A Cryptocurrency Framework for Decentralized Physical Space Networking

Abhijit Roy

Department of Computer Science

McGill University

Montréal, Québec, Canada

October, 2023

A thesis submitted to McGill University in partial fulfillment of the requirements of the degree of Masters of Computer Science.

Abstract

The world is rapidly undergoing digitization, with blockchain-based solutions playing a crucial role in providing a fully digital and decentralized approach to monetary transactions. Blockchains have proven to be versatile solutions in various domains, including healthcare and supply chain management. However, their potential in digitizing physical world locations and spaces has not been fully realized.

In a digital world that accurately represents the physical world, each physical entity, such as individuals with fitness trackers, smart vehicles, and smart spaces, would be represented by their digital counterparts. However, these digital representations must be accompanied by reliable physical world data, such as location coordinates for fitness trackers or the supported functionalities of smart spaces. Existing solutions that do not verify physical world data or rely on external hardware oracles to collect and process such data are not viable solutions.

To address the limitations of current solutions, we have developed an innovative, oracle-independent framework called the *Decentralized Physical Space Network*. DPSN leverages blockchain technology and verified crowd work to verify the physical location of

objects and the attributes of smart spaces. It consists of two integral components: *Location Mining*, which verifies the physical world location of objects, and *Space Mining*, which verifies the attributes and functionalities of smart spaces in the physical world. This approach creates a trustworthy connection between the digital representation of physical objects, smart spaces, and their respective locations and functionalities. Unlike oracle-based solutions, our approach does not rely on external trusted hardware oracles to collect and process physical world data. Instead, we opportunistically select participants to process the physical world data as needed. This allows us to overcome several limitations associated with oracle-based solutions, such as dependence on proprietary trusted hardware.

We have evaluated Location Mining in a virtual geospatial plane against various criteria, including throughput and object movement patterns. The results confirmed that the network correctly identified and rewarded honest and active participation. Additionally, we have used these findings to improve the protocol and ensure fair distribution of rewards, even in the presence of adversaries that can opportunistically program their movements to gain more rewards.

Abrégé

Le monde se numérise rapidement et les solutions basées sur «blockchain» jouent un rôle crucial en fournissant une approche entièrement numérique et décentralisée des transactions monétaires. «Blockchains» se sont révélées être des solutions polyvalentes dans divers domaines, notamment les soins de santé et la gestion de la chaîne d'approvisionnement. Toutefois, leur potentiel en matière de numérisation des lieux et des espaces du monde physique n'a pas encore été bien exploité.

Dans un monde numérique qui représente fidèlement le monde physique, chaque entité physique, comme les individus équipés de trackers de fitness, les véhicules intelligents et les espaces intelligents, serait représentée par son équivalent numérique. Toutefois, ces représentations numériques doivent être accompagnées de données fiables sur le monde physique, telles que les coordonnées de localisation des trackers de fitness ou les fonctionnalités prises en charge par les espaces intelligents. Les solutions existantes qui ne vérifient pas les données du monde physique ou qui s'appuient sur des oracles matériels externes pour ramasser et transformer ces données ne sont pas viables.

Pour répondre aux limites des solutions actuelles, nous avons développé un cadre innovant et indépendant de l'oracle appelé «*Decentralized Physical Space Network*». DPSN s'appuie sur la technologie «blockchain» et «verified crowd work» pour vérifier l'emplacement physique des objets et les attributs des espaces intelligents. Il s'agit de deux composantes intégrales : «*Location Mining*», qui vérifie l'emplacement des objets dans le monde physique, et «*Space Mining*», qui vérifie les attributs et les fonctionnalités des espaces intelligents dans le monde physique. Cette approche crée un lien de confiance entre la représentation numérique des objets physiques, les espaces intelligents et leurs emplacements et fonctionnalités respectifs. Contrairement aux solutions basées sur des oracles, notre approche ne dépend pas d'oracles matériels externes de confiance pour ramasser et transformer les données du monde physique. Au lieu de cela, nous sélectionnons de manière opportuniste des participants pour transformer les données du monde physique. Cela nous permet de surmonter plusieurs limitations associées aux solutions basées sur des oracles, telles que la dépendance à l'égard d'un matériel de confiance propriétaire.

Nous avons évalué «Location Mining» dans un plan géospatial virtuel en fonction de divers critères, notamment le débit et les modèles de mouvement des objets. Les résultats ont confirmé que le réseau identifiait et récompensait correctement la prticaipation honnête et active. En outre, nous avons utilisé ces résultats pour améliorer le protocole et garantir une distribution équitable des récompenses, même en présence d'adversaires qui peuvent

programmer leurs mouvements de manière opportuniste pour obtenir davantage de récompenses.

Acknowledgements

I would like to express my heartfelt gratitude to:

- Prof. Muthucumaru Maheswaran for his thorough guidance and constructive feedback throughout my time here in McGill and thousands of brainstorming sessions.
- Vishal Kulkarni for his invaluable contribution in drafting the initial version of Space Mining Algorithm.
- To my beloved family back in India, without whose support, my dream would have stayed a dream.
- An anonymous 6′2” god-sent angel and a friend who has always helped me untangle myself.
- All the amazing supportive people and friends here at McGill and back home.

If not for you all, this journey would certainly have been a different one and probably not as memorable.

Contents

List of Figures

List of Acronyms

AP	Access Points.
API	Application Programming Interface.
AR	Augmented Reality.
CoM	Call for Meeting.
DPSN	Decentralized Physical Space Network.
DSN	Decentralized Storage Network.
DTN	Decentralized Transaction Networks.
GPS	Global Positioning System.
IoT	Internet of Things.
LBS	Location Based Services.
MK	Meeting Keeper.
MK_0	Meeting Keepers at day 0.
MMORPG	Massive(ly) Multiplayer Online Role-Playing Games.
NFTs	Non-Fungible Tokens.

NGO	Non-Governmental Organizations.
PoE	Proof of Existence.
PoIs	Points of Interest.
PoL	Proof-of-Location.
PoM	Proof of Meeting.
PoS	Proof-of-Stake.
PoST	Proof of Space Time.
PoW	Proof-of-Work.
QoS	Quality of Service.
REST	Representational State Transfer.
RSU	Road Side Units.
SK	Space Keeper.
SM	Space Miners.
TCRs	Token Curated Registries.
VANET	Vehicular Ad Hoc Networks.

Chapter 1

Introduction

The world is rapidly digitizing, and blockchain-based cryptocurrencies are playing a significant role in this advancement. These cryptocurrencies offer a fully digital and decentralized approach to monetary transactions. Since the introduction of the pioneering cryptocurrency Bitcoin [1], cryptocurrencies have gained widespread attention and prompted extensive research to apply the underlying blockchain technology to domains such as healthcare [2], land registries [3], and anti-corruption efforts [4].

There are some existing solutions that have tried to digitize various aspects of the physical world using blockchain technology, but none of them have been successful in providing a complete and trustworthy framework for digitizing physical world entities. For example, several frameworks exist that maintain notarized land registries on the blockchain, however this is just a partial solution as the notarization authorities are off chain trusted entities.

Although, our framework, Decentralized Physical Space Network (DPSN), does not support high-stake scenarios such as ownership verification, it enables us to create trustworthy digital handles by verifying physical world location and attributes, without being dependent on off-chain entities.

In a digital world, all physical entities such as individuals with fitness trackers, drones, streetlights, and private parking spaces would be represented by unique digital handles. Ensuring the verification of these handles' reported locations and accurately capturing all the functionalities supported by the entities are crucial in this endeavor. *Proof-of-Location (PoL)* systems enable the verification of the sender's location, ensuring trustworthy location sharing [5]. Although PoL systems can address the first requirement, contemporary PoL frameworks are far from perfect. Furthermore, there is an absence of frameworks that can address the second requirement by creating trustworthy digital representations of physical spaces, that capture all their capabilities and attributes.

In PoL frameworks, physical entities are represented by their digital handles, and these frameworks provide assurance that the location provided by these handles is reliable. While there are several frameworks for creating and sharing PoLs, they often rely on trusted hardware or third parties known as *Oracles* [6] to capture and process physical world data. In this document, we provide a detailed analysis of these frameworks in chapter 3 and compare their approach to DPSN.

Our solution, Decentralized Physical Space Network (DPSN), overcomes the limitations

of existing PoL frameworks and introduces a novel approach to creating trustworthy digitized spaces. Our PoL framework, called *Location Mining*, verifies the location shared by all participants without relying on oracles to collect and process physical world data. Instead, DPSN selects a small group of participants to hold meetings where nearby participants' physical presence is verified through challenges. These meetings provide a powerful mechanism to verify participants' locations and prevent attacks such as *Location Spoofing*. Unlike oracle-based solutions, which are limited by hardware capabilities, DPSN can grow stronger with each new participant as we leverage active participation.

We also introduce *Space Mining*, which verifies the existence of advertised physical spaces and the accessibility of all their features through digital handles. DPSN selects participants to help verify spaces by interacting with them and logging observations to the blockchain. Accessing a space through its digital handle grants users access to various supported functionalities. These functionalities are implemented as *complementary functions* with well-defined transition logic and two acceptable states. For example, the complementary function *isOccupied* checks if a parking space is available and returns its occupancy state. The observations are later verified by DPSN to algorithmically determine the authenticity of the space. This protocol is part of our novel proof system called *Proof of Existence (PoE)*.

DPSN is built on the Algorand blockchain, which belongs to a family of cryptocurrency frameworks primarily focused on preventing *double spending*, similar to Ethereum and

Bitcoin. However, Algorand offers immediate finality, is free from forks, and has a very short time to finality [7]. This allows us to process data gathered from the physical world at a much faster rate. Additionally, since DPSN is a software-based framework, it is not subject to the large capital investment typically required for traditional hardware-based solutions.

By accurately binding physical locations to digital handles and capturing their capabilities, we can create trustworthy digital twins [8] [9] of physical entities that reflect any changes in their physical counterparts. This would not be possible if oracles were used to create only an initial digital image of an entity. DPSNs can host a variety of applications, including pervasive games, supply chain validation, validated crowd work in the physical world, space sharing, equipment sharing, and other applications related to spaces and things.

We provide comprehensive algorithms for both Location Mining and Space Mining. We have simulated Location Mining in a virtual geospatial plane using Python and conducted an in-depth analysis of the simulation, including aspects such as throughput, reward rate, and other key concepts to demonstrate its effectiveness. This work is supported by a comprehensive security analysis, covering major attacks and their respective safeguards, followed by an extensive list of possible DPSN applications.

Contribution of Authors

This work is a result of a collaboration between the author and Professor Maheswaran. Through continuous iteration over the underlying protocols, the system as a whole was improved. Although we have come a long way from the initial idea of Location Mining, the idea of meeting other participants during verified meeting belongs to Professor Maheswaran. From there, the author and Professor Maheswaran have collaborated to create a working algorithm for *Location Mining*.

As the author and Professor Maheswaran were in the final stages of designing Location Mining, we felt the need for verifying and adding smart physical spaces to the network. *Space Mining* was a result of this collaboration. Vishal Kulkarni, a colleague and a student under the supervision of Professor Maheswaran, joined us momentarily to refine Space Mining and provide insights that would make the protocol more secure and usable. Location Mining and Space Mining are further supported with an implementation plan, which was created by the author.

System design, implementation, data collection and result presentation were completed by the author. However, Professor Maheswaran's feedback was crucial in identifying the parameters that would have the largest impact for future stakeholders. Upon analysing the test results, we came across a vulnerability where participants could opportunistically program their movement patterns. *Randomized look-back* was collectively designed by the author and Professor Maheswaran to make the protocol resilient against such attacks.

We also discuss various attack vectors, and possible real-world applications of the protocol. Although these sections have been made by the author, Professor Maheswaran's feedback provided important insights such as outsider and an insider's perspective of how such frameworks could be attacked, a starting point for interesting applications such as drone-enabled deliveries.

Chapter 2

Background

There is a large variety of cryptocurrency frameworks today, most built to facilitate monetary transactions. The consensus algorithms used by different frameworks vary - some take longer to finalize the transactions while other are considerably faster. Transaction throughput, finality time, possibility of fork, disconnection tolerance are some of the key differentiating factors among the different prevalent frameworks. In order to have a richer contextualized discussion in the following chapters, we start with a brief and simplified overview of Blockchain, and then move on to providing an overview of two frameworks relevant to our work in this chapter.

2.1 Blockchain: A brief Overview

Blockchain is an immutable, decentralized and distributed ledger of transactions, stored across a distributed network of devices. It is essentially a sequence of blocks, where each block contains several transactions, and a block is connected to its predecessor through a secure hash. Since each block is connected to the previous block via a hash of the previous block, any attempts to change even minute details in previous blocks would invalidate all the future blocks.

After a transaction has been created by a participant, it is broadcast to its peers. Specialised participants, *miners*, gather such transactions, validate them and add them to their own blocks. Depending on the underlying consensus mechanism, some participants can acquire the rights to add the next valid block to the blockchain. If the resulting sequence of blocks is accepted by the majority(majority can mean several things, such as computing capacity, stake in the network, etc.) of the participants, we can conclude that a consensus has been reached on the order of transactions.

If the network cannot reach a consensus, a fork is created. Forks might be caused by software updates, propagation delays, etc. However such scenarios are undesirable and can cause several problems, such as unwanted delay and confusion caused by uncertainty in transactions' validity during a fork, double spending attacks [10], where adversaries try to spend the same token twice by creating conflicting transactions on separate forks.

2.2 Decentralized Transaction Network

Nowadays, DTNs have extended their reach beyond monetary transactions. DTNs have been applied in many different fields such as managing NFTs [11], Land Registries [12], and others. However, with its current capabilities, DTNs are only suited to exchange digital native tokens.

Different DTNs, although differing in implementation, share the common goal of creating a single version of the transaction history to prevent double spending [13] [10], a scenario in which an attacker tries to spend the same token twice by broadcasting two conflicting transactions to separate groups of peers.

The two most prominent DTNs, Bitcoin and Ethereum initially used similar *Proof-of-Work (PoW)* consensus mechanism to reach a globally accepted version of the total order of transactions, in other words, a consensus. Since then, Ethereum with its new version Ethereum 2.0 has moved to a new consensus mechanism, *Proof-of-Stake (PoS)*.

PoW in Bitcoin involves solving computationally complex puzzles to reach a consensus. This requires expending a significant amount of computational resources in order to get the rights to add a new block which is consistent with prior blocks. Moreover, it is almost impossible to revise records after they have been mined, because all blocks are connected via block hashes and changing them across the entire network requires a significant amount of computing resources [1].

Unlike PoW, in PoS a validator is selected based on their stakes in the network, that is,

the higher the stake, the higher the chances of getting selected as a validator. By attacking the network, the attacker will not only put their stake at risk, but they will also devalue their investment in the network.

2.3 Decentralized Storage Network

Filecoin protocol is a Decentralized Storage Network (DSN) built on a blockchain supported by an underlying native token. Clients spend tokens for storing and retrieving their data, and miners can earn tokens by storing and serving data [14].

Storage services on Filecoin must be both verifiable and auditable by making *Sybil Attacks* [15], *Outsourcing Attacks* [16] and *Generation Attacks* [16] impossible. This is achieved by two novel proof systems, *Proof-of-Replication* and *Proof-of-Spacetime*.

Proof-of-Replication allows a storage miner to prove that they are creating the required number of independent copies. *Proof-of-Spacetime* on the other hand, allows a user to check if the storage miner is storing the data for the promised time-period. These proofs are regularly added to the blockchain, where they are verified by other users.

Chapter 3

Related Works

Location plays a vital role in many day-to-day activities such as navigation, emergency services, healthcare services. It is also essential for many industries such as supply-chain management. However, traditional positioning services like Global Positioning System (GPS) have some major shortcomings, such as needing to connect to at least 4 satellites for accurate results [17], lack of verifiable location reports by the GPS sensors and being vulnerable to spoofing attacks [18].

We begin by taking a brief look at systems that use off-chain trusted oracles to process the physical world data as they highlight the drawbacks of off-chain verification. Then, we move on to the approach we have chosen, Proof-of-Location. PoL enables us to overcome the limitations of traditional positioning services providing a way to verify the location shared by any device [5]. We provide a detailed overview of leading PoL systems and frameworks

and describe what sets DPSN apart.

3.1 Off-chain verification frameworks

Several blockchain-based frameworks such as maintaining land-registries on blockchain [3] and anti-corruption facilitated by blockchain technology [4] depend on off-chain trusted authorities to verify physical world data so that it can be added to the blockchain. For example, several solutions that store notarized land-registries on the blockchain depend on off-chain notarization agencies to verify the documents before they can be added to the blockchain.

Such solutions are only as strong as the least trustworthy off-chain authority and pose a risk to the entire framework if a single authority is compromised. For example, lets assume an adversarial notary is successful in adding a few fake registries to the blockchain. This would not only be a spurious claim that cannot be backed in the physical world, it would also reduce the overall value of a system in which authentic registries co-exist with fake registries. Such attacks would be extremely difficult in DPSN, as adversaries would need to control a majority of the network to influence the consensus.

Since the verification authorities exist off-chain, the only reaction strategy to reduce the severity of attacks is to classify notaries as trustworthy or non-trustworthy, based on their observed behaviour. On the contrary, since all entities are part of the network in DPSN, we can take more direct measures by controlling the assets held by participants, as a function

of their observed behaviour.

3.2 FOAM

FOAM is an open protocol for decentralized, geo-spatial data markets. Its components are designed to provide spatial protocols, standards and applications that bring geo-spatial data to the blockchain and empower a consensus-driven map of the world [19]. PoL in FOAM has two components, *Static PoL* and *Dynamic PoL*.

On one hand, static PoL incentivizes the creation of Token Curated Registries (TCRs), which correspond to Points of Interest (PoIs) in the physical world, and is collectively curated by *Cartographers*. On the other hand, dynamic PoL uses proprietary hardware setup to generate PoLs for moving objects. Dynamic PoL uses Zones, which are formed up of at least 4 proprietary radio beacons, better known as Zone Anchors. Each anchor must set aside a collateral and synchronize their clocks with other anchors before participating in PoL activities. After the zone setup is complete, anchors can triangulate an object's location and add it to their local blockchain. *Verifiers* process these local claims and add them to the global blockchain.

To begin with, unlike FOAM, DPSN does not depend on proprietary hardware solutions to drive the consensus. Additionally, there are no fixed roles in DPSN, each participant has an equal opportunity to participate, even as miners when wanderers acquire sufficiently high PoST scores. However, this is not possible in FOAM, where certain roles such as Zone

Anchors are restricted by hardware capabilities. Unlike FOAM, DPSN is also free from the overhead of synchronization as it utilizes *rounds*, a logical time unit used in Algorand. Furthermore, since FOAM is built on Ethereum with time to finality set to 15 minutes [20], it is significantly slower when compared to DPSN which is built on Algorand with time to finality of around 4 seconds [7].

3.3 XY Oracle Network

XYO Network is a queryable network of untrusted nodes that constructs trustworthy responses which can be used in smart contracts, enabling developers to interact with the physical world as if it were an API [21]. It uses zero knowledge proofs to generate PoLs. PoL occurs in layers using *Proof of Origin* and *Bound Witnesses*.

Proof of Origin uses zero-knowledge proofs to establish that two or more pieces of data originated from the same source. Proof of Origin itself relies on Bound Witnesses, where two or more entities can prove that they were in close proximity by co-signing close range interactions. Using a complex network of nodes with varying capabilities, the network responds to queries issued to itself, fetching the answer with the highest confidence score.

Unlike DPSN where all wanderers can add data to the blockchain, only Diviners - a specialised component in XYO - add data to the main chain. Since only diviners respond to queries, they can selfishly censor valid responses, crippling or slowing down the network. Additionally, since logs with more independent Proof of Origin will have higher confidence

score, a device with multiple roles is discouraged in XYO. Furthermore, oracle [22] movement in XYO adversely affects the confidence score as well. The latter issues prevent XYO from providing a faithful interpretation of the highly mobile physical world, where each entity usually controls several identities.

3.4 Helium - A Decentralized Wireless Network

Helium is a decentralized wireless network that provides Internet connectivity to devices while allowing them to cost-effectively geo-locate themselves, without depending on power-hungry solutions such as satellite location [23]. PoL in Helium uses a combination of a novel Proof-of-Work algorithm, *Proof-of-Coverage* and *Proof-of-Serialization*.

Proof-of-Coverage continuously tries to prove if Hotspots are correctly reporting their location and coverage. To formulate Proofs-of-Coverage, challenges are periodically sent across the network, where each challenged miner responds back and also forwards the rest of the challenge after removing their layer, this process is repeated until the timeout or when the challenge reaches the final target miner. These responses, if successfully validated, result in Proof-of-Coverage for all participating miners. Proof-of-Serialization is used to establish a time consensus across the network.

Using these proof systems, Helium can generate PoLs for any device on the network. By comparing the timestamps of the reception of a data packet across different devices it can estimate the location of a device.

Although Helium sets out to provide many viable solutions, it still focuses on a hardware solution. Moreover, all the hardware devices on Helium must abide by the specifications set out by Helium and must be manufactured by authorized makers.

3.5 Blockchain Based Zero-Knowledge Proof of Location in IoT

In their work, Wei et al. propose a zero-knowledge proof of location (zk-PoL) framework to provide PoLs while protecting users' privacy [24]. It allows the users to have control over how much personal information they would like to share to obtain a service.

Users request Location Certificates from trusted Access Points (AP) which are used to generate PoLs. Users must then provide these proofs to Servers to get access to the server's LBS. Depending on the type of requested service, users can choose how much or how little of their personal information they would like to share. They implement hierarchical privacy protection, where each subsequent layer reveals additional details to the server. After a server verifies the proof, it provides the service to the user and adds a service record on the blockchain so that the same proof cannot be used again.

This however is not a perfect solution and there are a few drawbacks. To begin with, APs are assumed to be honest; however in case of a compromised AP, especially in the case where an AP is unaware that they have been compromised, adversaries can falsify proofs to

gain access to services offered by servers. Furthermore, the entire process described above suffers from significant processing delays, about 2 minutes, limiting its applicability in a highly mobile physical world.

3.6 Blockchain Based Proof of Location

Michele et al. propose a blockchain based, decentralized, infrastructure-independent PoL framework that guarantees user privacy preservation [25]. Blockchain and short-range wireless communication are the primary building blocks of their work.

They create a peer-to-peer network of primarily mobile devices that can exchange data with their neighbours over a short range wireless channel. A *Prover* gathers PoLs from its neighbouring witnesses, while *Witnesses* are the ones providing the PoLs. PoLs are generated by the witnesses as a result of successful short range interaction between the requesting prover and the witness. Thereafter, the witnesses share the generated PoL with the requesting prover. The witnesses and the prover also separately add the PoLs to the blockchain, making it available to all other participants, thereby completing a proof generation cycle.

Much like the works discussed before, this framework also has its own drawbacks. To begin with, witnesses only entertain PoL requests from previously known contacts, even if provers are present in close proximity, thereby limiting the chances of new provers having PoLs generated for them. Furthermore, there is a possibility of forks in the network, which not only imposes a memory constraint on mobile devices as they need to store the historical

transactions in order to locally resolve forks, but it also introduces momentary uncertainty in the network. Additionally, the methodology of selecting the best possible witnesses, although crucial in receiving back PoLs from contacted witnesses in a highly mobile physical world, is left ambiguous. Last but not the least, the strategy to find fake PoLs in a witness' neighbourhood, by trying to contact either the witness that generated the proof, or the requesting prover does not take into account the mobility of devices in the physical world, where devices often keep changing their neighbourhood. This might result in distrusting authentic PoLs if the prover-witness pair is not available at the time of the verification.

3.7 Vouch: A Secure Proof of Location Scheme for VANETs

Vouch presents a secure PoL framework for VANETs. It leverages the high-precision positioning capabilities and low latency inherent in 5G systems. Trusted Road Side Units (RSU) generate and distribute the PoLs to the *provers* and *verifiers* classify the received location beacons as plausible or implausible, based on the PoLs received [26].

PoL cycle starts when a vehicle that wants to prove its location, that is, a prover registers at a convenient RSU. If the registration request is valid, including the prover location in the registration request, the RSU sends an acknowledgment and starts sending periodic PoLs to the prover, which contains a timestamp, estimated location of the prover, error threshold

and confidence on the location. When a vehicle *beacons* their location, they also include the latest version of PoL received from the RSU. *Verifiers* are other vehicles that want to verify a prover's location claims. When they receive a beacon from a prover, they classify it based on the location in the beacon, location in the PoL, error threshold and confidence score, thus completing the PoL cycle.

Even though the preliminary system evaluation demonstrated promising results, there are still a few concerns that need to be addressed. Since PoL generation is dependant on trusted RSUs, the capability of vehicles to get their location verified is severely restricted in areas with limited 5G coverage. Moreover, since RSUs are trusted by design, it gives them the opportunity to fake location claims. Furthermore, since a plausibility check is performed at each received beacon, the network can be flooded with beacons, causing increased processing loads which might lead to dropped messages that can prove dangerous in safety-critical systems such as VANETs.

3.8 Decentralized proof of location in vehicular Ad Hoc networks

Vouch+ is an immediate successor to Vouch described in the previous section. Vouch+ is a decentralized PoL framework that allows any entity to be a proof provider, including vehicles and RSUs [27]. It also facilitates vehicles proving their location to others beyond

direct sensing range.

Like its precursor, the system broadly consists of *Provers*, *Proof Providers* and *Verifiers*, with similar roles. However, unlike Vouch, proof providers can be any entity that can sense a prover's location, not just limited to RSUs. When a prover wants to prove its location to other verifiers, it chooses a "trustworthy" proof provider and requests PoLs. If the request is successful, the proof provider will send a steady steam of PoLs to the requesting prover. Instead of sending these received PoLs to verifiers with each beacon, the provers only include new PoLs in the location beacons. Verifiers upon receiving the beacons, classify each beacon based on the data available in new, or stored PoLs and the location in the beacon.

Switching to a decentralized version might resolve the RSU availability issue that was inherent in the previous version. However, it introduces a few issues of its own. In the current implementation of Vouch+, the proof providers are trusted by the protocol. This can severely compromise the network if a sufficiently large population of adversarial proof providers are introduced. Furthermore, there is no alternative PoL generation strategy provided for sparsely populated areas. All these issues are gracefully handled by DPSN, during *Location Mining*.

Chapter 4

System Model and Assumptions

DPSN is a permissionless, open, decentralized framework by design; that is, there are no administrators to manage the network and anyone can participate in mining to drive the consensus. However, permissionless protocols must be carefully implemented as they are especially susceptible to *Sybil Attacks* [15].

This chapter provides a detailed description of all the assumptions that were taken into account for successfully implementing the current version of DPSN. These assumptions were not only essential in the development of a more focused solution, they also helped limit bad actors from undermining the effectiveness of the system. The latter half of this chapter is dedicated to a high-level overview of its various components and how they interact with each other to complete *Location Mining* and *Space Mining*. Finally, having discussed the underlying assumptions and a brief system overview, we move on to understanding why such

networks must be carefully bootstrapped.

4.1 Assumptions

- *There are an abundance of smart objects with short range wireless communication capabilities, distributed across the whole physical world. They can connect with others only if they are within each other's effective range.*

  We can explain how this is utilized by taking an example of a smart lamppost that also doubles as a wireless AP. The lamppost has a well-known location in the physical world. If a device can wirelessly connect to the lamppost, we can safely assume that this device is located within the lamppost's effective coverage range.

- *Spaces do not change their dimensions after being successfully added to DPSN.*

  Enabling a space to change its dimension after it has been successfully mined and added to DPSN can cause several issues. Changing the dimensions of a space might have a direct consequence on the services being offered by the space, such as storage. Additionally, Space Mining Fee is partially based on the dimensions of the space. Any attempts to try and cheat the system by altering a space's dimensions should be deemed malicious.

- *Only one challenge can be broadcast during a challenge period.*

  A challenge can be repeatedly broadcast during the entire challenge period. However, a

new challenge cannot be issued during an ongoing challenge. This prevents participants from hosting multiple meetings at the same time.

- *A wanderer can have only one Space-Time point at a given instant.*

  It should be physically impossible for a wanderer to be present at two drastically different coordinates at a given point of time. Although it is possible that malicious wanderers publish conflicting check-ins on the blockchain, it is assumed that such cases will not occur during the initial stages of DPSN.

- *Complementary functions have only two acceptable states.*

  For the current version of DPSN, all functionalities supported by spaces are encoded in terms of *complementary functions*, with exactly 2 acceptable states and well-defined transition logic. We acknowledge that there are scenarios that cannot be captured with this encoding format, such as a large space occupied by several entities not changing its current occupancy status if just a few entities move out. However, such complex interactions would introduce unwanted complexity in the initial stages, where we would like to focus on proving the effectiveness of the protocols.

4.2 System Model

DPSN has a peer-to-peer architecture with each node running the system components according to the role it plays in the network. DPSN relies on an underlay blockchain with a

very small time to finality, Algorand. We use the underlay blockchain as a tamper-resistant distributed ledger. The participants continuously log critical information to this ledger, this information cannot be changed even by the writing node - DPSN leverages this characteristic in its protocols. Additionally, DPSN utilises a logical time unit, *rounds*. Each round corresponds to the amount of time required to produce a block on Algorand, which is around 4 seconds. Furthermore, we define an *epoch* as a collection of several consecutive rounds.

The world is represented using two types of elements in DPSN: *smart spaces* and *objects*, and each one of them is assigned their unique digital handle. The objects can be moving or stationary and are called *wanderers* in DPSN. A wanderer's path of motion is referred to as a *journey*. As a wanderer undertakes a journey, it must check-in its current location to the blockchain at regular time intervals. Meaningful location is a combination of physical world coordinates and the time of the observed location; we can alternatively call them *Space-Time points*. DPSN maintains a complete log of such Space-Time points for all wanderers. For each wanderer, all Space-Time points will be unique, as time is a monotonically increasing quantity. For example, a stationary object will have the same space coordinate, but the time coordinate will always keep increasing.

The sequence of check-ins made by the wanderer is its journey. DPSN has journeys as first-class elements, that is, programs can be written on top of DPSN that would determine the valid check-ins for a journey. For example, we can require a wanderer to be in a specific

locality at all times (e.g., geofencing) by rejecting all check-ins outside the locality. This can also be used for traffic shaping by modeling journeys based on pre-defined routes or criteria, such applications would motivate wanderers to closely follow the criteria by rejecting any check-ins that do not match the requirements. Physical world scenarios, such as a police car patrolling a perimeter can be easily modeled using this approach.

Stationary objects are part of the infrastructure of spaces, such as, lampposts, cell phone towers, etc. A few such objects are chosen to be a part of DPSN as trusted elements. Further details about such objects are presented in section 4.3. It should, however, be emphasized that as a part of DPSN, the actions performed by stationary wanderers are no different from other wanderers.

The smart spaces can either be closed (e.g., a building), or open (e.g., a parking space). They can also be mobile (e.g., a bus) or stationary. A smart space is a collective of a physical space and several functionalities supported by the space. Each functionality is encoded in its respective *complementary function*, with exactly two acceptable states, and made accessible through its respective API. A verified smart space on DPSN is represented by its unique digital handle which gives authorized users access to all the functionalities supported by the space. For example, we can determine whether a space is occupied or free by using its API. Additionally, each space is managed by a Space Keeper (SK). SKs are not only responsible for managing the API, but also for initiating the process to get a space verified and added to DPSN.

Some wanderers will be recruited as Meeting Keeper (MK)s, who are responsible for holding meetings (i.e., simultaneous encounter of multiple objects in a particular space). During a meeting, its participants are close to each other, enabling them to communicate and coordinate their program execution with each other through short range wireless communication. The collective coordination can result in objects getting new capabilities or data after the meetings. For example, we can restrict the objects that are eligible to participate in a meeting. We can also provide *alibis* for all participants in a meeting. MKs generate *Proof of Meeting (PoM)s* for all wanderers that participated honestly in the meeting. At the end of each epoch, during *PoST update*, DPSN utilises the PoMs generated during the meetings, to verify the location of each participating wanderer.

To verify a space and add it to DPSN, some wanderers will be recruited as Space Miners (SM)s. During verification, SMs verify the space and all its functionalities, by interacting with it through its API and log their observations on to the blockchain. These responses are verified by DPSN for correctness, and if a space receives a threshold number of votes in favor of it, it is added to DPSN, making the space available to other wanderers and thus expanding the capabilities of the network.

Applications developed on top of DPSN can use journeys and meetings to constraint the behaviour of objects and spaces. However, they are not exclusively controlled by DPSN. Like cryptocurrencies co-existing with fiat currencies and other money transfer schemes, DPSN needs to co-exist with frameworks that manipulate the world. DPSN and the application

build on top of it are just some of the factors shaping the behaviour of objects, spaces, the people in the spaces, and the people associated with the objects.

4.3 Bootstrapping DPSN

Most wanderers that join DPSN start with a *Proof of Space Time (PoST) score of 0*. This score signifies the certainty about a wanderer's location, that is, the higher the score the more unlikely it is for a wanderer to lie about their location. Further details about PoST score and the protocol are discussed in the following chapter, Chapter 5. However, the definition provided above is sufficiently complete to further our discussion.

Apart from their role as wanderers, wanderers can also be selected by DPSN for activities that are essential to drive the consensus of the network. It is for this reason we *bootstrap* the network by adding in a few wanderers that start with a non-zero PoST score, reflecting their off-chain physical world trust (e.g., a police patrol car, a traffic light, a federal building's AP, etc.). Furthermore, when DPSN is setup, these wanderers are rewarded enough native tokens during *Initial Coin Offering*, to exclusively manage the consensus-driving activities until enough eligible MKs are created to make DPSN self-sufficient. We call such wanderers *Meeting Keepers at day 0 (MK_0)*.

Careful bootstrapping not only provides appropriate growth opportunities for wanderers, but it also limits the chances of a successful *Sybil Attack*. It can be especially devastating in the initial stages of a permissionless protocol such as DPSN, when there might not be a

definitive majority of honest participants to oppose such attacks.

Chapter 5

Location Mining in DPSN

Current PoL solutions suffer from major shortcomings such as dependency on oracles, large processing delays. In an attempt to overcome these challenges, we present a novel PoL scheme in this work. We call our PoL scheme *Location Mining*, as we are working or "mining" to verify wanderers' physical location. Location Mining can be further broken down into two broad stages, acquiring *Proof of Meeting (PoM)s* and *Proof of Space Time (PoST)* update. This chapter provides a detailed description of the entire Location Mining process and is further supported by the algorithmic representations of its stages in Algorithms 1 and 2 respectively. Figure 5.1 provides a high-level overview of the entire process. At the end of this chapter, we provide a blueprint for implementing Location Mining in the physical world.

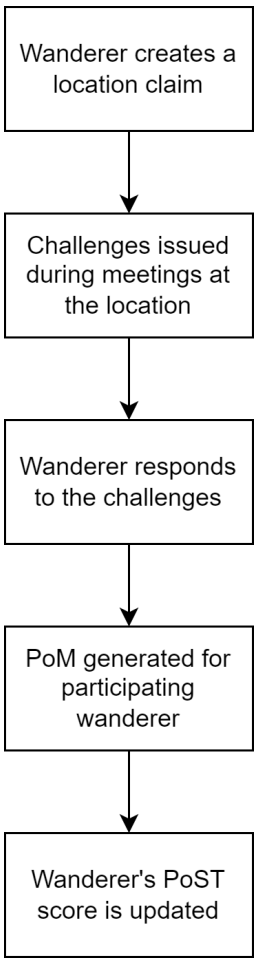

Figure 5.1: Location Mining overview

5.1 Proof of Meeting

As a wanderer moves in the physical world, it periodically makes location check-ins to the blockchain. We call these check-ins *Location claims*, as essentially a wanderer claims to own a particular Space-Time point by adding a claim to the blockchain. Additionally, each wanderer has a corresponding PoST score. PoST score is an indication of trust in a wanderer's location claims. A PoST score of 0 signifies that we do not have any confidence on a

wanderer's location claims, while a high PoST score signifies that we have high confidence on their claims. Generally, when a new wanderer joins DPSN, they start with a PoST score of 0 as the network cannot ascertain the truth of their claims based on the data available on the blockchain.

Location Mining is a perpetual and iterative process. Before each epoch, DPSN randomly selects wanderers with high PoST scores as Meeting Keepers (MKs). DPSN sends *Call for Meeting (CoM)* to all such selected MKs to signal the start of a new round of meetings in the vicinity of the MKs.

During the meetings, each MK broadcasts - multiple challenges, a nonce and a meeting ID - in its immediate vicinity using short range wireless communication. Wanderers who receive the broadcast are expected to correctly respond to the challenges and include the nonce and meeting ID with each of their responses. Including a nonce ensures that we secure the protocol against replay attacks [28]. Upon receiving the responses, each MK verifies them and buffers valid responses until the end of the meeting. When the meetings have concluded, each MK creates Proof of Meeting (PoM) for all the wanderers that responded to a quorum of challenges and adds them to the blockchain.

Challenges are simple problems that can be solved relatively quickly and effortlessly. However, a challenge is only broadcast for a short amount of time, ensuring that they cannot be transmitted over long distances and solved within the permitted time period. Furthermore, ensuring that each wanderer responds to a quorum of challenges ensures that

wanderers are not operating multiple personalities, wherein they would fail to meet this requirement. Algorithm 1 provides a step-by-step breakdown of PoM generation process.

During PoST update, the PoMs and their respective location claims are compared against each other in order to update a wanderer's PoST score. The PoST update protocol is described in the following section.

Selecting a subset of wanderers as MKs to gather and process the physical world data sets us apart from other oracle-based solutions. By adopting this approach, we can ensure that the digital handles accurately portray a wanderer's current behaviour, which would be difficult in oracle based solutions, where we assume honest participation as participants own licensed hardware components. Additionally, oracle based solutions are limited by their underlying hardware, which could severely impact the system performance in case a lot of new users join the network. However, that presents an opportunity for DPSN, as it would have a larger selection of MKs to choose from, which could improve the overall system coverage.

It is worth noting that due to wanderers' location relative to MKs, high mobility, random selection of MKs, it is possible that not all wanderers will be able to take part in meetings. However, this by itself would not negatively impact their PoST scores.

Input: W=$\{W_1, \ldots, W_A\}$; where W is the set of wanderers in DPSN.
W^+=$\{W_1^+, \ldots, W_B^+\}$; where W^+ is the set of wanderers with high PoST scores in DPSN.
LC_{W_i}=$\{LC_1, \ldots, LC_X\}$; where LC_{W_i} is the location claims made by wanderer W_i.
Output: PoMs for wanderers in valid meetings with MKs.
Parameters: C: Collateral in native token required for generating PoMs.
N: No. of challenges issued in a meeting.
Q: Quorum of challenges.
CoM: Call for Meeting.

forall *epoch* **do** `// DPSN Algorithm`
  Randomly select MKs from W^+ as MKs;
  forall *w in MKs* **do**
    Send CoM signal;
  end
end
forall *mk in MKs* **do** `// At Meeting Keepers`
  Broadcast Nonce, Challenge & Meeting ID;
end
EventListener *newResponse*:
  Check response and buffer valid responses till meeting end;
end
DPSN sends stop & aggregate signal;`// DPSN Algorithm`
forall *w in mk* **do** `// At Meeting Keepers`
  if *responded to at least Q challenges correctly* **then**
    Generate PoMs and set aside collateral C;
  endif
end

Algorithm 1: PoM algorithm

5.2 Proof of Space Time

PoST score is a measure of confidence in a wanderer's location claims. At the end of each epoch, DPSN updates the PoST scores of all the wanderers that should have had PoMs generated for them during the same epoch.

Location is only meaningful when combined with its corresponding time. We have already encountered several frameworks that introduce unwanted synchronisation delays to establish a shared sense of time. We can get around this limitation by utilising a logical time-unit, *rounds*, that is an integral part of Algorand.

At the end of each epoch, DPSN gathers all the location claims and PoMs accumulated during the same epoch. If a wanderer had a PoM generated for them by participating in meetings hosted by MKs, DPSN will try to find the corresponding location claims matching the coordinates and time of the meeting. If DPSN can find logs matching this criterion, it tries to find conflicting location claims made by the same wanderer, to determine whether they are spoofing. If no such conflicts are found, their PoST score is incremented. In all other cases, a wanderer's PoST score is decreased as they exhibited malicious behaviour by not complying with the protocol. These cases are explained as follows.

Wanderers are expected to periodically publish their location claims. If a wanderer fails do so, DPSN cannot verify the truth of their physical world location. Additionally, to safeguard against location spoofing, we compare the recent location claims of all wanderers to determine the feasibility of their location claims. Furthermore, if a wanderer participated

in a meeting, but did not have corresponding location claims in or around the meeting area, this effectively implies that they are trying to hide their location. Skipping meetings while being present in a MK's vicinity also lead us to a similar conclusion. All such actions try to undermine the protocol in some or the other way and introduce uncertainty in a wanderer's location, thus they amount to a PoST score penalty.

We acknowledge that it is reasonable if a wanderer does not always want to share their current location, due to concerns such as privacy issues, limited battery life, etc. Additionally, it would be unfair to penalize wanderers in cases such as inability to participate in meetings due to limited mobility capabilities. Although the current version of DPSN does not account for the former issue, it can be easily encoded through applications built on top of DPSN, that could add an optional *sleep* status, wherein they would not be penalized for not continuously sharing their location. This still leaves us with the issue of privacy protection, we discuss the implications of our design on privacy and respective safeguards in section 9.1. Moving on to the latter scenario, this is already handled in DPSN, as wanderers are expected to join random meetings in their current location. Constraints such as limited mobility, would only mean that such wanderers would have fewer opportunities to get their PoST scores updated.

DPSN rewards native tokens to all entities that have participated honestly in Location Mining. During each PoST update, DPSN also determines if a wanderer is eligible to receive the mining rewards. On accumulating PoMs from a minimum threshold number of unique MKs, wanderers and also the MKs that provided PoMs are rewarded. Rewards ensure

sustained incentives in truthfully reporting physical world location and participating in the network. Algorithm 2 provides a condensed view of the entire PoST update protocol.

5.3 Implementation blueprint

DPSN as a framework, gives its users a lot of flexibility. For example, it can be implemented on any DTN, wanderers could use any device to join as long it is capable of short-range wireless communication, encrypting and signing transactions and geo-locate themselves. For the best results however, we would suggest using a crypto framework with immediate finality and short time to finality, like Algorand. This section provides a blueprint to implement Location Mining in the physical world, and we assume that it is built for a metropolitan city, using Algorand.

Before we even begin holding our first meetings to verify wanderers' Space-Time coordinates, we need to bootstrap the network by adding few objects with off-chain, physical world trust as Meeting Keepers at day 0 (MK_0)s to the network, for example, police cars, smart traffic lights, APs in a government building. This selection must be made carefully, to guarantee good coverage across the city and to minimize the possibility of cheating. We also need to ensure that smart contracts, which holds regular meetings across the city and PoST updates, are up and running on the blockchain.

After successfully bootstrapping, the network will be ready to welcome new wanderers. Physical world objects, such as mobile phones, smart watches, can request to join the network

Input: W = $\{W_1, \ldots, W_N\}$; W is the set of wanderers in DPSN
$LC_{W_i} = \{LC_1, \ldots, LC_n\}$; LC_{W_i} is the location claims by W_i
$PoM_{W_i} = \{PoM_1, \ldots, PoM_m\}$; PoM_{w_i} is the set of PoMs for W_i; LC_n and PoM_m are for the latest epoch
$MK_{W_i} = \{MK_1, \ldots, MK_x\}$; MK_{W_i} is the set of MKs for wanderer W_i
Output: Updated for score for all wanderers in W.
Parameters: D : Log refresh time.
L : Location Mining Reward threshold.
C: Collateral in native token required for generating PoMs.

```
forall w in W do
    if w not in sleep mode then
        if l_{c_n} = ∅ or PoM_m = ∅ and w in meeting range then
            Reduce PoST score;
        else
            if l_{c_n} − l_{c_{n−1}} too big and currentTime mod D ≠ 0 then
                Reduce PoST score;
            else if not(PoM_m < loc > −LC_n ≈ 0) then
                Reduce PoST score;
            else
                Increase PoST score;
                if |MK_w| ≥ L then
                    Reward w;
                    forall mk in MK_w do
                        Return collateral C to mk;
                        Reward mk;
                        Remove mk from MK_w;
                    end
                endif
            endif
end
if time ≥ D then
    forall w in W do
        Clear LC_{W_i};
    end
endif
```

Algorithm 2: PoST Update algorithm

and have new digital handles created for them by paying on-boarding fees to the network. After joining, wanderers are given access to a smart contract that would help them regularly create location claims by publishing their location to the blockchain.

While moving across the city and regularly publishing their Space-Time coordinates, wanderers will also come across random short meetings, hosted by randomly selected MKs. As was detailed in the protocol earlier, wanderers are expected to actively participate in any such meetings they come across.

At the end of each epoch on the blockchain, the PoST update smart contract cross-verifies all wanderers' location claims and PoMs that were accumulated during the same epoch, and updates each wanderer's PoST score based on their observed behavior. This contract also verifies if a wanderer and the MKs that generated PoMs for the wanderer are eligible for mining rewards, and rewards them accordingly. Reward distribution logic should be carefully adjusted to keep the wanderers motivated enough to actively participate, but not so generous that they become lazy.

As more wanderers actively participate, we can expect to see a growth in the numbers of eligible MKs all across the city and the network would no longer need to depend exclusively on MK_0s for hosting meetings.

Chapter 6

Space Mining in DPSN

This research discusses several contemporary PoL frameworks and their limitations. *Location Mining* introduced a novel approach that effectively addresses many of these limitations. However, the same cannot be said for physical spaces. There is an almost inexplicable lack of frameworks that facilitate creation of trustworthy digital representation of physical spaces. Our novel approach, *Proof of Existence (PoE)*, aims at rectifying this problem.

In a digital world, even physical spaces would need to be represented by their digital handles. PoE ensures that a physical space exists at the correct location with all its advertised attributes, and all its features accessible through its digital handle. Each functionality supported by a space is encoded in *complementary functions*, with exactly 2 acceptable states and well-defined transition logic. *Space Mining* is a PoE framework. This chapter provides a closer look at Space Mining.

Space Mining has three distinct phases, starting with *Space Proposal initiation*, moving on to *Space Proposal verification* and finally concluding by *Response verification*. The output of each of the former two phases is fed-in to the subsequent phase. The proposal is either accepted or rejected at the end of Response verification phase. We begin our description by taking a closer look at the first of the three phases, Space Proposal initiation.

Before discussing a probable implementation route, we briefly discuss the limitations that need to be considered for future improvements, these must also be considered while implementing Space Mining in the physical world.

The final section of this chapter is dedicated to Space Mining implementation blueprint in the physical world, explained with a simple example of a smart car trunk, in the same metropolitan city introduced in the previous chapter, which now has functioning Location Mining.

Although we provide a clear, complete and detailed step-by-step description of the protocol. It is a fairly involved protocol, with complex interactions between several entities, due to which it could not be implemented with reasonable accuracy in the given time frame.

6.1 Space Proposal initiation

Space owners can choose to add the physical spaces owned by them to DPSN. By being a part of the network, the space, and by extension its owner can earn native tokens when it is

used by other wanderers, for example, a parking spot being rented out by other wanderers. Space owners need not necessarily be a part of DPSN, in which case spaces are represented by a distinct *Space Keeper (SK)*. Developing and managing the REST APIs for the space and creating a space proposal are a couple of the most important responsibilities of a SK.

To indicate their intent to be a part of DPSN, the SK creates and publishes an initial *Space Proposal*. This version of the proposal should contain the coordinates, dimensions and a list of functionalities supported by the space. Upon receiving the proposal, DPSN requests the SK to implement several abstract complementary functions, taking into consideration the capabilities of the space and the network requirements. One such abstract function, *isOccupied* is shown in Fig. 6.1, with exactly 2 complementary acceptable states and well-defined transition criteria.

Based on the specifications provided by DPSN, the SK implements REST API functions for all the requested functionalities, along with a *contents* function that helps in determining the contents of the space. Then, the SK sends the API endpoints for each function to DPSN.

DPSN requests a mining fee based on the complexity of the functionalities and the attributes of the space, such as, dimensions of the space. Requesting a mining fee not only discourages fake proposal creation and safeguards the network against *Denial-of-Service* attacks, but it also form a part of the reward for the *Space Miners (SMs)* that verify the space proposal.

After the SK pays the mining fee, the completed proposal is stored by DPSN, until

SMs are selected to verify the proposal, this process is described in the following section. Algorithm 3 provides a walk-through of the proposal initiation.

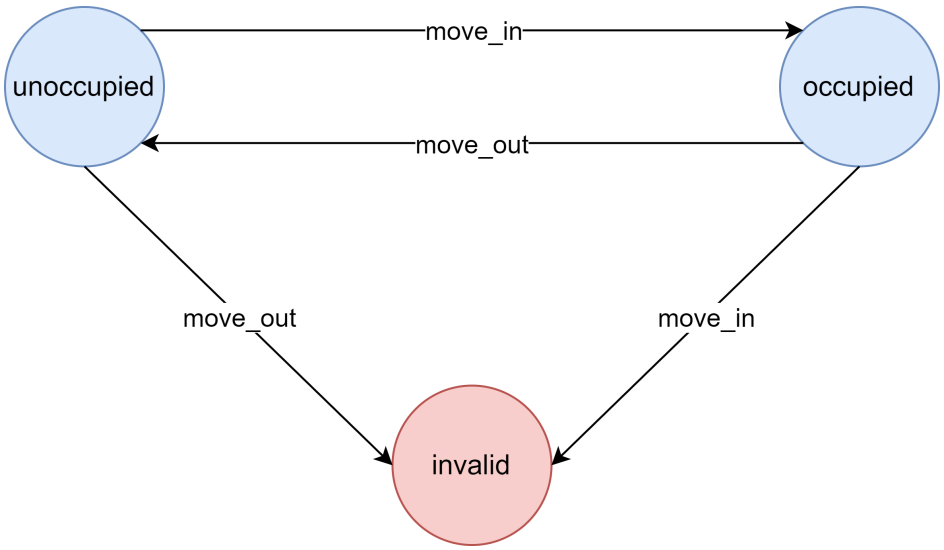

Figure 6.1: Complementary function: isOccupied

6.2 Space Proposal verification

DPSN selects wanderers with high PoST scores near the coordinates in the proposal as Space Miners (SMs). To find SMs in a neighbourhood, DPSN maintains a list of eligible wanderers in each locality, based on their PoST scores and recent PoMs.

Depending on the complexity of the proposal, the verification process is carried over several rounds. For each round, DPSN selects a subset of eligible wanderers as SMs and sends them the coordinates to the space. After moving to the requested location, the SMs request access to use the REST API endpoints supported by a space. DPSN verifies the recent

Input: SP_i: Initial Space Proposal created by the Space Keeper.
Output: SP_c: Complete Space Proposal ready for verification.
Parameters: S_x: Space x, that wants join DPSN.
SK_x: Space Keeper for Space x.
ABS_f: Set of abstract functions supported by DPSN.
M_x: Mining fee for proposal verification.
Contents API: Returns the contents of a Space.
SK_x creates a SP_i for S_x, which includes the coordinates, dimensions and supported features;
SK_x includes SP_i in a transaction and sends it to DPSN;
DPSN creates a subset, $a_{fx} \subseteq ABS_f$, based on SP_i and network requirements;
DPSN sends back a transaction to SK_x with a_{fx};
forall f *in* a_{fx} **do**
  SK_x implements REST API function with two complementary acceptable states;
end
SK_x also implements a *Contents* API for S_x;
SK_x sends back the API endpoints to DPSN;
DPSN requests M_x from SK_x;
SK_x sends M_x to DPSN;
DPSN stores the complete proposal, SP_c, on-chain for verification.

Algorithm 3: Space Proposal initiation

PoMs of the SM and grants access to the endpoints only if they have recently participated in nearby meetings. Although SMs have high PoST scores, we do not assume that they cannot feign their presence, making it physically effortful to even try to cheat.

For each round, DPSN creates a new challenge to verify a functionality. Challenges require a SM to either change the state, or report it if already changed. Additionally they are crafted in such a way that only one SM at max can change it to the expected state in any given round. We explain the verification process with the example of a simple complementary function, *open_close* which can either close a space when open and vice versa.

Let's assume that DPSN selects 3 SMs for a particular round, where it instructs them to close the space. The first wanderer reports the space to be open and closes it. The other wanderers just report that the space was already closed. This is a successful round as only one wanderer could change the state of the space. Let's take another example of a failed round. For this round, DPSN selects 5 SMs to verify the same functionality and expects them to close the space. However, two SMs report having found the space open, which should not be possible. Fig. 6.2 provides a graphical representation of this example.

We go through a few additional steps to ensure the security of the protocol and to make to make it harder for SMs and SKs to coordinate amongst each other, or even amongst themselves. To begin with, challenges are sent to each SM and are encrypted with their public key, making it impossible for others to know the challenge or other co-participants in a round, unless they are already colluding. Additionally, to prevent SMs from colluding with

each other, the space state reports are delayed until the round is over, making it difficult for other SMs in a round to refer to each other's responses and make educated guesses. SMs create response transactions reporting the observed state of the space, with a *FirstValid* and *LastValid* time in future, when the round would have ended, but do not publish it. They publish this transaction immediately after the round is over. To ensure that they have not changed their original response, we require them to publish a transaction with the hash of their response transaction, immediately after interacting with the space and creating the response transaction. Fig. 6.3 provides a simplified response transaction structure and timeline. The final phase, Response verification, performed by DPSN is discussed in the following section. Algorithm 4 provides a concise yet complete overview of the Space Proposal verification algorithm.

6.3 Response verification

DPSN waits until the end of all challenge rounds before verifying the responses sent by the SMs. To verify each functionality, DPSN searches for a minimum number of consecutive valid rounds, where SM responses match the required criteria.

DPSN creates a total order of all the received responses, and groups them in their respective rounds, based on the *challenge_txid* in their responses and the timestamp. Before checking the response, DPSN verifies if the hash of the response had already been timely published, if not, then the response is trivially rejected and the round is considered failed.

Input: SP_c: Complete Space Proposal stored in DPSN.
W^+: $\{W_1^+, \ldots, W_n\}$; where W^+ is the set of wanderers with high PoST scores in DPSN.
Output: Space interaction logs.
Parameters: C: Challenge question created at each round by DPSN.
R: Total number of rounds for verification.
$PoMs$: Proof-of-Meetings.
$FirstValid$: Round before which a transaction is not valid.
$LastValid$: Latest transaction acceptance round.
$counter$: Sequential counter present within the Space.
$(p_k, s_k)_{DPSN}$: DPSN's public-private key pair.
$(p_k, s_k)_{M_i}$: Public-Private key pair for miner M_i.
$Time(t_1, \ldots, t_y)$: Logical-time in terms of Algorand rounds.

DPSN maintains a locality based subset of wanderers with high PoST scores, $w_L^+ \subseteq W^+$;
forall *Challenge rounds in* R **do**
  Select a subset of miners, $m_x \subseteq w_L$;
  Generate a challenge, C to verify a feature;
  forall *miner in* m_x **do**
    Send SP_c to miner;
    Encrypt C with $p_{k_{miner}}$ and send it to the miner;
    Miner moves to the location in SP_c and requests access to the APIs;
    if $PoMs$ *are near* SP_c **then**
      Grant access to miner;
      Based on C, miner either changes the space state or just records the state;
      Miner creates a challenge response transaction, $resp_{tx}$, with future $FirstValid$ and $LastValid$ fields including the counter and the response, but does not send it immediately;
      Miner encrypts $resp_{tx}$ with $p_{k_{DPSN}}$;
      Miner creates and send a proof transaction with hash of $resp_{tx}$ to DPSN.
    endif
  end
  if *Challenge round is over* **then**
    forall *miner in* m_x **do**
      Send encrypted $resp_{tx}$ to DPSN.
    end
  endif
end

Algorithm 4: Space Proposal verification

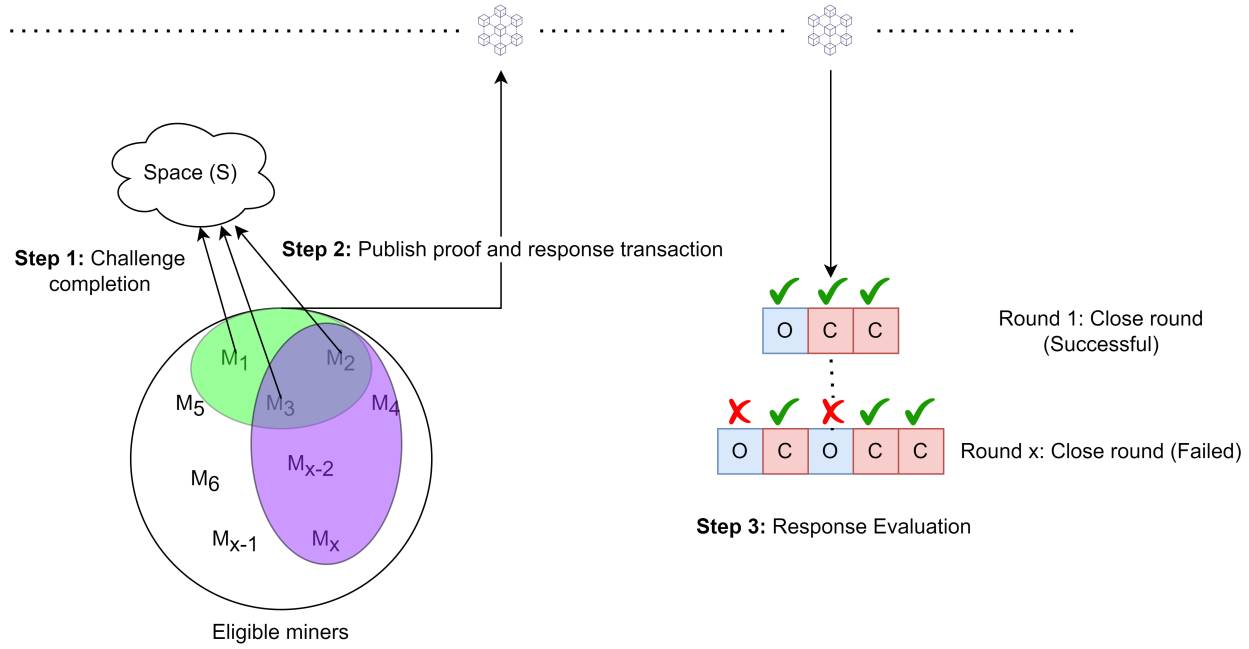

Figure 6.2: Space Mining Overview

A round is valid when two conditions are satisfied. Firstly, there should be only one expected response, while other SMs should just confirm the state change. Secondly, the state of a space should be consistent with the last valid round. There can be two possibilities when verifying these conditions, either it is the first valid round, so we assume the state to be true if the first condition is satisfied, or there already exists a previous valid round and compare the current state to it.

DPSN keeps a tally of subsequent valid rounds and the miners that participated in those rounds. If the chain of valid rounds is broken, DPSN discards the rounds it has already encountered and repeats the process until it reaches the desirable threshold of consecutive valid rounds, or encounters the *timeout*, at which point the proposal is rejected.

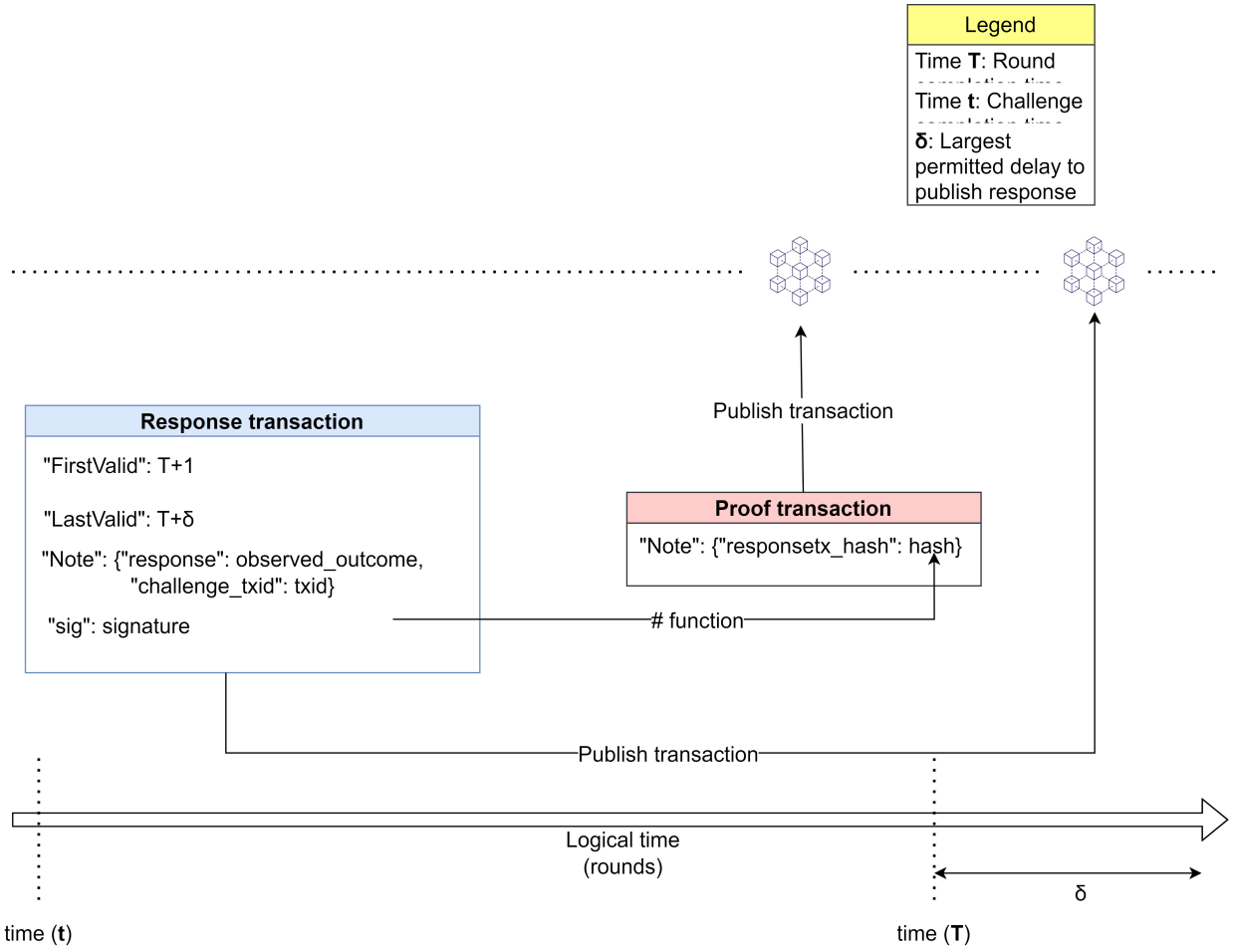

Figure 6.3: Space miner response structure and timeline

If the threshold of consecutive valid rounds is reached before the timeout, DPSN rewards all SMs the participated in the chain of valid rounds. At this point, the space is assigned a digital handle, which grants authorized users access to all its functionalities, and made available to all wanderers. Algorithm 5 provides a step-by-step breakdown of the Response verification stage.

6.4 Space Mining Limitations

Although we provide a in-depth protocol that attempts to capture and verify the physical world capabilities of smart space, there are a few limitations that must be considered during implementation and while refining the protocol.

There is a direct cost on the wanderers that are selected as Space Miners. SMs must accommodate verifying the space into their schedule, this might be quite challenging for wanderers with a busy and pre-defined schedule. To offset such costs, the rewards obtained from participating in verifying a successfully mined space need to be adjusted carefully, to ensure that the miners are compensated adequately for their efforts.

Although a space cannot be used on DPSN before it is verified and successfully mined, it could still be used by other physical world entities while it awaits verification. Presence of such entities during the verification would certainly interfere with the verification task.

Another option would be to restrict access only to miners, until the space is verified and added to DPSN. However, in such cases the space would always be aware that it is only being accessed by miners, and thus, it would put them in a position where they could leverage this knowledge by always giving the expected response to the miners.

There is one last detail that needs to be considered before we conclude this section and move on to the implementation plan. Depending on the functionality being verified, it might be necessary to restore the space back to its original state, after all the challenge rounds have completed. For example, if *isClean* is the complementary functionality being verified,

it would be inappropriate to conclude on a round where isClean returns False.

6.5 Implementation blueprint

This section provides a blueprint for Space Mining implementation in the physical world. We continue the same example of a metropolitan city from the previous chapter and assume they have a successful Location Mining system in place.

Since Space Mining is a rather complex protocol, taking a complex example of an apartment as a smart space for example, would detract from the actual value of protocol. Instead, we keep the example simple, and consider a car trunk as a smart space, which supports a single functionality *LockUnlock*, that does exactly what its name says, lock or unlock the car trunk. The car owner might have noticed the need for temporary storage in his neighbourhood. Intrigued by the fact that their unused car trunk could serve as a temporary storage, and that they could be rewarded when the trunk is used, they chose to add the trunk to the network.

To join the network, the owner must create an initial proposal, that includes the attributes of the space, such as its dimensions, location, and the functionalities supported by it. To signal their intent to join the network, they must create a transaction that includes the initial space proposal, and address it to DPSN. Upon receiving the initial proposal, DPSN invokes a smart contract that evaluates the initial proposal for correctness and requests the transaction sender to implement and return the API endpoint for LockUnlock functionality. The sender

Input: Interaction logs from miners.
Output: Proposal acceptance or rejection.
Parameters: *timeout*: Time limit to complete Space Mining.
Contents API: Returns the contents of a Space. $(p_k, s_k)_{DPSN}$: DPSN's public-private key pair.
$(p_k, s_k)_{M_i}$: Public-Private key pair for miner M_i.
$Time(t_1, \ldots, t_y)$: Logical-time in terms of Algorand rounds.
R: Total number of rounds for verification.
threshold: $2/3^{rd}$ of R rounds.

```
Set count := 0, miners := [];
if All challenge rounds are complete then
    while not timeout and count ≠ threshold do
        forall rounds in R do
            Gather all responses transactions and decrypt the data using p_{k_DPSN};
            Create a hash of the transaction data;
            Set responses := [];
            Set round_miners := [];
            if Received hash = Created hash then
                Add response to responses.
            endif
            if there is exactly one expected response in responses and it is consistent
             with last round then
                count := count + 1;
                Add miner to miners;
            else
                Reset count and round_miners;
            endif
            Add round_miners to miners;
        end
    end
    if count = threshold then
        Add space, S_x to DPSN;
        forall miner in miners do
            Reward miner;
        end
    endif
    Assign ID to the space and map all functionalities to it;
endif
```

Algorithm 5: Response verification

implements the complementary function for LockUnlock and returns the API endpoint. The contract requests a mining fee based on several criteria, like proposal complexity, space dimensions. After DPSN receives the payment from the sender, the completed proposal is stored on the blockchain.

To give each proposal a fair chance of being verified and to prevent SMs from being lazy as the proposed space is too far, DPSN selects several wanderers in the locality of the proposed space as SMs. It is of course possible to have a central database that stores the list of eligible wanderers in each locality, but that approach is vulnerable to the limitations of a centralized solution. Instead we can utilise a Decentralized Storage Network (DSN), like Filecoin, to store and retrieve that information.

DPSN verifies each functionality over several rounds, wherein it selects several nearby SMs. Each SM is expected to move close to the car trunk and interact with it through the API, with the aim of checking the specifications advertised in the proposal and the functionality being verified, in this case LockUnlock. After interacting with the car trunk, each SM publishes their observed output to the blockchain in a response transaction as soon as the round is over. This process is repeated for several rounds, and all the responses are stored on the blockchain.

When all the rounds have concluded, a smart contract to verify all the received responses is triggered. This contract creates a total order of all the received responses and organises them in their respective rounds. The responses received in each round is evaluated, until we

find a threshold number of consecutive successful rounds and the trunk is added to DPSN, or we reach timeout and the proposal is rejected.

Assuming that the trunk was successfully added on DPSN, this gives the space owner new opportunities of generating revenue. Additionally, it gives others the opportunity of utilising the car trunk for applications such as a secure package drop-off and collection point.

Chapter 7

Evaluation of DPSN

Based on the designs mentioned in the previous sections, various aspects of DPSN have been evaluated in a simulated environment. This section begins with describing the simulation setup, which contains details about how wanderers join DPSN and how it has been implemented. The rest of this section provides an in-depth explanation of the features supported by DPSN, starting with the essential discussion around throughput, PoST score growth, population coverage and ease of reward acquisition. This section concludes with an interesting discussion about Network difficulty, wherein similarities are drawn between Bitcoin [1] and DPSN, and how that has been utilised to improve DPSN.

7.1 Simulation setup

Location Based Services (LBS) [29] require location data to be available and processed at a rather fast pace, especially due to the high mobility of most physical world entities. As such, a blockchain with short time to finality plays an integral role in DPSN. DPSN is implemented on top of one such blockchain network, Algorand [30], that promises transaction finalization time of 2.5 - 4 seconds [7].

All physical world entities that would like to join our network, like any other cryptocurrency framework, will need a public key that serves as their unique Account ID, and also a private/ secret key to authorize transactions. Upon request, these addresses are generated by Algorand. It must be noted that accounts must be appropriately funded before transacting.

There are three ways to access Algorand blockchain, namely running a full node, running a quick node or interacting by means of a REST API. Depending on the application, each of them have their own advantages and disadvantages. In our case, APIs provide us with the best performance, without the additional overhead of node management. Algonode API and PureStake API have been used to implement DPSN.

The simulation is carried on a 40x40 square, virtual geospatial space. The space is then divided into 4 equal square partitions, each managed by a MK_0 placed exactly at the center of the partition. As a whole, the simulation includes 40 moving wanderers, and the 4 stationary MK_0s mentioned above.

7.2 System throughput

Wanderers regularly interact with DPSN and create several types of transactions, including:

- *Check-ins* that contain wanderer location data and must be regularly published on the blockchain, even by stationary wanderers.
- *Proof of Meeting (PoM)* created by Meeting Keepers to log successful meetings between wanderers and self.

Apart from these transactions created by the wanderers, DPSN itself creates several transactions on fulfilment of certain conditions, they include:

- *Reward* transactions that are created in favor of wanderers/ Meeting Keepers in recognition of their efforts to support DPSN, it also serves as an incentive for continued honest participation.
- *PoST update* transactions to increase or decrease a wanderer's PoST score, subject to the honesty of their self-proclaimed location claims.

To simulate long running interactions between wanderers and DPSN in the physical world, each iteration runs for 4 hours, while progressively increasing the number of wanderers in each iteration.

7.3 PoST growth

PoST score is the measure of the certainty in a wanderer's location claims, that is, wanderers with higher PoST scores are less likely to fake their location claims. A high PoST score is one of the pre-requisites to be selected as a miner. Moreover, wanderers are incentivized to maintain high scores at all times by rewarding successful mining activities.

All wanderers that join DPSN, with the exception of MK_0, start with a PoST score of 0 as the network does not know anything about their behaviour based on the data available, they can either be malicious or honest. Wanderers can attempt to update their PoST score by taking part in meetings held by randomly selected MKs across the map. During these meetings, the selected MKs generate PoMs for each honest wanderer in the meeting. While DPSN is running PoST update, it cross validates the coordinates in the location claims and the PoMs for each wanderer and updates their respective PoST scores accordingly.

In our implementation, we have chosen to update the PoST scores by a constant factor, rather than an increment rate which slows down the higher we go. Diminishing returns, that is, the latter approach, might induce lethargy or discourage wanderers to keep updating their scores once they have attained a sufficiently high PoST score. To get a better estimate of the PoST scores for highly active and interactive wanderers over a long period of time, we have adjusted this factor to be 10 times the normal rate.

Additionally, there is no strict upper limit to the PoST score a wanderer can accumulate. DPSN as a network is intended to have a diverse user base, with equally diverse skill-sets and

understanding. From a layman's perspective, it would be easier to decide between wanderers W_a and W_b with scores 15.8 and 17, rather than wanderers W_x and W_y with scores of 0.99814 and 0.99972.

7.4 Coverage control

A network such as DPSN, that can be used to support and augment *Location Based Services (LBS)*, needs to provide pervasive services. We can achieve that using coverage control. Through appropriate *coverage control* we can control the number of miners available for both location and space mining.

Controlling the number of miners selected in each epoch for *Location mining* effectively changes the geographical area where PoMs can be generated. As a result, the number of miners that get their location claims verified during PoST update also changes.

With regard to Space mining however, changing the coverage has a direct impact on the *Quality of Service (QoS)*. Depending on the proposal verification task, DPSN might choose more wanderers to ensure that the space is authentic and matches the specifications mentioned in their respective completed space proposals. It should be noted that increasing the coverage might increase the quality of the mined space, however it also increases the complexity of the proposal verification job.

Coverage control has its applications both in Reward acquisition and Network difficulty. However, due to its widespread implications it has been explained in its own separate section.

7.5 Reward acquisition

One of the main contributions of this work is Location mining. Location mining enables wanderers to have their location claims verified by joining meetings and having PoMs generated for them. At the end of each epoch, DPSN cross-verifies all the location claims and PoMs, during PoST update.

During PoST update, DPSN not only verifies the coordinates in the location claims against the respective wanderer's PoMs generated in that epoch, but DPSN also decides whether to reward all the involved parties, i.e., wanderers and meeting keepers. Wanderers receive a reward when they gather PoMs from a minimum number of unique meeting keepers. If a wanderer receives a mining reward in an epoch, all miners that aided the wanderer in generating PoMs are also rewarded.

The current version of DPSN is implemented on Algorand's testnet. In this version the rewards are handed out in algos (Algorand's native currency). Additionally, the native currency for DPSN, *Space coins* has already been implemented on Algorand. All accounts used in the simulation have also *opted-in* (Algorand terminology) to space coin asset, making them capable of transacting in space coins.

To approximately simulate various movement patterns in the physical world, the experiments have been performed in two modes. In the first mode, during each iteration which runs for 4 hours, all moving wanderers move in an unique movement patter, either circular, local (around the nearest stationary MK_0), oscillatory or random movement. In

the second mode, all moving wanderers are further sub-divided into 4 equal groups, with each group adhering to one movement type and all wanderers moving simultaneously.

7.6 Network difficulty

Bitcoin tries to maintain a block creation rate of one block every 10 minutes. It does this by adjusting the number of zeros at the start of the target hash to adjust the difficulty. This feature can also be used to resolve forks in favor of the fork with higher difficulty [13].

Network difficulty in DPSN is a measure of the difficulty by which rewards can be acquired, that is, the higher the difficulty, the harder it is for wanderers to get a reward. Although this might seem like a deterrent to DPSN's growth, it serves as an essential security feature in a network of distrusted parties.

Coverage control, introduced in one of the previous subsections can also be used to manage the network difficulty. However, it is a sufficiently large feature of DPSN and warrants a section of its own.

All of the features described in the following subsections, along with coverage control, if used appropriately have several useful applications. They can be used to manage network security by controlling reward distribution in DPSN if malicious activity is detected, or in case of a system bug. These features can also be used to naturally reduce mining rewards as and when DPSN matures, something similar to what Bitcoin does, where rewards are halved every 210,000 blocks [31].

7.6.1 Threshold control

As discussed in the previous sections, a wanderer can receive rewards during Location mining by getting their location claims and respective PoMs cross-verified during PoST update. Additionally, to receive rewards, a wanderer must have received PoMs from a minimum threshold number of unique meeting keepers.

Increasing this threshold makes it harder for wanderers to gain rewards as they need to physically move more and meetup with a larger variety of meeting keepers to become eligible to receive new rewards.

To test this feature, we gradually increase the threshold for the entire network and observe how the reward distribution changes in each iteration. As in the previous cases, each iteration here also lasts for 4 hours.

7.6.2 Randomized look-back

Randomized movement, wherein a wanderer meets the most amount of new entities is the ideal behaviour expected from most wanderers in DPSN. This behaviour is also best suited to acquire more rewards as their movement is random and there will be fewer chances of meeting the same meeting keepers again, allowing wanderers to easily achieve the minimum threshold requirement.

However, it might be possible to deceive the logic if a wanderer can correctly guess the minimum threshold of unique meeting keepers and program their movement according to the

following approach. Let's assume that the threshold for unique meeting keepers generating PoMs is L. Let's also assume that we have a total population of W wanderers. Ideally, L should be dynamic and wanderers' movement patterns should be randomized and unaffected by this threshold. If a wanderer guesses L correctly and finds a smaller subset of meeting keepers MK_L such that, $MK_L \subset W$ & $|MK_L| = L$, then they can just circle around this smaller subset of meeting keepers to get rewarded more often by traversing shorter distances.

To prevent this outcome where a wanderer can acquire comparatively higher rewards with limited physical movement, we have introduced *randomized look-back*. With *randomized look-back*, during PoST update, DPSN randomly looks-back to a number of previous rewards issued, this number is upper bounded by the the location mining threshold L. Instead of rewarding a wanderer when they've acquired PoMs from L unique meeting keepers, DPSN rewards a wanderer only if the new meetings keepers were not present in the look-back reward transactions and the net count of new meeting keepers is at least L.

To evaluate if this works as expected, rewards with random look-back has been implemented in iterations. It has also been verified using two modes. The modes are divided based on a similar approach as in Reward acquisition section. The only difference being that the rewards are distributed based on random look-back. Additionally, wanderer reward acquisition is compared to that when random look-back is not being used.

Chapter 8

Simulation Results

This section presents the findings observed during the simulation. All the following subsections follow the same general order as in the previous chapter. Section 8.1 shows the system throughput, taking into account all the different kinds of transaction in DPSN. Followed by section 8.2, which discusses wanderer PoST score growth for wanderers with different ages on DPSN. The next section, section 8.3 shows the effects of coverage control on the rewards and spatial coverage which closely relates to the availability of meeting keepers for wanderers. Section 8.4 starts by providing a detailed analysis of reward distribution and PoST updates with varying number of active wanderers and different movement schemes. Then we move on to a frequency comparison between rewards and PoST updates and how it can be leveraged. This section concludes by drawing out a relationship between average displacement and the number of rewards received. The last

section, section 8.5 is further sub-divided into two subsections with each dedicated to difficulty control in terms of threshold control and randomized look-back respectively, and their effects on reward distribution.

8.1 System throughput

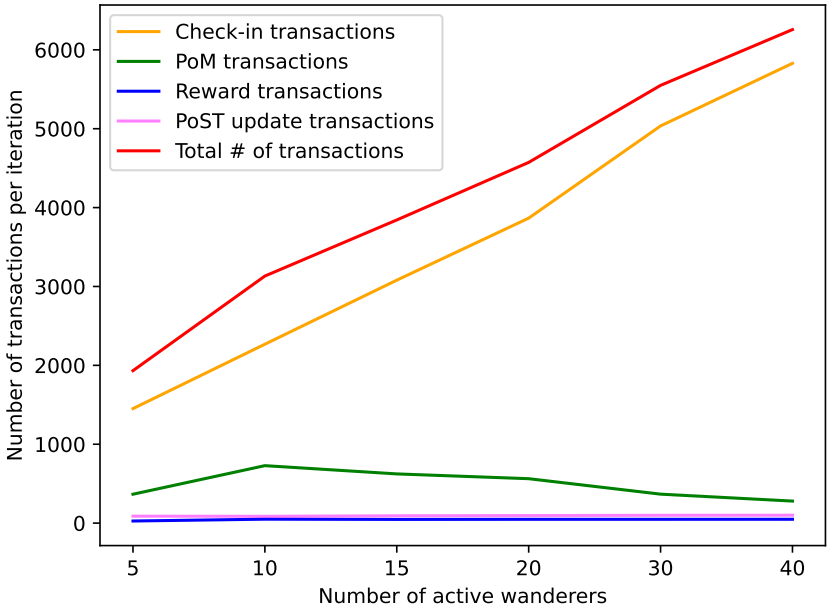

Figure 8.1: DPSN throughput

Wanderers create several types of transactions by interacting with DPSN. Additionally, DPSN also generates some transactions of its own. DPSN, being intended for devices with varied computing capabilities does not expect one single device to create these many transaction. However, the setup was intended to test the overall transaction volume of

several simulated wanderers with reasonable computational capabilities.

The results of the simulation are presented in Fig. 8.1. Throughout the simulation, there are 4 stationary wanderers and the number of mobile wanderers is slowly increased to a maximum of 40.

We can see that the total number of transactions closely follows the number of check-in transactions. In fact, check-in transactions should always be the largest volume of transactions as wanderers are expected to publish their location claims frequently.

The number of PoM transactions depends on the number of meeting keepers and the number of wanderers arriving at meetings. During each epoch, two wanderers have been selected as meeting keepers. The steady drop in PoM transaction count is not attributed to DPSN's performance. Rather, several threads contending with each other for system resources increases the time taken by relatively computationally intensive meeting threads where PoM transactions are created, reducing the overall throughput in terms of PoM transactions.

DPSN runs PoST update at the end of each epoch. While running PoST update for a wanderer, it also decided whether to reward that wanderer or not. During each PoST update, DPSN goes through all the PoMs and check-ins of a wanderer. Based on the correctness of the data, their PoST scores are updated (increase/ decrease) exactly once at the end of an epoch. The same is also corroborated in the results: we can see a gradual increase in PoST update transactions as they are equal to the number of active wanderers in an epoch.

Additionally, we can also see that the number of reward transactions is upper bounded by that of PoST update transactions.

The discussion about transaction throughput would be incomplete without discussing a bottleneck in how things work on Algorand. Although Algorand is a blockchain network with quick transaction finalization, a node must wait for about 4 seconds to receive a confirmation back from the network, which is still considerably quicker than many well established cryptocurrency networks like Bitcoin with probabilistic finality after 1 hour and about 15 mins for Ethereum [20]. Even though in the current implementation of DPSN all transactions are created on a separate thread, the thread must still wait for about 4 seconds before it can be terminated.

8.2 PoST growth

At the end of each epoch, DPSN gathers all PoMs and check-ins that were published in that epoch. It cross-verifies the coordinates published in the PoMs and check-ins for all wanderers and increases the score for plausible coordinates and reduces it otherwise. Additionally, one wanderer will receive a maximum of 1 PoST update at the end of each epoch, based on their participation in meetings with randomly selected meeting keepers.

We can observe three distinct groups of PoST score growth patterns from Fig. 8.2. The first group of 4 wanderers, are the wanderers that start with the highest PoST scores, between 11 - 16. These wanderers are about 140 days old (at the time of writing this document) and

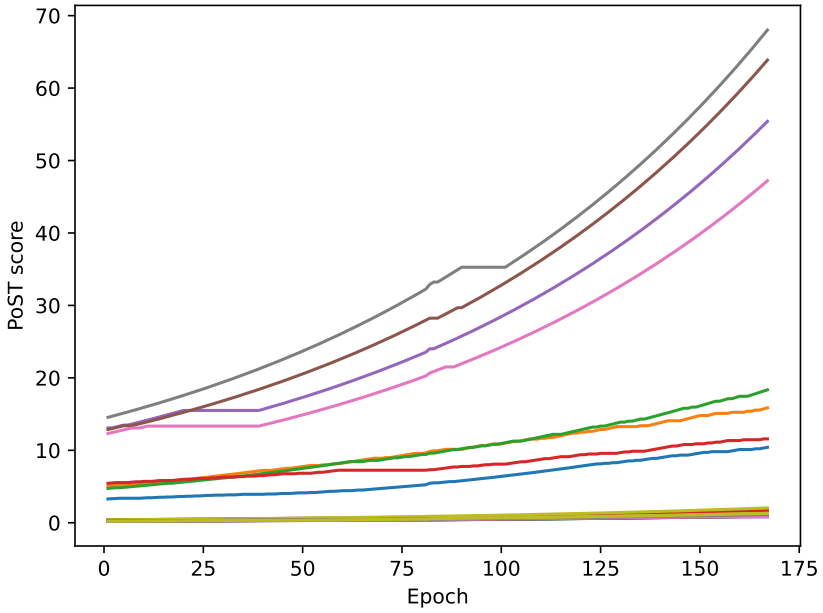

Figure 8.2: Wanderer PoST score growth

have been active ever since. The other wanderers have been active for 30 days, but they can be further sub-divided into two groups based on their activity levels. The first subgroup of 4 wanderers with starting PoST scores between 3 - 7, have been active for nearly 4 times as much before this experiment is performed, in comparison to the remaining wanderers, and thus, they start with higher PoST scores. The last remaining group belongs to the wanderers which are about 30 days old, with the least activity on DPSN, which is reflected in their PoST scores.

It is clear from the results, especially in the first group of wanderers with highest PoST scores that the PoST scores follow an exponential growth pattern, which is as expected due

to the constant update factor. We can also see that some PoST scores remain unchanged over several epochs, this happens when a wanderer did not join any meetings during those epochs and consequently didn't have any PoMs created for them.

8.3 Coverage control

Currently, at the start of each epoch, DPSN selects a random number of eligible wanderers with high PoST scores as meeting keepers. The numbers of meetings keepers to be selected is known as coverage. In future versions, coverage will change to meet the requirements of the spatial coverage of the physical world. It would try to maximize the capability of wanderers to have PoMs generated for them, while minimizing the overlapping areas between meeting keepers.

We have analyzed the effects of coverage control on two parameters, population coverage % and total number of rewards earned in an iteration, where each iteration lasts for 4 hours. The findings for the first comparison are presented in Fig. 8.3. To measure population coverage %, we start with one wanderer as meeting keeper in each epoch and slowly increase it to a maximum of 8 meeting keepers per epoch over subsequent iterations. Initially, with an effective coverage range of 10 units and one meeting keeper for a map of 1600 $unit^2$, DPSN can provide PoMs for 30% of the population, which will not be enough to sustain the network. As we increase the coverage, we see a gradual increase in the percentage of wanderers that can have PoMs generated for them, with a maximum of just under 90%

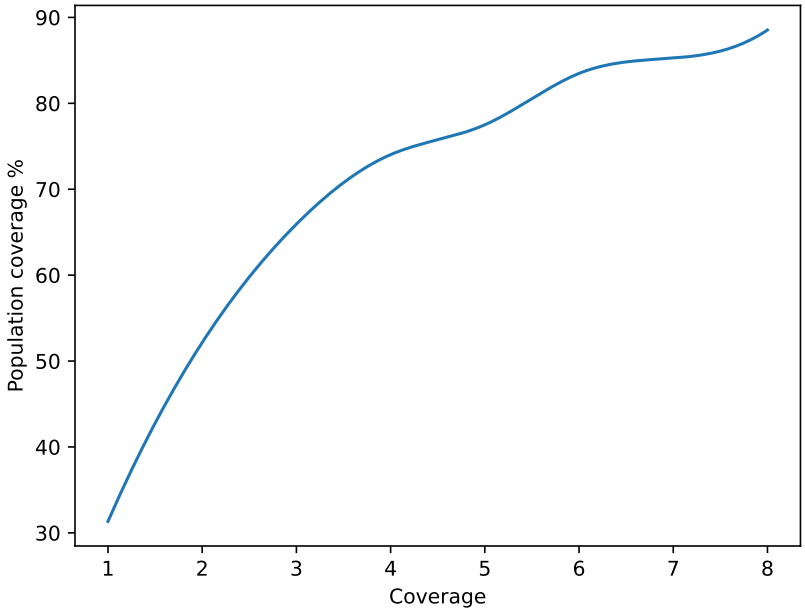

Figure 8.3: Population coverage % vs. Coverage

when then coverage is 8. We can also see that the rate of increase in terms of the population coverage % is not linear with respect to the coverage, this is caused by the overlap of areas managed by each meeting keeper. Overlaps result in lesser area being covered by the meeting keepers, effectively leading to fewer PoMs being generated.

In Fig. 8.4 we can see the effects of coverage change on reward distribution. We can see a general upward trend in the number of rewards accumulated over an iteration when coverage is gradually increased. This is in line with the expected results as increasing the coverage results in more PoMs being created by distinct meeting keepers. Additionally, during PoST update, more rewards are generated as not only more PoMs are processed, but they are also

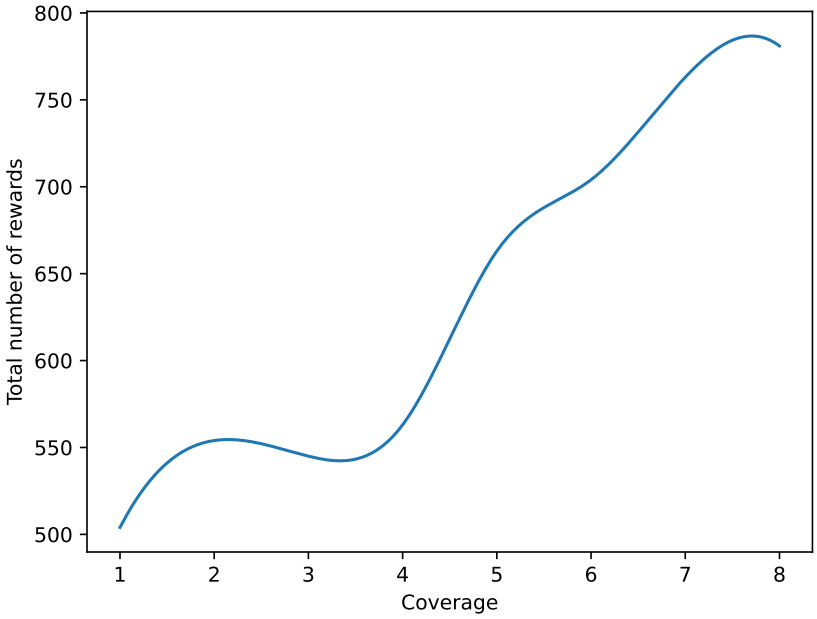

Figure 8.4: Total rewards vs. Coverage

being generated by an increasingly different collection of meeting keepers. However, we also see a slight drop in the total number of rewards, from 554 to 545, as the coverage changes from 2 to 3. The actual number of rewards and respective reward transactions is actually higher when coverage was 3, however, since the simulation was built using *daemon threads*, the simulation was terminated after a fixed time interval and the reward transactions were never published.

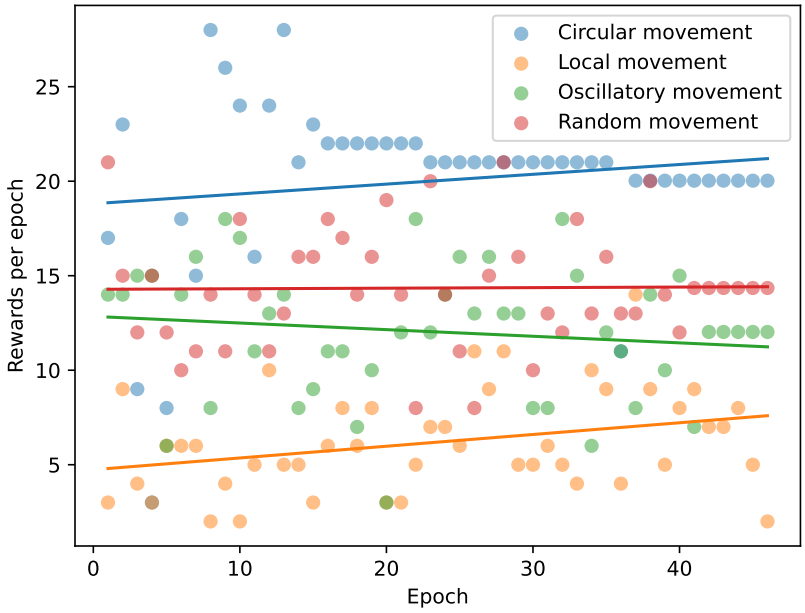

Figure 8.5: Rewards per epoch with unique movement pattern

8.4 Reward acquisition

The results in this section are from DPSN's two operating modes, unique movement mode and simultaneous movement mode. In each movement mode there are 4 movement types, circular movement, local movement, oscillatory movement and random movement. We will take a closer look at how these movement patterns affect reward distribution and PoST updates and provide a rich comparison across different movement modes. Figures 8.5, 8.6, 8.7, 8.8, 8.9 and 8.10 are supplied with lines of best-fit to aid our understanding of the data trends.

By comparing reward distribution in the two modes from figures 8.5 and 8.7, we can

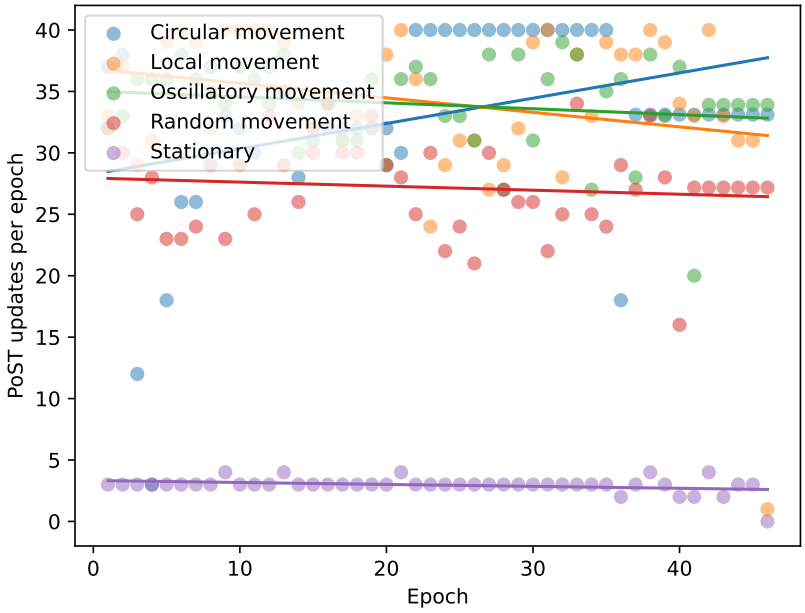

Figure 8.6: PoST updates per epoch with unique movement pattern

see that circular movement is the most rewarded movement type. This is because such wanderers uniformly circle around the entire area, centered at the midpoint, enabling them to meet many meeting keepers along the way. This might not be always desirable as random movement should be the most favored in DPSN since it should have the largest probability of meeting new meeting keepers. *Random look-back* looks into this issue and has been detailed in section 7.6.2. The least rewarded movement type, as expected, is local movement as the wanderers just move around the meeting keeper closest to them and thus, they have a small chance of meeting new meeting keepers. The next least rewarded movement type is oscillatory movement for similar reasons.

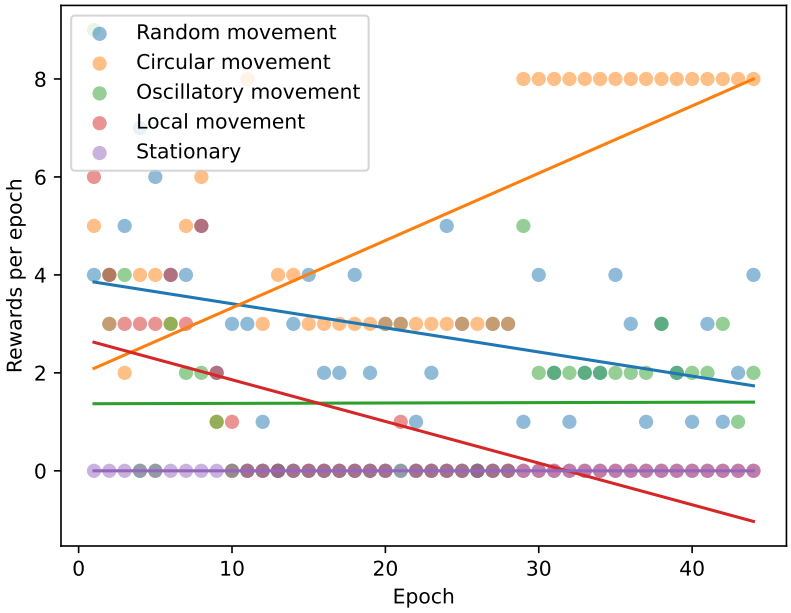

Figure 8.7: Rewards per epoch with simultaneous movement patterns

When we compare the PoST update frequency from figures 8.6 and 8.8, we can see that random movement is the lowest in both modes, while local and oscillatory movement perform better than random movement, and are almost comparable to circular movement. Local and oscillatory movement patterns are centered around one or two meeting keepers respectively, this increases the chances of having a meeting with a selected meeting keeper. Furthermore, wanderers with circular movement pattern move around the map with evenly distributed meeting keepers and always within an eligible meeting keepers' effective range. However, in case of random movement we cannot be certain about any of these; wanderers with random movement patterns can freely move in and out of blinds spots, where no meeting keepers are

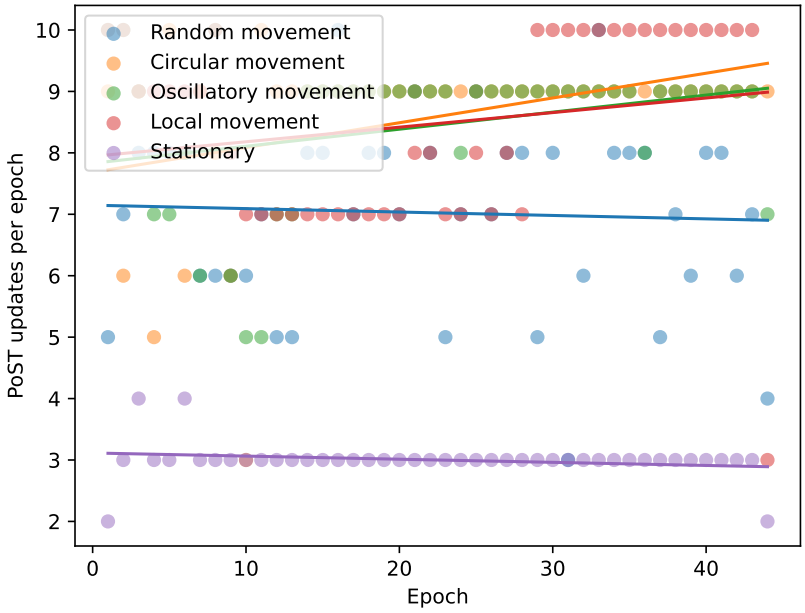

Figure 8.8: PoST updates per epoch with simultaneous movement patterns

available.

Wanderers might be rewarded during PoST update if certain conditions are satisfied. From figures 8.9 and 8.10 we can find out how often this happens. Both in unique movement and simultaneous movement modes, we can see that circular movement has the highest Rewards to PoST update ratio, followed by random, then oscillatory and finally local movement. This means that wanderers with circular movement have a shown a higher probability of getting rewarded during PoST updates, when compared to other movement patterns.

We can consolidate our findings from figures 8.5, 8.6, 8.7, 8.8, 8.9, 8.10 and present it in

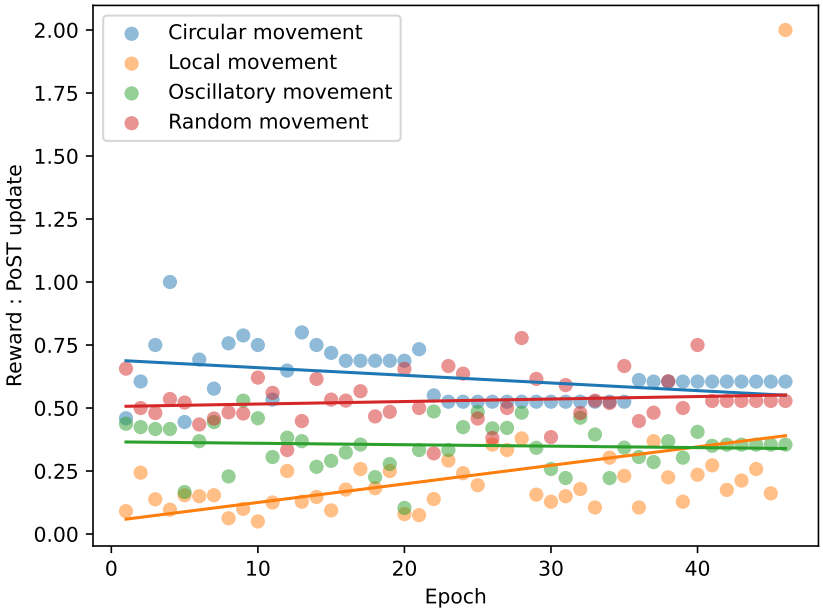

Figure 8.9: Number of rewards to PoST updates ratio with unique movement pattern

figures 8.11, 8.12 and 8.13. From the figures above, we already concluded that the wanderers with circular movement had the highest Rewards to PoST update ratio, that is, they had a higher chance of receiving most rewards in an epoch, it can be seen from figure 8.12 and 8.13. Furthermore, they almost always received the most rewards in each round, resulting in them accumulating the highest percentage of total rewards in each mode, this is shown in 8.11 and 8.13.

From Fig. 8.14 we try to establish a correlation between number of rewards earned and average wanderer displacement for different movement types. We have chosen to limit ourselves to simultaneous movement mode for this experiment. The blue lines represent the

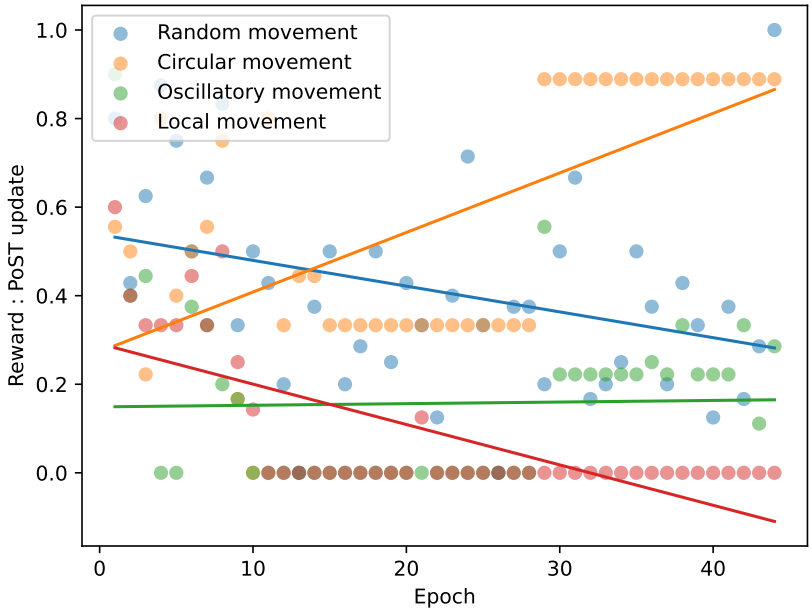

Figure 8.10: Number of rewards to PoST updates ratio with simultaneous movement patterns

averages in their respective axes. Wanderers following oscillatory movement pattern might be able to acquire a few rewards with low average displacement, but this depends on several factors such as the Location Mining Reward Threshold, which was set to 2 for this experiment and this facilitated this phenomenon, it also depends on the number of meeting keepers a wanderer oscillates or moves between, this behaviour and its countermeasure is discussed in section 7.6.2. Furthermore, sparse high rewards earned by wanderers with oscillatory movement doesn't affect the total rewards distribution, as established from Fig. 8.13. Other movement types all share the same expected results, that is, higher average displacement fetching higher rewards and vice versa.

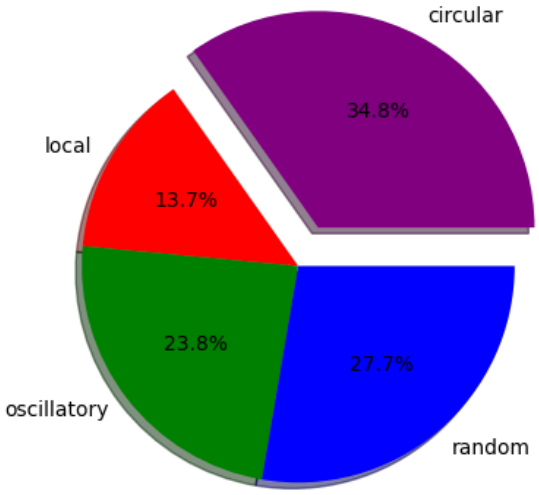

Figure 8.11: Total reward distribution with unique movement pattern

8.5 Network difficulty

Network difficulty can be used to control reward distribution across the network. Excluding coverage control discussed in a previous section, there are two other approaches to control network difficulty, *Threshold control* and *Randomized look-back*. The findings are reported in their respective sub-sections.

8.5.1 Threshold control

During PoST update, DPSN cross-verifies each wanderer's PoMs and check-ins accumulated over the last epoch. If the PoMs and the check-ins are consistent, their PoST scores are

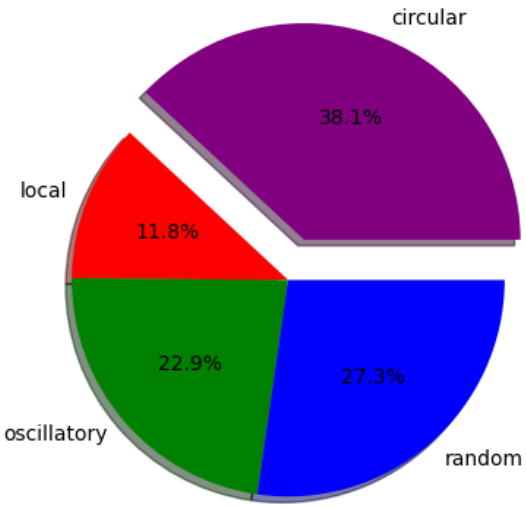

Figure 8.12: Average reward distribution per round with unique movement pattern

incremented and they also have a chance of being rewarded for completing location mining. If wanderers can accumulate PoMs from a minimum threshold number of meeting keepers, they are rewarded in a separate transaction during PoST update.

To test the effects of *Location Mining Reward Threshold* on Reward distribution, we have started with a threshold of 1, that is, wanderers will get rewarded during each PoST update. We have even tested the system by adjusting the threshold beyond the total number of unique eligible meeting keepers, which was 4 at the time this experiment was performed.

From the results in Fig. 8.15, we can see that indeed the number of PoST updates and Rewards are equal when the threshold is set to 1. This is explained by the fact that

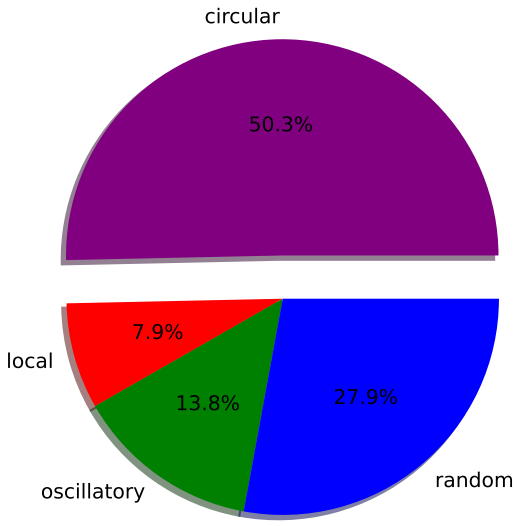

Figure 8.13: Total reward distribution with simultaneous movement patterns

wanderers need at least 1 meeting keeper to have PoMs generated for them. So, when the threshold is set to 1, each PoST update will also result in a reward. As we keep increasing the threshold, we can see that the rewards are almost halved with each increment, up to the threshold of 4.

Adjusting the threshold beyond the total number of eligible meeting keepers in the network essentially denies rewards to all wanderers as they will never be able to accumulate PoMs from sufficiently distinct meeting keepers, even if they meet with every possible meeting keeper. This will continue until new eligible meeting keepers emerge in the network such that their numbers cross the threshold, or quite simply if the threshold is

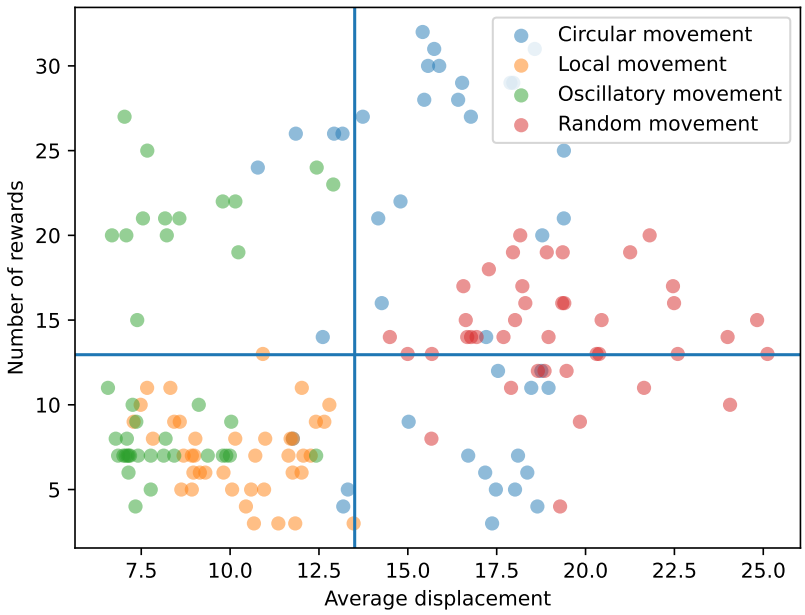

Figure 8.14: Total rewards vs. Average displacement

reduced.

8.5.2 Randomized look-back

Ideal behaviour for earning rewards in DPSN would be randomized movement in the physical world such that wanderers do not often meet each other, even if they do so, it should be after a fairly long time interval. However, if a wanderer can correctly guess the threshold, they can deceive the system to be rewarded more often by programming their movement. This is detailed in section 7.6.2.

From Fig. 8.16 and 8.17 we can see that wanderers with circular movement and random

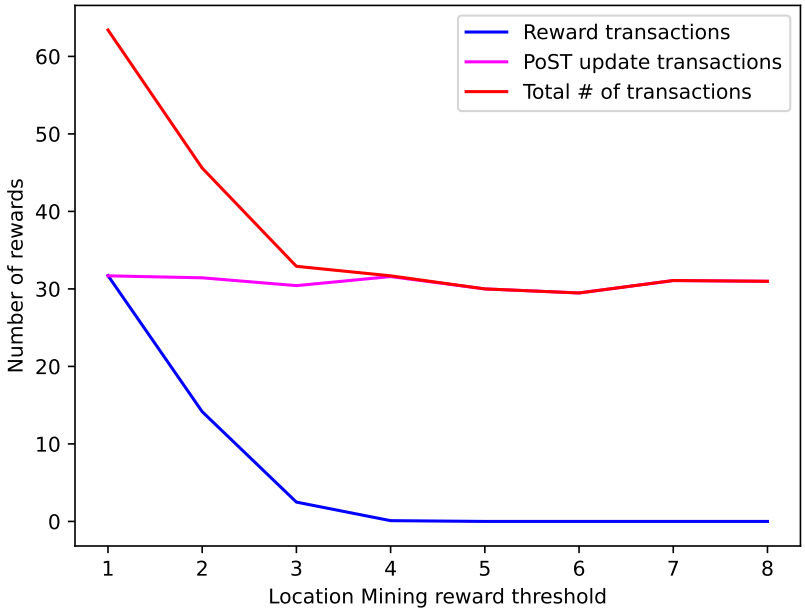

Figure 8.15: Location Mining reward threshold control

movement receive the most rewards, both during unique and simultaneous movement, followed by oscillatory then local movement patterns.

There are very few interesting findings that can be observed from just these two plots, but if we compare reward distribution with and without *Random look-back*, we get a better understanding of how it affects the network. When figures 8.5 and 8.7 are compared to their respective counterparts while using random look-back, we can see that the number of rewards for each movement type in both cases has decreased. This is due to the fact that while random look-back is being used, a wanderer only receives a reward if the new meeting keepers are not present in the look-back reward transactions and the net count of unique

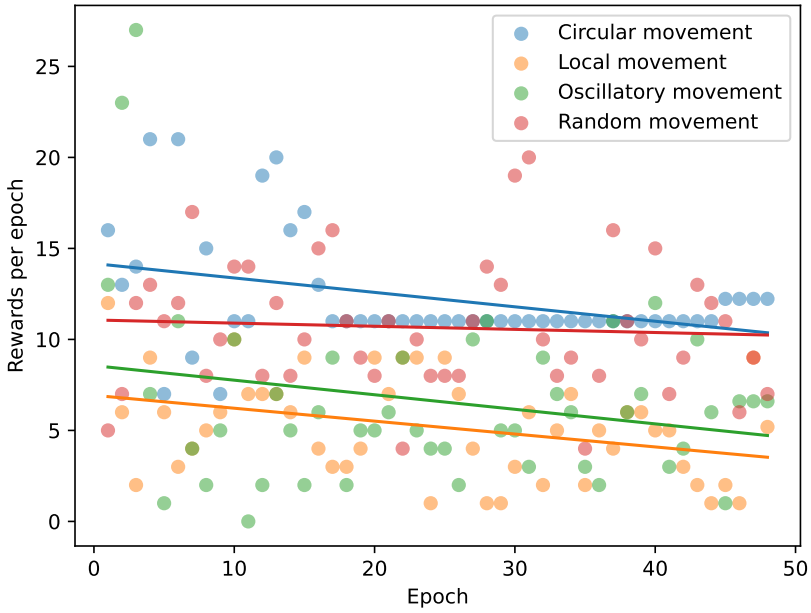

Figure 8.16: Rewards with unique movement pattern and random look-back

meeting keepers at least equals the threshold. This imposes a stricter condition than just the Location Mining Reward Threshold, resulting in fewer rewards, irrespective of the movement types.

Furthermore, we can see that using this approach, random and circular movement result in similar rewards. This is a big difference from the simulation results without random look-back where wanderers with circular movement were rewarded the most, irrespective of unique or simultaneous movement. This supports the argument for ideal movement behaviour in DPSN.

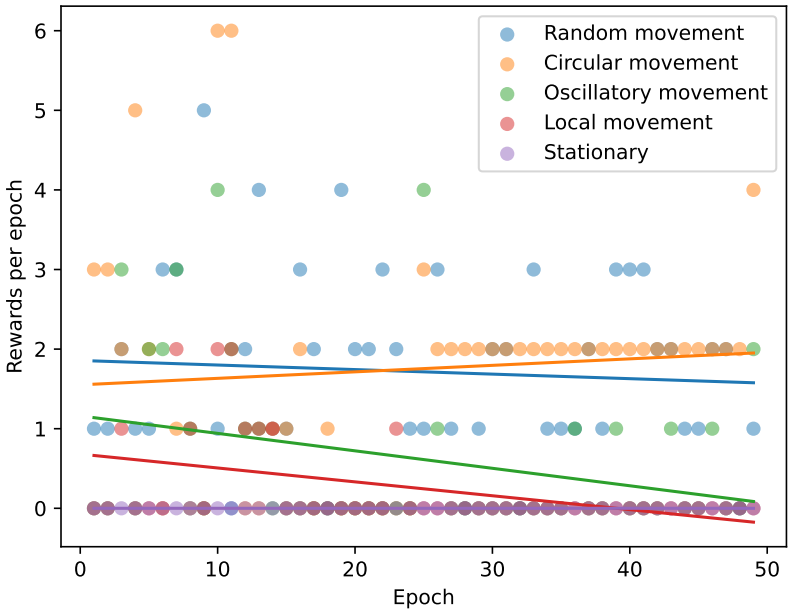

Figure 8.17: Rewards with simultaneous movement patterns and random look-back

Chapter 9

Security Analysis

Cryptocurrency frameworks as susceptible to various attacks, if not designed carefully. Generally, malicious actors try to attack such systems for monetary gains. Double spending, blockchain forks and Sybil Attacks are the most common issues in such frameworks. In a *double spending* attack, an attacker tries to spend the same token twice. Karame et al. provide a compelling study of double spending attack on Bitcoin [32]. Blockchain forks are not inherently problematic for all types crypto frameworks, however, they can be quite problematic when money is involved. Algorand is a cryptocurrency framework with immediate finality and no forks, so in DPSN, we don't have to deal with these attacks. Although Sybil Attacks are possible, it is gracefully dealt with in our implementation. This chapter provides a detailed account of several possible attacks that could be launched against DPSN, and how they are overcome. Due to the fast-paced

nature of the physical world, it would be impossible to include all attack vectors. Instead, we focus on the attacks that we believe would have the greatest impact on the network.

9.1 Privacy Protection

On DPSN, wanderers must regularly log their physical world location on the blockchain, by creating location claims. As the final step in Location Mining, during PoST update, PoMs are cross-verified against their respective location claims.

Apart from a wanderer's public key, there are no other public identifiers of a wanderer. Public keys however, are just pseudo-anonymous identities. If proper care is not taken, a wanderer's physical world identity can be revealed. This can have severe consequences as DPSN maintains a complete log of a wanderer's whereabouts. Predictable behavioural patterns, such as following the same path everyday, meeting with the same entities at fixed times, etc. facilitate such identity leaks. Furthermore, if a person is managing multiple identities on different devices, an attacker might be able to associate all the pseudo-anonymous identities to a physical world entity [33].

To prevent such attacks, it is advisable to regularly change into new identities on DPSN. Exchange services can be utilised that transfer the assets held by the old identity to the new one, including their PoST scores. We can implement a simple exchange smart contract by asking for the passphrase provided to a wanderer during account creation. If the contract can generate the public and private key using the passphrase, we can assume that the address

owner is requesting a new identity and then transfer all the assets to a new identity.

9.2 System Latency

Crypto frameworks, in general suffer from latency issues, for example it takes about an hour, or at times even more to finalize a transaction on Bitcoin [34]. Such latency is highly undesirable in a system that aims at capturing real-time events in the physical world. Let's explain the issue at hand with a hypothetical example, where DPSN is implemented on a Blockchain with a large time to finality. A wanderer's PoST score is updated at the end of each epoch, based on their behaviour. For PoST update to run, we would need to wait for all the PoM transactions to be finalised. We would also need to wait for an additional round to finalize the PoST update transactions. This could mean introducing several hours of delay for each update, based on the choice of the underlying blockchain. During this delay, all participants would be forced to either defer all their decisions for a several rounds, until the transactions are finalized, or they would need to take risks and operate with possibly erroneous values.

Algorand is a blockchain with immediate finality, that is, a transaction cannot be nullified after being accepted due to issues such as forking, and small time to finality of about 4 seconds [7]. Thus, we reduce the decision latency as much as possible. Additionally, the current version of DPSN is designed to verify recent historic location data provided by wanderers. Taking these two factors in consideration, current system performance is well

within expected bounds.

9.3 Sybil Attack

In an attempt to undermine the consensus driving mechanism of a crypto-framework, a handful adversaries can forge multiple identities. This is known as *Sybil Attack* and is a serious concern in crypto systems [15]. If left unchecked, the majority of the network can be driven by such attackers, enabling them to maliciously drive the consensus and create fake transactions that cannot be refuted due to the lack of an honest majority.

Wanderers drive the consensus on DPSN, by taking part in Location Mining and Space Mining. However, only wanderers with high PoST scores are selected as MKs during location mining, and SMs during space mining.

PoST score is a hard earned resource, as it requires a lot of physical movement in the physical world. As in Bitcoin, for a successful Sybil Attack, that is the 51% Attack [35], the attackers need to control at least 51% of the total hashing power of the network. This sheer amount, makes it almost impossible to successfully launch such attacks. Similarly, in DPSN, the attackers will need to control a good majority of wanderers with high PoST scores to drive the consensus. Although we cannot establish the exact number without further analysis, not only would reaching this number be extremely difficult due to the amount of physical movement it entails, but it would also be counter-intuitive for attackers to invest such heavily in a network they choose to devalue.

9.4 Location Mining: Collusion Attack

Native tokens are rewarded to wanderers and MKs for completing Location Mining. A wanderer, and consequently all the MKs that generated PoMs for the wanderer are rewarded when the wanderer meets a minimum threshold number of unique MKs. At first glance, it might seem like a viable strategy to form a small group of unique MKs and wanderers, such that they can always be rewarded for colluding.

Wanderers must participate in meetings hosted by randomly selected MKs. During the meetings, they must also respond to a majority of the challenges set by the MK, to have a PoM generated for them. Assuming that both the wanderer and the MKs are colluding, this can be achieved easily. However, while creating PoMs for wanderers, a MK must set aside a collateral that is returned along with a reward, if the wanderer acquires PoMs from a minimum threshold number of unique MKs. Since this threshold is randomly selected for each epoch, a group of colluding attackers can never be sure that they will get the rewards, much less their collateral. Furthermore, *Randomized Look-Back*, introduced in section 7.6.2, ensures that the unique set of MKs is not repeated in the recent past, further reducing the chances of a successful attack. With these measures in place, we can conclude that chances of such attacks being successful are quite small, and even if successful would result in small momentary gains that will quickly be overshadowed by the required collateral.

9.5 Location Mining: Multi-Personality Attack

A wanderer at the location of a meeting can try to have PoMs generated for wanderers that are outside the effective communication range of the hosting MK, or much simply they can try to manage multiple identities on a single device. In order to have PoMs generated for a wanderer, they must correctly respond to a majority of the challenges broadcast by the host MK during the meeting. Attackers can either try to respond by quickly switching between identities, or they can broadcast the challenge to the remote wanderer and send the response to the MK as soon as they receive an answer from the remote wanderer.

Since the challenges change in quick succession, attackers adopting any of the approaches discussed above will find it hard to respond to a majority of challenges. Additionally, the minimum threshold of correct responses is randomly decided at each meeting, so there is no way for a wanderer to know the minimum threshold unless both the wanderer and MK are colluding, this scenario has already been discussed in the previous section 9.4. In both approaches, attackers and co-conspirators will either have several missing PoMs or will have conflicting location claims and PoMs, resulting in a PoST score penalty.

All things said, appropriate challenge construction to resolve such attacks depends on several metrics such as complexity of the challenge, communication speed between peers, etc. It would be difficult to comment on the effectiveness of the current strategy and we will have to defer to future work that carefully constructs appropriate challenges and demonstrates their effectiveness in preventing major attacks.

9.6 Space Mining: Denial-of-Service Attack

Fake proposals can be created by attackers in an attempt to cause a *Denial-of-Service* in DPSN, with respect to Space Mining. SMs must physically travel to a proposed space's location, in order to complete verifying the proposal. By creating a large numer of convincing fake proposals, it might be possible that during certain time-periods, there are no free SMs left to verify authentic space proposals, as they are busy with fake ones, thereby causing a successful Denial-of-Service.

Such attacks will be infeasible and rather unprofitable for any attacker or group of attackers. To begin with, DPSN requests a substantial mining fee from the entity publishing the proposal, based on the complexity of the functionalities and the dimensions of the space. Even if we assume that attackers are willing to spend this amount just to create uncertainty and instability in the system, the mining fee will be distributed amongst the miners that helped remove fake proposals from the network, thus incentivizing honest participation.

9.7 Space Mining: Remote Access

In order to begin verifying a proposal, the space and its functionalities, a Space Miner (SM) must physically move close to the proposed space's coordinates and request access to the APIs supported by it. SMs can try to cheat by requesting access to the space and the APIs

when their physical world location is not in the immediate vicinity of the proposed space.

DPSN has several safeguards in place to prevent such scenarios. To begin with, SMs are selected from wanderers with high PoST scores, that is, they are less likely to lie about their physical world location. Additionally, for each round, DPSN selects SMs based on the proximity to the proposed space, so moving close to the space should not present a big challenge. Even so, considering the high mobility inherent in today's movement patterns, it is possible that SMs could have moved away before the verification began. In such cases, SMs would have to move close to the space before requesting access. DPSN verifies this by checking the recent PoMs generated for the SM by other Meeting Keeper (MK)s.

9.8 Space Mining: Remote Verification

We have argued in the previous section, section 9.7, how it would be difficult for Space Miners (SM) to gain access to the space and its API remotely. As an extension to the previous attack model, an attacker could gain access to the space and its APIs, but feign active participation by casting a random vote *remotely*.

We prevent such attacks from happening by verifying the PoMs generated for the SM by other MKs, when they were interacting with the space, and also when they completed the verification and created a response. If the PoM location supports the response location, that is, the PoMs were created in and around the proposed space, we accept the response. Otherwise, the round is discarded and the SM is penalized accordingly.

9.9 Space Mining: Collusion Attack

In-spite of all the measures discussed against cheating in Space Mining, we can only provide probabilistic assurances that such attacks would be mostly unsuccessful. In case of a successful attack, the worst case scenario would be when a fake space is advertised as a real physical space on DPSN. A fake space can either be a completely fake space that does not exist in the physical world, has restricted access or spaces that do not have the advertised capabilities. Although not entirely unavoidable, the success rate of this attack would be negligible and would certainly be unprofitable for the attackers, considering the degree of coordination and amount of physical movement it requires.

For a proposal to be verified successfully, several successive rounds of correct responses must be recorded. In order to achieve that, all responses in a round must not only match what DPSN expects it to be, it must also be consistent with the previous successful round's output state. It is assumed that the proposal creator's coordination will be needed to successfully execute this attack. Let's also assume the worst case scenario, where there is a clear majority of attackers in a neighbourhood and only attackers are selected for the first round. In this case they can generate any correct combination, as long as it does not conflict amongst each other and what DPSN is expecting, it will be accepted by the network. We can further go on to assume that this continues for several rounds, where the responses are carefully crafted such that DPSN accepts it. However, as soon as at least one honest SM is selected, things become challenging for the attackers.

The honest SM's responses will always reflect the current state of the space, of course if the space exists. If it does not, the honest SM can respond accordingly and all the responses from the previous round will be discarded. Assuming that the honest wanderer cannot trivially reject the proposal, they will respond with the correct state of the space, reflecting missing, limited or incorrect capabilities. Since the responses are encrypted and addressed directly to DPSN, other attackers cannot match their responses to that of the honest SM to make it a successful round, resulting in previous rounds always being rejected as soon as a honest SM is selected in any round. This mechanism mimics the *veto power* held by powerful institutions in the physical world.

We can conclude from the description we have seen so far, that such a scenario would be highly unlikely, and would require impeccably coordinated physical movement and dumb luck. This attack does not provide any substantial benefits for the colluding attackers, as the mining fee will simply be distributed amongst them, that is, there will be no monetary gain. They can only expect minor gains from swindling the space users for a few epochs.

A wanderer needs to pay before getting access to a space and its API. By adopting a *pay-as-you-go* model we can limit damages from such scenarios. Additionally, the wanderers can report such spaces. If a sufficient number of complaints are received, DPSN can remove the space and penalize all the wanderers that accepted the proposal.

Chapter 10

Applications

Location plays an integral part in our day-to-day lives, from finding the best route to our offices, to finding the trendiest restaurants in the neighbourhood. However, most Location Based Services (LBS) depend exclusively on GPS data, that can easily be *spoofed*. PoL frameworks provide us an alternative by enabling verification of location shared by any device on the framework. However, current PoL services do not always provide us with the best possible solution and are not suitable for physical world locations with limited coverage. Additionally, proprietary solutions lack the flexibility of being utilized in creating something new and useful. Apart from Location Mining, DPSN also supports Space Mining, giving us the first-hand opportunity to create trustworthy digitized spaces and extend the list of possible services that can be supported by DPSN.

The possibilities of creating useful applications on top of DPSN are numerous, certainly

far more that can be reasonably covered in this work. We limit ourselves by providing a non-exhaustive list of possibly interesting applications of Location Mining and Space Mining.

10.1 Traffic shaping

In chapter 4, we introduced *journeys* as a wanderer's path of motion, on which it is expected to regularly create location claims. We have also hinted upon the possibility of building applications built on top of DPSN that leverage journeys. In this section, we discuss several interesting applications of DPSN that can alter the wanderer traffic based on the application requirements.

Geofencing produces targeted results for a device, based on its position with respect to a virtual boundary in the physical world. By cross-verifying location claims made by a wanderer and the PoMs created for it, DPSN can ascertain the wanderer's location with a high degree of confidence. Current build of DPSN already implements geofencing, by rejecting location claims made outside a certain area. But, we can have different reaction strategies based on a wanderer's location. For example, we can easily achieve results similar to *Electronic Monitoring* and alert relevant authorities when a device steps out of a boundary, or stops updating their location.

New businesses dependant on a minimum volume of foot-traffic, such as grocery stores and restaurants, find it challenging to get kick-started. People are generally reluctant to try out new things, even more so, after the economic stress introduced over the last couple

of years. DPSN makes it possible to help these businesses. This can be easily achieved by encouraging wanderers to have PoMs generated from within the premises of the new business, by opportunistically selecting MKs, thus encouraging them to try it out. Although this provides us an interesting opportunity, it should be noted that DPSN was never intended as a replacement for skillful business planning.

10.2 Treasure Hunt games

Niantic, introduced us to an amazing Augmented Reality (AR) experience that leveraged physical world location and created a memorable experience through Pokémon GO. Since then, several AR games leveraging the same concept have showed up. These games suffer from several issues when physical world location is not accurately reflected in-game, that is, *Location Spoofing attacks*. With DPSN's approach to PoL, we can easily thwart such attacks. We do not delve deeper into this issue here as we have already analysed it in chapter 9. Instead, we shift our focus to some innovative solutions to everyday physical world problems, which we have tried to model as treasure hunt games.

One such application is *tours*. A tour can be individual, undertaken by a single wanderer, or in a group, undertaken by several wanderers. While journeys enable us to program the finer movements of a wanderer, using tours, we intend to program the sequence of meetings the wanderers have to attend. Tours can include different decision points at meetings, allowing wanderers to take different tours based on the program they

are running. By touring in groups or individually, wanderers can acquire new capabilities. We can draw several similarities between this approach of solving problems and *Raid parties* in Massive(ly) Multiplayer Online Role-Playing Games (MMORPG), where multiple players join forces opportunistically to complete objectives, take down bosses, etc. Food and supply distribution across different shelters would be a viable use case. With the increasing rate of poverty, coupled with the general lack of funds to sustain Non-Governmental Organizations (NGO), small organizations lack the resources to bring about any measurable differences. By appropriate implementation of tours, we can group several smaller NGOs in such a way that they can coordinate with each other, and have a much more positive impact on the lives of disadvantaged people. We can also control the order of the tour, based on the urgency and needs of shelters. Furthermore, we can support the cause by adjusting the density of meetings around these institutions, so that people are encouraged to travel by, in an attempt to destigmatize population at-risk.

Up until now, the applications we have discussed in this section leverage Location Mining. Now, we discuss an application, where Location and Space Mining are used in conjunction. Advertisements through billboards, flyers and coupons are still a major way of reaching potential customers. However, the problem with these strategies is the lack of metrics to suggest the effectiveness of the campaign. By monitoring the results from the supported APIs and the PoMs generated from within a space where the advertisement has been posted, we can gather several useful metrics, such as, the duration of the visit, peak hours, number of

visitors, etc. These metrics can be utilised by advertisers to generate targeted adverts that should be better-received by the visitors. In this case, the adverts and coupons itself can be treated as treasures that are collected by unsuspecting wanderers.

10.3 Drone-enabled delivery system

Contactless delivery using drones might become the preferred mode for receiving packages. However, drones generally have short flight distance due to limited battery life. The key components of DPSN can easily be leveraged to overcome this limitation and create a drone-enabled delivery systems.

DPSN maintains a list of wanderers with high accumulated PoST scores for each locality. By selecting such wanderers, or drones in this case, we can ensure the drones reliably broadcast their location when they go about delivering packages. We can also manage the package handover, or the final delivery using the spaces registered on DPSN. For example, a delivery drone can unlock a temporary storage space such as a car trunk, if authorized by DPSN. At a later point of time, the next authorized delivery drone can unlock the same storage and move it to the next checkpoint.

By going through a series of such handovers we can complete much more complex and long-distance delivery jobs, than what would have been possible with a limited number of drones owned by a single organization. Additionally, both the delivery drones and storage spaces can be selected opportunistically, this implies that the delivery agency that initially

accepted the delivery job need not necessarily own all the drones, nor the storage spaces that were utilized to complete the delivery. The agency can make this selection based on several factors such as the carrying capacity of the drone, the safety features offered by a storage space, or the *rental cost*. If this gains enough momentum, we can expect an equally large and competitive market for drone and storage space rentals, which in turn will facilitate cost-effective and profitable solutions for both the suppliers and the consumers.

10.4 Accountable Roaming and Alibis

Many tasks in physical spaces are carried out by roaming agents. For example, security guards secure premises to ensure that everything is in order and no suspicious activity is taking place within the premises. By verifying the location claims and PoMs of a wandering guard, we can determine whether the entire area is getting appropriate coverage.

Another similar, but arguably more useful use case of DPSN is *alibis*. Alibis are powerful tools that can be used to prove a party's innocence. DPSN provides us several opportunities to generate powerful alibis. During PoST update, DPSN already ensures that wanderers cannot be at two places far apart by checking the location claims and PoMs. Additionally, if a wanderer has participated in the network as a Space Miner, we can place the wanderer next to, or within the space during their interaction with the space, with a high degree of confidence. Furthermore, meetings with Meeting Keepers at day 0 (MK_0) solidifies a wanderers alibis as MK_0s are trustworthy physical world objects, such as an AP

in a government building.

10.5 Supply chain verification

Many items go through lengthy journeys, changing hands several time, before reaching us. Often, these items lack appropriate traceability. This issue is pervasive both in common products like chocolate, clothes, to much more rare and expensive products such as diamonds. Governments across the world are trying to find a good solution to this issue. Through DPSN, we provide a simple and effective solution.

To kick-start DPSN, Meeting Keepers at day 0 (MK_0) are selected based on their off-chain physical world trust. Routers in a government building, a lamppost with short range wireless communication capabilities, are a few good candidates to be selected as MK_0s. The network eventually grows to have more trusted wanderers as well. So, just by meeting with such wanderers along their journeys, DPSN becomes capable of keeping a track of how the object has been moving across the world. Since these records are tamper-resistant, we can always refer back to a product's path, right up to where it joined DPSN.

10.6 Space transformation

We can facilitate actions that take place in a verified space, transforming the space in some pre-defined acceptable form. A Space Owner, Space Keeper or the wanderer who has

reserved a space can post a transformation job on DPSN, detailing the job specifications. If a wanderer accepts the job, they are given access to the space so that they can complete the transformation. If the job creator is satisfied with the transformation, the wanderer who completed the transformation is paid according to the job specification.

We explain this with an example of a fenced, but uncovered parking area which is covered with snow, due to a recent snowstorm. The owner creates a cleaning job and posts it on DPSN. Nearby wanderers can find the job postings and accept it if they are capable of completing it. After the job is finished, the owner can verify the completion status and release the rewards if they are satisfied.

10.7 Space reservation

One of the more simpler, but approachable applications of DPSN is that wanderers can rent out verified spaces. Wanderers would be able to look-up verified spaces in their locality and request access through DPSN. Based on the requested capabilities and duration, Space Owners either reject the request or request an appropriate rent. Upon receiving the payment, the space, along with the requested functionalities are made available to the wanderer for the requested duration. Parking space, apartments, etc. can be rented using this approach.

Chapter 11

Conclusion and Future Work

This section begins by summarizing the design decisions taken to create protocols that would capture and verify physical world location and space data without depending on trusted third-party oracles, and the findings obtained by testing Location Mining protocol in a simulated environment. Although we provide interesting protocols to capture and verify physical world location and spaces, the work is not yet complete. In the final section of this chapter we provide a brief overview of the future improvements that could be made to further improve this work.

11.1 Conclusion

We are currently living in the Age of Digitization, characterized by constant innovation aimed at digitizing our day-to-day lives. However, during our initial work, we noticed a

lack of solutions that could reliably digitize physical locations and spaces without relying on oracles.

To address this issue, we developed an open and infrastructure-independent solution called Decentralized Physical Space Network (DPSN), which consists of two key components: *Location Mining* and *Space Mining*. Location Mining ensures a reliable association between a digital handle and its corresponding physical location, while Space Mining ensures a reliable association between a digital handle and the functionalities of its corresponding space.

In this document, we delve into the algorithms behind Location Mining and Space Mining. We evaluate Location Mining by simulating a network of wanderers in a virtual geospatial plane and manipulate various parameters to improve the protocol. We observe that the growth of the Proof of Space Time (PoST) score reflects the level of honest and active participation by the wanderers. Rewards play a crucial role in motivating wanderers to actively participate and contribute to the protocol, so we also evaluate how rewards are distributed among wanderers with different movement patterns. Upon closer examination, we discovered a bias towards a particular movement pattern that opportunistically programmed their movement around meeting keepers, potentially compromising fair reward distribution. To address this issue, we implemented a technique called "*Random look-back*", which significantly reduced the bias. We also successfully adjusted the difficulty of acquiring rewards, motivating wanderers in the physical world to work harder for their rewards.

Security is paramount in systems like these, especially when it comes to user privacy. Therefore, we provide a comprehensive analysis of DPSN's security, including how it handles major security issues such as privacy leaks. With the complete implementation of Location Mining and Space Mining, our framework opens up a world of opportunities and applications. These range from simple applications like renting spaces to more innovative and novel applications like a drone-enabled, opportunistic delivery system.

11.2 Future Work

Although we provide a detailed solution to the problems we set out to solve, it is far from perfect and needs several long arduous hours of development and adjustment, before we can make it available to the public. Here are the top candidates that would help in solidifying our framework:

- Since *Space Mining* is a rather complex protocol, it would prudent to encode the protocol and verifying its correctness and robustness with an automatic prover, before moving on to the implementation phase.

- Although *Space Mining* protocol has been fleshed out in great detail, and even supported by its own Security analysis, the time and complexity constraints made it infeasible to be implemented. The protocols might also need to be adjusted based on the feedback obtained from the simulations.

- On similar lines, *Space Mining*, would need a complete Result and Security analysis, to determine its effectiveness.

- Depending on the interaction between the finalized versions of *Location Mining* and *Space Mining* protocols, we might have new exploits or advantages. They would need to be carefully studied, leveraged and incorporated back into the protocols.

- The Security Analysis presented in this work, and enhanced by the future iterations, would need to be backed by concrete results, to demonstrate the effectiveness of the countermeasures and the protocols itself.

- Challenges issued during Location Mining need to be carefully constructed to safeguard the framework against multi-personality attacks. The effectiveness of the chosen approach also needs to be demonstrated through empirical analysis.

Bibliography

[1] S. Nakamoto, "Bitcoin: A peer-to-peer electronic cash system," *Decentralized business review*, pp. 1–9, 2008.

[2] M. Mettler, "Blockchain technology in healthcare: The revolution starts here," in *2016 IEEE 18th international conference on e-health networking, applications and services (Healthcom)*, pp. 1–3, IEEE, 2016.

[3] M. Shuaib, S. M. Daud, S. Alam, and W. Z. Khan, "Blockchain-based framework for secure and reliable land registry system," *TELKOMNIKA (Telecommunication Computing Electronics and Control)*, vol. 18, no. 5, pp. 2560–2571, 2020.

[4] A. I. Sanka and R. C. Cheung, "Blockchain: panacea for corrupt practices in developing countries," in *2019 2nd International Conference of the IEEE Nigeria Computer Chapter (NigeriaComputConf)*, pp. 1–7, IEEE, 2019.

[5] B. Consensus Encyclopedia, "Proof of location." `https://tokens-economy.gitbook.io/consensus/chain-based-proof-of-capacity-space/`

`dynamic-proof-of-location`, 2019.

[6] G. Caldarelli, "Understanding the blockchain oracle problem: A call for action," *Information*, vol. 11, no. 11, p. 509, 2020.

[7] J. Granados, "Reaching new transaction speeds on algorand." `https://developer.algorand.org/articles/reaching-new-transaction-speeds-on-algorand/`, Jun 2023.

[8] D. Jones, C. Snider, A. Nassehi, J. Yon, and B. Hicks, "Characterising the digital twin: A systematic literature review," *CIRP journal of manufacturing science and technology*, vol. 29, pp. 36–52, 2020.

[9] D. Gelernter, *Mirror worlds: Or the day software puts the universe in a shoebox... How it will happen and what it will mean.* Oxford University Press, 1993.

[10] G. O. Karame, E. Androulaki, M. Roeschlin, A. Gervais, and S. Čapkun, "Misbehavior in bitcoin: A study of double-spending and accountability," *ACM Transactions on Information and System Security (TISSEC)*, vol. 18, no. 1, pp. 1–32, 2015.

[11] F. Regner, N. Urbach, and A. Schweizer, "Nfts in practice–non-fungible tokens as core component of a blockchain-based event ticketing application," *Fortieth International Conference on Information Systems*, pp. 1–17, 2019.

[12] V. Thakur, M. Doja, Y. K. Dwivedi, T. Ahmad, and G. Khadanga, "Land records on blockchain for implementation of land titling in india," *International Journal of Information Management*, vol. 52, pp. 1–9, 2020.

[13] S. Barber, X. Boyen, E. Shi, and E. Uzun, "Bitter to better—how to make bitcoin a better currency," in *Financial Cryptography and Data Security: 16th International Conference, FC 2012, Kralendijk, Bonaire, Februray 27-March 2, 2012, Revised Selected Papers 16*, pp. 399–414, Springer, 2012.

[14] J. Benet and N. Greco, "Filecoin: A decentralized storage network," *Protoc. Labs*, vol. 1, pp. 1–36, 2018.

[15] J. R. Douceur, "The sybil attack," in *International workshop on peer-to-peer systems*, pp. 251–260, Springer, 2002.

[16] O. F. Cangir, O. Cankur, and A. Ozsoy, "A taxonomy for blockchain based distributed storage technologies," *Information processing & management*, vol. 58, no. 5, p. 102627, 2021.

[17] F. Aviation Administration, "Satellite navigation - gps - how it works." `https://www.faa.gov/about/office_org/headquarters_offices/ato/service_units/techops/navservices/gnss/gps/howitworks`, Jun 2022.

[18] A. Jafarnia-Jahromi, A. Broumandan, J. Nielsen, and G. Lachapelle, "Gps vulnerability to spoofing threats and a review of antispoofing techniques," *International Journal of Navigation and Observation*, vol. 2012, 2012.

[19] F. Space, "Foam whitepaper," *Foamspace Corp*, pp. 1–23, 2018.

[20] E. Org, "Single slot finality." `https://ethereum.org/fil/roadmap/single-slot-finality`, Aug 2023.

[21] A. Trouw, M. Levin, and S. Scheper, "The xy oracle network: The proof-of-origin based cryptographic location network," *XYO Netw., San Diego, CA, USA*, 2018.

[22] X. Xu, C. Pautasso, L. Zhu, Q. Lu, and I. Weber, "A pattern collection for blockchain-based applications," in *Proceedings of the 23rd European Conference on Pattern Languages of Programs*, pp. 1–20, 2018.

[23] A. Haleem, A. Allen, A. Thompson, M. Nijdam, and R. Garg, "A decentralized wireless network," *Helium Netw*, pp. 3–7, 2018.

[24] W. Wu, E. Liu, X. Gong, and R. Wang, "Blockchain based zero-knowledge proof of location in iot," in *ICC 2020-2020 IEEE International Conference on Communications (ICC)*, pp. 1–7, IEEE, 2020.

[25] M. Amoretti, G. Brambilla, F. Medioli, and F. Zanichelli, "Blockchain-based proof of location," in *2018 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 146–153, IEEE, 2018.

[26] F. Boeira, M. Asplund, and M. P. Barcellos, "Vouch: A secure proof-of-location scheme for vanets," in *Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, pp. 241–248, 2018.

[27] F. Boeira, M. Asplund, and M. Barcellos, "Decentralized proof of location in vehicular ad hoc networks," *Computer Communications*, vol. 147, pp. 98–110, 2019.

[28] P. Syverson, "A taxonomy of replay attacks [cryptographic protocols]," in *Proceedings The Computer Security Foundations Workshop VII*, pp. 187–191, IEEE, 1994.

[29] J. Schiller and A. Voisard, *Location-based services*. Elsevier, 2004.

[30] J. Chen and S. Micali, "Algorand," *arXiv preprint arXiv:1607.01341*, 2016.

[31] A. Hayes, "A cost of production model for bitcoin," *SSRN*, pp. 1–4, 2015.

[32] G. O. Karame, E. Androulaki, and S. Capkun, "Double-spending fast payments in bitcoin," in *Proceedings of the 2012 ACM conference on Computer and communications security*, pp. 906–917, 2012.

[33] M. Conoscenti, A. Vetro, and J. C. De Martin, "Blockchain for the internet of things: A systematic literature review," in *2016 IEEE/ACS 13th International Conference of Computer Systems and Applications (AICCSA)*, pp. 1–6, IEEE, 2016.

[34] Y. Kawase and S. Kasahara, "Transaction-confirmation time for bitcoin: A queueing analytical approach to blockchain mechanism," in *Queueing Theory and Network Applications: 12th International Conference, QTNA 2017, Qinhuangdao, China, August 21-23, 2017, Proceedings 12*, pp. 75–88, Springer, 2017.

[35] S. Sayeed and H. Marco-Gisbert, "Assessing blockchain consensus and security mechanisms against the 51% attack," *Applied sciences*, vol. 9, no. 9, p. 1788, 2019.