analysis of the routing protocol.  
 950 For ease of analysis, we simulate the network against  
 951 two types of threats: Black-holes and Grey-holes. As  
 952 previously explored in section 3.3, although we are ca-  
 953 pable of detecting many threats, our detection system  
 954 revolves around the same methods: deviation from ex-  
 955 pected activities. As a result, Black-holes allow us to  
 956 simulate complete data destruction, whereas Grey-holes  
 957 allow us to vary the probability of destruction, allowing  
 958 more or less packets to transition through the network.  
 959 This means that even with only two attacks, we can  
 960 hypothesise that the results would be similar with the  
 961 other attacks, since their consequences and subsequent  
 962 detection would be the same.  
 963 During our analysis, we used two network topologies,



pitching *AODV-Miner* against its older brother AODV. The first contains 100 nodes in an area of 300m×300m whereas the second contains only 30 nodes, in a smaller area of 150m×150m. This allows us to test our system in two different situations, where the possible route length significantly increases, as well as the number of potential malicious nodes. In both situations, we transmit 5 random data packages every minute, allowing the network time to perform route discovery, packet routing and blockchain dissemination

## 6. RESULTS

Our simulations allowed us to evaluate and analyse the overall functionalities and efficiency of our approach. By varying the topological layout, we could verify that our methodology would be able to handle different sized networks. We start our analysis by evaluating the functionality of the Reputation metric, before taking a gander at the routes themselves. Finally, we analyse how our method holds up against varying degrees of malicious activities, simulating both Black-hole and Grey-hole attacks.

### 6.1 Reputation Analysis

Figure 10a shows the evolution of a nodes reputation over time with varying degrees of malicious intentions. By using  $\alpha = 2$ , we double the weight of malicious activities in relation to *good* actions. This can be observe with 25% malicious activities, where the resulting reputation resides around the neutral 0.5 mark. As a result, the greater the malicious activities, the lower the reputation, with 75% and 100% practically indistinguishable. Furthermore, we can also notice that the reputation is established immediately after the first route expires, round about the 1 minute mark. We can also see that, although the values fluctuate, they remain in the same overall area throughout the simulation.

By varying the value of  $\alpha$ , presented in Figure 10b, we can observe its impact on the reputation. In this figure, we analyse the evolution of the reputation for 25% malicious activities. We can verify this by comparing the results of  $\alpha = 2$  with the 25% malicious activities from Figure 10. Immediately, we can confirm our hypothesis of the impact of  $\alpha$  as we can clearly observe that the greater the value, the lower the reputation. This is of-course also true in the opposite direction, with the corresponding results for lower values of  $\alpha$  finding themselves closer to the perfect reputation of 1. In essence, by acting on this variable we can actively influence the weight of all *bad* behaviour, instantly punishing a node for misbehaving, granting them forgiveness more swiftly.

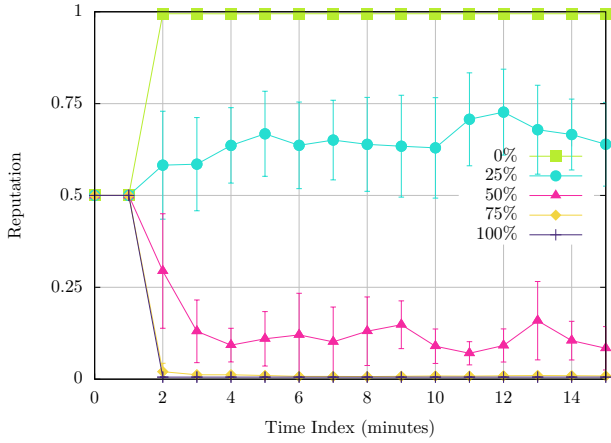
### 6.2 Route Analysis

By analysing the routing efficiency, we can determine if *AODV-Miner* can reach its goal of isolating as many malicious nodes as possible from the determined routes.

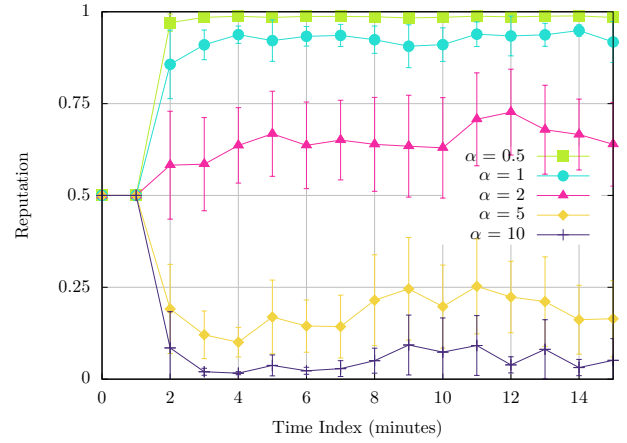
Figures 11 and 12 compare these results against the standard AODV protocol in a network of 30 and 100 nodes respectively. Firstly, we analyse the number of packets dropped ( $|Packets\ Sent| - |Packets\ Received|$ ), visible in Figures 11a and 12a. We can immediately see that there is a reduction in lost packets, with an overall increase in efficiency of 48% with 30 nodes, and 38% with 100 for a network with 10% malicious activities. Furthermore, these results are corroborated in Figures 11b and 12b, where we can see that *AODV-Miner* possesses a higher overall throughput than AODV for both typologies, whatever the percentage of malicious nodes. It is to be noted that not all drops can be prevented, since the reputation is computed on the fly, leaving time for malicious entities to cause mayhem. It is also possible that in some cases, traversing a node with a *link-cost* of 4, is still considered more efficient than five nodes with a cost of 1. However, there is a consequence to this increase in efficiency. Indeed, Figures 11c and 12c show a trade-off, where we may indeed have better efficiency, but at the cost of longer routes. In our network with only a 30 nodes this difference is minimal, however, by increasing the number of nodes we can see an increase in the number of hops. This is not the only cost of our implementation. Another is linked to the activities of the miners, since block validation and distribution increases the number of packets exchanged throughout the network. Our final analysis in Figures 11d and 12d demonstrates this increase, with both 30 and 100 node typologies possessing a significantly higher overhead, ending up around the 80% mark. Although this may seem high, it is a necessary evil to ensure that a higher percentage of data reaches its destination unscathed.

Thanks to these results, we can confirm that our method allows us to isolate and avoid malicious nodes, increasing the probability of data reaching its destination. Figure 13 illustrates this process in networks of 30 and 100 nodes, both with 25% exhibiting black hole characteristics, represented with thick outlines. By superimposing the computed reputation for all nodes, as well as the most used route by both AODV and *AODV-Miner*, we can visualise this increase in performance. In both networks, we can see that AODV attempts to take the shortest most direct route possible per its programming, which unfortunately results in encountering a malicious node. In contrast, *AODV-Miner* is capable of discovering a free trustworthy route between the source and destination, avoiding malicious entities. As we can see by the colour gradient, nodes have been attributed both high and low reputations, dependant on their activities during routing. By analysing Figure 13a, we can see that a total of eight nodes have been attributed reputations higher than the neutral 0.5, whereas three others have received low reputations. As stated previously, it is a necessary evil to allow messages to be lost to allow for the malicious activities to be detected and the reputation computed. This means that in this scenario,



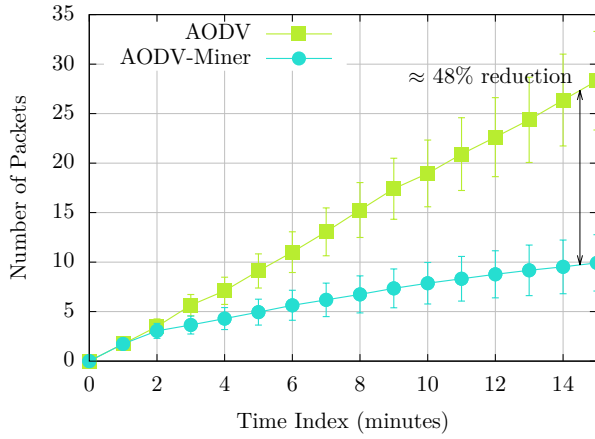


(a) Reputation overtime with varying degrees of malicious activities

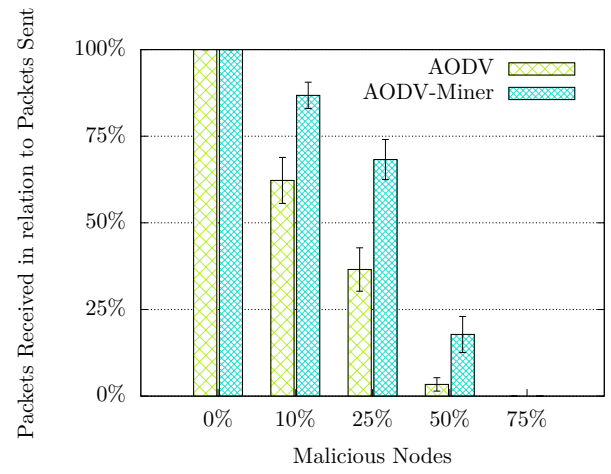


(b) Impact of  $\alpha$  with 25% malicious activity

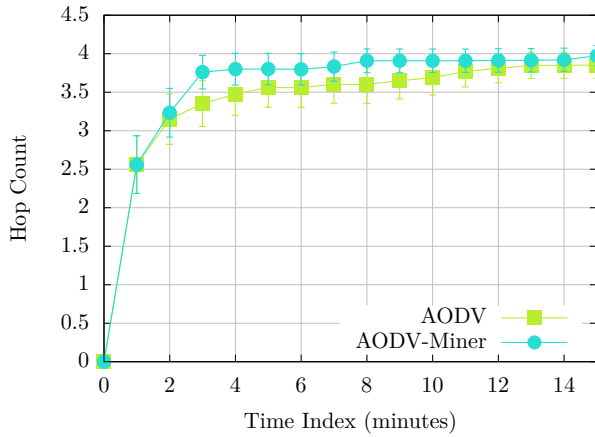
Fig. 10 – Evolution of node reputation



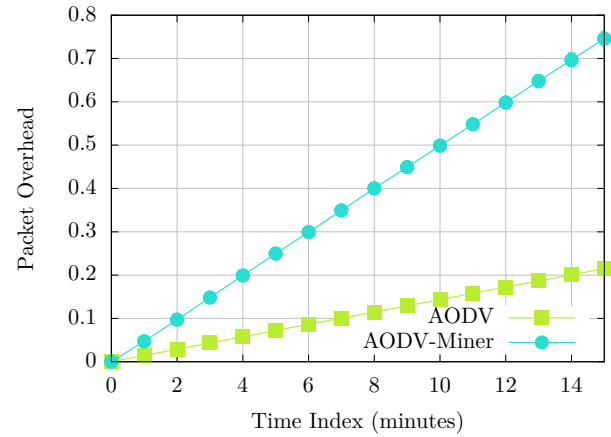
(a) Packets dropped with 10% malicious nodes



(b) Throughput with varying percentage of malicious nodes



(c) Average route length with 10% malicious nodes

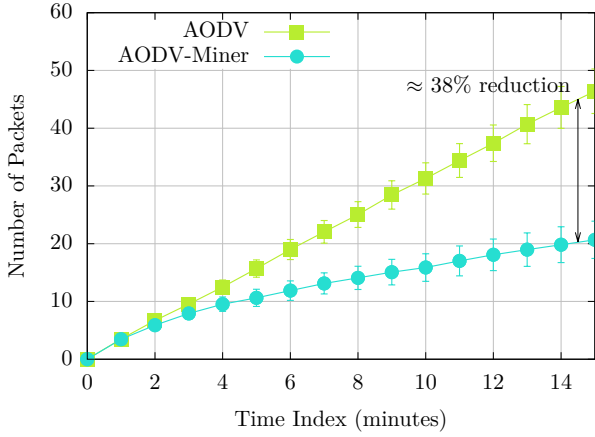


(d) Normalised overhead with 10% malicious nodes

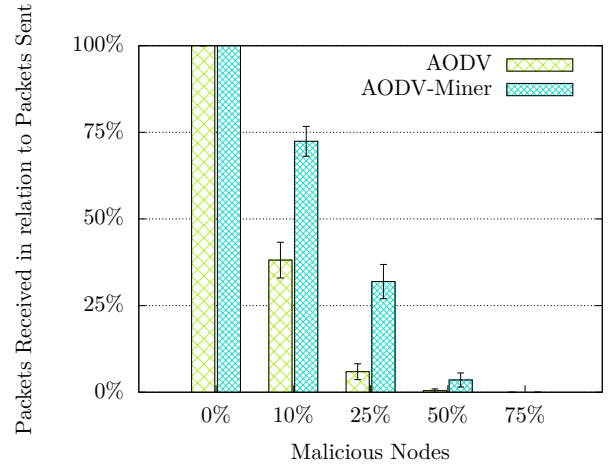
Fig. 11 – Routing efficiency between *AODV-Miner* and AODV with a network of 30 nodes

1075 three determined routes ended with all their data being  
 1076 lost before *AODV-Miner* was able to adapt. Of course,  
 1077 this effect is amplified the larger the network, and con-  
 1078 sequently the more malicious nodes are present. In con-  
 1079 trast, Figure 13b presents a significant sixteen nodes  
 1080 possessing a high reputation and seven with low values,

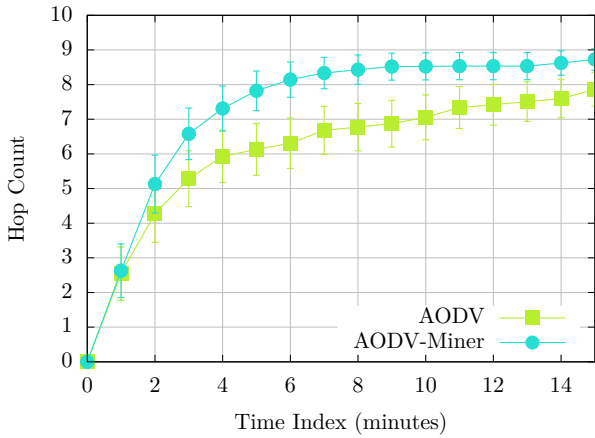
1081 four more than the smaller network. We can also see  
 1082 a cluster of four malicious nodes in the centre of the  
 1083 network separating the source from the destination, all  
 1084 of which have been detected and subsequently avoided.  
 1085 One final note is that, as is the case with AODV, the  
 1086 route selected may on occasion change due to various



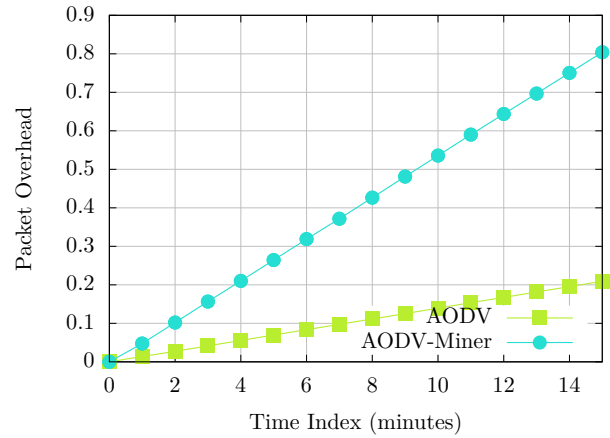
(a) Packets dropped with 10% malicious nodes



(b) Throughput with varying percentage of malicious nodes

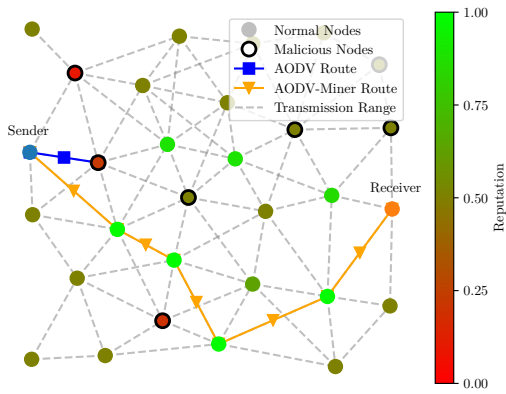


(c) Average route length with 10% malicious nodes

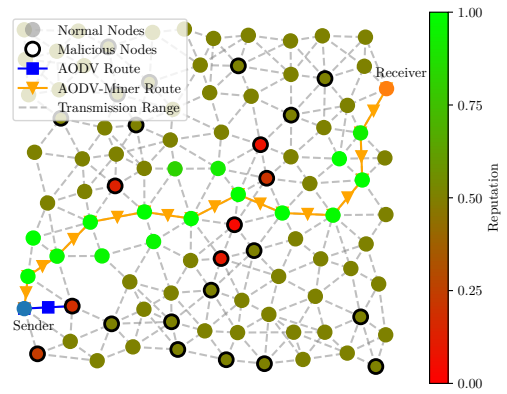


(d) Normalised overhead with 10% malicious nodes

**Fig. 12** – Routing efficiency between *AODV-Miner* and AODV with a network of 100 nodes



(a) 30 Nodes

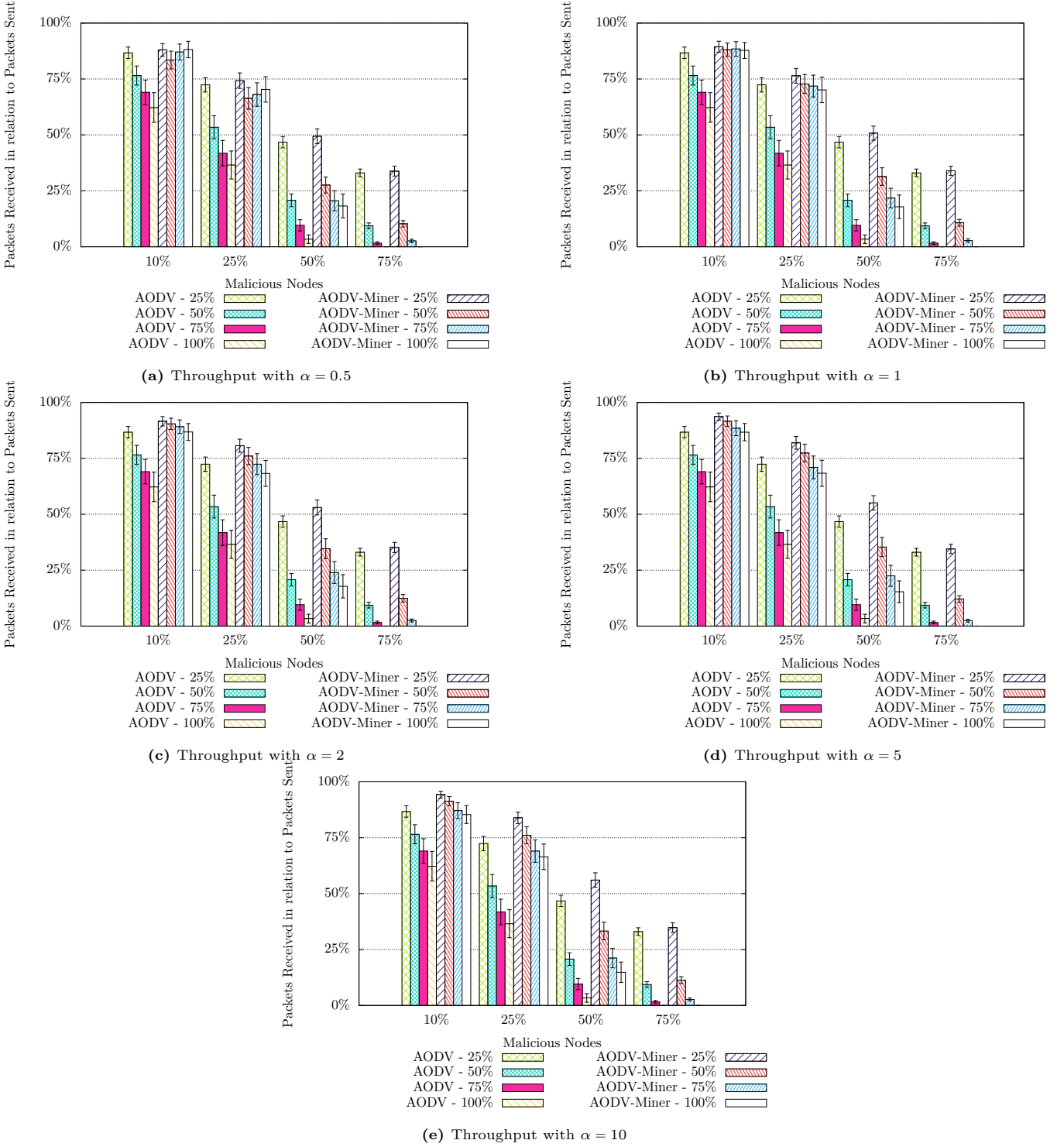


(b) 100 Nodes

**Fig. 13** – Visualisation of route reputation after 15 mins. with 25% malicious nodes

1087 reasons. We can see this with the fact that in both figures, there are nodes which have good reputations, and  
 1088 yet are not part of the most used route. This is possible where some RREQ messages are lost due to collisions,  
 1089 forcing the network to select an alternate route, or sim-  
 1090  
 1091

1092 ply arriving too late to change the selected route.



**Fig. 14** – Throughput comparison between *AODV-Miner* and *AODV* with a network of 30 nodes subjected to Grey-hole attacks

### 1093 6.3 Threat Adaptation

1094 The final aspect of our analysis concerns the ability of  
 1095 our system to adapt to different threat types. In this  
 1096 context, we pitch the *AODV-Miner* against varying de-  
 1097 grees of packet drops in a Grey-hole attack. Some Grey-  
 1098 hole attacks use packet selection to decide what data to  
 1099 destroy and what to let pass, also called Selective For-  
 1100 warding (*RTE06* in Table 1). In our case, we use inter-

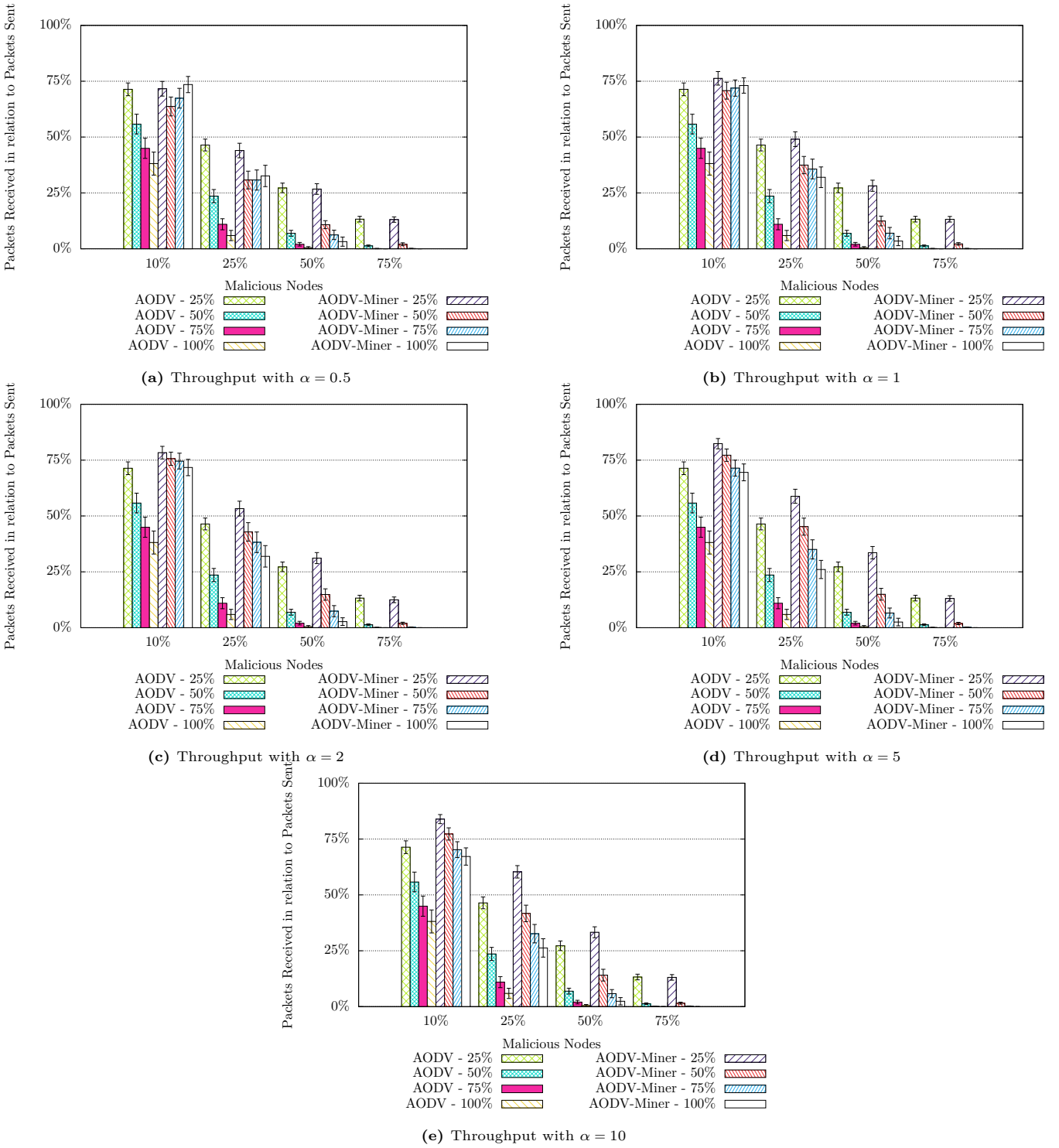
1101 nal probability functions to decide which packet to drop  
 1102 on each malicious node, each initialised with a differ-  
 1103 ent seed allowing different values of probability between  
 1104 them. Figure 14 shows an analysis of these activities for  
 1105 a network of 30 nodes and Figure 15 for a network of  
 1106 100 nodes.

1107 If we turn our attention to the analysis of the 30 node  
 1108 topologies, Figure 14 shows the different throughput lev-

els of AODV against *AODV-Miner* with varying numbers of malicious nodes, based on the Grey-hole probability in use. We also extend this analysis by comparing the results with different values of  $\alpha$ , thus showing its impact on the determination of the reputation and consequently the routing efficiency. We can see that in general, *AODV-Miner* performs well, keeping an overall throughput higher than the corresponding values of AODV. Naturally, the more nodes turn to the dark side, the harder it is for *AODV-Miner* to determine a free route, which we can see with the very slight increase in network efficiency. Figure 14a shows the results where  $\alpha = 0.5$  corresponding to a very forgiving network where malicious activities have half the impact of *good* activities. This means that a node needs to perform twice the amount of *bad* activities than *good* to warrant a decrease in its reputation. This can be confirmed in the results with 10% and 25% malicious nodes possessing a malicious probability of 50%, where the throughput drops slightly since on average the nodes drop every other packet they receive. However, the moment the percentage of packets dropped is higher than a ratio of 1 : 1, the throughput rises once more, increasing even higher when all packets are being destroyed, reaching the same value as 25% malicious probability. In contrast, Figure 14b represents the case where *good* and *bad* activities possess the same weight,  $\alpha = 1$ . Here we can see that, for 10% malicious nodes, the throughput decreases only slightly the higher the malicious probability, simply due to the need for packets to be dropped before the reputation can be computed. The rest of the results decrease in throughput the higher the probability, all the while remaining slightly higher, or on par, with the results from Figure 14a. However, we can already identify a slight decrease in throughput when all packets are being dropped when compared to the previous figure. Figure 14c shows the first analysis where malicious activities possess a higher weight to *good*, with  $\alpha = 2$ . Comparing with  $\alpha = 0.5$ , here nodes need to perform twice the amount of *good* actions than *bad*, to stabilise their reputation once more. We can observe that, contrary to the previous analyses, there is a distinct decrease in reputation the higher the malicious probability, all the while remaining higher or equal to AODV. However, once more we can see that once more, the throughput for 100% of packets being dropped is lower than the previous values of  $\alpha$ . On the other hand, due to the increase in malicious weight, the initial throughput with only 25% of nodes exhibiting malicious tendencies is higher than before. As a result, the higher the value of  $\alpha$ , the more weight is accorded to *bad* actions and the faster *AODV-Miner* can react. That being said, there is a point where we reach peak efficiency, and the throughput cannot increase any higher and even starts to decrease slightly. This is the case of Figures 14d and 14e with  $\alpha = 5$  and  $\alpha = 10$  respectively. We can see that the values remain extremely similar, with in some cases  $\alpha = 10$  presenting slightly lower results than  $\alpha = 5$ , am-

plifying the previous observations for 100% malicious probability. However, as stated previously, when the vast majority of the network has become one with the enemy, there is only so much that can be done to try and combat the issue. This is the case with 75% of nodes exhibiting malicious habits, where the results for all five values of  $\alpha$  are extremely close with very low throughput levels.

By analysing the results from networks of 100 nodes, presented in Figure 15, we can analyse and strengthen our hypotheses. First off, we can see that in general the larger network size has resulted in general decrease in throughput level, due to the presence of more malicious nodes, as illustrated in Figure 13b. By beginning our analysis once more with  $\alpha = 0.5$  in Figure 15a, we can see the same pattern as previously, where the throughput drops between 25% and 50% malicious probability with 10% malicious nodes, only to rise once more, this time surpassing the throughput with 25% probability. This is also the case with 25% malicious nodes, although the increase is more subtle than the 30 node network in Figure 14a. However, here we can see that for 25% malicious probability, the corresponding throughput is lower than that of AODV for all percentages of malicious nodes. This reinforces our hypothesis that a low value of  $\alpha$  makes the network more forgiving, meaning it takes longer to detect and isolate malicious nodes, resulting in them being used more often, dropping more packets. Furthermore, whereas AODV on occasion will change routes depending on which RREP returns first and the potential RREQ losses, *AODV-Miner* would continue to use the node, since it would receive a good reputation, as previously demonstrated in Figure 10b. Increasing the value of  $\alpha$  consequently increases the overall throughput, although some parallels with the low value of  $\alpha$  can still be made. This is the case for  $\alpha = 1$  in Figure 15b, where a similar phenomena can be observed with 10% malicious nodes, all the while possessing a generally higher throughput. By looking at the values for 25% malicious probability, we can see that *AODV-Miner* is once again higher than AODV, reinforcing our previous hypothesis. Increasing the influence of bad actions, visible in Figures 15c, 15d and 15e demonstrates the advantages but also disadvantages of higher values. If we turn our attention to the results for 25% malicious probability, we can see the corresponding throughput increases the higher the value of  $\alpha$ , also visible in the other two figures. However, the higher the malicious probability, the more the associated throughput seems to struggle, decreasing slightly the more  $\alpha$  rises, similarly to the network of 30 nodes. This can be explained by the fact that malicious nodes are detected quicker, the higher the value of  $\alpha$ , explaining the increase in throughput for 25% malicious probability. This advantage allows *AODV-Miner* to determine new routes constantly once a malicious node has been detected. Furthermore, with a malicious probability of 25%, on average 1 packet in 4 is dropped, meaning it is possible that for every four packets transmitted along



**Fig. 15** – Throughput comparison between *AODV-Miner* and *AODV* with a network of 100 nodes subjected to Grey-hole attacks

1225 the same route, up to *four* malicious nodes can be detected, increasing the efficiency of *AODV-Miner*. As a  
 1226 consequence, the higher the malicious probability, the longer it takes to detect and circumnavigate malicious  
 1227 nodes. In the previous example, a malicious probability of 50% would produce a drop rate of 1 in 2, meaning that for four packets we could potentially detect  
 1228 only *three*, further decreasing to *two* for 75%, ending  
 1229  
 1230  
 1231  
 1232

1233 up with only a *single* node when Black-holes are used.  
 1234 This means that it would take *AODV-Miner* potentially four times longer to identify malicious nodes when they drop all packets when compared to Grey-holes dropping  
 1235 only 25%. This delay would consequently manifest in a lower throughput, as more malicious nodes need to be encountered directly to identify a route. Finally, as already examined previously, a network where 75% of all  
 1236  
 1237  
 1238  
 1239  
 1240

1241 nodes are beyond hope, even by changing the route con-  
1242 stantly in an effort to reach the destination, it is highly  
1243 unlikely to find a clear route. This is illustrated by the  
1244 fact that *AODV-Miner* results in a lower throughput for  
1245 25% malicious probability than AODV, where the sig-  
1246 nificant presence of malicious nodes simply hinders the  
1247 overall performance.

## 1248 7. DISCUSSION & FUTURE WORKS

1249 As we have presented previously, *AODV-Miner* has pro-  
1250 vided some overall good results. By providing an analy-  
1251 sis against various degrees of grey-holes, we have demon-  
1252 strated the adaptability of our protocol and its abil-  
1253 ity to cope with different attack scenarios. However,  
1254 we are aware that this analysis possesses some limita-  
1255 tions. Firstly, our system revolves around an emulated  
1256 lightweight blockchain, basically assimilated to a dis-  
1257 semination tool only. This was motivated to allow us  
1258 to concentrate further on the validation miners them-  
1259 selves and their activities related to behavioural analy-  
1260 sis. Blockchain storage is a well known challenge when  
1261 it comes the IoT, where many applications are turning  
1262 towards cloud computing strategies to store their data  
1263 [28]. This means that the blocks themselves in our case  
1264 are not stored on the nodes due to the inherent hardware  
1265 limitations of IoT devices. Instead, the information is  
1266 simply extracted and used to update the *Node Reputa-*  
1267 *tion Tables*, before forwarding the blocks onwards. Our  
1268 consensus-based validation metric also responds to the  
1269 specificities of IoT devices, reducing computation and  
1270 energy consumption inherent to the *PoW* concept. **Sec-**  
1271 **ondly, we only concern ourselves with malicious nodes**  
1272 **infiltrating the routing process. This choice was moti-**  
1273 **vated by our interest to demonstrate the efficiency of**  
1274 **our module against such attacks, without the risk of**  
1275 **further compromise by a malicious party. However, the**  
1276 **protection of the validation process itself is one of our**  
1277 **current interests and we are proposing an extension to**  
1278 **this module to secure the PoW against malicious miners.**

1279 Our consensus-based reputation system has been pro-  
1280 posed and evaluated using AODV, since it provides both  
1281 a simple and efficient platform for analysis. However,  
1282 our approach has been realised in such a way that it can  
1283 be applied to every platform respecting certain require-  
1284 ments. Indeed, many new protocols have emerged since  
1285 its elaboration, each with their own advantages and se-  
1286 curity integration's. Our next step would be to fully  
1287 analyse the advantages and functionality of our system  
1288 with these new protocols, by integrating our consensus-  
1289 based reputation system into the route decision mak-  
1290 ing process itself. By comparing these results with our  
1291 AODV baseline, we can evaluate in a more in-depth con-  
1292 text the efficiency and functionality of our system. Fur-  
1293 thermore, by deploying our system on real devices, we  
1294 can extrapolate real-life results from the idealistic sim-  
1295 ulation environment, **as well as evaluate the impact of**  
1296 **the implementation itself. Through this experimenta-**

1297 **tion, we can extend our study to encompass further cri-**  
1298 **teria, such as the impact of the overhead on the energy**  
1299 **consumption and lifespan of the devices themselves.**

## 1300 8. CONCLUSION

1301 In this paper, we introduced a secure consensus-  
1302 based routing method using node reputation metrics  
1303 to identify the most trustworthy route available. The  
1304 consensus-based validation technique employed allows  
1305 us to accurately separate malicious nodes from the  
1306 masses, avoiding them in subsequent communications.  
1307 Furthermore, by using blockchain as a method for dis-  
1308 tributing the computed reputation throughout the net-  
1309 work, we assure that all nodes receive the correct and  
1310 valid reputation values for the entire network. Finally,  
1311 with the application of a reputation decay function-  
1312 ality, we provide the ability for the network to heal it-  
1313 self by re-introducing repaired and salvaged nodes with-  
1314 out user intervention. By implementing our module  
1315 in an AODV-like routing protocol, *AODV-Miner*, and  
1316 analysing the overall efficiency in multiple scenarios  
1317 with different network topologies and complexities, we  
1318 can demonstrate the adaptive capabilities of our net-  
1319 work. Through extensive simulations, we have not only  
1320 proved the increase in security and efficiency of *AODV-*  
1321 *Miner* in relation to AODV, but also the importance of  
1322 reputation-based routing in multi-hop networks. How-  
1323 ever, a significant increase in overhead forms a necessary  
1324 trade off in the strive for increased integrity and security  
1325 in routing activities.

## 1326 ACKNOWLEDGEMENTS

1327 This work was partially supported by a grant from  
1328 CPER DATA and by the European Union's Horizon  
1329 2020 Project "CyberSANE" under Grant Agreement  
1330 No. 833683 addressing the topic SU-ICT-01-2018.

## 1331 REFERENCES

- 1332 [1] L. Pycroft and T. Z. Aziz. "Security of implantable  
1333 medical devices with wireless connections: The  
1334 dangers of cyber-attacks". In: *Expert Review of*  
1335 *Medical Devices* 15.6 (2018). PMID: 29860880,  
1336 pp. 403–406. DOI: 10 . 1080 / 17434440 . 2018 .  
1337 1483235. eprint: [https://doi.org/10.1080/  
1338 17434440.2018.1483235](https://doi.org/10.1080/17434440.2018.1483235). URL: [https://doi.  
1339 org/10.1080/17434440.2018.1483235](https://doi.org/10.1080/17434440.2018.1483235).
- 1340 [2] J. Sengupta, S. Ruj, and S. Das Bit. "A Com-  
1341 prehensive Survey on Attacks, Security Issues and  
1342 Blockchain Solutions for IoT and IIoT". In: *Jour-*  
1343 *nal of Network and Computer Applications* 149  
1344 (2020), p. 102481. ISSN: 1084-8045. DOI: [https:  
1345 //doi.org/10.1016/j.jnca.2019.102481](https://doi.org/10.1016/j.jnca.2019.102481). URL:  
1346 [https://www.sciencedirect.com/science/  
1347 article/pii/S1084804519303418](https://www.sciencedirect.com/science/article/pii/S1084804519303418).



- [3] NARA. *Blockchain White Paper*. White Paper. National Archives and Records Administration, Feb. 2019.
- [4] A. M Antonopoulos. *Mastering Bitcoin: Programming the open blockchain*. ” O’Reilly Media, Inc.”, 2017.
- [5] S. R. Das, C. E. Perkins, and E. M. Belding-Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. July 2003. DOI: 10.17487/RFC3561. URL: <https://rfc-editor.org/rfc/rfc3561.txt>.
- [6] F. Bao, I.-R. Chen, M.J. Chang, and J.-H. Cho. “Hierarchical Trust Management for Wireless Sensor Networks and its Applications to Trust-Based Routing and Intrusion Detection”. In: *IEEE Transactions on Network and Service Management* 9.2 (2012), pp. 169–183. DOI: 10.1109/TCOMM.2012.031912.110179.
- [7] D. K. Bangotra, Y. Singh, A. Selwal, N. Kumar, and P. K. Singh. “A Trust Based Secure Intelligent Opportunistic Routing Protocol for Wireless Sensor Networks”. In: *Wireless Personal Communications* (2021), pp. 1–22.
- [8] N. Djedjig, D. Tandjaoui, F. Medjek, and I. Romdhani. “Trust-aware and cooperative routing protocol for IoT security”. In: *Journal of Information Security and Applications* 52 (2020), p. 102467. ISSN: 2214-2126. DOI: <https://doi.org/10.1016/j.jisa.2020.102467>. URL: <https://www.sciencedirect.com/science/article/pii/S2214212619306751>.
- [9] J. Tang, A. Liu, M. Zhao, and T. Wang. “An aggregate signature based trust routing for data gathering in sensor networks”. In: *Security and Communication Networks* 2018 (2018).
- [10] X. Li, P. Jiang, T. Chen, X. Luo, and Q. Wen. “A survey on the security of blockchain systems”. In: *Future Generation Computer Systems* 107 (2020). DOI: <https://doi.org/10.1016/j.future.2017.08.020>. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X17318332>.
- [11] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani. “Applications of Blockchains in the Internet of Things: A Comprehensive Survey”. In: *IEEE Com. Surveys Tutorials* 21.2 (2019). DOI: 10.1109/COMST.2018.2886932.
- [12] A. Moinet, B. Darties, and J.-L. Baril. “Blockchain based trust & authentication for decentralized sensor networks”. In: *ArXiv abs/1706.01730* (2017).
- [13] Yu Zeng, Xing Zhang, Rizwan Akhtar, and Changda Wang. “A Blockchain-Based Scheme for Secure Data Provenance in Wireless Sensor Networks”. In: *2018 14th International Conference on Mobile Ad-Hoc and Sensor Networks (MSN)*. 2018, pp. 13–18. DOI: 10.1109/MSN.2018.00009.
- [14] C. Machado and C. M. Westphall. “Blockchain incentivized data forwarding in MANETs: Strategies and challenges”. In: *Ad Hoc Networks* 110 (2021), p. 102321. ISSN: 1570-8705. DOI: <https://doi.org/10.1016/j.adhoc.2020.102321>. URL: <https://www.sciencedirect.com/science/article/pii/S1570870520306752>.
- [15] J. Yang, S. He, Y. Xu, L. Chen, and J. Ren. “A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks”. In: *Sensors* 19.4 (2019). ISSN: 1424-8220. DOI: 10.3390/s19040970. URL: <https://www.mdpi.com/1424-8220/19/4/970>.
- [16] H. Lazrag, A. Chehri, R. Saadane, and M. D. Rahmani. “A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT”. In: *Int. Conf. on Signal-Image Technology Internet-Based Systems (SITIS)*. 2019.
- [17] J. Wang, Y. Liu, S. Niu, and H. Song. “Lightweight blockchain assisted secure routing of swarm UAS networking”. In: *Computer Communications* 165 (2021), pp. 131–140. ISSN: 0140-3664. DOI: <https://doi.org/10.1016/j.comcom.2020.11.008>. URL: <https://www.sciencedirect.com/science/article/pii/S0140366420319885>.
- [18] G. Ramezan and C. Leung. “A blockchain-based contractual routing protocol for the internet of things using smart contracts”. In: *Wireless Communications and Mobile Computing* 2018 (2018).
- [19] C.-Y. Tseng, P. Balasubramanyam, C. Ko, R. Limprasittiporn, J. Rowe, and K. Levitt. “A Specification-Based Intrusion Detection System for AODV”. In: *Proceedings of the 1st ACM Workshop on Security of Ad Hoc and Sensor Networks*. SASN ’03. Fairfax, Virginia: Association for Computing Machinery, 2003, pp. 125–134. ISBN: 1581137834. DOI: 10.1145/986858.986876. URL: <https://doi.org/10.1145/986858.986876>.
- [20] S. Bhargava and D.P. Agrawal. “Security enhancements in AODV protocol for wireless ad hoc networks”. In: *IEEE 54th Vehicular Technology Conference. VTC Fall 2001. Proceedings (Cat. No.01CH37211)*. Vol. 4. 2001, 2143–2147 vol.4. DOI: 10.1109/VTC.2001.957123.
- [21] S. Gurung and S. Chauhan. “A Dynamic Threshold Based Algorithm for Improving Security and Performance of AODV under Black-Hole Attack in MANET”. In: *Wirel. Netw.* 25.4 (May 2019),

pp. 1685–1695. ISSN: 1022-0038. DOI: 10.1007/s11276-017-1622-y. URL: <https://doi.org/10.1007/s11276-017-1622-y>.

[22] L. Guillaume, J. van de Sype, L. Schumacher, G. Di Stasi, and R. Canonico. “Adding reputation extensions to AODV-UU”. In: *IEEE Symp. on Comm. and Vehicular Technology in the Benelux (SCVT)*. 2010.

[23] A. Jarjis and G. Kadir. “Blockchain Authentication for AODV Routing Protocol”. In: *2020 Second International Conference on Blockchain Computing and Applications (BCCA)*. 2020, pp. 78–85. DOI: 10.1109/BCCA50787.2020.9274452.

[24] C. Ran, S. Yan, L. Huang, and L. Zhang. “An improved AODV routing security algorithm based on blockchain technology in ad hoc network”. In: *EURASIP Journal on Wireless Communications and Networking* 2021.1 (2021), pp. 1–16.

[25] M. A. A. Careem and A. Dutta. “Reputation based Routing in MANET using Blockchain”. In: *Int. Conference on COMMunication Systems NETWORKS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450.

[26] E. Staddon, V. Loscri, and N. Mitton. “Attack Categorisation for IoT Applications in Critical Infrastructures, a Survey”. In: *Applied Sciences* 11.16 (2021). ISSN: 2076-3417. DOI: 10.3390/app11167228. URL: <https://www.mdpi.com/2076-3417/11/16/7228>.

[27] G. Oikonomou, S. Duquennoy, A. Elsts, J. Eriksson, Y. Tanaka, and N. Tsiftes. “The Contiki-NG open source operating system for next generation IoT devices”. In: *SoftwareX* 18 (2022), p. 101089. ISSN: 2352-7110. DOI: <https://doi.org/10.1016/j.softx.2022.101089>.

[28] A. Reyna, C. Martín, J. Chen, E. Soler, and M. Díaz. “On blockchain and its integration with IoT. Challenges and opportunities”. In: *Future Generation Computer Systems* 88 (2018), pp. 173–190. ISSN: 0167-739X. DOI: <https://doi.org/10.1016/j.future.2018.05.046>. URL: <https://www.sciencedirect.com/science/article/pii/S0167739X17329205>.

## AUTHORS



**Edward Staddon** is currently a PhD. Student in the FUN Team at Inria Lille-Nord Europe, France since Oct. 2019, undertaken as part of the H2020 CyberSANE project. He received his B.Sc. and M.Sc. degrees in Computer Science from Université Bretagne Sud, France in 2016 and 2018 respectively. His research interests

include Wireless Communications, Cyber-Security and the Internet-of-Things.



**Valeria Loscri** is a permanent researcher at Inria Lille since Oct. 2013. From Dec. 2006 to Sept. 2013, she was Research Fellow in the TITAN Lab of the University of Calabria, Italy. She received her MSc and PhD degrees in Computer Science in 2003 and 2007, respectively, from the University of Calabria and her Habilitation à Diriger des recherches in 2018 from Université de Lille

(France). Her research interests focus on emerging technologies for wireless communication. She is on the editorial board of IEEE COMST, TNB, Elsevier ComNet, JNCA. Since 2019, she is Scientific International Delegate for Inria Lille-Nord Europe.



**Nathalie Mitton** received MSc and PhD. degrees in Computer Science from INSA Lyon in 2003 and 2006 respectively. She is currently an Inria full researcher since 2006 and from 2012, the scientific head of the Inria FUN team. Her research interests focus on self-organization from PHY to routing for wireless constrained networks. She has published her research in more than 50

international revues and 120 international conferences. She is involved in H2020 CyberSANE project and in several TPC such as Infocom, PerCom, DCOSS (since 2019), ICC (since 2015), Globecom (since 2017). She also supervises several PhD students and engineers.