Threat detection, identification and quarantine in wireless IoT based Critical Infrastructures

Edward Staddon

Université de Lille
École Doctorale Mathématiques, sciences du numérique et de leurs interactions
Institut de Recherche Centre Inria de l'Université de Lille

Thèse préparée par **Edward Staddon**
en vue de l'obtention du grade de **Docteur en Informatique**
Discipline **Informatique et Applications**

Détection, identification et mise en quarantaine des menaces dans les infrastructures critiques sans fil basées sur l'IoT

Thèse soutenue le 9 décembre 2022 devant le jury composé de :

Thomas Noël	**Président**
Professeur des Universités, Université de Strasbourg	
Thomas Begin	**Rapporteur**
Maître de Conférences HDR, Université Claude Bernard Lyon 1	
Thierry Val	**Rapporteur**
Professeur des Universités, IRIT	
Nathalie Mitton	**Directrice de Thèse**
Directrice de Recherche, Inria	
Valeria Loscri	**Co-Directrice de Thèse**
Chargée de Recherche HDR, Inria	

University of Lille
Doctoral School MAthematics and DIgital Sciences
Research Institute Inria Centre of the University of Lille

Thesis prepared by **Edward Staddon**
with the view of obtaining the degree of **PhD in Computer Science**
Discipline **IT and Applications**

Threat Detection, Identification and Quarantine in Wireless IoT based Critical Infrastructures

---

Thesis defended the 9th December 2022 in-front of the jury comprised of:

Thomas Noël	**President**
Full Professor, Université de Strasbourg	
Thomas Begin	**Reviewer**
Associate Professor, Université Claude Bernard Lyon 1	
Thierry Val	**Reviewer**
Full Professor, IRIT	
Nathalie Mitton	**Supervisor**
Research Director, Inria	
Valeria Loscri	**Co-Supervisor**
Permanent Researcher, Inria	

"In memory of my Grandfather"

Résumé

Ces dernières années, les Infrastructures Critiques (CI) sont devenues la cible de divers cybercriminels aux objectifs néfastes. Grâce aux développements de nouveaux paradigmes technologiques tels que l'Internet des Objets (IoT), fournissant une large gamme de services allant des drones et des capteurs militaires aux dispositifs médicaux portatifs, de nouveaux vecteurs d'attaque ont émergé. Ainsi, les attaquants peuvent désormais cibler les vulnérabilités et caractéristiques des équipements IoT, ayant un impact sur la CI sous-jacente et plus largement affecter la population. Les appareils qui utilisent des technologies sans fil multi-sauts sont exposés à un risque d'attaques entre autre de routage, ciblant la transmission de données. C'est dans ce contexte que cette thèse s'inscrit, visant à fournir les outils nécessaires permettant de détecter et d'éviter ces menaces.

Pour combattre ces menaces, elles doivent être analysées, ouvrant ainsi la voie vers des systèmes de détection et de mitigation modernes. Grâce à de nouveaux systèmes de gestion d'incident, d'alerte et de réponse, tels que la plateforme proposée et développée dans le cadre du projet Européen H2020 CyberSANE, la protection des CI s'accroît. Dans cette thèse, nous portons notre attention sur la sécurité des réseaux IoT multi-sauts et proposons un algorithme de consensus fondé sur l'observation des nœuds du réseau, leur permettant d'analyser le comportement de leurs voisins. Les valeurs obtenues par le consensus sont ensuite distribuées à travers le réseau via la technologie blockchain, fournissant un système de stockage et de distribution à la fois transparent et sécurisé pour l'ensemble des nœuds du réseau. Ainsi, nous fournissons aux nœuds la capacité d'exprimer la fiabilité de leurs voisins en utilisant la notion de réputation, permettant la séparation rapide des entités malicieuses. Nous proposons également une méthode, permettant l'intégration de cette réputation dans plusieurs protocoles de routage multi-sauts. Grâce à cette approche, nous sommes capables d'influencer les algorithmes de sélection de chemins, privilégiant ainsi les routes les plus fiables quand cela est possible. En permettant à ce système d'intégration de s'adapter non seulement au protocole de routage, mais au réseau lui-même, nous pouvons détecter avec une précision maximale les équipements malicieux. Dans un premier temps, nous évaluons cette approche avec deux protocoles de routage réactif, Ad hoc On-Demand Distance Vector (AODV) et Dynamic Source Routing (DSR). Puis nous abordons comment l'intégrer avec des protocoles proactifs tels que Routing Protocol for Low-Power and Lossy Networks (RPL). Grâce à cette analyse, nous démontrons une augmentation de l'efficacité de routage dans le réseau entier ainsi que la réduction de l'impact des entités malicieuses sur le routage, grâce à notre module de consensus basé sur la réputation.

Enfin, nous proposons une extension à ce module avec la notion de quarantaine pour les nœuds malicieux. En définissant divers niveaux de menaces, nous pouvons influencer l'intensité du niveau de quarantaine ainsi que les conséquences directes des actions malicieuses entreprises. Cette extension pousse la sélection des routes encore plus loin, en isolant les menaces de haut niveau en les empêchant de participer ou de contribuer aux activités du réseau.

Abstract

In recent years, Critical Infrastructures (CIs) have become under siege from various cyber criminals with nefarious objectives. With the advancements of new technological paradigms such as the Internet-of-Things (IoT) providing a much wider range of services, from military drones and sensors to wearable healthcare devices, new attack vectors have emerged. As a result, attackers can target the vulnerabilities and characteristics of IoT devices, impacting the underlying CI and even innocent bystanders. These devices which use multi-hop wireless technologies are at risk of routing-based attacks, directly targeting data transmission. It is in this context that this thesis is situated, aiming to provide necessary tools to detect and avoid these threats.

To help combat these threats, they must first be analysed, paving the way for more modern and adaptable detection and mitigation systems. Thanks to novel incident handling, warning and response systems, such as the platform proposed and developed as part of the H2020 EU CyberSANE project, CI protection is ever increasing. In this thesis, we turn our attention to IoT multi-hop security, proposing a consensus-based observational algorithm allowing network nodes to analyse the behaviour of their neighbours. The values produced from this consensus metric are subsequently shared throughout the network using blockchain technology, providing a transparent and secure distribution and storage system for all network nodes. Thus, we grant the capability for these nodes to express the trustworthiness of these neighbours using the notion of reputation, quickly separating malicious entities from good ones. We also propose an integration method, allowing for these reputational values to be incorporated into multiple multi-hop routing protocols. Thanks to this approach, we are capable of influencing the protocol's path selection algorithms, privileging higher reputable routes where available. By allowing this integration system to not only adapt to the protocol at hand but also the network itself, we can allow maximum precision for the identification of malicious devices. Firstly, we evaluate this approach in conjunction with two reactive routing protocols, Ad hoc On-Demand Distance Vector (AODV) and Dynamic Source Routing (DSR). Subsequently, we also discuss the possible integration with proactive protocols, such as Routing Protocol for Low-Power and Lossy Networks (RPL). Thanks to this analysis, we demonstrate an increase in network routing efficiency as well as a reduction of malicious impact on network routing, thanks to our consensus-based reputation module.

Finally, we propose an extension to this module introducing the notion of quarantine for malicious nodes. By defining various threat levels, we can influence the severity of the quarantine response as well as the direct consequences of malicious actions. This extension pushes the path selection even further, effectively isolating high level threats, stopping them from partaking on contributing towards network-based activities.

Acknowledgements

I would like to begin by expressing my gratitude to Dr. Thomas Begin and Dr. Thierry Val for agreeing to review this manuscript as well as to Dr. Thomas Noel for agreeing to participate in the evaluation of my work. I know that in this period, timetables are very full and I appreciate you all taking the time out of your busy schedules to partake in my defence.

I would also like to thank my two supervisors Dr. Nathalie Mitton and Dr. Valeria Loscri for being by my side on this journey. I especially appreciate their unyielding guidance with suggestions at each stage of my evolution during my work as well as their patience, even more so when I was struggling with various difficulties both related to this PhD and outside. I would like to extend my appreciation to the members of the CyberSANE consortium for their aide in the various tasks, in particular Ana Maria Morales for her help in getting the CyberSANE dissemination train rolling in the correct direction.

Of course, it goes without saying that I give my thanks to Inria for the opportunity to study with them as well as wholeheartedly thank the other members of the FUN team for the moments we spent together. I have fond memories of our discussions around the office bar, during lunches as well as our various trips and events such as kebabs, Thursday badminton games and trips to the gym. Speaking of sports, words cannot express my gratitude to Emilie Bout, who was always there when I needed moral support and who kept me going these three years, even with my many many questions to which she always had an answer, even after her departure.

I would like to conclude my thanks with my family, who were always there for me whenever I needed, allowing me a getaway from the work environment to unwind. In particular, my girlfriend Alexia who put up with my work talk during our evening phone calls and kept me motivated during the evenings and weekends.

Thank-you all for all you have done in these past three years.

Contents

List of Figures

List of Tables

1 Introduction and Context

In this chapter, we provide an introduction to the work achieved in this thesis. We also present the context in which this work was performed, that of wireless Critical Infrastructures (CIs) and multi-hop IoT networks. We also provide an overview of the contributions towards this thesis and the CyberSANE H2020 European project, under which this work was performed. Finally, we list the different publications and deliverables produced presenting all achievements both towards CyberSANE, and the thesis itself.

1.1 Wireless Communications

The human race is inherently a very social species. Back before any wide spread communication methods were available, we used to meet up and communicate in person, gossiping about events and learning about each other. With the first appearance of the postbox back in 1653 in Paris came the first chance for communicating at a distance [1]. The more this concept caught on across Europe in the 1840s and 50s, the more it became apparent that keeping in touch with others was important to us.

This idea was further solidified with the invention of the telephone in 1876, allowing the first real-time communication at a distance. Although this idea took time to catch on, it eventually spread like wildfire, where in almost 100 years later, practically every household possessed a landline and would use it regularly. However, innovation didn't stop there and continued to find new ways to communicate. Following the invention of what would be called the internet, the first email networks began to forge in the 1970, expanding to proprietary "electronic mail systems" within ten years [2].

The technological advancements have continued up until today, where we now have the capabilities of making phone calls and sending emails directly from a small device in our pockets. However, these devices rely on one important invention: Wireless Communications. Through the manipulation of radio waves, it is possible to relay data in a binary form between two digital devices, allowing them to exchange information. With this discovery came a race to develop applications for such a technology, reserving certain frequencies for specific applications.

To solve this overuse issue, specific frequency ranges were provided for commercial and public use, referred to internationally as the Industrial, Scientific and Medical Radio Band (ISM Band). Due to these restrictions, many public wireless protocols have been developed specifically to function in this overcrowded band, including *Wi-Fi* [3], *Bluetooth* [4] as well as *802.15.4* Personal Area Networks (PANs) such as *Zigbee* [5]. Since the ISM Band is free access, it is, therefore, shared with a multitude of different devices, from home network devices to microwaves. As such, since any transmitted data is travelling through the public domain, all communications are at the risk of being attacked, such as being captured,

Figure 1.1: The first postbox on record from Paris in 1653 [1]

[1]: Wikipedia. *Post box*. Sept. 19, 2022. URL: https://en.wikipedia.org/wiki/Post_box (visited on Sept. 26, 2022)

[2]: Wikipedia. *History of email*. Sept. 21, 2022. URL: https://en.wikipedia.org/wiki/History_of_email (visited on Sept. 26, 2022)

[3]: Wikipedia. *Wi-Fi*. Oct. 9, 2022. URL: https://en.wikipedia.org/wiki/Wi-Fi (visited on Oct. 11, 2022)

[4]: Wikipedia. *Bluetooth*. Oct. 5, 2022. URL: https://en.wikipedia.org/wiki/Bluetooth (visited on Oct. 11, 2022)

[5]: Wikipedia. *Zigbee*. Oct. 5, 2022. URL: https://en.wikipedia.org/wiki/Zigbee (visited on Oct. 11, 2022)

Figure 1.2: The first electronic message was shared between these two PDP-10 computers in BBN Technologies in 1971, connected only through the ARPANET, the first wide-area packet-switched network implementing the TCP/IP suite [2]

[6]: Mathy Vanhoef and Frank Piessens. 'Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2'. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017. DOI: 10.1145/3133956.3134027

analysed, or even exploited. Furthermore, although wireless networks provide a much welcome freedom, no longer needing to be in-front of a desk all day, their wired brothers possessed slightly higher security where direct access to the infrastructure was required to access any communications. With wireless technologies, attackers can exploit the wireless radio range to interact with the target network without being directly adjacent to the target.

Protecting and securing wireless communications is an ongoing challenge, since the medium is both shared and inherently unprotected. Many solutions exist to protect the exchange of data, such as the common security protocols Wired Equivalent Privacy (WEP) and Wi-Fi Protected Access II (WPA2). However, these systems are not infallible and, when broken, lose their usefulness. This is even more significant when noting the widespread use of WPA2, which, after 14 years of certification, was broken in 2017 [6] with its replacement Wi-Fi Protected Access III (WPA3) only commencing its widespread deployment in 2020 where WPA3 became a mandatory requirement for "Wi-Fi CERTIFIED" devices. Furthermore, with the increase of wireless capabilities comes to ever increasing hunger for innovation, thus expanding existing devices capabilities through wireless technologies, entering into a new area of communications.

1.2 The Internet-of-Things

There are many devices in our everyday lives which make our lives easier. This is the case of a watch, showing the time; a phone, allowing to talk to our loved ones; a car, to take us places; and even a fridge, to keep our chocolates nice and fresh. However, with the wide spread adoption of wireless technologies, these well known and used "Things" are gaining extra functionalities and are capable of thinking for themselves. For instance, our watches are now capable of reading our heart rate and alerting medical services if something is wrong. On the other hand, our phones now possess the capabilities to surf the internet, even allowing us to watch videos whilst our car drives us autonomously to our destination after having told it to do so through voice control. Even our fridges aren't spared from this progress, providing us with a much needed method for adding new items to our shopping list, allowing us to never forget our chocolate again.

This interconnection of devices is known as the Internet-of-Things (IoT), and opens the door to new areas of application. Indeed, with these devices comes certain advantages, making lives easier for everyday citizens. This is the case for areas such as smart agriculture, where IoT devices can be deployed in fields, allowing the surveillance of crops without direct intervention from the farmer. By utilising their wireless communications capabilities, these sensors can even alert the fields irrigation system before the ground becomes too dry, thus keeping the crops alive.

Keeping things alive is a common use case for these devices. Indeed, IoT devices have a widespread use in the healthcare sector thanks to the deployment of wearable smart healthcare devices. Complementing the previous example of a smart watch, these devices are directly responsible

for the patients health, reporting back to the hospital with the patients vitals, even providing the capability for doctors to directly intervene upon the patients treatment. This is the case for smart pacemakers and smart insulin pumps, which not only perform the functions of their basic non-smart ancestors, also allow constant medical surveillance and alerts.

As a result, these devices possess certain characteristics limiting their functionalities. Indeed, due to their size and different use cases, as with the example of wearable devices, certain hardware restrictions are necessary. The most important of which is the use of batteries, severely limiting the computational power of these devices in an attempt to provide a long battery life. Furthermore, these devices possess limited storage space, thus limiting the applications and implementation options available.

Due to their specific characteristics, IoT devices also possess certain security principles which govern their use. Due to their unique nature, IoT security is an ever developing challenge where multiple security concerns must be identified to aid in the development of security systems and protection methods. Many of these security principles presented in Figure 1.3 are not specific to IoT applications and are shared with cyber systems, such as *Confidentiality*, *Integrity*, *Availability* and *Authentication* [7]. However, specific security features revolve around the different characteristics of IoT devices and networks [8].

[7]: Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. 'Internet of things (IoT) security: Current status, challenges and prospective measures'. In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. DOI: 10.1109/ICITST.2015.7412116

[8]: Wei Zhou, Yan Jia, Anni Peng, Yuqing Zhang, and Peng Liu. 'The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved'. In: *IEEE Internet of Things Journal* 6 (2019). DOI: 10.1109/JIOT.2018.2847733

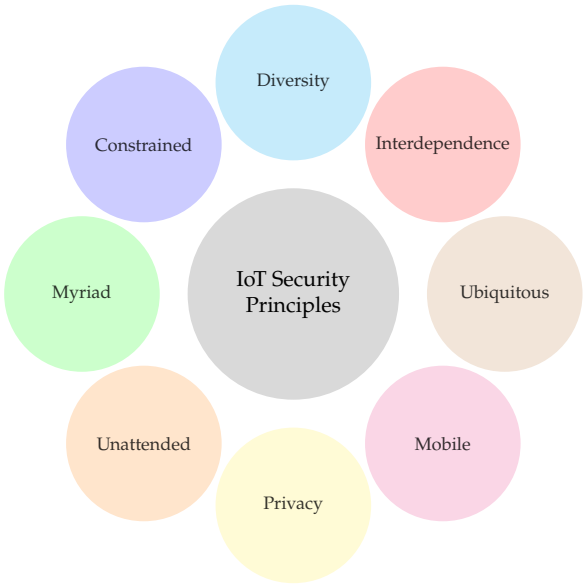


Figure 1.3: The eight security principals of the IoT

- **Interdependence**: These devices must function in tandem with others, allowing interconnections between each other. This is the case for Smart Home applications, for example with turning on a light when the light level drops below a certain threshold. In this case, an attacker can trick the sensor into believing the light level is higher, thus deactivating the bulbs in the vicinity, allowing them to penetrate into the accommodation undetected.
- **Diversity**: Depending on the application, the devices hardware changes (i.e., smart bulbs, plugs, switches, etc.). The more diversity

in the network, the higher the chance for specific vulnerabilities to exist an be exploited by a potential attacker.

- **Constrained**: As mentioned previously, IoT devices possess hardware limitations, such as limited energy reserves or computational capabilities and communication technologies. These limitations are dependant on both the manufacturer, as well as the devices application and use case (i.e., medical sensors with long battery life compared to military drones with powerful cameras)
- **Myriad**: Devices are easier to create and deploy in large quantities, increasing the network complexity. As a result, the more devices are in an IoT network, the higher the risk of compromise due once more to the large diversity of equipment, thus potential vulnerabilities.
- **Unattended**: In certain areas, such as agriculture or military applications, devices are deployed in remote inaccessible areas. This means they must function autonomously, capable of communicating with each other using their wireless protocols, without human interaction or supervision.
- **Privacy**: In many applications, IoT devices capture and process large quantities of personal data, such as wearable healthcare devices. This data ca be used against us by an attacker and must, therefore, be protected on both the device and during communications with other devices or servers.
- **Mobile**: With wearable devices comes the notion of mobility, where the IoT equipment must adapt to dynamic changes in its environment. To allow continued communications, devices can jump from one network to another, allowing communications with multiple devices.
- **Ubiquitous**: The increasing presence of IoT devices increases the risk of security related incidents based upon invalid human interaction. Information Technology (IT) issues are sometimes known as "the error is generally found between the chair and the keyboard" where human error is a contributing factor, meaning threats can come from many different angles including the user themself.

1.3 Thesis Context

In this section, we present the context in which this thesis is situated, providing a general idea towards the importance of security in these domains. Firstly, we explore the notion of Critical Information Infrastructure (CII) and why it is important to secure them against ever growing threats. We then take a look at the CyberSANE platform, a novel security solution to help secure these infrastructures against the cyber menace. Finally, we present the notion of multi-hop wireless networks, an area in constant expansion but not widely covered by current general security solutions.

1.3.1 Critical Information Infrastructures

When a cyber system is compromised, the objective of the attack can vary. One of the most common is to access private and secure information to either render it public or to sell it to the highest bidder. This was

Figure 1.4: The different sectors that are considered to be Critical Infrastructures (CIs) [9]

the case of the attack against a South Korean nuclear and hydroelectric plan in December 2014, where the attackers stole both technical reactor information as well as employee personal data [10]. Another possibility is to simply impact the operation of the target itself, rendering it essentially unusable and causing disruptions to its operational control as well as any products it may produce. This happened to the Saudi Arabia petrochemical plant back in August 2018, where the attacker sabotaged the plants production chain in a what is believed to be an attempt to cause an explosion [11].

These types of attacks leave a critical mark upon the target infrastructure due to their important and vital nature with regards to their respective nations. These CIs cover multiple sectors as shown in Figure 1.4, including healthcare, transport, energy and finance, as well as government based facilities which are often targets for cyber attacks. As a result, the critical nature of these infrastructures means that a successful attack could not only cause significant disruption to the nation itself, but in certain cases could even result in large numbers of civilian casualties.

With the advancements made in the areas of IoT, more and more CI dependant technologies are being deployed amongst the civilian population. This is the case of the aforementioned small wearable healthcare devices belonging to healthcare providers, such as smart vital monitors which share medical data with hospital staff and can, in the case of smart insulin pumps for example, make decisions regarding dosage, adapting to the patients biology. As such, in the hands of civilians these devices can result in deadly consequences if the devices were to become compromised. As a consequence, CI protection is paramount and part of many ongoing cyber security research projects [12].

[9]: Cybersecurity & Infrastructure Security Agency. *Identifying Critical Infrastructure During COVID-19*. Mar. 19, 2020. URL: https://www.cisa.gov/identifying-critical-infrastructure-during-covid-19 (visited on Sept. 25, 2022)

[10]: Justin McCurry. 'South Korean nuclear operator hacked amid cyber-attack fears'. In: *The Guardian* (2014). URL: https://www.theguardian.com/world/2014/dec/22/south-korea-nuclear-power-cyber-attack-hack (visited on Sept. 25, 2022)

[11]: Nicole Perlroth and Clifford Krauss. 'A cyber attack in Saudi Arabia failed to cause carnage, but the next attempt could be deadly'. In: *The Independent* (2018). URL: https://www.independent.co.uk/news/long_reads/cyber-warfare-saudi-arabia-petrochemical-security-america-a8258636.html (visited on Sept. 25, 2022)

[12]: Eleonora Viganò, Michele Loi, and Emad Yaghmaei. 'Cybersecurity of critical infrastructure'. In: *The Ethics of Cybersecurity*. Springer, Cham, 2020. DOI: 10.1007/978-3-030-29053-5_8

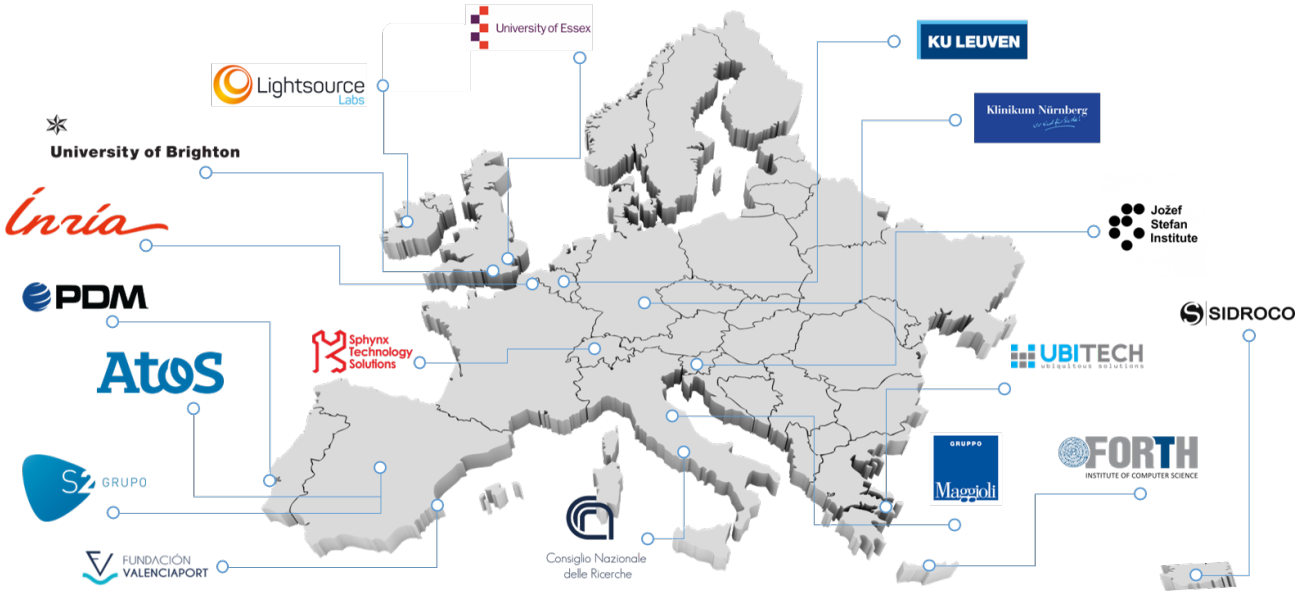


Figure 1.5: An overview of the 17 CyberSANE partners from 13 European countries

Figure 1.6: CyberSANE logo [13]

[13]: CyberSANE. *Website*. Dec. 1, 2019. URL: https://www.cybersane-project.eu/ (visited on July 28, 2022)

1.3.2 Critical Information Infrastructure Security Platform - CyberSANE

To respond to the threats targeting these platforms, a novel cyber security platform was proposed to enhance the security and resilience of Critical Information Infrastructures (CIIs). To achieve this, the European H2020 Project CyberSANE [13] was proposed, providing an innovative, knowledge-based, collaborative security and response dynamic system to support and guide security officers in their objectives. This project regroups 17 partners from 13 European countries shown in Figure 1.5, providing various levels of expertise from industry to academia, including three end user CIs to evaluate the CyberSANE system:

1. **Valenciaport Foundation**: The Spanish port of Valencia is the fifth largest European port. Their pilot revolves around the security of their container cargo transportation service, allowing CyberSANE to be tested in a maritime transportation scenario.
2. **Lightsource Labs Ltd**: The Irish based energy company is a global leader in the funding, development and long-term operation of solar PV projects. Their pilot concerning the security aspects of their solar energy management platform, allows CyberSANE to be confronting against attacks targeting the energy sector.
3. **Klinikum Nürnberg**: The German hospital is one of the largest municipal hospitals in Europe. With their pilot concerning the cyber-threat identification and communication in healthcare, they provide the tools necessary for CyberSANE to be tested in an important area, that of healthcare itself.

The CyberSANE System is capable of implementing all phases of the Cyber incident handling life-cycle for increasing the agility of the security professionals and encourages continuous learning. Using a horizontal business logic, CyberSANE system is composed by five components

Figure 1.7: An overview of the CyberSANE core architecture, presenting the five separate components, each with their own specifications and objectives

which integrate various existing tools provided by consortium partners to offer specific functionalities and features both, to the component and to the whole system. Additional business services such as advanced reports, notifications, and others, have been included in a unified manner, built upon the existing services. The CyberSANE Core, presented in Figure 1.7, interoperates with the so-called "CyberSANE Ecosystem", an architectural layer hosting all project partners' tools that provide a significant set of services and features for each of the main CyberSANE components:

- **LiveNet**: *Live Security Monitoring and Analysis*
  Responsible for real-time threat detection, this component is also capable of preventing and mitigating the effects of an infection or intrusion in the underlying CI. As a result, LiveNet serves as the primary interface between between the CI and the rest of the CyberSANE System.
- **DarkNet**: *Deep and Dark Web Mining and Intelligence*
  This component performs various reconnaissance type activities, through the exploitation and analysis of information embedded in user generated data from various electronic devices in an attempt to identify potential security risk or threat information. By using a mixture of textual and meta-data content taken from various electronic streams including social media as well as deep and dark web forums, DarkNet is capable of alerting for potential upcoming attacks or leaked data.
- **HybridNet**: *Data Fusion, Risk Evaluation and Event Management*
  This component is responsible for aiding in the analysis of security event data, by providing the necessary intelligence to perform both effective and efficient analysis. This information is derived from internally generated security information, as well as various reports

provided by both LiveNet and DarkNet. As a result, it contains three main elements: *Anomaly Detection Engine*, *Incident Analysis & Response*, as well as *Decision-Making, Warning and Notification*.

- **ShareNet**: *Intelligence and Information Sharing Dissemination*
  An important element of security is learning from our own mistakes, as well as those of others. This components aids with this task, by providing threat intelligence and information sharing capabilities, allowing the data to be exchanged with other CIs, as well as external involved parties. ShareNet also provided necessary trust information, allowing to identify the source of incoming information, as well as its trustworthiness.
- **PrivacyNet**: *Privacy and Data Protection Orchestrator*
  When sharing information publicly, it is important to respect the different privacy and data protection laws. To achieve this, this component takes on a managerial role, orchestrating the application of innovative privacy mechanisms, maximising the levels of confidentiality and data protection. As a result, PrivacyNet assures compliance with the highly-demanding provisions indicated in the GDPR, as well as different local variations imposed upon each infrastructure, thus protecting sensitive incident-related information both within and outside CIs

In this context, the work presented in this thesis concerns both LiveNet and HybridNet components.

1.3.3 Wireless Multi-Hop Networks

Wireless communications have become a part of our everyday lives. They allow us to check our emails on the fly and work from home from our patios, enjoying the summer sun. In this context, Wi-Fi networks function in a mode called *Infrastructure*, where direct communications with a, preferably secure, access point is needed. Although this functions very well, if we wanted to work from a bit further away down the garden, we may loose our Wi-Fi connection, resulting in no internet access at all.

In many scenarios, direct access to an access point, otherwise known as sink or gateway, is not always available. Indeed, in the case of smart agriculture for example, deploying multiple devices across multiple hectares would require a large number of access points to be installed, thus adding the subsequent problem of running the network cables. To resolve this issue, IoT devices can take on the functionalities of routers, allowing them to relay information onwards towards the intended destination. With this method, the farmer needs only to install a single sink at a central point in their farm and then all sensors would be able to relay the data back home.

To be able to determine the correct path to take, there are two options. Either the devices must be informed before hand as to the direction in which to forward data, or they must learn for themselves. In some cases, the former approach is used, where administrators preload the devices with static routing tables, providing them with the necessary information to return to the sink. However, many protocols exist to determine the best route to take back to the source, or to the requested destination if this is different.

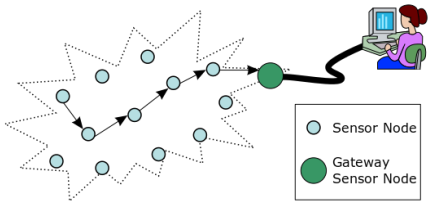


Figure 1.8: Multi-Hop Network example, where devices (light-blue) must relay information back to the gateway (**green**) for it to reach the internet or the user [14]

[14]: Wikipedia. *Multi-hop routing*. June 9, 2021. URL: https://en.wikipedia.org/wiki/Multi-hop_routing (visited on Oct. 17, 2022)

Whatever the protocol, they all employ path selection methods to help determine the best route by their standards, which could be the number of hops or another specific metric. In any-case, these generic protocols do not possess the necessary capabilities to detect malicious threats attacking the network, thus falling directly into the hands of the attackers. As a result, much work has been performed towards securing these protocols in the literature, each with their own advantages. However, many are devised with a single protocol in mind, thus removing the possibility of changing protocols without completely revamping the routing process. Thus weakness leaves a gap in multi-hop routing security, one that needs to be plugged to provide increased protection to these networks.

1.4 Contributions

In this thesis, two types of contributions were performed. Firstly, work was performed towards the elaboration of the CyberSANE platform, aiding the consortium with different aspects related to CI security. Secondly, an analysis towards the general context of this thesis was also performed.

1.4.1 CyberSANE Collaborative Activities

During the CyberSANE project, multiple contributions were provided towards the conception of the system platform. In total, three different areas of contribution were needed, each targetting different elements important to both the conception as well as the promotion of the project and resulting platform. These areas are as follows:

1. **Work-Package 2: User Requirements and Reference Scenarios**
   The work here revolved around a comprehensive analysis of existing incident handling techniques applied to CIIs. As a result, a literature review was performed, determining the different trends in threat detection, as well as the different categorisation techniques used to organise different threats against a specific target. This work resulted in the production of the background needed for the deliverable **D2.1 - Cyber Incident Handling Trend Analysis**.
2. **Work-Package 3: Live Security Monitoring and Analysis (LiveNet)**
   Here, we lead the work related to the study of the existing threat landscape towards CIIs in an effort to propose new threat models. During this task, a novel comprehensive threat taxonomy was produced for the creation of LiveNet's threat models. This new taxonomy was the main objective of the deliverable **D3.1 - Taxonomy of Threat Landscape for CIIs**.
3. **Work-Package 11: Dissemination, Exploitation, Sustainability and Market Take up**
   In this task we lead the dissemination activities, through the creation and upkeep of various digital sharing platforms including the CyberSANE website and social media accounts. This work resulted in three deliverables, **D11.2 - Initial**, **D11.4 - Intermediate** and **D11.6 - Final Report on Dissemination and Communication Activities**.

1.4.2 Thesis related Activities

As presented previously, CyberSANE provides a secure cyber security platform the the protection of CIs. Furthermore, with the increasing employment of IoT devices in their midst, these communications are in need of solutions. As already mentioned, the multi-hop wireless paradigm allows devices to relay information on-wards from a source to a destination. When coupled with IoT devices, these networks can provide a wide range of services, sometimes critical. It is here that our expertise comes into play, analysing problems and providing solutions to help secure these communications, thus reinforcing the underlying CI for CyberSANE.

To achieve this objective we provide detection, identification and quarantine capabilities for CyberSANE, allowing to mitigate threats targetting the routing process in multi-hop networks. There are three contributions towards this objective, presented as follows:

1. **Threat Detection and Identification**: we propose a trust-based methodology for the detection of threats perceived in wireless multi-hop networks. By taking inspiration from blockchain technology, we derive a consensus metric allowing accurate behavioural validation, extending the threat detection to the identification of the malicious node in the network. With the incorporation of blockchain distribution, the behavioural results, expressed as a reputational value can be shared securely throughout the network, informing all devices of the resulting analysis.
2. **Protocol Integration**: we propose a method for integrating this proposed "Miner" module into reactive multi-hop routing protocols, thus allowing the reputational values to influence the path selection process. The resulting intertwining allows the underlying routing protocol to select the most reputable route, avoiding malicious entities as much as possible.
3. **Threat Quarantine**: we propose an extension to the original reputation-module, providing further behavioural metrics allowing to determine the threat level posed by the malicious device. With this information, it is possible to dynamically isolate the threat, stopping it from partaking in any and all routing operations.

Figure 1.9: Contiki-NG logo [15]

[15]: Github. *Contiki-NG Github repository*. Aug. 4, 2021. URL: https://github.com/contiki-ng/contiki-ng (visited on Aug. 7, 2022)

[16]: George Oikonomou, Simon Duquennoy, Atis Elsts, Joakim Eriksson, Yasuyuki Tanaka, and Nicolas Tsiftes. 'The Contiki-NG open source operating system for next generation IoT devices'. In: *SoftwareX* 18 (2022). DOI: 10.1016/j.softx.2022.101089

[17]: Fredrik Osterlind, Adam Dunkels, Joakim Eriksson, Niclas Finne, and Thiemo Voigt. 'Cross-Level Sensor Network Simulation with COOJA'. In: *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. 2006. DOI: 10.1109/LCN.2006.322172

Simulation Environment

To accurately evaluate the previous contributions, we perform extensive simulations using the Contiki-NG Operating System (OS) [16] using the network simulator Cooja [17]. Contiki-NG is a next generation OS for resource constrained IoT devices, utilising an Request for Comments (RFC)-compliant, low-power IPv6 net-stack. This net-stack runs a comprehensive 6LoWPAN network layer along with a non-beacon-enabled always on CSMA Media Access Control (MAC) layer residing on an 802.15.4 radio layer.

The built-in simulator, Cooja, supports many different implementation types, simulating OS and network functionalities in a container module. Implemented in Java using the MSPsim emulator, Cooja provides accurate simulations of different hardware and radio transmitters. Figure 1.10

presents an overview of the simulators interface during a simulation. Thanks to the interface, we can easily depict the different communications between the network nodes, as well as the communications range for easy understanding. During our simulations, we simulated our module using the *Cooja* mote type, thus allowing accurate results from the MSPsim emulator on native Linux hardware running Ubuntu 18.04 OS.

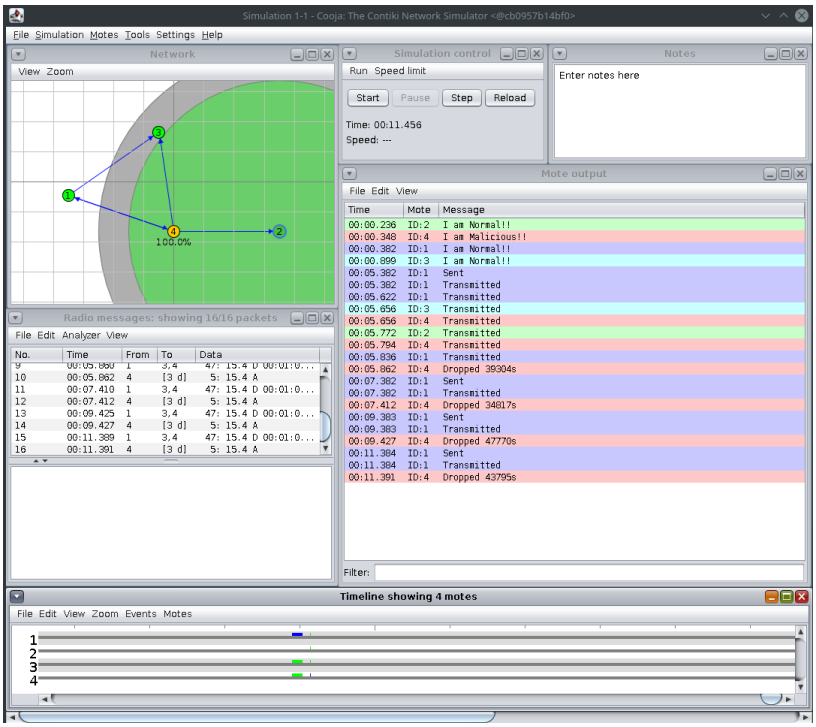


Figure 1.10: A screenshot of the interface of the Cooja network simulator

1.5 Thesis Outline

This thesis provides a comprehensive overview of the work achieved towards the security of multi-hop IoT networks, in the context of Cyber-SANE and CIs protection. It is organised into six individual chapters, each presenting a piece of the puzzle as follows:

We begin by providing an overview of the context in which this thesis resides in Chapter 2. The different aspects and information related to the security of wireless CIs are defined, including an introduction to the notion of a "cyber attack". This is follow up with an overview of the different detection methodologies at the disposal of security experts to detect threats in both CIs and IoT networks. We then take a look at how these systems can both be protected, as well as reduce the impact of an attack itself and how to recover from the post-attack fallout.

Our attention in Chapter 3 is then turned towards the notion of trustworthiness and how it can be applied to multi-hop networks in order to identify and avoid malicious devices. We commence by presenting the correlation between the notions of trust and reputation, illustrating how each influences the other. This is followed with an evaluation of how such methods can be adapted to IoT networks based upon previous studies in the area, in particular through the use of blockchain technology to distribute the reputational information throughout the network. From

here, we propose a consensus-based behavioural validation mechanism, capable of detecting threats and identifying the source during routing activities. This mechanism also provides extra security, assuring only valid information is distributed through the blockchain. We conclude this chapter with an evaluation of the evolution of reputation when subjected to attacks, illustrating its use in IoT network security.

From this work, in Chapter 4 we interest ourselves with integration methods, allowing to intertwine our previous module with existing routing protocols. In doing so, we can allow the notion of trustworthiness to help in the path selection process, guiding the underlying protocol towards selecting the best, most reputable route possible to reach a certain destination. We begin with an overview of existing multi-hop routing protocols in an effort to understand how they function, as well as their path selection algorithms. With this information, we propose a method allowing us to seemingly integrate our module with reactive protocols, all the while requiring as little modifications to the original as possible, thus keeping the path selection as original as possible. We then evaluate our proposition through extensive simulations with the Contiki-NG OS and the Cooja simulator with two reactive routing protocols, AODV and DSR, before proposing a theoretical integration model for proactive protocols, commencing with RPL.

In Chapter 5, we propose a rework of our original reputation module, introducing the notion of quarantine, thus providing the ability to isolate network nodes from routing activities. First, we introduce the importance of quarantine and its application in the world of IT security, in particular the different aspects for a good quarantine system with regards to IoT devices and how such a method would function. We also evaluate the possibility of multi-level quarantine, adapting itself to both the network and the severity of the threat posed by the malicious device. We then detail a replacement consensus methodology, allowing a more robust agreement mechanism as well as providing extra security measures to both detect and respond to threats during behavioural analysis. Finally, we demonstrate the impact on network efficiency with the same protocols as the previous chapter, comparing the difference in the results, in particular regarding the trade-offs', indicating the importance as well as difficulties with such an approach.

Finally, we conclude this work in Chapter 6, summarising the different contributions from our research. We start with an overview of the research and creations made towards the elaboration of the CyberSANE platform, before presenting the different contributions towards the general context of this thesis. We finally depart ways with an open mind, analysing the potential extensions to our work, increasing the protection of wireless multi-hop IoT networks.

1.6 List of Publications

Journals

- Edward Staddon, Valeria Loscri and Nathalie Mitton. "Attack Categorisation for IoT Applications in Critical Infrastructures, a

Survey". In: *Applied Sciences, MDPI*, 2021, 11 (16), pp.7228. DOI: 10.3390/app11167228

- Edward Staddon, Valeria Loscri and Nathalie Mitton. "A Consensus-Based Approach to Reputational Routing in Multi-Hop Networks". In: *ITU Journal on Future and Evolving Technologies - Innovative network solutions for future services*, 2023, Accepted.

International Conferences

- Edward Staddon, Valeria Loscri and Nathalie Mitton. "AODV-Miner: Consensus-Based Routing Using Node Reputation". In: *WiMob 2022 - 18th International Conference on Wireless and Mobile Computing, Networking and Communications*, Oct. 2022, Thessaloniki, Greece. URL: https://hal.inria.fr/hal-03787034/

National Conferences

- Edward Staddon, Valeria Loscri and Nathalie Mitton. "AODV-Miner : Routage par Consensus Basé sur la Réputation". In: *CORES 2022 – 7ème Rencontres Francophones sur la Conception de Protocoles, l'Évaluation de Performance et l'Expérimentation des Réseaux de Communication*, May 2022, Saint-Rémy-Lès-Chevreuse, France. URL: https://hal.archives-ouvertes.fr/hal-03659299/

Scientific Popularisation

- Edward Staddon, Nathalie Mitton and Valeria Loscri. "Comment marchent votre réseau wifi et vos appareils connectés – et pourquoi ils sont vulnérables aux attaques informatiques". In: *The Conversation*, 2021. URL: https://hal.inria.fr/hal-03273579

Project Deliverables

- Ribeiro, L. L., Campos, L. M., Osliak, O., Beyer, S., Zamarripa, S., Loscri, V., Mitton, N., Staddon, E., Hatzikou, M., Karantjias, A., Papastergiou, S., Karagiorgou, S., Athanatos, M., Spanoudakis, G., Karypidis, P. - A., Lytos, A., Ismail, U. and Mouratidis, H., "CyberSANE Project. Deliverable 2.1. Cyber Incident Handling Trend Analysis". Ed. by Nathalie Mitton. Research Report. Feb. 2020. URL: https://www.cybersane-project.eu/deliverables/#D2.1
- Ribeiro, L. L., Correia Loureiro, F., Ruiz, J. F., Becerra, C., Staddon, E., Mitton, N., Loscri, V., Papastergiou, S., Hatzikou, M., Boukis, G., Pruccoli, A., Tamburini, N., Serra, M., Papadogiannai, E., Athanatos, M., Kontakis, K., Saoulidis, T., Mouratidis, H. and Ismail, U., "CyberSANE Project. Deliverable 3.1. Taxonomy of threat landscape for CIIs". Ed. by Edward Staddon. Research Report. June 2020. URL: https://www.cybersane-project.eu/deliverables/#D3.1
- Ribeiro, L. L., Campos, L. M., Ruiz, J. F., Martinelli, F., Zamarripa, S., Mitton, N., Staddon, E., Papastergiou, S., Karantjias, A., Hatzikou, M., Karagiorgou, S., Athanatos, M., Papadogiannaki, E., Kontakis, K., Spanoudakis, G. and Karypidis, P.- A., "CyberSANE Project. Deliverable 5.1. Prevention and Response to Advanced Threats". Ed. by Konstantinos Kontakis and Georgios Spanoudakis. Research Report. June 2020. URL: https://www.cybersane-project.eu/deliverables/#D5.1

- Staddon, E., Mitton and N., Loscri, V., "CyberSANE Project. Deliverable 11.2. Initial Report on Dissemination and Communication Activities". Ed. by Edward Staddon, Nathalie Mitton and Valeria Loscri. Research Report. Aug. 2020. URL: https://www.cybersane-project.eu/deliverables/#D11.2
- Staddon, E., Mitton, N., Loscri, V., Vezakis, I. and Danilatou, V., "CyberSANE Project. Deliverable 11.4. Intermediate Report on Dissemination and Communication Activities". Ed. by Edward Staddon, Nathalie Mitton and Valeria Loscri. Research Report. Aug. 2021. URL: https://www.cybersane-project.eu/deliverables/#D11.4
- Ribeiro, L. L., Gallego Puyol, B., Staddon, E., Mitton, N., Loscri, V., Athanatos, M., Papadogiannaki, E., Georgopoulos, K., Corrêa, A. M., Yasar, B., O Connor, D. and Bordianu, R., "CyberSANE Project. Deliverable 11.6. Final Report on Dissemination and Communication Activities". Ed. by Edward Staddon. Research Report. Aug. 2022. URL: https://www.cybersane-project.eu/deliverables/#D11.6

Security in Wireless Multi-Hop Networks

2

In this chapter, we present a general overview of the context in which this thesis resides, in particular regarding the security of wireless multi-hop networks, relative to CIs. We commence evaluating the ever evolving threats against Wireless CIs, including the different methods to organise and categorise them for defence purposes. We continue with this train of thought, explaining the different methodologies for defending against various cyber attacks against CIs as well as IoT devices. Finally, we conclude this overview with a look into the different methods to protect CIs and IoT devices as well as learn from past events to increase the protection methods in the future.

2.1 Threats in Wireless Critical Infrastructures

As presented previously, IoT devices are being used in more domains due to their versatility and easy access for any sector or person. As a result, many existing devices are seeing new functionalities, such as smart fridges interfacing with shopping lists, autonomous cars, or smart healthcare devices which regulate dosage and medical alerts. However, with their incorporation means further steps to ensure their protection are needed, securing both users and infrastructures alike.

In this section, we present the ever evolving cyber space, delving into detail into the different stages making up an effective and deadly cyber attack. We also provide an overview of attack categorisation methods used in the literature and by security professionals in defence systems. Finally, we present the new areas of cyber threats, where portable IoT devices coupled with intelligent systems can pave to way to more adaptable, efficient and even deadly attackers.

2.1.1 An Ever Evolving Cyber-Space

A "kill chain" (originally used as a military concept related to the structure of a military strike) consists of multiple stages including target identification, strike force dispatch, attack plan and then finally the attack itself. Although this term is not universally accepted, the cyber kill chain model has seen some use in the area of IT security. This section describes the different steps in a cyber kill chain and how they can be explored to identify, detect and counter various cyber attacks.

As stated previously, the notion of *cyber attacks* is generally used to present an aggressive act towards a computer or an electronic device. However, the term represents much more than the attack itself. In [18], it is mentioned that cyber attacks are a grouping of five distinct steps, each with their own independent objectives towards the successful completion of an attack. Since these stages are critical to the success of a cyber attack, any defensive barrier erected against any single stage will cause a disruption in the attackers efforts. However, whether the system is

[18]: Linyuan Zhang, Guoru Ding, Qihui Wu, Yulong Zou, Zhu Han, and Jinlong Wang. 'Byzantine Attack and Defense in Cognitive Radio Networks: A Survey'. In: *IEEE Communications Surveys & Tutorials* 17 (2015). DOI: 10.1109/COMST.2015.2422735

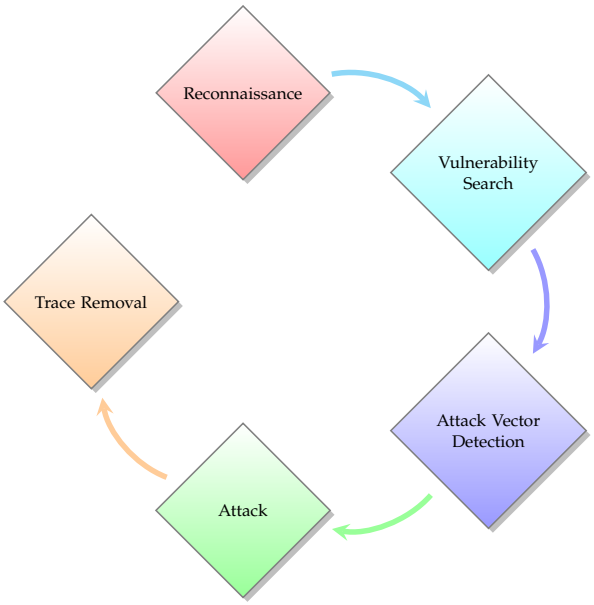


Figure 2.1: The different steps of a Cyber Attack, as explained in [18]

IoT-based or employs specific network protocols, these five attack steps remain somewhat the same, varying slightly dependant on the hardware or network specifications. The five categories are presented in Figure 2.1 and are detailed below.

- **Reconnaissance**: Similar to its military cousin, *Reconnaissance* is the act of gathering information regarding a specific target [19]. In a war zone scenario, soldiers will aim to discover the layout of the target environment, as well as the different infrastructures and vehicles possessed by the enemy but also discover critical targets, increasing their efficiency. In the cyber-space, the discovery of the network topology, the different software solutions and OSs used by the target, as well as the type of device itself grants the attacker the upper hand. Data such as Internet Protocol (IP) addresses, user names as well as firewall information and even home addresses or telephone numbers can be extracted and exploited [20].
- **Vulnerability Search**: By analysing the recovered information the attacker can gain even further knowledge Indeed, they can identify known vulnerabilities from the information, including software, OS or even network or hardware weak points [21]. Since certain vulnerabilities are susceptible to specific attacks, the attackers job becomes easier. Furthermore, with similar systems employed accross multiple infrastructures, it is possible that one attack on one system is possible on another, as previously seen with the *Ransomware* attacks against two French hospitals in 2021 which took place a week apart, with a third successfully prevented [22, 23].
- **Attack Vector Detection**: From these weaknesses, the best possible attack can be determined. Indeed, the attacker needs to gain entry into the target system to cause their disruption. With the expansion of the internet, practically every device is connected to the outside world, allowing remote attacks to take place. Armed with the list of vulnerabilities, the attacker can examine their target more in detail, determing the potential defences, and choosing the optimal vector dependant on the available vulnerabilities [24].

[19]: Nicholas R. Rodofile, Kenneth Radke, and Ernest Foo. 'Framework for SCADA Cyber-Attack Dataset Creation'. In: *Proceedings of the Australasian Computer Science Week Multiconference*. 2017. DOI: 10.1145/3014812.3014883

[20]: HP. Sanghvi and MS. Dahiya. 'Cyber reconnaissance: an alarm before cyber attack'. In: *International Journal of Computer Applications* 63 (2013). DOI: 10.5120/10472-5202

[21]: NIST CSRC. *Glossary - Vulnerability Definition*. Sept. 29, 2018. URL: https://csrc.nist.gov/glossary/term/vulnerability (visited on Sept. 22, 2022)

[22]: France 24. *Cyber attacks hit two French hospitals in one week*. Feb. 16, 2021. URL: https://www.france24.com/en/europe/20210216-cyber-attacks-hit-two-french-hospitals-in-one-week (visited on Sept. 22, 2022)

[23]: N. Pontier, M. Orre, and M. Martin. 'Cyberattaque à l'hôpital de Dax : exposition des faits, conséquences et retour d'expérience'. In: *Cancer/Radiothérapie* 26 (2022). DOI: 10.1016/j.canrad.2022.06.011

[24]: Joaquín Rodríguez. *CIPSEC - Most common attack vectors over Critical Infrastructures*. Jan. 26, 2018. URL: https://www.cipsec.eu/content/most-common-attack-vector-over-critical-infrastructures (visited on Aug. 26, 2020)

- **Attack**: Armed with this arsenal of knowledge now at the disposal, it is now possible to begin the assault. There is no fixed unique methodology to undertake such an attack, since the desired outcome as well as the system specifications vary. The previously recovered information, however, allows the attacker to determine the best possible methods to inflict the desired consequences. As a result, the exact actions undertaken depend on the specific attack utilised. For example, targeting a network device with a Denial-of-Service (DoS) attack won't necessitate the same actions on the part of the attacker when compared to, for example, a virus or a worm.
- **Trace Removal**: After the work comes the cleanup, where the attacker's objective is to cover their tracks to stop them being detected. Indeed, operations on IT devices leave log traces, which can be used by cyber security specialists [25] to determine what happend. To protect themself, the attacker can manipulate the available log files, either removing the specific traces of their attack, or deleting the file all together which is easily detectable. In many cases, some devices do not employ log files making the attacker's job easier such as IoT devices, which due to their hardware constraints cannot afford to waste their limited storage on ever expanding log files.

[25]: Neda Afzaliseresht, Yuan Miao, Sandra Michalska, Qing Liu, and Hua Wang. 'From logs to Stories: Human-Centred Data Mining for Cyber Threat Intelligence'. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2966760

Here, the contributions provided as part of this thesis target the attack phase itself, by providing a method to detect routing based attacks and avoid them as much as possible. With this approach, the attack itself can be avoided, thus reducing its overall impact.

2.1.2 Threat Categorisation

To allow security professionnals to respond to cyber attacks, they must first be identified. To aid in this, many threat categorisation methods have been proposed, allowing attacks to be categorised dependant on a specific common criteria. With this information, it is possible to analyse the attacker's strategies and design new dynamic counter actions to either identify system vulnerabilities in advance and fix them, or to quickly detect an attack underway and recover as soon as possible. However, since there are multiple types of criteria which can be used for categorisation, the choice is dependant on the intended use, but also the types of attacks at hand. For example, network-based attacks will not be categorised the same way as physical access to a device, since the conditions as well as the type of interactions are completely different.

Overall, we have documented eight distinct categorical methodologies, each with different approaches in structuring their attacks. Table 2.1 presents an overview of these categorisation approaches. For more information regarding the different threats, Appendix A contains an overview of each type presented in the different approaches hereafter.

National cyber emergency: An attack which causes sustained disruption of essential services or affects national security, leading to severe economic or social consequences or to loss of life.
Highly significant incident: An attack which has a serious impact on central government, essential services, a large proportion of the population, or the economy.
Significant incident: An attack which has a serious impact on a large organisation, or on either wider or local government, or which poses a considerable risk to central government or essential services.
Substantial incident: An attack which has a serious impact on a medium-sized organisation, or which poses a considerable risk to a large organisation or even wider or local government.
Moderate incident: An attack on a small organisation, or which poses a considerable risk to medium-sized organisations, or preliminary indications of cyber activity against a large organisaion or the government.
Localised incident: An attack on an individual or preliminary indications of cyber activity against a small or medium-sized organisation.

[26]: National Cyber Security Centre UK. *New Cyber Attack categorisation system to improve UK response to incidents*. Apr. 11, 2018. URL: https://www.ncsc.gov.uk/news/new-cyber-attack-categorisation-system-improve-uk-response-incidents (visited on Sept. 22, 2022)

Attack Severity

Probably the most basic categorisation method used in cyber security is the separation based upon the severity of the attack. In [26], the UK National Cyber Security Centre proposes an attack categorisation based

Table 2.1: Overview of the eight exiting threat categorisation methodologies and their types

Categorisation	Description
Attack Severity	Organised dependant on the severity of the attack or the threat level
Access Type	Organised dependant upon the type of access used by the attack
Attack Type	Organised dependant on the type overall type of attack
Attacker Position	Organised dependant on the attackers position relative to the victim
Attacker Implication	Organised dependant on the interaction between attacker and victim
Objective Oriented	Organised dependant on the overall goal of the specific attack
Network Layer Oriented	Organised dependant on the OSI layer where the attack resides
Use-Case Specific	Organised dependant on the specific use case

on six threat levels, from the lowest being a localised incident, to the highest corresponding to a national cyber emergency. This categorisation, however, is heavily dependant on both the type of environment and systems available, but also severely influenced by the organism designing such a proposition. Indeed, depending on the environment concerned, various attacks would cause more or less disruption.

Access Type

Another most basic form of attack categorisation is the separation based upon the type of access exploited. In this case, the type of access is defined as the basic interaction methods used to commit the attack. In [27], the authors define cyber attacks in their taxonomy as pertaining to two categories:

- **Physical**: physical interaction with an IT system.
  i.e., *tampering*, *hardware damage*, *physical access* to restricted information, etc.
- **Cyber**: threat or assault from a digital source.
  i.e., *malware*, *remote access*, etc.

An example of a *Physical* attack is the assault on California Power Station by a Sniper in April 2013 [28]. In our work, we turn towards cyber threats, pertaining to network-based intrusions.

Attack Type

Another categorisation is based around the type of attack itself. Indeed, there are many different attack methodologies that exist, however, multiple approaches concern the same type of attack. Here we have identified five different approaches to categorise by attack type.

- **DoS, Probing, Remote-to-Local (R2L), User-to-Root (U2R)**
  This approach is used in the presentation of different cyber attack detection strategies in [29], where they present the different types of cyber attacks that they consider. In [30] and [31], these categories are used to define attacks and traffic anomalies for use in attack and anomaly detection systems. This approach can be adapted to various types of networks with their specific limitations and characteristics. For example, in [32] the authors apply this categorisation to deep learning based attack detection on IoT based Fog computing whereas in [33], it is used to explain how Brute Force Attacks occur. Furthermore, it is also used in [34] during the

[27]: Jairo Giraldo, Esha Sarkar, Alvaro A. Cardenas, Michail Maniatakos, and Murat Kantarcioglu. 'Security and Privacy in Cyber-Physical Systems: A Survey of Surveys'. In: *IEEE Design & Test* 34 (2017). DOI: 10.1109/MDAT.2017.2709310

[28]: Rebecca Smith. *Assault on California Power Station Raises Alarm on Potential for Terrorism*. Feb. 4, 2014. URL: https://www.wsj.com/articles/assault-on-california-power-station-raises-alarm-on-potential-for-terrorism-1391570879 (visited on Sept. 23, 2022)

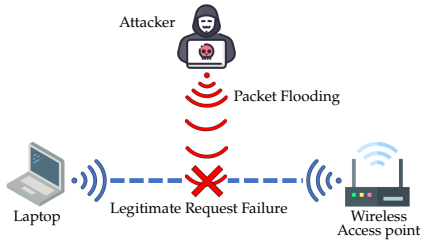


Figure 2.2: Illustration of the principal behind a DoS attack

[29]: Jamal Raiyn. 'A survey of cyber attack detection strategies'. In: *International Journal of Security and Its Applications* 8 (2014). DOI: 10.14257/ijsia.2014.8.1.23

[30]: Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 'Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges'. In: *Cybersecurity* 2 (2019). DOI: 10.1186/s42400-019-0038-7

[31]: Mohiuddin Ahmed. 'Intelligent Big Data Summarization for Rare Anomaly Detection'. In: *IEEE Access* 7 (2019). DOI: 10.1109/ACCESS.2019.2918364

[32]: Abebe Abeshu Diro and Naveen Chilamkurti. 'Distributed attack detection scheme using deep learning approach for Internet of Things'. In: *Future Generation Computer Systems* 82 (2018). DOI: 10.1016/j.future.2017.08.043

[33]: Deris Stiawan, Mohd. Yazid Idris, Reza Firsandaya Malik, Siti Nurmaini, Nizar Alsharif, and Rahmat Budiarto. 'Investigating Brute Force Attack Patterns in IoT Network'. In: *Journal of Electrical and Computer Engineering* 2019 (2019). DOI: 10.1155/2019/4568368

analysis of Machine Learning based network intrusion detection classifiers.

- **DoS, Man-in-the-Middle (MitM), Brute Force**
  This categorisation is presented in [35] to classify the most common recurring cyber attacks.
- **DoS, Replay, Deception**
  These three categories are used in [36] and [37] to categorise cyber-threats towards industrial Cyber Physical System (CPSs).
- **Active Eavesdropping, Scanning and Probing, Code Injection**
  This three-way categorisation is used in [38] to organise multiple stealth attacks against Critical Information Infrastructures.
- **Physical, Network, Software, Encryption**
  This approach is used in [39] to present the various security attacks possible against IoT networks and devices.

Here, we interest ourselves with routing-based attacks targeting data in transit throughout wireless ad hoc environments. Furthermore, we do not interest ourselves with how the malicious nodes appeared, either through deployment of false nodes or the infection of existing devices, through various methods such as tampering or R2L based attacks.

Attacker position

Another method for categorisation can be based upon the position of the attacker relative to the target system. As presented in [40], this can be reduced to two states: *Outside* and *Inside*.

- **Outside**: attack origin outside of the target network and infrastructure. These are generally network based, such as *eavesdropping* or *DoS* attacks.
- **Inside**: attack origin within the confines of the affected network and infrastructure. This includes attacks such as *compromised node* or *forced authentication*.

In [41], the authors specify that the attacker position can influence the methods used in various attacks. This approach is also used partly in [7], [42] and [43] to categorise attacks on IoT networks as well as Mobile Ad Hoc Networks (MANETs) and Software Defined Networks (SDNs), which are explained in more detail in Section 2.1.2.

As we stated, we are interested with routing based attacks, where malicious parties have integrated themselves into the network, making them inside attacks.

Attacker implication

Another common method for attack differentiation is the notion of attacker implication. This approach defines the level of interaction between the attackers themselves and the target system. We talk, therefore, of *Active* or *Passive* attacks.

- **Active**: the attacker is physically invested in their attack. Such attacks include *data tampering* or *DoS* [44, 45].

[34]: Ahmed Mahfouz, Deepak Venugopal, and Sajjan Shiva. 'Comparative Analysis of ML Classifiers for Network Intrusion Detection'. In: *Fourth International Congress on Information and Communication Technology*. 2020. DOI: 10.1007/978-981-32-9343-4_16

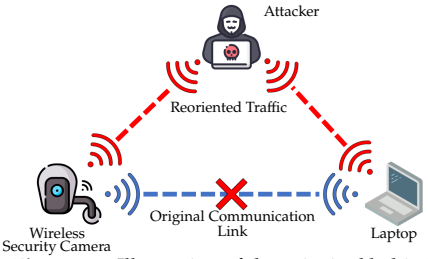


Figure 2.3: Illustration of the principal behind a MitM attack

[35]: Gurminder Kaur Jaideep Singh Simarpreet Kaur and Goldendeep Kaur. 'A Detailed Survey and Classification of Commonly Recurring Cyber Attacks'. In: *International Journal of Computer Applications* 141 (2016). DOI: 10.5120/ijca2016909811

[36]: Derui Ding, Qing-Long Han, Yang Xiang, Xiaohua Ge, and Xian-Ming Zhang. 'A survey on security control and attack detection for industrial cyber-physical systems'. In: *Neurocomputing* 275 (2018). DOI: 10.1016/j.neucom.2017.10.009

[37]: Magdi S. Mahmoud, Mutaz M. Hamdan, and Uthman A. Baroudi. 'Modeling and control of Cyber-Physical Systems subject to cyber attacks: A survey of recent advances and challenges'. In: *Neurocomputing* 338 (2019). DOI: 10.1016/j.neucom.2019.01.099

[38]: Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. 'Cyber Stealth Attacks in Critical Information Infrastructures'. In: *IEEE Systems Journal* 12 (2018). DOI: 10.1109/JSYST.2015.2487684

[39]: Jyoti Deogirikar and Amarsinh Vidhate. 'Security attacks in IoT: A survey'. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2017. DOI: 10.1109/I-SMAC.2017.8058363

[40]: Kahina Chelli. 'Security Issues in Wireless Sensor Networks: Attacks and Countermeasures'. In: *Proceedings of the World Congress on Engineering*. 2015. URL: http://www.iaeng.org/publication/WCE2015/WCE2015_pp519-524.pdf (visited on Oct. 17, 2022)

[41]: Inria. *Cybersecurity: Current challenges and Inria's research directions*. Tech. rep. Inria, 2019. URL: https://www.inria.fr/en/white-paper-inria-cybersecurity (visited on Oct. 17, 2022)

[7]: Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. 'Internet of things (IoT) security: Current status, challenges and prospective measures'. In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. DOI: 10.1109/ICITST.2015.7412116

[42]: Priyanka Goyal, Vinti Parmar, Rahul Rishi, et al. 'Manet: vulnerabilities, challenges, attacks, application'. In: *IJCEM International Journal of Computational Engineering & Management* 11 (2011). URL: http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.472.310&rep=rep1&type=pdf (visited on Oct. 17, 2022)

[43]: Olivier Flauzac, Carlos González, Abdelhak Hachani, and Florent Nolot. 'SDN Based Architecture for IoT and Improvement of the Security'. In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. 2015. DOI: 10.1109/WAINA.2015.110

[44]: Furrakh Shahzad, Maruf Pasha, and Arslan Ahmad. 'A Survey of Active Attacks on Wireless Sensor Networks and their Countermeasures'. In: *CoRR* abs/1702.07136 (2017). DOI: 10.48550/arXiv.1702.07136

[45]: Khaleel Ahmad. 'Classification of Internet Security Attacks'. In: *Proceeding of the 5th National Conference INDIACom-2011Bharti Vidyapeeth's Institute of Computer Applications and Management, New Delhi ISSN*. 2011. URL: https://www.researchgate.net/profile/Khaleel-Ahmad-3/publication/262494946_Classification_of_Internet_Security_Attacks/links/0a85e537def4c0c903000000/Classification-of-Internet-Security-Attacks.pdf (visited on Oct. 17, 2022)

[29]: Jamal Raiyn. 'A survey of cyber attack detection strategies'. In: *International Journal of Security and Its Applications* 8 (2014). DOI: 10.14257/ijsia.2014.8.1.23

[40]: Kahina Chelli. 'Security Issues in Wireless Sensor Networks: Attacks and Countermeasures'. In: *Proceedings of the World Congress on Engineering*. 2015. URL: http://www.iaeng.org/publication/WCE2015/WCE2015_pp519-524.pdf (visited on Oct. 17, 2022)

[27]: Jairo Giraldo, Esha Sarkar, Alvaro A. Cardenas, Michail Maniatakos, and Murat Kantarcioglu. 'Security and Privacy in Cyber-Physical Systems: A Survey of Surveys'. In: *IEEE Design & Test* 34 (2017). DOI: 10.1109/MDAT.2017.2709310

[46]: Noah J. Apthorpe, Dillon Reisman, Srikanth Sundaresan, Arvind Narayanan, and Nick Feamster. 'Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic'. In: *CoRR* abs/1708.05044 (2017). DOI: 10.48550/arXiv.1708.05044

[38]: Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. 'Cyber Stealth Attacks in Critical Information Infrastructures'. In: *IEEE Systems Journal* 12 (2018). DOI: 10.1109/JSYST.2015.2487684

[47]: Katharina Krombholz, Heidelinde Hobel, Markus Huber, and Edgar Weippl. 'Advanced social engineering attacks'. In: *Journal of Information Security and Applications* 22 (2015). DOI: 10.1016/j.jisa.2014.09.005

[48]: Emma J. Williams, Joanne Hinds, and Adam N. Joinson. 'Exploring susceptibility to phishing in the workplace'. In: *International Journal of Human-Computer Studies* 120 (2018). DOI: 10.1016/j.ijhcs.2018.06.004

- **Passive**: the attacker has no influence on the attack itself. These attacks are generally considered stealthy and are difficult to detect on their own. This includes threats such as *eavesdropping* or *traffic analysis* where information is recovered from observations, such as spying, and not direct interaction corresponding to a breach of privacy [29, 40]

Once more, since we concern ourselves with routing based attacks, we turn our attention towards active attacks, since attacks take an active stance within their networks.

Objective oriented

A different method for attack categorisation is the organisation based upon the objective of the specific attack. In total, we have identified six objective oriented approaches to attack categorisation.

- **Privacy**
  This categorisation is used in [27] to enumerate attacks which recover private information through various spying techniques.These attacks are even more significant dependant on the environment in which they reside, for example in an IoT network, such attacks can recover large quantities of data for analysis and exploitation [46].
- **Reconnaissance**, **Access**, **Malicious**, **Non-Malicious**, **Cyber Crime**, **Cyber Espionage**, **Cyber Terrorism**, **Cyber War**
  These categories are used in [29] to complement their existing methods.
- **Disconnection and Goodput Reduction**, **Side-Channel Exploitation**, **Covert-Channel Exploitation**
  [38] utilise these three categories to complement their existing approaches.
- **Hardware**, **Network**, **Human Factor**
  These categories complement the position of the attacker in [41], allowing to increase the categorisation potential. Some examples of attacks targetting the *Human Factor* are *Social Engineering*, tricking the user into compromising their system [47]; and *Phishing*, attempting to extract important information from a user, generally through email communications [48].
- **Interception**, **Interruption**, **Fabrication**, **Modification**
  [45] extend their previous categorisation using the attacker's implication with these four extra categories.
- **Access Control**, **Authentication**, **Availability**, **Confidentiality**, **Integrity**
  This categorisation is used in [49] to categorise and present various different security threats in wireless networks.

Here, we interest ourselves with attacks which can impact data being routed. This concerns both data and packet privacy as well as data throughput.

Network layer oriented

Network-based attacks can be categorised dependant on the different network layers of the Open Systems Interconnection (OSI) model [50]. This grants the possibility to associate specific attacks to the layer on which their primary impact takes place, such as *Jamming* [51] on the physical layer or a *DoS User Datagram Protocol (UDP)* flood [52] on the transport layer.

In practice, attacks are generally categorised on the first three layers (Physical to Network) as well as the seventh (Application). Since the OSI model is the inspiration behind the physical and protocol structure of IP based networks, these four layers can be used to separate the different threats towards different systems types, such as Wireless Sensor Networks (WSNs) [40] or Wireless Body Area Networks (WBANs) [53].

Here, our association is inherently transparent, where our routing based attacks take olace on the network layer itself.

Use case specific

The final method for categorisation is based upon the use case in which the attacks are taking place. This approach is, therefore, dependant on multiple factors, including the choice of hardware, software, the network paradigm, user interaction and most importantly the service provided. An overview of these categorisation methods by use case is presented in Figure 2.4. Here, we interest ourselves in-particular with Ah Hoc approaches, which are more in line with the context of this thesis. The other categories are available in Appendix A.

Wireless Ad-Hoc Networks

Wireless Ad-hoc Networks utilise the multi-hop network premice to allow the exchange of data, without the need for a direct link to a central infrastructure. This is the case for WSNs where IoT devices are utilised to capture sensory data, which is then relayed back to a sink-node, connected to the internet [54]. In some cases, the devices can become mobile, causing the need for these devices to be able to adapt and explore their surroundings. These networks are called Mobile Ad Hoc Networks (MANETs) and are an area of research in constant expansion.

With the need to relay information comes the risk of data loss due to malicious relay nodes. As a result, routing based attacks can have a significant impact as they can cause large scale disruptions during forwarding. There are three main types of routing based attacks in this situation:

- **Black-hole Attack**, which influences routing decisions to force all messages to transit to the compromised node itself to then be dropped, resulting in a DoS of varying intensity [55].
- **Wormhole Attack**, which creates an unauthorised long distance link between two compromised nodes, forwarding data from one end of the network to the other, disrupting routing efficiency as nodes on one end believe they are closer to nodes on the other end than they are [56].

[45]: Khaleel Ahmad. 'Classification of Internet Security Attacks'. In: *Proceeding of the 5th National Conference INDIACom-2011Bharti Vidyapeeth's Institute of Computer Applications and Management, New Delhi ISSN*. 2011. URL: https://www.researchgate.net/profile/Khaleel-Ahmad-3/publication/262494946_Classification_of_Internet_Security_Attacks/links/0a85e537def4c0c903000000/Classification-of-Internet-Security-Attacks.pdf (visited on Oct. 17, 2022)

[49]: Akhil Gupta and Rakesh Kumar Jha. 'Security threats of wireless networks: A survey'. In: *International Conference on Computing, Communication & Automation*. 2015. DOI: 10.1109/CCAA.2015.7148407

[50]: Glenn Surman. 'Understanding Security Using the OSI Model'. In: *SANS Institute Reading Room* (2002). URL: https://www.sans.org/white-papers/377/ (visited on Oct. 17, 2022)

[51]: Wenyuan Xu, Ke Ma, W. Trappe, and Yanyong Zhang. 'Jamming sensor networks: attack and defense strategies'. In: *IEEE Network* 20 (2006). DOI: 10.1109/MNET.2006.1637931

[52]: Samad S. Kolahi, Kiattikul Treseangrat, and Bahman Sarrafpour. 'Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13'. In: *2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)*. 2015. DOI: 10.1109/ICCSPA.2015.7081286

[40]: Kahina Chelli. 'Security Issues in Wireless Sensor Networks: Attacks and Countermeasures'. In: *Proceedings of the World Congress on Engineering*. 2015. URL: http://www.iaeng.org/publication/WCE2015/WCE2015_pp519-524.pdf (visited on Oct. 17, 2022)

[53]: Sagarika Chowdhury and Mainak Sen. 'Survey on Attacks on Wireless Body Area Network'. In: *International Journal of Computational Intelligence & IoT, Forthcoming* (2019). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3358378 (visited on Oct. 17, 2022)

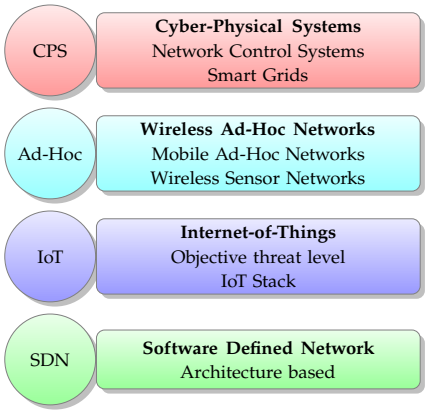


Figure 2.4: Presentation of the four methods used to categorise threats by use case

[54]: I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. 'Wireless sensor networks: a survey'. In: *Computer Networks* 38 (2002). DOI: 10.1016/S1389-1286(01)00302-4

- **Grey-hole Attack**, which functions in a similar fashion to Black-holes, except instead of dropping all passing messages, only a select few will be dropped, dependant on various metrics from random to specific message types [57].

A simple method of threat categorisation when multiple areas and domains are being evaluated is the separation based on the application used, in this case *Attacks on MANET* and *Attacks on WSN* as explained and presented in [29].

In our context, we interest ourselves with Black-hole attacks, impacting the overall routing efficiency.

Categorisation Overview

In total, 22 different approaches have been identified, spanning the eight categorisation types. Furthermore, with the expansion of IoT-based systems, more and more approaches are likely to emerge specific to the use case at hand. An overview of all previous approahces is presented in Table 2.2.

We can see that, not all approaches are popular and have been adopted by multiple entities. Indeed, the six *Objective Oriented* approaches were only used once each, meaning that these are either new propositions, or are not of interest. This isalso the same for some of the *Use-Case Specific* approaches, with the second IoT being slightly more popular than the former.

However, three types do stand out above the rest. Firstly, the categories relative to *Attacker Position* and *Implication* are widely used, especially since only a single approach has been identified in these categories. Indeed, determine where the treat is coming from as well as its potential impact due to the implication of the attacker is an important strategy. The notion of *Passive* and *Active* attacks has been extensively used in the literature, however, it is generally subservient to another categorisation method.

By combining the *Attacker's Implication* with other approaches, it can serve as a specialisation for attacks themselves, identifying more in detail the specific impact. We can see that this is the case of [7], which couples both these categories to increase the precision of their *Network*-based approach. Finally, we can conclude that the most common approach is based on the *Attack Type*, in particular *DoS*, *Probing*, *R2L*, *U2R*, which once again has been used in conjunction with the *Attacker's Implication* in [29].

2.1.3 Incorporation of Smart Systems

With the widespread adoption of machine learning techniques in IoT security, defensive methods are becoming "smarter". Indeed, with Intrusion Detection Systems (IDSs) relying on these methods, such systems are capable of reacting towards the different threats they encounter. Furthermore, with the wide variety of machine learning methods available, it is possible to find the best approach for the current objective. However,

[55]: Latha Tamilselvan and V. Sankaranarayanan. 'Prevention of Blackhole Attack in MANET'. In: *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*. 2007. DOI: 10.1109/AUSWIRELESS.2007.61

[56]: Viren Mahajan, Maitreya Natu, and Adarshpal Sethi. 'Analysis of wormhole intrusion attacks in MANETS'. In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. 2008. DOI: 10.1109/MILCOM.2008.4753176

[57]: V Shanmuganathan and T Anand. 'A survey on gray hole attack in manet'. In: *IRACST–International Journal of Computer Networks and Wireless Communications (IJCNWC)* 2 (2012). URL: https://www.academia.edu/download/45795271/2vol2no6.pdf (visited on Oct. 17, 2022)

[29]: Jamal Raiyn. 'A survey of cyber attack detection strategies'. In: *International Journal of Security and Its Applications* 8 (2014). DOI: 10.14257/ijsia.2014.8.1.23

[7]: Rwan Mahmoud, Tasneem Yousuf, Fadi Aloul, and Imran Zualkernan. 'Internet of things (IoT) security: Current status, challenges and prospective measures'. In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. DOI: 10.1109/ICITST.2015.7412116

Table 2.2: Detailed overview of all previously presented categorisation methods, organised by type and approach

Categorisation		Approach	References
Attack Severity		6 threat levels: Localised, Moderate, Substantial, Significant, Highly Significant & National Cyber Emergency	[26]
Access Type		Physical, Cyber	[27]
Attack Type		DoS, Probing, R2L, U2R	[29–34, 62]
		DoS, MitM, Brute Force	[35, 63]
		DoS, Replay, Deception	[36, 37]
		Active Eavesdropping, Scanning and Probing, Code Injection	[38]
		Physical, Network, Software, Encryption	[39]
Attacker Position		Outside, Inside	[7, 40–43]
Attacker Implication		Active, Passive	[7, 29, 40, 44, 45, 57]
Objective Oriented		Privacy	[27]
		Access, Malicious, Non-Malicious, Cyber Crime, Cyber Espionage, Cyber Terrorism, Cyber War	[29]
		Disconnection and Goodput Reduction, Side-Channel Exploitation, Covert-Channel Exploitation	[38]
		Hardware, Network, Human Factor	[41]
		Interception, Interruption, Fabrication, Modification	[45]
		Access Control, Authentication, Availability, Confidentiality, Integrity	[49]
Network Layer Oriented		OSI model, Layers 1 - 4 & 7	[40, 53]
Use-Case Specific	CPS - NCS	Attacks on Physical Components, Attacks on Communication Network	[64]
	CPS - Smart Grids	Power and Energy Layer, Computer/IT Layer, Communication Layer	[65]
	Wireless Ad-Hoc Networks	Attacks on MANET, Attacks on WSN	[29]
	IoT	Low-Level, Intermediate-Level, High-Level Security Issues	[66]
		IoT Stack Layers - Perception, Network, Application	[7, 67]
	SDN	SDN Architecture - Application Layer, Application-Control Interface, Control Layer, Control-Data Interface, Data Layer	[68]

machine learning methods can be categorised into multiple areas and will be explored in Section 2.2.1 with regards to IDSs. These areas have been extensively studied in the literature, paring different machine learning techniques with different defensive capabilities in order to increase the security of IT and IoT systems.

However, this is also the case on the other side of the fence. Indeed, where machine learning can be adopted on the victims side to help their defence, it can also be the same on the attacker's side, to help their attack become more efficient. In [58] the authors present an overview of these new "smart attacks", where machine learning algorithms can be used for either data analysis, allowing the attacker to gain further information about a target network. or behavioural detection, allowing the attacker to learn about the victims environment or habits. Furthermore, these methods can be used to render the attacks more efficient, such as increase the chance of a critical hit during the attack, or to reduce the energy consumption overtime allowing a mobile attacker to function for longer [59]. In the former case, these attacks themselves can also vary, concerning both passive threats, such as *eavesdropping*, as well as active threats such as *jamming attacks*, increasing the impact on the victim [60].

All this aside, although machine learning brings new tools to the table, allowing attackers and victims alike to adapt their strategies, this also provides another target for attacks. Since the objective for machine learning is to "learn" about a target element, allowing them to classify any input data, it is possible to influence this stage, causing the algorithm to provide false data. Indeed, by corrupting the data used during the training phase, the expected output can be influenced, allowing attackers to force their victims in a specific direction [61]. Thankfully, solutions exist to defend against these threats, however, they remain a reality and a potential target for malicious entities.

[58]: Emilie Bout, Valeria Loscri, and Antoine Gallais. 'How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey'. In: *IEEE Communications Surveys & Tutorials* 24 (2022). DOI: 10.1109/COMST.2021.3127267

[59]: Emilie Bout, Valeria Loscri, and Antoine Gallais. 'HARPAGON: An Energy Management Framework for Attacks in IoT Networks'. In: *IEEE Internet of Things Journal* 9 (2022). DOI: 10.1109/JIOT.2022.3172849

[60]: Emilie Bout, Alessandro Brighente, Mauro Conti, and Valeria Loscri. 'FOLPETTI: A Novel Multi-Armed Bandit Smart Attack for Wireless Networks'. In: *Proceedings of the 17th International Conference on Availability, Reliability and Security*. 2022. DOI: 10.1145/3538969.3539001

[61]: Mingfu Xue, Chengxiang Yuan, Heyi Wu, Yushu Zhang, and Weiqiang Liu. 'Machine Learning Security: Threats, Countermeasures, and Evaluations'. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2987435

2.2 Detection Methodologies

Detecting different cyber events is an active research domain. There are many different techniques for cyber-detection, both for a specific attack type or general event detection. Firstly, there are two large categories of cyber attacks where detection techniques differ: *Anomalies* and *Intrusion*. *Anomalies* are abnormal behavioural patterns detected in network traffic and internal IT systems. These patterns indicate something which has changed in the network, which could indicate either an external attack underway or even an intrusion. As its name states, in the same way as a thief can enter into a building without authorisation, an intrusion is the unauthorised access into a computer network or system, granting unrestricted access.

2.2.1 Intrusion Detection

In [30], the authors presented two types of IDS based on two different approaches. The first named Signature-based Intrusion Detection System (SIDS), is a knowledge-based detection system which uses pattern matching techniques from a database containing intrusion signatures.

[30]: Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 'Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges'. In: *Cybersecurity* 2 (2019). DOI: 10.1186/s42400-019-0038-7

It therefore compares packets in real time to known signatures, raising an alarm if an attack is detected. However, it is incapable of detecting zero-day attacks since no possible signature exists as well as multi-packet attacks due to its packet-by-packet analysis methodology. The other, Anomaly-based Intrusion Detection System (AIDS), uses multiple approaches to compare detected traffic to a "normal" traffic model and can therefore overcome the limitations of SIDS through machine-learning, statistical or knowledge-based methods. Its concept works on assuming that any deviation from the model is an anomaly, based on the assumption that malicious behaviours differ from that of typical users. With this technique, it is capable of detecting zero-day attacks and internal malicious activities such as abnormal users. Although difficult for cyber criminals to recognise, it does generate large quantities of false positives as any new activity is considered an intrusion.

The authors also explained two means of data inspection from either a host system or from network traffic. The first, Host-based Intrusion Detection System (HIDS) analyses log files, such as OS, firewall or database logs. It can, therefore, allow the detection of insider attacks with physical access to a machine. However, it is limited to machines upon which it is deployed, meaning a network wide coverage would need an HIDS on every device. On the network side, Network-based Intrusion Detection System (NIDS) extracts traffic through packet capture and is therefore capable of monitoring all devices on a certain part of the network and can also monitor external malicious activities before they spread to another system. Even with its multi-positional deployment possibilities, the inspection capabilities are highly influenced by data bandwidth as not all traffic can be examined as well as possible encryption technologies.

Evolutionary Algorithms

The first of the many different computational analyses is the notion of Evolutionary Algorithms. In [69], the authors presented the effectiveness of this approach in relation to anomaly detection. Evolutionary calculations, also known as *Genetic Algorithms*, use natural selection to optimise the analysis for a specific task. Created from training data, the overall results show good performance against various intrusions, however, a more efficient heuristic for chromosomal fitness would prove very valuable for detection techniques. In conjunction with Support-Vector Machine (SVM), an average detection rate of 99% can be achieved compared to other approaches, even though fully labelled input data is needed for training. It is also possible to evolve statistical rule-sets, evaluating each rule using a fitness function after each one is evolved with statistical continuous-valued input data. This allows the algorithm to ignore labelled elements in packet headers, as well as keeping the rule-sets small and efficient for detecting attacks.

[69]: Ayei E Ibor, Florence A Oladeji, and Olusoji B Okunoye. 'A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention'. In: *International Journal of Security and its Applications* 12 (2018). DOI: 10.14257/ijsia.2018.12.4.02

Machine Learning

Mentioned previously, various Machine Learning techniques can be used for both attacks and attack detection. These techniques need to be trained before they are capable of correctly differentiating target data. However,

there are different methods for how classification techniques learn, four of which are presented in [69]. Once these techniques have been correctly trained, they can be implemented, a few of which are demonstrated in [62].

[69]: Ayei E Ibor, Florence A Oladeji, and Olusoji B Okunoye. 'A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention'. In: *International Journal of Security and its Applications* 12 (2018). DOI: 10.14257/ijsia.2018.12.4.02

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

Learning Types

The first of the four paradigms presented in [69] is *Supervised Learning*. *Supervised Learning* necessitates the availability of pre-labelled training data, thus teaching the algorithm to differentiate between different categories using pattern recognition. This is the most common type of learning for classification purposes, thus multiple different algorithms exist, from SVM to Artificial Neural Networks.

The second is the complete opposite of the previous. *Unsupervised Learning* uses the same pattern recognition approach as previously, but uses unlabelled training data to build its own view of the overall problem. The data is first filtered to separate normal data before they are clustered in a specific number of groups, ending with a generated model from the resulting data. However, this approach contains a significant drawback, being the high computational requirements necessary for the classification, which is not always feasible dependent on the available hardware.

The third approach is a mix of the previous two. The *Semi-Supervised Learning* technique uses both labelled and unlabelled training data. It first models normal behaviour from a pre-labelled dataset, thus achieving the same view as the *Supervised* technique. It then uses an iterative process from the unlabelled data to reduce false alarm rate using similarity distances and dispersion rate of the initial result. The resulting probabilistic model produces overall good results with a high detection rate and low false positives.

Finally, we have the *Reinforcement Learning* type. As its name states, the model is trained through interaction with the surrounding environment. This allows the pattern recognition model to be specific to the platform on which the interactions with the environment have been taken, being rewarded for the best actions taken. It is suitable for solving sequential problems using a Markov decision process model. Using fuzzy Q-learning techniques, the algorithm can self-learn based on past attacks.

Implementations

Some of the possible algorithms used for attack detection are presented in [69] and [62]. The first is SVM which can be used to detect anomalous events. It derives hyperplanes from training data which maximises the separation margin between opposing classes. Although it is a *Supervised* technique, it can be adapted for *Unsupervised* functionality as well as *Semi-Supervised*. It can be used to monitor modifications and query's such as queries to Windows registry where deviation from normal registry access is considered anomalous. Using averaging techniques, SVM can ignore noise thus making the decision surface smoother. Since this approach reduces the number of support vectors, there is a reduced runtime.

The second is the notion of *Bayesian Network*. Like the previous, this approach can be used with either *Supervised* or *Unsupervised Learning*. Furthermore, it allows the modelling of domains containing uncertainty

in an efficient manner in a tree-like representation, meaning each child-node is dependent on the parent. However, the rate of false positives is elevated, due to the inability to process unusual but legitimate behaviour, such as an increase in Central Processing Unit (CPU) usage, or the need to aggregate different outputs for normality deduction when using probability analysis. An example of a *Bayesian Network* is the *Naive Bayes Classifier* which uses the Supervised Learning technique.

Another common occurrence is that of *Neural Networks*. Although they possess high computational requirements, their efficiency at resolving complex problems explain their use in many domains including image processing and cyber security defence systems. They can also be merged with other techniques from *Machine Learning* to *Statistical Analysis*. This combination can be used to create a hierarchical IDS using K-Nearest Neighbors (KNN) classifier. It is also possible to classify network traffic in real time, where different attacks correspond to different sets of artificial neurons which cover various sizes of area on the neuron map.

The last presented algorithm is the *Rule-Based* approach, which uses *Supervised* Learning. It learns the "normal" behaviour of a system and categorises any abnormality as a malicious anomaly. It is possible to train a *Rule-Base*d technique using single or multi-label learning algorithms, the latter being correlated with fuzzy clustering. It can therefore perform in-depth protocol analysis as well as deep packet inspection.

Statistical Analysis

Attack detection can also be achieved by using various statistical theories to analyse collected data. The authors of [62] explain this approach through two examples extracted from scientific literature. The first, *chi-square theory* is used for anomaly detection by creating a profile containing only normal events, thus any deviation from these events is deemed anomalous and a possible intrusion. The second is a processing unit capable of detecting rare attacks through *network traffic analysis*. The developed metric searches automatically for identical characteristics of different service requests, attributing each request an anomaly score. These scores are calculated based on the type of request, the length as well as the payload's distribution, raising an alarm once a customised threshold has been met. Many different types of techniques have been created for anomaly detection based on various statistical principals.

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

Mixture Model

The first type presented by in [62] is based on the concept that anomalies lie within large numbers of normal elements. Mixture Models possess elements which fall into two classes: possessing a small probability of δ or with a majority of elements with the probability $1-\delta$. An implementation example of this concept is anomaly detection from noisy data where the assumption resides on a set of system calls with a probability of $1-\delta$ being legitimate system use whereas intrusions possess a probability of δ.

Signal Processing Technique

Another type of statistical processing is Signal Processing Technique. An example of this technique is based on the detection of sudden changes

where anomalies are described as either corresponding to network failures and performance problems, or encompassing security-related issues such as DoS attacks. Network health functions are generated which raise alarms when anomalies are detected and are given a degree of abnormality normalised between 0 and 1. This technique is, however, a domain which has hardly been explored.

Principal Component Analysis (PCA)

Principal Component Analysis (PCA) is an easier method for the analysis of high dimensional network traffic datasets, using linear combinations of a number of random variables. It can be characterised as uncorrelated, with high to low sorted variance or with a total variance equal the that of the original data. The benefits of such an approach are that it is free from the assumption of statistical distribution while being able to reduce dimensions of data without the loss of important information. It also possesses a minimal computational complexity which allows it to be used in real-time anomaly detection.

Information Theory

Using Information-Theoretic measurements, it is possible to create appropriate anomaly detection models. Presented in [62], these models use five measurements to explain dataset characteristics:

- **Entropy**: This is the basic concept of Information Theory, which measures the uncertainty of a collection of data items.
- **Conditional Entropy**: The entropy measurement of the dataset given the entropy of the probability distribution.
- **Relative Entropy**: The entropy calculation between two probability distributions defined over the same class.
- **Relative Conditional Entropy**: The measured entropy of the dataset given the entropy between two probability distributions defined over the same class.
- **Information Gain**: Measurement of the information gain of an attribute or feature in the dataset.

From these measurement techniques, appropriate anomaly detection models can be designed Using a supervised anomaly detection approach, these measurements are used to validate new models to determine if they are suitable for testing a new dataset. nformation-Theoretic measurements have been determined to have increased performance, thus being suitable to create efficient anomaly detection models.

Using correlation analysis, it is possible to create different functionalities. Nonlinear Correlation Coefficient (NCC) similarity measurement is used to detect malicious network behaviours by extracting both linear and non-linear correlative information between network traffic. Where network traffic is concerned, data sets exist which possess non-linear correlations. Another measurement is Multivariate Correlation Analysis (MCA) which accurately characterises network traffic through the extraction of geometric correlations between network traffic features. Used mainly for DoS detection, this approach uses images converted from characterised network traffic instances. This is done through dissimilarity measurements, such as Earth Mover's Distance (EMD) which considers

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

cross-bin marching and also provides an accurate evaluation on the dissimilarity between different distributions.

Clustering

Another algorithm used for anomaly detection is the notion of Clustering. Clustering-Based detection uses unsupervised learning technology to differentiate between data points. The three main assumptions of this method are that:

- the clusters are formed using normal data and any new inputs not corresponding with existing clusters are deemed anomalous. However, when a cluster is formed with both normal and anomalous data,
- the normal data is considered to lie close to the clusters centroid whereas anomalies reside further away, making them detectable using a distance score,
- Since clusters vary in size, any whose size is smaller than a given threshold is considered anomalous leaving the thicker clusters to be considered as normal.

Many different types of clustering techniques are possible, however only two are presented in [62].

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

Regular Clustering

The first and most common clustering technique is Regular Clustering. Using various data processing methodologies, the main standing point of this technique is that clusters are generated from rows extracted from the training dataset. A few existing implementations summarised in [62] are K-Means clustering and a distance-based anomaly detection algorithm. All techniques explored possess varying degrees of accuracy (from 57% to 80%) but with a high false positive rate of approximately 22%.

Co-Clustering

Unlike Regular Clustering, Co-Clustering uses simultaneous processing of both rows and columns of the dataset to generate two sets of clusters. It also allows the definition of clustering criteria as well as optimisation and simultaneous row or column subset retrieval from the data matrix corresponding to specified criteria. Considered a dimensional reduction technique, the simultaneous grouping of rows and columns allow the preservation of information contained in the original data. There is also a significant reduction in computational complexity compared to other algorithms used in various clustering methods. Experimental results demonstrated that this technique is beneficial in the detection of DoS attacks with an overall 95% accuracy when trained with DoS specific data compared to 75% from generic datasets.

Detection Technique Used

All CIs make use of a telecommunication network to transmit data between both critical and non-critical components of their system. The necessity of guaranteeing the proper functionality of such network along with an effective security strategy has been noted in [70]. Their

[70]: Salvatore D'Antonio, Francesco Oliviero, and Roberto Setola. 'High-Speed Intrusion Detection in Support of Critical Infrastructure Protection'. In: *Critical Information Infrastructures Security*. Ed. by Javier Lopez. 2006. DOI: 10.1007/11962977_18

approach was composed of an IDS built on top of a customizable flow monitor, capable of monitoring network traffic at different levels of granularity and discovering ongoing attempts of compromise. Imitating IDS methodologies but focusing on CI Supervisory Control and Data Acquisition (SCADA) systems, [71] explored in the near past various machine learning methodologies optimized for the detection of malicious insider actors. Their work took into consideration multiple learning methods which were evaluated with the use of the Remote Technical Unit communications dataset and involved both normal operations and instances of command data injection attack scenarios. The protection of SCADA systems through machine learning algorithms have also been addressed in [72], where a novel One Class SVM (SVM) methodology was implemented to separate outliers (attack data) from the majority (normal data). The proposed methodology was superior in both performance and threat-detection terms compared with the rest of rule-based, Artificial Neural Network (ANN), Hidden Markov Model (HMM), and SVM techniques.

In [73], Behavioural observations and Big Data analysis techniques were utilized to detect anomalies in CI environments, aiming at providing an additional layer of defence against cyber attacks. The data used for the purposes of their study was based on normal and "infected" nuclear plant simulation scenarios, so that their feature extraction and data classification results be as realistic as possible. The outcomes of their work regarding the detection capabilities of the proposed solution were satisfactory enough, compared with the already traditionally used solutions. Another study, targeted especially towards the identification of cyber attacks on CIs like a water distribution system, developed a novel algorithm capable of detecting and localizing such attacks [74]. Their proposal took advantage of a deep learning neural network architecture composed of several autoencoders, which was able to successfully identify all attacks, -as well as the compromising of data integrity- of the BATtle of the Attack Detection ALgorithms open-source dataset [75].

Over the last few years, a couple of promising proactive detection methodologies have been reported in literature, which require less computational power and outperform prior existing solutions. Among them, [76] proposed the combination of Fuzzy hashing algorithms and clustering methodologies for the efficient detection of emerging ransomware threats, while [77, 78] described extensively several reasoning approaches, data mining and pattern matching techniques for the detection of abnormal network behaviour with satisfactory results [79, 80].

Knowledge and Datasets

One important factor, as presented in [81] is that attack detection is strongly influenced by the attacker's profile, but also the profile of the detection agent. All detection techniques are based on the computational analysis of data, due to our inability to process the thousands of lines of log files and look for various anomalies. However, each analysis was defined, programmed and configured by a human, thus making it dependant on that humans' abilities. A novice hacker will leave a lot more breadcrumbs than an expert when intruding into an IT system. On the same principal, a novice defender will find it difficult to detect the novice hacker and

[71]: Justin M. Beaver, Raymond C. Borges-Hink, and Mark A. Buckner. 'An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications'. In: *2013 12th International Conference on Machine Learning and Applications*. 2013. DOI: 10.1109/ICMLA.2013.105

[72]: S. L. P. Yasakethu and J. Jiang. 'Intrusion Detection via Machine Learning for SCADA System Protection'. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013. DOI: 10.14236/ewic/ICSCSR2013.12

[73]: William Hurst, Madjid Merabti, and Paul Fergus. 'Big Data Analysis Techniques for Cyber-threat Detection in Critical Infrastructures'. In: *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. 2014. DOI: 10.1109/WAINA.2014.141

[74]: Riccardo Taormina and Stefano Galelli. 'Deep-Learning Approach to the Detection and Localization of Cyber-Physical Attacks on Water Distribution Systems'. In: *Journal of Water Resources Planning and Management* 144 (2018). DOI: 10.1061/(ASCE)WR.1943-5452.0000983

[75]: Riccardo Taormina, Stefano Galelli, Nils Ole Tippenhauer, Elad Salomons, Avi Ostfeld, Demetrios G. Eliades, Mohsen Aghashahi, Raanju Sundararajan, Mohsen Pourahmadi, M. Katherine Banks, B. M. Brentan, Enrique Campbell, G. Lima, D. Manzi, D. Ayala-Cabrera, M. Herrera, I. Montalvo, J. Izquierdo, E. Luvizotto, Sarin E. Chandy, Amin Rasekh, Zachary A. Barker, Bruce Campbell, M. Ehsan Shafiee, Marcio Giacomoni, Nikolaos Gatsis, Ahmad Taha, Ahmed A. Abokifa, Kelsey Haddad, Cynthia S. Lo, Pratim Biswas, M. Fayzul K. Pasha, Bijay Kc, Saravanakumar Lakshmanan Somasundaram, Mashor Housh, and Ziv Ohar. 'Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks'. In: *Journal of Water Resources Planning and Management* 144 (2018). DOI: 10.1061/(ASCE)WR.1943-5452.0000969

[76]: Nitin Naik, Paul Jenkins, Jonathan Gillett, Haralambos Mouratidis, Kshirasagar Naik, and Jingping Song. 'Lockout-Tagout Ransomware: A Detection Method for Ransomware using Fuzzy Hashing and Clustering'. In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2019. DOI: 10.1109/SSCI44817.2019.9003148

[77]: Dharminder Kumar and D Fet. 'Performance analysis of various data mining algorithms: A review'. In: *International Journal of Computer Applications* 32 (2011). URL: https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.259.3569&rep=rep1&type=pdf (visited on Oct. 17, 2022)

[78]: Huaglory Tianfield. 'Data mining based cyber-attack detection'. In: *System simulation technology* 13 (2017). URL: https://researchonline.gcu.ac.uk/en/publications/data-mining-based-cyber-attack-detection (visited on Oct. 17, 2022)

practically impossible to detect the expert. The expert defender however, will detect the novice attacker with fewer difficulties and can potentially catch the expert. This notion is also relevant concerning the data used to train the different detection techniques.

Datasets are very important as they are used to train these different methods to recognise "normal" traffic, allowing the system to identify any anomalies. It is therefore important that the data contained in these sets be both clean and up-to-date. In [62], the authors presented some available datasets as well as various issues. The main problem is that there are very few public datasets available since privacy is an issue due to the packets being untouched. There is also the risk that the dataset be "corrupted" due to unintended and undetected infected traffic on the network. The DARPA/KDD public datasets, considered the benchmark in network analysis, are unfortunately limited due to the presence of artefacts which influence the detection process. They were also generated on an out-of-date Solaris-bases system, thus creating many important differences with modern systems.

A more modern publicly available set called ADFA-LD12 is also presented. It contains modern attack methodologies and was generated using an Ubuntu 11.04 system from 2011. The attacks target web-based tools on a realistic fully patched server forming an acceptable simulation of the real world, making it a possible successor to the DARPA/KDD dataset. The attacks available include password brute-force on an FTP/SSH connection, a Java-based meterpreter with the TikiWiki vulnerability exploit and a C100 web-shell with PHP remote file inclusion vulnerability. Other public data sets exist and are presented more in detail in [62].

Table 2.3 shows a detailed analysis and overview of existing datasets which can be used with IDSs. In CyberSANE, datasets are used to help train the detection systems utilised in LiveNet. In contrast, our work removes this constraint where our system does not need to be trained in advance as it is capable of learning the correct hops on the fly. However, these datasets could be utilised in emulators to test our system with real traces, proving its efficiency.

2.2.2 Threat Landscape & Modelling

Since the area of cyber security is in constant expansion, it is necessary to create elaborate methods to counter these threats. These methods gave way to the definition of a means of attack classification, known as a *Threat Taxonomy*. Since many exist, it is necessary to primarily examine the area before making the step forward towards the definition of a Taxonomy, capable of responding to specifications and characteristics of CIIs.

Threat Taxonomies and Classifications

Over the last few years, the worrisome growing of cybercrime and its corresponding threat landscape denoted the necessity of establishing a common way to properly identify and countermeasure such cyber threats. The threat actors behind those attacks aim to breach the protection data mechanisms and the privacy of things setup by the security experts,

[79]: Misty Blowers and Jonathan Williams. 'Machine Learning Applied to Cyber Operations'. In: *Network Science and Cybersecurity*. Ed. by Robinson E. Pino. Springer New York, 2014. DOI: 10.1007/978-1-4614-7597-2_10

[80]: Imen Brahmi, Sadok Ben Yahia, Hamed Aouadi, and Pascal Poncelet. 'Towards a Multiagent-Based Distributed Intrusion Detection System Using Data Mining Approaches'. In: *Agents and Data Mining Interaction*. Ed. by Longbing Cao, Ana L. C. Bazzan, Andreas L. Symeonidis, Vladimir I. Gorodetsky, Gerhard Weiss, and Philip S. Yu. 2012. DOI: 10.1007/978-3-642-27609-5_12

[81]: Noam Ben-Asher and Cleotilde Gonzalez. 'Effects of cyber security knowledge on attack detection'. In: *Computers in Human Behavior* 48 (2015). DOI: 10.1016/j.chb.2015.01.039

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

Table 2.3: Overview of the different Intrusion Detection Data Sets available and their caracteristics
✓*= on request
✓e= email contact needed
✓?= commercial use with permission
❋= no information found

Data Set	Information			Data		Traffic Type		Trace Contents				Data composition			References
	Year	Public	Size	Duration	Labelled	Normal	Attack	IoT	0-Day	Network	System	Nb Records	% Normal	% Attack	
ADFA-LD	2014	✓	2.3 MB.	❋	✓	✓	✓	✗	✓	✓	✓	❋	❋	❋	[84–87]
ADFA-WD	2014	✓	29.6 MB.	❋	✓	✓	✓	✗	✓	✓	✓	1 033 233	64%	36%	[86, 87]
ADFA-WD:SAA	2014	✓	403 MB.	❋	✓	✓	✓	✗	✓	✓	✓	❋	❋	❋	[86, 87]
AWID	2015	✓*	❋	108 h.	✓	✓	✓	✗	✗	✓	✗	210 900 113	97%	3%	[88, 89]
Booters	2013	✓	250 GB.	2 d.	✗	✗	✓	✗	✗	✓	✗	❋	0%	100%	[90]
Bot-IoT	2018	✓?	69.3 GB.	❋	✓	✓	✓	✓	✓	✓	✗	72 000 000	❋	❋	[91, 92]
Botnet	2014	✓*	13.8 GB.	❋	✓	✓	✓	✗	✗	✓	✗	≈ 915 944	69%	31%	[93, 94]
CAIDA	2007	✓*	21 GB.	1 h.	✗	✗	✓	✗	✗	✓	✗	❋	0%	100%	[95, 96]
CIC-DDoS 2019	2019	✓*	❋	2 d.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[96, 97]
CIC DoS	2017	✓*	4.6 GB.	24 h.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[98, 99]
CICIDS 2017	2017	✓*	51.1 GB.	5 d.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[100, 101]
CIDDS-001	2017	✓	380 MB.	28 d.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[102, 103]
CIDDS-002	2017	✓	200 MB.	14 d.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[102–104]
CDX	2009	✓	12 GB.	4 d.	✗	✓	✓	✗	✗	✓	✗	❋	❋	❋	[105, 106]
CTU-13	2013	✓	697 GB.	143 h.	✓	✓	✓	✓	✗	✓	✗	20 643 076	98%	2%	[107, 108]
DARPA	1999	✓	❋	25 d.	✓	✓	✓	✗	✗	✓	✓	❋	❋	❋	[109, 110]
Gure KDD Cup	2008	✓*	13.6 GB.	35 d.	✓	✓	✓	✗	✗	✓	✗	2 759 494	41%	59%	[111–113]
IRSC	2015	✗	❋	❋	✓	✓	✓	✗	❋	✓	✗	❋	❋	❋	[114]
ISCX 2012	2012	✓*	84.4 GB.	7 d.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[115, 116]
ISOT	2010	✓*	420 GB.	3 mnth.	✓	✓	✓	✓	✓	✓	✓	1 675 424	97%	3%	[117, 118]
KDD CUP 99	1998	✓	743 MB.	❋	✓	✓	✓	✗	✗	✓	✓	4 898 431	20%	80%	[112, 119–121]
Kent 2016	2016	✓*	12 GB.	58 d.	✗	✓	❋	✗	✗	✓	✓	1 648 275 307	❋	❋	[122, 123]
Kyoto 2006+	2006 to 2015	✓	19.2 GB.	10 y.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[124, 125]
LBNL	2005	✓	11 GB.	4 mnth.	✗	✓	✓	✗	✗	✓	✗	❋	❋	❋	[126, 127]
NDSec-1	2016	✓	869.2 GB.	❋	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[128, 129]
NGIDS-DS	2016	✓	❋	27 h.	✓	✓	✓	✗	✗	✓	✓	90 054 160	99%	1%	[130]
NSL-KDD	1998	✓	❋	❋	✓	✓	✓	✗	✗	✓	✓	5 209 458	20%	80%	[112, 119, 120, 131]
PU-IDS	2015	❋	❋	❋	✓	✓	✓	✗	✗	✓	✓	198 904	47%	53%	[132]
PUF	2018	❋	❋	3 d.	✓	✓	✓	✗	✗	✓	✗	298 463	❋	❋	[133]
SANTA	2014	✗	❋	❋	✓	✓	✓	✗	✓	✓	✓	❋	❋	❋	[134]
SSENET-2011	2011	❋	❋	4 h.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[135]
SSENET-2014	2014	❋	❋	4 h.	✓	✓	✓	✗	✗	✓	✗	❋	❋	❋	[136]
SSHCure	2014	✓	2.5 GB.	2 mnth.	✓	✗	✓	✗	✗	✓	✓	❋	0%	100%	[137, 138]
TON_IoT	2020	✓	69.8 GB.	❋	✓	✓	✓	✓	✗	✓	✓	32 153 175	29%	79%	[139, 140]
TRAbID	2017	✓	129 GB.	8 h.	✓	✓	✓	✗	✗	✓	✓	469 442 290	93%	7%	[141, 142]
TUIDS	2012	✓e	65.2 GB.	14 d.	✓	✓	✓	✗	✗	✓	✗	833 006	52%	48%	[143]
Twente	2008	✓	❋	6 d.	✓	✗	✓	✗	✗	✓	✗	❋	0%	100%	[144, 145]
UGR'16	2016	✓	236 GB.	6 mnth.	✓	✓	✓	✗	✗	✓	✗	≈ 16.9 B.	❋	❋	[146, 147]
UNIBS-2009	2009	✓e	27 GB.	3 d.	✗	✓	✗	✗	✗	✓	✗	❋	100%	0%	[148, 149]
Unified Host and Network	2017	✓*	150 GB.	90 d.	✗	✓	✗	✗	✓	✓	✓	❋	100%	0%	[150, 151]
UNSW-NB15	2015	✓	100 GB.	31 h.	✓	✓	✓	✗	✓	✓	✓	2 540 044	87%	13%	[120, 152, 153]

aspects which are of crucial importance and one of the key challenges across all types of industries. In order to efficiently deal with the aforementioned issues, a set of tier-based classification systems known as threat taxonomies were introduced, where an attacker's intent, objectives, strategies and defences are modelled and attributed to a given threat category. A taxonomy can be defined as the practice of classification of things or concepts including the principles that underlie such classification. Thus, threat taxonomies act as a useful median for the cyber security experts of any CII, to unify different organizational concepts and respond timely following a generic pattern which is ultimately independent of their premise's structure.

Introduction to Threat Taxonomies

The specification of a threat taxonomy must follow a mutually exclusive strategy where the threats of one category cannot overlap another category -while at the same time- the mapped categories have to be exhaustive enough to cover as many actors and threats as possible. Those actors and threats compose the main threat characteristics that usually come in several variations following an Identity and Access Management approach [82]. Threat taxonomies assist in establishing basic and advanced financial and security values, by measuring the "attractiveness" of an asset as well as discovering the potential exposure of other assets which are closely related with the compromised one. The classification scheme also helps to address specific periods or circumstances that take place within a CII, where the accessibility of an attacker is greater than usual and could result to the exploitation of a vulnerability, or gaining control of a reduced access service (e.g. during the scheduled maintenance of a system's component). These types of cyber-attacks, as well as any kind of attack, are usually generalized into a higher level of threats which acts as an abstraction mechanism to further identify key concepts between a threat-actor and the potentially compromised system [83]. Thanks to this categorisation, the rapidly increasing amount of cyber-security threats is conclusively gathered into a finite only set of threats like the deliberate or unintentional attack, the failure or outage of a system, the interception or nefarious activity, etc.

The heterogeneous nature of cyber security field gave birth to several threat taxonomies from various organizations, where each one of them took into consideration its own specific needs and created a tailored version of threat classification. [154] on account of European Union Agency for Network and Security Information (ENISA) proceeded to research which was able to successfully identify and record various threats that could apply within a wide range of industries. On the other hand, [155] defined a clustering taxonomy model where each cyber security aspect is classified across a finite set of cross-cutting areas among security and privacy, laws and regulations, technologies, applications and sectors. Even though their approach is quite well-structured, it lacks concepts from the European law landscape, and the number of the generated dimensions across each cluster introduces a high fragmentation of competencies. Following a similar approach, the technical committee of the European Telecommunications Standard Institute (ETSI) developed a cyber security taxonomy based on the current industry interests like the security technologies used, as well as the tools and techniques adopted [156]. Their approach classifies the CIs into the horizontal cybersecurity

[82]: Dennis Hollingworth. 'Towards threat, attack, and vulnerability taxonomies'. In: *IFIP WG*. 2003. URL: https://pdfs.semanticscholar.org/6e48/492edd56fbb45b6958feb360e8f879d950aa.pdf (visited on Oct. 17, 2022)

[83]: Luis Marinos, Adrian Belmonte, and Evangelos Rekleitis. 'ENISA's Threat Taxonomy'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. URL: https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy (visited on Oct. 17, 2022)

[154]: Cedric Lévy-Bencheton and Eleni Darra. 'Cyber Security and Resilience of Intelligent Public Transport: Good practices and recommendations'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. URL: https://www.enisa.europa.eu/publications/good-practices-recommendations (visited on Oct. 17, 2022)

[155]: National Institute of Standards and Technology (NIST). 'NIST CSRC Taxonomy'. In: *NIST Special Publication (SP) 800-30, Revision 1, Guide for Conducting Risk Assessments*. 2020. DOI: 10.6028/NIST.SP.800-30r1

[156]: ETSI. 'CYBER; Global Cyber Security Ecosystem'. In: *European Telecommunications Standard Institute (ETSI)*. 2020. URL: https://www.etsi.org/deliver/etsi_tr/103300_103399/103306/01.02.01_60/tr_103306v010201p.pdf (visited on Oct. 17, 2022)

domain, and addresses a short glossary of definitions and components, along with the management perspective that should be followed during a security incident. The presented management process resembles the National Institute of Standards and Technology (NIST) incident response phases and involves the clusters of identification, protection, detection, response, recovering, and sharing. However, a cluster does have to necessarily being interconnected with each other, allowing in that way a CI to specialize on a single or a few only subset of those clusters.

Existing Threat Taxonomies

Threat taxonomies respond to the necessity to offer a common language for conveying IT threats that could lead to cyber-attacks or cyber-incidents of any nature. Originally, threat taxonomies and catalogues were developed as an internal tool by different organizations related to Information Communication Technologies (ICT), used in the collection and consolidation of threat information. Regrettably, in the vast field of ICTs and computer science, there are many ways to classify cyber threats, depending on many factors, so in general, existing incident taxonomies belong to either of the following groups:

- Specific taxonomies developed by individual Computer Emergency Response Teams (CERTs)
- Universal, internationally recognized taxonomies

Several national CERTs have developed their way to classify cyber threats, some just based on internet security attacks (such as the one developed by the Latvian CERT NIC.LV, consisting of eleven types of cyber attacks), based probably on the team's experiences; and other taxonomies are established according to who reported the incident, as in the case of the CERT-Hungary team, whose classification consists of just four categories (incidents reported by 1-National CII Protection (CII), 2-CIIP of partners with Service Level Agreement (SLA), 3-International partners, 4-cooperating organizations). The value of these proprietary taxonomies is that they maximize the correlation with the team's needs and expectations, but they are not universally agreed or comparable with other taxonomies.

Indeed, many different taxonomical approaches, each with their different strengths and weaknesses, such as NIST's CSRC taxonomy [155], ENISA's threat taxonomy [83], the Taxonomy of Operational Cyber Security Risks (TOCSR) proposed in [157], and the Open Threat Taxonomy (OTT) developed from Enclave Security [158]. These different approaches are adapted to many different uses since they can take different forms, such as a mind mapping tree or a classical table. Moreover, taxonomies are designed to be applicable to the system which they are protecting, hence their elaboration by that system's cyber security experts. This means that a taxonomy applied to specific system architecture with identifiable specifications and requirements, will not necessarily be exploitable on another system which could possess both different architectural decisions as well as specifications. An evaluation of such comprehensive taxonomies for information technology threats has been recently conducted in [159].

CyberSANE Threat Taxonomies

Since CyberSANE's purpose is to cover multiple CIIs and share detection information between units, multiple taxonomies are not feasible. A

[155]: National Institute of Standards and Technology (NIST). 'NIST CSRC Taxonomy'. In: *NIST Special Publication (SP) 800-30, Revision 1, Guide for Conducting Risk Assessments*. 2020. DOI: 10.6028/NIST.SP.800-30r1

[83]: Luis Marinos, Adrian Belmonte, and Evangelos Rekleitis. 'ENISA's Threat Taxonomy'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. URL: https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy (visited on Oct. 17, 2022)

[157]: James Cebula, Mary Popeck, and Lisa Young. *A Taxonomy of Operational Cyber Security Risks Version 2*. Tech. rep. CMU/SEI-2014-TN-006. Software Engineering Institute, Carnegie Mellon University, 2014. URL: http://resources.sei.cmu.edu/library/asset-view.cfm?AssetID=91013 (visited on Oct. 17, 2022)

[158]: Enclave Security. *Open Threat Taxonomy (Version 1.1)*. Mar. 26, 2016. URL: https://www.auditscripts.com/free-resources/open-threat-taxonomy/ (visited on Sept. 24, 2022)

[159]: Steven Launius. 'Evaluation of Comprehensive Taxonomies for Information Technology Threats'. In: *SANS Institute* 1 (2018). URL: https://www.sans.org/white-papers/38360/ (visited on Oct. 17, 2022)

single taxonomy was, therefore, created to cover CIIs as a single system, decomposed into multiple sectors. The resulting product was based upon the format used by ENISA [83], thus giving a solid ground to begin construction of CyberSANE's core detection taxonomy. However, ENISA's taxonomical structure possessed certain limitations regarding its categorical choices. Indeed, certain threats such as DoS or MitM attacks were categorised as single individual attacks, whereas in practice there are numerous methods to perform either attack. It was thus decided to expand upon the existing threats presented by ENISA, presented in Table 2.4 to expand into multiple subcategories, thus differentiating between the high level threat, and the threat type.

It is immediately possible to identify that certain categories, such as "Eavesdropping" and "Nefarious Activity" are vast areas, regrouping practically all cyber threats. It was here that CyberSANE's Taxonomy was able to expand upon these categories, associating more specific threat types, as presented in Table 2.5.

[83]: Luis Marinos, Adrian Belmonte, and Evangelos Rekleitis. 'ENISA's Threat Taxonomy'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. URL: https://www.enisa.europa.eu/topics/threat-risk-management/threats-and-trends/enisa-threat-landscape/threat-taxonomy (visited on Oct. 17, 2022)

Table 2.4: The different High-Level Threat Categories in the Threat Taxonomy proposed by ENISA.

High-Level Threats
Physical
Accidental
Encironmental Disaster
Failure
Outage
Eavesdropping / Interception / Hijacking
Nefarious Activity

Table 2.5: The expansion of the ENISA Threat Taxonomy utilised within CyberSANE, allowing further specification of the threat type based on the high level associated threat.

High-Level Threat	Threat Type
Eavesdropping / Interception / Hijacking	Reconnaissance
	Eavesdropping
	Man-in-the-Middle
Nefarious Activity	Denial-of-Service
	Disruption
	Side-Channel
	Transmission Control Protocol
	Routing
	Authentication
	Confidentiality
	Wireles Sensor Network
	Data Integrity / Breach
	Software
	Malware
	Equipment
	Protocol
	Information Leak / False Information

High Level Threa	Threat Type	Threat ID	Threat	Threat Description	Comments	Class Description
PHYSICAL		PHY01	Unauthorised access	Unapproved access to physical placement of device		Threats pertaining to intentional / hostile acts of human intervention on the physical manifestation of IT systems
		PHY02	Hardware theft	Removal of hardware from location without prior approval/knowledge (Server, router, smartphone, tablet, ...)		
		PHY03	Data theft	Extraction of data from systems without prior approval/knowledge (documents, backups, ...)		
		PHY04	Vandalism	Damage to devices simply to prove a point, such as a disgruntled employee		
		PHY05	Sabotage	Damage or compromise devices to render system unusable, or introducing vulnerability (unauthorised wireless AP on private device, infection, reprogram, replace hardware, control device)		
		PHY06	Tampering	Extract sensitive information from device such as crypto keys or data contained on storage media		
		PHY07	Hardware exchange	Replace hardware on device such as PCB to contain a hardware trojan		
		PHY08	Terrorist attack	Mass scale physical attack to destroy all possible equipment, rendering maximum damage		
		PHY09	AP theft	Unauthorised removal or a wireless AP or a node in a wireless sensor network, removal of important relay nodes causes larger impact		
		PHY10	Fake AP	Installation of a fake AP in proximity to legitimate one, used to lure devices away from legitimate AP, can be used to recover data		
		PHY11	Rogue AP	Installation of an unsecure and unauthorised AP in network or directly on to server system, allows unrestricted access to network data		
		PHY12	Defect	Physical defect introduced into device during fabrication, accidental or on purpose (hardware trojan), causing instability or vulnerability		

Figure 2.5: Extract of the CyberSANE Threat Taxonomy, in particular the Physical threats, where we can see the different threats, each with their own ID, description and the overall class description of the high level threat.

This identification allowed the ability to categorise threats more precisely, increasing ease of use towards adding new threats or even new threat type categories. This expanded the taxonomy into more specific areas, such as IoT-based WSN, and even listing threats towards specific exchange

protocols in control systems. The resulting taxonomy contains currently 248 listed threats spread across a total of 22 threat categories. An extract of Physical level threats can be seen in Figure 2.5.

2.3 System Protection and Threat Reduction

Where cyber attacks are concerned, there are different means of handling the consequences depending on the impact of the attack. Firstly, we will examine the notion of *Incident Handling*, as well as different *Response Techniques*. In [160], NIST presents a guide to the Cyber-Incident Handling process. This concerns all elements necessary for an IT department to adequately detect, identify, respond, protect and recover. This also includes training users and operatives in the notions of security awareness, thus reducing the risk of attacks resulting from accidental erroneous operations. In [161], Cyber-Incident Handling in a cloud context is examined where it is explained that Incident Handling is part of a larger process called Incident Management [162]. Incident Handling itself contains four steps in order to adequately respond to an incident, presented in Figure 2.6.

[160]: Paul Cichonski, Tom Millar, Tim Grance, Karen Scarfone, et al. 'Computer Security Incident Handling Guide'. In: *NIST Special Publication* 800 (2004). URL: https://rms.koenig-solutions.com/Sync_data/Trainer/QMS1784-2020417482-NIST.SP.80061r2.pdf (visited on Oct. 17, 2022)

[161]: Nurul Hidayah Ab Rahman and Kim-Kwang Raymond Choo. 'A survey of information security incident handling in the cloud'. In: *Computers & Security* 49 (2015). DOI: 10.1016/j.cose.2014.11.006

[162]: Harry W. *Getting started with cyber incident management*. Sept. 19, 2019. URL: https://www.ncsc.gov.uk/blog-post/getting-started-with-cyber-incident-management (visited on Sept. 25, 2022)

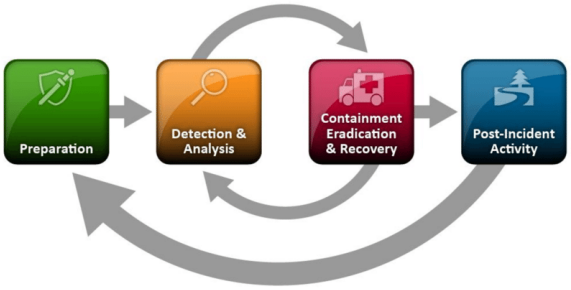


Figure 2.6: The life-cycle of the Incident Handling process.

2.3.1 Active Protection Methods

The first stage in preventing a cyber attack is to have a good defence strategy in place. This is the same as locking the front door at night, stopping another person from entering freely. However, this may not stop them directly, but would indeed slow them down and by setting up an alarm on said door, if the attacker forced through it, an alert would be raised and the police will arrive.

This is the same case with cyber systems. Indeed, since an attack can come from anywhere at anytime, it is important to be prepared and employ rigorous methods to ensure not only the systems are primed for defence, but also the personal behind it. Furthermore, detecing the intrusion is an important stage in both detecting an attack has taken place, as well as stopping it.

Preparation

The first step in Incident Handling is the *Preparation* phase. This phase is necessary to create a state of forensic readiness, thus reducing the impact of a security incident and allowing continuous functionality.

This preparation assures the system is well prepared for any unforeseen events. A popular approach to pre-incident preparation is the Information Security management, such as training and policy compliance. From a technological perspective, logical security control must be addressed. This is done by implementing certain number of protection schemes such as firewalls, vulnerability assessments or network monitoring. This is complemented by both physical and environmental protection systems. A Computer Security Incident Response Team (CSIRT) is also important, as they determine what has happened, what should be done to remedy such happenings and implement the chosen solution.

Detection and Analysis

The next phase is called *Detection and Analysis* and starts when suspicious or unusual events are detected or reported. These events can be anything from an unfamiliar file name to suspicious entries into a network system account and can originate from an automated IDS or from manual reporting. After an incident is detected, the resulting data is analysed to determine its validity and the potential impacts to core services. Risk management, containing assessment, mitigation and evaluation, is important to estimate potential damage and determine how to prioritise incident handling in multi-attack scenarios.

By providing behavioural detection and analysis capabilities to multi-hop networks, we provide the ability for CIs to observe and analyse the behaviour of IoT devices in multi-hop networks. Furthermore, by providing the network with the ability to determine "good" behaviour, we also allow devices to detect any anomalous behaviour, corresponding to a potential attack. Thus, we not only provide the possibility to determine that an attack is underway, but we can also identify the malicious device in question causing the disruption on network activities.

2.3.2 Infection Isolation

The next phase concerns the notion of Incident Response (IR) which occurs soon after the previous phase to mitigate harmful impacts in a quick and efficient way. Since no attack is the same, it is not possible to have a generalised response. The strategy must therefore be adapted from multiple possible criteria such as the potential damage or theft of resources, the necessity of service availability and the duration of the response impacting functionality. This response can be decomposed into three steps: *Containment*, *Eradication* and *Recovery*.

Containment Eradication can often be achieved at the same time, since shutting down a machine, isolating a contamination or blocking all incoming traffic can both isolate and remove a threat. The *Containment* choice is naturally dependent on the threat perceived, but other factors are also taken into account, such as the availability of the service impacted, the duration and resources needed, but also the need to preserve any evidence. This last element is significantly important as it must follow strict guidelines to be admissible in legal proceedings. Once contained, in depth *Eradication* can take place to remove any infected software

or resources before lifting returning the victim device back into the network.

Threat Quarantine

When it comes to infected devices, quick reaction is important. The quicker the reaction, the less of an impact will be perceived on the victim infrastructure and any data contained therein. Indeed, the notion of containment, also called "Quarantine" or "Isolation", is an effective method for separating a threat from other devices, thus reducing the risk of further contamination. An active example is naturally the notion of human quarantine, where by isolating infected people, we slow down the progression of the illness to others, potentially reducing their suffering. The notion of quarantine or isolation, however, aren't limited to the medical domain. Indeed, when applying the notion to the digital ecosystem, we see many similarities and studies of various methodologies, where isolating a device from network operations reduces the risks impacting passing communications [40]. Furthermore, different mitigation strategies also exist and are used in network communications, be it IoT-based or otherwise. These strategies include Network Isolation, Traffic Filtering and User Notification [44]. However, other solutions exist and have been analysed in the literature.

In [53], the authors propose an IoT-Botnet in Smart Homes detection and firewall-based isolation mechanism. This approach has the advantage of limiting network communications with the infected devices, but only functions where a central access point or controller is used. The authors of [161] also use a firewall-based system, allowing them to secure large-scale IoT networks or CPSs against malware spreading. However, they note that one of the encountered challenges is the deployment of these firewalls, placing them in locations where they would be the most efficient.

In [163], the authors propose a multistep process for detecting and isolating malicious nodes who perform Distributed DoS (DoS) attacks. Their methodology revolves around abnormal bandwidth usage, where an overly high transmission rate will mark the culprit as bad. In [30], the authors propose a modification to the IoT netstack, adding an "Isolation Layer", allowing channel isolation in mixed IoT networks. This approach has the advantage of being technology independent, allowing it to function what ever physical layer is in place, providing multiple levels of isolation. Its positioning between the link layer and the network layer, adds the extra advantage of isolating communications from malicious nodes. In a similar fashion, the authors of [69] propose an abstraction layer, allowing a logical view of the network containing only trusted devices. This approach used a white-listing initiative, where devices must authenticate prior to being accepted in the network, based upon the administration of a blockchain-based smart contract.

The authors of [62] turn their attention towards the spreading of malware in IoT networks and propose a secure patching framework to mitigate the threat. Their approach uses three interconnected modules, each based upon different elements in the network: edge gateway and backend; each with their own responsibilities, detecting compromised devices and raising the alert using remote attestation. Furthermore, they define three

[40]: Kahina Chelli. 'Security Issues in Wireless Sensor Networks: Attacks and Countermeasures'. In: *Proceedings of the World Congress on Engineering*. 2015. URL: http://www.iaeng.org/publication/WCE2015/WCE2015_pp519-524.pdf (visited on Oct. 17, 2022)

[44]: Furrakh Shahzad, Maruf Pasha, and Arslan Ahmad. 'A Survey of Active Attacks on Wireless Sensor Networks and their Countermeasures'. In: *CoRR* abs/1702.07136 (2017). DOI: 10.48550/arXiv.1702.07136

[53]: Sagarika Chowdhury and Mainak Sen. 'Survey on Attacks on Wireless Body Area Network'. In: *International Journal of Computational Intelligence & IoT, Forthcoming* (2019). URL: https://papers.ssrn.com/sol3/papers.cfm?abstract_id=3358378 (visited on Oct. 17, 2022)

[161]: Nurul Hidayah Ab Rahman and Kim-Kwang Raymond Choo. 'A survey of information security incident handling in the cloud'. In: *Computers & Security* 49 (2015). DOI: 10.1016/j.cose.2014.11.006

[163]: Tina Wu, Jules Ferdinand Pagna Disso, Kevin Jones, and Adrian Campos. 'Towards a SCADA forensics architecture'. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013. DOI: 10.14236/ewic/ICSCSR2013.2

[30]: Ansam Khraisat, Iqbal Gondal, Peter Vamplew, and Joarder Kamruzzaman. 'Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges'. In: *Cybersecurity* 2 (2019). DOI: 10.1186/s42400-019-0038-7

[69]: Ayei E Ibor, Florence A Oladeji, and Olusoji B Okunoye. 'A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention'. In: *International Journal of Security and its Applications* 12 (2018). DOI: 10.14257/ijsia.2018.12.4.02

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

levels of network isolation from trusted, possessing free communications; strict, where traffic is filtered to avoid contamination; and isolated, where all exiting communications are ignored.

Along similar lines, [81] looks at the effects of quarantine on worm infected nodes in WSNs as well as their recovery. Their proposed SEIQR model defines different states of infection, each influencing how the nodes are treated in the network: Susceptible, Exposed, Infected, Quarantined and Recovered. They identified a correlation between the rate of recovery and the number of infected nodes, where an increased recovery rate decreases the number of nodes infected.

[81]: Noam Ben-Asher and Cleotilde Gonzalez. 'Effects of cyber security knowledge on attack detection'. In: *Computers in Human Behavior* 48 (2015). DOI: 10.1016/j.chb.2015.01.039

The notion of quarantine models has been analysed in various aspects in the literature. Indeed, in [164] the authors analyse the previously proposed SEIQR model with various different incidence rates. Their analysis confirms the efficiency of quarantining infected nodes, where the higher the quarantine rate, the better the recovery process, reducing the overall infection rate. The authors of [165] analyse the effects of quarantining IoT devices, based upon the notion of network slicing, separating the normal IoT network from quarantined nodes. By using SDNs to configure and regulate traffic flow, the network operates transparently, with the different devices automatically redirected to their corresponding slice. By using a lightweight detection method, they can redirect suspicious devices to the quarantine slice, allowing a more in-depth analysis of their activities.

[164]: IT Governance UK. *Cyber Security Risk Management*. July 23, 2012. URL: https://www.itgovernance.co.uk/cyber-security-risk-management (visited on Sept. 25, 2022)

[165]: IT Governance UK. *Cyber Security Risk Assessment*. Apr. 30, 2019. URL: https://www.itgovernance.co.uk/cyber-security-risk-assessments (visited on Sept. 25, 2022)

Finally, in [166] the authors use a segregated trust architecture for a smart home environment, allowing all devices inside the network to trust each other, separating the internal network from the external. By using a trust model, based upon the device's reputation, known vulnerabilities, potential risk and the context, the home controller can allow a device to join the network or not. This extensive analysis, not only checks if the device is susceptible to attack through its known vulnerabilities, but also interacts with other devices, thus putting them at risk.

[166]: Petar Radanliev, D Charles De Roure, P Burnap, E Anthi, A Uchenna, L Maddox, Omar Santos, et al. 'Cyber Risk Management for the Internet of Things'. In: *Preprints 2019*. 2019. DOI: 10.20944/preprints201904.0133.v1

In our work, we provide the capability to avoid malicious nodes during routing activities, thus reducing the probability and impact of an attack. We also extend this by adapting the notion of quarantine to these networks, completely isolating malicious nodes from routing activities, thus protecting the network.

2.3.3 Operation Recovery

The final stage of any attack is the ability for a system or organisation to pick themselves up, dust themselves off and continue marching on. However, there are specific steps aims to help with the aftermath of an attack, be it with the recovery process of the affected infrastructure, or alerting to the attack and the continued learning process. When it comes to security incidents, there is a legal requirement to inform the national security agency that the attack has happened, giving an overview of the affected systems as an approximation of the overall damage. Thanks to post-incident methods, such as evidence gathering, this process is made easier. Furthermore, by learning from our mistakes and those of others, we can reduce the chance of such an attack happening again.

Incident Response Recovery

The last step of the IR stage is that of *Recovery*, which emphasises the importance of improved performance techniques and the utilisation of advanced backup technologies, such as online backup or cloud storage. To react quickly, an automated system capable of responding to an attack by deploying mitigation techniques dependent on the incident scenario is necessary. It is also important for the system to possess a low delay time between the detection and the response, especially in complex multistage attacks. An example is the Automated Intrusion Response System (AIRS) which possesses significant improvement in IR rate. There are three approaches to incident mapping: Static, Dynamic and Cost-Sensitive as presented in [161].

[161]: Nurul Hidayah Ab Rahman and Kim-Kwang Raymond Choo. 'A survey of information security incident handling in the cloud'. In: *Computers & Security* 49 (2015). DOI: 10.1016/j.cose.2014.11.006

Static Mapping

Static Mapping uses a pre-defined database, associating a specific incident alert to a specific response. The database is built from previous experience using probabilistic cognitive maps. Although its implementation is relatively simple, it doesn't protect itself entirely from a potential exploitation. Due to is static nature, attackers can circumvent the system by assuming a specific response to their attack. One way to rectify this vulnerability is to render the response strategy dynamic.

Dynamic Mapping

Using this new dynamic response strategy, Dynamic Mapping is capable of selecting a response based on the context of the incident and not a pre-defined response as before. This allows the system to dynamically adapt to attacks, making it harder to predict and therefore circumvent. However, this solution does not consider the damage or response cost, thus entertaining the possibility of an inappropriate response processing a larger cost than that of the incident itself. The necessity to compare max possible damage costs with those of possible responses increased the interest of cost-sensitive mapping.

Cost-Sensitive Mapping

A Cost-Sensitive Mapping technique is key to being able to balance both damage and response costs. This allows to reduce the cost of implementation as well as the amount of necessary resources, the temporal effectiveness and the cost of induced modifications. The three major cost factors which are examined are as follows:

- **Damage**: amount of damage to target resource
- **Response**: cost of acting on an intrusion
- **Operational**: cost of processing the IDS

We incorporate the notion of recovery into our module through the aspect of "Decay", allowing node reintegration after a period of inactivity. In doing to, we allow sanitised nodes to be reintegrated and participate once more in routing activities. We also extend this recovery to apply to nodes which have been identified as the most trustworthy, thus "decaying" their standing to return them to an equal footing after a long period of inactivity.

Post-Incident

After an incident has been resolved, Post-Incident processing must be done. This is the final phase where information and results are used as feed-back to improve future incident handling. Using Adaptive Incident Learning, the system and users have the ability to change and learn from past experiences. The information collected from the three previous phases is used to generate a Post-Incident Report explaining the incident as well as possible improvement recommendations in incident handling from both a technical and a managerial perspective. It is the least studded phase of the Incident Handling system.

After the dust has settled on networking activities, we provide the ability to analyse the past actions, to identify malicious parties through consensus, and distribute the information using blockchain. With this distribution, we provide the necessary information for the preparation phase, where nodes are aware of potential threats, and can, therefore, avoid them.

Digital Forensics

Incident Handling focuses mainly on responding to incident breaches without general consideration for evidence collection. This evidence could provide valuable data for current investigations but also the future prosecution of the offender. Using similar security tools as Incident Handling, Digital Forensics is a scientific discipline concerning the collection, analysis and interpretation of digital data connected with a computer security incident. Using legally admissible methods, the recovered data from compromised systems can help reconstruct incident facts but also can be used for risk mitigation in the Post-Incident phase. Integrating Digital Forensic analysis into the Detection and Analysis phase can facilitate the identification of key assets as well as vulnerabilities and threats which could be exploited. Appropriate and effective risk assessment and mitigation strategies help ensure the system is forensic ready, thus when an incident occurs, responding investigators know where potential evidence can be found. This facilitates efficient and timely incident response and forensic examination. A version compatible with CIs based on SCADA forensics architecture is presented in [163].

[163]: Tina Wu, Jules Ferdinand Pagna Disso, Kevin Jones, and Adrian Campos. 'Towards a SCADA forensics architecture'. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013. DOI: 10.14236/ewic/ICSCSR2013.2

2.4 Conclusion

To effectively counter a cyber attack, it is important to not only understand the attack itself, but also how it can take place. By studying the cyber attack life-cycle, security professionals can create effective defensive mechanisms to counter these threats. However, in order to do so it is necessary to categorise the different attacks, allowing quick and easy understanding of multiple factors, such as the attackers position, their implication or the overall impact. That being said, with the large number of possible threats against CIs as well as the incorporated IoT devices, there also is a large number of categorisation approaches, each adapted to the security professional who creates and uses it.

In the same idea, many different detection systems exist, each with their own advantages. IDSs can employ multiple analytical algorithms dependant on their basic use, such as using signatures (SIDS) or anomalies (AIDS) to detect intrusions. Be the algorithms statistics-based, relying on machine learning techniques or using Information theory or clustering, these methods must be trained to reach peak efficiency. With the exception of continuous learning methods, such as reinforcement learning, IDSs can pick from the large quantity of datasets to find one tailored to their application to help them learn the difference between normal operation and malicious intentions. Furthermore, by representing the different threats perceived against them using a *Threat Taxonomy*, it is possible to easily prioritise threat response.

Although IoT devices have been analysed in the context of CI, there hasn't been much analysis in the areas of multi-hop network, specifically applicable to these infrastructures. It is here that the work of this thesis has been performed, to provide the necessary tools for multi-hop networks to autonomously detect threats upon routing operations. Furthermore, with the addition of identification metrics, IoT devices in these networks would have the capabilities to identify the source of the threat, thus allowing network operations to adapt and continue as normal possible. By using some of the concepts from NIDS regarding traffic analysis, all the necessary information can be provided.

When it comes to threat response, the *Incident Handling* guide provides the necessary steps to respond to a threat, from preparation to recovery. An important phase is the notion of containment, allowing isolation of compromised systems. Although such isolation techniques are commonly used in operating systems and large infrastructures, in certain conditions this becomes more of a challenge, such as multi-hop networks. To tackle this challenge, we extend our work into threat detection and identification, effectively providing advanced quarantine methods to the network, thus allowing the IoT devices in the network to isolate the malicious intruders, increasing the overall efficiency and essentially, securing the CI from routing attacks on its relay infrastructure.

Consensus-Based Reputational Analysis

3

In Chapter 2, we provided an overview of the various aspects in the security in CIs, as well as those more targeted towards ad hoc wireless networks, from threats to threat response. In this chapter, we delve deeper into the security of these ad hoc IoT devices, taking a look at threat detection based around trust and reputation. We provide an overview of how such a method functions vis-à-vis wireless devices and how, by levying blockchain technology, the resulting reputation can be shared throughout the network.

During our analysis, we hypothesise that these networks utilise wireless technologies which support ad hoc communications, such as 802.11 WiFi or 802.15.4 *Zigbee*. Furthermore, we suppose that these networks employ modern network technologies, such as IPv6 on the network layer for data exchange and routing capabilities. We also theorise that the networks themselves form a connected mesh topology, where all nodes can be reached by all others through the use of various multi-hop routing protocols.

3.1 Behavioural-Based Observation

Behaviour is an important factor with regards to building trust with someone or something. Being social animals, through observing and analysing the behaviour of people around us, we in turn can adapt our own response to the environment. That being said, although trust is often reduced to a simple yes or no answer, it is actually more complicated for us humans. To be able to trust, we must be shown proof that the intended target of said trust possesses good intentions. However, these good intentions are also widely spread and come in a large variety of behavioural keys.

These keys can be adapted to electronic devices in various scenarios. For example, if we see someone being bad in the street, we will have a tendency to avoid them. The same can be performed in a networking context, where observing a device perform malicious intentions would result in the device being avoided in the future. In this section we present an overview of human trust and how it would be applied to IoT devices along with the discrepancies due to their operation. Finally, we will present how this behavioural analysis would allow devices to provide reputational values between them, identifying whether they can be trusted or not.

3.1.1 Human Inspired Trust

The Cambridge Dictionary defines trust as "*to believe that someone is good and honest and will not harm you, or that something is safe and reliable*" [167]. By analysing this notion in relation to human psyche, we can indeed identify that trust revolves around our impressions or feelings towards another

[167]: Cambridge Dictionary. *Trust*. Aug. 10, 2022. URL: https://dictionary.cambridge.org/dictionary/english/trust (visited on Sept. 4, 2022)

person. This interpersonal trust is the basis of human interactions and highly governs the company we keep, surrounding ourselves primarily with people in which we trust. Trust is also an important factor in a professional environment, as each person is but a cog in the wheel of the organisation which must keep turning. Indeed, if a single mesh on one of the cogs isn't correctly aligned, the whole machine would slow down and even break.

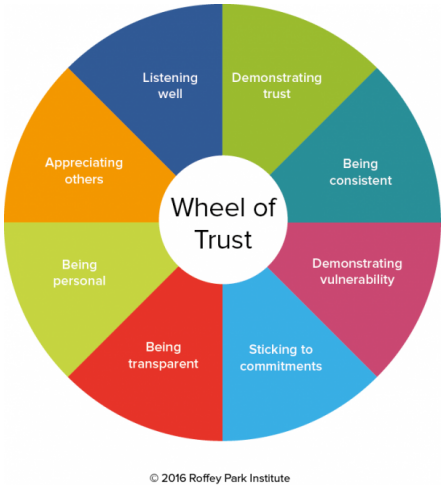


Figure 3.1: The eight behaviours that build trust [168]

[168]: Meysam Poorkavoos. *Eight behaviours that build trust*. Oct. 19, 2016. URL: https://www.roffeypark.ac.uk/knowledge-and-learning-resources-hub/eight-behaviours-that-build-trust/ (visited on Sept. 4, 2022)

Research into interpersonal trust has brought to light that there are multiple specific behaviours which both lead and contribute to trust [168]. Although this study concerns human interactions, in some cases, there are similarities which can be made towards the interactions between IoT devices. By analysing their importance towards personal trust, it is possible to also identify their importance and impact towards devices in a networking context. In total, eight different behaviours have been identified in [168], shown in Figure 3.1:

- **Trust**: Trust isn't given, but must be earned. One of the easiest ways to do so is through trust itself. Indeed, by trusting someone first, we show them that we have confidence in their actions, thus making it more likely for them to trust back.
- **Consistency**: Consistency must not be confused with repetition. By being consistent, we remain true to our beliefs and our previous actions. This may mean that we don't always perform the same tasks, but we act accordingly with regards to the current situation and respect the guidelines and specifications in place.
- **Vulnerability**: Everyone makes mistakes, but what is important is owning up to them. By accepting the responsibility of our actions and doing what's necessary to remediate any consequences, we show that we not only express remorse for our errors, but also that we know what went wrong and are willing to learn and correct the issue.
- **Commitments**: If we are asked or offer to do something, then it is important to do it. This shows our reliability and proves that we can be trusted with certain tasks and that we can be counted on.
- **Transparency**: By being both honest and open with other people, they are more likely to get a better understanding of who we are. This includes if we are not able to complete our previous commitments, especially if something unexpected got in the way. People are more likely to forgive when given the chance, especially when something wasn't our fault.
- **Being personal**: When interacting with others, it is important to be ourselves, all the while respecting our own and others privacy. By not closing down and actively interacting with others, we can let them know who we really are, allowing them to gain a better understanding of our personality.
- **Appreciation**: Respect is an important factor with human interactions. There are multiple levels of respect which must be earned, however, the first level is that of common courtesy. By showing our appreciation and respecting other people's actions, we show them the respect they are owed.
- **Listening**: Another element where we can show respect is towards people's opinions. Simply through the act of listening, we can gather a greater understanding of the issues, enter into debates

where opinions differ, allowing us all to grow and gain knowledge.

Although these aspects are specific to human nature, there are some similarities with IoT devices. In networking, showing *trust* in another device is implicit by design, where each node makes a *transparent commitment* towards routing data to the correct target. However, this trust or respect isn't clearly identifiable in the context of the IoT. As a result, network-based equivalents for certain aspects, such as *listening* or *being personal* can be represented by direct technological equivalents, such as constantly *listening* to traffic for routing purposes, as well as performing what is expected if it, such as routing correctly. All in all, IoT devices are generally designed with a specific goal in mind, from sensing (i.e., thermometer) to actively taking part in a specific task (i.e., car). Here, action *consistency* is important, where devices are expected to perform their specific task, where in many cases other interconnected systems rely on their input.

As we can see, these different behavioural types possess certain similarities with network devices. Furthermore, there are two important aspects which must also be analysed: how the trust is represented; and how the behaviour is analysed.

3.1.2 Operational Discrepancy

How trust is represented is generally specific to the case in use. For example, humans represent trust by our subsequent interactions with each other. If we trust our fellow man, we are more likely to interact with them than if there was not trust between us. Furthermore, our level of trust is not only directly linked to our observation of the aforementioned behavioural keys, but is also influenced by word of mouth. Indeed, if we hear that someone has done something bad, even though we haven't directly observed it, our potential trust level will diminish.

In this case, by basing our actions on what we have heard and not observed, we are being influenced by the notion of "reputation". Defined by the Cambridge Dictionary as "*the opinion that people in general have about someone or something, [...] based on past behaviour or character*" [169], we can see a correlation with the notion of trust. In particular, trust is based on the conceptualisation of goodness in an individual or entity based generally on behaviour. In this case, by analysing the past behaviour of said individual, we can determine the level of respect they receive. As a result, their reputation directly impacts the level of trust we convey, thus allowing a more precise idea of their reliability. Furthermore, how much we trust the source of this information is also important, as it defines how seriously we take the provided reputation and how much in turn we trust it.

This notion of reputation and trust is used often by businesses allowing them to promote products based upon the opinion of previous clients. This is the case of many search engines and platforms such as the well known booking service *Trivago*. As we can see in Figure 3.2, by searching for hotels in London we are proposed multiple options, each with grand names and luxurious images. However, looks can be deceiving so it is important to gain an insiders opinion on each of these stays. This is

[169]: Cambridge Dictionary. *Reputation*. Aug. 10, 2022. URL: https://dictionary.cambridge.org/dictionary/english/reputation (visited on Sept. 5, 2022)

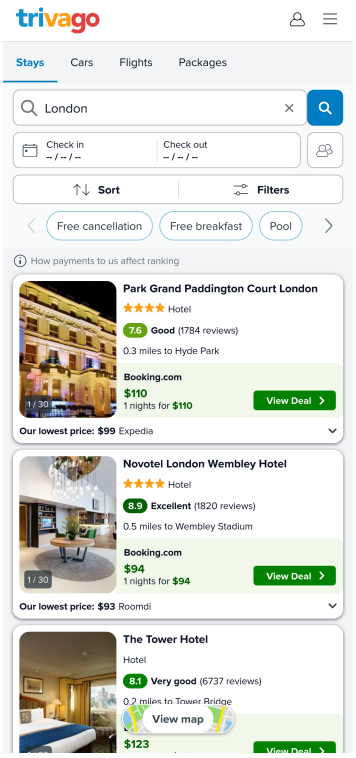


Figure 3.2: Trivago website showing a search for hotels in London, each with a rating and price [170]

[170]: Trivago. *Trivago hotel, taxi and flight booking service*. Feb. 28, 2011. URL: https://www.trivago.com/ (visited on Aug. 9, 2022)

where the notions of reputation and trust enter into play. Here, each hotel possesses an overall rating score between *0* and *10*, indicating the average opinion of their clients with a higher value representing a higher reputation and thus level of trust. In this instance, we can see that the *Park Grand Paddington Court London* hotel possesses a score of *7.6*, whereas the following *Novotel London Wembley Hotel*'s score is slightly higher at *8.9*. Calculated based on the reviews left by previous clients, the resulting score provides an objective overview of the target location, allowing future clients to quickly create an opinion. Furthermore, if they so wish they can delve into the history of the hotel through the different reviews, thus gaining an even further understanding as to the scores, indicating for example if the choice or marmalade served at breakfast in the Paddington hotel is a contributing factor.

This concept is used in many areas, from the tourism industry to online shopping, and influences our actions on a day-to-day basis. We can, therefore, conclude that by analysing the behaviour patterns of either a person or an entity, at a specific moment in time, we can determine their current level of trustworthiness. However, by expanding this analysis to an overview of their direct history, we can build a detailed profile based upon their reputation, which in turn can heavily influence the amount of trust provided.

3.1.3 Network Reputation

As shown previously, there are many behavioural aspects to the definition of trust which can be adapted to IoT network devices. Furthermore, by providing a determination of a devices reputation, it would be possible to determine the level of trust, based upon an analysis of their previous actions. The main goal of this task, is to provide each network node with the information needed to make an informed decision about which neighbour it should communicate with. As a result, the nodes would be able to select a more reputable neighbour, thus increasing the reliability and overall integrity of the network.

This is the case of [171] where the authors use trust-based methods to identify nodes in the network, based on their previous activities. By evaluating multiple types of activities based on node social interactions and Quality-of-Service (QoS), *trust* profiles can be build for each network device. This would allow other nodes to evaluate the trustworthiness of their neighbours based upon these profiles, influencing the selection of which node to interact with. In a similar fashion, [172] integrates this trust-based functionality into a routing protocol for WSN. For each action taken by the intermediate nodes, a trust value is computed by its neighbours, providing intelligence on the nodes operations. To do so, multiple aspects are taken into account, such as the node's *consistency* with regards to data forwarding, their *transparency* in acknowledging previous packets as well as their *commitment* to continued operation trough their overall energy consumption. Armed with this analysis, the resulting value is then used to determine the most trustworthy candidate to relay the data throughout the network.

As stated previously, both reputation and trust metrics can be expressed in multiple fashions dependant on the use case and situation at hand. An

[171]: Fenye Bao, Ing-Ray Chen, MoonJeong Chang, and Jin-Hee 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 (2012). DOI: 10.1109/TCOMM.2012.031912.110179

[172]: Deep Kumar Bangotra, Yashwant Singh, Arvind Selwal, Nagesh Kumar, and Pradeep Kumar Singh. 'A Trust Based Secure Intelligent Opportunistic Routing Protocol for Wireless Sensor Networks'. In: *Wireless Personal Communications* (2021). DOI: 10.1007/s11277-021-08564-3

example of this is provided by the authors of [173], where neighbouring behavioural patterns are evaluated using inter-node cooperation. Relying not only on their direct observations, but also on recommendations passed from their neighbours, nodes are capable of determining the trustworthiness of the node in question. By using metrics specific to the device itself, such as energy consumption, its honesty with regards to *transparency* of malicious intent as well as its selfishness in its *consistency* and *commitment* to network operations. On the other hand, the authors of [174] use a signature-based methodology, allowing to validate passing data integrity. Their Aggregate Signature based Trust Routing (ASTR) scheme, coupled with a lightweight aggregate signature-based detour routing scheme, allows the exchange of abstract transmission related information, providing the capability to verify that data reaches the intended sink. In doing so, the sender can influence the trust level of the routing path, identifying if data was lost, or on the contrary arrived intact. Another solution presented in [175] proposes an energy efficient hierarchical routing protocol for WSN which utilises trust management to enhance the networks ability to defend against attacks. In this situation, malicious entities are identified through the observational analysis of network sensor nodes, by associating a trust value directly to the node, influencing the selection of cluster heads.

[173]: Nabil Djedjig, Djamel Tandjaoui, Faiza Medjek, and Imed Romdhani. 'Trust-aware and cooperative routing protocol for IoT security'. In: *Journal of Information Security and Applications* 52 (2020). DOI: 10.1016/j.jisa.2020.102467

[174]: Jiawei Tang, Anfeng Liu, Ming Zhao, and Tian Wang. 'An aggregate signature based trust routing for data gathering in sensor networks'. In: *Security and Communication Networks* 2018 (2018). DOI: 10.1155/2018/6328504

[175]: Weidong Fang, Wuxiong Zhang, Wei Yang, Zhannan Li, Weiwei Gao, and Yinxuan Yang. 'Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks'. In: *Digital Communications and Networks* 7 (2021). DOI: 10.1016/j.dcan.2021.03.005

The authors of [176] utilise the reputation-metric to influence the routing decision based upon the level of trustworthiness of each intermediate node. By observing and analysing routing activities, they are capable of identifying a list of *good* and *bad* actions, which are then securely distributed throughout the network using blockchain technology. Thanks to the history provided by the blockchain, they can compute a nodes reputation as a non-linear sigmoid function using the historical weighted average determined from the different blocks.

[176]: Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450

However, there are areas of improvement where this work is concerned. Indeed, by basing our proposition upon their previous study, we can provide further capabilities in particular towards the distinction between *good* and *bad* actions, as well as protocol integration, specific to AODV in their case. Thus, we not only extend their basis for the computation of reputation, but also provide a different angle into the integration with multiple protocols, allowing network and protocol adaptability.

3.1.4 Blockchain-Based Distribution

The notion of blockchain is not new and has been steadily gaining in popularity in recent years due to their wide spread use in many cryptocurrencies, such as the renowned Bitcoin [178]. The blockchain is a decentralised immutable ledger, openly shared and accessible by all necessary participants as explained in [179]. The data stored in the blockchain differs dependant on the use case, for example in cryptocurrencies, the data is a list of monetary transactions performed with the corresponding currency. This data is stored in the form of "blocks" containing both the contents and a header with information related to the block itself as well as other blocks in the chain, some of which are illustrated in Figure 3.4.

An advantage of the blockchain is its immutability, which is achieved through the utilisation of block header *hashes*. Indeed, each block contains

Figure 3.3: Bitcoin cryptocurrency logo [177]

[177]: Wikipedia. *Bitcoin*. Oct. 6, 2022. URL: https://en.wikipedia.org/wiki/Bitcoin (visited on Oct. 15, 2022)

[178]: Andreas M Antonopoulos. *Mastering Bitcoin: Programming the open blockchain*. "O'Reilly Media, Inc.", 2017. URL: https://books.google.com/books?hl=en&lr=&id=MpwnDwAAQBAJ&oi=fnd&pg=PP1&dq=Mastering+Bitcoin:+Programming+the+open+blockchain&ots=wR9pqoAYG1&sig=obi1pv2WIkHHtvj1-RAIo5EG-bY (visited on Oct. 17, 2022)

[179]: NARA. *Blockchain White Paper*. Tech. rep. National Archives and Records Administration, 2019. URL: https://www.archives.gov/files/records-mgmt/policy/nara-blockchain-whitepaper.pdf (visited on Oct. 17, 2022)

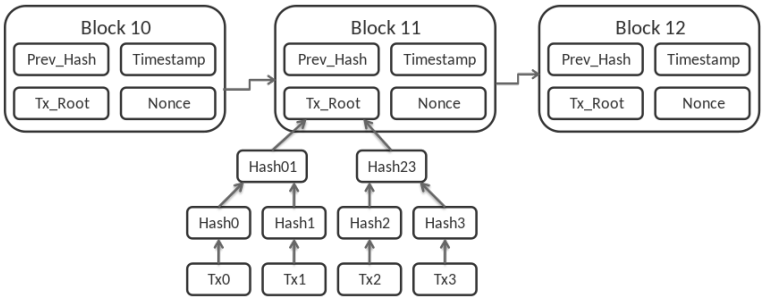


Figure 3.4: The structure of the blockchain in a Bitcoin application where each block is linked to its successor, all containing the hash of its predecessor. Each block also contains a timestamp of the blocks creation as well as a Nonce used during the PoW. There is also the Merkle tree root value contained in *Tx_Root*[180]

[180]: Wikipedia. *Blockchain*. Oct. 14, 2022. URL: https://en.wikipedia.org/wiki/Blockchain (visited on Oct. 15, 2022)

[181]: Wikipedia. *Merkle tree*. Sept. 9, 2022. URL: https://en.wikipedia.org/wiki/Merkle_tree (visited on Oct. 15, 2022)

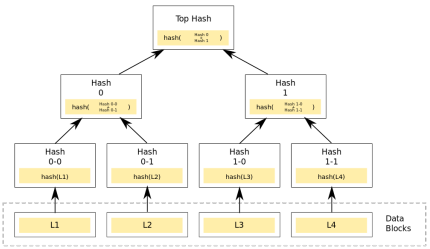


Figure 3.5: The format of a binary Merkle hash tree built from four data blocks [181]

[182]: Xiaoqi Li, Peng Jiang, Ting Chen, Xiapu Luo, and Qiaoyan Wen. 'A survey on the security of blockchain systems'. In: *Future Generation Computer Systems* 107 (2020). DOI: 10.1016/j.future.2017.08.020

[183]: Muhammad Salek Ali, Massimo Vecchio, Miguel Pincheira, Koustabh Dolui, Fabio Antonelli, and Mubashir Husain Rehmani. 'Applications of Blockchains in the Internet of Things: A Comprehensive Survey'. In: *IEEE Communications Surveys & Tutorials* 21 (2019). DOI: 10.1109/COMST.2018.2886932

[184]: Axel Moinet, Benoît Darties, and Jean-Luc Baril. 'Blockchain based trust & authentication for decentralized sensor networks'. In: *arXiv preprint arXiv:1706.01730* abs/1706.01730 (2017). DOI: 10.48550/arXiv.1706.01730

[185]: 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. DOI: 10.1109/MSN.2018.00009

[186]: Caciano Machado and Carla Merkle Westphall. 'Blockchain incentivized data forwarding in MANETs: Strategies and challenges'. In: *Ad Hoc Networks* 110 (2021). DOI: 10.1016/j.adhoc.2020.102321

[187]: Jidian Yang, Shiwen He, Yang Xu, Linweiya Chen, and Ju Ren. 'A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks'. In: *Sensors* 19 (2019). DOI: 10.3390/s19040970

the *hash* of their predecessor's header, essentially linking them together in a *chain*, hence the name. The blockchain also contains a hash of its data in the form of a *Merkle Tree* [181], the process of which is shown in Figure 3.5, providing a secure and efficient verification method for large quantities of data.

Furthermore, the blockchain employs devices known as "miners" to perform validation activities on data blocks befor ethey are inserted. This mechanism involves confirming the validity of the data through a heavy cryptographic computation called Proof-of-Work (PoW), which is confirmed by other miners through a consensus mechanism before being added to the blockchain. Another advantage of blockchain technology is its aforementioned immutability, as analysed by the authors of [182]. This has lead to it being used in other areas, such as IoT security as show in [183]. However, the specifics of IoT devices and their networks give way to many challenges, one of them being the Proof-of-Work (PoW) mechanism itself, impacting both the limited energy and computation capacities, impacting the devices lifetime.

That being said, the blockchain has seen its fair share of attention in the area of security. Indeed, in [184], the authors use the blockchain as a secure data structure, providing authentication and trust services in the IoT. Chosen for its secure and distributed nature, the blocks contain lists of public cryptographic keys as well as digital signatures and node peer information, enforcing trust between nodes. In [185], the blockchain is used as a decentralised network to increase data integrity and authenticity. Here, the blockchain monitors traffic, storing and encrypting packet origins, which are subsequently verified by the base station to confirm network integrity. Blockchain has also seen some recent interest in the areas of routing, with many different methods employed to increase overall security [186].

An example is the work performed by the authors of [187]. Here the blockchain stores information related to the data transmission, allowing all nodes to participate in determining the "legality" of the exchanges. In [188], the authors use the blockchain to store and share the status of the network in real-time to enhance the routing process. By checking the list of transactions, nodes can determine the most efficient route, thus avoiding congested areas and nodes. This technology has also been used in Unmanned Aircraft Systems as in [189], improving both routing activities and authentication. Here, a lightweight blockchain deployment is used, providing each drone with identification and authentication information. The authors of [190] propose a novel routing protocol based

on blockchain contractual methodology. By using the ledger to store smart contract addresses indicating when routing is needed, routes can be offered and determined when needed.

[188]: Hilmi Lazrag, Abdellah Chehri, Rachid Saadane, and Moulay Driss Rahmani. 'A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT'. In: *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. 2019. DOI: 10.1109/SITIS.2019.00072

[189]: Jian Wang, Yongxin Liu, Shuteng Niu, and Houbing Song. 'Lightweight blockchain assisted secure routing of swarm UAS networking'. In: *Computer Communications* 165 (2021). DOI: 10.1016/j.comcom.2020.11.008

[190]: Gholamreza Ramezan and Cyril Leung. 'A Blockchain-Based Contractual Routing Protocol for the Internet of Things Using Smart Contracts'. In: *Wireless Communications and Mobile Computing* 2018 (2018). DOI: 10.1155/2018/4029591

3.2 Activity Reputation

As we have seen, the notion of reputation is based upon direct observation as well as the public history of the target entity. In the case of IoT multi-hop routing, their history is key to determining their level of trust. Through this analysis, we are capable of gaining an accurate impression of the nodes trustworthiness, based amongst others upon its reliability. However, when analysing history we must also be aware of a generally underestimated enemy: time itself.

In this section, we explain how we can derive a reputation value based upon a nodes history. From here, we extend this through an explanation of the influence time has upon the determination of reputation as a whole, as well as its significance in an IoT network. We then build on this once more to propose a method for allowing the reputation value to be affected overtime, based upon the last action perceived for the target node.

3.2.1 Reputation-Based Return

The notion of *good* or *bad* actions are determined from the behavioural analysis performed during routing. These binary actions allow to differentiate between what is expected which represents a *good* action, and anything else where any deviation would be considered as *bad*. As a result, the more actions are in either category, the more the reputational scale will tip towards one side or the other. In short, the greater the amount of *good* actions, the higher the reputation we become and vice-versa.

$$S^R_{good_n} = \sum_{i=1}^{W^R_n} good\ actions_{n^R_i} \tag{3.1}$$

$$S^R_{bad_n} = \sum_{i=1}^{W^R_n} bad\ actions_{n^R_i} \tag{3.2}$$

We define $S_{good^R_n}$ and $S_{bad^R_n}$ as the sum of *good* and *bad* routing actions respectively for node n, as computed in Equation 3.1 and Equation 3.2. We also define W^R_n as the size of the routing 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^R_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 and thus, react accordingly.

Armed with the quantity of *good* and *bad* actions during the defined time frame, we can calculate the nodes reputation. The reputation R^R_n as shown in Equation 3.3 is defined in $[0, 1]$ and is expressed as a non-linear

sigmoid function We define the exponent δ_n^R as shown in Equation 3.4, itself specified in $[-1.\ 1]$, which in turn represents the weighted value of the relation between $S^R_{good_n}$ and $S^R_{bad_n}$.

$$R_n^R = \frac{1}{1 + e^{-\delta_n^R}} \tag{3.3}$$

$$\delta_n = \beta \times \frac{S^R_{good_n} - \alpha \times S^R_{bad_n}}{S^R_{good_n} + \alpha \times S^R_{bad_n}} \tag{3.4}$$

To further extend our analysis, we introduce two variables for the calculation of δ_n^R. The first of these is β, which which corresponds to the sensitivity factor influencing the sigmoid function. Previously used in [176], its purpose is to increase the impact of variations produced by the weighted calculation between *good* and *bad* actions. In this situation, we define $\beta = 8$ as shown in [176] which shows an adequate influence in the reputations distribution The second is α which is defined as the weight of malicious routing 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.

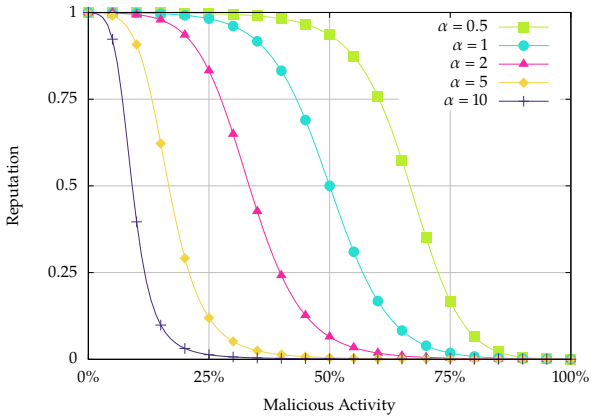


Figure 3.6: The influence of α upon the evolution of the reputation in relation to the proportion of malicious actions perceived.

Figure 3.6 allows us to visualise the impact of the choice of α in relation to the malicious intentions of the corresponding node. We can see that our assumption is indeed justified, where the higher the value of α, the quicker and steeper the drop in reputation. One way to view this is to look at what can be considered a neutral reputation, *0.5*. With $\alpha = 1$ (**light-blue** circles), a neutral reputation is met for 50% malicious activities, corresponding to $S^R_{good_n} = S^R_{bad_n}$. If we half this value to $\alpha = 0.5$ (**green** squares), it would take double the number of *bad* actions to match a single *good* one, corresponding to a malicious activity of approximately 66%, i.e., $2 \times S^R_{good_n} = S^R_{bad_n}$. Going the other way and doubling the value to $\alpha = 2$ (**pink** triangles), we can see that this interaction in inverted, where it would take double the number of *good* actions to reach forgiveness for a single *bad* one, corresponding to approximately 33% malicious activity, i.e., $S^R_{good_n} = 2 \times S^R_{bad_n}$. By increasing the value of α even further to $\alpha = 5$ (**yellow** diamonds) and $\alpha = 10$ (**purple** pluses), we can see that the network becomes more and more unforgiving, heavily penalising a node if it takes a single bad action.

3.2.2 Device Profiling

Depending on which way you look, time can both be considered as a companion or an enemy. Indeed, time allows us to grow and evolve, but it can also take us further away from times and people, lost in the past. Thankfully, we can still remember the good times thanks to our memories. Unfortunately, the more time passes, the more these memories decay, leaving our recollection of events and feelings slightly altered. This of course not only impacts ourselves, but also the recommendations and suggestions we give to others.

Since the notion of reputation relies heavily upon the past behaviour of the target entity, temporal decay can heavily impact the result. Examples of this can be seen in our everyday lives, including the interactions with our friends and family. Indeed, if we haven't seen someone for a very long time, it is possible that we will not recognise them during our next encounter due to their possible evolution. This means that our initial perception all those years ago becomes more and more inaccurate as time goes by. Due to our memories fading and mixing together, it is possible that our recollection of something we currently perceive as being bad, might have been completely the opposite at that time it occurred. For example, a restaurant which was considered bad a few years prior, may now be considered excellent, even to the stage of possessing several MICHELIN stars. As a result, this time variation must play a prominent point in the computation and analysis of the reputation.

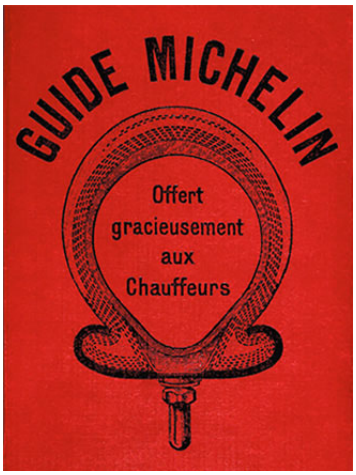


Figure 3.7: Original Michelin Guide from 1900: "Graciously offered to drivers" [191]

[191]: MICHELIN Guide. *About Us*. July 10, 2019. URL: https://guide.michelin.com/gb/en/about-us (visited on Sept. 5, 2022)

Whereas humans have a tendency to mis-remember or even sometimes forget important things, electronic devices do not. Dependant on the use case and implementation, electronic devices keep logs of their previous actions, thus keeping an explicit record of any potential errors. In the case of NIDS, network traffic logs provide an overview of the perceived exchanges, allowing the identification of any misbehaving network devices. As a result, once an action is perceived and written "bits-on-drive" as it were, they will remain so until explicitly removed. By constructing a reputational profile of the corresponding device, we can clearly identify not only its malicious actions, but also impact its reputational standing, allowing other devices to be aware of its mishaps. However, since the actions will remain almost indefinitely, one mishap a long time ago would still impact their standing in the future. As such, a method of expressing the human concept of forgiveness as well as temporal bias would allow devices to have a second-chance, but also make sure that only those devices with good reason are to be trusted.

3.2.3 Temporal Decay

As we have seen, the principles of temporal variance can also be applied to devices in the cyberspace. There are multiple causes for behavioural change in electronic devices, all the way from firmware policy updates to simple algorithmic bugs. However, in the domain of cyber-security, another risk is that of compromise by a third external party, rendering the device essentially malicious. That being said, hackers generally possess a specific target for their actions which, if not reachable or their presence is detected, they will abandon the infected device in favour of a new point of attack. This means that a previously compromised device, for example

causing problems during routing, would once again function as normal and could be allowed back into network operations.

Another possibility concerns the opposite principal, where a node with an almost perfect reputation but which hasn't partaken in routing for a while can pose a security risk. Indeed, since this device managed to achieve a good standing in the network it would be explicitly trusted the next time it is needed. As a result, this device could be considered as a perfect target for attack, allowing the attacker to immediately impact network operations, at least until the device is detected and its reputation impacted. One way to remedy this would be the addition of a *Reputation Decay* metric, whose goal would be to achieve what us humans can do: decay the resulting reputation the more time has passed since their last use. As stated, there are two main advantages to this method:

1. Allow the reintegration of sanitised malicious nodes back into the network. Thanks to their low reputation, these devices will be avoided from partaking in routing activities. However, through the use of reputation decay, unused devices can once again prove their allegiance and start to regain their reputational standing.
2. Remove explicit trust from unused good nodes. On the same idea, unused nodes which possessed very good or exceptional good values would be trusted explicitly upon their return. By allowing their reputation to decay over time, they will once again be back on the same level as their network colleagues, thus returning to a level status quo.

In either case, the objective is to return the computed value to what can be considered a "neutral" amount. Since the reputation is computed in $[0, 1]$, it is logical to consider the centre point, 0.5 as being the starting neutral value. This would allow all nodes to commence on equal footing, allowing good nodes to take steps forwards, leaving malicious nodes falling behind. However, in contrast to human reputational decay, where over time our recollection of historical events may become tainted, electronic records remain intact. As a result, the reputational decay doesn't alter the list of *good* or *bad* actions, but simply serves as a means for granting nodes second chances. We define $Rd^R_{n_t}$ as the routing reputation decay of node n at time t where t^R_n represents the time stamp of the last reputational update, i.e., the last action performed by n.

$$Rd^R_{n_t} = (t - t^R_n) \times (\frac{\lambda^R_n}{t^R_{\frac{1}{2}n}}) \tag{3.5}$$

$$R_{n_t} = R^R_n - Rd^R_{n_t} \tag{3.6}$$

There are multiple methods of decay used in the scientific domain, especially in the area of particle physics and chemistry, where the term "*half-life*" is very common. We take inspiration from these other uses and define two new variables to allow for precise customisation of the decay rate. Firstly, we define λ^R_n as the routing decay factor for node n itself, influencing the rate at which the reputation will decrease during a certain time period. This time period is defined as the reputation's half-life $t^R_{\frac{1}{2}n}$, allowing to increase or decrease the time needed. As a result, reputational convergence towards 0.5 can occur either sooner or later, impacting the rate at which malicious nodes can be reintroduced. Armed

with this decay information, we can compute the final reputation R_{n_t} of node n at time t.

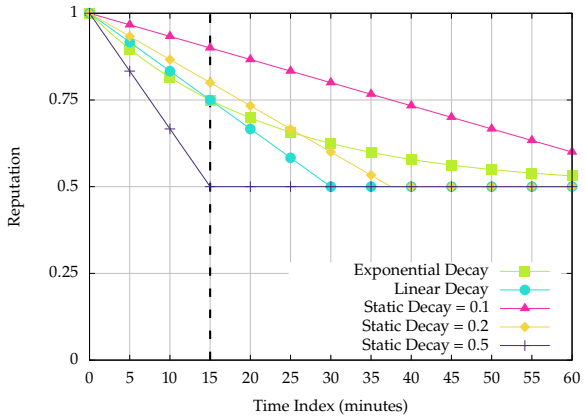


Figure 3.8: Visualisation of the different methodologies for representing the temporal decay of reputation from static linear approaches to the well know exponential decay used in physics and chemistry, with a half-life of 15 mins.

As stated, multiple decay factors or half-life's can be utilised, significantly impacting the rate of decay. Figure 3.8 illustrates a few examples of decay, all with a half-life of 15 minutes. Here, there are three indistinct types of decay:

- **Exponential Decay**: Reduces the concerned value by half after every passage of time equal to the entities *half-life*. Being in exponential form, the value will slow its descent the more time passes, inching closer and closer to 0, all the while never reaching it. It is shown with the **green** squares.
- **Linear Decay**: Uses the same notion as the exponential decay, where the initial value is reduced by half after *half-life* has passed, only this time the rate continues, reaching 0 after $2 \times$ *half-life*. This representation is shown with the **light-blue** circles. We can clearly see that it intersects with the exponential decay at the 15 minute mark.
- **Static Decay**: Reduces the value by a specific amount after each *half-life*. By varying the corresponding value, the rate of decrease itself be increased or decreased. Here we show three versions of this rate with $\delta = 0.1$ (**pink** triangles), $\delta = 0.2$ (**yellow** diamonds) and $\delta = 0.5$ (**purple** pluses). We can see that the higher the rate of decay, the quicker the value drops, visible with a decay of 0.5 where the reputation reaches neutral after 15 minutes have passed. On the other hand, a rate of 0.1 leave the reputation relatively high, which still hasn't reached the neural value after one hour.

Armed with the correct decay value Rd_n based-upon the timestamp of their last observed action, it is now possible to compute the final reputation for the concerned node. Algorithm 1 provides an overview of the algorithm used to allow the reputation to centre itself around 0.5 instead of 0.

Firstly, the reputation R_n is reduced by 0.5, thus placing the centre point on the equivalent of 0.5 reputation. The decay value Rd_n is then applied directly to the reputation, returning it back to 0. By stopping it from decaying further in either direction, we block the final reputation value at neutral. Finally, the reputation is once again increased by 0.5, thus returning it all back to its original position, with the reputation decayed back towards the neutral 0.5.

Algorithm 1 Calculation of the node n's decayed reputation back to neutral value of 0.5

```
1: function Reputation Decay(R_n, Rd_n)
2:     reputation ← R_n − 0.5
3:     if reputation > 0 then
4:         reputation ← max(0, reputation − Rd_n)
5:     else
6:         reputation ← min(0, reputation + Rd_n)
7:     end if
8: return reputation + 0.5
9: end function
```

3.3 Network-Wide Consensus

Now armed with the capability of computing the trustworthiness of network nodes, we must turn our attention to the matters of behavioural analysis and result distribution. As stated, the reputation is calculated based upon the number of *good* and *bad* actions perceived during the validation process. However, although these distinctions are generally clear to us humans, IT devices need a little more help in their determination. As such, we start by presenting a the principles used to determine this distinction, as well as the methods employed to confirm the activities in real-time. Finally, we present the two stage methodology used for validating the analysed behaviour and distributing the resulting values through the network using blockchain technology.

3.3.1 Role Distribution

When participating in routing activities, there is generally only two distinctions: either the node is a router, or not such as the source or destination. However, to allow the capability for behavioural analysis, the exact actions of each node in the network must be analysed as they occur. Thankfully, IoT networks leverage wireless technologies for communications purposes, allowing data exchange between distant devices without the need for physical networking infrastructure. That being said, the main advantage of these technologies is also one of its weaknesses. Indeed, by using radio waves traversing through the open environment, they are susceptible to external stimuli.

This is the same principal which impacts human vocal communications. Thanks to our voices, we are capable of exchanging information much quicker and more efficiently than using written methods. However, if another discussion were to occur directly adjacent to the first, or a loud siren sounding off, our capability of hearing our interlocutor's voice is severely hindered, or even blocked. Furthermore, since voices can carry, it is also possible to overhear a conversation between two people for example sitting at a bar, all the while sitting in a booth on the other side of the room. To allow our system to analyse the behaviour of routing nodes, we take advantage of the latter, granting the ability for nodes to eavesdrop on the routing actions performed around them. However, this leaves two problems to be resolved:

1. Which nodes can perform this observation.

2. How to separate *good* actions from *bad*.

To answer these problems, we propose a new routing role called *Miner*. Inspired from the miners utilised in blockchain to validate and distribute blocks, we adapt their functionalities to allow behavioural validation and result distribution. Algorithm 2 presents the sequence of steps used to resolve determine which nodes can be considered *Miners*, as well as the construction of *Route Validation Tables (RVTs)*, utilised for behavioural validation. These RVTs are an important part of the validation process, providing the necessary information to the Miners for validation. Indeed, by providing the exact sequence of hops needed to travel in either direction along the route, the Miner's can be sure of the intentions of each relay node.

Algorithm 2 Miner Selection run at node n_i upon reception of a routing control or discovery packet for route $[src \rightarrow dst]$

```
1:  n_j ← packet source
2:  n_{j-1} ← packet destination
3:  n_{j+1} ← next hop for n_j towards dst
4:
5:  if n_{j-1} = me or src = me then                ▷ n_i is part of route
6:      Set role_{src→dst} as Router
7:      Drop RVT entries for route
8:      Add n_j as next hop along route
9:  else if dst ≠ me then                  ▷ n_i is a neighbour of a Router
10:     if role_{src→dst} = Router then          ▷ Already a router for route
11:         Drop packet and exit
12:     end if
13:     Set role_{src→dst} as Miner
14:     Add new forward and reverse RVT entries
15:     Add n_{j-1} as reverse hop towards src
16:     Add n_{j+1} as forwards hop towards dst
17: end if
```

To resolve the first issue, we can utilise the existing routing protocols to our advantage. Since the objective of our work is to identify trustworthy nodes to partake in routing, we can utilise routing control information to help determine role distribution, here shown for the route $src \rightarrow dst$. However, as stated the *Miner* role is a separate new addition to the routing process, meaning it is exclusive with the other two for a given route. As a result, a node cannot take on multiple roles for a single route but can only be either a router, a Miner, or simply not participate in routing at all. However, as is already the case where nodes can participate in multiple routes at once, nodes can also take on multiple roles, as long as they are each for individual routes. Thanks to the previously granted eavesdropping capabilities, nodes can pickup and analyse passing routing information from neighbouring devices participating in routing. In doing so, we grant network nodes the capabilities to determine their own role in routing.

However, the previous role information is not the only treasure hidden in the routing protocols discovery packets. Indeed, with route determination being a dynamic process, various routing protocols utilise different methods to discover routes to their destination. However, since they all utilise the wireless medium, their contents can be analysed further

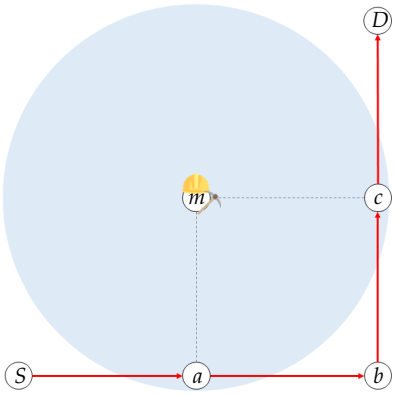


Figure 3.9: Network example to illustrate the construction of the RVT entries in Table 3.1 from the point of view of m whose communications range is represented in light-blue.

Table 3.1: Contents of both forwards and reverse RVT tables for route $S \rightarrow D$ from the point of view of Miner m in the network presented in Figure 3.9

Forwards	Reverse
a ⟶ b	c ⟶ b
c ⟶ D	a ⟶ S

by the new Miners, thus answering our second problem. By analysing the routing information provided by a specific node, it is possible to determine to whom they would forward packets along a specific route. As a result, Miners can construct RVTs, containing the layer two MAC address of the expected hop in both directions, back towards src via n_{j-1} and forwards towards dst via n_{j+1}. By concatenating all overhead routing information for a specific route, it is possible to construct the exact sequence of hops needed to route data from src to dst through the Miners neighbouring nodes. Table 3.1 shows the RVT for the route $S \rightarrow D$ as seen by Miner m, represented in Figure 3.9. We can see that, even though only two relay nodes are in communications range, by following Algorithm 2, m is capable of extracting the necessary information and constructing both forwards and reverse hop sequences. Here, the MAC addresses are utilised as the source and destination IP addresses would naturally reflect the source and destination of the requested route, leaving only the MAC addresses to identify the different point-to-point exchanges.

3.3.2 Observational Validation

Now armed with the capability to determine the correct behaviour of their neighbouring nodes, each Miner can start the validation process. As stated previously, this process is split into two distinct phases: behavioural validation and result distribution. Each stage possesses its own objectives and specifications, each important towards the correct validation and sharing of routing activities. Naturally, before any results can be shared throughout the network, they must first be determined by to the Miners.

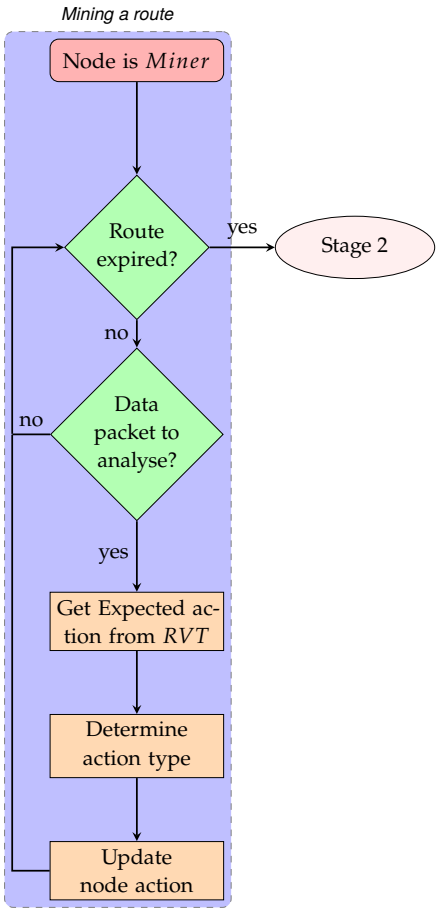


Figure 3.10: Validation flowchart - Stage 1

The objective of this validation process, also called *mining a route*, is to observe the routing behaviour of each neighbouring node and identify each forwarding action as *good* or *bad* Furthermore, we also grant the capability of a more extensive analysis, allowing the Miner's to also analyse the contents of each packet, easily identifying of it was modified by the router. Figure 3.10 shows the flowchart process of this first stage.

Since determined routes all possess a specific lifetime, the validation process also runs for the same duration, allowing a single process to take place for all routing activities utilising the same established route. This means that the process can either result in a single observation or hundreds, depending on the amount of data needing forwarding. In any case, each Miner observes the surrounding wireless medium for any data packet belonging to the observed route. When a packet is observed, the miner computes a hash value of the data packet. This hash not only allows the verification of data integrity, but also serves to recognise previously seen packets using a passing packet buffer. By adding a new entry into the buffer buf_{pkt}, the Miner can follow the specific packet on its journey through its neighbours.

If the received packet isn't already listed as invalid in the buffer, then the corresponding RVT is extracted from the Miner's onboard route table. By storing the RVTs in a route table, the Miner is also capable of validating not only the route $src \rightarrow dst$, but also any response from $dst \rightarrow src$. However, it is important to note that since the RVTs are associated to a specific route, only packets explicitly from src towards dst can be validated. It is, therefore, not possible to validate the routing

activities of nodes forwarding packets from an intermediate node which took advantage of the pre-existing route to dst. In any case, if the Miner has no record of the route $[src \rightarrow dst]$ or the requested RVT is empty, the transmitter is considered to be performing a replay attack, and its activities automatically identified as malicious. As a result, the number of *bad* actions for the transmitter bad_{llsrc}, identified by its MAC address $llsrc$ is incremented by one, also setting the packet buffer entry buf_{pkt} as invalid.

If on the other hand all is well, the next hop is extracted from the RVT and compared to the MAC destination address of the packet $lldst$. If these two differ, then the relay node is considered to be performing a redirection attack and considered malicious, incrementing once more bad_{llsrc}. If not, then the node is considered to be expressing valid behaviour and is treated as such, incrementing the number of *good* actions $good_{llsrc}$ by one.

Once the observed route is no longer used and expires, the Miner's prepare for the next phase. To do so, they perform one final verification on their observations, in particular the packet buffer entries. As stated, a packet's buffer entry bug_{pkt} allows the Miner to follow the packets journey along the route. As a result, it also allows the identification of lost packets. If, once the route has expired, there are still entries in the packet buffer which haven't reached the end of this portion of their journey, they are considered to have been destroyed by the last encountered hop and their number of *bad* actions is increase for each dropped packet. Once the final actions for each router have been totted up, the Miners clear their RVTs to allow the detection of replay attacks, and prepare for the block validation phase.

3.3.3 Result Validity Computation

Before we are able to distribute the results throughout the network, they first must be validated. We propose and define a consensus based validation mechanism, allowing Miners in proximity with each other to vet and confirm the work of their neighbours. The overall process is presented in Figure 3.11, following on from the first stage presented previously.

To begin, when a route expires all associated Miners collect and combine all observed actions into a temporary block ready to be shared across the network. To reduce the risk of collision or cross transmission, each Miner arms a random back-off timer. From this point, there are two possible outcomes: either the timer expires, or a block is received. If the former occurs, the Miner simply transmits their block in a *Verification* packet of type 2, the structure of which is presented in Figure 3.12, and awaits a response from another Miner. We can see here that each packet contains a unique type, identification number and timestamp. Here we use unique identification numbers combined with the timestamp of transmission to detect both replayed blocks, as well as in the case of crossover, ignore packets with older timestamps. To allow the Miners to validate the blocks contents, we provide the IP addresses of the route's source and destination, as well as the address of the originating Miner for identification purposes and the number of entries within the block.

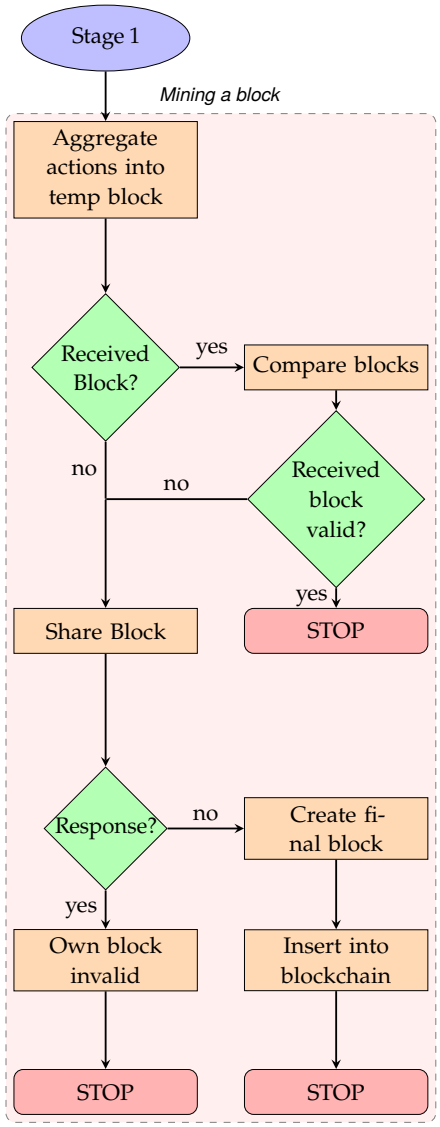


Figure 3.11: Validation flowchart - Stage 2

The block itself contains three elements: the observed node's IP address, the number of *good* actions and the number of *bad* actions.

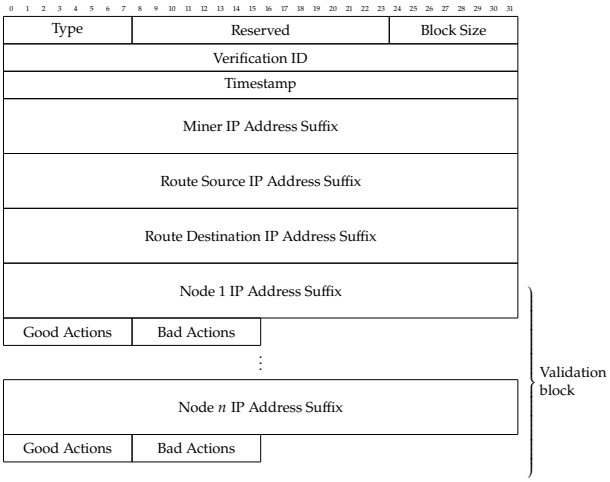


Figure 3.12: The structure of a *Miner Verification* packet used during the consensus phase.

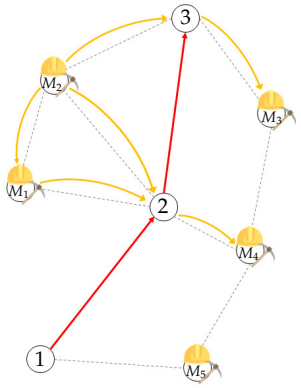


Figure 3.13: Example of Miner validation process where only miners in two-hop range can validate each other, corresponding to miners observing the same portion of the route. The block exchanged from miner M_2 is represented in **orange** for the route observed in **red**.

Since our objective is to validate the contents of the *Verification* packet's block, we can limit its transmission to only Miners in proximity to the same routing nodes. To achieve this, we set the Time To Live (TTL) field of the encapsulating IPv6 packet to 2, thus allowing the routing nodes to relay the packet onwards, reaching only the neighbouring Miners, as shown in Figure 3.13. Here, we can see that the block transmitted by Miner M_2 who has validated nodes 2 and 3, shown with the **orange** arrows, reaches all other miners except M_5, who is only in range of node 1, meaning it cannot validate M_2's block. Furthermore, we function on a "no-news–good-news" principle, where a response to the *Verification* packet identifies an error with the blocks contents. By combining these two approaches, we can effectively reduce the validation overhead, all the while permitting the validation of the blocks themselves.

It is possible that a Miner receives another block before their back-off timer expires, in which case their analysis is modified. Firstly, they analyse the contents of the *Verification* packet and compare them with their own block to determine their validity. Since Miner's can observe multiple routers at a time, a received verification packet may contain nodes unknown to the receiver. As a result, the analysis is only performed on the common nodes. To do so, the Miner determines a validity ratio, representing the percentage of actions contained therein which are confirmed by this Miner, which in other words are identical. By using a validity threshold, the Miners can determine if a block is to be considered valid or not. For example, a threshold of 80%, would allow a block containing *five* observations to pass validation with at most *one* incorrect node.

If the block is considered valid, the Miners perform a second computation, aiming to determine the efficiency factor of the received block. Since we wish to reduce the overhead as much as possible, it would be beneficial to insert as fewer blocks as possible into the blockchain. To determine this, the ratio of common nodes from the received block P_B and the Miner's own temporary block P_M is calculated, shown in Equation 3.7 and Equation 3.8 respectively, with B corresponding to the nodes in the received block and M those in the Miner's own.

$$P_B = \frac{|M \cap B|}{|M|} \tag{3.7}$$

$$P_M = \frac{|M \cap B|}{|B|} \tag{3.8}$$

By comparing the resulting values, we can determine which block is more efficient than the other. In this case, if $P_B < P_M$, the Miner's own block is considered more efficient. This means that the Miner will transmit its own block in response to a received block in two situations:

1. The received block's validity ratio is below the acceptable validity threshold, rendering the block invalid.
2. The received block is valid, but its efficiency ratio is lower than the Miner's own block.

By transmitting its own block, the Miner considers itself being valid, thus waiting for a response from another miner to either overrule its block, or of not, to then insert its block into the blockchain. Furthermore, by keeping an eye on the nodes in its contained block, the Miner's are capable of determining which have been overridden by other Miners and which haven't. If this is the case, the Miner considers itself as the *ultimate authority* for these missing nodes and prepares to insert them into the blockchain. It is also impossible to note that a Miner can only transmit its *validation* block once during the validation phase, thus avoiding infinite loops, or Miners trying to force their blocks thanks to the identification of the Miner's IP address. An in-depth example of the validation phase based upon two scenarios and topologies is available in Appendix B.

Consensus Formalisation

As stated previously, the validation phase revolves around either the transmission of a block, or the reception of another. As a result, the outcome of the validation phase varies, dependant on the outcome of this first step. Indeed, depending on which Miner wakes up and transmits their *Verification* packet first, decides how the network will react. As such, predicting the exact outcome, based upon the first Miner is difficult.

To help in this, we have represented this phase using Markov Chain Theory. The corresponding probability graph is presented in Figure 3.14 and shows the possible sequence of events as well as the different Miner states.

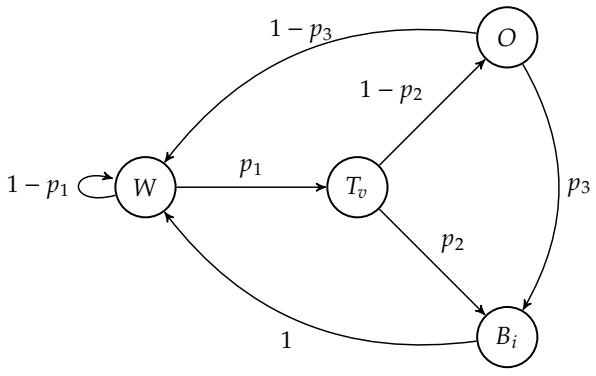


Figure 3.14: Formalisation of the consensus algorithm using Markov Chain Theory to represent the changes between the different stages shown in the Figure 3.11

Here we define four possible states:

- **Waiting (W)**: The phase in which all nodes commence at the start of the validation phase, corresponding to Miners waiting for either a block to be received they can override, of for their back-off timer to expire.
- **Transmit Validation (T_v)**: The Miner transmits their temporary block for validation and awaits a response.
- **Overridden (O)**: The Miner has received a response to their initial *validation* packet, overriding it and marking it as invalid.
- **Block Insertion (B_i)**: The Miner contains nodes in its block which haven't been overridden and are, therefore, inserted into the blockchain.

From here, we can define the state transition matrix as the following:

$$P = \begin{Vmatrix} 1-p_1 & p_1 & 0 & 0 \\ 0 & 0 & 1-p_2 & p_2 \\ 1-p_3 & 0 & 0 & p_3 \\ 1 & 0 & 0 & 0 \end{Vmatrix} \tag{3.9}$$

When performing validation, Miners present in state W, can either pass to T_v or remain where they are. The probability of passing onwards, p_1 is defined as follows:

Proposition 3.3.1 *Let all variables be defined as:*

- $B_{received}$ = *block received*
- Δt_1 = *maximum back-off timer for Miner wake-up*
- B_{mined} = *Miner's block*
- Δt_2 = *validation timer*
- n = *a node*

Definition 3.3.1 *Let p_1 be the probability of transitioning from state W to state T_v where:*

- $\nexists B_{received}$ *since* Δt_1*; OR*
- B_{mined} *efficiency* $>$ $B_{received}$ *efficiency; OR*
- $\nexists B_{received}$ *since* Δt_2 *AND* $\exists n \in B_{mined}$ *where* $n \neq$ *confirmed*

If one of these probabilities is true, then the Miner transitions from W to T_v, otherwise they remain where they are.

Once in state T_v, the Miner transmits its block for validation. Here there are two possibilities: either the Miner transitions into B_i and injects its block into the blockchain, or it moves into O and is overridden by another. The probability of passing into B_i, p_2 is defined as follows:

Definition 3.3.2 *Let p_2 be the probability of transitioning from state T_v to B_i where:*

- $\nexists B_{received}$ *since* Δt_2

In this case, if no block is received, the miner transitions into B_i, otherwise they move into O.

Once the Miner injects their block, they automatically return to their waiting state W. However, if the miner has been overridden and moved into state O, they once again have two possibilities: inject its block into the blockchain by transitioning into B_i, or returning back to the waiting state W. The probability of passing into B_i, p_3 is defined as follows:

Definition 3.3.3 *Let p_3 be the probability of transitioning from state O to state B_i where:*

- $\nexists B_{received}$ *since* Δt_2*; AND*

- $\exists n \in B_{mined}$ *where* $n \neq$ *validated*

If both of these conditions are met, then the Miner injects its block into the blockchain in state B_i, otherwise they return back to their waiting state W.

It is noticeable that probability p_1 contains a condition which is the same as p_3. Indeed, both transitions depend on the reception of a block within a specific time frame, where not all of the Miner's nodes have been overridden. However, the more time passes during the validation phase, the more this probability fluctuates. Furthermore, if this probability was determined as *false* during p_1, it is possible for it to become *true* during the second, due to the evolution of the Miners in the vicinity.

3.3.4 History Dissemination

The blockchain is formed of a list of "blocks", linked together by each block containing the hash of its predecessor. Each block contains a data payload containing the list of validated transactions. The validation is performed by "miners", specific devices who confirm the data through a cryptographic computation called Proof-of-Work (PoW).

Thanks to the Miners, we now possess the list of node actions in "block" form. To allow all nodes to accurately determine the trustworthiness of their neighbours, they all must possess the valid list of actions for these nodes. For this, we utilise blockchain technology due to its secure data sharing capabilities. Furthermore, we redefine the PoW mechanism, replacing it with our aforementioned validation process, allowing the confirmation of observed activities, all the while minimising the impact on the nodes themselves. Thus, once all actions have been parsed, analysed and validated, the selected Miners prepare to distribute their blocks throughout the network. To do so, they construct a *Share* packet with type 1, the format of which is presented in Figure 3.15.

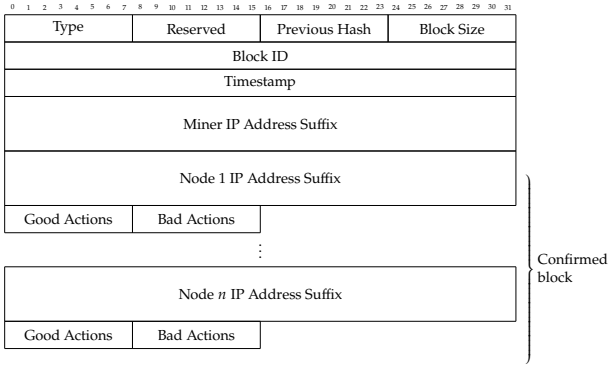


Figure 3.15: The structure of a *Miner Share* packet used to insert a block into the blockchain.

Following the format used with the *Verification* packet, the share packet also contains a type field as well as the IP address of the originating Miner. Furthermore, it also contains the block size, indicating how many nodes are contained within the block, as well as a unique identifier and the timestamp of the transmission. The main difference with the *Verification* packet is the addition of a previous hash field. Indeed, as presented above, blocks in the blockchain include the has of its predecessor, allowing to enforce and confirm the immutability of the chain. Finally, the confirmed block itself follows the same format as the validation block, containing the IP address of each confirmed node, along with the number of *good* and *bad* actions.

This packet represents the format which is inserted into the blockchain and subsequently that which all nodes in the network can parse and extract the contents to update their local action caches. This means that the full history of each node is effectively stored in the blockchain, whereas

each individual node only keeps a snippet to compute a nodes reputation. As a result, if a node wishes to increase the history window W_n, it is possible to extract the previous actions from the blockchain, allowing the nodes to keep an accurate overview of all routing activities.

3.4 Theoretical Observation

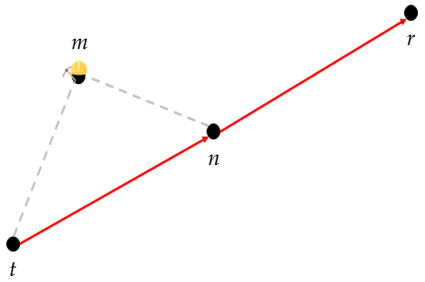


Figure 3.16: Network topology used to evaluate the evolution of the reputation of node n using Miner m when participating in routing activities for the route $t \longrightarrow r$, shown in **red**

To grasp the concept of node reputation, we simulated a network of four nodes using the Cooja simulator for Contiki-NG, previously defined in Chapter 4.4.1. Each simulation ran for *15 minutes*, with one transmitter t, one receiver r, one malicious relay node n and one observational Miner m. The topology used is shown in Figure 3.16 and is coincidently the one depicted in the simulator in Chapter 4.4.1. Thanks to this network, we can evaluate how the reputation of node n is impacted based upon its actions over time. To achieve this, t transmits five packets at two second intervals every minute, commencing 30 seconds after the start of the simulation. This provides ample time for the Miner to perform its validation phase and distribute its block before the next transmission. For the duration of these simulations we set the window size W_n to 10, taking into account the last 10 actions at all time. This means that with an update to the reputation every minute, the activities achieved at during the first minute, would be lost when reaching the eleventh. Furthermore, we define the malicious weight $\alpha = 2$, doubling the weight of malicious actions compared to good actions as explained previously To provide a complete overview, we vary n degree of malicious actions, expressed as the percentage of malicious packets dropped, allowing to compare how the reputation is impacted. This study is shown in Figure 3.17.

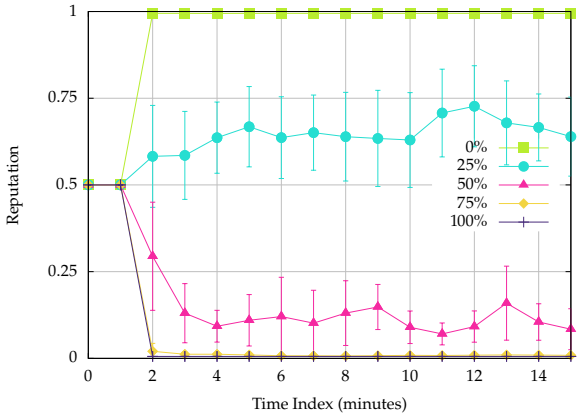


Figure 3.17: Evolution of a nodes reputation over-time dependant on the degree of malicious intentions. These results were achieved with $\alpha = 2$

For this analysis we varied the malicious intentions of n in stages of 25%, from a perfect router (0%) to a representative of the dark side (100%). As expected, when no attack is performed, represented with the **green** squares, the reputation rises to 1 and stays there for the entire 15 minutes. As the malicious probability increases, however, the resulting reputation falls with it. When dropping one packet out of four, shown with the **light-blue** circles corresponding to 25% malicious intentions, the reputation decreases to around 0.6. We can also see that the reputation is not stable, fluctuating with the varying degrees of malicious actions performed by node n. However, when the malicious probability reaches 50%, shown with the **pink** triangles, the reputation has almost reached

rock bottom, resting at around the 0.15 mark. When the node rises towards peak maliciousness, with 70% (**yellow** diamond) and finally 100% (**purple** plus), the reputation bottoms out, indicating there is no distinction between the two.

Thanks to this analysis, we can see how the reputation evolves overtime, based upon the malicious actions of node n. However, we defined $\alpha = 2$, immediately increasing the weight of malicious actions. To illustrate the impact α has on the reputation, we vary its value using the same parameters as shown in Figure 3.6, while keeping a malicious activity rate of 25%. This evaluation is performed in Figure 3.18.

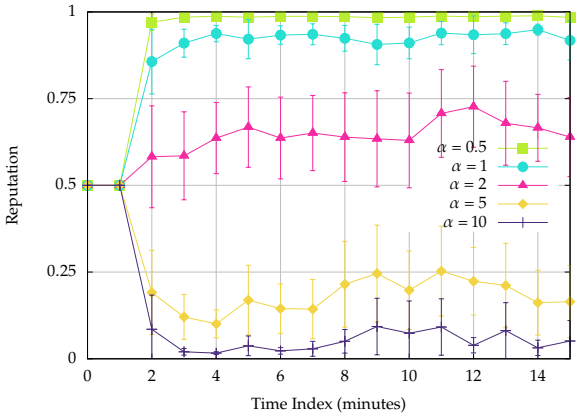


Figure 3.18: Study of the importance and impact of α on the evolution of a nodes reputation overtime. These results were achieved with a malicious activity degree of 25%

First off, we can see that the values for $\alpha = 2$, represented with the **pink** triangles, corresponds to those from 25% malicious activities in Figure 3.17, shown with the **light-blue** circles. This confirms the association between the two graphs. By varying the value of α, we can either increase or decrease the impact of the malicious actions. By looking at the results for $\alpha = 1$ (**light-blue** circles) and $\alpha = 0.5$ (**green** squares), we can confirm that the malicious actions taken by n cause less disruption, wit the reputation remaining generally above 0.85, reaching near maximum with $\alpha = 0.5$. Consequently, the opposite can be observed with α is increased, with $\alpha = 5$ (**yellow** diamonds) and $\alpha = 10$ (**purple** pluses), where the reputation drops to below 0.25.

As a result, we can conclude that the value of α actively influences the the reputation of a node whose intentions are to disrupt traffic. However, even with a static value of α, the level of malicious intent expressed by the disruptive node is easily distinguishable. That being said, the higher the value of α, the more the reputation drops, where a node expressing a malicious activity of 25% with $\alpha = 10$, possesses a lower reputation than a malicious node with 50% malicious activities and $\alpha = 2$. To increase the impact of malicious actions, we decided to remain with $\alpha = 2$, thus allowing nodes to be impacted by their past choices, all the while giving them a chance to return back.

3.5 Conclusion

In this chapter, we explored the notion of trustworthiness, applying its characteristics to IoT networks using the notion of reputation. We explored this notion and proposed a metrics to compute a network

nodes reputation, based upon a customisable portion of its history W_n. Furthermore by allowing malicious actions to increase or decrease their impact upon a nodes standing through the notion of malicious weight α, we allow for easy customisation of the reputation function. To be able to determine a devices reputation, their behaviour must also be analysed. We, therefore, propose a new type of "role" in the routing process called "Miner", inspired from blockchain. These Miners analyse the actions taken by neighbouring nodes participating in routing, identifying $good$ and bad actions from their Route Validation Table (RVT). To allow intra-Miner validation, we propose consensus algorithm, allowing Miners to agree amongst themselves as to the validity of their findings. Using this method, we allow only validated behavioural information to be shared amongst the network, confirming the computed reputational values. Finally, we levy blockchain technology to contain and distribute this behavioural analysis throughout the network, replacing its generic PoW with our consensus algorithm. As a result, we not only distribute correct, validated information but also provide a secure and transparent medium for information sharing.

Routing Protocol Integration 4

Knowing the route to take to reach a destination is important, not only in IT but also in our daily lives. To solve this, we utilise maps and GPSs, allowing us to determine the best route possible. However, these maps all rely on existing data, meaning they are already aware of all possible routes to get from every and any point, to every other. This is the case of internet-based routing, where all routers along the internet's back-bone are aware of their different links available and which links lead to which destination.

When it comes to ad hoc multi-hop networks, however, omniscient routing isn't always possible. Indeed, in some cases devices can be preloaded with routing information, but it is generally impractical and difficult dependant on their specific use case. As a result, many different types of multi-hop routing protocols have been proposed over the years, all providing different advantages and using different methods, such as "I want to know all now" or "I will search when needed". Furthermore, these protocols utilise different path selection algorithms, allowing them to determine the best path for them, based on their specific criteria, which can be influenced to incorporate new metrics without changing the underlying protocol.

In this chapter, we present an overview of different types of multi-hop routing protocols, defining how they function in these networks. We complete this overview with an analysis of the path selection algorithms utilised and proposed in the literature, before proposing an integration metric based upon a nodes reputation as presented in Chapter 3. Finally, we provide a detailed overview of how this new metric, called *link-cost* would integrate with two *Reactive* routing protocols (AODV and DSR). We then demonstrate its overall efficiency through in-depth evaluations during multiple simulations. We conclude this overview with a theoretical analysis of how this metric could be adapted and integrated into other protocol types, such as *Proactive* routing, in particular with RPL.

4.1 Multi-Hop Routing

In multi-hop networks, a direct route from source to destination doesn't always exist. Contrary to our everyday one-hop network, such as our personal Wi-Fi networks where every device is directly in communication with the base station, here the access point, this is not the case in ad hoc networks. Indeed, here devices rely on each other to relay information from one side of the network to each other, passing the data onwards towards the destination. These intermediate devices basically take on the role of routers and are responsible for ensuring their entrusted data goes where it is needed.

For these routers to know where to transmit their data they must first be informed where the destination can be found. The routers making up the back-bone of the internet all possess very complete and complex

routing tables, indicating to them in which direction they must relay their entrusted data. However, these routers are generally administered, providing the routers with the necessary information allowing them to make the right decision. This is not the case in multi-hop wireless ad hoc networks, where each device is left to its own accord and must learn about its surroundings in order to participate in networking activities.

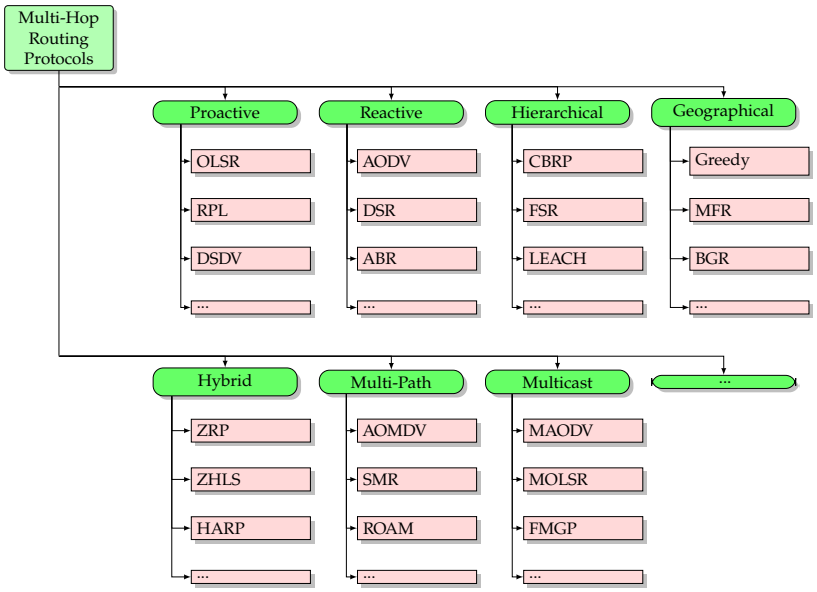


Figure 4.1: Overview of some multi-hop routing protocol categories as well as some examples of protocols for each category. Not all categories have been depicted

To allow devices to exchange data in these networks, many different types of routing protocols have been proposed over the years [192]. Figure 4.1 presents some of these protocols, organised into different categories, dependant on their underlying operation. In this section, we provide an overview of the two most common and well know protocol types: *Proactive* and *Reactive*.

[192]: Eiman Alotaibi and Biswanath Mukherjee. 'A survey on routing algorithms for wireless Ad-Hoc and mesh networks'. In: *Computer Networks* 56 (2012). DOI: 10.1016/j.comnet.2011.10.011

4.1.1 Proactive Routing

Proactive routing protocols are some of the most well known and explored in the scientific and networking communities. Also called *table-driven*, these protocols utilise routing tables upon each node to store route based information as to how to reach individual nodes. However, these tables must be both populated and maintained on a regular basis, so as to allow the network to remain connected and packets to reach their intended destination. To achieve this, *Proactive* protocols perform active searches, distributing routing information to their neighbours, informing them of who they can reach. From this, each neighbour can update their own table entries to reflect this new information, keeping individual entries for each network node.

Although this is their advantage, individual routing tables is also one of their weaknesses. Indeed, due to the one-hop table updates, nodes must constantly share table changes, which not only take time to be propagated throughout the network, but also significantly increase the traffic overhead. In many cases, these "fresh" table entries are not even used, as not every node required to contact every other in the network. Furthermore, table scaling is another issue, where large networks result

in significantly heavy routing tables, taking up valuable memory reserves on each node.

Here, we present two *Proactive* routing protocols: OLSR and RPL.

OLSR

The Optimized Link State Routing Protocol (OLSR), defined in *RFC 3626* [193] from 2004, utilises periodic message exchange to discover neighbouring nodes. These "*hello*" messages provide important information regarding the communication capabilities of all neighbours, including the status of the link between the transmitter and receiver. However, these messages also serve another purpose, allowing the nodes to determine the most optimal and efficient relay nodes in its neighbourhood.

By utilising this *link-state* approach, the "*hello*" messages also contain the list of neighbours of the transmitter, thus providing the list of two-hop nodes to all receivers and so on. From here, each node determines, thanks to an optimisation algorithm, the lowest possible number of one-hop neighbours to reach all two-hop neighbours, called MultiPoint Relays (MPRs), as shown in Figure 4.2. These MPRs provide not only flooding overhead reduction, allowing control packets to be forwarded in the most efficient way, reaching all nodes in the network with the lowest number of transmissions, but can also help with routing.

By sharing detailed topology information through the use of "*Topology Control (TC)*" packets associated with the data from the neighbourhood discovery, each node can construct routing table entries for every and all nodes in the network. Indeed, each table entry indicates to which node the data packet must be forwarded to reach the intended destination, along with its distance corresponding to the number of hops. With the proactive nature of OLSR, new links can be created or lost, thus causing the routing table to constantly evolve, reflecting the current network topology as best as possible.

[193]: Thomas H. Clausen and Philippe Jacquet. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626. 2003. DOI: 10.17487/RFC3626

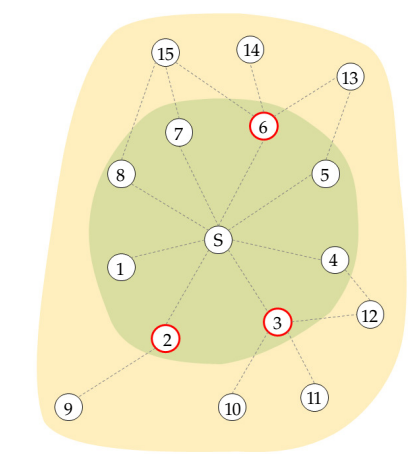


Figure 4.2: Network discovery using OLSR with the selection of MPRs (**red** outline) in one-hop nodes (**green**) to reach all two-hop nodes (**orange**)

RPL

Compared to OLSR, the Routing Protocol for Low-Power and Lossy Networks (RPL) functions slightly differently. Defined in *RFC 6550* [194] in 2012, RPL is a distance vector-based protocol for IEEE 802.15.4 networks and is the standard protocol used by Contiki-NG. Supporting a wide range of operational modes including many-to-one but also one-to-one, RPL can create up-to-date network routes by continuously exchanging path related information. To forward data on-wards, RPL creates logical tree-like topologies, called Directed Acyclic Graphs (DAGs), as shown in Figure 4.3.

Each DAG is comprised of one or more Destination-Oriented DAGs (DAGs), each with a specific root node called sink. Within these DODAGs, each node is assigned a specific rank, corresponding to the distance in hops between itself and the DODAG root. Whereas OLSR was developed with IPv4 in mind, RPL utilises various types of Internet Control Message Protocol for IPv6 () control packets for both tree definition and path selection. Each topology is created and maintained via DODAG

[194]: Roger Alexander, Anders Brandt, JP Vasseur, Jonathan Hui, Kris Pister, Pascal Thubert, P Levis, Rene Struik, Richard Kelsey, and Tim Winter. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. 2012. DOI: 10.17487/RFC6550

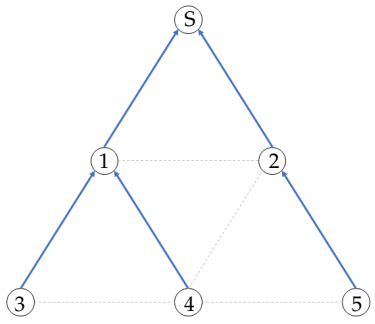
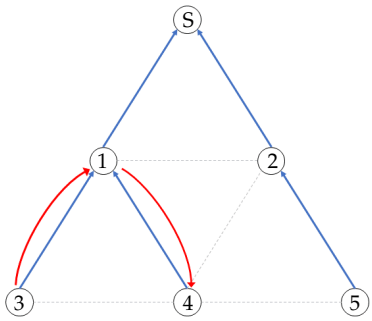


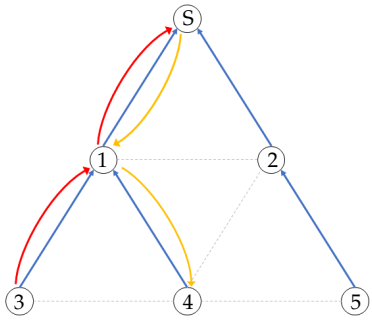
Figure 4.3: Representation of a DAG RPL network topology where the **blue** arrows represent the child-parent relationship

Information Object (DODAG) control packets which are produced by each node, containing a routing metric, such as link quality or energy reserves, and an indicator process for the selection of the parent node from its neighbours. Further to this, RPL also provides the capability for nodes to "announce" themselves to the root node, identifying them as a potential destination through the use of Destination Advertisement Object (DAO) control packets, which are propagated throughout the network.

From this information, parent nodes can construct downwards routes back to the source, however, in the context of IoT devices, memory is often a limiting factor. To resolve this, two operating modes have been included, *storing* and *non-storing*, each impacting the overall functionality of the DODAG as illustrated in Figure 4.4. In *storing* mode, each parent creates a routing table entry for each DAO received from its sub-DODAG, allowing it to respond directly, as shown in Figure 4.4a. On the other hand, in *non-storing* mode this is not the case, and packets must be returned all the way to the DODAG root which uses *source routing* techniques to include the expected path to the intended destination directly into the IP header, as shown in Figure 4.4b. With these two modes, there is a trade-off for the saved memory storage, with regards to network overhead where some un-necessary links have been used.



(a) RPL *storing* mode with the **red** arrow representing the routed packet



(b) RPL *non-storing* mode with the **red** arrow representing the routed packet to the sink and the **orange** arrow the *source route* from the sink to the destination

Figure 4.4: Comparison of the two RPL modes for a packet transmitted from node 3 to node 4

4.1.2 Reactive Routing

Another well known protocol type is that of *Reactive* routing, otherwise known as *on-demand*. These function with the opposite premise to *Proactive* protocols, where instead of actively maintaining routes to all other nodes, paths are only determined if and when they are needed. To achieve this, when a path is requested towards a destination, these protocols use a route discovery process to determine the sequence of hops needed to reach the target node. An advantage to this approach, compared to *Proactive* routing, is that routes can quickly adapt to changing topologies, with no need to wait for routing tables to be updated, allowing them to function in dynamic environments.

That being said, this approach means that route discovery takes place every-time a new route is needed, increasing the transmission delay until the discovery process is completed. However, with the reduced need for constant route information exchange, the routing overhead is considerably lower than *Proactive* methods. This does also add the advantage of being scalable, since route information is only stored when it is needed for a short duration, allowing for much longer routes in much larger networks.

In this section, we present two well known *Reactive* protocols: AODV and DSR.

AODV

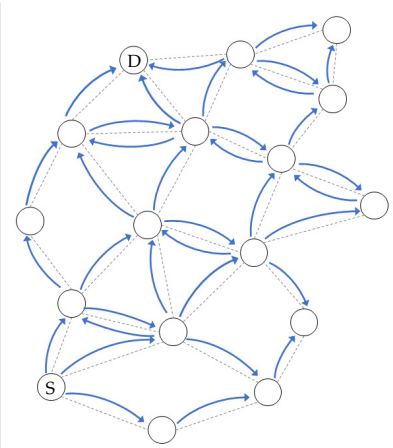
The Ad hoc On-Demand Distance Vector (AODV) routing protocol is probably one of the most known and explored multi-hop protocols to date. Originally proposed in 1999 and defined in *RFC 3561* [195] in

[195]: Samir R. Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. 2003. DOI: 10.17487/RFC3561

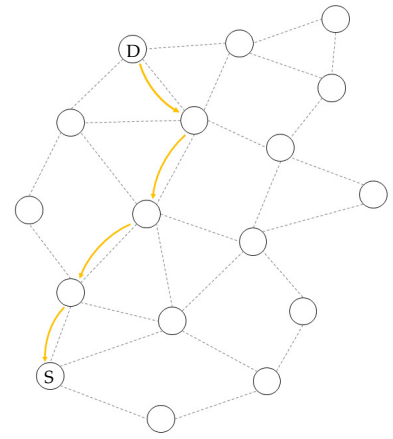
2003, AODV relies on route discovery and maintenance methods to both determine available paths, as well as their continuous availability. Its main functionality is that of route discovery which consists of flooding the network with Route Requests (RREQs) in an attempt to identify an available path towards a requested destination. Upon receiving such a message, intermediate nodes simply relay RREQs onwards, in accordance with their path selection algorithm, propagating the route discovery throughout the network in a wave, as shown in Figure 4.5a.

When the requested destination receives the RREQ, it responds back to the request originator, considered as the route's source, with a Route Reply (RREP). Contrary to RREQs, RREPs are sent via unicast, travelling along the reverse route all the way back to the source node, illustrated in Figure 4.5b. Along the way, each and every intermediate node creates two routing table entries, indicating the next hops needed to transmit a packet in either direction. As stated, each node only keeps the routes as long as they are valid, the lifetime of which is directly extracted from the received RREP and provided by the route's destination.

With this approach, upon receiving a data packet, nodes can extract the corresponding route from their tables and transmit the data to the next hop, allowing it to jump across the network. However, it is possible that an intermediate node either looses its connection with the next hop, due to possible mobility, or the entry simply expires, at which time an Route Error (RERR) is returned back to the source and the discovery process recommences again. Originally proposed with IPv4 similar to OLSR, AODV has received a draft proposition for an update towards IPv6-based networks [196], allowing it to function with modern IoT devices.



(a) Flooding of RREQ packets



(b) Unicast response with RREPs

Figure 4.5: Route discovery process with AODV for a route between *S* and *D*

DSR

Following on the same concepts as AODV, the Dynamic Source Routing (DSR) protocol introduces some specific characteristics which change its overall functionality. Proposed in *RFC 4728* [197] in 2007, DSR utilises RREQ and RREP packets to discover routes between two distant nodes. However, contrary to AODV, DSR doesn't employ intermediate routing tables to determine to whom packets must be passed at each step, but instead utilises the notion of *source routing*. This process consists of including the sequence of hops directly into the packet itself, essentially providing the exact directions needed to reach the intended destination.

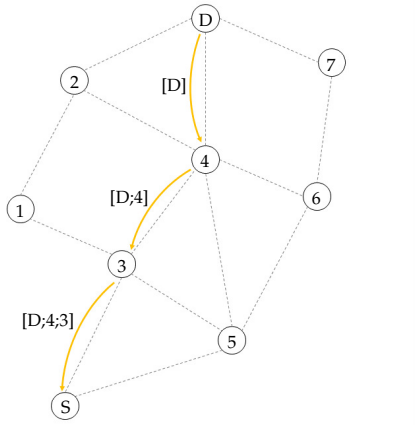
To achieve this, DSR takes advantage of IPv4 extension headers [198], providing the necessary information directly on the network layer, facilitating routing. During discovery, all nodes function almost identically to AODV, relaying the packets onwards whilst following their path selection algorithm. However, instead of creating a routing table entry for the route, the node simply includes its own IP address directly into the RREQ header. The same notion is also applied to RREP packets which remain unicast back to the route originator, with each intermediate node including their own address directly into the header, as shown in Figure 4.6a.

With this principal, the source node can possess multiple possible routes towards their destination, allowing the choice of route dependant on their specifications. Thus, when the source wished to transmit a packet

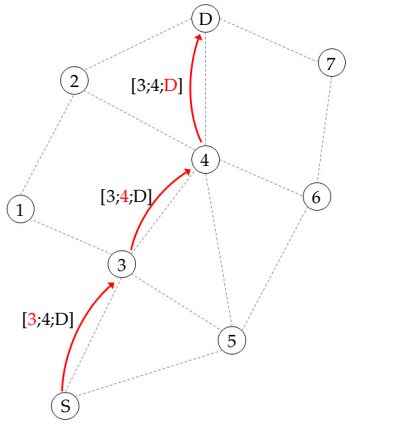
[196]: Charles E. Perkins and Elizabeth M. Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing for IP version 6*. Internet-Draft draft-perkins-aodv6-01. Work in Progress. Internet Engineering Task Force, 2001. URL: https://datatracker.ietf.org/doc/draft-perkins-aodv6/01/ (visited on Oct. 17, 2022)

[197]: Yih-Chun Hu, Dave A. Maltz, and David B. Johnson. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728. 2007. DOI: 10.17487/RFC4728

[198]: Jon Postel. *Internet Protocol*. RFC 791. 1981. DOI: 10.17487/RFC0791



(a) RREP response with hop-by-hop information



(b) Data packet with the *source routing* header

Figure 4.6: Route discovery between *S* and *D* using DSR and the utilisation of *source routing* in the data packet based on the hop-by-hop contents of the RREP

to the destination, the exact sequence of hops, represented by their IP addresses, is included in a *source routing* IP header, which is analysed on each hop, as demonstrated in Figure 4.6b. Since DSR functions using IP headers, it can also be adapted for use in IPv6 networks, since IPv6 possesses updated header formats, including source routing extensions allowing easy DSR integration [199].

4.2 Reputation-Based Route Selection

All routing protocols employ different path selection algorithms to determine the best path to their destination. In many cases, these are left at the discretion of the implementation, however, most protocols possess specific metrics to achieve their objectives, be them quality preserving, throughput protection or simply distance related. As a result, it is possible to replace or influence these algorithms to incorporate different metrics into the routing process, such as our previous reputational metrics from Chapter 3.

In this section, we present an overview of the route selection process, presenting some different methods used in routing protocols. We then propose an integration method based around a nodes overall cost, allowing their reputation to influence the path selection algorithm.

4.2.1 Route Selection Process

As we have seen previously, each routing protocol utilises various metrics to determine the best route towards the requested destination. The route selection process can depend on either network statistics, or from external stimuli. The authors in [200] present an improved route selection technique for multi-rate ad hoc networks. Here, their objective is to select a route which maximises both the reliability as well as the throughput, allowing data to be shared at an optimal rate. By allowing the network to determine and exploit links with the highest effective capacity, data transmission times can be reduced.

QoS is an important element in networking, as it can define which packets can be sent to who and how. This is the case with modern internet infrastructure, where packets of high importance are given priority over lesser importance. As a result, routing activities can also be influenced, as presented in [201]. Indeed, by incorporating a set of QoS constraints directly into path selection algorithms themselves, it is possible to privilege more efficient routes for high priority data flows. Furthermore, by categorising the type of data itself based on its importance and assigning it a priority, routers can make an informed decision as to which path. This approach can also be applied to other domains, such as vehicular communications as presented in [202] where the authors use QoS to determine how data is to be offhanded to roadside infrastructures. In this case, it is possible that low priority data is not relayed directly to the static infrastructure, based upon a number of factors, but instead transmitted to another vehicle to be relayed onwards.

Although there are many different methods to determine which route to take, the most common and most basic is to simply select the shortest

[199]: Bob Hinden and Dr. Steve E. Deering. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460. 1998. DOI: 10.17487/RFC2460

[200]: Baruch Awerbuch, David Holmer, and Herbert Rubens. 'High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks'. In: *Wireless On-Demand Network Systems*. Ed. by Roberto Battiti, Marco Conti, and Renato Lo Cigno. 2004. DOI: 10.1007/978-3-540-24614-5_19

[201]: Fernando Kuipers, Piet Van Mieghem, Turgay Korkmaz, and Marwan Krunz. 'An overview of constraint-based path selection algorithms for QoS routing'. In: *IEEE Communications Magazine* 40 (2002). DOI: 10.1109/MCOM.2002.1106159

[202]: Yasir Saleem, Nathalie Mitton, and Valeria Loscri. 'A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning'. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. 2021. DOI: 10.1109/IWCMC51323.2021.9498708

possible route. This is the case, for example, of AODV. Indeed, both AODV discovery packets (RREQ and RREP) contain a field called "Hop Count", which is incremented each time the packet is forwarded in either direction. The contents of the RREQ hop count field actively influence how the discovery packets are forwarded onwards, allowing intermediate nodes to relay only those containing the lowest number of hops. As a result, the destination node can respond along the route containing the lowest value. Thanks to this field, all intermediate nodes are also informed of the distance between themselves and both the route source and destination, allowing them to update their internal caches at will.

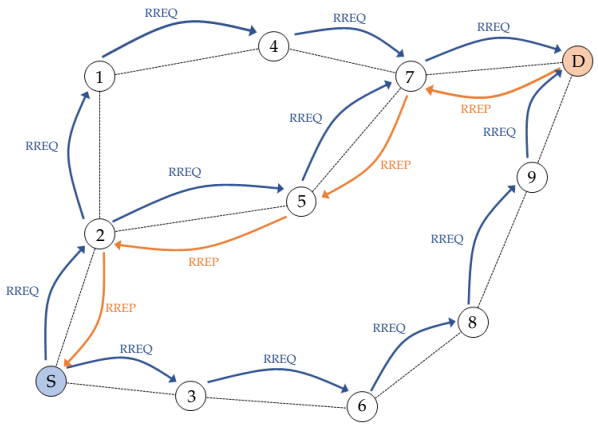


Figure 4.7: Route discovery based upon the shortest route length using a hop counter, as used in AODV

Figure 4.7 presents the discovery process used by AODV in a network with three possible routes of varying lengths. We can see that the RREQ packets are transmitted from the source node S to the destination D, hopping from node to node. However, the RREP response is only transmitted back along a single route, here via $7 \rightarrow 5 \rightarrow 2$ corresponding to the lowest number of hops possible.

4.2.2 Link-Cost Integration

Taking influence from how QoS can help determine the best route based on external criteria, the same concept can be applied to the notion of node reputation, previously presented in Chapter 3. However, in order to do so a reputation mapping methodology must be determined in order to convert the reputational values to a more easily usable format. To do so, we once more take inspiration from the authors of [176]. In this work, the authors propose an adaptation to AODV, where they use the computed reputation to influence the *hop-count* field in the RREQ and RREP packets.

[176]: Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450

When a node receives an RREQ or RREP, the normal functionality would be to increase the *hop-count* field by one, and relay the packet onwards. Along the way, packets with higher *hop-count* values are discarded, allowing only the shortest routes to be propagated. The approach in [176] proposes to update the contents of the *hop-count* field to no longer represent the route's length, but instead its overall *cost* in terms of trust, called *link-cost*. Computed based on a nodes reputation, the *link-cost* value effectively represents the inverse trust ratio represented as a

floating point value between $[0, 1]$, where a higher cost represents an untrustworthy route. By using artificial route shortening, i.e. assigning lower values to nodes with a good reputation, their approach is capable of influencing the generic AODV path selection algorithm. Indeed, by allowing it to continue selecting the lowest value from the now *link-cost* field, they are effectively selecting more reputable, sometimes longer routes. However, this computation possesses certain limitations, such as representing floating point values with a single byte, which results in a loss of precision. Furthermore, with the possibility of a cost of 0 at each hop, infinite discovery loops are a possibility, severely hindering the propagation of good routes throughout the network.

Our approach functions using the same basic principals, however, we propose an inverted computation method where instead of reducing the cost with higher reputation, we define a base cost which we increase the lower the reputation. To do so, we update the *link-cost* function using an adaptable scaling function, allowing a customisable level of precision, based upon the specifications and constraints of the routing protocol in use.

$$C_n = \lfloor (1 - R_{n_t}) \times (C_{max} - (C_{min} - 1)) + C_{min} \rfloor \tag{4.1}$$

We define C_n as the *link-cost* between node n and the current node, with R_{n_t} corresponding to the reputation of said node at time t post decay. Since R_{n_t} is normalised between 0 and 1, we can proportionately scale the reputation to our liking. We, therefore, define C_{min} and C_{max} as the minimum and maximum values for the resulting cost. By defining $C_{min} = 1$, we assure that there is always an increase in the *link-cost* for a route, even for a node with a perfect reputation, thus removing the aforementioned risk of infinite cost calculation loops. Finally, the cost is reduced to the nearest natural number, less than or equal to the calculated value.

As stated, we also provide the scaling function with the ability to adapt to the routing protocol at hand. This is achieved through the modification of C_{max}, effectively increasing or decreasing the precision of the *link-cost* function. Since some protocols, such as AODV embark the *link-cost* in their packet headers, the maximum value must also depend on the capacity of the corresponding field, reducing the risk of overflow.

$$C_{max} = \lceil \frac{2^{f_{size}} - 1}{L_{max}} - 1 + C_{min} \rceil \tag{4.2}$$

To adapt C_{max} to the specific case, we require two external variables: f_{size}, the storage field size in bits and L_{max}, the maximum possible route length (i.e., maximum number of hops). By calculating the relation ship between these two values, we can be assured that the total *link-cost* computed across L_{max} hops will reduce the risk of overflow, suddenly resulting in a very low cost route, tricking the route selection algorithm. To understand its significance, we will take a look at how it would adapt for incorporation with AODV. As stated previously, the *hop-count* field in AODV's RREQ and RREP packets is one byte in size. As a result, we can determine that the maximum value for the storage field is $2^8 - 1 = 255$. From here, if we define the maximum route length as $L_{max} = 32$, allowing at most for 32 nodes to participate in a single route, we would result in:

$$C_{max} = \lceil \frac{255}{32} - 1 + 1 \rceil = 8 \tag{4.3}$$

allowing for eight different values to represent each nodes reputation. On the other hand, if we increase $L_{max} = 64$ to correspond to the maximum value of TTL used in networking, the maximum cost would be:

$$C_{max} = \lceil \frac{255}{64} - 1 + 1 \rceil = 4 \tag{4.4}$$

effectively halving the precision of the cost function.

To illustrate this further, we can evaluate how C_{max} would be impacted if, for example, AODV stored the *link-cost* across two bytes instead of one. First off, we can determine the maximum value for this larger storage field is $2^{16} - 1 = 65535$, a significantly higher value than with only one byte. With this much larger field size, the maximum cost also increases. If we once more perform the calculation with $L_{max} = 32$, we end with a *link-cost* of

$$C_{max} = \lceil \frac{65\,535}{32} - 1 + 1 \rceil = 2\,048 \tag{4.5}$$

an increase in precision by a factor of 510, significantly increasing the precision C_n. This is confirmed when we double the maximum number of hops once more with $L_{max} = 64$:

$$C_{max} = \lceil \frac{65\,535}{64} - 1 + 1 \rceil = 1\,024 \tag{4.6}$$

where we once again see that C_{max} is halved in a similar fashion to Equation 4.4.

To put these values into perspective, we can compare the respective *link-cost* evolution for both field sizes of one and two bytes, with maximum route size of $L_{max} = 64$. These *link-cost* values are computed based on the reputation calculation performed in Chapter 3.2.1 and shown in Figure 4.8.

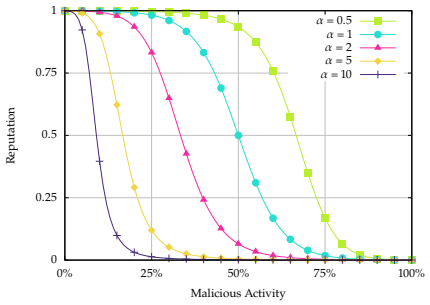
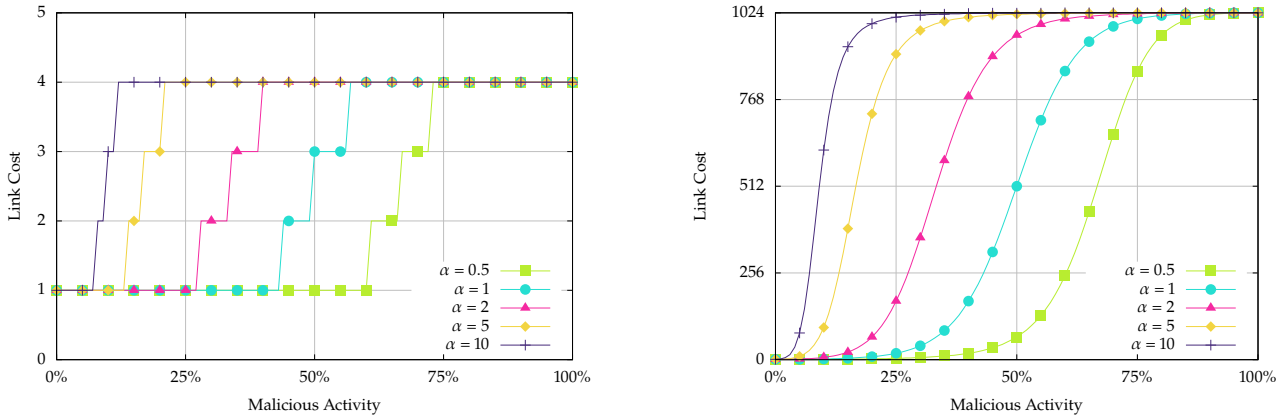


Figure 4.8: Reminder of the evolution of a nodes reputation, previously shown in Figure 3.6 in Chapter 3.2.1

Figure 4.9 shows a side by side comparison of the *link-cost* evolution, based upon the maximum value of C_{max}. We can see that with a lower precision, as shown in Figure 4.9a, the increase in *link-cost* is performed in staged, steadily increasing over time towards the maximum value of 4. However, with a much higher precision, as sown in Figure 4.9b, the evolution heavily resembles the inverse of the reputation, as illustrated in Figure 4.8. This is the case for example of the evolution where $\alpha = 0.5$ (**green** squares), where with $C_{max} = 4$, the *link cost* doesn't increase until the malicious activities reach 60%. On the other hand, with $C_{max} = 1\,024$ this evolution starts much sooner, approximately around the 10% mark, allowing the network to react much sooner.



(a) $C_{max} = 4$ corresponding to a field size of one byte

(b) $C_{max} = 1\,024$ corresponding to a field size of two bytes

Figure 4.9: Study of the evolution of the *link-cost* based on the reputational values from Figure 4.8 with varying levels of precision for a maximum route length of $L_{max} = 64$

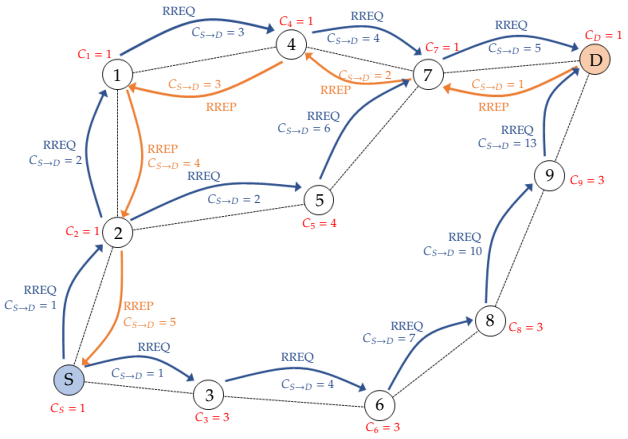


Figure 4.10: Theoretical route discovery utilising reputation-based *link-cost* to determine the most reputable route, illustrated with AODV

Now armed with the ability to determine the *link-cost* of a node, we once again return to the same network shown previously, illustrated once more in Figure 4.10 using AODV. Previously, we showed that the route selection process would determine the route $S \longrightarrow 2 \longrightarrow 5 \longrightarrow 7 \longrightarrow D$ to be the best option, as it is indeed the shortest most direct route. However, if node 5 suddenly turned to the dark size, it could severely impact routing operations. Here we show what would happen if utilising the *link-cost* metric during route discovery. As stated, node 5 has been identified as malicious thanks to our behavioural analysis methodology presented in the previous chapter, resulting in it receiving a low reputation. As shown in Figure 4.9a, the corresponding *link-cost* value for a misbehaving node with a low reputation is the maximum possible value, here 4. As a result, the previous route is artificially lengthened, inciting AODV to select a more reputable route, at the cost of a larger number of hops. Thus, during discovery, the *link-cost* value for the route is effectively larger at a total of 6 than that of the top route, itself at 4. As a consequence, the selected route swaps instead to $S \longrightarrow 2 \longrightarrow 1 \longrightarrow 4 \longrightarrow 7 \longrightarrow D$ instead, an increase of one hop but also in terms of trustworthiness. As for the unused nodes, as stated previously we provide a neutral reputation to all nodes at the start of their life. Here, since they haven't been used, this neutral reputation remains, providing a *link-cost* of 3 for each node, as seen in Figure 4.9a

and easily identifiable with $\alpha = 1$ (**light-blue** circles).

4.3 Protocol Integration

Thanks to the *link-cost* metric, it is possible to influence routing protocols into selecting more reputable paths over the most direct. However, before the cost values can be computed, the nodes reputation must also be determined. As stated in Chapter 3, this reputation is calculated based on the number of *good* and *bad* actions issued from the behavioural observation phase. Furthermore, this distinction is achieved by the Miners gathering the expected sequence of events from passing route discovery or control packets. Unfortunately, not all the necessary information is contained in these packets, making it difficult in some cases to visualise the big picture.

As a result, some adaptations to the routing protocols must be performed in order to fully incorporate the reputational consensus module. In this section, we present the integration process for two *Reactive* protocols, AODV and DSR along with modifications to the current packet structure. Furthermore, we provide a theoretical overview of how this module can be adapted to function in conjunction with *Proactive* routing protocols, in this case RPL.

4.3.1 AODV-Miner

Previously, we used AODV to illustrate the functionalities of the *link-cost* metric. By integrating this approach directly with AODV, along with the reputation-based consensus mechanism, it is possible to influence the route selection process in a similar fashion to [176]. Thanks to its RREP packets which identify the path to be taken, neighbouring Miners can gain an overview of the correct sequence of steps in order to reach the destination. However, these packets are missing some important information which requires some adaptations to the existing packet structure. Furthermore, some characteristics of AODV will need to be adapted to allow efficient evaluation of both behaviour as well as the selection of the most efficient route.

[176]: Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450

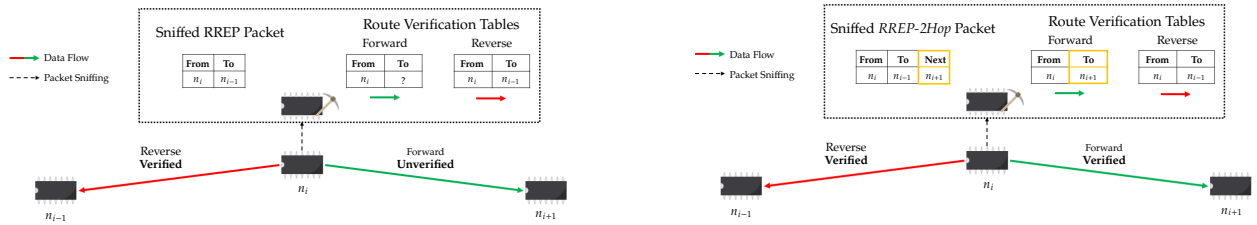
Here, we present the new additions and updates to AODV, which we call *AODV-Miner*.

Methodology

As stated, AODV's RREP packets are the perfect place for the Miner's to recover the sequence of hops. However, the purpose of these packets is to return back along the selected route towards the source, indicating to each node along the way to whom they must transmit their packets. In this case, only part of the route is contained in RREPs, i.e., informing node $n-1$ to transmit towards n.

This problem is illustrated in Figure 4.11a. As we can see, our Miner is in communications range with node n_i, meaning it can overheard all of its transmissions. Since its goal is to validate the behaviour of n_i when

routing information towards the destination, it must be able of assessing the hop towards n_{i+1}. However, as we can see, the contents of the sniffed RREP packet only contain the information to complete the reverse RVT entry back towards the source. As a result, only the reverse hop can be validated for n_i, leaving its forward hop unknown.



(a) Validation with RREP

(b) Validation with RREP-2Hop

Figure 4.11: Illustration of the need for RREP-2Hop for the construction of forward RVTs

To remedy this snag, we propose an update to the RREP packet structure to include the information regarding the next expected hop from the point of the of the packet's transmitter. This new packet format called RREP-2Hop is presented in Figure 4.12. We can see that there are three new elements contained in this packet. Firstly, we can see the addition of the aforementioned *link-cost* field, effectively replacing the previously used *hop-count*. Next up is the addition of the IP address of the next hop, as well as its layer two MAC address. As stated previously, the behavioural observation is based upon the MAC addresses of each participating node, thus granting Miners the ability to add the missing hop to their RVTs. Since AODV allows the creation of dynamic opportunistic routing table entries based upon received RREP packets, by providing the IP address, we allow the creation of 2hop routes, if the node so desires. So as to allow seeming-less integration with AODV, we also include the addition of a new *Miner flag*, allowing nodes to quickly identify if either AODV or *AODV-Miner* is in use and parse the packet accordingly.

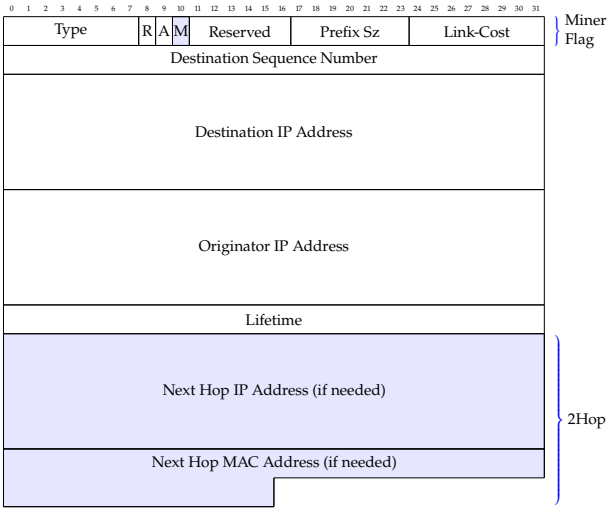


Figure 4.12: Proposition for an updated RREP packet structure for AODV containing the next expected hop information, called RREP-2Hop

Figure 4.11b presents how the proposition of RREP-2Hop provides the missing information to the Miner. We can confirm that, thanks to the addition of the next hop address, the Miner in question is finally capable of constructing the forwards RVT entry, thus allowing it to effectively validate the hop towards the route's destination.

One final update regarding the generic AODV functionality is the analysis of RREQ packets by the route's destination. Indeed, depending on the implementation, the routes destination can either perform an analysis on the received RREQs, only responding to that with the lowest *hop-count*, or it simply responds to all, thus pushing the final decision back to the source. However, due to our implementation and the need for the Miner's to gain an accurate overview of the route, we added some extra conditions. Firstly, upon receiving an RREQ, the destination awaits for a random duration, thus providing time for other requests to arrive. At this point, the destination discards those requests containing high *link-costs*, thus only keeping the most efficient possible. After no further requests have been received for a while, the destination creates its RREP-2Hop reply and unicast's it back towards the source. From this point onwards, any further RREQs received for the same route are automatically discarded by the destination, thus reducing the risk of RVT corruption.

4.3.2 DSR-Miner

Having provided a proof of concept with AODV, we can turn our attention towards integrating our consensus module into other protocols. In this case, DSR comes to mind. Indeed, DSR uses the same basic functionality of RREQs and RREPs, all the while changing how the paths themselves are determined [203]. This means that the basic Miner functionality remains the same, where they rely on the contents of RREP packets to inform them of the expected next hop. The main difference here is both how these packets are formed and how they are used to select a route.

[203]: David B. Johnson and David A. Maltz. 'Dynamic Source Routing in Ad Hoc Wireless Networks'. In: *Mobile Computing*. Ed. by Tomasz Imielinski and Henry F. Korth. Springer US, 1996. DOI: 10.1007/978-0-585-29603-6_5

In this section, we present how this new adaptation of DSR called *DSR-Miner* functions, as well as the various updates needed to the general packet structure and constraints imposed by DSR.

Methodology

Since DSR relies on RREQ and RREP packets to discover routes to a certain destination, the principals resemble those of *AODV-Miner*. However, DSR utilises different methods to reach its goals compared to AODV, the most apparent of which is the use of IPv4 headers to contain routing information. As stated previously, DSR only stores route information at the source node, providing a hop-by-hop route in the IP header, indicating the expected next hop to each individual node. As a result, this approach imposes strict restrictions on how DSR can function, especially when adapted and utilised on 6LoWPAN networks.

As defined in *RFC 4919* [204], 6LoWPANs are networks comprised of resource constrained IoT devices conforming to the IEEE 802.15.4-2003 standard. One of its main characteristics is to provide low cost communications using as little power as possible, thus imposing restrictions and constraints on the data transported. The main limitation in our case is the reduced Maximum Transmission Unit (MTU) size of 127 bytes, allowing for a payload of only 102 bytes. This means that, since IP headers cannot be fragmented, the size of the DSR options is severely reduced as it can only contain a limited supply of addresses before the IP packet is too large. It is also important to node that, in the case of the *Source Routing* header,

[204]: Gabriel Montenegro, Christian Schumacher, and Nandakishore Kushalnagar. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*. RFC 4919. 2007. DOI: 10.17487/RFC4919

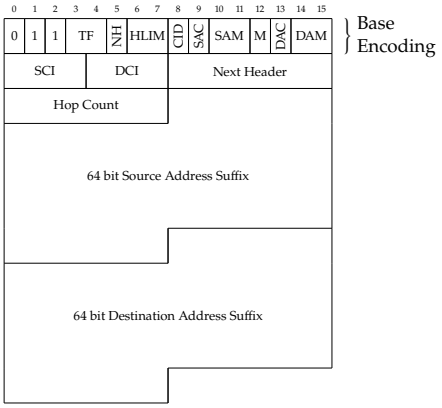


Figure 4.13: Maximum size of a compressed IPv6 header using `LOWPAN_IPHC` for use in 6LoWPANs

[205]: Pascal Thubert and Jonathan Hui. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. RFC 6282. 2011. DOI: 10.17487/RFC6282

Table 4.1: Different size values in bytes for all IP and DSR headers, as well as transport layer and payload size

Type	Size
$MTU_{payload}$	102
$IPHC$	21
DSR_{header}	4
$DSR_{src\ option}$	12
$DSR_{rreq\ option}$	4
$DSR_{rrep\ option}$	3
UDP_{header}	8
$data$	7
$IPv6_{compressed}$	8
MAC	6

the objective is to transfer data to a destination, meaning the 102 byte restriction must also contain the IP payload, including transport layer headers and data, hindering even further the possible route length.

Furthermore, 6LoWPANs function with IPv6 addresses, thus imposing an even bigger handicap on DSR. Indeed, since IPv6 addresses are four times larger than IPv4 addresses, the route length is effectively reduced to a quarter capacity. Thankfully, 6LoWPANs utilises header compression, as defined in [205], allowing the standard IPv6 header to be reduced down to as little as 2 bytes, using `LOWPAN_IPHC` compression. In our simulation environment, Contiki-NG utilises this compression methodology using stateful, context-based reduction for the IPv6 addresses. Based upon this information, we can determine the maximum size the compressed header will be, the format of which is presented in Figure 4.13.

Here, we can see that the first line corresponds to the base encoding, the minimum compressed size possible. However, here there are multiple elements which cannot be compressed. Firstly, we can see the presence of *Source Context Identifier (SCI)* and *Destination Context Identifier (DCI)* fields, used to identify the context aware stateful IPv6 prefix. Secondly, the *Next Header* identifier is clearly stated, since DSR in its current format cannot be compressed using `LOWPAN_NHC` for next header compression. Next, we see that the *Hop Count* field is also clearly marked. Indeed, although `LOWPAN_IPHC` allows for the hop count to be compressed, this is only the case for three values: 1, 64 and 255, thus all other values must be explicitly stated. Finally, both the *Source* and *Destination* IPv6 addresses are also included, here in a compressed form. `LOWPAN_IPHC` compression allows to vary the IPv6 size dependant on the context. In this case, since all nodes are part of the same network and, therefore, contain the same network prefix, the first half of the address can be omitted. This means that a nodes IP address can be represented using only 64 bits, half of the full size, allowing to fit two addresses in the space of one. It is notable that the source or destination address can be omitted during compression, but this is only the case for the first and last hops, as they are derived from the MAC layer addresses for the respective fields. As a result, we can conclude that the maximum size for the compressed header is 21 bytes in its largest format, almost reducing the original format by half.

From here, we can calculate the maximum number of hops which would be included in the DSR *Source Route* option. To aid in this computation, we also update the standard DSR header options, replacing all IPv6 addresses with a compressed version, confirming to the compression used by `LOWPAN_IPHC`, thus allowing to double the number of addresses which can be inserted. We only evaluate this option due to the insertion of the transport layer protocol, here UDP followed by the data payload, making the overall packet larger than the RREQ or RREP options. We can calculate the number with the following equation:

$$max\ hops = \frac{MTU_{payload} - IPHC - DSR_{header} - DSR_{src\ option} - UDP_{header} - data}{IPv6_{compressed}} \tag{4.7}$$

Firstly, we calculate the number of bytes available to store the hop-by-hop addresses in the *Source Route* option header. For this, we subtract the concatenation of all header sizes, including the compressed IPv6 header,

DSR standard and *Source Option* headers along with the UDP header and total payload size, extracted from the Cooja simulation implementation. Finally, the number of hops $max\ hops$ can be determined by simply dividing the available space by the size of the compressed IPv6 addresses. A brief overview of all sizes used is shown in Table 4.1 and the computation is as follows:

$$\begin{aligned} max\ hops &= \frac{102 - 21 - 4 - 4 - 8 - 7}{8} \\ &= \frac{58}{8} \\ &\approx 7 \text{ hops} \end{aligned} \tag{4.8}$$

As a result, we can conclude that at most, seven compressed IPv6 addresses can be contained in the DSR *Source Route* option, thus reducing the overall possible route length. However, as seen previously with AODV, basic RREPs do not contain all the necessary information needed for the Miners to construct their forward RVTs. Indeed, the standard RREP format contains a concatenation of next hop addresses leading back to the destination. Although this is sufficient for routing purposes, it only provides the IP address for the next hop, whereas the Miners require the MAC address to accurately validate routing on a hop-by-hop basis. To combat this issue, we propose an adaptation of the RREP-2Hop for DSR option headers, shown in Figure 4.14.

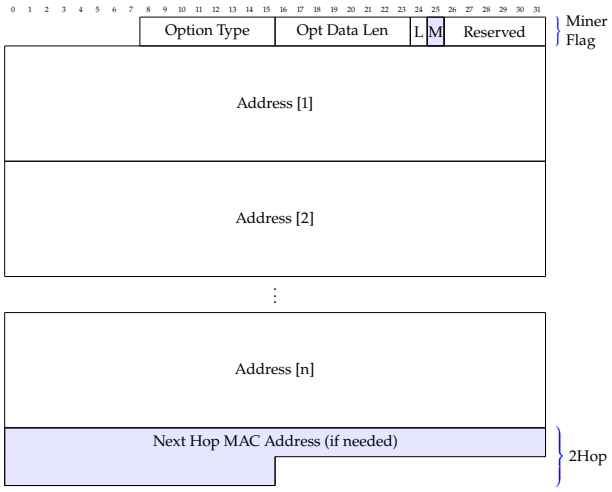


Figure 4.14: Proposition for an updated RREP option header for DSR containing the next expected hop information, called RREP-2Hop

As we can see, similar to previously we have included two new additions to the header structure. The first is the addition of the next hop MAC address, allowing the Miner to construct the corresponding next hop for the transmitter node in their forward RVT. Compared to AODV, since the header already contains a list of IP addresses used to construct the source route, the next hop IP address is not needed. To allow to identify which version of RREP is being parsed, we propose the addition once more of a Miner flag, allowing to determine if the last value in the header is an IP a MAC address. As a result, this addition reduces the maximum number of possible hops the RREP header can take. However, as we can see in Equation 4.9, the number of hops with the addition of the MAC address is equal to that that from the DSR *Source Route* option if the destination address is omitted, meaning that the functionality is not affected.

$$
\begin{aligned}
max\ hops &= \frac{MTU_{payload} - IPHC - DSR_{header} - DSR_{rrep\ option} - MAC}{IPv6_{compressed}} \\
&= \frac{102 - 21 - 4 - 3 - 6}{8} \\
&= \frac{58}{8} \\
&\approx 8 \text{ hops including destination address}
\end{aligned}
\tag{4.9}
$$

Since DSR doesn't directly integrate shortest path metrics into its headers, there is no direct integration for the *link-cost* value during routing. As explained previously, DSR functions on a "first come - first serve" basis, meaning subsequent RREQs are dropped by the receiving node. Our implementation of *DSR-Miner* changes this approach to be more in line with AODV. Upon receiving a packet with an RREQ header option, the node calculates the associated *link-cost* for all addresses contained in the hop list. This value is calculated and compared for all subsequent RREQs, allowing the route with the lowest *link-cost* to be propagated towards the destination. The same operation is performed by the destination node, where instead of responding to all received requests, the destination waits for a certain duration before responding to the route with the lowest overall cost. After this point, all subsequent requests are dropped by the destination, so as not to corrupt the Miner's RVTs.

Furthermore, since this *link-cost* value isn't stored directly in the headers themselves, there is no immediate restriction for storage space. As a result, we decided to store the value using a two-byte variable, significantly increasing the overall precision, as previously illustrated. For this implementation, we set the maximum path length $L_{max} = 255$, as defined in *RFC 4728* [197], with the maximum field size of 16, $2^{16} - 1 = 65535$. This means our value of C_{max} increases, allowing more precision than with AODV. We calculate C_{max} for DSR as follows:

[197]: Yih-Chun Hu, Dave A. Maltz, and David B. Johnson. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728. 2007. DOI: 10.17487/RFC4728

$$C_{max} = \lceil \frac{2^{f_{size}} - 1}{L_{max}} - 1 + C_{min} \rceil \tag{4.10}$$

$$
\begin{aligned}
C_{max} &= \lceil \frac{65535}{255} - 1 + 1 \rceil \\
&= 257
\end{aligned}
\tag{4.11}
$$

With a maximum *link-cost* value $C_{max} = 257$, we have a lot more precision over a node's reputation than with *AODV-Miner*. Figure 4.15 shows the representation of this link cost, with the version used by *AODV-Miner* shown in Figure 4.16.

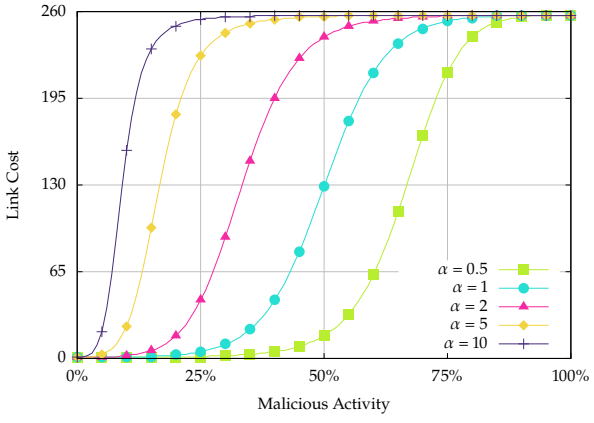


Figure 4.15: Study of the evolution of the *link-cost* based on the reputational values from Figure 4.8 as defined with the parameters for *DSR-Miner* with $C_{max} = 257$ and $L_{max} = 255$

As a result, we confirm that by using a larger storage field for the *link-cost* value, we can increase the overall precision. However, by also increasing the maximum number of hops to $L_{max} = 255$, we decrease the precision compared to our previous analysis shown in Figure 4.9b. It is to be noted that as our previous analysis shows, only seven hops at most are possible before the overall packet is dropped due to buffer overflow. However, in the interest of remaining as true to the specifications as possible, we kept the maximum path length value as defined in the original RFC.

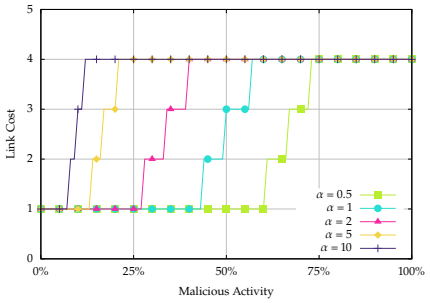


Figure 4.16: Reminder of the representation of a nodes reputation using the *link-cost* metric with $C_{max} = 4$ as used by *AODV-Miner*, previously shown in Figure 4.9a in Section 4.2.2

4.3.3 RPL-Miner

We have discussed how our approach can be integrated into *Reactive* routing protocols, in particular AODV and DSR Since these protocols select their routes only when needed, it is, therefore, easy to influence the routing process each and every-time a path is needed. As a result, it would be interesting to evaluate this approach coupled with other types of protocols, such as *Proactive* ones.

In this section, we propose a theoretical extension of the *Proactive* protocol RPL, incorporating our consensus-methodology to help determine the best parent to whom transit their packet. We name this theoretical protocol: *RPL-Miner*.

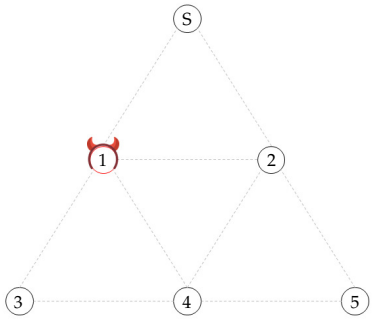
Methodology

As presented previously in Section 4.1.1, the RPL routing protocol functions differently to that of AODV and DSR. Instead of discovering routes when needed, RPL maintains the network by using DODAGs, a tree-like topology spanning down from a root node. With such an approach, nodes can simply determine if their destination is part of their own sub-DODAG in which case the data can be relayed to the next child, or if not it is relayed to the nodes DODAG parent, and the cycle continues. If, however, the RPL instance is functioning in *non-storing* mode, then the data will always be transmitted to the DODAG parent, all the way to the DAG sink node, which then determines the route to take to the destination node, even if they are a child of the source.

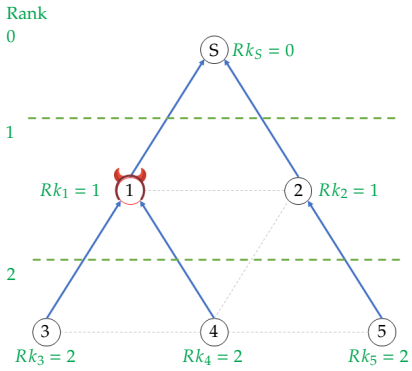
In any case, routing takes place dependant on the branches of the different DODAG trees in either direction. Thus, to influence the routing itself, it would be necessary to influence how the DODAG trees are both created, and how they function. Indeed, two notions must be explored: reputation integration, influencing the parent selection process and thus packet routing; and behavioural observation, with the dynamic selection of Miners.

Parent Selection

DODAGs are generated around the concept of node *rank*, where each node is provided with a fixed value based upon a specific criteria. This value is generally represented by the nodes position in the network relative to the DAG's sink, thus removing the possibility of loops appearing. By utilising this value, nodes can determine both their own parents and children, using the general concept of lower values representing parents,



(a) RPL network before DAG construction



(b) RPL network after basic DAG construction based on each node's rank Rk_n

Figure 4.17: Basic construction of a DAG in RPL in a network of 5 nodes, with node 1 being malicious. Each **blue** arrow corresponds to to the child-parent relationship in the DAG

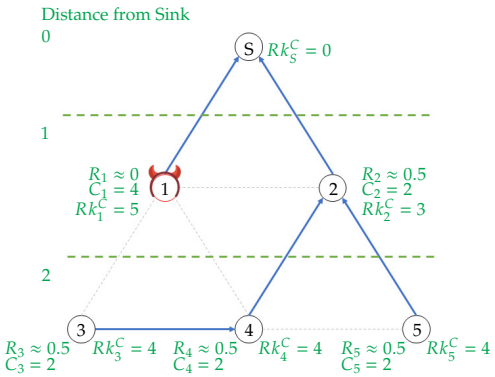


Figure 4.18: Integration of the *link-cost* (C_n) based on each node's reputation (R_n) into the RPL DAG construction process, influencing each nodes *link-cost rank* Rk^C_n

Comment 4.3.1

We suppose that we employ $C_{max} = 4$ as previously demonstrated in Section 4.2.2, spreading the reputation across four *link-cost* values, from 1 (high reputation) to 4 (low reputation).

which are deemed closer to the sink; and higher values representing children, which are considered as closer to the tree's leaves. Since DAGs represent logical topologies, it is possible to influence the parent selection process, inciting nodes to select another as parent when one is available, lengthening the overall distance to the sink, but potentially avoiding malicious nodes.

The basic topology selection process is demonstrated in Figure 4.17 with Figure 4.17a representing the initial network status. Here we can see that the further away from the sink node, the higher the corresponding rank, representing the distance in hops. As a result, nodes will select the neighbour with the lowest rank to become their parent, indicating them of such. This means that each node is aware of not only their parent, but are also informed of their direct children due to the same process, thus allowing the DAG to be formed as shown in Figure 4.17b. However, in this scenario node 1 is indeed malicious, meaning that nodes 3 and 4 will be transmitting data directly into the hands of an attacker.

This is where the *link-cost* can come into play once more. Previously, this value was used to influence the route discovery process, providing the ability to avoid malicious nodes. With RPL, it is possible to influence the choice of parent by influencing the value of each node's rank, artificially increasing it the lower the reputation. So as to avoid routing loops, the *link-cost* value can simply be added to the nodes position relative to the sink, thus guiding nodes to select a parent which is closer to the sink if no other possibility is available.

We define the updated *link-cost rank* of node n, Rk^C_n as follows:

$$Rk^C_n = C_n + Pos_n \tag{4.12}$$

where C_n corresponds to the *link-cost* of node n and Pos_n its position relative to the sink in number of hops. From here, the generic algorithm aiming to select a node with a lower rank as parent can still take over, forming a connected DAG with malicious nodes pushed as far towards leaf positions as possible. The impact of this can be seen in Figure 4.18 based upon the same topology as previously.

Here we can see that, although the position of each node is the same, the computed *link-cost rank* has increased. Firstly, we can see that thanks to the reputational value, the malicious node, 1, has been identified and flagged as malicious, receiving a low reputation of approximately 0, thus increasing its *link-cost* to 4. By computing the *link-cost rank*, we can see that its value is now higher than that of its previous children nodes, 3 and 4. From here, by simply selecting the neighbour with the lowest *link-cost rank*, a new DAG can be formed, possessing a clear path circumnavigating around the malicious entity.

Miner Determination

As we have demonstrated, we can influence the DAG construction process, allowing malicious nodes to be avoided as parents and allowing nodes the possibility to have a clear route to the sink. However, since this relies on the nodes reputation, the Miner role distribution and behavioural analysis must also be evaluated. Furthermore, contrary to previously, RPL includes some new challenges, notably the presence of two modes:

storing and *non-storing*. As a result, the overall functionality of the network with regards to data routing changes, meaning the Miner's must also adapt and act differently dependant on the storage mode utilised.

Previously, node role selection was a simple task dependant on the established route by basing the selection on passing RREPs. With RPL, routes are known before hand, making the simple distinction between miner and router more complicated. Indeed, the role distribution process used in AODV and DSR, allowed nodes to take on either role in a fixed capacity for each route, providing the network with stability and clear distinction of who the Miner's were. In this case, the lines become slightly blurry, as Miner's can only be determined based upon overheard data traffic. To solve this, we update and repurpose the specifications of the Miner role, adapting it to a situation where every and all route is known in advance.

The first major modification is how the Miner's are determined. In Chapter 3.3, we state that each Miner is associated with a specific route, and remains so for that route's lifetime. When it expires, the resulting consensus validation confirms the observations, which are inserted into the blockchain updating the node's reputations. However, in RPL since there is no explicit route discovery, Miners cannot be associated with any route. To solve this, we propose an update to the role selection algorithm, presented in Algorithm 3.

Algorithm 3 Miner selection algorithm used in *RPL-Miner*, mixes role selection and behavioural validation for *Proactive* protocols.

$llsrc \leftarrow$ MAC address of the transmitter
$lldst \leftarrow$ MAC address of the receiver
$dst \leftarrow$ IP address of the destination
$hash \leftarrow$ sha-256 hash of $packet$
$type \leftarrow$ the content type of the $packet$ (RPL, data, etc.)

if $type \neq$ data **then** ▷ Control Packets
    Analyse $packet$ as normal and exit
end if
if $lldst =$ me **then** ▷ node is part of route
    set $role_{dst}$ as $Router$
    Drop all $good_{dst}$ and bad_{dst} actions for $hash$
    Process $packet$ as normal and exit
else if $role_{dst} = Router$ **then** ▷ $packet$ already routed
    Drop $packet$ and exit
end if
▷ node is considered $Miner$
$RVT \leftarrow$ get validation table for $llsrc$
$nextHop_{pkt} \leftarrow$ next hop from RVT towards dst ▷ $llsrc$ parent or child
if $lldst \neq nextHop_{pkt}$ **then** ▷ invalid next hop - Malicious behaviour
    Increment $bad^{hash}_{llsrc_{dst}}$
else ▷ Valid behaviour
    Increment $good^{hash}_{llsrc_{dst}}$
end if

By allowing all neighbouring nodes to dynamically take on the role of Miner when a data packet is overheard, we can assure continual observation with predefined and determined routes. Here, we no longer

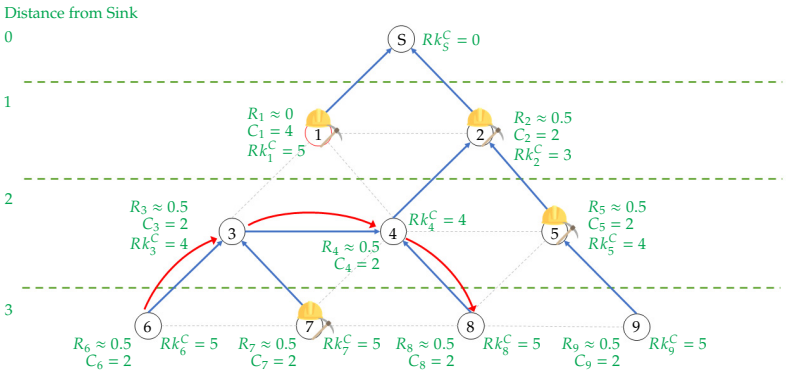


Figure 4.19: Illustration of the Miner distribution for the route 6 → 8 in RPL *storing* mode, where the **red** arrow shows the packets path from source to destination

associate a Miner with a specific route but instead utilise a dual association, with a specific destination and a specific packet. Previously, we defined a constraint where nodes cannot mine the route in which they are participating, to reduce the impact of potential conflict of interest. In this case, since routes are always known, we reduce this constraint, allowing Miners to dynamically mine the route taken by a packet, but only if they weren't involved in the routing itself. This is represented in the algorithm where the node received a packet for routing, and drops the list of *good* and *bad* actions associated with said packet, without impacting the previous observations for this route.

Another specificity of RPL is its two operational modes: *storing* and *non-storing*; allowing intermediate nodes to route data, or the sink node to perform source routing instead. In this first case, a single route can be determined from $src \rightarrow dst$, similar to those used in the previous protocols. However, in *non-storing* mode, data is passed up the DAG to the sink, before coming back down with the help of a source routing extension similar to DSR, providing a hop-by-hop map towards the destination. By only utilising the destination address, we can differentiate between a direct route, and the knowledge of the different intermediate nodes. Indeed, during routing only the destination address is of use with the source only needed for potential replies.

Since all routes are already known, by simply looking up the destination, either in the DODAG of a specific node or simply by interrogating the DAG sink, a path can be found. Furthermore, previously the Miner's began the consensus validation phase following the route's expiration, however, here there is no direct equivalent. To solve this, we can set the Miner's timeout function to a minimum recurring value, which can be customised to shorted the reactivity delay as well as decrease the nodes history size, or increase it potentially causing more disruption in the mean time, but with more detailed observations in the node's history.

Another important modification is the construction of the Miner's RVTs. Indeed, with the lack of RREPs, the RVT entries must be both constructed dynamically from the DAG construction and constantly kept up-to-date by analysing DAO control packets. However, once more the contents of these RVTs also varies dependant on the operating mode. Indeed, the Miners must be aware of which mode the RPL instance is running, allowing to differentiate between "normal" routing in *storing* mode and sink source routing in *non-storing* mode.

Firstly, we will present how the RVTs and Miners would function in *storing* mode, which would also be applicable to *non-storing* as well. However, we will also discuss a compressed, reduced version for use in such a network, applying the same concept as *non-storing*, to use as little storage space as possible. Figure 4.19 presents how a packet is routed in *storing* mode, along with the routing tables of each node. We can see that if node 6 wishes to contact node 8, node 4 knows that node 8 is its direct child, simply routing the packet directly to the destination. Indeed, we can see that each node's routing table contains a detailed representation of their own DODAGs, allowing them to determine if the destination is known to them, or not in which case the packet is passed to their parent.

These table entries are provided using DAO control packets, providing routing information from parent to parent, all the way to the sink. By allowing nodes to overhear passing DAOs, it is possible to extend the generic RVT to represent the DODAGs of all neighbouring nodes. Table 4.2 shows gives an overview of these DODAG-RVT tables for nodes 7 and 1 from Figure 4.19. We can see that node 7 in Table 4.2b is a neighbour of nodes $3, 4, 6, 8$, all nodes involved in this route. As a result, their DODAG-RVT must contain the relationship information for all these nodes, including their parents and contents of their own DODAG. Here we can see that for each neighbour, we list their direct children as well as sub-children, allowing the node 7 to validate the next hop for all sub-children, without the need to know the exact topology thereafter. The same can be said for node 1 in Table 4.2b, only this time the contents of its table concern the full DAG as the sink itself is its neighbour, however, it is not.

On the other hand, node 1, which is the neighbour of nodes $s, 2, 3, 4$ has a higher position than node 7, indecently being one of the sinks direct children. This means that its DODAG-RVT would be much more extensive than node 7 since it would contain the entire DAG due to the sink being a direct neighbour. Armed with this information, each Miner can determine if the actions of neighbouring nodes is indeed valid, based upon their own image of the surrounding DODAGs.

As stated previously, this approach would also function in *non-storing* mode, since all information is passed up towards the sink, allowing nodes to create their DODAG-RVTs. Indeed, Miners can simply verify that when a packet is transmitted towards the sink for routing, the packet is passed to the correct parents on the way. When it comes to the reply, the Miners can perform two verifications:

- Check of the performed hop corresponds to the provided hop from the *source routing* header
- Check of the hop in question is expected, extracted from the DODAG-RVT

However, the concept of *non-storing* is to remove the storage need from intermediate nodes, moving the routing table up to the sink. Figure 4.20 shows the same network as previously, only with the sink node possessing the routing table instead. We can see that, contrary to previously, node 4 doesn't transmit directly to node 8, but instead passes the data to its parent, all the way up to the sink. From here, the sink retransmits the

Table 4.2: DODAG-RVT entries for nodes 1 and 7

(a) Node 1

Parent	Node	Children	Sub-Children
4	3	6	
		7	
2	4	3	6, 7
		8	
S	2	4	3, 6, 7, 8
		5	9
	S	2	3, 4, 5, 6, 7, 8, 9
		1	

(b) Node 7

Parent	Node	Children	Sub-Children
4	3	6	
		7	
2	4	3	6, 7
		8	
3	6		
4	8		

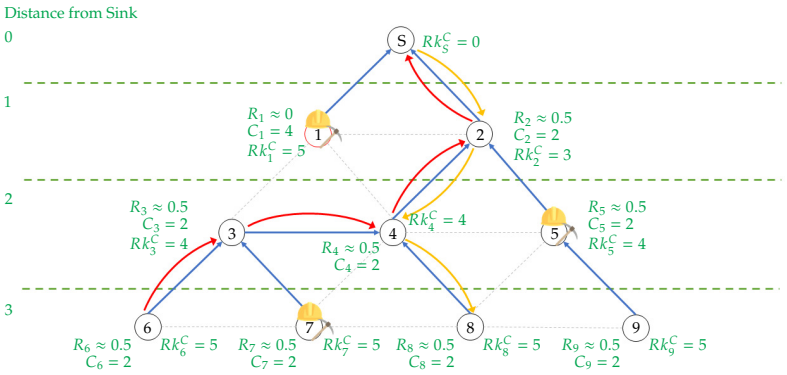


Figure 4.20: Illustration of the Miner distribution for the route 6 → 8 in RPL *non-storing* mode, where the **red** arrow shows the packets path from the source to the sink, ad the **orange** arrow the *source route* from the sink to the destination

data to the intended destination, with an included *source routing* header indicating the hops needed to reach the destination node.

To allow a *non-storing* version of *RPL-Miner*, we can simply reduce the information contained in the RVTs to contain only the list of direct parents for each neighbour node. With this, for each transmission, the Miners would be able to verify if the hop used towards the sink corresponds to the correct hop from the parent-RVT when no *source routing* header is present. However, the return journey would have reduced precision, where instead of verifying the expected hop from the DODAG-RVT, Miners would only be able to check if the receiver is a direct child of the transmitter, indicating next-hop *source route* deviation.

Table 4.3: Parent-RVT entries for nodes 1 and 7

(a) Node 1

Parent	Node
S ←	2
2 ←	4
2 ←	5
4 ←	3
4 ←	8
3 ←	6
3 ←	7

(b) Node 7

Parent	Node
4 ←	3
2 ←	4
3 ←	6
4 ←	8

Table 4.3 presents these updated tables for the same Miners as previously, 7 and 1. We can see that the overall table size is much simpler than previously in Table 4.2 due to the lack of precision regarding the various sub-children. As a result, we can see that both can validate the route towards the sink, as the hops are parent-to-parent. Furthermore, by reading the table in the other direction, it is possible to verify that each node relays the data towards a node which is one of their children. It is important to note that since all DAO packets are routed to the sink to construct its routing table, nodes such as 1 can also see the expected parent of nodes 5, 6, 7, 8 and 9, meaning that the parent-RVT itself needs to be maintained to reduce overflow. This means keeping only entries which concern the neighbours of the node, thus excluding node 9 as itself or its parent are too far out of range.

General Considerations

Compared to *Reactive* protocols, *Proactive* versions provide the ability for route to be constructed dynamically throughout the lifetime of the network. Indeed, since routes are constantly updated with fresh values, it is possible to influence this process towards more reputable nodes using the same *link-cost* concept as previously. In the case of *RPL-Miner*, by associating te *link-cost* value with the nodes distance from the sink, we can influence the DAG construction through node rank manipulation. In doing so, we can provide the same capabilities as previously, allowing malicious nodes to be avoided.

However, the reputation of each node must be calculated to influence the rank correctly. As a result, the Miners and their RVTs must also evolve. Contrary to *AODV-Miner* and *DSR-Miner*, the RVT no longer contains the exact hop sequence, but instead an overview of the relationships of each neighbour node. Furthermore, RPL possesses two modes, allowing routing information to be stored or not on each intermediate node. This process can be mirrored in *RPL-Miner* on the precision of the relationships in the RVT, where *storing* mode would allow for a complete detailed overview of each nodes sub-DODAG in their so called DODAG-RVT. On the other hand, *non-storing* mode would reduce this information to basic interactions between parent and child, reducing the contents of their parent-RVTs, thus the overall storage space required.

This theoretical proposition illustrates how our reputational consensus module could be adapted to *Proactive* protocols, such as RPL to form *RPL-Miner*. However, to verify this approach, it would need to be simulated against the same scenarios as *AODV-Miner* and *DSR-Miner*.

4.4 Efficiency Evaluation

In the previous section, we presented how our consensus-based reputation module can be integrated into two *Reactive* protocols, as well as a theoretical integration with a *Proactive* protocol, all using our *link-cost* metric. To evaluate how our module impacts the network integrity with both *Reactive* protocols, we performed multiple in depth simulations, pitching the networks against various levels of malicious entities and threats. In this section, we evaluate the efficiency of both *AODV-Miner* and *DSR-Miner* in two distinct simulation scenarios.

4.4.1 Simulation Environment

As stated previously, we utilised the Contiki-NG operating system as a basis for our implementations, allowing them to be simulated with their Cooja simulator. Due to Contiki-NG's specifications, such as the use of an IPv6 net stack using a 6LoWPAN network layer along with an 802.15.4 radio layer, our simulated environment resembles our hypothesis from Chapter 3. In this case, we implemented both AODV and DSR following their specifications from their respective RFCs [196, 197]. In order to gain an overview of their functionalities, we devised two distinct network based scenarios:

- **Medium size network**: In this scenario, we analyse the behaviour of the consensus module in a network of 30-nodes, spaced out in an area of $150m \times 150m$.
- **Large size network**: In this scenario, we push our module further, testing its reactivity and resilience in a larger and denser network of 100-nodes, contained in an area of $300m \times 300m$.

Table 4.4 presents an overview of the different simulation parameters used. In total, we simulated numerous series of 100 simulations during 15 minutes each on 100 different topologies for each network scenario,

[196]: Charles E. Perkins and Elizabeth M. Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing for IP version 6*. Internet-Draft draft-perkins-aodv6-01. Work in Progress. Internet Engineering Task Force, 2001. URL: https://datatracker.ietf.org/doc/draft-perkins-aodv6/01/ (visited on Oct. 17, 2022)

[197]: Yih-Chun Hu, Dave A. Maltz, and David B. Johnson. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728. 2007. DOI: 10.17487/RFC4728

Black-hole: routing based attack, where a malicious device drops all passing messages leavin no survivors.
Grey-hole: similar to Black-holes, only dropping messages dependant on a specifc criteria (i.e., probability, type, length, source/destination, etc.).

Table 4.4: Collection of parameters used during the simulations of the reputational consensus module using Cooja

Parameter	Setting
Area	$\{150m^2, 300m^2\}$
Number of nodes (N)	{30, 100}
Malicious Activity (P_{Ma})	{0% → 100%}
Malicious Weight (α)	{0.5, 1, 2, 5, 10}
Link-cost field size (f_{size})	{8, 16}
Max Length (L_{max})	{64, 255}
Distribution	Random uniform
Transmission Range	50m
Window Size (W_n^R)	5
Reputation Decay (λ_n^R)	Linear
Reputation Half-Life ($t^R_{\frac{1}{2}n}$)	15 min.
Initial Reputation	0.5
Number of Simulations	100
Simulation Duration	15 min.
Messages per Transmission	5
Transmission Interval	1 min.
Message Interval	2 sec.

generated using random uniform distribution in their defined areas. Furthermore, we assure that these topologies are 100% connected, allowing each and every node to reach all others in the network. During each simulation, we confront the routing operations directly against two types of attacks: Black-holes and varying probability Grey-holes. In doing so, we increase the width of our routing-based threat landscape, all the while allowing our system to be tested against basic drop-or-forward attacks. In a similar fashion to the reputation evaluation, each scenario utilises a transmission rate of five packets sent at two second intervals every minute, starting 30 seconds after the start of each simulation. In doing so, we allow enough actions to take place to create an accurate behavioural picture of each and every node, as well as provide the necessary time for the Miners to perform their validation, and distribute the resulting blocks in the network. Finally, some parameters are scenario or protocol specific, such as the number of nodes or the *link-cost* field size. For these varying parameters, we will define them at the start of each scenario.

During our analysis, we utilise various evaluation metrics to determine the efficiency of our approach. These metrics are defined below:

1. **Packets Dropped**: $|Packets_{received}| - |Packets_{sent}|$
   This metric allows to evaluate the temporal impact of malicious nodes throughout the simulations, where a lower value represents a higher level of data integrity.
2. **Throughput**: $\frac{|Packets_{received}|}{|Packets_{sent}|}$
   This metric provides an overview of the efficiency of the routing as a whole, where the percentage value represents the overall portion of packets received by the destination.
3. **Route Length**: nb_{hops}
   This metric grants the ability to identify the average number of hops taken during routing, this representing the impact and potential consequences of influencing the routing itself.
4. **Packet Overhead**: $nb_{transmissions}$
   By representing the total number of transmissions, we can identify how much of an impact the consensus mechanism has on network operations, in particular network traffic.

4.4.2 Simulation - Scenario I

We begin our evaluation within various networks containing 30-nodes. Firstly, we analyse the behaviour of *AODV-Miner* within these scenarios, before performing the same analysis with *DSR-Miner*.

AODV-Miner

We start our simulations pitching AODV against *AODV-Miner*. As we stated previously, the size of the *link-cost* field is known at one byte. Furthermore, we decided to set the maximum possible path length $L_{max} = 64$ to correspond with widely used general TTL value as explained previously. As a result, the *link-cost* metric is that presented in Figure 4.9a, with $C_{max} = 4$. The overall parameters utilised by *AODV-Miner* are shown in Table 4.5.

Table 4.5: Simulation parameters for the first scenario with *AODV-Miner*

Parameter	Setting
Area	150m×150m
N	30
P_{Ma}	{0% → 100%}
α	{0.5, 1, 2, 5, 10}
f_{size}	8 bits
L_{max}	64
Distribution	Random uniform
Tr Range	50m
W_n^R	5
λ_n^R	Linear
$t_{\frac{1}{2}n}^R$	15 min.
Initial R_n	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

We commence our evaluation by analysing the impact of our approach towards general routing efficiency, thus allowing us to determine if *AODV-Miner* is capable of reaching its goal of identifying and avoiding as many malicious nodes as possible. For this evaluation, we compare AODV and *AODV-Miner* when they are submitted to Black-hole routing attacks. The first metric we can evaluate is the number of packets dropped, shown in Figure 4.21.

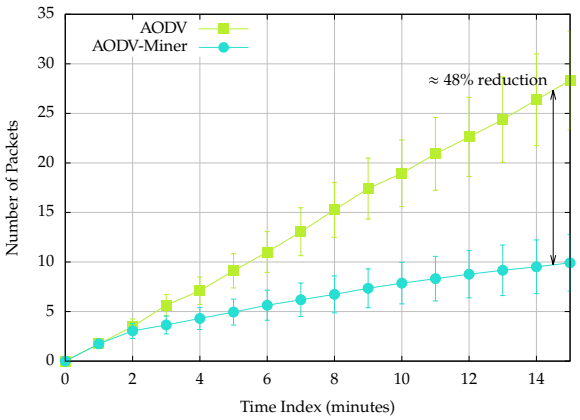


Figure 4.21: Number of packets dropped by both *AODV-Miner* and AODV in a network of 30 nodes with 10% of them expressing malicious tendencies

As we can see, after 15 minutes *AODV-Miner* (**light-blue** circles) has reached approximately 10 packets dropped, whereas AODV (**green** squares) on the other hand is just short of 30. During these simulations, *AODV-Miner* saw a reduction in drops by approximately 48%, almost half that of AODV. Furthermore, by analysing the tendency of the **light-blue** circles, we can see that *AODV-Miner* is slowly stabilising, while AODV continues to rise at a steady pace.

We can confirm these results by analysing the overall throughput between both protocols, as shown in Figure 4.22. We can immediately see that *AODV-Miner* (**light-blue** hashes) has a significant increase in overall throughout when compared to AODV (**green** crosses), even when half of the network is actively trying to disrupt routing operations. Here it is already clear that our reputational approach has provided an increase in

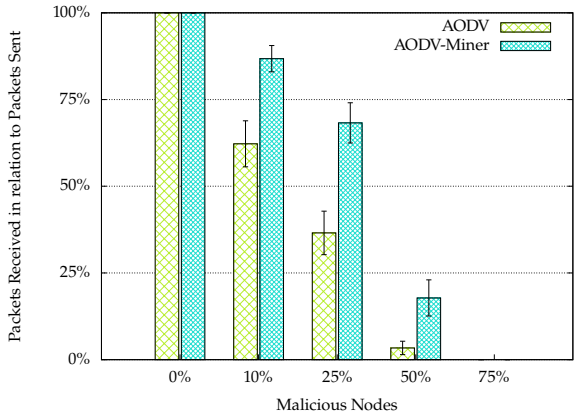


Figure 4.22: Throughput of *AODV-Miner* and AODV in a network of 30-nodes with varying percentage of malicious presence

efficiency to *AODV-Miner*. However, there are some trade-offs for this increase in efficiency.

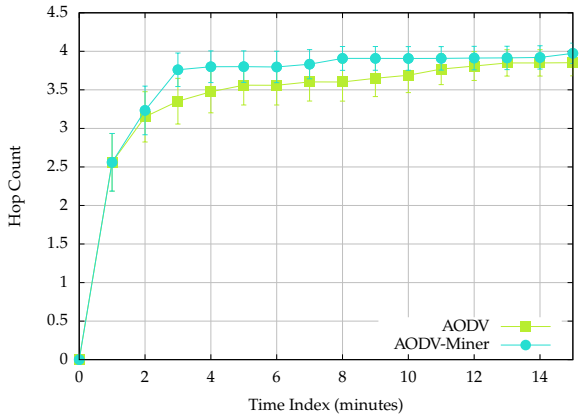


Figure 4.23: Average route length of *AODV-Miner* and AODV in a network of 30 nodes with 10% malicious

The first trade-off is that of overall route length, presented in Figure 4.23. Here we can see that overall, the number of hops per route is higher in the case of *AODV-Miner* than with AODV. This confirms our initial hypothesis that by replacing the shortest route algorithm with the most reputable, the overall length increases on average by 5.2%, thus remaining very affordable.

This isn't the only increase observed with *AODV-Miner*. Indeed, Figure 4.24 presents the normalise overhead between *AODV-Miner* and AODV. It comes at no surprise that overall, due to the need to exchange packets for the validation process to occur, their is a significant increase in packet transmissions. Although this effectively causes some impact on resource constrained devices, our previous results that this trade-off does provide the network with a much needed increase in data protection.

Thanks to these results, we can confirm that our method allows us to identify and avoid malicious nodes, increasing the probability of data reaching its destination. Figure 4.25 illustrates this process presenting one of the 100 network topologies used in this scenario where 25% of nodes perform Black-hole attacks on passing data, 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 effectively visualise the increase in performance. We can see that AODV naturally attempts to take the shortest most direct route possible

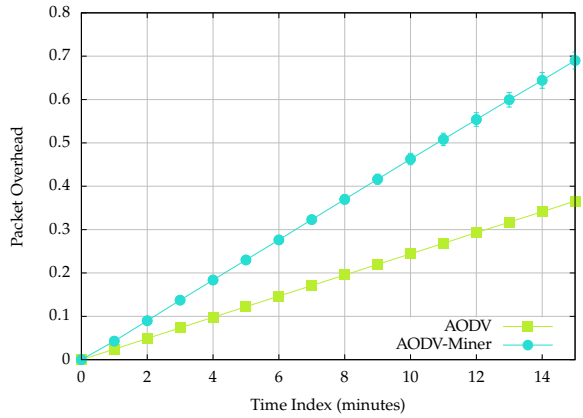


Figure 4.24: Normalised overhead of *AODV-Miner* and AODV in a network of 30 nodes with 10% malicious

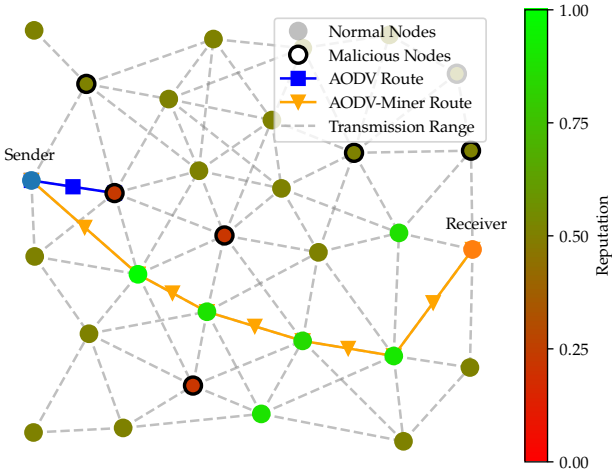


Figure 4.25: Visualisation of route reputation after 15 mins. with *AODV-Miner* and AODV in a network of 30 nodes, 25% of which are malicious

per its programming, which results in immediately encountering a malicious node. In contrast, *AODV-Miner* is capable of discovering a free trustworthy route between the source and destination, avoiding all malicious entities in its path. As we can see by the colour gradient, nodes have been attributed both high (**green**) and low (**red**) reputations, dependant on their activities during routing. By analysing the number of **green** nodes, we can see that a total of eight have been attributed reputations higher than the neutral 0.5. On the other hand, we can see that the three nodes represented in **red** having received low reputations are part of the malicious entities plaguing the network. Since our system is based around behavioural observation, it is a necessary evil to allow messages to be lost in order for the malicious activities to be detected and flagged as untrustworthy through their reputation. This means that in this scenario, three separate routes ended with all their data being lost before *AODV-Miner* was able to adapt and find a valid route.

However, it is important to note that not all good nodes represented in **green** are part of the most used route by *AODV-Miner*. Indeed, three other nodes have received high reputations, with another receiving one slightly higher than neutral. We can, therefore, conclude that *AODV-Miner* found and utilised multiple routes to reach the destination, one of which was early on the simulations, thus resulting in a decayed reputation at the end, all the while remaining higher than its unused peers.

Threat Adaptation

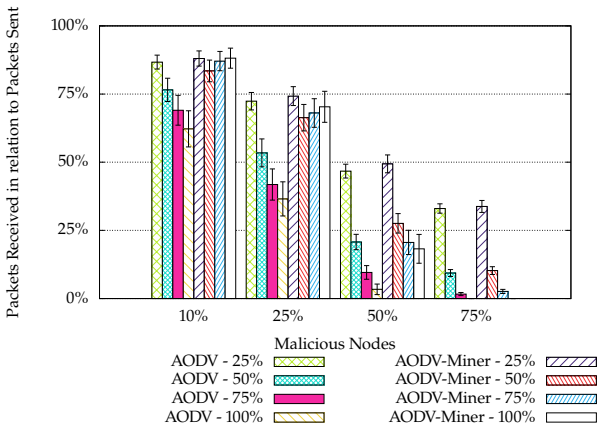
To simulate how our consensus-module can keep with with varying types of threats, we modify the malicious probability of each malicious node, allowing us to simulate Grey-holes. By varying the drop probability, we can increase or decrease the impact of the routing attacks, allowing us to visualise how *AODV-Miner* adapts. Furthermore, we can also introduce our threat weight α into play. Indeed, by performing each simulation run with a different value for α, we can directly visualise its impact on not only the identification of malicious nodes, but also the selection and exploitation of the best possible route when compared to AODV. This analysis is performed in Figure 4.26, with the same varying values of α as shown in the previous chapter: $\alpha = \{0.5, 1, 2, 5, 10\}$.

Naturally, we can see that 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 4.26a shows the results where $\alpha = 0.5$ corresponding to a very forgiving network where *bad* 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.

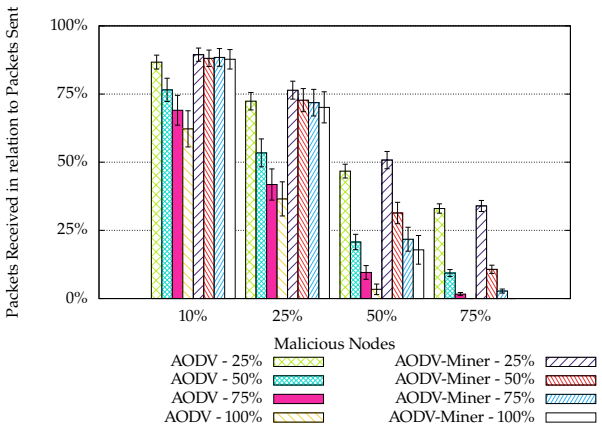
In contrast, Figure 4.26b represents the case where *good* and *bad* activities posses the same weight, $\alpha = 1$. Here we can see that, for 10% malicious nodes, the throughput decreases only slightly the higher the malicious probability, simply due to the need for packets to be dropped before the 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 4.26a. However, we can already identify a slight decrease in throughput when all packets are being dropped when compared to the previous figure.

Figure 4.26c shows the first analysis where malicious activities possess a higher weight to *good*, with $\alpha = 2$, the same value as used in the previous efficiency analysis. 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 previously observed, 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 the throughput where 100% of packets are being dropped is lower than for the previous values of α. 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 α, the more weight is accorded to *bad* actions and the faster *AODV-Miner* can react.

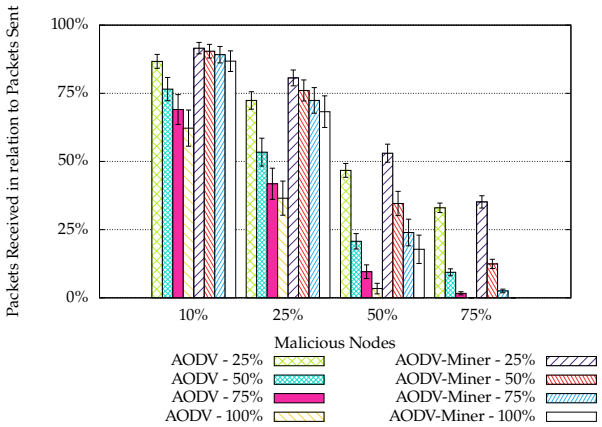
That being said, there is a point where we reach peak efficiency, and the throughput cannot increase any higher and even starts to decrease slightly. This is the case of Figure 4.26d and Figure 4.26e with $\alpha = 5$ and $\alpha = 10$ respectively. We can see that the values remain extremely similar,



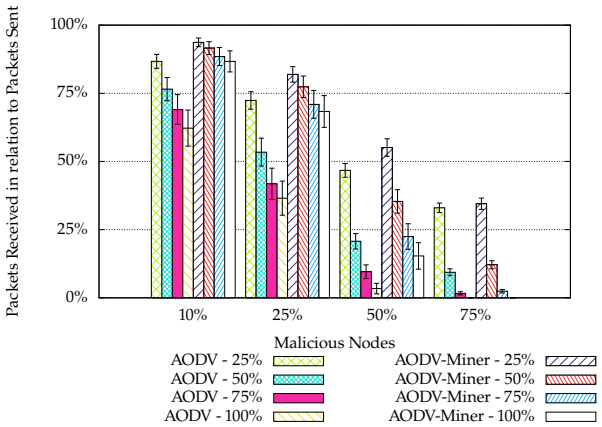
(a) Throughput with $\alpha = 0.5$



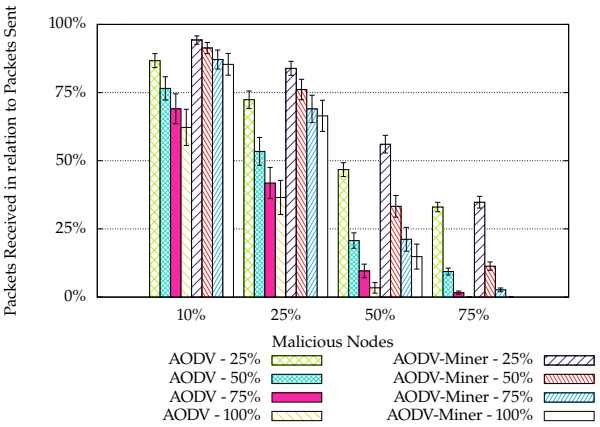
(b) Throughput with $\alpha = 1$



(c) Throughput with $\alpha = 2$



(d) Throughput with $\alpha = 5$



(e) Throughput with $\alpha = 10$

Figure 4.26: Throughput comparison between *AODV-Miner* and AODV in a network of 30 nodes, subjected to varying probability Grey-hole attacks

with in some cases $\alpha = 10$ presenting slightly lower results than $\alpha = 5$, amplifying 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 α are extremely close with very low throughput levels.

DSR-Miner

Compared to *AODV-Miner*, since *DSR-Miner* does not embark the *link-cost* in its discovery process itself, there is no immediate limitation regarding field size. Thus, we settled on an increased value of two bytes, allowing for a maximum cost of $C_{max} = 257$, due to the much longer maximum path size of $L_{max} = 255$. The overall simulation parameters are presented in Table 4.6.

Table 4.6: Simulation parameters for the first scenario with *DSR-Miner*

Parameter	Setting
Area	150m×150m
N	30
P_{Ma}	{0% → 100%}
α	{0.5, 1, 2, 5, 10}
f_{size}	16 bits
L_{max}	255
Distribution	Random uniform
Tr Range	50m
W_n^R	5
λ_n^R	Linear
$t_{\frac{1}{2}n}^R$	15 min.
Initial R_n	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

In the same way as previously, we commence our evaluation with the analysis of routing efficiency between *DSR-Miner* and DSR. Here, we once again utilise Black-holes to illustrate perfect malicious entities, dropping every and all packets they encounter. First, we evaluate the number of packets dropped, presented in Figure 4.27.

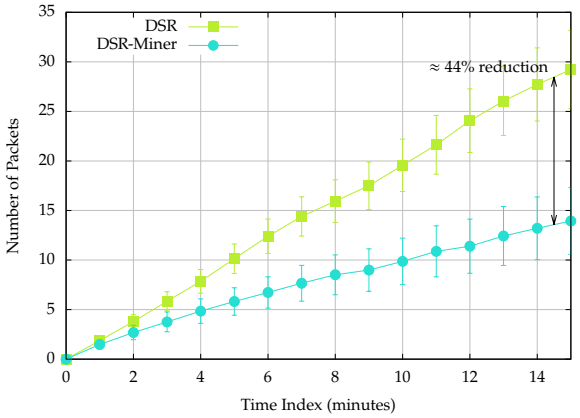


Figure 4.27: Number of packets dropped by both *DSR-Miner* and DSR in a network of 30 nodes with 10% of them expressing malicious tendencies

When compared to the results for *AODV-Miner* shown in Figure 4.21, we can see that *DSR-Miner* (**light-blue** circles) has dropped approximately 14 packets, slightly more then previously wereas DSR alone dropped just short of 30, similar to AODV. That being said, an overall reduction in drop rates can be observed, with approximately 44% less packets dropped, slightly smaller than *AODV-Miner*. However, whereas *AODV-Miner* slowly stabilised overtime, *DSR-Miner* keeps an overall upwards tendency, all the while more horizontal than DSR. What we can also see is that the results for DSR itself are quite unstable, showing that the protocol alone possessed slight difficulties in keeping stable routes.

By taking a look at the overall throughput between both implementations of DSR, shown in Figure 4.28, we can grasp a better impression of this efficiency increase. Indeed, we can see that *DSR-Miner* (**light-blue** hashes) has achieved a higher overall throughput than DSR (**green** crosses). Although this increase isn't as pronounced as with *AODV-Miner*, it is

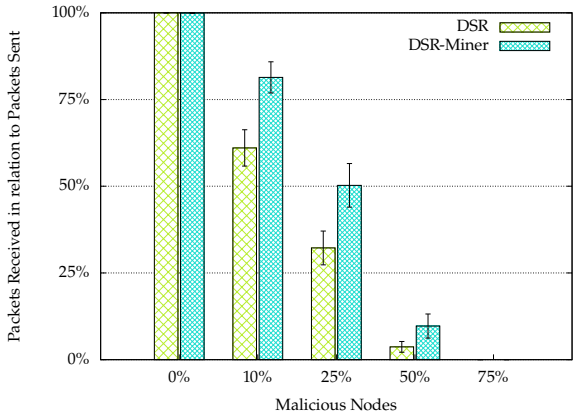


Figure 4.28: Throughput of *DSR-Miner* and DSR in a network of 30-nodes with varying percentage of malicious presence

still present across all levels of disruption, when a clear route is still a possibility. Once again, we can see that this reputational approach has its merits, however, the choice of underlying routing protocol naturally influences how the best routes are selected.

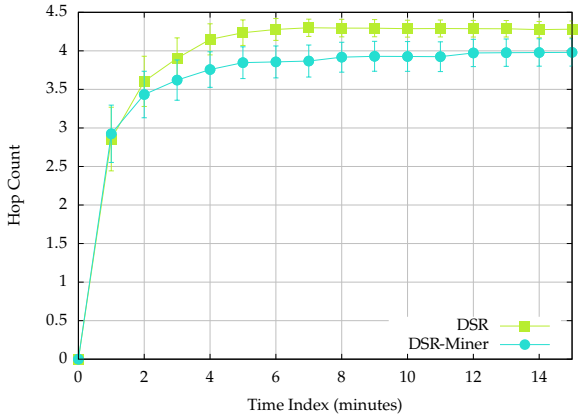


Figure 4.29: Average route length of *DSR-Miner* and DSR in a network of 30 nodes with 10% malicious

Previously, we analysed the overall route length as a trade-off with our reputation module. However, as shown in Figure 4.29, this is not the case with *DSR-Miner*. Due to the implementation specifics with DSR explained previously, it is possible for DSR to select longer routes outright, due to it not using metrics such as hop-count, but functioning on a "first come - first serve" basis. This means that with the addition of the *link-cost* metric with *DSR-Miner*, shorter routes are on occasion selected, when available. That being said, it nevertheless reinforces our previous conclusion that thanks to this metric, we influence the route selection for more trustworthy routes, which significantly impact the length of the routes themselves. On the other hand, there is still one trade-off with our approach.

Indeed, the normalised overhead between *DSR-Miner* and DSR, as shown in Figure 4.30, still shows an increase. That being said, the increase ratio is significantly reduced compared to *AODV-Miner*. This can be explained once again with the specifics of DSR's implementation with regards to route discovery. Since the routes selected by AODV were generally shorter at the start (illustrated in Figure 4.23), DSR naturally transmitted more packets during routing. As a result, the operation itself is a success, all the while showing a slight decreased overhead ratio than previously.

Once again, these simulation results confirm that our reputation-based

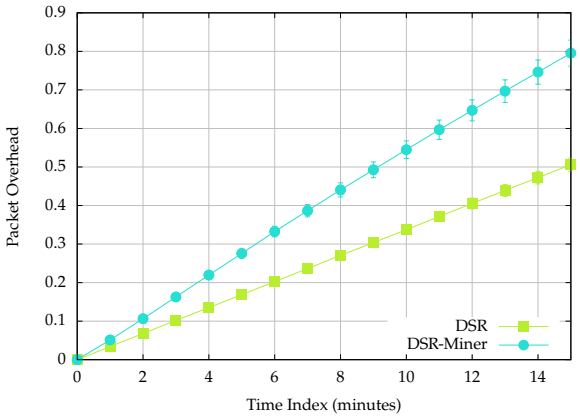


Figure 4.30: Normalised overhead of *DSR-Miner* and DSR in a network of 30 nodes with 10% malicious

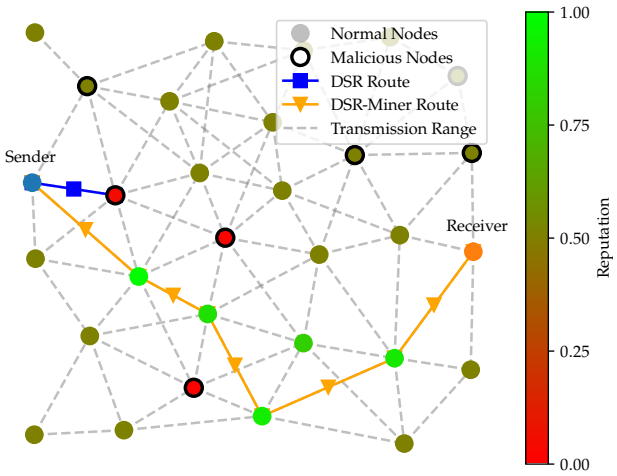


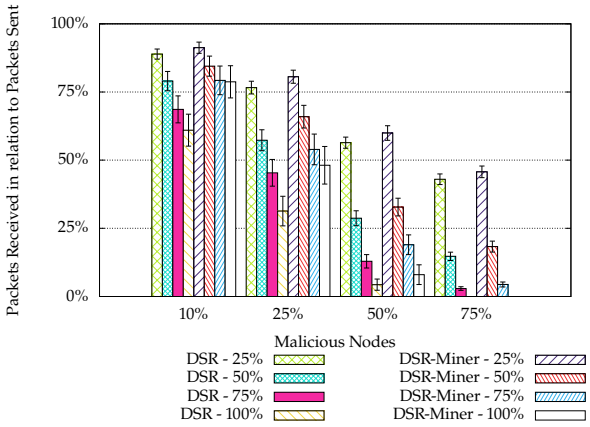
Figure 4.31: Visualisation of route reputation after 15 mins. with *DSR-Miner* and DSR in a network of 30 nodes, 25% of which are malicious

approach allows the identification and avoidance of malicious nodes. However, as shown the path selection algorithm differs from that used by AODV, meaning the most used routes also changes. This is visualised in Figure 4.31 where we present the same 30-node topology as previously. Once again, we can see the 7 malicious nodes represented with thick outlines, making up 25% of the network, all performing Black-hole attacks. Taking a look at the different routes, we can see that DSR has followed in the footsteps of AODV and has gone for the most direct route, falling right into the hands of a malicious node.

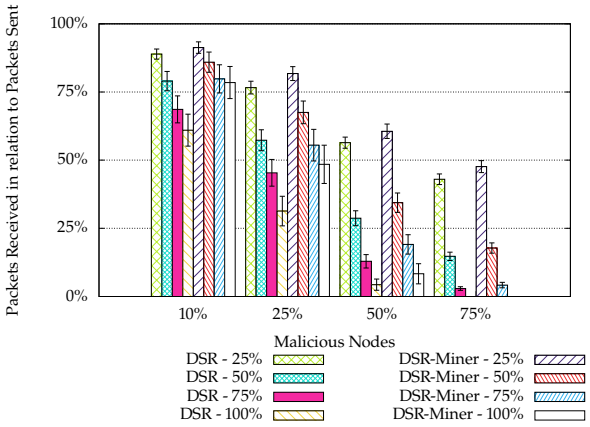
On the other hand, *DSR-Miner* has used approximately five different routes, three of them ending in malicious nodes, which have received a low (**red**) reputation similar to *AODV-Miner*. Indeed, contrary to *AODV-Miner*, *DSR-Miner* has presented less deviations, with only one node varying between routes, all of which have received a good (**green**) reputation. This can be illustrated by the implementation choices used by by DSR and *DSR-Miner* with regards to RREP responses, as presented previously. All in all, we have once again demonstrated that our consensus-module is capable of influencing the route selection process allowing the underlying protocol to detect and avoid attacks.

Threat Adaptation

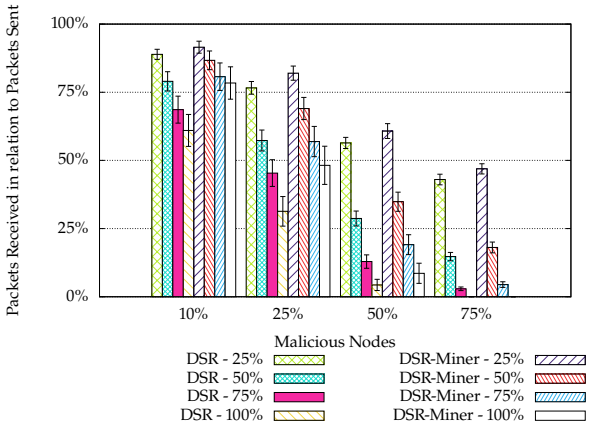
Having demonstrated the efficiency of our module against Black-hole attacks, we can now evaluate its adaptability against various degrees of Grey-holes. As previously, we can vary the drop probability, steadily



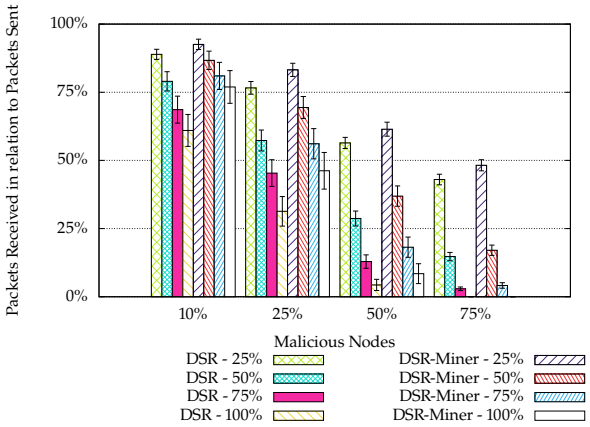
(a) Throughput with $\alpha = 0.5$



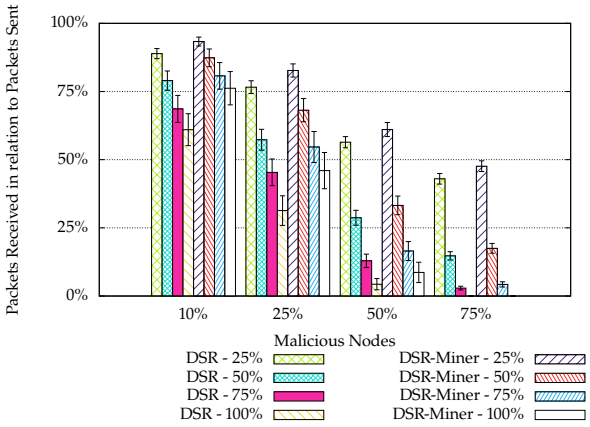
(b) Throughput with $\alpha = 1$



(c) Throughput with $\alpha = 2$



(d) Throughput with $\alpha = 5$



(e) Throughput with $\alpha = 10$

Figure 4.32: Throughput comparison between *DSR-Miner* and DSR in a network of 30 nodes, subjected to varying probability Grey-hole attacks

increasing the impact of the attack itself all the way to Black-holes, as studied previously. Here, we visualise how this probability, coupled with different values of malicious weight α, can directly influence how *DSR-Miner* adapts to its surrounding. This analysis is presented in Figure 4.32, once again varying the Grey-holes probability for the different values of α where $\alpha = 0.5, 1, 2, 5, 10$.

Following on from the results previously observed with *AODV-Miner*, *DSR-Miner* takes longer to adapt the more malicious nodes are present. This can be seen with Figure 4.32a representing a forgiving network with $\alpha = 0.5$, where 10% *bad* nodes reacts quicker, with a higher throughput increase than 50%. These results increase slightly with $\alpha = 1$ shown in Figure 4.32b where we can see that even with 75% of nodes no longer acting on the side of good, the throughput is slightly higher than previously.

When reaching higher values of α as shown in Figure 4.32c, Figure 4.32d and Figure 4.32e with $\alpha = 2$, $\alpha = 5$ and $\alpha = 10$ respectively, the more the results begin to normalise. Indeed, contrary to *AODV-Miner*, here the increase seems to peak with $\alpha = 2$, after which the overall throughput starts to dwindle. An example of this is with 100% malicious activities with *DSR-Miner*. With 10% malicious nodes manifest *bad* intentions, the throughput value descends slightly towards 75%. This is also visible with 25% malicious nodes, where the throughput itself decreases below 50%.

Discussion

In these simulations, we have demonstrated that both *AODV-Miner* and *DSR-Miner* are capable of identifying malicious entries present in the network, using our previously presented behavioural mechanism. Furthermore, thanks to the *link-cost* function presented previously, we are capable of influencing the generic path selection used in both AODV and DSR, allowing us to instead utilise a more trustworthy path, avoiding malicious devices. By analysing both protocols against Black-hole attacks in 30-node networks, we have demonstrated the impact of our approach with a reduced network size and density, increasing the overall level of routing efficiency. This increase, however, comes at a cost, that of increased network overhead, due to the exchanged between Miners as part of the consensus validation phase.

On the other hand, we have demonstrated that our module also influences the average route length, due to it forcing different paths from normal. The main notion of interest here is that with *AODV-Miner*, this average length increases, enforcing that we no longer look for the shortest route but the most reputable. However, this is not the case for *DSR-Miner*, where saw a slight decrease in route length. This discrepancy can be explained through the specific implementation of DSR with "first come - first serve" as explained previously, compared to AODV. Indeed, since our goal is to select a route with the lowest possible *link-cost*, DSR took on some characteristics from AODV which inherently tries to select the shortest route possible.

We also performed an evaluation of the impact of α based on both malicious node distribution and Grey-hole impact. We can see that for

both protocols, α influenced both the response time of the path selection algorithm, all the while keeping the average throughput for all levels of Grey-hole higher than the base protocol. That being said, higher values of α allow the network to punish malicious nodes more severely, providing slight increases with certain malicious intentions.

All in all, we can confirm that our approach provides the necessary information for both AODV and DSR to avoid malicious threats and increase data integrity and network efficiency.

4.4.3 Simulation - Scenario II

Previously, we evaluated both protocols in various network topologies containing 30-nodes. Here, we increase the network surface and density, upping the network to 100-nodes, increasing the overall challenge encountered by our module. By not only increasing the number of nodes, we have not only provided longer routes for the evaluation, but we have also increased the overall network density. As a result, multiple new routes can be perceived, where more interconnections are present in the different topologies. Furthermore, with more connections comes more Miners, thus increasing the complexity and demand of the validation algorithm putting more strain on the underlying consensus. This means we can also determine how the network consensus algorithm can cope with this increase in potential exchanges, as well as the reputation values extracted from the larger quantity of inserted blocks in the blockchain.

AODV-Miner

Once more, we present the specific simulation parameters for *AODV-Miner* in Table 4.7. All others remain identical to the previous scenario, allowing a comparison of effectiveness between the two.

Table 4.7: Simulation parameters for the second scenario with *AODV-Miner*

Parameter	Setting
Area	300m×300m
N	100
P_{Ma}	{0% → 100%}
α	{0.5, 1, 2, 5, 10}
f_{size}	8 bits
L_{max}	64
Distribution	Random uniform
Tr Range	50m
W_n^R	5
λ_n^R	Linear
$t_{\frac{1}{2}n}^R$	15 min.
Initial R_n	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

Once again, we begin our evaluation by analysing the routing efficiency, this time in a much larger network. Once again, AODV and *AODV-Miner* are evaluated through the use of Black-hole attacks. The first metric once again is the number of packets dropped, shown in Figure 4.33.

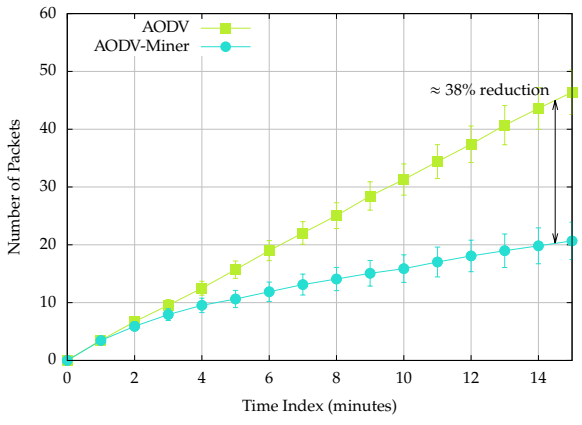


Figure 4.33: Number of packets dropped by both *AODV-Miner* and AODV in a network of 100 nodes with 10% of them expressing malicious tendencies

Similar to the previous scenario, we can see that after 15 minutes *AODV-Miner* (**light-blue** circles) has possesses a lower number of dropped packets than AODV (**green** squares). There, we can see that a total of just over 20 packets have been dropped by *AODV-Miner*, compared to approximately 48 with AODV. Overall, we saw a reduction in drops by approximately 38%, 10% less than the previous scenario, all the while remaining at approximately half of AODV at the end of the simulations. Furthermore, we identify once more that *AODV-Miner* is once again stabilising overtime, contrary to AODV's continuous increase.

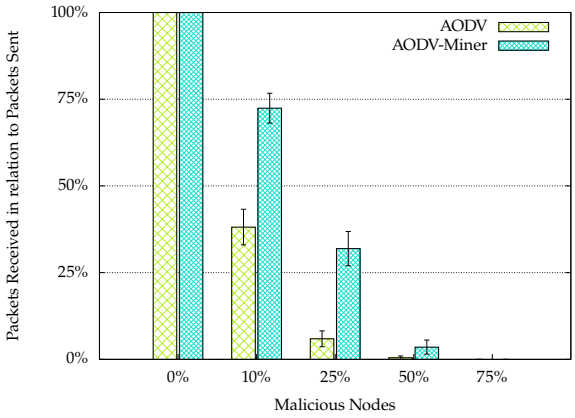


Figure 4.34: Throughput of *AODV-Miner* and AODV in a network of 100-nodes with varying percentage of malicious presence

Once again, an analysis of the overall throughput between these two protocols, presented in Figure 4.34, allows us to confirm this analysis. We can immediately see once more that *AODV-Miner* (**light-blue** hashes) possesses higher throughput values than AODV. However, contrary to previously, the higher the malicious overtake is on the network, the small this increase becomes, due to the network's overall dimensions. All in all, *AODV-Miner* is still more efficient than *AODV* when it comes to reducing the impact of malicious attacks on the network.

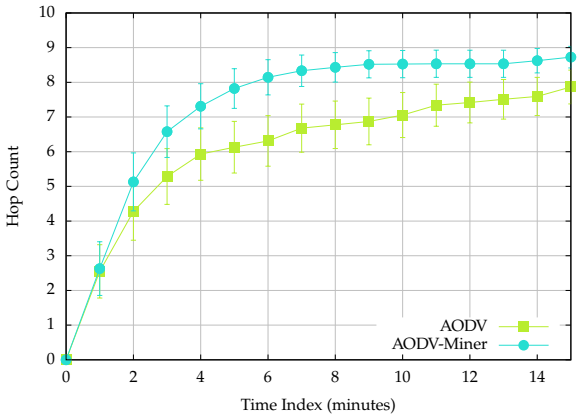


Figure 4.35: Average route length of *AODV-Miner* and AODV in a network of 100 nodes with 10% malicious

Next, we can look at the trade-offs for this increase in efficiency, starting with the overall route length, presented in Figure 4.35. Again, the number of average hops is much higher with *AODV-Miner* than with AODV, indicating that the reputational metrics are playing their part in the route selection process. Furthermore, we can see that this increase in hops is proportionate to the network size, with a much higher difference here than with 30-nodes.

Consequently, the normalised packet overhead has also increased, as

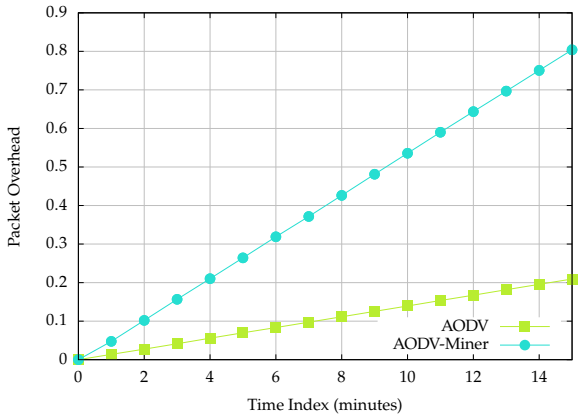


Figure 4.36: Normalised overhead of *AODV-Miner* and AODV in a network of 100 nodes with 10% malicious

shown in Figure 4.36. Similarly, this is a consequence of the increased network size for multiple reasons. The first, is more Miners are needed to observe the routes which from the start are much longer. This also impacts the transmissions made during the validation process as more Miners need to exchange to confirm their observations. The second is, with more Miners also comes the increased number of blocks needing to be disseminated throughout the network.

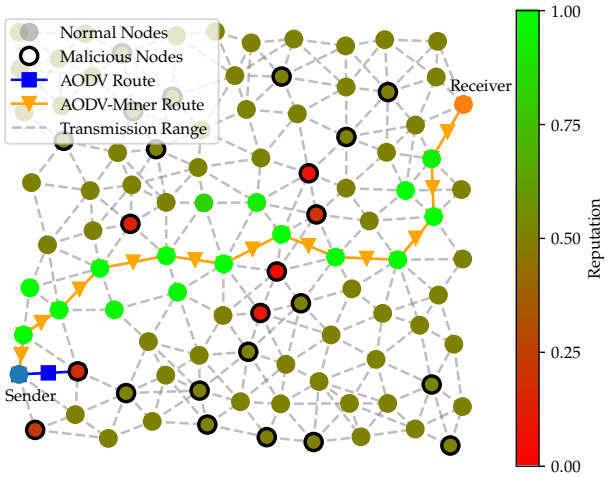


Figure 4.37: Visualisation of route reputation after 15 mins. with *AODV-Miner* and AODV in a network of 100 nodes, 25% of which are malicious

As expected, the trade-offs are amplified in a larger network which consequently is also the case for the number of malicious nodes present. To visualise this, Figure 4.37 presents a significant 16 nodes possessing a high reputation, represented in **green** and 7 with lower values, represented in **red**, four more than the network of 30-nodes. We can also see a cluster of four malicious nodes in the centre of the network separating the source from the destination, all of which have been detected and subsequently avoided. One final note is that, as is the case with AODV, the route selected may on occasion change due to various reasons, from interference to random MAC layer back-off values. We can see this with the fact that once again there are nodes which possess good reputations and yet are not part of the most used route. Indeed, this is possible to to route change, from either encountering malicious nodes, as is the case in the centre quadrant, or simply network variations, causing them to jump an unneeded node, as seen on the right hand side. Furthermore, we can once again see that one node in the upper-left-centre quadrant is shaded in a much deeper shade of **green**, indicating that it was used some time

ago, giving its reputation time to decay.

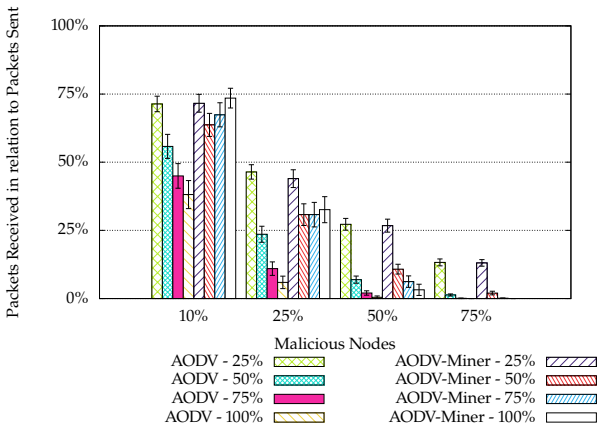
Threat Adaptation

The next comparison provides an overview of how *AODV-Miner* is capable of adapting to varying threat levels, here represented by using varying probability-based Grey-holes. As stated previously, we can utilise the value of α to influence how the network responds to each threat, allowing to identify if the best value changes dependant on the size of the network. Once again, we use the same base values of $\alpha = \{0.5, 1, 2, 5, 10\}$, with the analysis results presented in Figure 4.38.

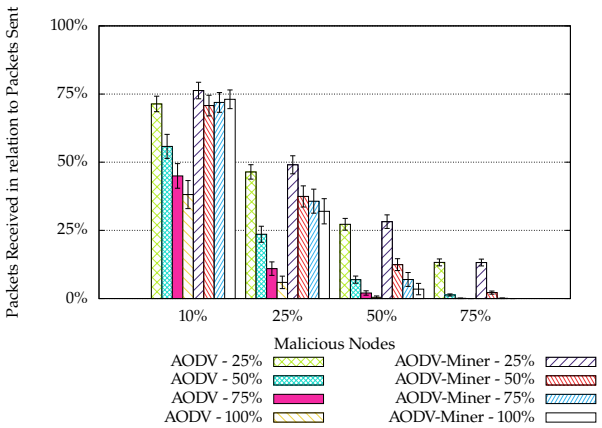
From this analysis, we can strengthen our initial hypothesis. First off, we can see that in general the larger network size has resulted in an overall decrease in throughput level, due to the presence of more malicious nodes, as illustrated in Figure 4.37. By beginning our analysis once more with $\alpha = 0.5$ in Figure 4.38a, 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 when dropping 100% of packets. This is also the case with 25% malicious nodes, although the increase is more subtle than the 30-node network in Figure 4.26a. 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 α 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 3.18.

Increasing the value of α consequently increases the overall throughput, although some parallels with the low value of α can still be made. This is the case for $\alpha = 1$ in Figure 4.38b, 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.

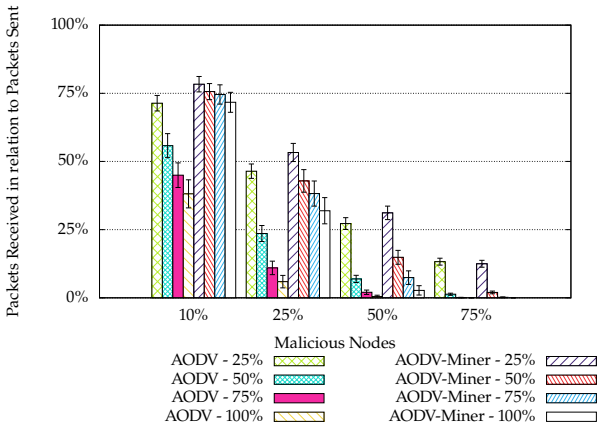
Increasing the influence of bad actions, visible in Figure 4.38c, Figure 4.38d and Figure 4.38e demonstrates the advantages but also the disadvantages of higher values. If we turn our attention to the results for 25% malicious probability, we can see the corresponding throughput increases the higher the value of α, 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 α 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 α, 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 the same route, up to *four* malicious nodes can be detected, increasing the efficiency of *AODV-Miner*. As a consequence, the higher the malicious probability, the longer it takes to



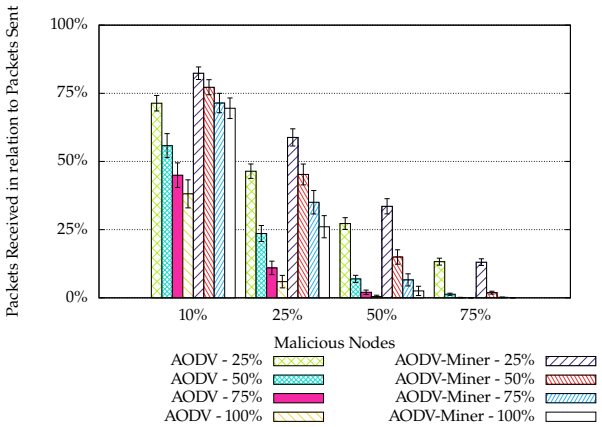
(a) Throughput with $\alpha = 0.5$



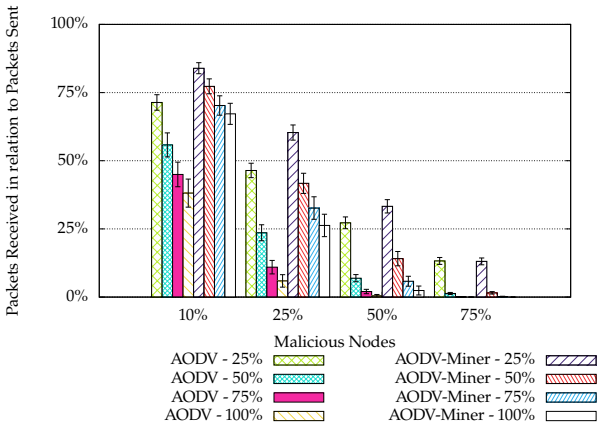
(b) Throughput with $\alpha = 1$



(c) Throughput with $\alpha = 2$



(d) Throughput with $\alpha = 5$



(e) Throughput with $\alpha = 10$

Figure 4.38: Throughput comparison between *AODV-Miner* and AODV in a network of 100 nodes, subjected to varying probability Grey-hole attacks

detect and circumnavigate malicious nodes, indicating that a dynamic adjustment of α might be beneficial.

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 only *three*, further decreasing to *two* for 75%, ending up with only a *single* node when Black-holes are used. 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 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 nodes are beyond hope, even by changing the route constantly in an effort to reach the destination, it is highly unlikely to find a clear path to the destination. This is illustrated by the fact that *AODV-Miner* results in a lower throughput for 25% malicious probability than AODV, where the significant presence of malicious nodes simply hinders the overall performance.

DSR-Miner

Out next set of simulation scenarios revolves around 100-node networks, allowing to test the implementation in a much larger scenario. However, in this case 100-node networks are more complicated to be simulated with DSR. As stated previously, DSR utilises IPv6 extension headers to contain the information for route discovery as well as routing. Indeed, the *Source Routing* header contains the exact sequence of hops needed to go from the source to the destination, thus eliminating the need for intermediate nodes to store routing information. Although this is one of DSR's advantages, this is also its main weakness.

As we have mentioned, DSR was devised for IPv4 networks where the address size is four times smaller than IPv6. This means that more space is needed in the header to contain the relative information. Thus, even with address compression inspired from `LOWPAN_IPHC` header compression utilised in 6LoWPANs, the maximum possible route is reduced to only 7 hops, as demonstrated in Section 4.3.2. Although this is sufficient for 30-node networks, this is insufficient with 100-nodes. Indeed, the analysis of *AODV-Miner* with 100-nodes presented in Section 5.4.3 shows an average hop-count in the vicinity of 8 to 9 hops.

Due to this difference, we can conclude that DSR would not be able to establish routes across all 100 topologies, thus severely impacting the overall results. Thus, we can conclude that DSR is not adapted to 6LoWPANs, in particular with the use of IPv6 addresses. For illustration purposes, with the same MTU payload size of 102 as previous, by using IPv4 addresses, the number of hops would be able to increase to 14, easily allowing routing activities in large networks. Going the other way, without our IPv6 compression, the number of possible hops would be halved, allowing only 3 hops per route, rendering it practically impossible for any routing activities to take place, unless in very small networks.

Discussion

Here, we evaluated our module in larger and denser networks of 100-nodes. However, due to limitations on the implementation, it is not possible to simulate DSR with the current system, thus stopping us from demonstrating the continuous impact of our module. Thankfully, this is not the case of AODV, allowing us instead to focus on *AODV-Miner*, demonstrating once more the impact of our module on network efficiency and integrity. Indeed, can see that the overall efficiency against Black-holes is still higher than AODV, however, this increase is slightly reduced compared to our first scenario. We can, therefore, conclude that the network size itself directly influences the efficiency level, due to more malicious entities being present in the network and the more time needed for our module to detect a more trustworthy route.

Once more, both average route length and packet overhead are much higher with *AODV-Miner* with a significant increase compared to the smaller network, due again to network complexity and the existence of more routes and more Miners per route. We also illustrated once more the impact of α upon the detection process against different levels of Grey-hole. However, we can also notice that here the results are slightly lower than in the first scenario, similar to the overall routing efficiency. It is even noticeable that in some cases, the network complexity causes our module to struggle in finding a clear path, sometimes ending in malicious dead ends, thus reducing the throughput compared to normal.

However, once again we can confirm that our module increases the overall network efficiency, allowing AODV to continue to avoid as many threats as possible.

4.5 Conclusion

In this chapter, we addressed the issue of integrating the reputational values computed in Chapter 3, with existing multi-hop routing protocols, in particular the most popular: RPL, AODV and DSR. We examined existing protocols, as well as the generic path selection algorithms to determine how these paths are formed throughout the network, the most common of which is the least number of hops. By proposing a *link-cost* metric to influence this value based upon a nodes reputation, it is possible to trick the algorithms into avoiding malicious nodes. We illustrate this approach in relation to two *Reactive* routing protocols: AODV and DSR However, to grant the system the capabilities of behavioural observation, both protocols received small updates to their existing packet formats, providing much needed information to the Miner module. We then performed in-depth simulations of both protocols in networks of 30-nodes, pitching both *AODV-Miner* and *DSR-Miner* against Black-hole and Grey-hole attacks. We also extended this analysis to networks of 100-nodes with *AODV-Miner* due to size constraints with *DSR-Miner* header size. We saw an increase in efficiency against Black-holes by $\approx$ 48% and $\approx$ 44% with both protocols respectively in networks of 30-nodes, and $\approx$ 38% in networks of 100-nodes with *AODV-Miner*. Through this analysis we can confirm the impact and importance of our module with *Reactive* protocols with regards to routing security. Finally, we performed

a theoretical overview of a possible integration with *Proactive* protocols with RPL, utilising the *link-cost* to influence node "rank" which is used in the constriction of RPL Directed Acyclic Graphs (DAGs). This overview also proposed an extension to the Miner module, in particular the RVTs as well as the behavioural analysis itself, allowing the validation of traffic on *Proactive* networks.

5 Adaptive Network Quarantine

With the ability to detect threats when they occur during routing, it is indeed possible to adapt the protocol to function at peak functionality. Indeed with the *link-cost* function, it is possible to effectively artificially extend the route's length based on each used nodes reputation. However, in some cases this does not eliminate the threat. As demonstrated previously, it is possible that using a malicious node with a high link cost (i.e., 4), would still be considered better than using five nodes with a low link cost (i.e., 1 each). This is a limitation of the reputation and *link-cost* metrics as they do not modify the functionality of the routing protocols, but instead influence the decision making process.

In the medical domain, when a threat on a biological organism is detected, a threat response is triggered, generally lead by the organisms immune system. When we come to human healthcare, immune response is not taken lightly and in many cases the way of life of the infected individual is affected (i.e., severe symptoms, isolation, etc.). Examples of this have been seen recently with the COVID-19 pandemic, where people who have been exposed to the virus were required to self-isolate and quarantine themselves for a certain period of time. In doing so, the person's immune system was given enough time to respond to the threat, all the while reducing the spread of the illness.

This functionality is also present in cyber-security. Indeed, many anti-virus systems possess a "Quarantine" or "Vault" in which malicious files are kept, an example of which is shown in Figure 5.1. Its main goal is to stop the user, who is on occasion unaware of the threat, from accessing the infected data or running a malicious program, which would cause further damage or spread of the virus. In this case, the files are quarantined indefinitely, or until the user either deletes them from their system, or removes if from quarantine. Threat isolation is also operable in network environments, where servers can be artificially isolated from each other if one of them has either been compromised, or is in a weakened state (same as an immunocompromised human). However, the only methods which is infallible against network based threats is complete network isolation, in other words the device is disconnected from all network access.

Although very efficient, this method is basically impossible for a device whose main functionality is to operate in a networking environment, such as IoT networks. That being said, it is still possible to adapt a form of quarantine to a distributed networking scenario, granting the ability for all nodes to determine and respond the malicious threats. Furthermore, by adding quarantine functionalities to our already existing reputation system, we can add the ability to isolate nodes from network operations when they post too much of a threat. However, the notion of "too much of a threat" needs to be defined in such a manner that it is both verifiable and robust. By updating the existing reputation metric, we can make sure to isolate extremely malicious entities, not only from network operations, but also from activities related the consensus-module itself.

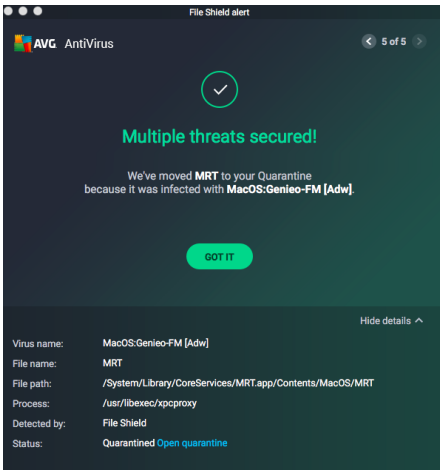


Figure 5.1: AVG AntiVirus threat alert about multiple threats on a MacOS system. Once alerted to the user, the infected files have been moved into the AntiVirus' quarantine facility for isolation [206]

[206]: AVG AntiVirus. *AVG Support Community - MacOS:Genieo-FW[Adw]*. Sept. 7, 2018. URL: https://support.avg.com/answers?id=9060N000000g9SNQAY (visited on Aug. 29, 2022)

As stated, the goal of IT quarantine is to isolate devices from each other. In a network context, this means prevent the concerned device from participating in network functions, such as routing or data transmission. In the case of multi-hop networks, routing through intermediate devices causes a significant security risk as they must be explicitly trusted to relay the data intact to the correct destination.

We have shown with the consensus-based reputation metric that we are capable of not only detecting if a malicious action takes place, but also identify the origin of these actions. Through reputation-based routing, we can avoid using these entities, thus reducing the overall impact on the network. However, in many cases, due to the use of a link-cost function, the routing protocol still selects a route using the malicious node, as its cost is still lower than using normal nodes (i.e., malicious node with link-cost of 4 is equal to 2 normal nodes with link-cost of 2, or 4 good nodes with link-cost of 1). As a result, although the system functions correctly, it is not sufficient to isolate and avoid all malicious nodes.

This is where the notion of quarantine enters play. By placing a malicious node into quarantine, we indicate to the network that it must not be used for routing activities. Thus, whatever the link-cost of the surrounding nodes, the route will naturally avoid the bad entity, further increasing the performance. However, nodes must not be put into quarantine prematurely, as isolating good nodes following false positives will impact the network more than not using quarantine metrics.

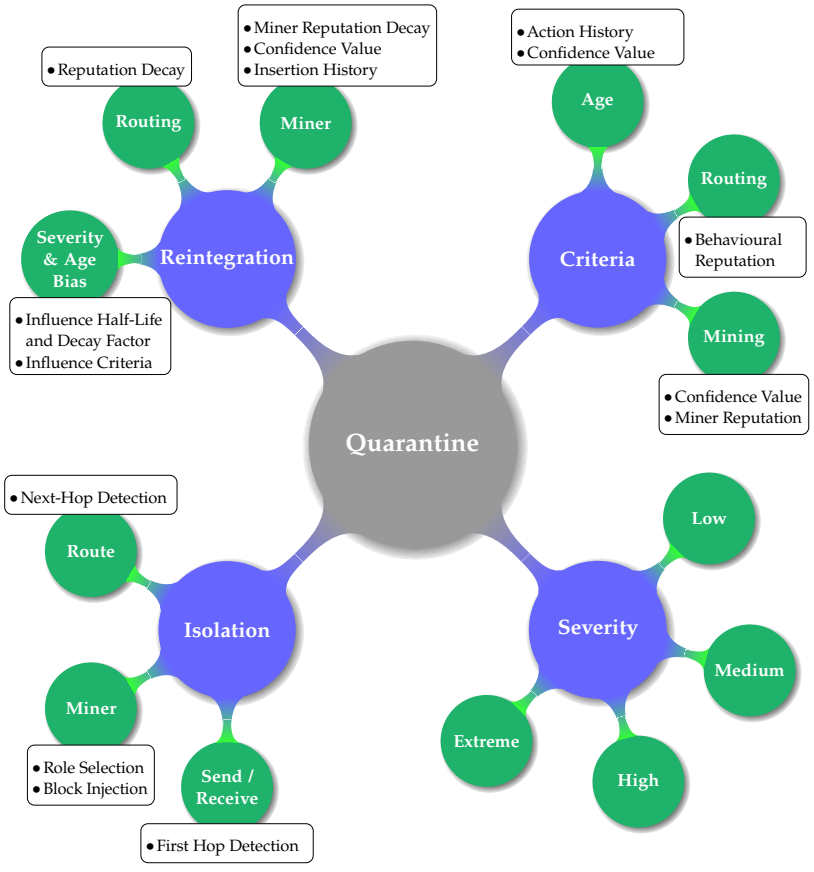


Figure 5.2: Mind-mapping of the ideas around the notion of network quarantine, based upon the Miner module

Figure 5.2 presents an overview of the proposed Quarantine system. There are four overall categories, each corresponding to a specific element of the system, depicted in **blue**:

- **Criteria**: The elements which determine if a node is to be put into quarantine.
- **Severity**: The different levels of quarantine, and their consequences on a node's activities
- **Isolation**: The act of stopping a node from communicating and participating in network activities based upon its level of threat severity
- **Reintegration**: Removing a node from quarantine and allowing it back into the network

Each of these elements contains specific characteristics which influence the functionality of network Quarantine, visible in **green**. We will present these elements one by one and will also define their respective characteristics.

5.1 Increasing Network Security

The first important stage to protect a network from malicious actions is the ability to detect and respond to various threats. By providing the necessary tools and functionalities to not only detect routing threats, but also threats towards the mining process, it is possible to separate good nodes from bad. With this distinction, it would be possible to enforce quarantine restrictions on these nodes, effectively removing them from the network and reducing their accepted operations.

5.1.1 Detection Criteria

For a node to be put into quarantine, certain characteristics must be taken into account. First off, their activities with regards to how they route data packets must influence the trust level of the node. Thanks to the approach presented in Chapter 3, these activities can be monitored with malicious nodes being identified and subsequently avoided. However, with dynamic role selection, nodes can also become miners for a specific route, where they are charged with observing the behaviour and update the action list of their neighbouring routing nodes. As presented previously, this role is crucial to the core functionality and as a result must not be taken for granted. This means that a new metric determining the trust level and the confidence value of the node must be computed, separate from routing reputation impacting specifically these activities as a Miner. This value, neutral at the start similar to the routing reputation, will also allow role selected based upon their level of confidence, allowing only the most trustworthy nodes to Mine for a route. That being said, this alone is not sufficient as Miners can still be compromised.

To combat this, we propose an upgrade to the consensus methodology based upon Byzantine Fault Tolerance (BFT) presented in Section 5.3, solidifying the selection of the most efficient Miner for block insertion. Previously, this determination was made based upon passing blocks, where Miner's analysed the contents and identified if they were more efficient or not, transmitting their own if this was the case. With this approach, however, not all Miners were aware of what was happening, as each Miner woke up at a random moment in time to share its data.

This means it is possible for a Miner not to wake up if it was overridden by another Miner before hand. By allowing all Miners to wake up and share their data with each other, they will all gain an overview of what has occurred in their vicinity. As a result, not only will they be able to better determine the most efficient in the group, but it also provided the ability to confirm the Mining activities of each neighbouring Miner.

The selection of this most efficient Miner can also be determined based on other criteria than routing or mining actions. Indeed, it is also possible to take into account how long the node has resided and acted in the network. The routing reputation is calculated based upon a small portion of the node's history thanks to the reduced window size W_n^R as seen in Equation 5.1. As a result, with a window size of 5, the node only needs to perform 5 single correct actions to erase the malicious history and return to a good standing in the network. By adding a lifetime value which will always evolve, based upon the ratio of *good* to *bad* actions, even if the node has performed no bad actions in a certain time, it would be possible to determine its overall trustworthiness. By providing this ratio in the form of a confidence value, it would be possible to influence the reputation of both routing and mining, as well as the role selection process, thus allowing the network to always remember previous achievements.

$$S^R_{good_n} = \sum_{i=1}^{W_n^R} good\ actions_{n_i^R}$$
$$S^R_{bad_n} = \sum_{i=1}^{W_n^R} bad\ actions_{n_i^R} \tag{5.1}$$

These equations are reminders of Equation 3.1 and Equation 3.2 from Chapter 3

5.1.2 Adaptive Quarantine

When it comes down to it, isolating a node from routing activities is a rather simple procedure. Indeed, thanks to the use of the *link-cost* derived from a nodes routing reputation, it is already possible to remove certain nodes from routing activities if the value is too high. In doing so, it is possible to not only block transmissions originating from a malicious sender, but also avoid the routing protocol using a malicious node to forward data on-wards. This is not the case, however, for blocking a node from receiving data, since the destination is rarely analysed to determine its trustworthiness. By performing such an analysis on the first hop, not only would it allow to block any unnecessary traffic, but it would also allow the introduction of a new error type into the routing protocol, indicating that the requested receiver is malicious and has been quarantined. However, no error should be provided if the source is malicious, as this would inform the attacker that the network is aware of their actions, which would impact the overall efficiency as they would most likely shift targets.

Isolating Miners on the other hand is slightly more complex. Firstly, it would be possible to influence the choice of Miners during the role selection process, allowing nodes to take on the role ONLY if their trust and confidence values are in a certain range. However, this doesn't stop nodes which force the mining of routes to inject malicious information into the chain. To stop this, we can utilise our previously presented upgrade of the consensus mechanism, to allow fellow miners to unanimously confirm a Miner's trustworthiness in the same fashion as the routing nodes. Furthermore, if a malicious Miner still proceeds to inject a block into the blockchain, other miners will recognise this irregularity as they didn't agree on it and, therefore, refuse to relay it on-wards, decreasing their Mining reputation as a consequence.

5.2 Dynamic Response Severity

Now that nodes have been isolated from network operations, it is necessary to analyse their activities and determine how they can be reintegrated. Indeed, as is the case with medical quarantine, the duration depends heavily on the nature of the illness, as do the restrictions imposed on the patients quarantine. These aspects can be transferred also to network security, where a detected threat can receive a corresponding threat level, thus easily identifying what it can and cannot do. Furthermore, based also on its history, how said node is reintegrated back into networking society can also be adjusted, allowing the network to continue to function, all the while increasing the severity of the threat response.

5.2.1 Overall Threat-Level

Quarantine is generally thought of as a binary action: isolated or not. However, we can add varying degrees of isolation to the network based upon the severity of the threat posed by the node. Table 5.1 shows an overview of the threat level and the actions which a node would be allowed to take in the network, as presented previously. As we can see, the higher the level, the less activities the nodes can perform up to the extreme case of complete isolation and no longer being able to participate in any activities. For both Low and Medium levels, we add the extra element where the node will only be allowed to contribute if no other solutions are possible (denoted by the ✓*). This is currently the case with the reputation metric, where with the increase of the *link-cost*, nodes are generally avoided, but can still be used if no other route exists, or it is simply too long. As a result, the reputation approach corresponds to the equivalent of a low-level quarantine.

Security Level	**Network Actions**			
	Send Data	*Receive Data*	*Route Data*	*Mine a Route*
None	✓	✓	✓	✓
Low	✓	✓	✓*	✓*
Medium	✓	✓	✓*	✗
High	✓	✓	✗	✗
Extreme	✗	✗	✗	✗

Table 5.1: Definition of the different degrees of Threat Severity and their impact on network actions.
✓ = *accepted*
✓* = *if no better option exists*
✗ = *denied*

The different levels are determined by the activities and criteria of the node, in particular the confidence value which will evolve forever throughout the lifetime of the node, as well as both the precise routing and mining reputation. Since the reputation values are computed based on the most recent actions, they will allow the selection of the immediate threat level, allowing to isolate the node as soon as possible. However, thanks to the confidence value, we would be able to stop malicious nodes from increasing the standing in an attempt to corrupt the network. Indeed, the lower the confidence value, the more chance their would be of a higher threat level, thus severely punishing recurring offenders.

5.2.2 Progressive Device Reintegration

The final element of network quarantine is the reintegration of isolated nodes back into the network, corresponding to the recovery phase of

Incident Handling, previously presented in Chapter 2.3.3. As explained, this stage generally revolves around resetting the system to a prior stage, thanks to services such as backups or cloud storage. However, here we look more towards short term reintegration, where devices can be reintroduced back into the network. Thus, we turn slightly off the beaten path, and not interest ourselves with system recovery, but towards an advanced method for device reincorporation into the network.

This can first off be performed following the level of threat severity, where nodes will only progress one level at a time. This lengthy process will stop malicious nodes performing constant stop-and-start type attacks where they impact network performance, stop whilst the reputation heals itself, then restart their attack. In our case, constant and prolonged attacks will severely decrease the confidence value, thus increasing the threat level to which the node is transferred, increasing the time needed to return to normal operations.

As shown previously, *Reputation decay* allows a nodes reputation to return to neutral overtime without touching the actions performed. Here the system will remain the same, however, we can incorporate the level of severity into the mix. Indeed, the higher the threat level, the longer the decay takes. By changing the reputations half-life and decay factor the higher the threat level, we can increase the length of time before the node can return to normal operation. Table 5.2 shows an overview of how such a procedure would function, with suggestive values for implementation, based upon the reputation reintegration presented previously. Here we can see that the higher the threat, the higher the value of $t_{\frac{1}{2}n}$ the reputation half-life for node n, increasing the time needed for the reputation to return to normal. Here, the value is doubled when the threat severity reaches *High*, this increasing the impact of high level threats. Furthermore, we can see that λ decreases, indicating the decree of decay for the corresponding half-life. As a result, the more the threat level increases, the more the corresponding node is punished, subsequently increasing its quarantine time. It is to be noted that here, both threat levels *None* and *Low* possess the same decay factor and half-life. This choice was made as the reputation decay function, previously shown in Chapter 3.2.3, corresponds to both of these threat levels, thus allowing us to extend the base functionality.

Table 5.2: Definition of the different degrees of Threat Severity from Table 5.1 and how they would impact with the reintegration decay functionalities, here decay half-life (in seconds) and the decay factor.

Security Level	**Decay Values**	
	Half-Life ($t_{\frac{1}{2}n}$)	*Decay-Factor* (λ_n)
None	900	0.25
Low	900	0.25
Medium	900	0.2
High	1800	0.15
Extreme	1800	0.1

We can also add a bias based upon the age of the node, in particular the nodes overall confidence value. With a high value, we can determine that this node has been good for the vast majority of its lifetime, and as a result we can shorten the return to normal. However, this would only be useable on lower-level threat node, as no node is protected from being compromised or impersonation. Thus, the confidence value will only influence the decay rate if the severity is below a certain level, thus allowing the benefit of the doubt to older trustworthy nodes, but only if they haven't performed a large attack. With the reputation value

decaying back to neutral, we can also begin to reintegrate Miners with their mining responsibilities. To do so, we can allow the node to become a miner once more only after a certain duration where it hasn't performed any malicious activities. This reduces the impact as a malicious routing attack will cause trouble, but trustworthy miners will detect and once more quarantine the culprit. However, if the malicious node infiltrates the Miners, it can corrupt the blockchain, thus impacting routing more severely by quarantining normal nodes and hiding malicious activities.

5.3 Advanced Network Consensus

The approach presented previously in Chapter 3 allows to achieve a basic consensus with as few communications as possible. However, this consensus is inherently flawed, as there is no protection as to the identity and legitimacy of the participating Miners. Furthermore, the system functions on a non-existent trust system, where Miners are expected to respect having their blocks overridden, without continued verification. This means that malicious Miners could continue to send their blocks for validation until no response is received, marking them as the overall winner, leaving them to insert their invalid data into the blockchain, corrupting the reputation at will. As such, a review of this consensus methodology is necessary, not only to render it more robust, but also providing the capabilities to analyse and validate the activities of the Miners. With this, it would be possible to identify malicious entities and inform surrounding Miners who is authorised to insert into the Blockchain, thus combating and correcting potential malicious insertions.

This new consensus mechanism replaces the entirety of the aforementioned block validation phase, leaving the primary behavioural validation mechanism unscathed, simply proposing an updated metric combating the problems mentioned above. To accurately achieve this, this new mechanism, inspired by how BFT tolerates errors whilst maintaining consensus, is itself split into two parts: *Routing Activities Validation* and *Miner Activities Validation*. In this section, we will take a look at both parts individually, explaining and illustrating their functionalities and why they are not only necessary, but provide a much-needed increase in robustness. An in-depth illustration using three distinct scenarios is available in Appendix C.

5.3.1 Byzantine Problem

To understand how this consensus mechanism functions, we need to take a look at the fundamental concepts behind BFT, in this case the well documented *Byzantine Generals Problem*. The general concept revolves around the need for military generals to agree on their planned assault against a target, all the while being able to detect attempts to deceive the armies by traitors. This problem based upon game theory has been formalised and expressed for the domain of IT system reliability in [207] and is a fundamental concept of how systems function today.

[207]: Leslie Lamport, Robert Shostak, and Marshall Pease. 'The Byzantine Generals Problem'. In: *Concurrency: The Works of Leslie Lamport*. Association for Computing Machinery, 2019. DOI: 10.1145/3335772.3335936

In short, the authors define that IT components should be able to tolerate false or lack of information provided by a compromised or faulty piece of hardware of a connected system. Since such failures become more and more unavoidable the more systems are inter connected, it is important to protect and allow continued operation no matter what. Indeed, when looking at distributed systems, the overall system should be able to continue if a certain proportion of its components no longer function correctly. This approach is also used in modern networking and forms one of the backbones of internet structure, where if one or more routers fail, there is virtually no, or very little impact for the users.

If we adapt this to a multi-hop network paradigm, we can see that the fundamental concepts are quite similar. Indeed, the network should be able to function even if some of the nodes are compromised or have failed. However, due to the nature of multi-hop networks, it is not always possible to guarantee the continued function of the network as a whole dependant on which node is the problem. For example, in an evenly distributed network, one node no longer functioning will have little or no impact, whereas a dual sided network transiting through a single point would be completely disrupted if this central point was compromised.

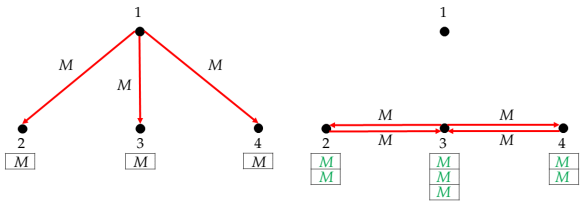


Figure 5.3: Illustration of the BFT *Byzantine Generals Problem* where all generals are loyal.

To understand how the Byzantine Consensus functions, we will present an example of the *Byzantine Generals Problem* in a network context. Figure 5.3 shows an overview of the Byzantine Consensus methodology in a four-node network with node 1 being the commander in chief and the other three being their generals. Here, we have node 1 which sends a message M to nodes 2, 3 and 4, representing the commander in chief giving orders to their army commanders. These nodes exchange the received message amongst themselves to confirm it is valid. In this case, all three have received the same value and as a result achieved consensus amongst themselves, agreeing that the message M is valid, continuing their operation. In this case, the generals would be able to follow the commander's orders with no problem.

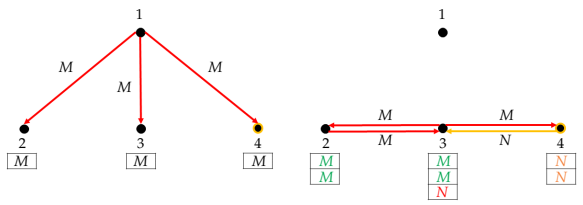


Figure 5.4: Illustration of the BFT *Byzantine Generals Problem* where one of the generals is malicious and attempts to corrupt the commanders orders.

The advantage of such a process is the apparition of a malicious entity, as seen in Figure 5.4 where one of the generals, here node 4, has defected to the enemy. We can see that node 1 sends the same message M to all nodes, but node 4 wants instead to send N. During the exchange with their neighbouring nodes, node 3 identifies it has received two messages containing M and one containing N. As a result, it determines that M is

the correct message and that node 4 sent malicious data. Even though one of the generals attempts to stop the corresponding attack, thanks to node 1 being in range of the other generals, they are able to determine the correct orders issued by the commander and attack as planned. With this approach, the third general is also capable of raising the alert and removing the fourth general from their position.

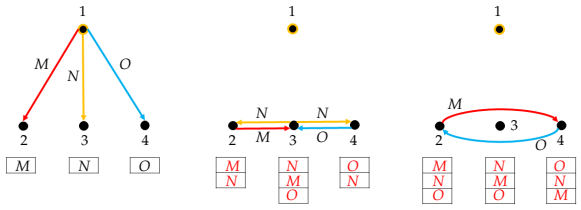


Figure 5.5: Illustration of the BFT *Byzantine Generals Problem* where the commander in chief gives different orders to each of their generals in an attempt to cause disruption in the ranks.

Thanks to this approach, the network is able to determine the correct course of action even when the data corruption is attempted. However, it is also capable of determining complete malicious input and not to take action at all as a result. This is the case in Figure 5.5, where the commander in chief, node 1, is a traitor and attempts to cause disruption by sending different orders to its generals. We can see that the during the validation stage, all three nodes have received different data from their neighbours and are not able to reach consensus. In an attempt to determine the correct course of action, generals 2 and 4 exchange their data with each other, resulting in all three receiving different data inputs. As a result, no consensus can be reached, identifying that the data source is malicious. In this case, the generals decide not to attack as they have all received different orders and cannot coordinate their strategy. As a result, they are also able to identify that something is wrong with the chain of command and as a consequence, relieve the commander of their duties.

By adapting this problem, we are able to detect the differences in the analysed routing activities, presented in Section 5.3.2, but also any Miner attempting to corrupt the validation process, defined in Section 5.3.3. It is, however, important to note that although this process is quite fast as presented in [208], it is also very costly as the number of transmissions is proportionate to the number of neighbours around the transmitter, thus increasing exponentially with the size of the network.

[208]: Miguel Castro, Barbara Liskov, et al. 'Practical byzantine fault tolerance'. In: *Third Symposium on Operating Systems Design and Implementation (OsDI)*. 1999. URL: https://www.usenix.org/legacy/publications/library/proceedings/osdi99/full_papers/castro/castro.ps (visited on Oct. 17, 2022)

5.3.2 Action Validation

The primary goal of this first stage is to perform the correct validation of the determined routing activities as perceived by the surrounding miners and is illustrated in Figure 5.6. This approach, therefore, performs a mix of the Byzantine methodology with the previous validation approach used in the Miner module in Chapter 3. Instead of using random chance to determine which Miner would wake up first and begin the validation process, here all nodes transmit their findings to their neighbours simultaneously. By keeping the same 2-hop limit, we continue to allow the data to reach only the Miners which have analysed the same portion of route. The main difference here is that each Miner, therefore, constructs a table with all received values called *consensus matrix*, as previously seen above with the Byzantine problem in Section 5.3.1. With all received

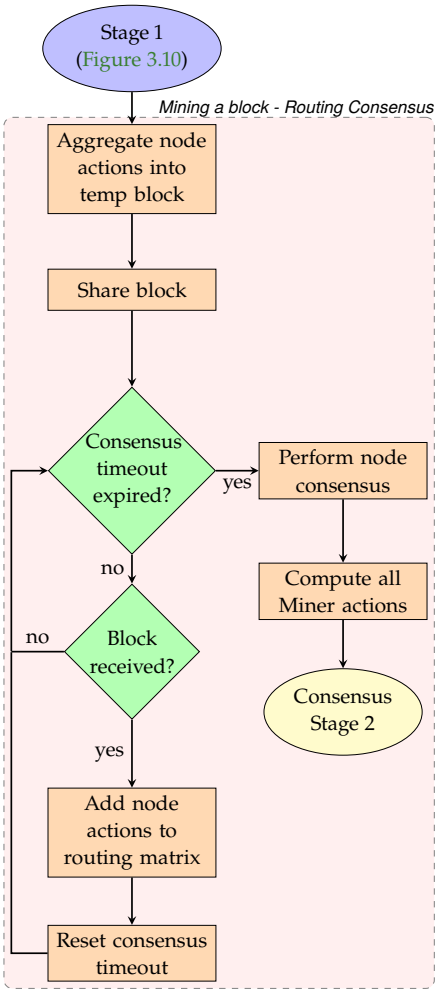


Figure 5.6: Routing Consensus Flowchart

actions present in the matrix, the Miners can achieve consensus not only on the actions contained therein, but also determine the actions of the Miners associated with the observations, ready for stage two.

To achieve this new validation methodology, new packet structures needed to be devised, replacing the original *Verification* packet presented in Chapter 3. The first of these new packets is the *Route Validation* packet with type 4, essentially replacing the old *Verification* packet of type 2. This new packet's structure is presented in Figure 5.7. On close inspection, not much has changed between the two formats. Since we don't need to rely on the correct order of appearance of each packet, due to all Miners analysing them together, neither a unique ID nor timestamp is needed. As a result, they have been removed from this structure, leaving more room for node actions. Is is important to note that, as was presented previously, the packet structures themselves have been slightly compressed, containing only the IPv6 suffix, as we presume all network nodes possess the same IP prefix, allowing to save overall space, increasing the payload size.

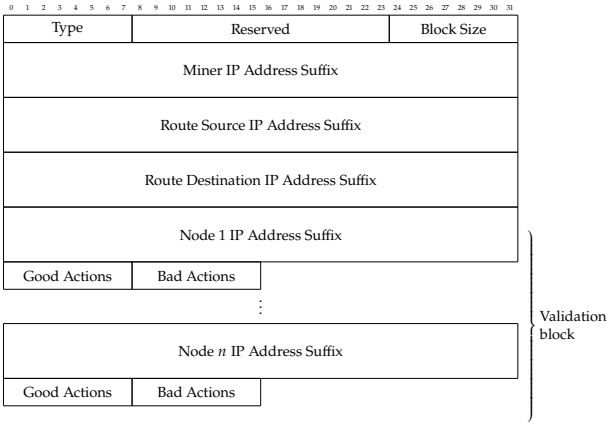


Figure 5.7: The structure of a *Route Validation* packet used to allow routing validation and Miner action consensus.

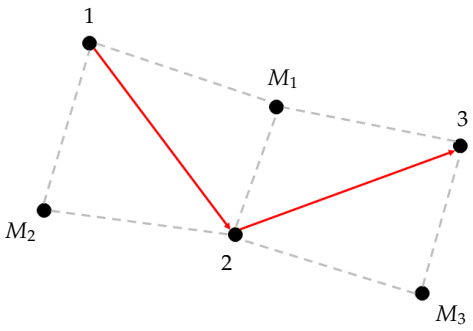


Figure 5.8: Six-Node network with a three node route and three Miners with the respective communications ranges depicted by the dotted grey lines.

To illustrate this process, we will use the network presented in Figure 5.8 with Table 5.3 presenting the network's *Consensus Matrix*. As we can see, this network possesses a three-node route with three Miners covering the entire route, labelled M_1 to M_3. Each miner transmits its block containing the list of observed activities up to 2-hops, reaching in this case the other two miners. Each miner constructs a *Consensus Table* corresponding to its 2-hop view of the route, listing the contents of all received blocks, associated with the Miner to which they belong.

The matrix presented in Table 5.3 contains the consensus tables of all three Miners for illustration, however, in practice the Miners would only

	M_1	M_2	M_3
M_1	1, 2, 3	1, 2, 3	1, 2, 3
M_2	1, 2	1, 2	1, 2
M_3	2, 3	2, 3	2, 3
Validated	**1, 2, 3**	**1, 2, 3**	**1, 2, 3**

Table 5.3: Consensus algorithm on mined nodes to confirm and validate all actions perceived by the neighbouring 2-hop Miners

be aware of their corresponding column. For each received block, the Miner adds the contained routing nodes and their actions to the table along with their own observations. Each unique node is analysed along with its actions in an attempt to reach consensus where the most frequent entries are considered the most valid. As we can see, all three Miners have reached a consensus on the activities of nodes 1, 2 and 3 as their corresponding lists of actions are present at least twice, the minimum necessary for consensus. This is an advantage compared to the previous approach, where in this case M_2 is capable of validating the activities of node 3, thanks to the data provided by both M_1 and M_3.

During this stage, each Miner also determines a preliminary opinion on the activities of the other Miners in range, based upon the actions they provided. For instance, if M_3 were to provide false or incorrect data on the activities of node 2, then both M_1 and M_2 would detect this discrepancy, identifying M_3 as malicious. Since the consensus mechanism validates the actions which are dominant in all received (i.e., most present), the correct actions of node 2 would be determined by the other Miners, since they both also validated that nodes activities.

However, there is the possibility where only a single Miner is in range of a specific node. In this case, this lone miner is considered the *ultimate authority* over this node, and its actions are considered valid by default. For this case to happen, the network topology must either be scarce, or the route runs along edge nodes, thus isolating a Miner from its neighbours. Although a rare occurrence, it remains possible dependant on the network at hand, however, this doesn't impact the underlying consensus mechanism. Appendix C.1 possesses a node in this situation, illustrating the notion of *ultimate authority*.

5.3.3 Miner Validation

Thanks to the new consensus approach for the results of the behavioural analysis, the Miners have not only reached consensus regarding the routing nodes, but have also identified Miners which performed malicious mining. However, alone this is insufficient as one Miner can identify all others as malicious in an attempt to disrupt network operations, such as using "Disinformation" attacks. Thus, another consensus stage is necessary, this time to determine and validate the list of Miner actions, illustrated in Figure 5.9.

The previous network stage used a 2-hop reach for the consensus data, allowing the Miners of a single section of route to converse with each other. However, to be able to correctly reach consensus regarding Miner activities, this hop range must be extended. Indeed, all Miners having received a block have performed their analysis on the block's owners' activities. As a result, it is necessary to double the hop limit for this Consensus, providing Miners who analysed the same Miner's actions with the opportunity to exchange and reach consensus. We also take

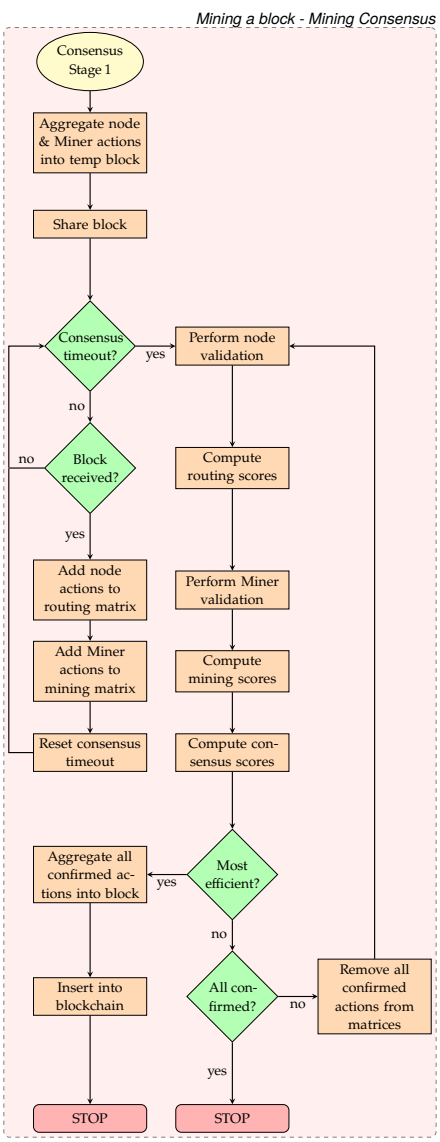


Figure 5.9: Miner Consensus Flowchart

advantage of this stage to share the list of confirmed node actions with other Miners, as it can be used to reduce the number of blocks inserted into the blockchain.

To distribute the combination of validated results from the routing phase, as well as the observed Miner actions, a new packet format was created, called *Miner Validation*. This packet, identified by type 4 and presented in Figure 5.10 contains the same base structure as the *Route Validation* packet with some slight variations. Firstly, as we can see, the packet's payload itself contains the new addition of Miner actions after the list of nodes. However, since the payload is now a mix of two types of nodes, another update to the *Block Size* field is needed. In this case, the *Reserved* field is split in two, with the second byte being replaced with the *Number of nodes*. Subsequently, the *Block Size* field is replaced with the number of Miners, allowing the size of both halves of the payload to be known. By representing them separately, it allows for accurate parsing, as well as the potential detection of a partially utilised block, where only nodes or only Miners are included. With this packet, all the necessary information is provided to the Miners to continue with their validation.

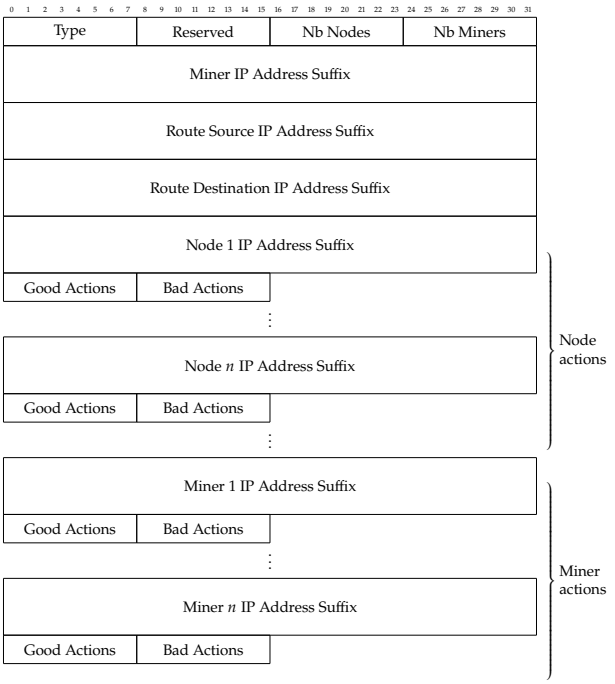


Figure 5.10: The structure of a *Miner Validation* packet used to allow routing validation and Miner action consensus.

This validation step, therefore, contains once again two parts. The first is the validation and concatenation of the received node actions as well as the computation of the corresponding routing score, which is used to identify the most efficient Miner for activity analysis. The second, is the same analysis performed in the previous validation phase, only this time on the activities of the miners. Once complete, the corresponding Miner score is computed, before both scores are added together to determine the overall validation score, allowing the identification of the most efficient Miner for inserting the block into the blockchain.

	M_1	M_2	M_3
M_1	1, 2, 3	1, 2, 3	1, 2, 3
M_2	1, 2, 3	1, 2, 3	1, 2, 3
M_3	1, 2, 3	1, 2, 3	1, 2, 3
Concatenated	**1, 2, 3**	**1, 2, 3**	**1, 2, 3**

Table 5.4: Consensus algorithm on confirmed nodes issued from the consensus stage to determine all validated actions perceived by all Miners in a 4-hop radius

Route Validation

The results of the routing validation phase as presented in Table 5.4. In this case, since all three Miners determined the same results, the differences aren't easily determined. However, for each received block the Miner once more adds the list of node actions to the table for the corresponding Miner. The main difference is that, instead of consensus, the values are simply concatenated to determine the list of all nodes which have been validated up to 5-hops away (4-hop miner validation plus the 1-hop for behavioural validation). Finally, each Miner computes the first of two efficiency scores, called the *Routing Score* for all Miners present in the validation matrix, including itself.

$$S_m^R = \frac{|R_{mined_m}|}{|\sum_{i=1}^{B} R_{received_i}|} \tag{5.2}$$

This score S_m^R is computed in Equation 5.2 as the ratio of nodes confirmed by the Miner m to that of all nodes from all received blocks combined. Here we specify R_{mined_m} as the list of validated nodes from the first consensus stage by this Miner and $\sum_{i=1}^{B} R_{received_i}$ the concatenation of all unique nodes from each received block $R_{received_i}$ with B the number of received blocks. The resulting routing scores are presented in Table 5.5 in fraction form for ease of calculation. Since all Miners were in range of each other, the overall routing score for each is the same in this example.

	M_1	M_2	M_3
M_1	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
M_2	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
M_3	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$

Table 5.5: Calculation of each Miner's *Routing Score* based upon the results of the routing validation stage, using Equation 5.2

Miner Consensus

The next phase is the Mining consensus, which bears a strong resemblance to the approach used for the Routing Consensus Matrix, computed in Table 5.3. Here, instead of the list of actions, the list of good and bad Miner actions are provided, allowing a consensus to be reached regarding the overall reliability of the Miners. This process is illustrated in Table 5.6, where we can see that, since all three Miners were in 2-hop range of each other, all three can validate the activities of the others. However, as can be seen each Miner cannot include itself in its own consensus calculation, thus avoiding a conflict of interest. Although this may seem like a conflict of interest, it does server a purpose in this scenario.

Indeed, by including oneself, it allows the other Miners to confirm the malicious intentions of said Miner, due to the difference between valid and invalid actions perceived. Furthermore, it is also possible that no

other Miners were in 2-hop range, thus meaning no one can confirm the Miner's actions. As a result, the lone Miner is considered the *ultimate authority* on its own actions, similar to previously with the routing actions.

Table 5.6: Consensus algorithm on all perceived Miner actions determined from the routing consensus stage to determine all confirmed and validated mining actions perceived by all Miners in a 4-hop radius

	M_1	M_2	M_3
M_1	M_2, M_3	M_2, M_3	M_2, M_3
M_2	M_1, M_3	M_1, M_3	M_1, M_3
M_3	M_1, M_2	M_1, M_2	M_1, M_2
Validated	M_1, M_2, M_3	M_1, M_2, M_3	$M_1, M_2, M_3,$

Once all Mining actions have been validated, the Miners can compute the second efficiency score, called the *Mining Score*.

$$S_m^M = \frac{|M_{mined_m} \cap M_{validated}|}{|M_{validated}|} \tag{5.3}$$

$$\begin{aligned} S_{M_m} &= \frac{|(M_1; M_2) \cap (M_1; M_3)|}{|(M_1; M_3)|} \\ &= \frac{|(M_1)|}{|(M_1; M_3)|} \\ &= \frac{1}{2} \end{aligned} \tag{5.4}$$

This score S_m^M is computed in a similar fashion to Equation 5.2, only this time concerning the Miner activities, as shown in Equation 5.3. However, whereas the routing score is determined from the results of two operations, routing consensus and routing validation, the mining score is computed from the single analysis of Miner consensus. The concept still remains the same, that of determining for each Miner the ratio of validated Miners in their block to the total number of validated Miners from all blocks resulting from the 4-hop consensus, $M_{validated}$. This means that if a Miner has mined miners M_1 and M_2, but the validation result contains M_1 and M_3, then the resulting score would be as shown in Equation 5.4 as there are only two validated Miners and only one Miner in common, M_1. This score can be calculated easily for all received blocks, granting the Miner a point of view as to the level of efficiency of all other Miners. The mining scores are presented in Table 5.7 in the same fashion as previously.

Table 5.7: Calculation of each Miner's *Mining Score* based upon the results of the mining consensus stage, using Equation 5.3

	M_1	M_2	M_3
M_1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
M_2	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
M_3	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$

Since the goal of this stage is the detection of the malicious Miners, the resulting actions are used to compute their reputation which is subsequently included in the analysis. Indeed, if a Miner has been determined as bad during the first consensus stage, its results will not be trusted in this phase, increasing the minimum number of similar values to reach consensus. Furthermore, its reputation is used in the computation of the overall score, thus allowing a very efficient malicious Miner to be excluded from block dissemination activities. The Reputation is computed following the same formulas used in the behavioural analysis in Chapter 3.

$$S^M_{good_m} = \sum_{i=1}^{W_m^M} good\ actions^M_{m_i} \tag{5.5}$$

$$S_{bad^M_m} = \sum_{i=1}^{W_m^M} bad\ actions^M_{m_i} \tag{5.6}$$

Firstly, we calculate $S^M_{good_m}$ and $S^M_{bad_m}$ the sum of good and bad miner actions for Miner m as shown in Equation 5.5 and Equation 5.6. Similarly, to the behavioural routing reputation, we define an action window time

frame W_m^M corresponding to the number of mining actions taken into account in the calculation. We have separated this from the routing window W_n^R so that they can be adapted dynamically dependant on the severity quarantine level, impacting both routing and mining separately. The mining reputation of Miner m, R_m^M is

$$R_m = \frac{1}{1 + e^{-\delta_m}} \tag{5.7}$$

$$\delta_m = \beta \times \frac{S_{good_m} - \gamma \times S_{bad_m}}{S_{good_m} + \gamma \times S_{bad_m}} \tag{5.8}$$

The reputation on the other hand follows the exact same formula as the behavioural reputation, with some slight modifications. Indeed, the Miner reputation R_m^M is still computed as a sigmoid function defined in $[0, 1]$ with the exponent δ_m representing the weighted value between $S_{good_m}^M$ and $S_{bad_m}^M$, defined itself in $[-1, 1]$. In comparing δ_m to δ_n in Equation 5.9, we see that the sensitivity factor β influencing the sigmoid function is still present. However, the weight of malicious actions α has been replaced with γ, once again separating the configuration of the routing reputation from the mining reputation. In this case, since no malicious actions were simulated, the resulting reputation for all three miners is 1.

$$\delta_n = \beta \times \frac{S_{good_n}^R - \alpha \times S_{bad_n}^R}{S_{good_n}^R + \alpha \times S_{bad_n}^R} \tag{5.9}$$

This equation is a reminder Equation 3.4 from Chapter 3

$$S_m^C = R_{m_t} \times (S_m^R + S_m^M) \tag{5.10}$$

The final consensus score S_m^C is computed using Equation 5.10 and shown in Table 5.8. By adding the routing score value S_m^R with the mining score value S_m^M , we can determine the best overall Miner based not only on the length of the neighbouring route, but also the density of the Miner distribution in its vicinity, allowing the blocks to be distributed in as fewer hops as possible. Furthermore, by multiplying this score with the node's reputation R_{m_t}, we allow the overall score to be impacted based upon the Miners reputation. As a result, a bad miner in an idealistic position will have less chances of being selected for block distribution, protecting even further the validation process.

Table 5.8: Calculation of each Miner's final *Consensus Score* based upon the *Routing* and *Mining Scores* calculated in Table 5.5 and Table 5.7

		M_1	M_2	M_3
M_1	**Routing**	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
	Reputation	1	1	1
	Score	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$
M_2	**Routing**	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
	Reputation	1	1	1
	Score	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$
M_3	**Routing**	$\frac{3}{3}$	$\frac{3}{3}$	$\frac{3}{3}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{3}$
	Reputation	1	1	1
	Score	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$	$\frac{5}{3} = 1.667$

In this case, the network remains very simple with all nodes in range of each other. All in all, no one Miner is better than the other resulting in an even score across the board. In such as case, a random back-off is used to allow one Miner to wake up first and insert its block into the blockchain, solidifying the behavioural analysis, but also the results of the Miner validation.

Block Dissemination

Another main difference with the previous methodology is the contents of the blocks inserted into the blockchain. Indeed, previously the blocks only contained the list of actions performed by the different routing nodes. Here, we also include the actions of the Miners directly in the same block, as shown in Equation 5.11. This can, however, evolve and adapt to new metrics, where it is also possible to define a separate blockchain, specific for the Miner's activities allowing to separate the routing analysis from that of the Miners. That being said, we chose to incorporate the contents into the same blockchain, thus reducing overall traffic and reducing the computation needed to extract and analyse from multiple sources.

$$M_1 \text{ or } M_2 \text{ or } M_3 = \begin{pmatrix} 1,2,3 \\ M_1, M_2, M_3 \end{pmatrix} \tag{5.11}$$

We use an updated format of the previously explained *Share* packet presented in Chapter 3.3.4, with the new format illustrated in Figure 5.11. As we can see here, the structure resembles a mix of the *Share* packet structure as well as the *Miner Validation* packet. Indeed, we find elements such as the *Type* field which contains the same value as previously 1, but also the *Previous Hash*, *Block ID* and *Timestamp* from the *Share* packet, as well as the separated list of node and Miner actions taken from the *Miner Validation* packet. However, there is one main difference with this new format. Whereas before the *Reserved* field took up two bytes and could, therefore, be split in two, this is not the case here. As a result, we decided on splitting the *Block Size* field itself, allowing half a Byte to represent the number of nodes and number of Miners. Although this is a reduction from the *Miner Validation* packet, this still allows for 15 nodes and Miners to be shared per block.

5.3.4 Secondary Consensus

Thanks to the two consensus validation phases, all routing activities can be both confirmed and included for transmission. However, this is not the case for all mining activities as only one consensus phase was performed, with no subsequent validation and concatenation of results. Indeed, in some cases, due to Miners being isolated from each other, it is possible that only one Miner is capable of confirming the activities of one or more others. As a result, during the final consensus phase these Miners would be excluded from the validated results, as only one Miner can vouch for them.

To resolve this deadlock, after the initial blocks have been inserted by the selected Miners from the consensus score, another Miner consensus phase is performed. Here, all confirmed Miners are omitted from the exchange,

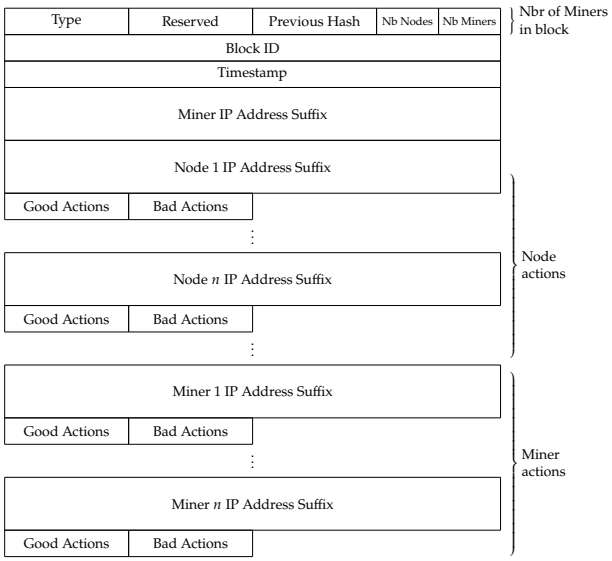


Figure 5.11: The updated structure of a *Miner Share* packet used to insert a block into the blockchain from the Quarantine metric, containing both nodes and Miners.

allowing only Miners with unconfirmed observations to share with their 4-hop neighbours. In this case, the nodes are exchanged using the same *Miner Consensus* packet used previously, only with an updated type value, 5, to indicate the secondary consensus. The rest of the consensus functions as normal, allowing the confirmation of Miner actions and the calculation of the resulting Miner and subsequent consensus scores. Finally, the selected Miners transmit their blocks with the missing values they can confirm, using a back-off timer if needed.

An example of this is presented in Appendix C.2, where the network used results in a separation between the Miners, thus needing a secondary consensus phase to resolve the missing values.

5.3.5 Broadcast Traffic Reduction

Due to the addition of a new analysis phase, extending the range to 4-hops on top of the original 2-hop range, there is a significant increase in traffic. This increase is a necessary sacrifice, as the Byzantine analysis requires a high level of exchanges between all involved parties to reach network wide consensus. However, it is possible to reduce the amount of traffic needed, all the while keeping the overall functionality the same. This can be achieved by allowing multicast data to be potentially embarked on existing packets, thus reducing the number at the cost of larger and longer transmissions.

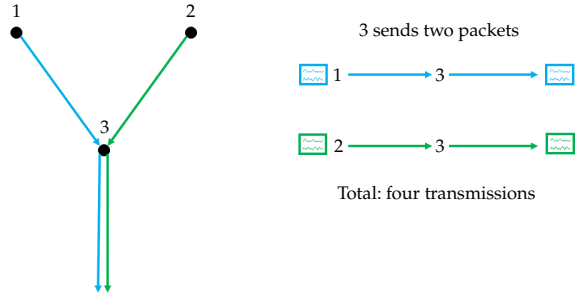


Figure 5.12: Illustration of how Multicast functions where two nodes send a multicast packet to the same IP address, which are relayed by another node.

To combat this, we propose a modification to the way nodes relay multicast packets. The standard operation of multicast is presented in Figure 5.12. We can see that nodes 1 and 2 send data to a multicast address which is received by node 3. In standard operation, node 3 will analyse the data, perform any actions it may be required to do and simply pass the data on-wards. As a result, we can see that we have a total of *four* transmissions for *two* packets since node 3 sends these multicast packets individually.

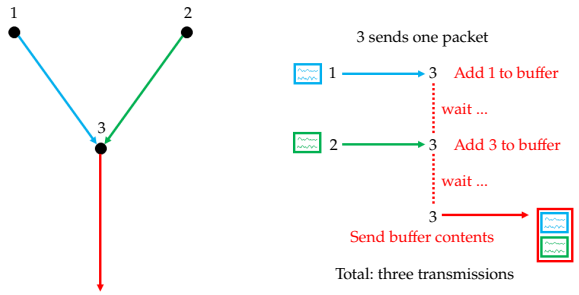


Figure 5.13: Illustration of how the *Multicast Relay* principal would function, where two nodes send a multicast packet to the same IP address, which are stored and relayed onwards in a single payload by another node.

Our proposition defines a new way of functioning with regards to forwarding multicast packets called *Multicast Relay*, illustrated in Figure 5.13. As we can see, nodes 1 and 2 send their multicast data as normal, only here node 3 changes how it reacts. Instead of forwarding the data blindly on-wards, node 3 instead adds it to a *multicast buffer* and activates a timeout function. When this timeout expires, the contents of the buffer a pushed to the network and on-wards towards their destination. However, if a new packet is received before then for the same multicast destination, it can also be added to the buffer, resetting the timeout. By using this method, we can continue to transmit the same data to the same destination, all the while reducing the number of transmissions in the meantime with a slight increase in message delay.

It is important to note, however, that this relay method will only function with packets going to the same multicast destination, since each destination IP would possess its own buffer entry. It would also be possible to allow the relay technology to be activated on demand, allowing certain multicast IPs to use it whilst others would function normally. As stated previously, we use multicast IPs to allow the broadcasting of blocks throughout the network for Miners to receive. Since these blocks aren't needed in real-time and can arrive with a small delay, it would be possible to use such a method to reduce the overhead.

Scenarios:
Medium size network: a network of 30-nodes, spaced out in an area of $150m \times 150m$.
Large size network: a larger and denser network of 100-nodes, contained in an area of $300m \times 300m$.

Assumptions:
- topologies 100% connected
- transmission rate of five packets sent every minute at two second intervals

Attacks:
Black-hole: routing based attack, where a malicious device drops all passing messages leaving no survivors.
Grey-hole: similar to Black-holes, only dropping messages dependant on a specific criteria (i.e., probability, type, length, source/destination, etc.).

5.4 Efficiency Evaluation

We perform a preliminary evaluation of this new module in the same conditions as the evaluation of *AODV-Miner* and *DSR-Miner* in Chapter 4.4. We illustrate the impact of this Quarantine module with the previous results in two new versions of both protocols: *AODV-Miner Quarantine* and *DSR-Miner Quarantine*.

Parameter	Setting
Area	$\{150m^2, 300m^2\}$
Number of nodes (N)	{30, 100}
Malicious Activity (P_{Ma})	{0% → 100%}
Malicious Weight (α & γ)	2
Link-cost field size (f_{size})	{8, 16}
Max Length (L_{max})	{64, 255}
Distribution	Random uniform
Transmission Range	50m
Window Size (W_n^R & W_m^M)	5
Reputation Decay (λ_n^R & λ_m^M)	{0.25, 0.25, 0.2, 0.15, 0.1}
Reputation Half-Life ($t_{\frac{1}{2}n}^R$ & $t_{\frac{1}{2}m}^M$)	{900, 900, 900, 1800, 1800}
Initial Reputation	0.5
Number of Simulations	100
Simulation Duration	15 min.
Messages per Transmission	5
Transmission Interval	1 min.
Message Interval	2 sec.

Table 5.9: Collection of parameters used during the simulations of the quarantine module using Cooja

5.4.1 Simulation Environment

Once again, we utilise Contiki-NG and Cooja to perform our simulations, analysing our Quarantine module against the same two scenarios as previously, with the list of parameters presented in Table 5.9. We base our analysis on the same topologies presented in Chapter 4.4.1, allowing a base point for comparison between our modules when confronted against the same Black-hole and Grey-hole attacks. One major change compared to previously, is the use of the adaptive reintegration method, where the half-life and decay factor of both the routing and mining reputation progresses dependant on the threat level of the corresponding node. In this preliminary analysis, we only utilise two of the four metrics previously defined:

1. **Packets Dropped**: $|Packets_{received}| - |Packets_{sent}|$
   This metric allows to evaluate the temporal impact of malicious nodes throughout the simulations, where a lower value represents a higher level of data integrity.
2. **Route Length**: nb_{hops}
   This metric grants the ability to identify the average number of hops taken during routing, this representing the impact and potential consequences of influencing the routing itself.

5.4.2 Simulation - Scenario I

We begin our evaluation once again with the 30-node networks, starting with our analysis of *AODV-Miner Quarantine*, before that of *DSR-Miner Quarantine*.

AODV-Miner Quarantine

We commence our analysis by comparing the results of *AODV-Miner Quarantine* with those of *AODV-Miner*. For continuity, we maintain a use of $C_{max} = 4$ as previously presented in Chapter 4.4.2. An overview of the parameters used in these simulations are shown inTable 5.10.

Table 5.10: Simulation parameters for the first scenario with *AODV-Miner Quarantine*

Parameter	Setting
Area	150m×150m
N	30
P_{Ma}	{0% → 100%}
α & γ	2
f_{size}	8 bits
L_{max}	64
Distribution	Random uniform
Tr Range	50m
W_n^R & W_m^M	5
λ_n^R & λ_m^M	{0.25, 0.25, 0.2, 0.15, 0.1}
$t_{\frac{1}{2}n}^R$ & $t_{\frac{1}{2}m}^M$	{900, 900, 900, 1800, 1800}
Initial R_n & R_m	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

Our first evaluation concerns routing efficiency against Black-hole attacks, allowing us to identify if our Quarantine module is capable of further increasing the efficiency of our Miner module in *AODV-Miner*. We commence once more with the evolution of the number of packets dropped, shown in Figure 5.14

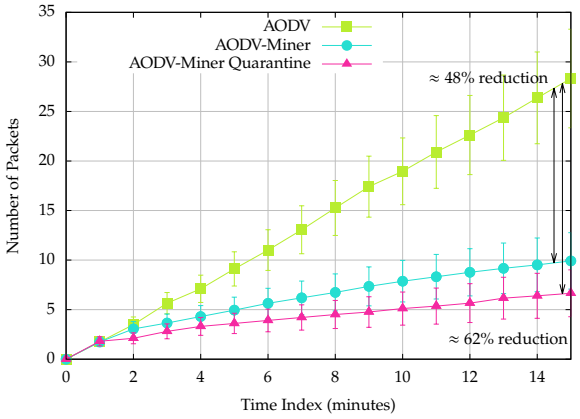


Figure 5.14: Number of packets dropped by *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 30 nodes with 10% of them expressing malicious tendencies

As we can see, from the first minute of the simulation, *AODV-Miner Quarantine* (**pink** triangles) remains below the number of drops observed with *AODV-Miner* (**light-blue** circles). At the end of the simulations, our Quarantine module rests at approximately three less packets dropped than our original Miner module, decreasing the overall drop rate by approximately 62% compared to the 48% seen previously. Furthermore, both *AODV-Miner* and *AODV-Miner Quarantine* follow a similar tendency, stabilising over time all the while minimising the number of drops. Thus, we can conclude that this initial version of the Quarantine module is capable of completely circumnavigating malicious nodes in the network, whereas before they were only avoided if possible.

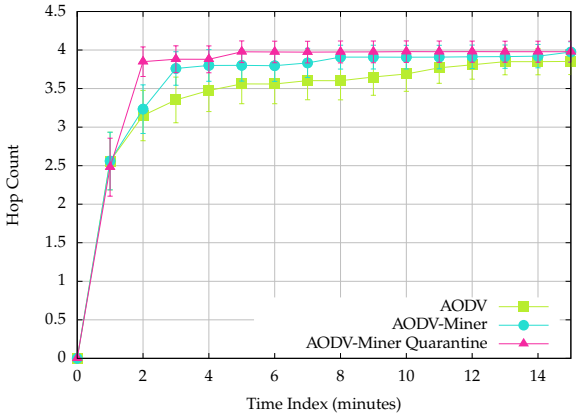


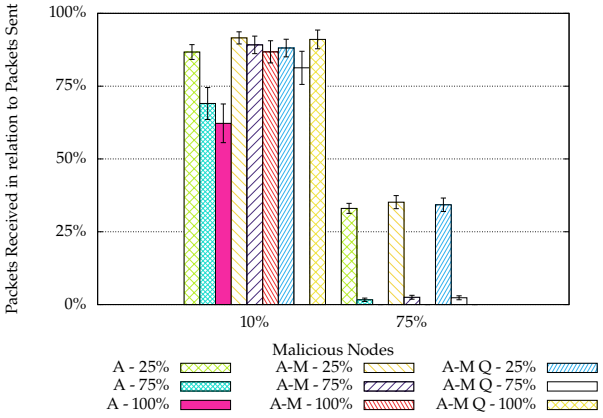
Figure 5.15: Average route length of *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 30 nodes with 10% malicious

As shown previously, one of the main trade-off's of our approach is the increase in overall route length. In Figure 5.15, we observe this increase and analyse the impact of our Quarantine module. As expected, the results of *AODV-Miner Quarantine* show an immediate impact to the overall route length, almost reaching the maximum values observed after the first two minutes. However, the maximum route length does not surpass that of *AODV-Miner*, simply increasing its rate of growth. Thus,

we can conclude that its impact is minimal, all the while increasing the efficiency and integrity of our system, and the network itself.

Threat Adaptation

Once again, we simulated the effect of Grey-holes on our modules. However, in this preliminary analysis we interest ourselves with the results for α and $\gamma = 2$, where 10% and 75% of the network nodes are malicious, each dropping 25%, 75% and 100% of data packets. As stated previously, this analysis allows us to determine how our new module reacts to different levels of malicious intentions in the network. The results are presented in Figure 5.19.



A → AODV
A-M → AODV-Miner
A-M Q → AODV-Miner Quarantine

Figure 5.16: Extract of throughput comparison between *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 30-nodes, subjected to varying probability Grey-hole attacks with α and $\gamma = 2$

As we can see, with 10% malicious nodes *AODV-Miner Quarantine* possesses a higher throughput level with 100% malicious activities than *AODV-Miner*, confirming the results from the previous analysis. However, this is not the case for lower values of impact, here 25% and 75% malicious activities. Indeed, *AODV-Miner Quarantine* struggles to reach the throughput level of *AODV-Miner*, all the while resulting with a higher level than the standard AODV implementation. This can be explained due to the introduction of threat severity, where the system may take longer to converge towards identifying the malicious entities. Furthermore, it is also possible that the current value of α and $\gamma = 2$ is not best adapted to this scenario, further increasing the convergence time.

The same can be said for 75% malicious entities. However, in this situation is important to note that when the majority of the network is compromised, there is only so much which can be done before the network crumbles. Here we can see that, even though *AODV-Miner Quarantine* has a slightly lower throughput that *AODV-Miner*, it still remains higher than that of AODV.

To further explore these results, it would be interesting to expand this analysis in line with that performed in Chapter 4.4.2, testing different values of α and γ. We can also increase the different percentages of malicious network takeover and malicious intentions to provide further points of comparison with the results of *AODV-Miner*. However, it is important to note that this new Quarantine module significantly expands the protection provided to the network. Indeed, whereas previously we only protected against malicious routing actions, here we also protect against malicious Miners compromising the integrity of both the consensus and the blockchain's contents. Thus, further analysis into this

increased protection would also allow to quantify the increased security provided to the network as a whole.

DSR-Miner Quarantine

Once again, we move on to the evaluation of our Quarantine module along side DSR. In a similar fashion with *AODV-Miner Quarantine*, we once again utilise $C_{max} = 257$, allowing a fair evaluation of *DSR-Miner Quarantine* along side its little brother. An overview of the parameters used in this evaluation are presented in Table 5.11.

Table 5.11: Simulation parameters for the first scenario with *DSR-Miner Quarantine*

Parameter	Setting
Area	150m×150m
N	30
P_{Ma}	{0% → 100%}
α & γ	2
f_{size}	8 bits
L_{max}	64
Distribution	Random uniform
Tr Range	50m
W_n^R & W_m^M	5
λ_n^R & λ_m^M	{0.25, 0.25, 0.2, 0.15, 0.1}
$t_{\frac{1}{2}n}^R$ & $t_{\frac{1}{2}m}^M$	{900, 900, 900, 1800, 1800}
Initial R_n & R_m	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

Unsurprisingly, we commence our evaluation by analysing the overall routing efficiency provided by *DSR-Miner Quarantine*, compared to *DSR-Miner*. First-off, we evaluate the number of packets dropped when confronted against Black-holes, illustrated in Figure 5.17.

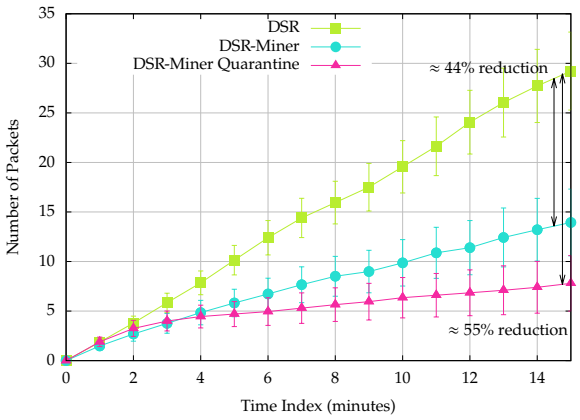


Figure 5.17: Number of packets dropped by both *DSR-Miner Quarantine*, *DSR-Miner* and DSR in a network of 30 nodes with 10% of them expressing malicious tendencies

We can see that once again, the addition of our Quarantine module has further increased the routing efficiency, decreasing the number of packets dropped by approximately 55% with *DSR-Miner Quarantine* (**pink** triangles) compared to 44% with *DSR-Miner* (**light-blue** circles). However, we can see that as of minute one, *DSR-Miner Quarantine* had more packet drops than *DSR-Miner*. This can be explained once more by the time needed for our module to converge towards the malicious entity before successfully placing it into quarantine. Indeed, although our Quarantine module gets off to a rocky start, it eventually finds its footing at around minute four, and outperforms our previous module.

By evaluating the route length trade-off, presented in Figure 5.18, we can see the convergence time mentioned previously. Indeed, from minute one to minute four, the overall route length is much lower with *DSR-Miner Quarantine* than *DSR-Miner*. We can summarise that when coupled with DSR, our module required additional time to be able to identify and isolate the malicious entities. However, after our Quarantine module has managed to converge upon the malicious nodes, the hop count increased to that of *DSR-Miner*, remaining slightly higher for the duration of the simulations. We can, therefore, conclude that the addition of our Quarantine module has increased network integrity, all be it with possessing a convergence period necessary to reach optimal efficiency.

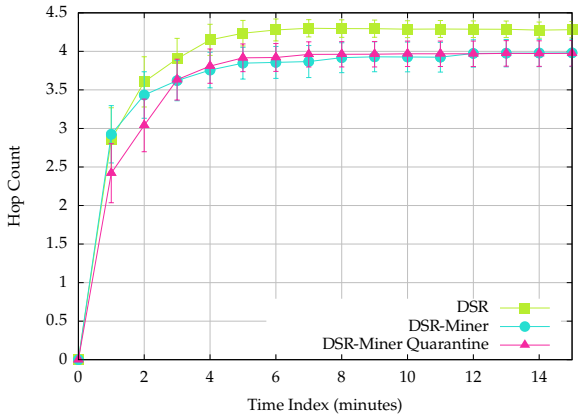
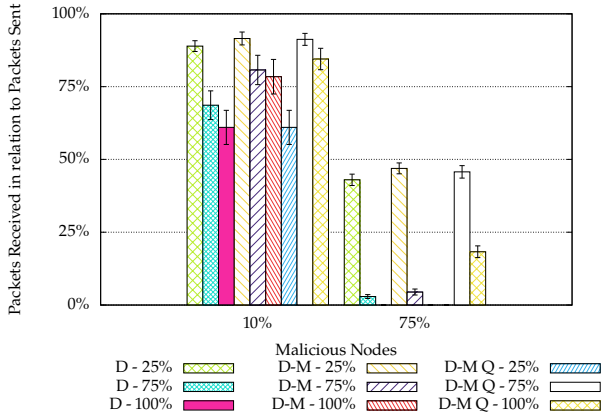


Figure 5.18: Average route length of *DSR-Miner Quarantine*, *DSR-Miner* and DSR in a network of 30 nodes with 10% malicious

Threat Adaptation

We finalise our analysis of *DSR-Miner Quarantine* by evaluating how it can adapt to when confronted with varying degrees of Grey-hole attacks. Once again, we note that this analysis is a preliminary evaluation of our module, thus we limit the simulations to α and $\gamma = 2$, with malicious nodes forming 10% and 75% of the network, with each dropping 25%, 75% and 100% of packets. Figure 5.19 presents the results of this analysis.



D → DSR
D-M → DSR-Miner
D-M Q → DSR-Miner Quarantine

Figure 5.19: Extract of throughput comparison between *DSR-Miner Quarantine*, *DSR-Miner* and DSR in a network of 30-nodes, subjected to varying probability Grey-hole attacks with $\alpha = 2$

As illustrated with *AODV-Miner Quarantine*, here we can see that with 100% of packets dropped with 10% malicious nodes, *DSR-Miner Quarantine* possesses a higher throughput than its non Quarantine counterpart. However, the main difference here is that not only has 75% malicious activities significantly increased in throughput levels, but it also reached a significantly higher level when 75% of the network has been compromised. Indeed, the results match that of 25% malicious activities, whereas here, the throughput level is through the floor with the Quarantine module. These results can be explained once more with the need for the system to converge towards accurately identifying malicious parties. As a result, it is a possibility that, whilst using DSR, our Quarantine module encounters significant difficulties in accurately identifying who to put into isolation.

To determine the cause of these results, a further in-depth analysis would be needed, also allowing us to evaluate the importance and impact of α and γ in the same was as previously. As stated previously, the increased security provided to the consensus mechanism is something which

mustn't be ignored, and possesses its own weight vis-a-vis the results on routing itself. Again, an interesting analysis would be to confront the consensus mechanism itself against different types of threats, such as malicious Miners themselves, to illustrate the importance of this new addition.

Discussion

With these preliminary simulation results, we have illustrated that our two implementations, *AODV-Miner Quarantine* and *DSR-Miner Quarantine* both possess an increase in network robustness when confronted with the same attacks as previously. However, it is important to note that the results do present some interesting questions. Indeed, the threat adaptation analysis brought to light some intricacies of the Quarantine module with different levels of attack. The results of *AODV-Miner Quarantine* demonstrated a small misalignment in terms of convergence time, where the Quarantine module, in some cases, couldn't accurately identify the malicious node, thus resulting in a lower throughput. These results were accentuated with *DSR-Miner Quarantine*, thus bringing attention to the need for a further in-depth analysis.

Firstly, it would be necessary to perform a full threat adaptation analysis on both protocol versions, utilising all the different values of both α and γ. With this study, we can accurately identify the convergence time needed to reach consensus with regards to the malicious node, but also extend our evaluation of both variables themselves. At the same time, an analysis of the interactions between the base protocol would be beneficial, as we have seen that AODV and DSR operate differently, adding extra variables to the equation. Furthermore, we could extend this threat analysis to include novel threat types, such as malicious Miners themselves, allowing us to reinforce one of the main advantages of this approach: the protection against a corrupted consensus. Finally, we could also evaluate how the consensus and convergence mechanism adapts towards a moving threat, such as a Worm which spreads out from a predefined patient zero and attempts to infect the whole network. Through device isolation, it would be possible to quarantine a portion of the network itself, thus stopping the spread in its tracks.

That being said, it is safe to say that, although further work is needed towards this Quarantine module, our preliminary results have shown an increase in security for both AODV and DSR.

5.4.3 Simulation - Scenario II

In the previous section, we analysed our Quarantine module in small scale scenarios possessing only 30-nodes. As performed in the previous chapter, we extend this analysis to much larger and denser networks of 100-nodes each.

As stated in Chapter 4.4.3, due to the limited capabilities of DSR with regards to large scale IPv6 networks, this analysis only concerns AODV.

DSR uses *Source Routing*, embarking the list of hops in an IP header which is not fragmentable. Since our simulations use a 6LoWPAN netstack with an IEEE 802.15.4 radio layer, the maximum MTU size is limited (102 bytes). As a result and even with the use of IPv6 compression, the max route length is also severely limited to ≈ 7 hops. As illustrated, this is insufficient for the topologies utilised in this analysis and thus, DSR must be omitted.

AODV-Miner

We follow on the previous analysis utilising the same overall analysis, adapting the configuration for the larger network size. Table **??** presents the parameters used in this analysis.

Table 5.12: Simulation parameters for the second scenario with *AODV-Miner Quarantine*

Parameter	Setting
Area	300m×300m
N	100
P_{Ma}	{0% → 100%}
α & γ	2
f_{size}	8 bits
L_{max}	64
Distribution	Random uniform
Tr Range	50m
W_n^R & W_m^M	5
λ_n^R & λ_m^M	{0.25, 0.25, 0.2, 0.15, 0.1}
$t_{\frac{1}{2}n}^R$ & $t_{\frac{1}{2}m}^M$	{900, 900, 900, 1800, 1800}
Initial R_n & R_m	0.5
Nb Sims	100
Duration	15 min.
Msg / Tr	5
Tr Interval	1 min.
Msg Interval	2 sec.

Routing Efficiency

We commence our evaluation with how the Quarantine module adapts to a much larger, denser network with regards to routing efficiency. We again utilise Black-holes as attackers and illustrate their impact on the number of dropped packets with *AODV-Miner Quarantine*, compared to *AODV-Miner*. Figure 5.20 shows the results of this analysis.

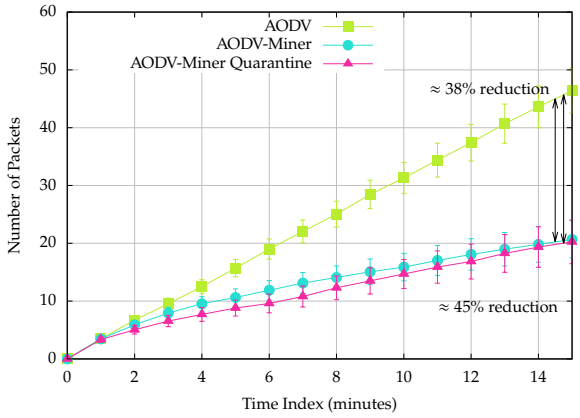


Figure 5.20: Number of packets dropped by both *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 100 nodes with 10% of them expressing malicious tendencies

We can see that *AODV-Miner Quarantine* (**pink** triangles) possesses a lower number of drops than *AODV-Miner* (**light-blue** circles) with approximately 45% in reduction compared to the previous 38%. However, whereas in the previous scenario the Quarantine results were easily distinguishable from those without, here we can see that the two start to blend towards the end of the scenario. That being said, our module is still capable of increasing the robustness of the network, all the while providing important Miner protection.

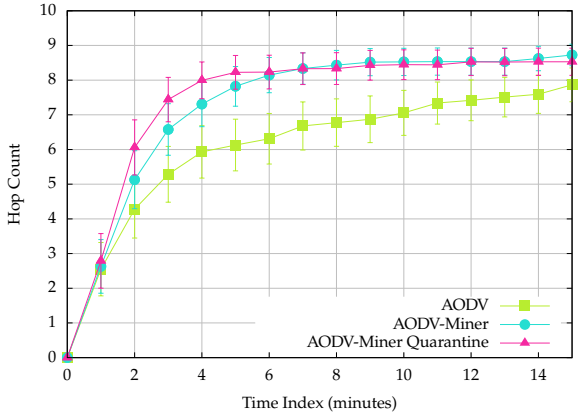


Figure 5.21: Average route length of *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 100 nodes with 10% malicious

To emphasise these results, we can take a look once again at the average route length, presented in Figure 5.21. It comes at no surprise that *AODV-Miner Quarantine* once more reaches the maximum route length at a faster rate. However, it can be seen that towards the end of the simulations, the route length remains generally constant, falling slightly below that of

AODV-Miner. Further analysis on a longer scenario would bring more points for comparison, and see if our Quarantine module remains more robust than its predecessor.

Threat Adaptation

We conclude the analysis of *AODV-Miner Quarantine* by analysing the adaptability of the Quarantine module in these larger 100-node networks. As stated previously, this analysis provides a preliminary overview of the modules capabilities. As a result, we only interest ourselves with the values of α and $\gamma = 2$, as well as 10% and 75% of the network performing the Grey-hole attacks with 25%, 75% and 100% of malicious activities. These results are presented in Figure 5.22.

A	→	AODV
A-M	→	AODV-Miner
A-M Q	→	AODV-Miner Quarantine

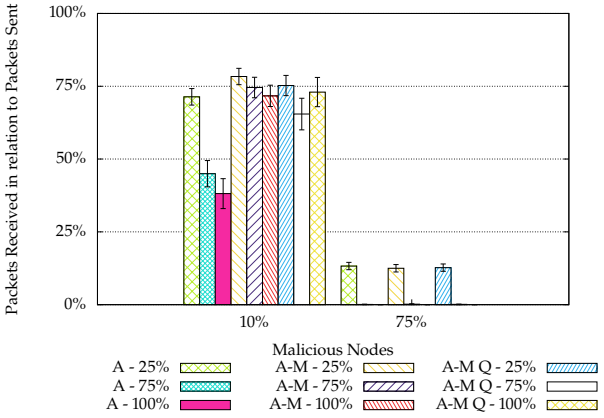


Figure 5.22: Extract of throughput comparison between *AODV-Miner Quarantine*, *AODV-Miner* and AODV in a network of 100-nodes, subjected to varying probability Grey-hole attacks with $\alpha = 2$

We can see once again that not all results from *AODV-Miner Quarantine* are higher than those of *AODV-Miner*. Indeed, only the results for 100% malicious activities with 10% malicious nodes show improvement over the original Miner module. As seen with the previous results, the number of packets dropped with *AODV-Miner Quarantine* is a lot closer to *AODV-Miner*. We can, therefore, hypothesise that our Quarantine module needs longer with its current configuration to accurately determine the malicious parties, and act accordingly. Unfortunately, the results from a 75% compromised network remain the same, where there is simply no free route available towards the destination.

To confirm our hypothesis, which is also the case for the previous scenario, an in-depth analysis of the impact of both α and γ should be performed. Furthermore, as stated previously, a longer simulation time would also allow to visualise the convergence time needed. However, it is to be noted that although the results for 25% and 75% malicious activities with *AODV-Miner Quarantine* fall below those of *AODV-Miner*, they still remain above those of AODV without either module. This emphasises that our system does indeed have a positive impact on the security of the network, with the added bonus of protecting against malicious Miners.

Discussion

Through the evaluation of our Quarantine module with a much larger and denser networks of 100-nodes, we were able demonstrate that *AODV-Miner Quarantine* still possesses the edge and further reinforces network security. However, this increase is much more limited that that observed

in the first scenario. This can be explained firstly with the much larger number of nodes available in the network, meaning that in some cases, the most optimal route was already in use. As a result, it is possible that only a small number of networks benefited from node isolation.

By extending our analysis to different threat types, different malicious weight values as well as different simulation durations, we would be able to gain a more in-depth view of the workings of this module. Moreover, as stated previously, it would be important to overview the interactions with both AODV as well as DSR from the previous scenario. In doing so, the interactions with the underlying protocol could be effectively strengthened, thus providing even better results in the future.

This being said, we can still confirm that our Quarantine module still increases the overall network efficiency, providing the necessary tools to AODV to not only avoid threats, but isolate them from networking activities.

5.5 Conclusion

In this chapter, we presented an extension of the consensus-based reputation module in Chapter 3, allowing to adaptively quarantine network threats. We discussed the intricacies of network quarantine, in particular the four important aspects with our proposition: Criteria, Threat Severity, Threat Isolation and Node Reintegration. Thanks to these four elements, it is possible to provide all nodes in the network with the proper tools to assure network operations. We also provide a redefinition of the consensus module, based upon BFT and the *Byzantine Generals Problem*. Through this new methodology, we provide a more robust platform for the validation of network activities, reducing the risks of malicious Miners corrupting proceedings. Furthermore, we also grant the ability to collectively determine the validity and trustworthiness of each network Miner, thus allowing to protect against bad Miners from causing mayhem and disrupting network operations. The combat the increased network traffic flow brought on by the Byzantine consensus, we propose a replacement multicast core called *Multicast Relay*, allowing intermediate nodes to forward multiple multicast payloads in a single packet. Finally, we performed a preliminary analysis through multiple simulations comparing the results with those achieved previously for both *AODV-Miner Quarantine* and *DSR-Miner Quarantine*. We saw an increase in efficiency against Black-holes by $\approx$ 62% and $\approx$ 55% with both protocols in networks of 30-nodes, and $\approx$ 45% with *AODV-Miner Quarantine*, higher than previously observed with *AODV-Miner* and *DSR-Miner*. Furthermore, our preliminary analysis into threat adaptation with Grey-hole attacks, showed our module needs time to converge towards the malicious entity itself. Further analysis of this convergence through extended simulation time, variable configuration and threats would provide the answers needed to fully understand and exploit the notion of Quarantine in wireless multi-hop networks.

Conclusion and Perspectives 6

The security of Critical Infrastructures (CIs) is paramount due to the importance of both their infrastructure and their objectives. Understanding the threats posed against these infrastructures grants security personnel the ability to prepare for a potential aggression, thus preparing their defences. With the deployment of the CyberSANE cyber security platform for CI protection, these security professionals are provided with an extensive library of functionalities and tools for protecting against threats, alerting when they occur and learning from past mishaps.

With the incorporation of small Internet-of-Things (IoT) devices, their characteristics and vulnerabilities must be taken into account. These devices are on occasion deployed in networks where direct communications aren't possible, forcing them to utilise the multi-hop network paradigm. However, entrusting data to unknown third parties is a risk, both for data privacy and network integrity. It is within these two objectives that the work performed in this thesis is situated.

In this chapter, we provide an overview of the contributions, relative to the CyberSANE project and towards this thesis itself. We then analyse these contributions, looking towards the future to determine the next possible steps short term, as well as significant advancements which could be achieved long term. Finally, we conclude this thesis with some closing remarks

6.1 General Conclusion

The work achieved in this thesis revolves around securing the routing process in wireless multi-hop ad hoc networks. This word was undertaken as part of the CyberSANE project, contributing towards the LiveNet component. Furthermore, some research was directed towards the construction of the components themselves, once more concerning LiveNet but also towards HybridNet. Here we commence by providing an overview of this research provided to the CyberSANE components, before we review the research activities pertaining to routing security.

6.1.1 CyberSANE Collaboration

As stated, the work towards the CyberSANE project concerned various research-based contributions, regarding current methods and systems. As a result, this work was performed at the start of the project, and intertwined with the state of the art performed fur the thesis work itself. Indeed, this work contributed towards the elaboration of both the CyberSANE evaluation process, as well as the LiveNet and HybridNet detection capabilities. Furthermore, during the projects lifetime, we performed the various dissemination activities, from the elaboration and upkeep of the project's website, but also the various social media accounts and campaigns.

The main technical contributions were towards literature study and idea development in two areas:

1. **Analysis of Cyber Incident Handling Trends**
   We performed an in-depth analysis into the different methods for handing incidents across IT systems, in particular those related to CIs. Through this study, we performed an overview of different threat organisational techniques, leading to different categorisation methods both used and exploited in the research domain, as well as by security experts. Furthermore, we analysed the different methods used in threat detection with Intrusion Detection Systems (IDSs), providing a detailed overview of existing metrics, methods and algorithms. Since these detection systems rely on input, it is possible to utilise existing security related datasets to help in training and testing, of which we performed an analysis of existing datasets, along with their contents. Finally, we presented how these elements would integrate with CI security, in particular towards use in the CyberSANE platform.
2. **CII Threat Taxonomy**
   To achieve its objectives of real-time threat detection, the LiveNet component of the CyberSANE system relies on input from external sources, called threat models, to identify different attacks. To create these models, an overview of the threat landscape was performed, presenting and analysing the different taxonomies available, along with their uses and potential weaknesses. Since these taxonomies are generally related to a specific system, we proposed a novel extended taxonomy, allowing coverage of multiple types of CIs, giving LiveNet the upper hand in both adaptation and comprehensiveness. The proposed taxonomy contains a total of 248 individual threats, organised into 22 threat categories, allowing for future expansion to include new CIs and was recognised by the *European Union's Innovation Radar* [209].

[209]: EU Innovation Radar. *Integrated incident handling and response for critical infrastructure*. Dec. 1, 2020. URL: https://www.innoradar.eu/innovation/40427 (visited on Oct. 17, 2022)

6.1.2 Thesis Proposition

From the initial analysis performed within CyberSANE we were able to advance with our main objective, concerning a secure method for allowing wireless multi-hop networks. Here, three main objectives, derived from CyberSANE's own, were defined to secure the exchange of data amongst IoT devices, which are summarised below:

1. **Threat Detection**: providing a secure and robust method for detecting threats in real-time based upon network activities.
2. **Origin Identification**: following the detection of the perceived threat, provide the capabilities for tracing the attack back to the source device in the network.
3. **Infection Quarantine**: reduce the malicious devices capabilities of partaking in networking activities in the IoT network itself, all the way to complete isolate from all network traffic.

From these three points, we have devised a consensus-based reputational module which can be associated with different multi-hop routing protocols to reinforce the base protocols abilities to exchange data in a

secure manner. Furthermore, by leveraging blockchain style dissemination capabilities, the resulting detection and identification information, represented by a devices reputation, can be shared amongst the network, allowing all devices to be aware of any existing threat. In total, the contributions of this thesis can be summarised as follows:

- We proposed a review of existing node-based reputation metrics, allowing accurate depiction of a devices trustworthiness based on their physical actions towards packet forwarding. We incorporate these metrics with consensus-based behavioural observation, allowing all network devices to work together to determine the trustworthiness of each other in the network. Through simulations, we have shown how this "*Miner module*"'s reputation metric functions with regards to the level of malicious intent of an infected node, allowing the network to gather an accurate overview of the node at hand. Furthermore, by allowing the behavioural history to be expanded or reduced, it is possible to adapt the approach to the device's characteristics, all the while maximising the precision of the consensus-module. Finally, with the incorporation of the reputational results with blockchain technology through a redefinition of the Proof-of-Work (PoW) concept, we can share the results with all neighbouring nodes, allowing all devices to possess an accurate overview of network trustworthiness.
- We proposed an adaptation to the Ad hoc On-Demand Distance Vector (AODV) routing protocol, allowing the reputation-based consensus module to integrate seamlessly with the existing routing functionalities, all the while influencing the path selection protocol to select the most trustworthy route available. We also proposed updates to the AODV packet structure, allowing the behavioural analysis module to gain knowledge of AODV's routing decisions, allowing it to accurately identify the routing behaviour. Simulations of the resulting combination between AODV and our "*Miner module*", called *AODV-Miner* showed an increase in network throughput across the board, demonstrating the *AODV-Miner* is capable of detecting malicious threats and their source before avoiding them as much as possible. We also demonstrated the versatility of our module against different threats with varying levels of malicious intentions and impact on the network itself, illustrating how the behavioural module is capable of keeping up with attackers.
- We extended the routing protocol interface, revamping its contact points for it to integrate with other reactive routing protocols without serious modification. Through an analysis in conjunction with Dynamic Source Routing (DSR), we demonstrated once more that the Miner module in *DSR-Miner* can not only adapt to the network and threats thrown at it, but also the protocol's only eccentricities and limitations. We also analysed a possible extension into the realm of proactive protocols, analysing the possible interface with Routing Protocol for Low-Power and Lossy Networks (RPL), thus opening the door towards securing other types of protocols.
- We reworked the isolation method component of the consensus-module to allow physical and logical quarantine of malicious nodes, effectively isolating and stopping them from partaking in any network related activities. Through the definition of severity metrics and the addition of both detection and identification capabilities

for malicious devices during the behavioural analysis phase, we can provide more information to the network, boosting overall trustworthiness. Furthermore, by extending the severity indicators to correspond with different levels of quarantine, malicious devices can receive punishments equal in ratio to the impact of their malicious actions, from a slap on the wrist with reduced participation to complete and utter isolation, stopping them from performing any network action. We also revamped the consensus method to reinforce the consensus itself for the results of the behavioural analysis, but also for the incorporation of the aforementioned security upgrade for the behavioural analysis phase itself. We finally performed a preliminary analysis into the capabilities of this Quarantine module alongside both AODV and DSR previously studdied. The proposed protocols *AODV-Miner Quarantine* and *DSR-Miner Quarantine* showed an increase in network security, all the while providing much needed robustness to the consensus mechanism.

6.2 Perspectives

As shown, our reputational-based consensus module can be integrated into reactive routing protocols, providing the necessary tools to influence the normal protocols path selection algorithms to select the most trustworthy path, all the while non compromising the protocol's core functionalities. That being said, there are still areas which could benefit from further analysis, increasing the overall efficiency with regards to IoT devices and networks, as well as other protocols and systems.

6.2.1 Short Term

There are multiple short term objectives which would be beneficial towards the evaluation of our module. Of course, the finalisation of the integration towards *RPL-Miner*, as well as a potential expansion towards other proactive protocols, such as OLSR, goes without saying. The same is also for transitioning from simulations to experimentation, which can be achieved through the FIT IoT-LAB platform [210] which natively supports Contiki-NG. Furthermore, some of these protocols are capable of supporting mobile devices, meaning evaluating our system in this context would also be beneficial.

[210]: Future Internet Testing Facility. *FIT IoT-LAB - The Very Large Scale IoT Testbed*. July 3, 2020. URL: https://www.iot-lab.info/ (visited on Sept. 25, 2022)

Although the quarantine expansion adds a much needed increase in security with regards to Miner tampering, however, it also comes at a theoretical cost, that of increased overhead from the consensus mechanism. Although this increase is normal when adding new elements and methods to existing algorithms, this overhead shouldn't be a severe handicap. This is the case with the quarantine expansion, which increased the already high overhead from the first consensus-metrics. Thus, an analysis of further methods to reduce this overhead without impacting the consensus efficiency is important to increase the lifetime of the IoT devices in the network.

6.2.2 Long Term

As previously presented, reactive routing protocols are adapted towards mobile environments, allowing routes to be discovered even when devices are moving around. During our simulations we relied on static topologies to validate our module and protocol integration. By extending these simulations to include dynamic topologies, we can simulate networks such as Mobile Ad Hoc Networks (MANETs), thus enhancing our expansion into other network types. However, the Cooja simulator no longer natively supports mobile topologies, meaning that a conversion of the implementation towards a mobile simulator might be necessary. Furthermore, this could also be included in the experimentation proposition, however, the choice of platform also incurs certain limitations. Indeed, if the experimentation is performed on a dedicated test-bed, mobile topologies would be simple to emulate on a small scale network.

Our contributions revolve around multi-hop IoT networks where devices are generally left to themselves, thus making autonomous decisions is necessary. However, it would be possible to adapt this approach to other types of networks, in particular where multiple large scale routers are deployed. Although these networks are generally administered, it is possible for a malicious party to compromise a routing table, thus causing the infected router to redirect its traffic. With the incorporation of our reputational module, other routers can identify a traitor in their midst, updating their routing tables to avoid the culprit. That being said, an overhaul of the behavioural scouting methods would need to be done, but this would provide the advantage of a more robust blockchain system for value distribution.

One final proposition is that since our module inserts observations into a blockchain allowing secure and distributed sharing of results, it is possible to utilise this to our advantage. For instance, if the blockchain was stored on a server under the control and protection of CyberSANE, it would be possible to integrate our Miner module into both the LiveNet and HybridNet components. Indeed, with this integration, LiveNet would receive alerts through block analysis to identify malicious actions undertaken in the network. It would also be possible to provide other information, such as network integrity and module efficiency, all computed from the blockchain. With this information, HybridNet would be able to propose solutions that the security expert can utilise, such as increasing the impact of network quarantine or even manually selecting a node to be isolated, all conveyed to the different nodes through the blockchain, thus authenticating the command's origin. This would also provide information such as new nodes appearing in the network from behavioural analysis, as well as detect unauthorised node movement, based upon the Miner which observed its activities. Finally, thanks to the network integrity overview, the security professional would be able to gain oversight as to the percentage of corruption in the network, thus allowing to completely shutdown and isolate the network from the main infrastructure, if this value falls too low.

6.3 Closing Remarks

Critical Infrastructure security is an important element with the advancements related in both IT and networking. Solutions like CyberSANE provide a much needed hand to security professionals whose work is becoming more and more difficult as threats evolve. With attackers becoming more and more cunning and available tools and equipment being easier to find, threats are themselves becoming more and more advanced as well as portable, meaning an attacker can implement a sophisticated self contained autonomous attack on a small device such as a Raspberry Pi and take it directly to the target location. It is, therefore, important for both detection and mitigation systems, as well as security professionals to constantly evolve, providing new solutions to counter the advancements made by malicious parties.

However, due to their limited resources, IoT devices aren't always granted the same courtesy with regards to security implementations. In many cases, these devices are the weak link in complex infrastructures, allowing attackers to slip in through a cracked window at the rear of the property, whilst the front door is deadlocked and under constant surveillance. Thus, proposing solutions to these devices is important and of significant interest in the scientific community. Their limited capabilities as well as autonomous nature render many solutions difficult to implement, not only with regards to functionality but also wear and tear on the devices themselves, effectively reducing their lifetime.

By proposing unified solutions, allowing for IoT protection as well as for CIs, it is possible to reduce the impact of attackers, thus returning the control of our small devices to us. Indeed, CyberSANE proposes open Application Programming Interfaces (APIs), allowing new solutions such as our consensus-module to be integrated directly into different components, providing the necessary information for the platform as well as the security user. Thanks to this solution, CI security is entering into a new era, reinforcing existing methods and providing new tools to make sure all equipment, be them IT or IoT, is secure against the ever evolving cyber menace. That being said, further study is necessary to clear up some grey areas, as well as to determine the best possible compromise between enforcing security at a minimal cost.

Appendix

A

Threat Type Overview

In this appendix, we present an overview of the different threat types presented in Chapter 2.1.2. We present each one individually, organised by categorisation type conform to the previous presentation.

A.1 Attack Type

- **Denial-of-Service (DoS)**: Attack aiming to deny legitimate access to a shared resource such as a web server or wireless access point [211].
- **Probing**: Reconnaissance method to obtain information from a target system, [62].
- **Remote-to-Local (R2L)**: Gain unauthorised remote access to an IT system using obtained user credentials.
- **User-to-Root (U2R)**: Illegaly access an administrative role on a target system, from which high-level attacks can be performed. **Man-in-the-Middle (MitM)**: The attacker places themself between two victim devices, forcing traffic to transit through them, allowing unlimited access to passing data to perform what analysis or attack they desire, as shown in Figure 2.3 [63].
- **Brute Force**: Iterative attack trying all possible keystrokes to find a match and break or decrypt login credentials, [212].
- **Replay**: Attack using legitimate captured data subsequently resent to the destination [213].
- **Deception**: Also called *Integrity attacks* or *False Data Injection (FDI)*, introduce false information into a machine, forcing it to perform invalid operations [214, 215].
- **Active Eavesdropping**: Active spying attack, forcing traffic to transit via them increasing their spying efficiency [216].
- **Scanning**: Recover various types of system or neteork related information through direct observation [217].
- **Code Injection**: Introduction of malicious code onto a device through unsecured inputs [218].
- **Physical**: Attacks resulting from direct physical access with the target device.
- **Network**: Attacks achieved through an active network connection or from a remote source on passing communications.
- **Software**: Any and all attacks made through malicious programs running on the victim device.
- **Encryption**: Threats which aim to break encryption systems to recover private keys.

[211]: A. D. Wood and J. A. Stankovic. 'Denial of service in sensor networks'. In: *Computer* 35 (2002). DOI: 10.1109/MC.2002.1039518

[62]: Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016

[63]: Mauro Conti, Nicola Dragoni, and Viktor Lesyk. 'A Survey of Man In The Middle Attacks'. In: *IEEE Communications Surveys & Tutorials* 18 (2016). DOI: 10.1109/COMST.2016.2548426

[212]: NIST CSRC. *Glossary - Brute Force Password Attack Definition*. Aug. 6, 2020. URL: https://csrc.nist.gov/glossary/term/brute_force_password_attack (visited on Sept. 22, 2022)

[213]: Paavan Rughoobur and Leckraj Nagowah. 'A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare'. In: *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*. 2017. DOI: 10.1109/ICTUS.2017.8286118

[214]: Qingyu Yang, Dou An, Rui Min, Wei Yu, Xinyu Yang, and Wei Zhao. 'On Optimal PMU Placement-Based Defense Against Data Integrity Attacks in Smart Grid'. In: *IEEE Transactions on Information Forensics and Security* 12 (2017). DOI: 10.1109/TIFS.2017.2686367

[215]: Yao Liu, Peng Ning, and Michael K. Reiter. 'False Data Injection Attacks against State Estimation in Electric Power Grids'. In: *ACM Transactions on Information and System Security (TISSEC)* 14 (2011). DOI: 10.1145/1952982.1952995

[216]: Yong Zeng and Rui Zhang. 'Active eavesdropping via spoofing relay attack'. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016. DOI: 10.1109/ICASSP.2016.7472059

[217]: Zhen Ling, Kaizheng Liu, Yiling Xu, Yier Jin, and Xinwen Fu. 'An End-to-End View of IoT Security and Privacy'. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017. DOI: 10.1109/GLOCOM.2017.8254011

[218]: Dimitris Mitropoulos and Diomidis Spinellis. 'Fatal injection: a survey of modern code injection attack countermeasures'. In: *PeerJ Computer Science* 3 (2017). DOI: 10.7717/peerj-cs.136

A.2 Objective Oriented

- **Access**: Attacks which attempt to gain unauthorised access to a device
- **Malicious**: Deliberate attempts to compromise a system, generally providing an advantage to the attacker
- **Non-Malicious**: Resulting from accidental damage or mishandling, causing various degrees of difficulties for system operations
- **Cyber Crime**: Small attacks where the goal was for the attackers personal gain
- **Cyber Espionage**: spying activities where information recovery is the primary objective
- **Cyber Terrorism**: attacks which cause significant damage and disruption to both people and property
- **Cyber War**: Sttacks between nations, where the nation itself is the attacking entity aiming to gain significant advantages over their victim
- **Disconnection and Goodput Reduction**: Attacks aiming to disconnect devices from the network, stopping them from communicating, or by severely impacting and reducing the operational efficiency, called *Goodput* [219]
- **Side-Channel Exploitation**: External methods to extract important information regarding internal device operations
- **Covert-Channel Exploitation**: Exploiting weaknesses in the device configuration where authorised information is shared between two cooperating entities, all the while breaching security policies [220]
- **Hardware**: Attacks aiming to significantly impact or exploit the various devices hardware characteristics
- **Human Factor**: Attacks made by the user, intentionally or not by not adhering to complicated security guidelines, or even attacks that target the users themselves
- **Interception**: Intercepting passing messages breaching confidentiality between the two communicating parties
- **Interruption**: Attacks aiming to stop or impact correct network activities
- **Fabrication**: Injects false packets from captured or generated data
- **Modification**: Changes the values of communications in real time
- **Access Control**: Any method used to break or take advantage of existing *Access Control* methods
- **Authentication**: Attacks aiming to break authentication algorithms
- **Availability**: Reduction of the capacity for users to use the various services
- **Confidentiality**: Attacks aiming to gain access to private communications to extract data
- **Integrity**: Attacks aiming to compromise legitimate data by forging authenticated messages

[219]: Néstor J. Hernández Marcano, Chres W. Sørensen, Juan A. Cabrera G., Simon Wunderlich, Daniel E. Lucani, and Frank H. P. Fitzek. 'On Goodput and Energy Measurements of Network Coding Schemes in the Raspberry Pi'. In: *Electronics* 5 (2016). DOI: 10.3390/electronics5040066

[220]: NIST CSRC. *Glossary - Covert Channel Definition*. Apr. 12, 2020. URL: https://csrc.nist.gov/glossary/term/covert_channel (visited on Sept. 22, 2022)

A.3 Use-Case

Cyber-Physical System

The first use case is relative to Cyber Physical System (CPS), which is defined by NIST as a device possessing interactions between both engineered components and various processes through the use of integrated physics and logic circuits [221]. Such devices can be employed in multiple areas and domains, mixing the physical world with the cyber verse. For example, these devices could be employed in healthcare in elderly peoples homes to assist in every day life, or be controlled and maintained in an intensive care unit.

[221]: NIST CSRC. *Glossary - cyber-physical system(s)*. Apr. 3, 2020. URL: https://csrc.nist.gov/glossary/term/cyber_physical_systems (visited on Sept. 23, 2022)

These devices are vulnerable to threats targetting various sensors or actuators, making their detection a priority [222]. An examples of such an attack is the Stuxnet worm [223] attack against the industrial control systems in a uranium enrichment plant in Iran [224].

[222]: Chaoqun Yang, Zhiguo Shi, Heng Zhang, Junfeng Wu, and Xiufang Shi. 'Multiple Attacks Detection in Cyber-Physical Systems Using Random Finite Set Theory'. In: *IEEE Transactions on Cybernetics* 50 (2020). DOI: 10.1109/TCYB.2019.2912939

[223]: Stamatis Karnouskos. 'Stuxnet worm impact on industrial cyber-physical system security'. In: *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*. 2011. DOI: 10.1109/IECON.2011.6120048

[224]: James P. Farwell and Rafal Rohozinski. 'Stuxnet and the Future of Cyber War'. In: *Survival* 53 (2011). DOI: 10.1080/00396338.2011.555586

Here we look at two subcategories of CPSs: *Networked Control System (NCS)* and *Smart Grids*.

- **Networked Control System (NCS)**: used in different CIs, these devices rely on shared networking systems to communicate between components, inheriting the networks weaknesses [225]. Attacks here can be separated into two categories, differentiating attacks on *physical components* from those on *network communications* [64]. This is, however, insufficient to categorise certain attacks. As explained in [64], compromised sensors can send false data, making the initial attack both *physical* and *network*-based. An example of this is the attack on the Maroochy Water Services in Queensland, Australia in 2000 where a wireless link to the wastewater pumps was exploited resulting in untreated waste being released [226].
- **Smart Grids**: these CIs use smart devices to measure energy consumption and adapt the distribution as needed. Many attacks exist on these infrastructures, such as *DoS* or *jamming* [227] as well as *MitM* or *False Data Injection (FDI)* [228], taking down three power suppliers causing the 2015 Ukraine Blackout. A method of categorisation for attacks against Smart Grids is proposed in [65], separating attacks into three distinct categories: *Power and Energy Layer*, *Computer/IT Layer* and *Communication Layer*. The *Power and Energy Layer* category, concerns attacks against both control stations and equipment, such as FDI aiming to corrupt control station operations. The second category, *Computer/IT Layer*, concerns attacks targetting the IT devices themselves running command software, allowing either attacks directly on the software itself, or through the use of malicious programs, such as malware. The third and final category evolves around the *Communication Layer*, dividing the attention between threats targetting the communication protocols themselves and generic network attacks, such as DoS and friends.

[225]: Xian-Ming Zhang, Qing-Long Han, Xiaohua Ge, Derui Ding, Lei Ding, Dong Yue, and Chen Peng. 'Networked control systems: a survey of trends and techniques'. In: *IEEE/CAA Journal of Automatica Sinica* 7 (2020). DOI: 10.1109/JAS.2019.1911651

[64]: Eman Mousavinejad, Fuwen Yang, Qing-Long Han, and Ljubo Vlacic. 'A Novel Cyber Attack Detection Method in Networked Control Systems'. In: *IEEE Transactions on Cybernetics* 48 (2018). DOI: 10.1109/TCYB.2018.2843358

[227]: Subham Sahoo, Sukumar Mishra, Jimmy Chih-Hsien Peng, and Tomislav Dragičević. 'A Stealth Cyber-Attack Detection Strategy for DC Microgrids'. In: *IEEE Transactions on Power Electronics* 34 (2019). DOI: 10.1109/TPEL.2018.2879886

[228]: Gaoqi Liang, Steven R. Weller, Junhua Zhao, Fengji Luo, and Zhao Yang Dong. 'The 2015 Ukraine Blackout: Implications for False Data Injection Attacks'. In: *IEEE Transactions on Power Systems* 32 (2017). DOI: 10.1109/TPWRS.2016.2631891

The Internet-of-Things

Since IoT networks are becoming more and more common, their security concerns are also increasing. Indeed, with the wide spread deployment in areas such as CIs, they are becoming a prime target for attackers due

to their inherent characteristics. To allow for IoT-based attacks to be categorised, two distict methods have been proposed:

- **Low-Level**, **Intermediate-Level**, **High-Level**
  This approach is utilised in [66], separating the threats based upon which layer on the OSI model they are found.
- **Perception**, **Network**, **Application**
  The same concept of layer separation can be used on the IoT stack itself, using the layers specific to these devices and applications, as utilised in [7]. [67] uses the same categorisation approach, all the while using different terminology for the categories based upon the Attack Vector itself: *Hardware*, *Communication Links* and *Interfaces/Services*

[66]: Minhaj Ahmad Khan and Khaled Salah. 'IoT security: Review, blockchain solutions, and open challenges'. In: *Future Generation Computer Systems* 82 (2018). DOI: 10.1016/j.future.2017.11.022

[67]: Mardiana binti Mohamad Noor and Wan Haslina Hassan. 'Current research on Internet of Things (IoT) security: A survey'. In: *Computer Networks* 148 (2019). DOI: 10.1016/j.comnet.2018.11.025

Here we present the different types utilised in IoT.

- **Low-Level**: Encompasses all threats against the lowest network layers, *Physical* and *Data-Link*, but also against the *hardware* itself
- **Intermediate-Level**: Attacks against *Network* and *Transport* layers related activities, such as routing or session management
- **High-Level**: Attacks revolving around the applications themselves which are running on the various network devices
- **Perception**: Also called the *Sensors* layer, concerns all operations on the sensor nodes themselves, from data collection to processing and transmission
- **Application**: Deals with the data itself, allowing the creation of the "smart" environment, all the while protecting the data's authenticity as well as integrity and confidentiality.

Software Defined Network

Although many security mechanisms exist such as firewalls or detection and prevention systems, they are deployed along the internet edge, protecting the enclosed network from external attacks. However, the borderless architecture in use by IoT devices bypasses such systems and raises many security concerns. One method to combat such risks is the introduction of Software Defined Network (SDN) to encompass and regulate routing decisions in the network itself [43]. However, like all other systems connected to the internet, SDNs are also susceptible to various types of attack:

- **Reconnaissance**: The attacker can observe and analyse various vulnerabilities in the SDN system, allowing them to possibly penetrate into the system.
- **Data Exfiltration Attack**: Once the attacker has gained access to the system, they can recover and extract compromising data as well as security credentials to the rest of the system.

When it comes to SDNs, the authors of [229] propose a categorisation method to differentiate attacks dependant on the SDN architecture layers [68] which are affected or targeted by the attack. This method is comprised of five distinct categories:

[229]: Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. 'Sdn Security: A Survey'. In: *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*. 2013. DOI: 10.1109/SDN4FNS.2013.6702553

[68]: Othmane Blial, Mouad Ben Mamoun, and Redouane Benaini. 'An Overview on SDN Architectures with Multiple Controllers'. In: *Journal of Computer Networks and Communications* 2016 (2016). DOI: 10.1155/2016/9396525

- **Application Layer**: The highest architectural layer containing the various network applications used for network monitoring and control. On this layer, attacks target the applications themselves, such

as unauthenticated application access, or resulting configuration errors, such as no policy or fake policy enforcement.

- **Application-Control Interface**: Encompasses a collection of open source APIs which in-turn en-globes all communications between the *Application Layer* and the *Control Layer* below
- **Control Layer**: Considered as the most important and intelligent section of an SDN architecture. Its goal is to forward the different rules from the *Application Layer* to the *Data Layer* through the many different controllers at its disposal
- **Control-Data Interface**: Ensures the connection between the *Control Layer* and the *Data Layer* through the use of various protocols. Once again, since this interface conveys data from the *Control Layer* to the *Data Layer*, it is susceptible to controller and data related attacks
- **Data Layer**: represents the entirety of network forwarding devices whose rules are retrieved from the *Control Layer* through the connected interface

B

Consensus Validation of Routing Activities

As explained in Chapter 3.3, the validation methodology is composed into two stages: Behavioural Validation and Block Validation, otherwise known as *Mining a route* and *Mining a block*. In this appendix, we turn our attention towards the second stage, in particular the consensus algorithm previously presented in Chapter 3.3.3. Here, we extend the previous presentation and analysis of the function of this algorithm by performing a theoretical analysis using two scenarios. Each scenario possesses a different network size and topology as follows:

1. **Scenario** A - 10-node interconnected network with a six-node route and four Miners
2. **Scenario** B - 13-node network with an eight-node route and five Miners, separated down the centre line with three Miners on one side and two on the other with each subset out of block exchange range with the other

Through this analysis, we can illustrate in detail how this consensus mechanism works when pitched against larger networks and varying topologies. As stated previously, the objective of this validation process is to confirm the actions of the Miner (i.e., the mined actions are correct), as well as determine the most efficient block for network dissemination. In this case, we suppose that all Miners are not malicious, and that all actions are correct. This allows us to concentrate on the efficiency computation, determining the blocks which are disseminated, as well as the exact sequence of events leading up to this result.

B.1 Scenario A

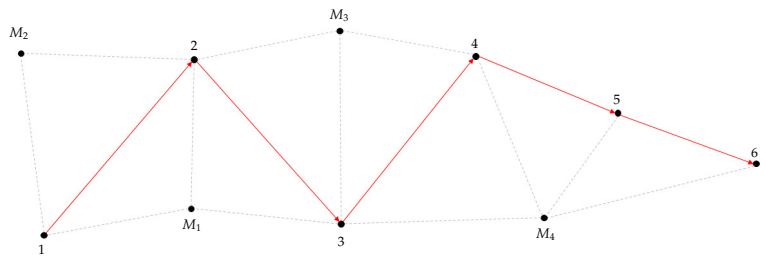


Figure B.1: 10-Node network topology with a six-node route and four Miners

One of the problems with this approach is the numerous possible outcomes based on which node woke up first. Indeed, in some cases, no extra blocks are needed as the most efficient node transmits their block first, thus reducing the overhead. But, in others, multiple blocks need

exchanging, increasing the transmissions and making the validation process prediction more complicated. Here we perform this analysis on the network illustrated in Figure B.1, varying which node wakes up first and the results of the different analytical stages. This is a theoretical topology, devised for the sole purpose of being used to theoretically test and validate the consensus methodology.

For reference, Table B.1 has been included with the number of transmissions needed for each Miner to transmit data up to 2-hops based upon the aforementioned figure. These values are necessary as it will allow us to calculate the number of transmissions made during validation, based upon which Miner is communicating. Furthermore, broadcasting a block throughout the network would mean using 10 transmissions as every node would retransmit the block at least once.

Table B.1: Number of transmissions performed for a packet sent with $TTL = 2$ by each Miner

Miners	2-hop
M_1	4
M_2	3
M_3	4
M_4	5

M_1 first

When M_1 wakes, it transmits its block to all neighbouring Miners at a 2-hop distance. Each Miner upon receipt, analyses the contents to determine the validity and efficiency factor. Table B.2 shows the thought process behind the consensus methodology. The column M_1 corresponds to the results of M_1's block transmission. As we can see, not all the nodes are contained in M_1's block, for analysis M_4 contains three nodes which haven't been validated. To determine what to do, the other Miners calculate the values of P_B and P_M, as shown in Analysis B.1.1.

Table B.2: Validation sequence if M_1 is the first Miner to wake up and the first to transmit their block for validation

		M_1	M_4	Final	
M_1	1		✗	✗	– Transmit
	2		✗	✗	
	3		✓	✓	
M_2	1	✓		✓	– No Transmit
	2	✓		✓	
M_3	2	✓	✗	✓	– No Transmit
	3	✓	✓	✓	
	4	✗	✓	✓	
M_4	3	✓		✗	– Transmit
	4	✗		✗	
	5	✗		✗	
	6	✗		✗	

Analysis B.1.1

$$
\begin{aligned}
M_2 &\left\{ \begin{array}{l} P_M = \frac{|(1,2)\cap(1,2,3)|}{|(1,2,3)|} = \frac{|(1,2)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(1,2)\cap(1,2,3)|}{|(1,2)|} = \frac{|(1,2)|}{|(1,2)|} = \frac{2}{2} = 100\% \end{array} \right\} P_B \text{ higher, no transmit} \\
M_3 &\left\{ \begin{array}{l} P_M = \frac{|(2,3,4)\cap(1,2,3)|}{|(1,2,3)|} = \frac{|(2,3)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(2,3,4)\cap(1,2,3)|}{|(2,3,4)|} = \frac{|(1,2)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit} \\
M_4 &\left\{ \begin{array}{l} P_M = \frac{|(3,4,5,6)\cap(1,2,3)|}{|(1,2,3)|} = \frac{|(3)|}{|(1,2,3)|} = \frac{1}{3} = 33\% \\ P_B = \frac{|(3,4,5,6)\cap(1,2,3)|}{|3,4,5,6)|} = \frac{|(3)|}{|(3,4,5,6)|} = \frac{1}{4} = 25\% \end{array} \right\} P_M \text{ higher, transmit}
\end{aligned}
$$

As we can see, both M_2 and M_3 determine M_1's block to be more efficient or equal to their own, so they do not overrule it with theirs. However, that is not the case with M_4, which has deemed their block as more efficient and has labelled it for transmission. The corresponding computation is

visible in the column M_4 of Table B.2 and the results of the efficiency calculation are shown in Analysis B.1.2.

Analysis B.1.2

$$M_1 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_1 \text{ has been overruled by } M_4$$

$$M_2 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ is out of 2-hop range of } M_4$$

$$M_3 \left\{ \begin{array}{l} P_M = \frac{|(2,3,4) \cap (3,4,5,6)|}{|(3,4,5,6)|} = \frac{|(3,4)|}{|(3,4,5,6)|} = \frac{2}{4} = 56\% \\ P_B = \frac{|(2,3,4) \cap (3,4,5,6)|}{|(2,3,4)|} = \frac{|(3,4)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \end{array} \right\} P_B \text{ higher, no transmit}$$

As a result, since M_3 doesn't override M_4, it considers itself as the winner and broadcasts its block. However, as we can see in the last column of Table B.2 not all the nodes in M_1's block have been confirmed by M_4. This means, that M_1 is the authority over nodes 1 and 2 and since no Miner has overridden them, M_1 also broadcasts its block as well. The resulting transmissions are shown in Analysis B.1.3.

Analysis B.1.3

$$M_4 = (3, 4, 5, 6)$$
$$M_1 = (1, 2)$$

We can now determine the number of transmissions resulting from the overall exchange, as presented in Analysis B.1.4. As a result, we can see that M_1 and M_4 transmitted their blocks for validation, with 4 and 5 transmissions respectively as presented in Table B.1. Furthermore, we also see that since two blocks were inserted, the number of transmissions is multiplied by the number of Miners, here two. The total value for this exchange is, therefore, 29 transmissions with two blocks.

Analysis B.1.4

$$M_1(4) + M_4(5) + (2 \times 10) = 29$$

M_2 first

If M_2 were to wake up first, then the network would operate differently, as is visualised in Table B.3. As we can see, the first column corresponds to M_2, which in this case doesn't reach M_4 due to it being too far away. As a result, it is possible for M_4 to awaken before the consensus process following M_2's completed transmission, however, we suppose that is not the case for this analysis and the entire computation finishes before M_4 ever wakes up. The analysis of the reception of M_2's block is presented in Analysis B.1.5.

Table B.3: Validation sequence if M_2 is the first Miner to wake up and the first to transmit their block for validation. Contains two potential situations, where either M_1 or M_3 responds first, separated with double lines.

		M_2	M_1	M_4	Final		M_3	M_4	Final	
M_1	1	✓		✗	✗		✓	✗	✓	
	2	✓	╳	✗	✗	– Transmit	✓	✗	✓	– No Transmit
	3	✗		✓	✓		✓	✓	✓	
M_2	1	╳	✓	╳	✓	– No Transmit	✗	╳	✗	– Transmit
	2		✓		✓		✓		✓	
M_3	2	✓	✗	✓	✓			✗	✗	– Transmit
	3	✗	✓	✓	✓	– No Transmit	╳	✓	✓	
	4	✗	✓	✓	✓			✓	✓	
M_4	3		✓		✗		✓		✗	– Transmit
	4	╳	✗	╳	✗	– Transmit	✓	╳	✗	
	5		✗		✗		✗		✗	
	6		✗		✗		✗		✗	

Analysis B.1.5

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(1,2)|}{|(1,2)|} = \frac{|(1,2)|}{|(1,2)|} = \frac{2}{2} = 100\% \\ P_B = \frac{|(1,2,3)\cap(1,2)|}{|(1,2,3)|} = \frac{|(1,2)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \end{array} \right\} P_M \text{ higher, transmit}$$

$$M_3 \left\{ \begin{array}{l} P_M = \frac{|(2,3,4)\cap(1,2)|}{|(1,2)|} = \frac{|(2)|}{|(1,2)|} = \frac{1}{2} = 50\% \\ P_B = \frac{|(2,3,4)\cap(1,2)|}{|(2,3,4)|} = \frac{|(2)|}{|(2,3,4)|} = \frac{1}{3} = 33\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_4 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_4 \text{ is out of 2-hop range of } M_2$$

Analysis B.1.6

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\}$$

M_2 overruled by M_3

$$M_3 \left\{ \begin{array}{l} P_M = 66\% \\ P_B = 66\% \end{array} \right\}$$

Same values, no transmit

$$M_4 \left\{ \begin{array}{l} P_M = 33\% \\ P_B = 25\% \end{array} \right\}$$

P_M higher, transmit

$$M_1 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\}$$

M_1 overruled by M_4

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\}$$

M_2 out of 2-hop range of M_4

$$M_3 \left\{ \begin{array}{l} P_M = 56\% \\ P_B = 66\% \end{array} \right\}$$

P_B higher, no transmit

The main difference here is the results of the consensus determine both M_1 and M_3 are more efficient than M_2 and thus must transmit their blocks instead. This is illustrated in Table B.3 where M_1 is visible on the left-hand side, after the double-lined separator, and M_3 on the right, again after the double lined separator. We shall not analyse in detail the results of M_1 responding first, as these are identical to what has previously been presented when M_1 woke up first, but a brief recap of M_1 and subsequent actions is visible in Analysis B.1.6. The overview, however, is visible in Table B.3, where we can see that, as is the case with M_1 waking up first, both M_1 and M_4 transmit their blocks. Instead, we shall analyse the new case where M_3 responds instead, which is presented in Analysis B.1.7.

Analysis B.1.7

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(2,3)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(1,2,3)\cap(2,3,4)|}{|(1,2,3)|} = \frac{|(2,3)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ has been overruled by } M_3$$

$$M_4 \left\{ \begin{array}{l} P_M = \frac{|(3,4,5,6)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(3,4)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(3,4,5,6)\cap(2,3,4)|}{|3,4,5,6)|} = \frac{|(3,4)|}{|(3,4,5,6)|} = \frac{2}{4} = 50\% \end{array} \right\} P_M \text{ higher, transmit}$$

Here we can see that once again M_4 is the more efficient and, therefore, transmits their block again for validation, overruling M_3. As a result, since both M_2 and M_3 have been overruled, only M_1 can respond to M_4. The results are presented in Analysis B.1.8.

Analysis B.1.8

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(3,4,5,6)|} = \frac{|(3)|}{|(3,4,5,6)|} = \frac{1}{4} = 25\% \\ P_B = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(1,2,3)|} = \frac{|(3)|}{|(1,2,3)|} = \frac{1}{3} = 33\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ has been overruled by } M_3$$

$$M_3 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_3 \text{ has been overruled by } M_4$$

We can see that M_1 does not override M_4, meaning M_4's block is inserted into the blockchain. However, in the final column on the right of Table B.3, we can see that M_2 possesses node 1 which hasn't been validated, and M_3 has node 2. Since M_3 overrode M_2 originally, these Miners will only transmit blocks containing the nodes which haven't been validated with no new efficiency calculation performed. As a result, the final transmissions for this scenario are shown in Analysis B.1.10. The final transmissions for the other scenario, where M_1 transmits instead of M_3 are identical to those when M_1 woke up first and are summarised in Analysis B.1.9.

Analysis B.1.9

$$M_4 = (3,4,5,6)$$
$$M_1 = (1,2)$$

Analysis B.1.10

$$M_4 = (3,4,5,6)$$
$$M_3 = (2)$$
$$M_2 = (1)$$

In all, two independent scenarios occur when M_2 is the first Miner to awaken, dependant on which Miner responds first. The overall exchanges are presented in Analysis B.1.11, once again using the number of transmissions previously presented in Table B.1. As we can see, a total of three exchanges were needed to reach consensus, with two blocks for the first case and three for the second, resulting in 32 and 42 total transmissions respectively.

Analysis B.1.11

$$M_2(3) + \begin{cases} M_1(4) + M_4(5) + (2 \times 10) = 32 \\ M_3(4) + M_4(5) + (3 \times 10) = 42 \end{cases}$$

M_3 first

In this third scenario, M_3 is the lucky one to wake up first. The resulting communications and analysis are portrayed in Table B.4, with the left most column corresponding to the results of M_3's transmission. The analysis of M_3's block is presented in Analysis B.1.12.

Table B.4: Validation sequence if M_3 is the first Miner to wake up and the first to transmit their block for validation. Contains two potential situations, where either M_1 or M_2 responds first, separated with double lines.

		M_3	M_4	M_1	**Final**		M_2	M_1	**Final**	
M_1	1	✗	✗		✗		✓		✗	
	2	✓	✓		✗	– Transmit	✓		✗	– Transmit
	3	✓	✓		✗		✓		✗	
M_2	1	✗		✓	✓			✓	✓	
	2	✓		✓	✓	– No Transmit		✓	✓	– No Transmit
M_3	2		✗	✓	✓		✓	✓	✓	
	3		✓	✓	✓	– No Transmit	✓	✓	✓	– No Transmit
	4		✓	✓	✓		✓	✓	✓	
M_4	3	✓		✓	✓			✓	✓	
	4	✓		✗	✗	– Transmit		✗	✗	– Transmit
	5	✗		✗	✗			✗	✗	
	6	✗		✗	✗			✗	✗	

Analysis B.1.12

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(2,3)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(1,2,3)\cap(2,3,4)|}{|(1,2,3)|} = \frac{|(2,3)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_2 \left\{ \begin{array}{l} P_M = \frac{|(1,2)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(2)|}{|(2,3,4)|} = \frac{1}{3} = 33\% \\ P_B = \frac{|(1,2)\cap(2,3,4)|}{|(1,2)|} = \frac{|(2)|}{|(1,2)|} = \frac{1}{2} = 50\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_4 \left\{ \begin{array}{l} P_M = \frac{|(3,4,5,6)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(3,4)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(3,4,5,6)\cap(2,3,4)|}{|3,4,5,6)|} = \frac{|(3,4)|}{|(3,4,5,6)|} = \frac{2}{4} = 50\% \end{array} \right\} P_M \text{ higher, transmit}$$

We can see that once again, M_4 is selected as the most efficient, transmitting its block and overriding that of M_3. The resulting analysis is presented in Analysis B.1.13.

Analysis B.1.13

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(3,4,5,6)|} = \frac{|(3)|}{|(3,4,5,6)|} = \frac{1}{4} = 25\% \\ P_B = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(1,2,3)|} = \frac{|(3)|}{|(1,2,3)|} = \frac{1}{3} = 33\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ is out of 2-hop range of } M_4$$

$$M_3 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_3 \text{ has been overruled by } M_4$$

We can see that M_4 still possesses the highest block efficiency and M_1 cannot override it. In the previous cases, this was the end of the consensus analysis as all nodes were validated and M_4 was the winner. However, if we look at Table B.4, in particular the column corresponding to M_4, we can see that node 1 was never validated as M_1 and M_2 never sent their blocks for validation. The algorithm employed is capable of detecting if certain nodes haven't been validated, allowing to force a Miner to transmit their block to allow this validation. Since both M_1 and M_2 are responsible for node 1, both can be woken up. As a result, dependant on which transmits first, the actions of the Miners change once more. We will first take a look at what happens if M_1 is the first to transmit their block, presented in Analysis B.1.14.

Analysis B.1.14

$$M_2 \left\{ \begin{array}{l} P_M = \frac{|(1,2)\cap(1,2,3)|}{|(1,2,3)|} = \frac{|(1,2)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(1,2)\cap(1,2,3)|}{|(1,2)|} = \frac{|(1,2)|}{|(1,2)|} = \frac{2}{2} = 100\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_3 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_3 \text{ has been overruled by } M_4$$

$$M_4 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_4 \text{ has been overruled by } M_1$$

Since both M_3 and M_4 have, therefore, been overridden only M_2 can analyse the contents of M_1's block. However, as we can see, the result does not allow it to override the it, meaning M_1 is the final Miner to insert their block into the blockchain.

As stated, another possibility is for M_2 to awaken before M_1. Here, we take a look at that eventuality, presented in Analysis B.1.15.

Analysis B.1.15

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(1,2)|}{|(1,2)|} = \frac{|(1,2)|}{|(1,2)|} = \frac{2}{2} = 100\% \\ P_B = \frac{|(1,2,3)\cap(1,2)|}{|(1,2,3)|} = \frac{|(1,2)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \end{array} \right\} P_M \text{ higher, transmit}$$

$$M_3 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_3 \text{ has been overruled by } M_4$$

$$M_4 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_4 \text{ is out of range of } M_2$$

In this case, M_1 is more efficient than M_2, meaning that it will once again transmit its block for validation. However, since all other Miners have been overridden, there is no one left to perform validation, meaning that M_1 is chosen by default. A summarised overview of this is provided in Analysis B.1.16.

Analysis B.1.16

$$M_2 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} M_2 \text{ overruled by } M_1$$

$$M_3 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} M_3 \text{ overruled by } M_4$$

$$M_4 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} M_4 \text{ overruled by } M_1$$

In either final case, the same Miners are selected for block distribution. Indeed, M_1 is the final Miner selected and by consulting the final columns for both scenarios in Table B.4, we can see that M_4 still possesses control over un validated nodes, meaning it will also transmit its block as well. Thus, the final transmissions for both scenarios are presented in Analysis B.1.17.

Analysis B.1.17

$$M_1 = (1, 2, 3)$$
$$M_4 = (4, 5, 6)$$

The resulting overall number of transmissions for M_3 waking up first is defined in Analysis B.1.18. Although the same number and even the same blocks are transmitted, the consensus journey taken is not the same. As a result, in the second case when M_2 transmits before M_1, we increase the number of transmissions from 33 to 36. We can identify that this is a

potential flaw in the consensus methodology, as M_2's transmission here is redundant as M_1 overrides it in any case.

Analysis B.1.18

$$M_3(3) + M_4(5) + \begin{cases} M_1(4) + (2 \times 10) = 33 \\ M_2(3) + M_1(4) + (2 \times 10) = 36 \end{cases}$$

M_4 first

In the previous three scenarios, M_4 has always had its block inserted into the blockchain. In this situation, we will evaluate what would happen if it was the first Miner to awaken and transmit its block before all others. The resulting block validation is presented in Table B.5 and the analysis of M_4's block is presented in Analysis B.1.19.

Table B.5: Validation sequence if M_4 is the first Miner to wake up and the first to transmit their block for validation. Contains two potential situations, where either M_1 or M_3 responds first, separated with double lines.

		M_4	M_2	M_1	**Final**		M_3	**Final**	
M_1	1	✗	✓		✗		✓	✓	
	2	✓	✓	╳	✗	– Transmit	✓	✓	– No Transmit
	3	✓	✓		✗		✓	✓	
M_2	1	╳	╳	✓	✓		✗	✗	
	2			✓	✓	– No Transmit	✓	✓	– Transmit
M_3	2	✗	✓	✓	✓			✗	
	3	✓	✓	✓	✓	– No Transmit	╳	✗	– Transmit
	4	✓	✓	✓	✓			✗	
M_4	3			✓	✓		✓	✓	
	4	╳	╳	✗	✗	– Transmit	✓	✓	– Transmit
	5			✗	✗		✗	✗	
	6			✗	✗		✗	✗	

Analysis B.1.19

M_1: $P_M = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(3,4,5,6)|} = \frac{|(3)|}{|(3,4,5,6)|} = \frac{1}{4} = 25\%$, $P_B = \frac{|(1,2,3)\cap(3,4,5,6)|}{|(1,2,3)|} = \frac{|(3)|}{|(1,2,3)|} = \frac{1}{3} = 33\%$ — P_B higher, no transmit

M_2: $\varnothing$, $\varnothing$ — No calculation as M_2 is out of 2-hop range of M_4

M_3: $P_M = \frac{|(2,3,4)\cap(3,4,5,6)|}{|(3,4,5,6)|} = \frac{|(3,4)|}{|(3,4,5,6)|} = \frac{2}{4} = 56\%$, $P_B = \frac{|(2,3,4)\cap(3,4,5,6)|}{|(2,3,4)|} = \frac{|(3,4)|}{|(2,3,4)|} = \frac{2}{3} = 66\%$ — P_B higher, no transmit

Analysis B.1.20

M_1: $P_M = 100\%$, $P_B = 66\%$ — P_M higher, transmit

M_3: $P_M = 50\%$, $P_B = 33\%$ — Same values, no transmit

M_4: $\varnothing$, $\varnothing$ — M_4 out of 2-hop range of M_2

As expected, no other Miners are capable of overloading M_4, as previously observed. However, M_2 is still out of range, meaning it hasn't yet received a block, thus, it transmit its block to its neighbouring Miners. This means the resulting analysis is the same as if M_2 would have transmitted its block first, thus we won't analyse it here once more but the results are available in M_2's column of Table B.5 and a summary is visible in Analysis B.1.20.

As seen previously, both M_1 and M_3 would seek to override M_2 as their blocks are more efficient. We will first analyse what would happen if M_1 would respond first, shown in Analysis B.1.21. In either case, since

M_4 would receive the responding block, it would also believe it is being overridden.

Analysis B.1.21

$$M_2 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ has been overruled by } M_3$$

$$M_3 \left\{ \begin{array}{l} P_M = \frac{|(2,3,4)\cap(1,2,3)|}{|(1,2,3)|} = \frac{|(2,3)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(2,3,4)\cap(1,2,3)|}{|(2,3,4)|} = \frac{|(1,2)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_4 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_4 \text{ has been overruled by } M_3$$

As we can see, M_3 does not overrule M_1, meaning that M_1 will once more join M_4 in sending its block to the network. We will now take a look at what would happen in M_3 would respond instead, presented in Analysis B.1.22.

Analysis B.1.22

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3)\cap(2,3,4)|}{|(2,3,4)|} = \frac{|(2,3)|}{|(2,3,4)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(1,2,3)\cap(2,3,4)|}{|(1,2,3)|} = \frac{|(2,3)|}{|(1,2,3)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_2 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_2 \text{ has been overruled by } M_3$$

$$M_4 \left\{ \begin{array}{c} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_4 \text{ has been overruled by } M_3$$

In the same spirit as previously, M_1 does not override M_3 since both blocks have an equal efficiency. The main difference here is that, as is visible in the far-right column in Table B.5, M_2 has not been overridden for the activities of node 1, whereas M_1 has been overridden due to receiving M_3's block, stopping it from overruling M_2's original transmission.

As a result, these two cases once again have different final blocks needing distribution. The final transmission for the first case, where M_1 transmits first are the same as previously and, therefore, are summarised in Analysis B.1.23. However, the results for the second case where M_3 responds first are shown in Analysis B.1.24.

Analysis B.1.23

$$M_1 = (1, 2, 3)$$
$$M_4 = (4, 5, 6)$$

Analysis B.1.24

$$M_3 = (2, 3, 4)$$
$$M_4 = (4, 5, 6)$$
$$M_2 = (1)$$

Indeed, in this case we have three block transmissions compared to two in the previous situation. As a result, the final number of transmissions for M_4 waking first is defined in Analysis B.1.25. We can see that the number of transmissions needed for the consensus algorithm remains the same. Only the number of blocks here varies by one, meaning the

total number of transmissions increases from 32 when M_1 responds, to 42 when M_3 responds instead.

Analysis B.1.25

$$M_4(5) + M_2(3) + \begin{cases} M_1(4) + (2 \times 10) = 32 \\ M_3(4) + (3 \times 10) = 42 \end{cases}$$

Final Results

From this analysis, we can determine the possible sequence of events performed by the Miners during validation for this network. We can also calculate the number of blocks needed as well as their different contents and the total number of Transmissions. Analysis B.1.26 shows an overview of the previous analysis for all four possible wake up scenarios.

Analysis B.1.26

$$M_1 \text{ first } \rightarrow M_1(4) + M_4(5) + (2 \times 10) = 29 \text{ Transmissions with 2 blocks}$$

$$M_2 \text{ first } \rightarrow M_2(3) + \begin{cases} M_1(4) + M_4(5) + (2 \times 10) = 32 \text{ Transmissions with 2 blocks} \\ M_3(4) + M_4(5) + (3 \times 10) = 42 \text{ Transmissions with 3 blocks} \end{cases}$$

$$M_3 \text{ first } \rightarrow M_3(3) + M_4(5) + \begin{cases} M_1(4) + (2 \times 10) = 33 \text{ Transmissions with 2 blocks} \\ M_2(3) + M_1(4) + (2 \times 10) \\ \qquad = 36 \text{ Transmissions with 2 blocks} \end{cases}$$

$$M_4 \text{ first } \rightarrow M_4(5) + M_2(3) + \begin{cases} M_1(4) + (2 \times 10) = 32 \text{ Transmissions with 2 blocks} \\ M_3(4) + (3 \times 10) = 42 \text{ Transmissions with 3 blocks} \end{cases}$$

As we can see, with four wake-up scenarios we end up with seven different possibilities overall with varying sequences of events. However, we can observe that the number of transmissions is comprised within $[29; 42]$ with the number of blocks being generally two, with only two scenarios needing three. From this, we can compute the average number of transmissions across all seven scenarios, leaving us with on average 35 per validation for this topology.

B.2 Scenario *B*

The next topology is presented in Figure B.2. Here, there is an eight-node route numbered from 1 to 8, once more indicated by the red arrows. The five surrounding nodes are the Miners, numbered from M_1 to M_5 and their interconnections indicated with the grey dotted line. Furthermore, it is important to note that in this topology, the network is semi-separated in two, with three Miners on one side and two on the other, impacting how the consensus mechanism functions. Contrary to the layout used in Scenario *A*, this topology is extracted from [176], where it is used to demonstrate the mining actions of the authors approach. By using this topology, we can evaluate how our metric functions on an existing and confirmed topology.

[176]: Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450

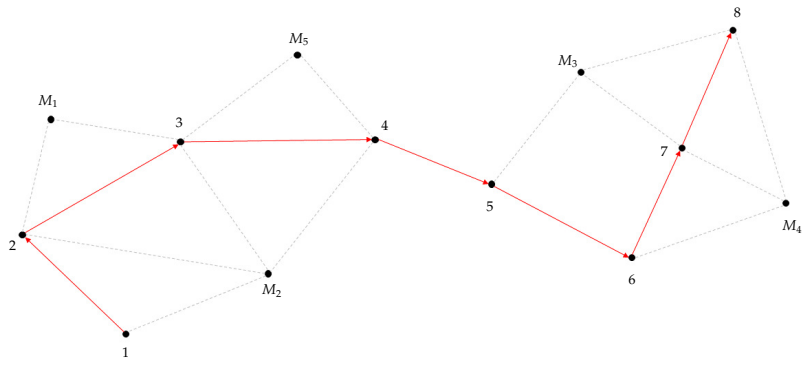


Figure B.2: 13-Node network topology with an eight-node route and five Miners. The network itself is semi-separated, blocking Miners from communicating between both sides

Although at first sight this network seems more complex than the previous, due to the separation between the miners, this actually renders the consensus computation easier. Indeed, since the Miners only transmit their blocks up to 2-hops during analysis, it is not possible for M_1, M_2 or M_5, to reach M_3 or M_4 and vice-versa. This means that two consensus negotiations can take place at the same time, one between M_1, M_2 and M_5 and the other between M_3 and M_4. We will analyse all five potential consensus results, before aggregating the results to gain an overview of the network.

As we previously used in Section B.1, we provide the number of transmissions needed for each Miner for each 2-hop analysis, visible in Table B.6. We can also determine that every block that will be inserted into the blockchain will need to be transmitted a total of 13 times to traverse the whole network.

Table B.6: Number of transmissions performed for a packet sent with $TTL = 2$ by each Miner

Miners	2-hop
M_1	3
M_2	5
M_3	4
M_4	4
M_5	3

We will start our analysis on the left side of the network, with Miners M_1, M_2 and M_5, before turning our attention to the other side and Miners M_3 and M_4.

M_1 First

We will start off by analysing what would happen if M_1 were to wake up first and transmit their block. The resulting validation is presented in Table B.7 and the analysis of M_1's block is performed in Analysis B.2.1.

		M_1	M_2	**Final**	
M_1	2	✗	✓	✓	– No Transmit
	3		✓	✓	
M_2	1	✗		✗	– Transmit
	2	✓	✗	✗	
	3	✓		✗	
	4	✗		✗	
M_5	3	✓	✓	✓	– No Transmit
	4	✗	✓	✓	

Table B.7: Validation sequence if M_1 is the first Miner to wake up and the first to transmit their block for validation.

Analysis B.2.1

$$M_2 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3,4)\cap(2,3)|}{|(2,3)|} = \frac{|(2,3)|}{|(2,3)|} = \frac{2}{2} = 100\% \\ P_B = \frac{|(1,2,3,4)\cap(2,3)|}{|(1,2,3,4)|} = \frac{|(2,3)|}{|(1,2,3,4)|} = \frac{2}{4} = 50\% \end{array} \right\} P_M \text{ higher, transmit}$$

$$M_5 \left\{ \begin{array}{l} P_M = \frac{|(3,4)\cap(2,3)|}{|(2,3)|} = \frac{|(3)|}{|(2,3)|} = \frac{1}{2} = 50\% \\ P_B = \frac{|(3,4)\cap(2,3)|}{|(3,4)|} = \frac{|(3)|}{|(3,4)|} = \frac{1}{2} = 50\% \end{array} \right\} \text{Same values, no transmit}$$

Here, we can see the M_2 determines its own block as more efficient than M_1's. Thus, M_2 takes the stand and transmits its block to the other two. The analysis is presented in Analysis B.2.2.

Analysis B.2.2

$$M_1 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} \text{No calculation as } M_1 \text{ has been overruled by } M_2$$

$$M_5 \left\{ \begin{array}{l} P_M = \frac{|(3,4)\cap(1,2,3,4)|}{|(1,2,3,4)|} = \frac{|(3,4)|}{|(1,2,3,4)|} = \frac{2}{4} = 50\% \\ P_B = \frac{|(3,4)\cap(1,2,3,4)|}{|(3,4)|} = \frac{|(3,4)|}{|(3,4)|} = \frac{2}{2} = 100\% \end{array} \right\} P_B \text{ higher, no transmit}$$

In this case, M_5 considers M_2 to be more efficient overall and thus does not overrule it. As we can see in the final results of Table B.7, all nodes are validated with only M_2 needing to transmit their block. We can, therefore, determine that the final transmission for this case, shown in Analysis B.2.3.

Analysis B.2.3

$$M_2 = (1, 2, 3, 4)$$

The final number of transmissions needed to share these results is presented in Analysis B.2.4. We can see that for this part of the network to reach consensus, 21 transmissions are needed in total with only one block inserted into the blockchain and two consensus calculations. The number of transmissions for each Miner is once again taken from the provided previously Table C.16.

Analysis B.2.4

$$M_1(3) + M_2(5) + (1 \times 13) = 21$$

M_2 First

Since M_2 was the most efficient block in the previous case, this time we will analyse how the Miners will respond if it is the first to wake up and transmit its block. The results of the consensus are visible in Table B.8 and the analysis of M_2's block is presented in Analysis B.2.5.

		M_2	Final	
M_1	2	✓	✓	– No Transmit
	3	✓	✓	
M_2	1		✗	– Transmit
	2		✗	
	3		✗	
	4		✗	
M_5	3	✓	✓	– No Transmit
	4	✓	✓	

Table B.8: Validation sequence if M_2 is the first Miner to wake up and the first to transmit their block for validation.

Analysis B.2.5

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(1,2)\cap(1,2,3,4)|}{|(1,2,3,4)|} = \frac{|(1,2)|}{|(1,2,3,4)|} = \frac{2}{4} = 50\% \\ P_B = \frac{|(1,2)\cap(1,2,3,4)|}{|(1,2)|} = \frac{|(1,2)|}{|(1,2)|} = \frac{2}{2} = 100\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_5 \left\{ \begin{array}{l} P_M = \frac{|(3,4)\cap(1,2,3,4)|}{|(1,2,3,4)|} = \frac{|(3,4)|}{|(1,2,3,4)|} = \frac{2}{4} = 50\% \\ P_B = \frac{|(3,4)\cap(1,2,3,4)|}{|(3,4)|} = \frac{|(3,4)|}{|(3,4)|} = \frac{2}{2} = 100\% \end{array} \right\} P_B \text{ higher, no transmit}$$

In this case, the consensus is very quick and straight forward as neither Miner is able to overrule M_2. As we can see, M_2's block once more contains all the node activities necessary for sharing, meaning neither M_1 nor M_5 needs to insert a block to fill in for a missing node. Since M_2 was the winner previously, its block is therefore the same as previously, summarised in Analysis B.2.6. The resulting number of transmissions is also a lot easier and is presented in Analysis B.2.7. Since only one block was shared for validation, it is naturally the only one to be inserted into the blockchain. Thus, only 18 transmissions are needed in total for consensus to be reached.

Analysis B.2.6

$$M_2 = (1, 2, 3, 4)$$

Analysis B.2.7

$$M_2(5) + (1 \times 13) = 18$$

M_5 First

Since M_1 and M_5 possess the same proportion of nodes, both covered by M_2 it is not surprising that the results of this analysis are highly predictable. However, we will still progress with the analysis to demonstrate how the system functions. The results are, therefore, presented in Table B.9 and the analysis of M_5's block is presented in Analysis B.2.8.

		M_5	M_2	Final	
M_1	2	✗	✓	✓	– No Transmit
	3	✓	✓	✓	
M_2	1	✗		✗	– Transmit
	2	✗		✗	
	3	✓		✗	
	4	✓		✗	
M_5	3		✓	✓	– No Transmit
	4		✓	✓	

Table B.9: Validation sequence if M_5 is the first Miner to wake up and the first to transmit their block for validation.

Analysis B.2.8

$$M_1 \left\{ \begin{array}{l} P_M = \frac{|(2,3)\cap(3,4)|}{|(3,4)|} = \frac{|(3)|}{|(3,4)|} = \frac{1}{2} = 50\% \\ P_B = \frac{|(2,3)\cap(3,4)|}{|(2,3)|} = \frac{|(3)|}{|(2,3)|} = \frac{1}{2} = 50\% \end{array} \right\} \text{Same values, no transmit}$$

$$M_2 \left\{ \begin{array}{l} P_M = \frac{|(1,2,3,4)\cap(3,4)|}{|(3,4)|} = \frac{|(3,4)|}{|(3,4)|} = \frac{2}{2} = 100\% \\ P_B = \frac{|(1,2,3,4)\cap(3,4)|}{|(1,2,3,4)|} = \frac{|(3,4)|}{|(1,2,3,4)|} = \frac{2}{4} = 50\% \end{array} \right\} P_M \text{ higher, transmit}$$

It come as no surprise that once more, M_2 is the most efficient Miner in range. Since in the previous two cases, M_2 has not been overridden, the analysis of its block remains the same, summarised in Analysis B.2.9. The results, however, are still visible in Table B.9. Once again, the transmitted block is the same as previously and summarised in Analysis B.2.10, so we will turn our attention to the calculation of the total number of transmissions, presented in Analysis B.2.11. As expected, the number of transmissions is the same as when M_1 transmitted first, with 21 transmissions needed to reach consensus and share M_2's bloc with the network.

Analysis B.2.9

$$M_1 \left\{ \begin{array}{l} P_M = 50\% \\ P_B = 100\% \end{array} \right\} P_B \text{ higher, no transmit}$$

$$M_5 \left\{ \begin{array}{l} \varnothing \\ \varnothing \end{array} \right\} M_5 \text{ overruled by } M_2$$

Analysis B.2.10

$$M_2 = (1, 2, 3, 4)$$

Analysis B.2.11

$$M_5(3) + M_2(5) + (1 \times 13) = 21$$

M_3 First

We now turn our attention to the other side of the network, this time with M_3 waking up first. Compared to previously, only two miners cover this section of the network. Furthermore, neither cover the exact same nodes, meaning they both possess unique nodes of which they are the only validators, which will impact the results of the consensus and the inserted blocks. The results are visible in Table B.10 with the analysis of M_3's block in Analysis B.2.12.

Table B.10: Validation sequence if M_3 is the first Miner to wake up and the first to transmit their block for validation.

		M_3	M_4	**Final**	
	5		✗	✗	
M_3	7		✓	✓	– Transmit
	8		✓	✓	
	6	✗		✗	
M_4	7	✓		✗	– Transmit
	8	✓		✗	

Analysis B.2.12

$$M_4 \left\{ \begin{array}{l} P_M = \frac{|(6,7,8)\cap(5,7,8)|}{|(5,7,8)|} = \frac{|(7,8)|}{|(5,7,8)|} = \frac{2}{3} = 66\% \\ P_B = \frac{|(6,7,8)\cap(5,7,8)|}{|(6,7,8)|} = \frac{|(7,8)|}{|(6,7,8)|} = \frac{2}{3} = 66\% \end{array} \right\} \text{Same values, no transmit}$$

As we can see, M_4 doesn't override M_3's block, leaving M_3 to believe it is the most efficient. However, as we can see M_3 doesn't cover node 6, meaning that since M_4 doesn't receive any other blocks with an analysis of node 6's activities, it will transmit its own block for validation. As a

result, since this transmission overrides M_3's block, it doesn't perform an analysis on M_4's data, which accepts its block by default, due to no response. Although this solved the problem of node 6, this opens another problem for node 5, which is under the responsibility of M_4, meaning that it must transmit its block after all containing the activities of the missing node. The resulting transmissions are, therefore, presented in Analysis B.2.13.

Analysis B.2.13

$$M_4 = (6, 7, 8)$$
$$M_3 = (5)$$

The total number of transmissions is defined in Analysis B.2.14. Since both Miners shared their blocks for validation, the maximum number of consensus transmissions is reached. Furthermore, with not one block like previously but with both Miners inserting their blocks, the number of transmissions reaches the maximum possible value for this network, resulting in 34 in total for both consensus, and block insertion.

Analysis B.2.14

$$M_3(4) + M_4(4) + (2 \times 13) = 34$$

M_4 First

The final analysis is on the same part of the network as previously, only this time we reverse the wake-up order of the two Miners. In this case, we consider that M_4 wakes-up before M_3 and transmits its block for validation. The results of the consensus validation are visible in Table B.11 and the analysis of M_4's block is presented in Analysis B.2.15.

		M_3	M_4	**Final**	
M_3	5	✗		✗	
	7	✓		✗	– Transmit
	8	✓		✗	
M_4	6		✗	✗	
	7		✓	✓	– Transmit
	8		✓	✓	

Table B.11: Validation sequence if M_4 is the first Miner to wake up and the first to transmit their block for validation.

Analysis B.2.15

$$M_3 \left\{ \begin{array}{l} P_M = 66\% \\ P_B = 66\% \end{array} \right\} \text{Same values, no transmit}$$

It comes as no surprise once more that M_4 is not overridden and its block prepared for insertion into the blockchain. However, although M_3 decided to not transmit their block initially, since no other blocks are received which allow the validation of node 5, M_3 transmits its own. As seen previously, due to M_4 being overriden by M_3's transmission, the analysis of M_3's block will not incur a response, resulting in M_3 determining its block as valid. However, as we can see in the final column of Table B.11, node 6 is once again non validated, meaning M_4 must still transmit its block containing this nodes actions. This means that the number of blocks and the transmitting Miners are the same as previously,

with only the contents that differ. The final transmissions are presented in Analysis B.2.16

Analysis B.2.16

$$M_3 = (5, 7, 8)$$
$$M_4 = (6)$$

Since the consensus process was practically identical to previously, the total number of transmissions is also the same, only with the order of blocks revered, as presented in Analysis B.2.17. We can conclude, therefore, that this side of the network will always operate at the maximum number of transmissions, 34 with two consensus validations and two inserted blocks.

Analysis B.2.17

$$M_4(4) + M_3(4) + (2 \times 13) = 34$$

Final Results

Thanks to this analysis, we can determine the sequence of communications needed by the Miners on either side of the network. Since these operate concurrently, they do not inflict or encroach on each other, meaning that for each possibility on one side, any of the other are possible. As a result, we have decided to visualise the overall results, dependant on which node wakes up first between M_1, M_2 and M_5. This choice was made since there are only two nodes on the other side of the network, making the visualisation of the results much easier. Analysis B.2.18 shows an overview of the previously analysed wake-up scenarios.

Analysis B.2.18

$$M_1 \text{ first } \rightarrow M_1(3) + M_2(5) + (1 \times 13) + \begin{cases} M_3(4) + M_4(4) + (2 \times 13) \\ M_4(4) + M_3(4) + (2 \times 13) \end{cases}$$
$$= 55 \text{ Transmissions with 3 blocks}$$
$$M_2 \text{ first } \rightarrow M_2(5) + (1 \times 13) + \begin{cases} M_3(4) + M_4(4) + (2 \times 13) \\ M_4(4) + M_3(4) + (2 \times 13) \end{cases}$$
$$= 52 \text{ Transmissions with 3 blocks}$$
$$M_5 \text{ first } \rightarrow M_5(3) + M_2(5) + (1 \times 13) + \begin{cases} M_3(4) + M_4(4) + (2 \times 13) \\ M_4(4) + M_3(4) + (2 \times 13) \end{cases}$$
$$= 55 \text{ Transmissions with 3 blocks}$$

As we can see, although there were five wake up scenarios in total, three took place on one side of the network and two on the other at the same time. As a result, we end up with a total of six different possibilities. That being said, even with these multiple different sequences of events, we

can see that in total, three blocks were used across all scenarios. We can also see that the number of transmissions is comprised between [52; 55] with an average being 54 over all six possibilities.

C

Byzantine Consensus Validation of Routing Activities

In Chapter 5.3, we proposed an update to the consensus-methodology used previously to allow validation of both blocks and Miners, based upon the Byzantine problem. In this appendix, explore this dual consensus algorithm, extending the previous presentation by performing a theoretical analysis using three scenarios. This analysis follows on from Appendix B using the same two scenarios, allowing for an efficiency comparison. Furthermore, we include the addition of a third, more complex network, which wasn't analysed with the previous consensus algorithm. The three scenarios are as follows:

1. **Scenario** *A* - 10-node interconnected network with a six-node route and four Miners
2. **Scenario** *B* - 13-node network with an eight-node route and five Miners, separated down the centre line with three Miners on one side and two on the other with each subset out of block exchange range with the other
3. **Scenario** *C* - 22-node fully connected network possessing an eight-node route with 13 Miners as well as three relay nodes, helping to distribute the blocks throughout the network

Through this analysis, we will illustrate in detail how this double consensus mechanism works, allowing the validation of both passing blocks, but also the validation activities of the Miner For this analysis, we assume that all Miners are not malicious and the mined actions are valid. By doing this, we can visualise simply how the process unfolds, allowing the network as a whole to come to a consensus regarding nodes as well as the Miners. Furthermore, we can determine the most efficient valid Miner in the network, allowing them to add their block to the blockchain.

C.1 Scenario *A*

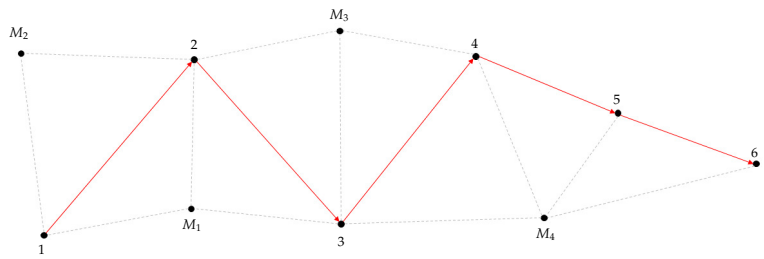


Figure C.1: 10-Node network topology with a six-node route and four Miners

Contrary the analyses performed in Appendix B, only one possible outcome can come from this method. Whereas previously the first Miner

to wake-up in the network determined how its neighbours reacted, here the consensus mechanism is based upon the entire fleet of Miners sharing their blocks. As a result, all Miners come to the same conclusion, meaning their is no ambiguity as to the correct course of action. Furthermore, since all Miners possess an overview of the expected results, any Miner which attempts to tamper with the results is immediately detected, their block dropped and their reputation decreased. Figure C.1 presents the same theoretical topology as used previously upon which this analysis will take place.

2-Hop Routing Consensus

We start off with the first consensus stage: Routing Validation presented in Table C.1, where the columns correspond to the vision of each Miner and the rows the contents of the received blocks. What we can immediately see is that Miners M2 and M4 are out of 2-hop communications range, so they cannot participate in the consensus for each other. Furthermore, nodes 5 and 6 are only mined by M4, meaning they are never validated by consensus, therefore, leaving M4 to be considered the authority over these two nodes. As a result, each Miner manages to validate different sets of nodes overall, adding some more complexity to the overall process. As explained before, since no malicious activities are expressed here the Miners mark all Miner actions as being valid. All un-validated nodes have been coloured **red** for ease of identification.

Table C.1: Consensus matrix representing the different blocks received up to 2-hops, allowing the identification of all validated and un validated nodes

	M_1	M_2	M_3	M_4
M_1	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3
M_2	1, 2	1, 2	1, 2	
M_3	2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4
M_4	3, 4, 5, 6		3, 4, 5, 6	3, 4, 5, 6
Validated	**1, 2, 3, 4**	**1, 2, 3**	**1, 2, 3, 4**	**2, 3, 4, 5, 6**

4-Hop Miner Consensus

Routing Validation

With this first stage complete, we enter the second stage with the 4-hop Miner validation. Firstly, we have a look at the second routing validation phase, presented in Table C.2. We can immediately see that, compared to previously, Miners M2 and M4 have been able to share routing information between each other. Furthermore, M4 has included the un-validated results for nodes 5 and 6, since during the consensus round, no other nodes contained these values, thus as the authority on them only this node can compute the values. We can also see that the concatenated values contain all nodes for the whole route, meaning all Miners are now capable of inserting the entire action list into the blockchain.

Table C.2: Validation matrix representing the distribution of confirmed routing information up to 4-hops, allowing the concatenation of all validated nodes

	M_1	M_2	M_3	M_4
M_1	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
M_2	1, 2, 3	1, 2, 3	1, 2, 3	1, 2, 3
M_3	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
M_4	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6
Concatenated	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**

Before the Miners can insert their blocks into the blockchain, they must calculate the routing score based upon the validation results, shown in Table C.3. Here once more we can see the scores represented in fraction format for ease of understanding and calculation. We can, therefore, understand how the calculation is performed. By taking a look at M1 for example, we can see it has a score of $\frac{4}{6}$, where in Table C.2 we can see that M1's block contained four nodes[1] whereas there are six total concatenated nodes[2]. By looking at the routing scores, we can see that M 4 has the highest since it contains the largest portion of confirmed nodes in its own block.

1: (1, 2, 3, 4)

2: (1, 2, 3, 4, 5, 6)

	M_1	M_2	M_3	M_4
M_1	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
M_2	$\frac{3}{6}$	$\frac{3}{6}$	$\frac{3}{6}$	$\frac{3}{6}$
M_3	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
M_4	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$

Table C.3: Calculation of the "Routing Score", allowing the identification of the most efficient routing Miner, possessing the highest ratio of mined to confirmed nodes. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

Miner Consensus and Validation

	M_1	M_2	M_3	M_4
M_1	M_2, M_3, M_4	M_2, M_3, M_4	M_2, M_3, M_4	M_2, M_3, M_4
M_2	M_1, M_3	M_1, M_3	M_1, M_3	M_1, M_3
M_3	M_1, M_2, M_4	M_1, M_2, M_4	M_1, M_2, M_4	M_1, M_2, M_4
M_4	M_1, M_3	M_1, M_3	M_1, M_3	M_1, M_3
Validated	M_1, M_2, M_3, M_4	M_1, M_2, M_3, M_4	M_1, M_2, M_3, M_4	M_1, M_2, M_3, M_4

Table C.4: Consensus matrix associated the Miners which have been validated with that which performed the validation, based upon the routing routing validation matrix in Table C.2

Now we have computed the routing score, we can turn our attention to the Miner validation process, shown in Table C.4. As we can see here, all Miners are able to validate themselves with each other. By coupling this with the results of the routing validation matrix in Table C.2, each Miner is capable of inserting a single block, containing all the results of the behavioural analysis, as well as the Miner evaluation. To determine which Miner is best suited, however, we first need to calculate the Miner score based upon the consensus results, as shown in Table C.5. Once again, the results are in fraction form, this time with M_1 and M_3 possessing the highest score.

	M_1	M_2	M_3	M_4
M_1	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
M_2	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{3}{6}$	$\frac{2}{4}$
M_3	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
M_4	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$

Table C.5: Calculation of the "Miner Score", allowing the identification of the most efficient Miner "validator", possessing the highest ratio of analysed to validated Miners. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

Score and Block Distribution Determination

The final step is the calculation of the overall consensus score in Table C.6. As we can see, the results for the routing and mining scores have been recovered from Table C.3 and Table C.5 respectively. Furthermore, as stated previously, since no malicious nodes are operating in this scenario, the reputation value is equal to 1 as we assume that all Miners have previously shown valid behaviour. As a result, we can see that two Miners

possess the highest score, meaning that both M1 and M3 are equally efficient. To resolve this deadlock, each Miner sets a random timer, thus allowing the first to wake to insert their block into the blockchain. Upon receipt of this inserted block, the other Miner will cancel its timeout and return to normal function.

Table C.6: Calculation of the "Consensus Score", a concatenation of the "Routing" and "Miner Scores", factoring in the reputation of the Miner in question to determine the most efficient for block insertion. The resulting most efficient Miners are highlighted in **blue**.

		M_1	M_2	M_3	M_4
M_1	**Routing**	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
	Mining	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
	Reputation	1	1	1	1
	Score	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$
M_2	**Routing**	$\frac{3}{6}$	$\frac{3}{6}$	$\frac{3}{6}$	$\frac{3}{6}$
	Mining	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$
	Reputation	1	1	1	1
	Score	1	1	1	1
M_3	**Routing**	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
	Mining	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$	$\frac{3}{4}$
	Reputation	1	1	1	1
	Score	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$	$\frac{17}{12} = 1.417$
M_4	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$	$\frac{2}{4}$
	Reputation	1	1	1	1
	Score	$\frac{4}{3} = 1.333$	$\frac{4}{3} = 1.333$	$\frac{4}{3} = 1.333$	$\frac{4}{3} = 1.333$

As determined, only one block is necessary to cover the entire analysis performed on this route. The transmission itself is presented in Analysis C.1.1.

Analysis C.1.1

$$M_1 \text{ or } M_3 = \begin{pmatrix} 1,2,3,4,5,6,7,8 \\ M_1, M_2, M_3, M_4 \end{pmatrix}$$

Table C.7: Representation of the total number of transmissions needed per validation phase with $TTL = 2$ and $TTL = 4$, as well as the included number of transmissions needed to broadcast, equal to the size of the network, here 10

Miners	**2-hop**	**4-hop**	**Total**
M_1	4	9	**13**
M_2	3	8	**11**
M_3	4	10	**14**
M_4	5	9	**14**
			52
Broadcast Size			**10**
Number of Insertions			**1**
			10
Total Transmissions			**62**

The final analysis we can perform is regarding the overall number of transmissions. Each consensus stage possesses its own transmission phase, followed by the transmission of the final block. Each phase also possesses its own number of hops necessary to function at peak efficiency. Table C.7 shows the overview of the number of transmissions based upon the number of hops. We can see that the routing validation phase uses 2-hops whereas the Miner validation phase was performed using 4-hops in total. Thanks to this network being so small, the number of hops is relatively low. Finally, we must add the number of transmissions corresponding to the broadcasting of the final block throughout the network. Since only one block is sufficient, the number of transmissions is limited to the number of nodes in the network, in this case 10. As a result, the total number of transmissions for this topology reaches 62, approximately 1.7 times higher when compared to the reputation consensus mechanism[3], presented in Appendix B.1.

3: ≈ 35

C.2 Scenario *B*

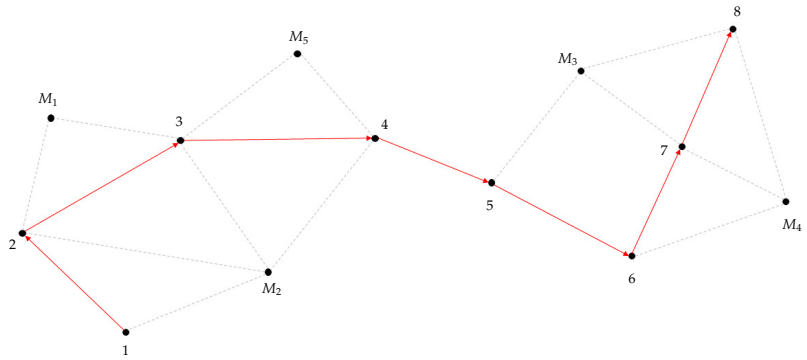


Figure C.2: 13-Node network topology with an eight-node route and five Miners. The network itself is semi-separated, blocking Miners from communicating between both sides

The next topology is presented in Figure C.2, possessing an eight-node route with five Miners, previously utilised in [176]. In comparaison with Scenario *A*, this topology possesses a distinct separation between two subsets of Miners. As determined in Appendix B.2, this separation causes some issues during validation, where each subset cannot interact with the other as they are out of 2-hop range. However, with the addition of a 4-hop range consensus, this separation is reduced, granting the capability of all miners to learn about their 4-hop neighbours, and thus reach consensus.

[176]: Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450

2-Hop Routing Consensus

Once again, we begin by performing the routing validation, the results of which are presented in Table C.8. As stated, due to the network's topology there are a lot more Miners which are out of 2-hop range of each other, reducing the capabilities of the network to reach consensus during this phase. Furthermore, there are three nodes which cannot be validated via consensus, node 1, 5 and 6 which can only be confirmed by their respective Miners M2, M 3 and M4. As a result, these Miners position themselves as the ultimate authority of these nodes. We can also see that, as is visible in the topology, the two sides of the network do not mix, which influences how the second stage Miner validation progresses.

	M_1	M_2	M_3	M_4	M_5
M_1	2, 3	2, 3			2, 3
M_2	1, 2, 3, 4	1, 2, 3, 4			1, 2, 3, 4
M_3			5, 7, 8	5, 7, 8	
M_4			6, 7, 8	6, 7, 8	
M_5	3, 4	3, 4			3, 4
Validated	**2, 3, 4**	**1, 2, 3, 4**	**5, 7, 8**	**6, 7, 8**	**2, 3, 4**

Table C.8: Consensus matrix representing the different blocks received up to 2-hops, allowing the identification of all validated and un validated nodes

4-Hop Miner Consensus

Routing Validation

Table C.9 shows the first step of the Miner validation, concerning the validation of the routing activities. We can see here that once more, it

is not possible for all the Miners to communicate together, here M1 and M4. Furthermore, as M4 is the authority on node 6, we can see that M 1 never receives the concerned validation, omitting it from its concatenated block, whereas the other four miners possess all eight routing nodes. This allows us to evaluate the score calculation in conditions where not all routing nodes are known or can be validated.

Table C.9: Validation matrix representing the distribution of confirmed routing information up to 4-hops, allowing the concatenation of all validated nodes

	M_1	M_2	M_3	M_4	M_5
M_1	2, 3, 4	2, 3, 4	2, 3, 4		2, 3, 4
M_2	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
M_3	5, 7, 8	5, 7, 8	5, 7, 8	5, 7, 8	5, 7, 8
M_4		6, 7, 8	6, 7, 8	6, 7, 8	6, 7, 8
M_5	2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4
Concatenated	**1, 2, 3, 4, 5, 7, 8**	**1, 2, 3, 4, 5, 6, 7, 8**	**1, 2, 3, 4, 5, 6, 7, 8**	**1, 2, 3, 4, 5, 6, 7, 8**	**1, 2, 3, 4, 5, 6, 7, 8**

Now armed with the list of concatenated nodes for validation, each Miner can compute the scores for routing efficiency, shown in Table C.10. Naturally, Miners M 1 and M4 cannot calculate the scores of each other since no blocks have been received. Furthermore, since M1 is not aware of 6's existence, the score is calculated with one less node in mind. However, as we can see in this context the result remains the same with M2 being identified as the most efficient Miner for the routing validation.

Table C.10: Calculation of the "Routing Score", allowing the identification of the most efficient routing Miner, possessing the highest ratio of mined to confirmed nodes. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

	M_1	M_2	M_3	M_4	M_5
M_1	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$		$\frac{3}{8}$
M_2	$\frac{4}{7}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{4}{8}$
M_3	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
M_4		$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
M_5	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{5}{6}$

Miner Consensus and Validation

Table C.11 shows the computation and results of the Miner validation process. We can see that overall, each miner is able to reach a consensus on different Miner actions. We can once more see that M3 and M4, similarly to previously when they were the authority over specific routing nodes, here they are the authority over each other, meaning that no other Miners were in 2-hop range of them during the routing validation phase, which is confirmed by Table C.8. As a result, only they possess the other in their list of validated Miners.

Table C.11: Consensus matrix associated the Miners which have been validated with that which performed the validation, based upon the routing routing validation matrix in Table C.9

	M_1	M_2	M_3	M_4	M_5
M_1	M_2, M_5	M_2, M_5	M_2, M_5		M_2, M_5
M_2	M_1, M_5	M_1, M_5	M_1, M_5	M_1, M_5	M_1, M_5
M_3	M_4	M_4	M_4	M_4	M_4
M_4		M_3	M_3	M_3	M_3
M_5	M_1, M_2	M_1, M_2	M_1, M_2	M_1, M_2	M_1, M_2
Validated	M_1, M_2, M_5	M_1, M_2, M_5	M_1, M_2, M_4, M_5	M_1, M_3	M_1, M_2, M_5

The resulting scores of the Miner validation are calculated in Table C.12. Since each Miner has validated a different number of Miners during consensus, the determined scores vary. However, one element remains, that M1, M 2 and M5 all possess the same high score, irrelevant of the calculating Miner. That being said, M4 is the only exception to this rule.

Indeed, since it is out of range of M 1, it is impossible to determine its efficiency score. Furthermore, in this case since it was able to validate only two Miners, it was able to give itself a high score as well during this phase. Unfortunately, this is not the case for all other Miners, where since M4 as well as M3 only validated each other. This means that, from the point of view of the other Miners, their blocks possess 0 efficiency and are not even considered for distribution.

	M_1	M_2	M_3	M_4	M_5
M_1	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$		$\frac{2}{3}$
M_2	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{1}{2}$	$\frac{2}{3}$
M_3	0	0	$\frac{1}{4}$	0	0
M_4		0	0	$\frac{1}{2}$	0
M_5	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{1}{2}$	$\frac{2}{3}$

Table C.12: Calculation of the "Miner Score", allowing the identification of the most efficient Miner "validator", possessing the highest ratio of analysed to validated Miners. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

Score and Block Distribution Determination

Due to the specificities of the network's topology, the overall consensus score in Table C.13 is a lot more complex than previously. That being said, even with the difficulties and challenges of this network, we can still see that all five Miners determine the same result that M2 is the most efficient Miner in the network and shall transmit its block containing all received results.

		M_1	M_2	M_3	M_4	M_5
M_1	**Routing**	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$		$\frac{3}{8}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$		$\frac{2}{3}$
	Reputation	1	1	1	1	1
	Score	$\frac{23}{21} = 1.095$	$\frac{25}{24} = 1.042$	$\frac{7}{8} = 0.875$	0	$\frac{25}{24} = 1.042$
M_2	**Routing**	$\frac{4}{7}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{4}{8}$	$\frac{4}{8}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{1}{2}$	$\frac{2}{3}$
	Reputation	1	1	1	1	1
	Score	$\frac{26}{21} = 1.238$	$\frac{7}{6} = 1.167$	1	1	$\frac{7}{6} = 1.167$
M_3	**Routing**	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
	Mining	0	0	$\frac{1}{4}$	0	0
	Reputation	1	1	1	1	1
	Score	$\frac{3}{7} = 0.429$	$\frac{3}{8} = 0.375$	$\frac{5}{8} = 0.625$	$\frac{3}{8} = 0.375$	$\frac{3}{8} = 0.375$
M_4	**Routing**		$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
	Mining		0	0	$\frac{1}{2}$	0
	Reputation	1	1	1	1	1
	Score	0	$\frac{3}{8} = 0.375$	$\frac{3}{8} = 0.375$	$\frac{7}{8} = 0.875$	$\frac{3}{8} = 0.375$
M_5	**Routing**	$\frac{3}{7}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$	$\frac{3}{8}$
	Mining	$\frac{2}{3}$	$\frac{2}{3}$	$\frac{2}{4}$	$\frac{1}{2}$	$\frac{2}{3}$
	Reputation	1	1	1	1	1
	Score	$\frac{23}{21} = 1.095$	$\frac{25}{24} = 1.042$	$\frac{7}{8} = 0.875$	$\frac{7}{8} = 0.875$	$\frac{25}{24} = 1.042$

Table C.13: Calculation of the "Consensus Score", a concatenation of the "Routing" and "Miner Scores", factoring in the reputation of the Miner in question to determine the most efficient for block insertion. The resulting most efficient Miners are highlighted in **blue**.

Secondary Consensus

Unfortunately, since M2 cannot confirm nor validate all the Miners, another consensus stage must be performed on the missing elements. To

determine which Miners are to insert their blocks into the blockchain, they can all perform once more Miner consensus matrix and score calculation, as presented in Table C.14 and Table C.15. As we can see, M 2 has been excluded from this second consensus as it has already transmitted its block to the network. Furthermore, we can see that no actual consensus can be reached regarding M 3 and M4 since, as already explained, no other nodes were able to validate their behaviour. Since all routing nodes have been inserted into the blockchain, no further analysis on the routing activities needs to be performed, so the results of the Miner validation are used on their own. As we can see, only M3 and M4 are capable of validating their own blocks, so they prepare them and transmit at random when ready, thus completing the validation of all Miners.

Table C.14: Secondary Consensus matrix associated the Miners which could not be validated and thus, couldn't be included in the primary validation phase

	M_1	M_2	M_3	M_4	M_5
M_1					
M_2					
M_3	M_4		M_4	M_4	M_4
M_4			M_3	M_3	M_3
M_5					
Validated	∅	∅	M_4	M_3	∅

Table C.15: Secondary Calculation of the "Miner Score", allowing to determine which Miners should insert the Miners which couldn't be validated during the primary validation phase

	M_1	M_2	M_3	M_4	M_5
M_1					
M_2					
M_3	0		1	0	0
M_4				1	0
M_5					

In total three block insertions took place to cover the entire routing validation process and Miner validation. These blocks are presented in Analysis C.2.1:

Analysis C.2.1

$$M_2 = \begin{pmatrix} 1,2,3,4,5,6,7,8 \\ M_1, M_2, M_5 \end{pmatrix}$$
$$M_3 = \begin{pmatrix} \varnothing \\ M_4 \end{pmatrix}$$
$$M_2 = \begin{pmatrix} \varnothing \\ M_3 \end{pmatrix}$$

Table C.16: Representation of the total number of transmissions needed per validation phase with $TTL = 2$ and $TTL = 4$, as well as the included number of transmissions needed to broadcast, equal to the size of the network, here 13

Miners	2-hop	4-hop	Total
M_1	3	8	**11**
M_2	5	10	**15**
M_3	4	10	**14**
M_4	4	7	**11**
M_5	3	10	**13**
			64
Broadcast Size			**13**
Number of Insertions			**3**
			39
Total Transmissions			**103**

To finalise this scenario, we can look once more at the overall number of transmissions needed. Table C.16 possesses the number of transmissions for the two consensus stages with 2-hop and 4-hop messages. Furthermore, since we know that three blocks were inserted, the number of transmissions is three time the number of nodes in the network, in this case 13, taking the block transmissions up to 39. In total, the number of transmissions amounts to 103, increasing the number of transmissions by a factor of 1.9 compared to the reputation consensus[4] in Appendix B.2.

4: ≈ 54

C.3 Scenario *C*

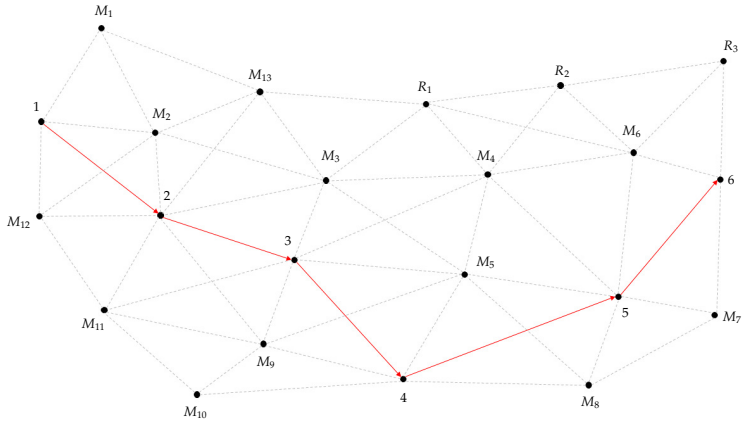


Figure C.3: 22-Node network topology with a six-node route and 13 Miners. The network is extremely dense and contains three relay nodes for dissemination

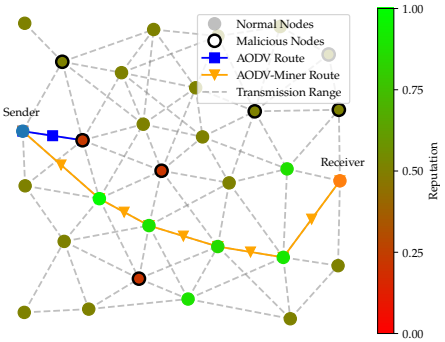


Figure C.4: Visualisation of route reputation after 15 mins. with *AODV-Miner* and AODV in a network of 30 nodes, 25% of which are malicious, previously presented in Figure 4.25

This final topology is a subnet extracted from one of the 100 30-node networks topologies used in the simulations in Chapter 4.4.2, visible in Figure C.4. The selected route for our scenario was one of the three routes selected by AODV-Miner during the simulations and possesses six nodes, numbered from 1 to 6. Thanks to the density, a total of 13 Miners are present, numbered from M1 to M13. Furthermore, contrary to the previous networks, this topology also has the presence of three relay nodes, which are instrumental in the dissemination of the different blocks by the Miners. This theoretical analysis will allow us to test on a real topology which will be used in subsequent simulations. Furthermore, its high density will also allow us to test how the Miners interact with each other during the various consensus stages.

Due to its size and complexity, it is impossible to accurately determine how the network would react using the reputation model from Appendix B. Since this version is dependent on which Miner wakes up first, plus which miner will respond to the initial block if any, we can approximate that with 13 Miners, we could potentially see more than 100 scenarios, an extremely complex task to perform by hand.

2-Hop Routing Consensus

The first stage of the quarantine byzantine consensus is the routing validation. The routing consensus matrix is presented in Table C.17 where each column represents the view of each Miner and the rows the different blocks which have been received. As we can see, due to the large size of the network, the corresponding matrix has become a lot more complex. With a total of 13 different Miners, each surveying at most three nodes, we have a much larger base for analysis. That being said, the main advantage of so many Miners is the potential for a complete consensus operation, allowing us to validate as many node activities as possible. However, we can see that due to the size of the network, not all Miners are able to communicate with each other with a 2-hop range. Oh the other hand, we can see that there are only a few instances of nodes not being able to be validated by Miners. For instance, Miners M4 and

M5 are unable to validate the activities of node 1, but they are able to confirm those of five other nodes.

Table C.17: Consensus matrix representing the different blocks received up to 2-hops, allowing the identification of all validated and un validated nodes

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	1	1	1									1	1
M_2	1, 2	1, 2	1, 2	1 , 2	1 , 2				1, 2	1, 2	1, 2	1, 2	1, 2
M_3	2, 3	2, 3	2, 3	2, 3	2, 3	2, 3		2, 3	2, 3		2, 3	2, 3	2, 3
M_4		3, 5	3, 5	3, 5	3, 5	3, 5	3, 5	3, 5	3, 5		3, 5		3, 5
M_5		3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5	3, 4, 5		3, 4 , 5
M_6			5, 6	5, 6	5, 6	5, 6	5, 6	5, 6					5, 6
M_7				5, 6	5, 6	5, 6	5, 6	5, 6					
M_8			4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5	4, 5			
M_9		2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4			2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4	2, 3, 4	
M_{10}		4			4			4	4	4	4	4	
M_{11}		2, 3	2, 3	2, 3	2, 3				2, 3	2, 3	2, 3	2, 3	2, 3
M_{12}	1, 2	1, 2	1, 2						1, 2	1, 2	1, 2	1, 2	1, 2
M_{13}	2	2	2	2	2	2					2	2	2
Valid.	**1, 2**	**1, 2, 3, 4, 5**	**1, 2, 3, 4, 5**	**2, 3, 4, 5, 6**	**2, 3, 4, 5, 6**	**2, 3, 4, 5, 6**	**3, 4, 5, 6**	**2, 3, 4, 5, 6**	**1, 2, 3, 4, 5**	**1, 2, 3, 4, 5**	**1, 2, 3, 4, 5**	**1, 2, 3, 4**	**1, 2, 3, 5**

4-Hop Routing Consensus

Routing Validation

With the results of this first consensus stage, the Miners can create another block and transmit it at a 4-hop distance for the next phase, the Miner validation. First off, however, we take a look at the analysed routing activities from the first stage and perform the first step of our net analysis, the validation of the routing consensus and calculation of the routing score. The routing validation is presented in Table C.18. First off, we can see that M 7 is in a position where it cannot reach M1 or M13, which is visible in the topology with M7 isolated on the right and the other two Miners on the top left. Furthermore, although during consensus certain nodes actions couldn't be validated for certain Miners, the validation phase has allowed them all to confirm all 6 routing nodes in the network.

Thanks to this validation phase, we can confirm that all Miners possess a global view of the route and can insert all observed actions into the blockchain. However, before that they must compute the routing score based upon the results of the routing validation, the results of which are visible in Table C.19. As we can see, a substantial majority of Miners have received the same high score. This basically means that these Miners are in a position where they are easily reachable with regards to the concerned route. It also comes to no surprise that the three Miners which do not have a high score are the three which cannot inter communicate and, by observing the network topology, we can see that they are positioned on the far outskirts of the network, reducing their efficiency and overall reach.

Miner Consensus and Validation

To be able to determine the most efficient Miner(s) in the network, we first need to perform the Miner validation. This is presented in Table C.22

Table C.18: Validation matrix representing the distribution of confirmed routing information up to 4-hops, allowing the concatenation of all validated nodes

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	1, 2	1, 2	1, 2	1, 2	1, 2	1, 2		1, 2	1, 2	1, 2	1, 2	1, 2	1, 2
M_2	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
M_3	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
M_4	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6
M_5	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6
M_6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6
M_7		3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6	3, 4, 5, 6		3, 4, 5, 6
M_8	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6	2, 3, 4, 5, 6
M_9	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
M_{10}	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
M_{11}	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
M_{12}	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4		1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4	1, 2, 3, 4
M_{13}	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5	1, 2, 3, 4, 5
Concat.	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**	**1, 2, 3, 4, 5, 6**

where we can see the substantial matrix formed by the validation process. Indeed, due to the large number of Miners and the close proximity between them, many Miners are callable of validating multiple others. This is easily perceivable not only by the size of the matrix, but also the results in the validation row where we can see that all Miners are capable of validating every Miner for this route.

To grasp a better understanding of the complexity and efficiency of this matrix, we can calculate the resulting Miner Score, visible in Table C.20. As we can see, compared to the previous routing scores in Table C.19, only two miners have received the highest score. As for the rest of the Miners, the scores are evenly distributed amongst them, all the while with M1 and M7 still bringing up the rear. One major difference, however, is the score increase on M12, which was in a position of weakness in regards to routing validation but is in close proximity to multiple Miners. This brings to light the importance of utilising both the routing score and mining score, as a Miner could be in an isolated position regarding the route itself, but easily inter connected with the other Miners.

Score and Block Distribution Determination

Now armed with the overall scores, we can compute the overall efficiency factor of the Miners relative to both their routing and mining activities. The results of this are presented in Table C.23.

As we can see, two miners have been identified as the most efficient: M 3 and M5. Coincidentally, these are the same Miners which received the highest mining score in Table C.20. Thanks to this consensus method, only one of these nodes need transmit a block as it is already aware of

Table C.19: Calculation of the "Routing Score", allowing the identification of the most efficient routing Miner, possessing the highest ratio of mined to confirmed nodes. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$		$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$
M_2	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_3	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_4	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_5	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_6	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_7		$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$		$\frac{5}{6}$
M_8	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_9	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_{10}	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_{11}	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
M_{12}	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$		$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
M_{13}	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$

Table C.20: Calculation of the "Miner Score", allowing the identification of the most efficient Miner "validator", possessing the highest ratio of analysed to validated Miners. The highest scores are identified with a **blue** background, emphasising the different levels of consensus.

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$		$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$
M_2	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$
M_3	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$
M_4	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$
M_5	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$
M_6	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$
M_7		$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$		$\frac{4}{13}$
M_8	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$
M_9	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$
M_{10}	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$
M_{11}	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$
M_{12}	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$		$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$
M_{13}	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$

Table C.21: Representation of the total number of transmissions needed per validation phase with $TTL = 2$ and $TTL = 4$, as well as the included number of transmissions needed to broadcast, equal to the size of the network, here 22

Miners	2-hop	4-hop	Total
M_1	4	15	19
M_2	7	19	26
M_3	8	22	30
M_4	8	22	30
M_5	8	22	30
M_6	7	20	27
M_7	4	15	19
M_8	5	19	24
M_9	7	21	28
M_{10}	4	18	22
M_{11}	6	19	25
M_{12}	5	15	20
M_{13}	6	21	27
			327
Broadcast Size			**22**
Number of Insertions			**1**
			22
Total Transmissions			**349**

all routing activities as well as the results of the Miner consensus. Thus, both M3 and M5 will arm a random timeout, after which the winner is the Miner which wakes up first and inserts its block into the blockchain before the other. The final block is presented in Analysis C.3.1.

Analysis C.3.1

$$M_3 \text{ or } M_5 = \begin{pmatrix} 1,2,3,4,5,6 \\ M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13} \end{pmatrix}$$

To complete this final scenario, we can calculate the overall number of transmissions needed for both block dissemination and the whole consensus process. Since only a single block is inserted, its cost is equal to the number of nodes in the network, here 22. Thanks to Table C.21, we can determine that in total 327 transmissions are needed to reach consensus on both the routing and the mining activities, bringing the total number of transmissions up to 349.

Table C.22: Consensus matrix associated the Miners which have been validated with that which performed the validation, based upon the routing routing validation matrix in Table C.18

	M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	×	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}	M_2, M_3, M_{12}, M_{13}
M_2	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_3, M_4, M_5, M_9, M_{10}, M_{11}, M_{12}, M_{13}$
M_3	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_4, M_5, M_6, M_8, M_9, M_{11}, M_{12}, M_{13}$
M_4	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$	$M_2, M_3, M_5, M_6, M_7, M_8, M_9, M_{11}, M_{13}$
M_5	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$	$M_2, M_3, M_4, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{13}$
M_6	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$	$M_3, M_4, M_5, M_7, M_8, M_{13}$
M_7	×	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	M_4, M_5, M_6, M_8	×	M_4, M_5, M_6, M_8
M_8	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$	$M_3, M_4, M_5, M_6, M_7, M_9, M_{10}$
M_9	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$	$M_2, M_3, M_4, M_5, M_8, M_{10}, M_{11}, M_{12}$

M_{10}	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$	$M_2, M_5, M_8, M_9, M_{11}, M_{12}$
M_{11}	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$	$M_2, M_3, M_4, M_5, M_9, M_{10}, M_{12}, M_{13}$
M_{12}	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	×	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$	$M_1, M_2, M_3, M_9, M_{10}, M_{11}, M_{13}$
M_{13}	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_{11}, M_{12}$
Valid.	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$	$M_1, M_2, M_3, M_4, M_5, M_6, M_7, M_8, M_9, M_{10}, M_{11}, M_{12}, M_{13}$

Table C.23: Calculation of the "Consensus Score", a concatenation of the "Routing" and "Miner Scores", factoring in the reputation of the Miner in question to determine the most efficient for block insertion. The resulting most efficient Miners are highlighted in **blue**

		M_1	M_2	M_3	M_4	M_5	M_6	M_7	M_8	M_9	M_{10}	M_{11}	M_{12}	M_{13}
M_1	**Routing**	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	×	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$	$\frac{2}{6}$
	Mining	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	×	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	0	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641	$\frac{25}{39} =$ 0.641
M_2	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1

	Score	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$
M_3	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$
M_4	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$	$\frac{9}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$	$\frac{119}{78} = 1.526$
M_5	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$	$\frac{10}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$	$\frac{125}{78} = 1.603$
M_6	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$
M_7	**Routing**	✗	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	✗	$\frac{5}{6}$
	Mining	✗	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	$\frac{4}{13}$	✗	$\frac{4}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	0	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	$\frac{38}{39} = 0.974$	0	$\frac{38}{39} = 0.974$
M_8	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$	$\frac{107}{78} = 1.372$

M_9	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$
M_{10}	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$	$\frac{6}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$	$\frac{111}{78} = 1.295$
M_{11}	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$
M_{12}	**Routing**	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	✕	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$	$\frac{4}{6}$
	Mining	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	✕	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$	$\frac{7}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$	$\frac{47}{39} = 1.205$
M_{13}	**Routing**	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$	$\frac{5}{6}$
	Mining	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$	$\frac{8}{13}$
	Reputation	1	1	1	1	1	1	1	1	1	1	1	1	1
	Score	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$	$\frac{113}{78} = 1.449$

Bibliography

[6] Mathy Vanhoef and Frank Piessens. 'Key Reinstallation Attacks: Forcing Nonce Reuse in WPA2'. In: *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. 2017. DOI: 10.1145/3133956.3134027 (cited on page 2).

[7] Rwan Mahmoud et al. 'Internet of things (IoT) security: Current status, challenges and prospective measures'. In: *2015 10th International Conference for Internet Technology and Secured Transactions (ICITST)*. 2015. DOI: 10.1109/ICITST.2015.7412116 (cited on pages 3, 19, 22, 23, 146).

[8] Wei Zhou et al. 'The Effect of IoT New Features on Security and Privacy: New Threats, Existing Solutions, and Challenges Yet to Be Solved'. In: *IEEE Internet of Things Journal* 6 (2019). DOI: 10.1109/JIOT.2018.2847733 (cited on page 3).

[10] Justin McCurry. 'South Korean nuclear operator hacked amid cyber-attack fears'. In: *The Guardian* (2014). (Visited on Sept. 25, 2022) (cited on page 5).

[11] Nicole Perlroth and Clifford Krauss. 'A cyber attack in Saudi Arabia failed to cause carnage, but the next attempt could be deadly'. In: *The Independent* (2018). (Visited on Sept. 25, 2022) (cited on page 5).

[12] Eleonora Viganò, Michele Loi, and Emad Yaghmaei. 'Cybersecurity of critical infrastructure'. In: *The Ethics of Cybersecurity*. Springer, Cham, 2020, pp. 157–177. DOI: 10.1007/978-3-030-29053-5_8 (cited on page 5).

[16] George Oikonomou et al. 'The Contiki-NG open source operating system for next generation IoT devices'. In: *SoftwareX* 18 (2022). DOI: 10.1016/j.softx.2022.101089 (cited on page 10).

[17] Fredrik Osterlind et al. 'Cross-Level Sensor Network Simulation with COOJA'. In: *Proceedings. 2006 31st IEEE Conference on Local Computer Networks*. 2006. DOI: 10.1109/LCN.2006.322172 (cited on page 10).

[18] Linyuan Zhang et al. 'Byzantine Attack and Defense in Cognitive Radio Networks: A Survey'. In: *IEEE Communications Surveys & Tutorials* 17 (2015). DOI: 10.1109/COMST.2015.2422735 (cited on pages 15, 16).

[19] Nicholas R. Rodofile, Kenneth Radke, and Ernest Foo. 'Framework for SCADA Cyber-Attack Dataset Creation'. In: *Proceedings of the Australasian Computer Science Week Multiconference*. 2017. DOI: 10.1145/3014812.3014883 (cited on page 16).

[20] HP. Sanghvi and MS. Dahiya. 'Cyber reconnaissance: an alarm before cyber attack'. In: *International Journal of Computer Applications* 63 (2013). DOI: 10.5120/10472-5202 (cited on page 16).

[23] N. Pontier, M. Orre, and M. Martin. 'Cyberattaque à l'hôpital de Dax : exposition des faits, conséquences et retour d'expérience'. In: *Cancer/Radiothérapie* 26 (2022). DOI: 10.1016/j.canrad.2022.06.011 (cited on page 16).

[25] Neda Afzaliseresht et al. 'From logs to Stories: Human-Centred Data Mining for Cyber Threat Intelligence'. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2966760 (cited on page 17).

[27] Jairo Giraldo et al. 'Security and Privacy in Cyber-Physical Systems: A Survey of Surveys'. In: *IEEE Design & Test* 34 (2017). DOI: 10.1109/MDAT.2017.2709310 (cited on pages 18, 20, 23).

[29] Jamal Raiyn. 'A survey of cyber attack detection strategies'. In: *International Journal of Security and Its Applications* 8 (2014). DOI: 10.14257/ijsia.2014.8.1.23 (cited on pages 18, 20, 22, 23).

[30] Ansam Khraisat et al. 'Survey of Intrusion Detection Systems: Techniques, Datasets and Challenges'. In: *Cybersecurity* 2 (2019). DOI: 10.1186/s42400-019-0038-7 (cited on pages 18, 23, 24, 38).

[31] Mohiuddin Ahmed. 'Intelligent Big Data Summarization for Rare Anomaly Detection'. In: *IEEE Access* 7 (2019). DOI: 10.1109/ACCESS.2019.2918364 (cited on pages 18, 23).

[32] Abebe Abeshu Diro and Naveen Chilamkurti. 'Distributed attack detection scheme using deep learning approach for Internet of Things'. In: *Future Generation Computer Systems* 82 (2018). DOI: 10.1016/j.future.2017.08.043 (cited on pages 18, 23).

[33] Deris Stiawan et al. 'Investigating Brute Force Attack Patterns in IoT Network'. In: *Journal of Electrical and Computer Engineering* 2019 (2019). DOI: 10.1155/2019/4568368 (cited on pages 18, 23).

[34] Ahmed Mahfouz, Deepak Venugopal, and Sajjan Shiva. 'Comparative Analysis of ML Classifiers for Network Intrusion Detection'. In: *Fourth International Congress on Information and Communication Technology*. 2020. DOI: 10.1007/978-981-32-9343-4_16 (cited on pages 18, 19, 23).

[35] Gurminder Kaur Jaideep Singh Simarpreet Kaur and Goldendeep Kaur. 'A Detailed Survey and Classification of Commonly Recurring Cyber Attacks'. In: *International Journal of Computer Applications* 141 (2016). DOI: 10.5120/ijca2016909811 (cited on pages 19, 23).

[36] Derui Ding et al. 'A survey on security control and attack detection for industrial cyber-physical systems'. In: *Neurocomputing* 275 (2018). DOI: 10.1016/j.neucom.2017.10.009 (cited on pages 19, 23).

[37] Magdi S. Mahmoud, Mutaz M. Hamdan, and Uthman A. Baroudi. 'Modeling and control of Cyber-Physical Systems subject to cyber attacks: A survey of recent advances and challenges'. In: *Neurocomputing* 338 (2019). DOI: 10.1016/j.neucom.2019.01.099 (cited on pages 19, 23).

[38] Lorena Cazorla, Cristina Alcaraz, and Javier Lopez. 'Cyber Stealth Attacks in Critical Information Infrastructures'. In: *IEEE Systems Journal* 12 (2018). DOI: 10.1109/JSYST.2015.2487684 (cited on pages 19, 20, 23).

[39] Jyoti Deogirikar and Amarsinh Vidhate. 'Security attacks in IoT: A survey'. In: *2017 International Conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC)*. 2017. DOI: 10.1109/I-SMAC.2017.8058363 (cited on pages 19, 23).

[40] Kahina Chelli. 'Security Issues in Wireless Sensor Networks: Attacks and Countermeasures'. In: *Proceedings of the World Congress on Engineering*. 2015. (Visited on Oct. 17, 2022) (cited on pages 19–21, 23, 38).

[41] Inria. *Cybersecurity: Current challenges and Inria's research directions*. Tech. rep. Inria, 2019. (Visited on Oct. 17, 2022) (cited on pages 19, 20, 23).

[42] Priyanka Goyal, Vinti Parmar, Rahul Rishi, et al. 'Manet: vulnerabilities, challenges, attacks, application'. In: *IJCEM International Journal of Computational Engineering & Management* 11 (2011). (Visited on Oct. 17, 2022) (cited on pages 19, 20, 23).

[43] Olivier Flauzac et al. 'SDN Based Architecture for IoT and Improvement of the Security'. In: *2015 IEEE 29th International Conference on Advanced Information Networking and Applications Workshops*. 2015. DOI: 10.1109/WAINA.2015.110 (cited on pages 19, 20, 23, 146).

[44] Furrakh Shahzad, Maruf Pasha, and Arslan Ahmad. 'A Survey of Active Attacks on Wireless Sensor Networks and their Countermeasures'. In: *CoRR* abs/1702.07136 (2017). DOI: 10.48550/arXiv.1702.07136 (cited on pages 19, 20, 23, 38).

[45] Khaleel Ahmad. 'Classification of Internet Security Attacks'. In: *Proceeding of the 5th National Conference INDIACom-2011Bharti Vidyapeeth's Institute of Computer Applications and Management, New Delhi ISSN*. 2011. (Visited on Oct. 17, 2022) (cited on pages 19–21, 23).

[46] Noah J. Apthorpe et al. 'Spying on the Smart Home: Privacy Attacks and Defenses on Encrypted IoT Traffic'. In: *CoRR* abs/1708.05044 (2017). DOI: 10.48550/arXiv.1708.05044 (cited on page 20).

[47] Katharina Krombholz et al. 'Advanced social engineering attacks'. In: *Journal of Information Security and Applications* 22 (2015). DOI: 10.1016/j.jisa.2014.09.005 (cited on page 20).

[48] Emma J. Williams, Joanne Hinds, and Adam N. Joinson. 'Exploring susceptibility to phishing in the workplace'. In: *International Journal of Human-Computer Studies* 120 (2018). DOI: 10.1016/j.ijhcs.2018.06.004 (cited on page 20).

[49] Akhil Gupta and Rakesh Kumar Jha. 'Security threats of wireless networks: A survey'. In: *International Conference on Computing, Communication & Automation*. 2015. DOI: 10.1109/CCAA.2015.7148407 (cited on pages 20, 21, 23).

[50] Glenn Surman. 'Understanding Security Using the OSI Model'. In: *SANS Institute Reading Room* (2002). (Visited on Oct. 17, 2022) (cited on page 21).

[51] Wenyuan Xu et al. 'Jamming sensor networks: attack and defense strategies'. In: *IEEE Network* 20 (2006). DOI: 10.1109/MNET.2006.1637931 (cited on page 21).

[52] Samad S. Kolahi, Kiattikul Treseangrat, and Bahman Sarrafpour. 'Analysis of UDP DDoS flood cyber attack and defense mechanisms on Web Server with Linux Ubuntu 13'. In: *2015 International Conference on Communications, Signal Processing, and their Applications (ICCSPA'15)*. 2015. DOI: 10.1109/ICCSPA.2015.7081286 (cited on page 21).

[53] Sagarika Chowdhury and Mainak Sen. 'Survey on Attacks on Wireless Body Area Network'. In: *International Journal of Computational Intelligence & IoT, Forthcoming* (2019). (Visited on Oct. 17, 2022) (cited on pages 21, 23, 38).

[54] I.F. Akyildiz et al. 'Wireless sensor networks: a survey'. In: *Computer Networks* 38 (2002). DOI: 10.1016/S1389-1286(01)00302-4 (cited on page 21).

[55] Latha Tamilselvan and V. Sankaranarayanan. 'Prevention of Blackhole Attack in MANET'. In: *The 2nd International Conference on Wireless Broadband and Ultra Wideband Communications (AusWireless 2007)*. 2007. DOI: 10.1109/AUSWIRELESS.2007.61 (cited on pages 21, 22).

[56] Viren Mahajan, Maitreya Natu, and Adarshpal Sethi. 'Analysis of wormhole intrusion attacks in MANETS'. In: *MILCOM 2008 - 2008 IEEE Military Communications Conference*. 2008. DOI: 10.1109/MILCOM.2008.4753176 (cited on pages 21, 22).

[57] V Shanmuganathan and T Anand. 'A survey on gray hole attack in manet'. In: *IRACST–International Journal of Computer Networks and Wireless Communications (IJCNWC)* 2 (2012). (Visited on Oct. 17, 2022) (cited on pages 22, 23).

[58] Emilie Bout, Valeria Loscri, and Antoine Gallais. 'How Machine Learning Changes the Nature of Cyberattacks on IoT Networks: A Survey'. In: *IEEE Communications Surveys & Tutorials* 24 (2022). DOI: 10.1109/COMST.2021.3127267 (cited on page 24).

[59] Emilie Bout, Valeria Loscri, and Antoine Gallais. 'HARPAGON: An Energy Management Framework for Attacks in IoT Networks'. In: *IEEE Internet of Things Journal* 9 (2022). DOI: 10.1109/JIOT.2022.3172849 (cited on page 24).

[60] Emilie Bout et al. 'FOLPETTI: A Novel Multi-Armed Bandit Smart Attack for Wireless Networks'. In: *Proceedings of the 17th International Conference on Availability, Reliability and Security*. 2022. DOI: 10.1145/3538969.3539001 (cited on page 24).

[61] Mingfu Xue et al. 'Machine Learning Security: Threats, Countermeasures, and Evaluations'. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.2987435 (cited on page 24).

[62] Mohiuddin Ahmed, Abdun Naser Mahmood, and Jiankun Hu. 'A survey of network anomaly detection techniques'. In: *Journal of Network and Computer Applications* 60 (2016). DOI: 10.1016/j.jnca.2015.11.016 (cited on pages 23, 26–29, 31, 38, 143).

[63] Mauro Conti, Nicola Dragoni, and Viktor Lesyk. 'A Survey of Man In The Middle Attacks'. In: *IEEE Communications Surveys & Tutorials* 18 (2016). DOI: 10.1109/COMST.2016.2548426 (cited on pages 23, 143).

[64] Eman Mousavinejad et al. 'A Novel Cyber Attack Detection Method in Networked Control Systems'. In: *IEEE Transactions on Cybernetics* 48 (2018). DOI: 10.1109/TCYB.2018.2843358 (cited on pages 23, 145).

[65] Ghada Elbez, Hubert B. Keller, and Veit Hagenmeyer. 'A New Classification of Attacks against the Cyber-Physical Security of Smart Grids'. In: *Proceedings of the 13th International Conference on Availability, Reliability and Security*. 2018. DOI: 10.1145/3230833.3234689 (cited on pages 23, 145).

[66] Minhaj Ahmad Khan and Khaled Salah. 'IoT security: Review, blockchain solutions, and open challenges'. In: *Future Generation Computer Systems* 82 (2018). DOI: 10.1016/j.future.2017.11.022 (cited on pages 23, 146).

[67] Mardiana binti Mohamad Noor and Wan Haslina Hassan. 'Current research on Internet of Things (IoT) security: A survey'. In: *Computer Networks* 148 (2019). DOI: 10.1016/j.comnet.2018.11.025 (cited on pages 23, 146).

[68] Othmane Blial, Mouad Ben Mamoun, and Redouane Benaini. 'An Overview on SDN Architectures with Multiple Controllers'. In: *Journal of Computer Networks and Communications* 2016 (2016). DOI: 10.1155/2016/9396525 (cited on pages 23, 146).

[69] Ayei E Ibor, Florence A Oladeji, and Olusoji B Okunoye. 'A Survey of Cyber Security Approaches for Attack Detection, Prediction, and Prevention'. In: *International Journal of Security and its Applications* 12 (2018). DOI: 10.14257/ijsia.2018.12.4.02 (cited on pages 25, 26, 38).

[70] Salvatore D'Antonio, Francesco Oliviero, and Roberto Setola. 'High-Speed Intrusion Detection in Support of Critical Infrastructure Protection'. In: *Critical Information Infrastructures Security*. Ed. by Javier Lopez. 2006. DOI: 10.1007/11962977_18 (cited on page 29).

[71] Justin M. Beaver, Raymond C. Borges-Hink, and Mark A. Buckner. 'An Evaluation of Machine Learning Methods to Detect Malicious SCADA Communications'. In: *2013 12th International Conference on Machine Learning and Applications*. 2013. DOI: 10.1109/ICMLA.2013.105 (cited on page 30).

[72] S. L. P. Yasakethu and J. Jiang. 'Intrusion Detection via Machine Learning for SCADA System Protection'. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013. DOI: 10.14236/ewic/ICSCSR2013.12 (cited on page 30).

[73] William Hurst, Madjid Merabti, and Paul Fergus. 'Big Data Analysis Techniques for Cyber-threat Detection in Critical Infrastructures'. In: *2014 28th International Conference on Advanced Information Networking and Applications Workshops*. 2014. DOI: 10.1109/WAINA.2014.141 (cited on page 30).

[74] Riccardo Taormina and Stefano Galelli. 'Deep-Learning Approach to the Detection and Localization of Cyber-Physical Attacks on Water Distribution Systems'. In: *Journal of Water Resources Planning and Management* 144 (2018). DOI: 10.1061/(ASCE)WR.1943-5452.0000983 (cited on page 30).

[75] Riccardo Taormina et al. 'Battle of the Attack Detection Algorithms: Disclosing Cyber Attacks on Water Distribution Networks'. In: *Journal of Water Resources Planning and Management* 144 (2018). DOI: 10.1061/(ASCE)WR.1943-5452.0000969 (cited on page 30).

[76] Nitin Naik et al. 'Lockout-Tagout Ransomware: A Detection Method for Ransomware using Fuzzy Hashing and Clustering'. In: *2019 IEEE Symposium Series on Computational Intelligence (SSCI)*. 2019. DOI: 10.1109/SSCI44817.2019.9003148 (cited on page 30).

[77] Dharminder Kumar and D Fet. 'Performance analysis of various data mining algorithms: A review'. In: *International Journal of Computer Applications* 32 (2011). (Visited on Oct. 17, 2022) (cited on page 30).

[78] Huaglory Tianfield. 'Data mining based cyber-attack detection'. In: *System simulation technology* 13 (2017). (Visited on Oct. 17, 2022) (cited on page 30).

[79] Misty Blowers and Jonathan Williams. 'Machine Learning Applied to Cyber Operations'. In: *Network Science and Cybersecurity*. Ed. by Robinson E. Pino. Springer New York, 2014, pp. 155–175. DOI: 10.1007/978-1-4614-7597-2_10 (cited on pages 30, 31).

[80] Imen Brahmi et al. 'Towards a Multiagent-Based Distributed Intrusion Detection System Using Data Mining Approaches'. In: *Agents and Data Mining Interaction*. Ed. by Longbing Cao et al. 2012. DOI: 10.1007/978-3-642-27609-5_12 (cited on pages 30, 31).

[81] Noam Ben-Asher and Cleotilde Gonzalez. 'Effects of cyber security knowledge on attack detection'. In: *Computers in Human Behavior* 48 (2015). DOI: 10.1016/j.chb.2015.01.039 (cited on pages 30, 31, 39).

[82] Dennis Hollingworth. 'Towards threat, attack, and vulnerability taxonomies'. In: *IFIP WG*. 2003. (Visited on Oct. 17, 2022) (cited on page 33).

[83] Luis Marinos, Adrian Belmonte, and Evangelos Rekleitis. 'ENISA's Threat Taxonomy'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. (Visited on Oct. 17, 2022) (cited on pages 33–35).

[84] Gideon Creech and Jiankun Hu. 'Generation of a new IDS test dataset: Time to retire the KDD collection'. In: *2013 IEEE Wireless Communications and Networking Conference (WCNC)*. 2013. DOI: 10.1109/WCNC.2013.6555301 (cited on page 32).

[85] Gideon Creech and Jiankun Hu. 'A Semantic Approach to Host-Based Intrusion Detection Systems Using Contiguousand Discontiguous System Call Patterns'. In: *IEEE Transactions on Computers* 63 (2014). DOI: 10.1109/TC.2013.13 (cited on page 32).

[86] Gideon Creech. 'Developing a high-accuracy cross platform Host-Based Intrusion Detection System capable of reliably detecting zero-day attacks'. PhD thesis. University of New South Wales. Engineering & Information Technology, 2014. DOI: 10.26190/unsworks/16615 (cited on page 32).

[88] Constantinos Kolias et al. 'Intrusion Detection in 802.11 Networks: Empirical Evaluation of Threats and a Public Dataset'. In: *IEEE Communications Surveys & Tutorials* 18 (2016). DOI: 10.1109/COMST.2015.2402161 (cited on page 32).

[90] José Jair Santanna et al. 'Booters — An analysis of DDoS-as-a-service attacks'. In: *2015 IFIP/IEEE International Symposium on Integrated Network Management (IM)*. 2015. DOI: 10.1109/INM.2015.7140298 (cited on page 32).

[91] Nickolaos Koroniotis et al. 'Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset'. In: *Future Generation Computer Systems* 100 (2019). DOI: 10.1016/j.future.2019.05.041 (cited on page 32).

[93] Elaheh Biglar Beigi et al. 'Towards effective feature selection in machine learning-based botnet detection approaches'. In: *2014 IEEE Conference on Communications and Network Security*. 2014. DOI: 10.1109/CNS.2014.6997492 (cited on page 32).

[97] Iman Sharafaldin et al. 'Developing Realistic Distributed Denial of Service (DDoS) Attack Dataset and Taxonomy'. In: *2019 International Carnahan Conference on Security Technology (ICCST)*. 2019. DOI: 10.1109/CCST.2019.8888419 (cited on page 32).

[98] Hossein Hadian Jazi et al. 'Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling'. In: *Computer Networks* 121 (2017). DOI: 10.1016/j.comnet.2017.03.018 (cited on page 32).

[100] Iman Sharafaldin, Arash Habibi Lashkari, and Ali A. Ghorbani. 'Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization'. In: *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP*. 2018. DOI: 10.5220/0006639801080116 (cited on page 32).

[102] Markus Ring et al. 'Flow-based benchmark data sets for intrusion detection'. In: *Proceedings of the 16th European Conference on Cyber Warfare and Security (ECCWS)*. ACPI, 2017, pp. 361–369. (Visited on Oct. 17, 2022) (cited on page 32).

[104] Markus Ring et al. 'Creation of flow-based data sets for intrusion detection'. In: *Journal of Information Warfare* 16 (2017). (Visited on Oct. 17, 2022) (cited on page 32).

[105] Benjamin Sangster et al. 'Toward Instrumenting Network Warfare Competitions to Generate Labeled Datasets'. In: *Proceedings of the 2nd Conference on Cyber Security Experimentation and Test*. 2009. (Visited on Oct. 17, 2022) (cited on page 32).

[107] S. García et al. 'An empirical comparison of botnet detection methods'. In: *Computers & Security* 45 (2014). DOI: 10.1016/j.cose.2014.05.011 (cited on page 32).

[109] Richard Lippmann et al. 'The 1999 DARPA off-line intrusion detection evaluation'. In: *Computer Networks* 34 (2000). DOI: 10.1016/S1389-1286(00)00139-0 (cited on page 32).

[111] Iñigo Perona et al. 'Service-Independent Payload Analysis to Improve Intrusion Detection in Network Traffic'. In: *Proceedings of the 7th Australasian Data Mining Conference - Volume 87*. 2008. (Visited on Oct. 17, 2022) (cited on page 32).

[112] Santosh Kumar Sahu, Sauravranjan Sarangi, and Sanjaya Kumar Jena. 'A detail analysis on intrusion detection datasets'. In: *2014 IEEE International Advance Computing Conference (IACC)*. 2014. DOI: 10.1109/IAdCC.2014.6779523 (cited on page 32).

[114] Richard Zuech et al. 'A New Intrusion Detection Benchmarking System'. In: *The Twenty-Eighth International FLAIRS Conference*. 2015. (Visited on Oct. 17, 2022) (cited on page 32).

[115] Ali Shiravi et al. 'Toward developing a systematic approach to generate benchmark datasets for intrusion detection'. In: *Computers & Security* 31 (2012). DOI: 10.1016/j.cose.2011.12.012 (cited on page 32).

[117] Sherif Saad et al. 'Detecting P2P botnets through network behavior analysis and machine learning'. In: *2011 Ninth Annual International Conference on Privacy, Security and Trust*. 2011. DOI: 10.1109/PST.2011.5971980 (cited on page 32).

[119] Mahbod Tavallaee et al. 'A detailed analysis of the KDD CUP 99 data set'. In: *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*. 2009. DOI: 10.1109/CISDA.2009.5356528 (cited on page 32).

[120] Sarika Choudhary and Nishtha Kesswani. 'Analysis of KDD-Cup'99, NSL-KDD and UNSW-NB15 Datasets using Deep Learning in IoT'. In: *Procedia Computer Science* 167 (2020). DOI: 10.1016/j.procs.2020.03.367 (cited on page 32).

[122] Alexander D. Kent. 'Cyber security data sources for dynamic network research'. In: *Dynamic Networks and Cyber-Security*. World Scientific, 2016. Chap. Chapter 2, pp. 37–65. DOI: 10.1142/9781786340757_0002 (cited on page 32).

[124] Jungsuk Song et al. 'Statistical Analysis of Honeypot Data and Building of Kyoto 2006+ Dataset for NIDS Evaluation'. In: *Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security*. 2011. DOI: 10.1145/1978672.1978676 (cited on page 32).

[126] Ruoming Pang et al. 'A First Look at Modern Enterprise Traffic'. In: *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*. 2005. DOI: 10.1145/1330107.1330110 (cited on page 32).

[128] Frank Beer et al. 'A new Attack Composition for Network Security'. In: *10. DFN-Forum Kommunikationstechnologien*. 2017. (Visited on Oct. 17, 2022) (cited on page 32).

[130] W. Haider et al. 'Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling'. In: *Journal of Network and Computer Applications* 87 (2017). DOI: 10.1016/j.jnca.2017.03.018 (cited on page 32).

[132] Raman Singh, Harish Kumar, and RK Singla. 'A reference dataset for network traffic activity based intrusion detection system'. In: *International Journal of Computers Communications & Control* 10 (2015). DOI: 10.15837/ijccc.2015.3.1924 (cited on page 32).

[133] Rohini Sharma, R.K. Singla, and Ajay Guleria. 'A New Labeled Flow-based DNS Dataset for Anomaly Detection: PUF Dataset'. In: *Procedia Computer Science* 132 (2018). DOI: 10.1016/j.procs.2018.05.079 (cited on page 32).

[134] Charles Wheelus et al. 'A Session Based Approach for Aggregating Network Traffic Data – The SANTA Dataset'. In: *2014 IEEE International Conference on Bioinformatics and Bioengineering*. 2014. DOI: 10.1109/BIBE.2014.72 (cited on page 32).

[135] A.R. Vasudevan, E. Harshini, and S. Selvakumar. 'SSENet-2011: A Network Intrusion Detection System dataset and its comparison with KDD CUP 99 dataset'. In: *2011 Second Asian Himalayas International Conference on Internet (AH-ICI)*. 2011. DOI: 10.1109/AHICI.2011.6113948 (cited on page 32).

[136] Sangeeta Bhattacharya and S. Selvakumar. 'SSENet-2014 Dataset: A Dataset for Detection of Multi-connection Attacks'. In: *2014 3rd International Conference on Eco-friendly Computing and Communication Systems*. 2014. DOI: 10.1109/Eco-friendly.2014.100 (cited on page 32).

[137] Rick Hofstede et al. 'SSH Compromise Detection Using NetFlow/IPFIX'. In: *SIGCOMM Computer Communication Review* 44 (2014). DOI: 10.1145/2677046.2677050 (cited on page 32).

[139] Abdullah Alsaedi et al. 'TON_IoT Telemetry Dataset: A New Generation Dataset of IoT and IIoT for Data-Driven Intrusion Detection Systems'. In: *IEEE Access* 8 (2020). DOI: 10.1109/ACCESS.2020.3022862 (cited on page 32).

[141] Eduardo K. Viegas, Altair O. Santin, and Luiz S. Oliveira. 'Toward a reliable anomaly-based intrusion detection in real-world environments'. In: *Computer Networks* 127 (2017). DOI: 10.1016/j.comnet.2017.08.013 (cited on page 32).

[143] Monowar H. Bhuyan, Dhruba Kumar Bhattacharyya, and Jugal Kumar Kalita. 'Towards Generating Real-life Datasets for Network Intrusion Detection'. In: *International Journal of Network Security* 17 (2015). (Visited on Oct. 17, 2022) (cited on page 32).

[144] Anna Sperotto et al. 'A Labeled Data Set for Flow-Based Intrusion Detection'. In: *IP Operations and Management*. Ed. by Giorgio Nunzi, Caterina Scoglio, and Xing Li. 2009. DOI: 10.1007/978-3-642-04968-2_4 (cited on page 32).

[146] Gabriel Maciá-Fernández et al. 'UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs'. In: *Computers & Security* 73 (2018). DOI: 10.1016/j.cose.2017.11.004 (cited on page 32).

[148] F. Gringoli et al. 'GT: Picking up the Truth from the Ground for Internet Traffic'. In: *SIGCOMM Computer Communication Review* 39 (2009). DOI: 10.1145/1629607.1629610 (cited on page 32).

[150] Melissa JM Turcotte, Alexander D Kent, and Curtis Hash. 'Unified Host and Network Data Set'. In: *Data Science for Cyber-Security*. World Scientific, 2019. Chap. Chapter 1, pp. 1–22. DOI: 10.1142/9781786345646_001 (cited on page 32).

[152] Nour Moustafa and Jill Slay. 'UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)'. In: *2015 Military Communications and Information Systems Conference (MilCIS)*. 2015. DOI: 10.1109/MilCIS.2015.7348942 (cited on page 32).

[154] Cedric Lévy-Bencheton and Eleni Darra. 'Cyber Security and Resilience of Intelligent Public Transport: Good practices and recommendations'. In: *European Union Agency for Network and Information Security (ENISA)*. 2016. (Visited on Oct. 17, 2022) (cited on page 33).

[155] National Institute of Standards and Technology (NIST). 'NIST CSRC Taxonomy'. In: *NIST Special Publication (SP) 800-30, Revision 1, Guide for Conducting Risk Assessments*. 2020. DOI: 10.6028/NIST.SP.800-30r1 (cited on pages 33, 34).

[156] ETSI. 'CYBER; Global Cyber Security Ecosystem'. In: *European Telecommunications Standard Institute (ETSI)*. 2020. (Visited on Oct. 17, 2022) (cited on page 33).

[157] James Cebula, Mary Popeck, and Lisa Young. *A Taxonomy of Operational Cyber Security Risks Version 2*. Tech. rep. CMU/SEI-2014-TN-006. Software Engineering Institute, Carnegie Mellon University, 2014. (Visited on Oct. 17, 2022) (cited on page 34).

[159] Steven Launius. 'Evaluation of Comprehensive Taxonomies for Information Technology Threats'. In: *SANS Institute* 1 (2018). (Visited on Oct. 17, 2022) (cited on page 34).

[160] Paul Cichonski et al. 'Computer Security Incident Handling Guide'. In: *NIST Special Publication* 800 (2004). (Visited on Oct. 17, 2022) (cited on page 36).

[161] Nurul Hidayah Ab Rahman and Kim-Kwang Raymond Choo. 'A survey of information security incident handling in the cloud'. In: *Computers & Security* 49 (2015). DOI: 10.1016/j.cose.2014.11.006 (cited on pages 36, 38, 40).

[163] Tina Wu et al. 'Towards a SCADA forensics architecture'. In: *1st International Symposium for ICS & SCADA Cyber Security Research 2013 (ICS-CSR 2013) 1*. 2013. DOI: 10.14236/ewic/ICSCSR2013.2 (cited on pages 38, 41).

[166] Petar Radanliev et al. 'Cyber Risk Management for the Internet of Things'. In: *Preprints 2019*. 2019. DOI: 10.20944/preprints201904.0133.v1 (cited on page 39).

[171] Fenye Bao et al. '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 (2012). DOI: 10.1109/TCOMM.2012.031912.110179 (cited on page 46).

[172] Deep Kumar Bangotra et al. 'A Trust Based Secure Intelligent Opportunistic Routing Protocol for Wireless Sensor Networks'. In: *Wireless Personal Communications* (2021). DOI: 10.1007/s11277-021-08564-3 (cited on page 46).

[173] Nabil Djedjig et al. 'Trust-aware and cooperative routing protocol for IoT security'. In: *Journal of Information Security and Applications* 52 (2020). DOI: 10.1016/j.jisa.2020.102467 (cited on page 47).

[174] Jiawei Tang et al. 'An aggregate signature based trust routing for data gathering in sensor networks'. In: *Security and Communication Networks* 2018 (2018). DOI: 10.1155/2018/6328504 (cited on page 47).

[175] Weidong Fang et al. 'Trust management-based and energy efficient hierarchical routing protocol in wireless sensor networks'. In: *Digital Communications and Networks* 7 (2021). DOI: 10.1016/j.dcan.2021.03.005 (cited on page 47).

[176] Maqsood Ahamed Abdul Careem and Aveek Dutta. 'Reputation based Routing in MANET using Blockchain'. In: *2020 International Conference on COMmunication Systems & NETworkS (COMSNETS)*. 2020. DOI: 10.1109/COMSNETS48256.2020.9027450 (cited on pages 47, 50, 71, 75, 158, 171).

[178] Andreas M Antonopoulos. *Mastering Bitcoin: Programming the open blockchain*. "O'Reilly Media, Inc.", 2017. (Visited on Oct. 17, 2022) (cited on page 47).

[179] NARA. *Blockchain White Paper*. Tech. rep. National Archives and Records Administration, 2019. (Visited on Oct. 17, 2022) (cited on page 47).

[182] Xiaoqi Li et al. 'A survey on the security of blockchain systems'. In: *Future Generation Computer Systems* 107 (2020). DOI: 10.1016/j.future.2017.08.020 (cited on page 48).

[183] Muhammad Salek Ali et al. 'Applications of Blockchains in the Internet of Things: A Comprehensive Survey'. In: *IEEE Communications Surveys & Tutorials* 21 (2019). DOI: 10.1109/COMST.2018.2886932 (cited on page 48).

[184] Axel Moinet, Benoît Darties, and Jean-Luc Baril. 'Blockchain based trust & authentication for decentralized sensor networks'. In: *arXiv preprint arXiv:1706.01730* abs/1706.01730 (2017). DOI: 10.48550/arXiv.1706.01730 (cited on page 48).

[185] Yu Zeng et al. '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. DOI: 10.1109/MSN.2018.00009 (cited on page 48).

[186] Caciano Machado and Carla Merkle Westphall. 'Blockchain incentivized data forwarding in MANETs: Strategies and challenges'. In: *Ad Hoc Networks* 110 (2021). DOI: 10.1016/j.adhoc.2020.102321 (cited on page 48).

[187] Jidian Yang et al. 'A Trusted Routing Scheme Using Blockchain and Reinforcement Learning for Wireless Sensor Networks'. In: *Sensors* 19 (2019). DOI: 10.3390/s19040970 (cited on page 48).

[188] Hilmi Lazrag et al. 'A Blockchain-Based Approach for Optimal and Secure Routing in Wireless Sensor Networks and IoT'. In: *2019 15th International Conference on Signal-Image Technology & Internet-Based Systems (SITIS)*. 2019. DOI: 10.1109/SITIS.2019.00072 (cited on pages 48, 49).

[189] Jian Wang et al. 'Lightweight blockchain assisted secure routing of swarm UAS networking'. In: *Computer Communications* 165 (2021). DOI: 10.1016/j.comcom.2020.11.008 (cited on pages 48, 49).

[190] Gholamreza Ramezan and Cyril Leung. 'A Blockchain-Based Contractual Routing Protocol for the Internet of Things Using Smart Contracts'. In: *Wireless Communications and Mobile Computing* 2018 (2018). DOI: 10.1155/2018/4029591 (cited on pages 48, 49).

[192] Eiman Alotaibi and Biswanath Mukherjee. 'A survey on routing algorithms for wireless Ad-Hoc and mesh networks'. In: *Computer Networks* 56 (2012). DOI: 10.1016/j.comnet.2011.10.011 (cited on page 66).

[193] Thomas H. Clausen and Philippe Jacquet. *Optimized Link State Routing Protocol (OLSR)*. RFC 3626. 2003. DOI: 10.17487/RFC3626 (cited on page 67).

[194] Roger Alexander et al. *RPL: IPv6 Routing Protocol for Low-Power and Lossy Networks*. RFC 6550. 2012. DOI: 10.17487/RFC6550 (cited on page 67).

[195] Samir R. Das, Charles E. Perkins, and Elizabeth M. Belding-Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing*. RFC 3561. 2003. DOI: 10.17487/RFC3561 (cited on page 68).

[196] Charles E. Perkins and Elizabeth M. Royer. *Ad hoc On-Demand Distance Vector (AODV) Routing for IP version 6*. Internet-Draft draft-perkins-aodv6-01. Work in Progress. Internet Engineering Task Force, 2001. 8 pp. (Visited on Oct. 17, 2022) (cited on pages 69, 87).

[197] Yih-Chun Hu, Dave A. Maltz, and David B. Johnson. *The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4*. RFC 4728. 2007. DOI: 10.17487/RFC4728 (cited on pages 69, 80, 87).

[198] Jon Postel. *Internet Protocol*. RFC 791. 1981. DOI: 10.17487/RFC0791 (cited on pages 69, 199).

[199] Bob Hinden and Dr. Steve E. Deering. *Internet Protocol, Version 6 (IPv6) Specification*. RFC 2460. 1998. DOI: 10.17487/RFC2460 (cited on pages 70, 199).

[200] Baruch Awerbuch, David Holmer, and Herbert Rubens. 'High Throughput Route Selection in Multi-rate Ad Hoc Wireless Networks'. In: *Wireless On-Demand Network Systems*. Ed. by Roberto Battiti, Marco Conti, and Renato Lo Cigno. 2004. DOI: 10.1007/978-3-540-24614-5_19 (cited on page 70).

[201] Fernando Kuipers et al. 'An overview of constraint-based path selection algorithms for QoS routing'. In: *IEEE Communications Magazine* 40 (2002). DOI: 10.1109/MCOM.2002.1106159 (cited on page 70).

[202] Yasir Saleem, Nathalie Mitton, and Valeria Loscri. 'A Vehicle-to-Infrastructure Data Offloading Scheme for Vehicular Networks with QoS Provisioning'. In: *2021 International Wireless Communications and Mobile Computing (IWCMC)*. 2021. DOI: 10.1109/IWCMC51323.2021.9498708 (cited on page 70).

[203] David B. Johnson and David A. Maltz. 'Dynamic Source Routing in Ad Hoc Wireless Networks'. In: *Mobile Computing*. Ed. by Tomasz Imielinski and Henry F. Korth. Springer US, 1996, pp. 153–181. DOI: 10.1007/978-0-585-29603-6_5 (cited on page 77).

[204] Gabriel Montenegro, Christian Schumacher, and Nandakishore Kushalnagar. *IPv6 over Low-Power Wireless Personal Area Networks (6LoWPANs): Overview, Assumptions, Problem Statement, and Goals*. RFC 4919. 2007. DOI: 10.17487/RFC4919 (cited on pages 77, 198).

[205] Pascal Thubert and Jonathan Hui. *Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks*. RFC 6282. 2011. DOI: 10.17487/RFC6282 (cited on page 78).

[207] Leslie Lamport, Robert Shostak, and Marshall Pease. 'The Byzantine Generals Problem'. In: *Concurrency: The Works of Leslie Lamport*. Association for Computing Machinery, 2019, pp. 203–226. DOI: 10.1145/3335772.3335936 (cited on page 113).

[208] Miguel Castro, Barbara Liskov, et al. 'Practical byzantine fault tolerance'. In: *Third Symposium on Operating Systems Design and Implementation (OsDI)*. 1999. (Visited on Oct. 17, 2022) (cited on page 115).

[211] A. D. Wood and J. A. Stankovic. 'Denial of service in sensor networks'. In: *Computer* 35 (2002). DOI: 10.1109/MC.2002.1039518 (cited on page 143).

[213] Paavan Rughoobur and Leckraj Nagowah. 'A lightweight replay attack detection framework for battery depended IoT devices designed for healthcare'. In: *2017 International Conference on Infocom Technologies and Unmanned Systems (Trends and Future Directions) (ICTUS)*. 2017. DOI: 10.1109/ICTUS.2017.8286118 (cited on page 143).

[214] Qingyu Yang et al. 'On Optimal PMU Placement-Based Defense Against Data Integrity Attacks in Smart Grid'. In: *IEEE Transactions on Information Forensics and Security* 12 (2017). DOI: 10.1109/TIFS.2017.2686367 (cited on page 143).

[215] Yao Liu, Peng Ning, and Michael K. Reiter. 'False Data Injection Attacks against State Estimation in Electric Power Grids'. In: *ACM Transactions on Information and System Security (TISSEC)* 14 (2011). DOI: 10.1145/1952982.1952995 (cited on page 143).

[216] Yong Zeng and Rui Zhang. 'Active eavesdropping via spoofing relay attack'. In: *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2016. DOI: 10.1109/ICASSP.2016.7472059 (cited on page 143).

[217] Zhen Ling et al. 'An End-to-End View of IoT Security and Privacy'. In: *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*. 2017. DOI: 10.1109/GLOCOM.2017.8254011 (cited on page 143).

[218] Dimitris Mitropoulos and Diomidis Spinellis. 'Fatal injection: a survey of modern code injection attack countermeasures'. In: *PeerJ Computer Science* 3 (2017). DOI: 10.7717/peerj-cs.136 (cited on page 143).

[219] Néstor J. Hernández Marcano et al. 'On Goodput and Energy Measurements of Network Coding Schemes in the Raspberry Pi'. In: *Electronics* 5 (2016). DOI: 10.3390/electronics5040066 (cited on page 144).

[222] Chaoqun Yang et al. 'Multiple Attacks Detection in Cyber-Physical Systems Using Random Finite Set Theory'. In: *IEEE Transactions on Cybernetics* 50 (2020). DOI: 10.1109/TCYB.2019.2912939 (cited on page 145).

[223] Stamatis Karnouskos. 'Stuxnet worm impact on industrial cyber-physical system security'. In: *IECON 2011 - 37th Annual Conference of the IEEE Industrial Electronics Society*. 2011. DOI: 10.1109/IECON.2011.6120048 (cited on page 145).

[224] James P. Farwell and Rafal Rohozinski. 'Stuxnet and the Future of Cyber War'. In: *Survival* 53 (2011). DOI: 10.1080/00396338.2011.555586 (cited on page 145).

[225] Xian-Ming Zhang et al. 'Networked control systems: a survey of trends and techniques'. In: *IEEE/CAA Journal of Automatica Sinica* 7 (2020). DOI: 10.1109/JAS.2019.1911651 (cited on page 145).

[226] Jill Slay and Michael Miller. 'Lessons Learned from the Maroochy Water Breach'. In: *Critical Infrastructure Protection*. Ed. by Eric Goetz and Sujeet Shenoi. 2008. DOI: 10.1007/978-0-387-75462-8_6 (cited on page 145).

[227] Subham Sahoo et al. 'A Stealth Cyber-Attack Detection Strategy for DC Microgrids'. In: *IEEE Transactions on Power Electronics* 34 (2019). DOI: 10.1109/TPEL.2018.2879886 (cited on page 145).

[228] Gaoqi Liang et al. 'The 2015 Ukraine Blackout: Implications for False Data Injection Attacks'. In: *IEEE Transactions on Power Systems* 32 (2017). DOI: 10.1109/TPWRS.2016.2631891 (cited on page 145).

[229] Sandra Scott-Hayward, Gemma O'Callaghan, and Sakir Sezer. 'Sdn Security: A Survey'. In: *2013 IEEE SDN for Future Networks and Services (SDN4FNS)*. 2013. DOI: 10.1109/SDN4FNS.2013.6702553 (cited on page 146).

Webography

[1] Wikipedia. *Post box*. Sept. 19, 2022. URL: https://en.wikipedia.org/wiki/Post_box (visited on Sept. 26, 2022) (cited on page 1).

[2] Wikipedia. *History of email*. Sept. 21, 2022. URL: https://en.wikipedia.org/wiki/History_of_email (visited on Sept. 26, 2022) (cited on pages 1, 2).

[3] Wikipedia. *Wi-Fi*. Oct. 9, 2022. URL: https://en.wikipedia.org/wiki/Wi-Fi (visited on Oct. 11, 2022) (cited on page 1).

[4] Wikipedia. *Bluetooth*. Oct. 5, 2022. URL: https://en.wikipedia.org/wiki/Bluetooth (visited on Oct. 11, 2022) (cited on page 1).

[5] Wikipedia. *Zigbee*. Oct. 5, 2022. URL: https://en.wikipedia.org/wiki/Zigbee (visited on Oct. 11, 2022) (cited on page 1).

[9] Cybersecurity & Infrastructure Security Agency. *Identifying Critical Infrastructure During COVID-19*. Mar. 19, 2020. URL: https://www.cisa.gov/identifying-critical-infrastructure-during-covid-19 (visited on Sept. 25, 2022) (cited on page 5).

[13] CyberSANE. *Website*. Dec. 1, 2019. URL: https://www.cybersane-project.eu/ (visited on July 28, 2022) (cited on page 6).

[14] Wikipedia. *Multi-hop routing*. June 9, 2021. URL: https://en.wikipedia.org/wiki/Multi-hop_routing (visited on Oct. 17, 2022) (cited on page 8).

[15] Github. *Contiki-NG Github repository*. Aug. 4, 2021. URL: https://github.com/contiki-ng/contiki-ng (visited on Aug. 7, 2022) (cited on page 10).

[21] NIST CSRC. *Glossary - Vulnerability Definition*. Sept. 29, 2018. URL: https://csrc.nist.gov/glossary/term/vulnerability (visited on Sept. 22, 2022) (cited on page 16).

[22] France 24. *Cyber attacks hit two French hospitals in one week*. Feb. 16, 2021. URL: https://www.france24.com/en/europe/20210216-cyber-attacks-hit-two-french-hospitals-in-one-week (visited on Sept. 22, 2022) (cited on page 16).

[24] Joaquín Rodríguez. *CIPSEC - Most common attack vectors over Critical Infrastructures*. Jan. 26, 2018. URL: https://www.cipsec.eu/content/most-common-attack-vector-over-critical-infrastructures (visited on Aug. 26, 2020) (cited on page 16).

[26] National Cyber Security Centre UK. *New Cyber Attack categorisation system to improve UK response to incidents*. Apr. 11, 2018. URL: https://www.ncsc.gov.uk/news/new-cyber-attack-categorisation-system-improve-uk-response-incidents (visited on Sept. 22, 2022) (cited on pages 17, 23).

[28] Rebecca Smith. *Assault on California Power Station Raises Alarm on Potential for Terrorism*. Feb. 4, 2014. URL: https://www.wsj.com/articles/assault-on-california-power-station-raises-alarm-on-potential-for-terrorism-1391570879 (visited on Sept. 23, 2022) (cited on page 18).

[87] University of New South Wales. *The ADFA Intrusion Detection Datasets*. Oct. 31, 2013. URL: https://research.unsw.edu.au/projects/adfa-ids-datasets (visited on Sept. 23, 2022) (cited on page 32).

[89] University of the Aegean. *AWID Dataset*. Feb. 28, 2015. URL: https://icsdweb.aegean.gr/awid/ (visited on Sept. 23, 2022) (cited on page 32).

[92] University of New South Wales. *Bot-IoT Dataset*. June 2, 2021. URL: https://research.unsw.edu.au/projects/bot-iot-dataset (visited on Sept. 23, 2022) (cited on page 32).

[94] University of New Brunswick - Canadian Institute for Cybersecurity. *Botnet dataset*. Sept. 30, 2015. URL: https://www.unb.ca/cic/datasets/botnet.html (visited on Sept. 23, 2022) (cited on page 32).

[95] CAIDA - Center for Applied Internet Data Analysis. *The CAIDA "DDoS Attack 2007" Dataset*. Feb. 24, 2010. URL: https://www.caida.org/data/passive/ddos-20070804_dataset.xml (visited on Sept. 23, 2022) (cited on page 32).

[96] University of New Brunswick - Canadian Institute for Cybersecurity. *DDoS Evaluation Dataset (CIC-DDoS2019)*. Sept. 6, 2019. URL: https://www.unb.ca/cic/datasets/ddos-2019.html (visited on Sept. 23, 2022) (cited on page 32).

[99] University of New Brunswick - Canadian Institute for Cybersecurity. *CIC DoS dataset*. Nov. 10, 2017. URL: https://www.unb.ca/cic/datasets/dos-dataset.html (visited on Sept. 23, 2022) (cited on page 32).

[101] University of New Brunswick - Canadian Institute for Cybersecurity. *Intrusion Detection Evaluation Dataset (CIC-IDS2017)*. Dec. 7, 2017. URL: https://www.unb.ca/cic/datasets/ids-2017.html (visited on Sept. 23, 2022) (cited on page 32).

[103] Hochschule Coburg. *CIDDS - Coburg Intrusion Detection Data Sets*. May 24, 2017. URL: https://www.hs-coburg.de/forschung/forschungsprojekte-oeffentlich/informationstechnologie/cidds-coburg-intrusion-detection-data-sets.html (visited on Sept. 23, 2022) (cited on page 32).

[106] United States Military Academy - West Point. *CDX 2009 dataset*. Nov. 11, 2011. URL: https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets (visited on Sept. 23, 2022) (cited on page 32).

[108] Stratosphere Lab. *CTU-13 Dataset*. Feb. 6, 2018. URL: https://www.stratosphereips.org/datasets-ctu13 (visited on Sept. 23, 2022) (cited on page 32).

[110] Lincoln Laboratory - Massachusetts Institute of Technology. *DARPA*. July 15, 2000. URL: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset (visited on Sept. 23, 2022) (cited on page 32).

[113] UPV/EHU Aldapa - Faculty of Informatics. *Gure KDD Cup*. May 17, 2018. URL: http://www.sc.ehu.es/acwaldap/gureKddcup/ (visited on Sept. 24, 2022) (cited on page 32).

[116] University of New Brunswick - Canadian Institute for Cybersecurity. *Intrusion Detection Evaluation Dataset (ISCXIDS2012)*. Nov. 10, 2017. URL: https://www.unb.ca/cic/datasets/ids.html (visited on Sept. 24, 2022) (cited on page 32).

[118] University of Victoria - ISOT Research Lab. *Datasets*. July 10, 2012. URL: https://www.uvic.ca/engineering/ece/isot/datasets/ (visited on Sept. 24, 2022) (cited on page 32).

[121] University of California - Information and Computer Science. *KDD Cup 1999 Dataset*. June 20, 2006. URL: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html (visited on Sept. 24, 2022) (cited on page 32).

[123] Advanced Research in Cyber Systems. *Kent 2016*. 2016. URL: https://csr.lanl.gov/data/cyber1/ (visited on Sept. 24, 2022) (cited on page 32).

[125] Kyoto University. *Kyoto 2006+*. Nov. 1, 2006. URL: https://www.takakura.com/Kyoto_data/ (visited on Sept. 24, 2022) (cited on page 32).

[127] LBNL/ICSI Enterprise. *LBNL*. June 30, 2013. URL: http://icir.org/enterprise-tracing/Overview.html (visited on Sept. 24, 2022) (cited on page 32).

[129] Network & Data Security. *NDSec-1*. May 31, 2020. URL: https://www2.hs-fulda.de/NDSec/NDSec-1/ (visited on Sept. 24, 2022) (cited on page 32).

[131] University of New Brunswick - Canadian Institute for Cybersecurity. *NSL-KDD dataset*. 2022-09-23. Nov. 10, 2017. URL: https://www.unb.ca/cic/datasets/nsl.html (cited on page 32).

[138] R Hofstede. *SSH datasets*. Oct. 23, 2014. URL: https://www.simpleweb.org/wiki/index.php/SSH_datasets (visited on Sept. 24, 2022) (cited on page 32).

[140] University of New South Wales. *TON_IoT Datasets*. June 2, 2021. URL: https://research.unsw.edu.au/projects/toniot-datasets (visited on Sept. 23, 2022) (cited on page 32).

[142] Security & Privacy Laboratory. *TRAbID - Datasets*. Aug. 20, 2017. URL: https://secplab.ppgia.pucpr.br/?q=trabid (visited on Sept. 24, 2022) (cited on page 32).

[145] Security & Privacy Laboratory. *Twente - Datasets*. Feb. 15, 2019. URL: https://www.simpleweb.org/wiki/index.php/Labeled_Dataset_for_Intrusion_Detection (visited on Sept. 24, 2022) (cited on page 32).

[147] NESG - Network Engineering & Security Group. *UGR'16 - Datasets*. June 6, 2017. URL: https://nesg.ugr.es/nesg-ugr16/ (visited on Sept. 24, 2022) (cited on page 32).

[149] University of Brescia. *UNIBS-2009 - Datasets*. Apr. 3, 2010. URL: http://netweb.ing.unibs.it/~ntw/tools/traces/ (visited on Sept. 24, 2022) (cited on page 32).

[151] Advanced Research in Cyber Systems. *Unified Host and Network Data Set*. Sept. 11, 2017. URL: https://csr.lanl.gov/data/2017/ (visited on Sept. 24, 2022) (cited on page 32).

[153] University of New South Wales. *UNSW-NB15 Dataset*. June 2, 2021. URL: https://www.unsw.adfa.edu.au/unsw-canberra-cyber/cybersecurity/ADFA-NB15-Datasets/ (visited on Sept. 24, 2022) (cited on page 32).

[158] Enclave Security. *Open Threat Taxonomy (Version 1.1)*. Mar. 26, 2016. URL: https://www.auditscripts.com/free-resources/open-threat-taxonomy/ (visited on Sept. 24, 2022) (cited on page 34).

[162] Harry W. *Getting started with cyber incident management*. Sept. 19, 2019. URL: https://www.ncsc.gov.uk/blog-post/getting-started-with-cyber-incident-management (visited on Sept. 25, 2022) (cited on page 36).

[164] IT Governance UK. *Cyber Security Risk Management*. July 23, 2012. URL: https://www.itgovernance.co.uk/cyber-security-risk-management (visited on Sept. 25, 2022) (cited on page 39).

[165] IT Governance UK. *Cyber Security Risk Assessment*. Apr. 30, 2019. URL: https://www.itgovernance.co.uk/cyber-security-risk-assessments (visited on Sept. 25, 2022) (cited on page 39).

[167] Cambridge Dictionary. *Trust*. Aug. 10, 2022. URL: https://dictionary.cambridge.org/dictionary/english/trust (visited on Sept. 4, 2022) (cited on page 43).

[168] Meysam Poorkavoos. *Eight behaviours that build trust*. Oct. 19, 2016. URL: https://www.roffeypark.ac.uk/knowledge-and-learning-resources-hub/eight-behaviours-that-build-trust/ (visited on Sept. 4, 2022) (cited on page 44).

[169] Cambridge Dictionary. *Reputation*. Aug. 10, 2022. URL: https://dictionary.cambridge.org/dictionary/english/reputation (visited on Sept. 5, 2022) (cited on page 45).

[170] Trivago. *Trivago hotel, taxi and flight booking service*. Feb. 28, 2011. URL: https://www.trivago.com/ (visited on Aug. 9, 2022) (cited on page 45).

[177] Wikipedia. *Bitcoin*. Oct. 6, 2022. URL: https://en.wikipedia.org/wiki/Bitcoin (visited on Oct. 15, 2022) (cited on page 47).

[180] Wikipedia. *Blockchain*. Oct. 14, 2022. URL: https://en.wikipedia.org/wiki/Blockchain (visited on Oct. 15, 2022) (cited on page 48).

[181] Wikipedia. *Merkle tree*. Sept. 9, 2022. URL: https://en.wikipedia.org/wiki/Merkle_tree (visited on Oct. 15, 2022) (cited on page 48).

[191] MICHELIN Guide. *About Us*. July 10, 2019. URL: https://guide.michelin.com/gb/en/about-us (visited on Sept. 5, 2022) (cited on page 51).

[206] AVG AntiVirus. *AVG Support Community - MacOS:Genieo-FW[Adw]*. Sept. 7, 2018. URL: https://support.avg.com/answers?id=9060N000000g9SNQAY (visited on Aug. 29, 2022) (cited on page 107).

[209] EU Innovation Radar. *Integrated incident handling and response for critical infrastructure*. Dec. 1, 2020. URL: https://www.innoradar.eu/innovation/40427 (visited on Oct. 17, 2022) (cited on page 136).

[210] Future Internet Testing Facility. *FIT IoT-LAB - The Very Large Scale IoT Testbed*. July 3, 2020. URL: https://www.iot-lab.info/ (visited on Sept. 25, 2022) (cited on page 138).

[212] NIST CSRC. *Glossary - Brute Force Password Attack Definition*. Aug. 6, 2020. URL: https://csrc.nist.gov/glossary/term/brute_force_password_attack (visited on Sept. 22, 2022) (cited on page 143).

[220] NIST CSRC. *Glossary - Covert Channel Definition*. Apr. 12, 2020. URL: https://csrc.nist.gov/glossary/term/covert_channel (visited on Sept. 22, 2022) (cited on page 144).

[221] NIST CSRC. *Glossary - cyber-physical system(s)*. Apr. 3, 2020. URL: https://csrc.nist.gov/glossary/term/cyber_physical_systems (visited on Sept. 23, 2022) (cited on page 145).

Nomenclature

α	Weight of malicious routing actions
β	Sensitivity factor of reputation sigmoid function
δ_n^M	Weight relation between good and bad mining actions in sigmoid function for node n
δ_n^R	Weight relation between good and bad routing actions in sigmoid function for node n
γ	Weight of malicious mining actions
λ_n^M	Decay factor for mining reputation for node n
λ_n^R	Decay factor for routing reputation for node n
B	Collection of nodes extracted from received block
C_{max}	Maximum cost of link between two nodes
C_{min}	Minimum cost of link between two nodes
C_n	Cost of link with node n
f_{size}	Number of bits in link-cost field size
L_{max}	Maximum route length
M	Collection of mined nodes
N	Number of nodes in the network
n	Singular network node
P_{Ma}	Percentage of malicious actions
R_{n_t}	Overall reputation of node n at time t, post decay
$R_{n_t}^M$	Reputation of node n's mining activities at time t, post decay
$R_{n_t}^R$	Reputation of node n's routing activities at time t, post decay
R_n^M	Calculated mining reputation of node n
R_n^R	Calculated routing reputation of node n
$Rd_{n_t}^M$	Reputation decay of node n's mining activities at time t
$Rd_{n_t}^R$	Reputation decay of node n's routing activities at time t
S_m^C	Consensus score of Miner m
S_m^M	Mining score of Miner m
S_m^R	Routing score of Miner m

$S^M_{bad_n}$ Sum of W^M_n bad mining actions of node n

$S^R_{bad_n}$ Sum of W^R_n bad routing actions of node n

$S^M_{good_n}$ Sum of W^M_n good mining actions of node n

$S^R_{good_n}$ Sum of W^R_n good routing actions of node n

$t^M_{\frac{1}{2}m}$ Decay half-life for mining reputation in seconds for miner m

$t^R_{\frac{1}{2}n}$ Decay half-life for routing reputation in seconds for node n

t^M_m Timestamp of the last mining reputation update for Miner m

t^R_n Timestamp of the last routing reputation update for node n

W^M_n Action window of node n's mining activities

W^R_n Action window of node n's routing activities

Special Terms

Numbers

6LoWPAN IPv6 over Low-Power Wireless Personal Area Networks conforming to the IEEE 802.15.4-2003 standard, as defined in [204]. 10, 77, 78, 87, 104, 130

A

AIDS Anomaly-based Intrusion Detection System. 25, 42

ANN Artificial Neural Network. 30

AODV Ad hoc On-Demand Distance Vector. vii, viii, 12, 47, 65, 68, 69, 71–77, 79–81, 83, 87, 89–96, 98–105, 126, 127, 130–133, 137, 138, 175

AODV-Miner Reputational consensus-based implementation of AODV. 75–77, 80, 81, 87, 89–96, 98–105, 124–127, 131–133, 137, 175

AODV-Miner Quarantine Quarantine implementation of AODV-Miner. 124–133, 138

API Application Programming Interface. 140, 147

ASTR Aggregate Signature based Trust Routing. 47

B

BFT Byzantine Fault Tolerance. 109, 113–115, 133

Black-hole Routing-based attack, where an attacker destroys all packets which come into their possession, causing localised network blackout. 21, 22, 87–90, 94, 96, 98, 99, 104, 105, 124–126, 128, 131, 133

C

CERT Computer Emergency Response Team. 34

CI Critical Infrastructure. vii, viii, 1, 5–11, 15, 29, 30, 33, 34, 37, 41–43, 135, 136, 140, 145

CII Critical Information Infrastructure. 4, 6, 9, 31, 33–35, 136

CIIP CII Protection. 34

Contiki-NG is an operating system for resource-constrained devices in the Internet-of-Things. 10, 12, 62, 67, 78, 87, 125, 138

Cooja is an open source network simulator using the Contiki-NG OS. 10–12, 62, 79, 87, 88, 125, 139

CPS Cyber Physical System. 19, 38, 145

CPU Central Processing Unit. 27

CSIRT Computer Security Incident Response Team. 37

CSMA Carrier Sense Multiple Access is a MAC protocol allowing network communications using traffic avoidance methods such as listening to the wireless medium prior to transmission. 10

CyberSANE is a Cyber Security Incident Handling, Warning and Response System for the European Critical Infrastructures, EU H2020 project under grant agreement No 833683. vii, viii, 1, 4, 6, 7, 9–12, 31, 34, 35, 135, 136, 139, 140

D

DAG Directed Acyclic Graph. 67, 81, 82, 84–86, 106

DAO Destination Advertisement Object. 68, 84–86

DarkNet *Deep and Dark Web Mining and Intelligence* component, recovers and analyses user generated data from various electronic sources, including the Deep and Dark Web.. 7, 8

DCI Destination Context Identifier. 78

DDoS Distributed DoS. 38

DIO DODAG Information Object. 67

DODAG Destination-Oriented DAG. 67, 68, 81, 84–87

DoS Denial-of-Service. 17–19, 21, 22, 28, 29, 35, 38, 143, 145

DSR Dynamic Source Routing. vii, viii, 12, 65, 68–70, 75, 77–81, 83, 84, 87, 94–99, 104, 105, 128–130, 133, 137, 138

DSR-Miner Reputational consensus-based implementation of DSR. 77, 80, 87, 89, 94–98, 105, 124, 128, 129, 133, 137

DSR-Miner Quarantine Quarantine implementation of DSR-Miner. 124, 125, 128–130, 133, 138

E

EMD Earth Mover's Distance. 28

ENISA European Union Agency for Network and Security Information. 33–35

ETSI European Telecommunications Standard Institute. 33

F

FDI False Data Injection. 143, 145

G

Grey-hole Routing-based attack, variation of the Black-hole attack where an attacker destroys packets which come into their possession based either on probability (i.e., dropping 25% of packets), or based upon selected criteria (i.e, dropping only acknowledgement packets), causing varying degrees of disruption to network operations. 22, 87, 88, 92, 93, 96–99, 102–105, 124, 125, 127, 129, 132, 133

H

HIDS Host-based Intrusion Detection System. 25

HMM Hidden Markov Model. 30

HybridNet *Data Fusion, Risk Evaluation and Event Management* component, provides the intelligence for the analysis of security events, derived from data acquired from LiveNet and DarkNet.. 7, 8, 135, 139

I

ICMP Internet Control Message Protocol. 67

ICMPv6 Internet Control Message Protocol for IPv6. 67

ICT Information Communication Technologies. 34

IDS Intrusion Detection System. 22, 24, 27, 30, 31, 37, 40, 42, 136

IoT Internet-of-Things. vii, viii, 1–5, 8, 10–12, 15–22, 24, 35, 37–39, 41–46, 48, 49, 54, 63, 68, 69, 77, 107, 135, 136, 138–140, 145, 146

IP Internet Protocol. 16, 21, 56–59, 61, 68–70, 76–79, 116, 124, 130

IPv4 Version 4 representation of a network IP address, 32 bits in length represented in decimal format, each byte separated by a decimal point, as defined in [198]: w.x.y.z (i.e., 128.93.162.214). 67, 69, 77, 78, 104

IPv6 Version 6 representation of a network IP address, 128 bits in length represented in hexadecimal format, every two bytes separated by a colon, as defined in [199]: aaaa:bbbb:cccc:dddd:eeee:ffff:gggg:hhhh (i.e. fd00::201:1:1:1). 10, 43, 58, 67, 69, 70, 78, 79, 87, 104, 116, 130

IR Incident Response. 37, 40

ISM Band Industrial, Scientific and Medical Radio Band. 1

IT Information Technology. 4, 12, 15, 17, 18, 24, 30, 34, 36, 54, 65, 113, 114, 136, 140, 143, 145

K

KNN K-Nearest Neighbors. 27

L

LiveNet *Live Security Monitoring and Analysis* component, capable of preventing and detecting threats, serving as an interface between the underlying CI and the CyberSANE system.. 7–9, 31, 135, 136, 139

M

MAC Media Access Control. 10, 56, 57, 76, 78, 79, 101

MANET Mobile Ad Hoc Network. 19, 21, 22, 139

MCA Multivariate Correlation Analysis. 28

MitM Man-in-the-Middle. 19, 35, 143, 145

MPR MultiPoint Relay. 67

MTU Maximum Transmission Unit. 77, 104, 130

N

NCC Nonlinear Correlation Coefficient. 28
NCS Networked Control System. 145
NIDS Network-based Intrusion Detection System. 25, 42, 51
NIST National Institute of Standards and Technology. 34, 36, 145

O

OCSVM One Class SVM. 30
OLSR Optimized Link State Routing Protocol. 67, 69, 138
OS Operating System. 10–12, 16, 25
OSI Open Systems Interconnection. 21, 146
OTT Open Threat Taxonomy. 34

P

PAN Personal Area Network. 1
PCA Principal Component Analysis. 28
PoW Proof-of-Work. 48, 61, 64, 137
PrivacyNet *Privacy & Data Protection Orchestrator* component, manages and orchestrates the application of privacy mechanisms, maximising the confidentiality and data protection in compliance with GDPR.. 8

Q

QoS Quality-of-Service. 46, 70, 71

R

R2L Remote-to-Local. 18, 19, 22, 143
RERR Route Error. 69
RFC Request for Comments. 10, 67–69, 77, 80, 81, 87
RPL Routing Protocol for Low-Power and Lossy Networks. vii, viii, 12, 65, 67, 68, 75, 81–84, 86, 87, 105, 106, 137
RPL-Miner Reputational consensus-based implementation of RPL. 81, 83, 86, 87, 138
RREP Route Reply. 69–72, 75–79, 83, 84, 96, 102
RREP-2Hop Update to the Route Reply packet structure to allow for validation of routing activities. 76, 77, 79
RREQ Route Request. 69, 71, 72, 77, 78, 80, 102
RVT Route Validation Table. 55–57, 64, 76, 77, 79, 80, 84–87, 106

S

SCADA Supervisory Control and Data Acquisition. 30, 41
SCI Source Context Identifier. 78
SDN Software Defined Network. 19, 39, 146, 147
ShareNet *Intelligence and Information Sharing and Dissemination* component, provides the necessary threat intelligence and information sharing capabilities within CIs and other parties.. 8
SIDS Signature-based Intrusion Detection System. 24, 25, 42
SLA Service Level Agreement. 34
SVM Support-Vector Machine. 25, 26, 30

T

TC Topology Control. 67
TCP Transmission Control Protocol. 35
TOCSR Taxonomy of Operational Cyber Security Risks. 34
TTL Time To Live. 58, 73, 89

U

U2R User-to-Root. 18, 22, 143
UDP User Datagram Protocol. 21, 78, 79

W
WBAN Wireless Body Area Network. 21
WEP Wired Equivalent Privacy. 2
WPA2 Wi-Fi Protected Access II. 2
WPA3 Wi-Fi Protected Access III. 2
WSN Wireles Sensor Network. 21, 22, 35, 39, 46, 47