Enabling Distributed Intelligence in the Internet of Things with IOTA Tangle and Mobile Agents

Tariq Alsboui

School of Computing and Engineering

University of Huddersfield

A thesis submitted to the University of Huddersfield in partial fulfilment of the requirements for the degree of Doctor of Philosophy

Supervisors: Dr. Yongrui (Louie) Qin

and Professor. Richard Hill

January 2022

To my mother and father,

my wife and my little princess,

my brother,

who made all of this possible,

for their endless encouragement and patience.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.

Tariq Alsboui

January 2022

Acknowledgements

I would like to express the deepest thanks and appreciation to **Dr. Yongrui (Louie) Qin**, and **Professor. Richard Hill**. Their enthusiasm, insight and guidance were invaluable assets that encouraged me to proceed well beyond my initial ideas. This thesis would not have been completed without their guidance, persistent help, and continuous support.

I would like to thank **Dr. James Dyer** and **Dr. Hussain Al-Aqrabi** for the insightful comments, support, and help in structuring the thesis for better presentation.

I also would like to thank my parents for the support, and encouragement that they have provided me throughout my study, which was a key factor in the achievement I have made. I also would like to thank my brothers, sisters, and friends for their continuous support along the way.

Last but not least I would like to thank my wife Ghayda for her encouragement and support in which she offered a kind assistance, during the final stages of my PhD.

Copyright Statement

List of Publications

1. Tariq Alsboui, Yongrui Qin, Richard Hill, Hussain Al-Aqrabi. (2021). *Distributed Intelligence in the Internet of Things: Challenges and Opportunities*. SN Comput. Sci, 4:10.1007/s42979-021-00677-7, 2021. https://doi.org/10.1007/s42979-021-00677-7.

   My contribution: I proposed the idea of categorising the distributed intelligence approaches, collected and summarised all the articles described in the survey paper, analyzed and compared all the summarised articles according to the identified IoT challenges, and drafted the manuscript with the support of the supervision team. All authors discussed the findings and commented on the final manuscript.

2. Tariq Alsboui, Yongrui Qin, Richard Hill, Hussain Al-Aqrabi. (2020). *Enabling distributed intelligence for the internet of things with IOTA and mobile agents*. Computing, 102(6):1345–1363, 2020. https://doi.org/10.1007/s00607-020-00806-9.

   My contribution: I proposed the idea of the framework for enabling distributed intelligence in the IoT domain, summarised all related work, developed the distributed algorithms, created the design of the algorithms, implemented the framework and conducted some experiments to verify the effectiveness of the proposed framework and analyzed the results, and drafted the manuscript with the support of the supervision team. All authors discussed the algorithms, results, findings, and commented on the final manuscript.

3. Tariq Alsboui, Yongrui Qin, Richard Hill. (2019). *Enabling Distributed Intelligence in the Internet of Things Using the IOTA Tangle Architecture*. In M. Ramachandran, R. Walters, G. Wills, V. Méndez Muñoz, & V. Chang (Eds.), Proceedings of the 4th International Conference on Internet of Things, Big Data and Security: IoTBDS 2019 (Vol. 1, pp. 392-398). SciTePress. https://doi.org/10.5220/0007751403920398.

   My contribution: I proposed the idea of integrating the IOTA Tangle and mobile agents to enable distributed intelligence in the IoT domain, drafted the initial design of the framework, and drafted the manuscript with the support of the supervision team. All authors discussed the initial idea and commented on the final manuscript.

4. Tariq Alsboui, Yongrui Qin, Richard Hill, Hussain Al-Aqrabi. (2020). *Towards a scalable iota tangle-based distributed intelligence approach for the internet of things*. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, Intelligent Computing-Proceedings of the 2020 Computing Conference, Volume 2, AI 2020, London, UK, 16-17 July 2020, volume 1229 of Advances in Intelligent Systems and Computing, pages 487–501. Springer, 2020.

   My contribution: I proposed the idea of the scalable iota tangle-based distributed intelligence approach for the IoT domain, drafted the design of the approach, implemented the approach and conducted experiments to verify the effectiveness of the proposed approach and analyzed all the results, and drafted the manuscript with the support of the supervision team. All authors discussed the results, findings, and commented on the final manuscript.

5. Tariq Alsboui, Yongrui Qin, Richard Hill, Hussain Al-Aqrabi. (2021). *An energy efficient multi-mobile agent itinerary planning approach in wireless sensor networks*. Computing, 103(9):2093–2113, 2021. https://doi.org/10.1007/s00607-021-00978-y.

My contribution: I proposed the idea of the multi-mobile agent itinerary planning approach in WSNs, implemented it and analyzed the results to verify how effective the approach, and drafted the manuscript with the support of the supervision team. All authors discussed the algorithms, results, and commented on the final manuscript.

Abstract

Widespread adoption of smart Internet of Things (IoT) devices is accelerating research for new techniques to make IoT applications scalable, energy-efficient, and capable of working in mission-critical use cases. IoT devices are characterised by limited resources, such as power consumption and memory storage, which is a fundamental challenge of IoT applications. Distributed Intelligence (DI) is an area of research within the field of IoT and is seen as a practical route towards the decentralisation of IoT architectures. Enabling DI is a challenging task in IoT because it needs to ensure scalability, and energy-efficiency due to resource constraints. These challenges requires a new solutions to be investigated. There is a wide body of literature about DI in the IoT. These approaches are dealing with a particular challenge and identified as an effective and efficient in achieving that challenge. However, there are few attempts to enable DI in IoT. The aim of this thesis is to develop a scalable, and energy efficient solution for enabling DI in the IoT. This aim is achieved through the development of high-level and low-level intelligence techniques to support DI. This thesis contributes towards the design of a new framework that ensures scalability and energy-efficiency of IoT applications. The developed hybrid Mobile-Agent Distributed Intelligence Tangle-Based framework (MADIT) represents the novel contribution of the work. The aim of which is to offer low-level and high-level intelligence for IoT applications. The low-level intelligence along with IOTA Tangle-based intelligence form the distributed intelligence in the IoT domain. The low-level intelligence is achieved through the use of multi-mobile agents to collect transactions data and high-level intelligence is achieved through the use of

Tangle-based architecture. The framework evaluates a Proof-of-Work computation offloading mechanism that performs costly computations on behalf of constrained IoT devices for efficacy with regard to energy efficiency and transaction throughput. The Proof-of-Work offloading computation mechanism improves efficiency and speed of processing, while saving energy consumption. In addition, this thesis proposes a new energy efficient Graph-based Static Mutli-Mobile Agent Itinerary Planning approach (GSMIP). The GSMIP applies Directed Acyclic Graph (DAG) related techniques and divides sensor nodes into different groups based on the routes defined by mobile agents itineraries. Mobile agents follow the predefined routes and only collect data from the groups they are responsible for. The proposed solution can be easily generalised to different application domains, and is less complex than many other existing approaches. The simplicity of the solutions neither demands great computational efforts nor large amounts of energy consumption. The experimental findings demonstrate the effectiveness and superiority of the proposed approach compared to the existing approaches in terms of energy consumption and task delay (time).

Table of contents

List of figures

List of tables

Chapter 1

Introduction

1.1 Research Background

The Internet of Things or the IoT is an emerging world-wide network of interconnected physical-heterogeneous smart objects (e.g., wearable-sensors, environmental sensors, and connected devices) that are uniquely addressable and are available through networking technologies such as WiFi, Bluetooth, and others [1]. By 2030, the study predicts that the IoT will rise exponentially, for example, by about 125 billion connected devices to the Internet [2–4]. As a result, this poses several challenges in terms of providing timely delivery, data volume, speed, confidentiality, and scalability [5, 6].

There are several features available for IoT applications: First, *sensing* the environment; Second, *communication* between objects for efficient data transfer; and Third, *computation* typically carried out to produce necessary raw data information.

The advent of the IoT enables a new paradigm that binds the physical objects on the Internet to form pervasive networks that allow sensing and medicating environments to respond to dynamic stimuli [1], often known as Cyber-Physical Systems (CPS) [7]. The IoT was also demonstrated by the Auto-ID centre that immediately recognises physical objects in the supply chain via Radio-Frequency Identification RFID technology and Electronic Product

Code (EPC). Such systems have already demonstrated the potential to enhance the quality of life by turning cities into *smart* cities [8], homes into *smart* homes [9] and campuses into *smart* campuses [10].

Despite the fast adoption of IoT in industry, scalability, resilience, energy efficiency, and security are the main challenges of adopting IoT [11]. In fact, the researchers reported in [12] that security is one of the top ten IoT challenges.

Enhancing CPS capabilities to schedule, analyse, and solve goal-directed issues allows complex IoT systems to be managed and ultimately optimised [13]. Such systems often require significant computational power.

Distributed Ledger Technology (DLT) is an emerging technology that allows the ledger to be shared, replicated, and synchronized in a distributed manner among participant nodes [14]. It allows a huge number of nodes in the distributed ledger network to be able to agree on adding new transactions and recording transactions without requiring a central entity. One example of DLT is the IOTA technology which relies on a new architecture called the tangle that indicates high scalability, no transaction fees developed for the IoT.

Wireless Sensor Networks (WSN) are a group of spatially distributed sensors deployed over an area of interest to monitor environmental conditions such as temperature, humidity and sound [15]. Sensor nodes are mainly concerned with sensing and transmitting data to central locations for further processing. WSN is considered as one of the most important enabling technologies of the IoT [1]. Sensor nodes are characterised by limited resources such as power consumption, processing capabilities, and memory storage. The usage of WSNs has been approved in several application areas such as military applications, healthcare, and monitoring system.

Mobile Agent (MA) technology has been put forward to cope with the resource constraints of WSNs such as energy consumption since it allows mobile agents to be dispatched to collect data from sensor nodes rather than sensor nodes transmit their data, which consumes

energy of sensor nodes. However, the way in which mobile agents are routed among sensor nodes must be intelligently planned to reduce energy consumption and improve information accuracy. The main issue of WSN is how to create an itinerary plan for MAs to collect data [16]. Itinerary planning can be specified as the route of the MAs when visiting sensor nodes. It displays the sequence of the source nodes to be visited through the MAs migrations trip.

1.2 Research Objectives

This research project aims to develop a scalable and energy-efficient distributed intelligence framework for the IoT. The framework adopts the IOTA tangle architecture and multi-mobile agents in order to enable distributed intelligence whilst minimising energy- consumption and ensuring scalability. Also, the framework develops a new multi-mobile agent itinerary planning approach that is scalable and energy-efficient.

The objectives are summarised as follows:

1. To examine and identify common requirements of existing distributed intelligence approaches in IoT.

2. To develop a distributed intelligence framework using multi-mobile agents and IOTA technology as an efficient architectural technique to facilitate local interaction, collection, aggregation of transactions data and it enables the deployment of IoT applications that are scalable and energy efficient.

3. To evaluate an existing Proof of Work (PoW) offloading mechanism for efficacy with regard to energy efficiency and transaction throughput.

4. To develop a new energy-efficient multi-mobile agent itinerary planning mechanism by partitioning Directed Acyclic Graph (DAG) into groups and allowing mobile agents to visit a particular group.

1.3 Research Motivation and Challenges

The problem of enabling distributed intelligence imposes significant challenges. This is due to the fact that it is distributed in nature, involves both processing and communication, energy-efficiency and placing distributed functionality through multiple layers in the IoT architecture. Another challenging problem related to distributed intelligence with current IoT networks is that of scalability and this is because the number of devices connected through an IoT network is expected to reach billions of devices. Therefore, current centralised systems to deal with connecting different nodes in the network will turn into a bottleneck. This would require a large investments into servers that will handle the large amount of information and the entire network can turn down if the server becomes unavailable.

Several techniques have been developed over the past few years to support distributed intelligence; however, some techniques rely on cloud computing architecture, which would result in a system failure due to a central point of control. In addition, a cloud based distributed intelligence architecture poses additional challenges. The application is deployed in an untrusted environment and data can be tampered with. The nature of deployment in the cloud is inefficient due to bandwidth usage and not suitable for time critical applications that requires fast response e.g., target tracking. An ideal solution should shift from the central point of control and be decentralised.

To address the above-mentioned problems, developing a distributed intelligence framework that is suitable for various IoT applications is the main motivation of this research. The developed framework aims at enabling distributed intelligence at both high and low levels of the architecture. At the high level, a tangle-based architecture is adopted to handle

transactions in an efficient way, while the low level employs multi-mobile agents to cater for node level communications. The framework integrates several features into a single model including; scalability, eliminating redundant data, and facilitating cooperation among various IoT devices. Integrating these features into a distributed intelligence architecture would ensure scalability, energy-efficiency and provide a promising way of overcoming the above-mentioned obstacles.

Although, reasonable results have been obtained from the mobile agent itinerary planning techniques in WSNs, however, it is important to note that the problem of itinerary planning for mobile agent is the most challenging issue in multi-mobile agent, mainly due to the fact that mobile agent efficiency depends on the itinerary planning, determining the optimal number of mobile agents and partitioning the sensor network into groups. It has been proved that planning itineraries for mobile agent is an NP-hard problem [17–21]. Therefore, in this study, a new multi-mobile agent itinerary planning approach is proposed to address the shortcomings of the existing multi-mobile agent itinerary planning approaches and enhance the overall performance of the network by extending the network lifetime.

1.4 Contributions

Aiming at enabling distributed intelligence in the IoT with less resource usage, the research is focused on the development of a scalable framework that supports distributed intelligence at two levels including: high-level and low-level. In addition, the research proposes a new multi-mobile agent itinerary planning approach that is scalable and energy-efficient. The main contributions of the thesis are summarised as follows:

1. **A new distributed intelligence framework called the Mobile-Agent Distributed Intelligence Tangle-Based Approach (MADIT)** has been developed that can effectively overcome the issues of scalability, high energy consumption and redundant data. The

framework uses a tangle based architecture to handle transactions data and employs a multi-mobile agent to cater for node level communications to collect transactions data.

2. **A new source grouping technique** has been developed based on a Directed Acyclic Graph (DAG). It applies DAG related techniques and partitions the network into several groups efficiently based on the routes defined by mobile agents itineraries. Mobile agents follow the predefined routes and only collect data from the groups they are responsible for.

3. **Multi-mobile agents itinerary planning technique** has been developed for scheduling mobile agents to collect data from sensor nodes in an intelligent way. The proposed itinerary planning mechanism has shown outstanding performance and outperforms the most widely used mobile agent itinerary planning mechanisms.

4. **As a result of the above contributions**, enabling distributed intelligence by adopting the IOTA tangle and mobile agent as an efficient technique becomes possible. This claim was thoroughly evaluated and supported by an experimental study that used real-life parameters. Experimental results show that the work outperforms its best rival in the literature.

1.5 Research Methodology

The methodology that is used in this thesis is a pure computer science research method [22]. The original contributions are developed through a new framework, theory and distributed algorithms. The way in which the approach was developed is split into four working packages. The first package addresses the research background and the requirements of the project. Two are technical working packages. The final working package is concerned with writing up the thesis.

- Work Package 1: Background of the Research.

  Initially, the research started by reviewing the state of the art distributed intelligence approaches in IoT and the problem domain to be understood. The literature review described the strengths and limitations of each distributed intelligence approach and used the following database digital libraries: ACM Digital library, IEEE Xplore, ScienceDirect and Springer-Link.

- Work Package 2: Distributed Intelligence Framework.

  This particular working package is mainly concerned with the design of the framework. It clearly describes the main components of the proposed framework and how these components will interact with each other to accomplish the objectives of the research. This working package consists of four phases as follows:

  1. Defining the role of each component in the proposed framework.
  2. Multi-Mobile agent for collecting transaction data at low-level.
  3. Tangle based architecture to deal with transactions data.
  4. A Proof of Work computation offloading mechanism for efficacy with regard to energy efficiency and transaction throughput.

- Work Package 3: Multi-Mobile Agent Itinerary Planning Approach.

  This working package is concerned with the development of a new energy-efficient mobile agent itinerary planning approach. This work package is composed of two phases as follows:

  1. A new approach to multi-mobile-agent itinerary planning that manages resources in terms of energy efficiency and scalability.

2. It uses a Directed Acyclic Graph (DAG) related techniques to generate mobile agent itinerary planning by dividing the DAG into groups and allocating mobile agents to each group.

- Work Package 4: Thesis Writing Up.

  This working package is concerned with writing up the thesis that is based on the results of all of the above working packages.

1.6 Thesis Outline

In addition to the introduction chapter, this thesis consists of five other chapters, each described as follows:

1. **Chapter 2: Literature Review**

   This chapter reviews existing distributed intelligence approaches in the IoT focusing on their advantages and disadvantages. It also defines the inspiration and obstacles that underlie the use of distributed intelligence in the IoT. Furthermore, it presents a summary of several representative distributed intelligence research deployments according to the following categorisation: Cloud-Computing, Mist-Computing, Distributed Ledger Technology (DLT), Service Oriented Computing and Hybrid, followed by an evaluation of distributed intelligence approaches. In addition, it presents the state-of-the-art mobile agent itinerary planning approaches. Finally, possible future research directions are presented in this chapter followed by a summary.

2. **Chapter 3: IOTA Distributed Ledger Technology**

   This chapter presents the background of distributed ledger technology. It presents the theories and components of IOTA technology and describes the features and working principles of the IOTA tangle. This chapter also presents the suitability of the IOTA

tangle for the IoT. In addition, it provides three different application scenarios showing the benefits of IOTA for the IoT. Finally, it presents the lessons learned followed by a summary of the chapter.

3. **Chapter 4: A Distributed Intelligence Framework for the IoT with IOTA and Mobile Agents**

   This chapter presents a new distributed intelligence framework for the IoT, which integrates the IOTA tangle and mobile agents. Then, it presents the use of mobile agents to assist in enabling distributed intelligence. This chapter also describes the proposed framework along with the developed distributed algorithms, followed by an implementation of the framework. In addition, it evaluates the framework and compares it with the baseline method, followed by a summary of the chapter.

4. **Chapter 5: An Energy Efficient Multi-Mobile Agent Itinerary Planning Approach**

   This chapter presents a new multi-mobile agent itinerary planning approach that is scalable and energy efficient. It also describes the anatomy of a mobile agent. Then, it presents the proposed multi-mobile agent itinerary planning approach. In addition, it provides the benefits of the proposed approach and a comparison against alternative approaches followed by a summary of the chapter.

5. **Chapter 6: Conclusion and Future Work**

   This chapter presents a summary of all of the contributions presented in this thesis. It provides conclusions on the overall research and outlines potential future research directions to extend and enhance the work further.

Chapter 2

Literature Review

Widespread adoption of smart IoT devices is accelerating research for new techniques to make IoT applications secure, scalable, energy-efficient, and capable of working in mission-critical use cases that require the ability to function offline. In this context, the novel combination of Distributed Ledger Technology (DLT) and Distributed Intelligence (DI) is seen as a practical route towards the decentralisation of IoT architectures. This chapter surveys DI techniques in IoT and commences by briefly explaining the need for DI, and proposing a comprehensive taxonomy of DI in IoT. This taxonomy is then used to review existing techniques and to investigate current challenges that require careful attention and consideration. Based on the taxonomy, IoT DI techniques can be classified into five categories based on the factors that support distributed functionality and data acquisition: Cloud-Computing, Mist-Computing, Distributed-Ledger-Technology, Service-Oriented-Computing and Hybrid. Existing techniques are compared and categorised mainly based on related challenges, and the level of intelligence supported. This chapter evaluates more than thirty current research efforts in this area. It defines many significant functionalities that should be supported by DI frameworks and solutions. The work assists system architects and developers to select the correct low-level communication techniques in an integrated IoT-to-DLT-to-cloud system architecture. The benefits and shortcomings of different DI approaches are

presented, which will inspire future work into automatic hybridisation and adaptation of DI mechanisms. Finally, open research issues for distributed intelligence in IoT are discussed. The research presented in this chapter was published in [23, 24].

2.1 Introduction

Given the potential benefits of Distributed Intelligence (DI), a number of issues related to general DI approaches, such as distrust, lack of scalability, energy-efficiency and poor identification of potential participants, where the privacy of the participants still needs to be solved [25]. Traditional approaches to DI, e.g., those using Cloud Computing, are inadequate for dynamic IoT environments. The vague clauses in Cloud Computing service agreements and unclear technical specifications may result in consumers of cloud services to be unable to discover trustworthy cloud services. Consequently, DI in conjunction with DLT, has been proposed to address these kinds of issues.

DI technology support for IoT applications can be categorised into five broad categories: Cloud-Computing, Mist-Computing, Distributed- Ledger-Technology, Service-Oriented-Computing and Hybrid. The major differences among these categories are described as follows. In cloud computing, DI processing functionality and data is controlled by single entities and only sent to the cloud for further processing. With mist computing, part of DI processing functionality and data is processed at the extreme edge of a network that typically consists of micro-controllers and sensors. By working at the extreme edge, mist computing can harvest resources such as computation and communication capabilities available on sensor nodes. With distributed ledger technology, the processing functionality and data is distributed among all participant nodes. With service oriented computing, processing components are provided as services and distributed at all levels of the system. Eventually, the hybrid method is a combination of two or more of the four categories listed above.

2.2 Distributed Intelligence in the IoT

This section outlines Distributed Intelligence (DI) and illustrates the need for DI in the IoT by identifying a number of the critical factors that determine the challenges of the IoT. Furthermore, DI approaches are categorised into five broad categories, and cast these categories in a detailed taxonomy. Fig. 2.1 graphically illustrates the dimensions that have been identified. Note that, DI is one of the most critical efforts to use the ever-increasing amount of data brought back by IoT nodes deployed with sensors to achieve a detailed and expensive task of finding, analysing and identifying the information needed.

According to [13] distributed intelligence is defined as "Cooperation between devices, intermediate communication infrastructures (local networks, access networks, global networks) and/or cloud systems in order to optimally support IoT communication and IoT applications". This chapter expands the concept of distributed intelligence in a border perspective, which includes data management, device management, resource constraints management, optimising communication and computation and scalability. Therefore, distributed intelligence can be defined as a set of techniques that allows processing functionality to be distributed, enables collaboration between smart objects and mediates data exchange to optimise communication for IoT applications. This is a description of consciousness in which the term distributed intelligence is used throughout this chapter.

It is required that intelligence should be optimally distributed and activated in order to be able to invoke functions such as processing, security, quality of service (QoS) and to build the communication infrastructure. Distributed intelligence involves offloading some computation and data processing to devices in which less data is transmitted to central collection points. Thereby, reducing bandwidth requirements and increases throughput. Through distributed intelligence, the right communication and processing functionality will be available at the right place and at the right time [13]. There are a number of factors that are required in order

to enable distributed intelligence such as resource constraints, scalability, security, privacy, offline capability, and interoperability. These factors are discussed in detail in Section 2.3

Distributed intelligence has the potential to overcome many of the IoT technical challenges, such as scalability, resource constraints, security, privacy and offline capability [26]. DI involves the distribution of processing functionality, cooperation among IoT devices and is concerned with identifying where functionality should be invoked.

Additional intelligence is required to optimally service a range of IoT applications and user requirements. This intelligence applies not only to data processing, but also to security, privacy, network configuration, quality of service and many more. There is therefore no single reliable place where this intelligence is triggered or placed. It can spread from the same devices to the Cloud/DLT/Fog according to the situation. It is expected that intelligence will be distributed through various locations to achieve maximum performance or functionality. In addition, this involves both in-network processing and networking elements.

The organisation of nodes plays a crucial role in DI as it defines, along with other factors, including (1) cost, which is the amount of energy needed to collect raw data, (2) accuracy, which is the level of coverage, and (3) reliability, which includes timeliness. The organisation of nodes may be either centralised or hierarchical, data gathered by all nodes is forwarded to a gateway (e.g., Raspberry Pi, Arduino) utilising single-hop or multi-hop communication in the centralised approach [27]. Nonetheless, there is lack of scalability in this approach, which is a critical concern for IoT applications. It is unreliable, and creates traffic bottlenecks and delays in transmission or congestion, particularly in areas across gateway nodes [8, 27]. In order to overcome the problems of centralised approaches, IOTA tangle has been introduced as a promising solution to achieve a longer network lifetime and to provide better scalability [25].

A taxonomy of distributed intelligence approaches in IoT is described in Fig. 2.1, which depicts the whole taxonomy that describes the IoT challenges. Then, it presents DI intelli-

gence levels as low-level and high-level focusing on processing functionality and data. Finally, distributed intelligence approaches are classified into five broad categories. The following paragraphs provide a full description about the dimensions that have been identified.

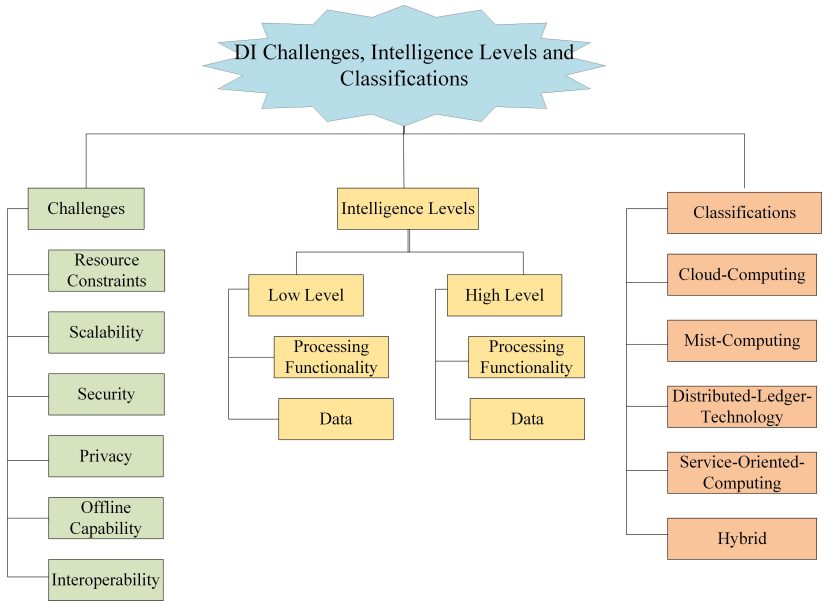

Fig. 2.1 A taxonomy of DI challenges, intelligence levels and Classifications in IoT

– Challenges: Connected IoT devices in the coming future lead to a number of fundamental challenges, e.g., resource constraints, poor scalability, security, offline capability, privacy and interoperability [28–30] as well as the massive amount of data produced by IoT. These also create large demands upon network resources [30].

– Intelligence Levels: is classified into two parts, including low-level and high-level. The former refers to node level communications in which processing functionality is distributed among nodes in the network and data processing occurs within the network, i.e., in-network processing. The latter uses high level nodes in the architecture, i.e., nodes with rich resources to process and handle data.

– Classifications: DI technology support for IoT applications is classified into five broad categories: Cloud-Computing, Mist-Computing, Distributed-Ledger-Technology (DLT), Service-Oriented-Computing and Hybrid technology.

2.3 The Need for Distributed Intelligence in the IoT

2.3.1 Resource Constraints

Resource Constraints are referred to IoT devices that are specifically designed with limited power, limited storage capabilities, and limited processing. These limited resources makes DI a challenging distributed task. IoT systems generate large quantities of data, generating a high demand for network resources [30]. IoT devices tend to be small and equipped with batteries to maintain the balance between the effective span of their lifetime and the potential costs of device replacement.

As a result, these devices are typically subject to strict constraints on their power consumption and available hardware resources. Efficient use of IoT devices energy would maintain a prolonged network lifetime. Energy harvesting [31], computation offloading mechanism [32] and management of the wake-up-sleep cycle [33] are important techniques that are effective in saving energy-consumption of constrained IoT devices.

2.3.2 Scalability

Scalability can be defined as the ability of the network to meet the increasing demands of the network. It is a fundamental requirement of any IoT system to handle the capability of the growing amount of work. It can be categorised as Vertical Scaling and Horizontal Scaling. The former is intended to upgrade the existing network devices by including more (e.g., power, RAM, CPU) [34]. The latter, is concerned with expanding the network by introducing more nodes.

In line with the predictions made in [3, 4] IoT is continually changing and growing to meet ever-increasing demands. Therefore, future technologies should be very flexible in dealing with billions of things or smart objects that are inevitably connected to the Internet.

2.3.3 Security

Security refers to the act of securing IoT devices and the networks they are connected to. The aim is to protect the entire system, which represents an IoT installation [35]. Authentication is concerned with identifying users and devices in a network and granting access to authorised people, whereas confidentiality ensures that data is protected by preventing the unauthorised disclosure of information. Availability guarantees that an IoT system and data can be accessed by authenticated users whenever needed [36].

Security remains one of the most fundamental challenges [37–41]. This is believed to be the most challenging and crucial obstacle to IoT. In addition, device security is another fundamental challenge that determines the successful implementation of IoT applications [42]. In such circumstances, authentication is especially of great concern, given the harm that could occur from a possibly malicious processing device and unauthenticated device attacks (Referred to outside device attacks)in an IoT system [43]. Ensuring the robustness of any IoT system against hacking is critical.

2.3.4 Privacy

Privacy can be defined as the ability of the system to properly ensure that any data/information is protected and remains confidential. IoT devices must have capability to send their data over the network. Hence, some IoT devices may capture private and disclose sensitive information so that they may pose a risk for the system [44]. According to [45] private information can be further categorised as follows: 1) personal information: Such as National Security Number.

2) Sensitive information: Such as salary. It should be ensured that these two types of data are private so that individual's information cannot be revealed without appropriate permission.

One main application that requires careful design of privacy is the healthcare, where patient information is delicate, and user privacy is concerned. In addition, the privacy leakage of user data is usually the ultimate concern, in particular with regard to sensitive data (e.g., the location and movement trajectory information) [46]. A possible solution would be to define who can access that data and in what form the data should be.

2.3.5 Offline Capability

It is also referred to as resilience. It can be defined as the system's ability to operate effectively in mission-critical scenarios. For example, all capabilities do not change if the Internet is not available. Consequently, in case if the system cannot connect to the internet, offline capability remains extremely important. IoT applications that place the intelligence in a cloud based system will ultimately become unavailable upon the shutdown of the Internet connectivity.

Creating a distributed intelligence approach that handles and processes data in the cloud is inefficient if the cloud becomes unavailable, the system should have the ability to function offline in this critical situation. Therefore, the main functionality of any IoT system should be placed within the network. This results in using simple local processing, which is still possible to have an operational system with less functionality. Hence, the distribution of intelligence is desirable and should be supported.

2.3.6 Interoperability

Interoperability can be defined as the ability of software to communicate with one another for effective exchange and process of information [47]. It should be tackled through multiple layers of services to enable software and devices to interact seamlessly with each other. This ensures straightforward integration.

Interoperability is the outcome of a range of critical problems, including vendors lock-in, the difficulty of developing IoT applications that work directly in cross-domains and or cross-platforms and also the challenges of IoT communication for non-interoperable IoT devices. Also, several manufacturers provide a wide selection of technology in its devices, and these devices on the market are unlikely to be directly compatible.

2.4 State-of-the-Art of Distributed Intelligence in IoT

In this part, a review of the recent approaches on distributed intelligence is presented. These approaches are summarised and compared using the challenges presented in Section 2.2. As aforementioned, there are five categories of distributed intelligence approaches.

2.4.1 Cloud Computing DI

In cloud computing approaches to DI, generally two layers are considered: the cloud and the end devices. Data is processed in the cloud (high-level intelligence) and devices equipped with sensors, are responsible for sensing the environment (low-level intelligence). In the simplest form, data is stored, processed and transferred to the cloud for further processing rather than connected devices [48]. The cloud is driven by a centralised design architecture and the functionality is managed in the cloud.

This is inefficient for the application that requires real time-decisions. For example, for autonomous vehicles, real-time decisions are critical. To overcome some of the inherent problems, [49] have adopted Amazon Web Services (AWS) and introduced a framework for smart traffic control. The framework is based on a public cloud AWS IoT. The components of the system include: AWS IoT, lambda, dynamoDB, kinesis, and cloud watch [49]. To be specific, AWS IoT is responsible for collecting data from the environment. The dynamoDB is responsible for collecting and storing data. This ensures what is beyond the endpoints is

updated in a timely manner. The cloud watch is a platform for monitoring AWS services and is responsible for debugging AWS services in run-time. The proposed framework is energy-efficient with the use of MQ Telemetry Transport (MQTT) protocol and is scalable and secure. However, privacy and offline capability are not supported.

Similarly, a distributed intelligence approach that leverages the AWS IoT platform is proposed in [50] for connected vehicles. The approach authenticates data according to five business rules. It deploys six unique Amazon services that store various details about cars health, trip and owners. The approach is energy efficient and achieves privacy. However, it lacks support for offline capability.

A distributed intelligence architecture that consists of three components is proposed by the CARMA project [51]. Each components of the architecture is responsible for a specific task. In carma vehicle, various sensors are connected together to collect data. The carma edge is responsible for processing of the data via one or two machines. Finally, the CARMA core is a cloud-based backend system that is based on public cloud resources and supports services and information storage. The proposed framework is not suited for time-critical applications, and lacks scalability.

In [52], the author proposes a cloud-based solution with the benefits of fog computing. The key concept is to introduce two more security features on fog gateway devices, such as *Decoys* and user behavior *Profiling*. The Decoys features involve putting the legitimate user in highly prominent locations in order to identify the dubious entry. Therefore, two of the new features are being added on top of the existing features of cloud security. The proposed approach achieves better security. However, other IoT related challenges such as energy consumption and offline-capability are not well supported.

A distributed intelligence solution, called PROTeCt-privacy architecture for IoT has recently been proposed in [53]. The architecture improves user privacy by implementing privacy enforcement at the IoT devices instead of at the gateway. Consequently, the proposed

approach improves both system security and fault tolerance, since it removes the single point of failure (gateway). It also decreases the amount of data that must be encrypted in order to secure the data transmitted by IoT devices. Therefore, the amount of data that must be transmitted also decreases. The architecture is energy efficient because less amount of data is transmitted to a gateway node. However, it lacks scalability and offline capability.

Similarly, the authors of [54] proposed Cloud-IoT distributed intelligence architecture, using an efficient and secure data acquisition scheme. The data collected through the terminals is divided into blocks, sequentially encrypted and processed with the corresponding access subtree until it is forwarded to the cloud in parallel. The proposed approach reduces the time, cost and security. However, the approach does not take into consideration the power usage of IoT devices, lacks offline capability, and scalability.

2.4.2 Mist Computing DI

In Mist Computing approaches to DI, part of DI functionality and data is processed at the extreme edge of a network that consists of sensors and micro-controllers. By working at the extreme edge, mist computing can harvest resources with the help of computation and communication capabilities available on the sensors [55]. In its simplest form, the gateway applies functionalities and rules for monitoring the health of local nodes, execution of computationally extensive tasks and filtering application parameters [56].

The authors in [57] recently proposed a heterogeneous five-layer mist, fog, and cloud-based architecture that is responsible for managing and routing (near-real-time) effectively. Data processing from offline / batch mode is also supported. In this framework, mist computing is responsible for checking if the data needs to be processed or not by applying certain-basic rules and the offloading mechanisms when needed. Software-defined networking (SDN) and link adaption-based load balancing are used in the heterogeneous framework. The framework ensures efficient resource utilisation while achieving optimal resource al-

locations. The proposed framework is energy efficient, can eliminate redundant data and provides a fast-response to certain events. However, the offline capability mechanism is not well supported. Furthermore, the framework does not include how scalability is achieved in the perception layer.

A framework based on mist computing is proposed by the authors of [58]. The framework consists of four layers: the layer of data abstraction, the layer of data extraction, the transmission layer and the layer of aggregation/integration, where each layer is dedicated to performing a specific task. The data extraction layer is responsible for extracting data from IoT devices. In the data abstraction layer, data is encapsulated into a JavaScript Object Notation (JSON) format rather than transmitting raw data via it. The data transmission layer is responsible for transmitting and receiving information through any radio and has a mist nodes where the abstracted payload of JSON-SenML [58] is transmitted to the microcontroller via the radio attached. The proposed framework efficiently achieves interoperability and is energy efficient. However, it lacks scalability and offline capability. Furthermore, privacy and security are not supported in their design.

Another work is proposed by the authors in [59], a framework that consists of four layers, including: IoT physical devices layer, mist nodes layer, fog nodes layer and cloud nodes layer. IoT layer sends data to the mist nodes, which are responsible for processing the data. The cloud node layer is responsible for heavy computation tasks. The proposed framework is energy efficient since it uses mist node and has less latency. However, it lacks scalability at the physical layer, and does not support security, privacy and offline capability.

The authors [60] have taken that work a step further by introducing an offloading computation mechanism among mist nodes using the MQ Telemetry Transport (MQTT) protocol that does not require service orchestration. Similar to the above work, a generic framework based on mist computing is proposed in [61]. The framework consists of mist nodes that

process data at the extreme-edge and provide mobile devices to share the networking and computational mechanisms as services in a versatile manner.

The framework is called a mobile embedded platform (mePaaS) and the essence of it lies in the architecture of Enterprise Service Bus (ESB). mePaaS nodes lend their hardware resources on the basis of a service level agreement (SLA) to others. Also, it utilises a plugin module-based method to enable nodes to perform computational processes specified through their requesters. mePaaS is capable of implementing a workflow that makes the service modules available to complete the requesting tasks. mePaaS requests may submit a request package consisting of the process flow specified in the standard workflow model (e.g., BPMN) and input parameters with custom algorithms. The proposed framework is energy efficient because of the use of mist nodes that processes data at the extreme edge. However, other related IoT challenges, such as offline capability, privacy, security and scalability, are not well considered in their design.

In [26], the authors described the fog computing architecture in support of distributed intelligence in the IoT. Fog nodes are considered as hardware and software architecture. In hardware, fog nodes are mainly installed on gateways, and appliances. In software, fog nodes are described as a virtual machine. Reliability, bandwidth and security are enhanced. However, it has been identified that security and privacy in fog computing remains as an issue [62–64]. In addition, how the approach is implemented and evaluated is not described. Also, it lacks a mechanism that deals with interoperability.

The work in [65] uses device-driven and human driven, to decrease energy usage and latency. Machine learning is adopted to identify user behaviors and data gathering from sensors are adjusted to decrease data transmission. Furthermore, some of the local tasks are offloaded between fog nodes in need to decrease energy usage. The proposed approach is considered to be energy efficient and supports less latency. However, it does not support

scalability and interoperability. Furthermore, a way of exchanging information between sensor nodes is not supported.

2.4.3 Distributed Ledger Technology DI

In Distributed Ledger Technology (DLT) approaches to DI, the functionality and data are distributed among all participant nodes. DLT serves as a shared, digital infrastructure for applications beyond financial transactions. DLT enables the operation of highly scalable, available and append-only distributed databases (known as the distributed ledger) in an untrustworthy environment [66].

Recently, the authors in [67–69], proposed an approach in support of distributed intelligence. In [67], the system focuses on privacy and security. The privacy leakage is avoided due to the fact that gateway requires the user to add consent before anyone gets access to the data. Authentication and secure management of privacy are ensured using digital signatures. The proposed approach achieves security and privacy. However, the offline capability is not well supported in their approach and IoT resource constraints i.e., power consumption are not taken into consideration. The approach also lacks scalability and interoperability.

Similarly, a distributed intelligence approach is proposed in [68], the architecture consists of six main components. The proposed approach is scalable, secure, energy efficient, lightweight and supports transparency, where low-level details are hidden. However, the offline-capability is not considered, and the elimination of redundant data coming from WSNs still remains unsolved. It also lacks a mechanism to deal with interoperability.

IOTA tangle architecture is an evolving DLT platform aimed at addressing transaction costs, mining and scalability issues (in the context of Blockchain technology) [66], that are related to IoT. The architecture of a tangle, which is central to IOTA has been proposed to achieve DI.

For example, in [70] a distributed intelligence approach that adopts the IOTA protocol is proposed. It establishes an infrastructure network for smart homes, paying particular attention to scalability. All of the home IoT nodes in the system are linked with neighbouring nodes to exchange information and ensure synchronisation with the ledger. The approach is only suitable for small scale applications and would lead to higher energy to be consumed in all nodes since proof of work computation is performed on local IoT nodes. The offline capability is not supported and is not decentralised because it's fully based on a coordinator.The approach also does not support interoperability in the design of the architecture.

Similarly, the work proposed in [71] in which a macro standardisation of energy consumption per transaction for the IOTA decentralised cryptocurrency is introduced. They measure the IOTA's Proof of Work (PoW) on local node. The energy consumption of App usage was analysed in different ways including battery states and Android Debug Bridge (ADB). The battery states is a tool included in the Androied framework that collects batter data on devices, while ADB is responsible for dumping the collected battery data to develop a machine and create a report. The results indicates that PoW offloading can save energy consumption of IoT devices. However, privacy, scalability and security are not considered in their design.

In [72], the authors proposed a system architecture that consumes the available computational resources of public volunteer devices for solving the expensive computational puzzles. The system architecture consists of several components including: core client, tangle subscriber, database and scheduler service each of which is responsible for a specific task. The core client is concerned with maximising the resource usage of the device. It is also responsible for communicating with the server services for requesting tasks and for reporting results within deadlines. The tangle subscriber is concerned with publishing every new transaction that it receives and adds it to the ledger and the subscriber service gets notified about the new transactions continuously. Upon receiving new transactions, the subscriber service

adds all new valid 3 transactions to the database. The database component is concerned with storing every new transaction published by the tangle. The scheduler service is concerned with handling the initiated requests and results from all the clients. The proposed system is energy efficient since it offloads heavy computational tasks from constrained IoT devices.

A novel Dual Signature Masked Authentication Message (DSMAM) is proposed in [73]. It ensures that data is generated by the trusted IoT device i.e, authenticity and enhances the classical IOTA masked authentication message (MAM). The users can securely and privately share messages with each other using the MAM channel. The communication can take place using the distribution of three input parameters 'Root', 'Public Key of IoT device' and a 'SideKey'. The receiver can then fetch the data message payload from the respective MAM channel. Only the valid receiver having the correct combination of these three input parameters can fetch and decode the message in the encrypted packet of MAM payload. The proposed approach is energy energy efficient since it uses a Proof of Work (PoW) computation offloading and supports privacy and security. However, scalability is not considered in their design.

In [74] a system architecture is proposed to ensure privacy. The architecture adopts a cuckoo filter in the IOTA lightweight client to avoid address reuse upon pruning of the tangle history. The lightweight client node holds a cuckoo filter that consists of all addresses used previously to receive funds. If the user requests a wallet to generate an address to receive funds, the cuckoo filter is checked if it contains an address, if it does not, then the address is returned as a fresh address, and a copy of the address is sent to the cuckoo filter to avoid future reuse. The system ensures the privacy of the users in an efficient way.

A general purpose decentralised attribute based access control mechanism using IOTA tangle is introduced in [75]. The technique ensures that owner defines and manages access control over his objects. It describes a security policies and the level of authorisation granularity of access rights and stores them on the tangle. In this way, it guarantees distributed

auditability and prevents the user from fraudulently denying the granted access rights. In the case of access request, the owner sends the authorisation token to the requester only if the requester meets the conditions defined in the access control policy. The proposed access control mechanism is scalable and achieves privacy by using masked authenticated messaging.

Most recently, an approach to distributed intelligence is introduced in [25]. The approach is called a Scalable Distributed Intelligence Tangle-based approach (SDIT). The approach is concerned with solving some of the IoT issues such as scalability, energy usage and decentralisation by adopting the IOTA protocol. A computation offloading mechanism has been developed to ensure that constrained IoT devices do not engage in performing heavy computation tasks. The proposed approach is scalable, energy-efficient and decentralised. However, security and elimination of redundant data are not considered. Also, they outline to develop a mechanism to deal with interoperability and offline capability as part of their future work.

Similar to [70, 25], the work presented in [76] in which a distributed intelligence architecture is introduced. The architecture consists of three main components, including IoT nodes, super-node, and Masked Authenticated Messages (MAM). IoT nodes are responsible for sensing the environment. The super-nodes are mainly concerned with aggregating data and packaging them into a transaction, which then are sent to the IOTA network. MAM is mainly responsible for managing access control over the data stored in the tangle. The approach achieves privacy with the use of MAM. However, the approach lacks an offline capability and interoperability mechanisms that are critical to IoT applications, and is neither energy efficient nor scalable due to the lack of an efficient clustering mechanism.

2.4.4 Service Oriented Computing DI

In Service Oriented Computing approaches to DI, the functionality is supported as services that are distributed in all levels of the system. In addition, it ensures that software components are re-usable and interoperable through service interfaces.

Recent work is introduced in [77]. The LEONORE system to support distributed intelligence. LEONORE is built up using a service-oriented architecture and supports several application components in large-scale IoT deployments. The LEONORE framework works according to two phases: push-based and pull-based. The pull-based is responsible to independently propose a run time method, while provisioning of push-based, responsible for providing control for the application by providing software updates and maintains security. The proposed framework is energy efficient and scalable. However, offline capability, security and privacy are not well supported.

A service oriented computing is developed for the agriculture application in [78]. The architecture contains components that are related to farming and farmers such as monitoring of the farm and it describes how farming should utilise such components. In this way, it is likely to reduce the expenditure for farming, minimise the labor, improve the crop yielding and suitability of the crop for a particular soil. However, the proposed approach lacks scalability, offline capability and privacy, which are fundamental requirements for IoT.

Similar to the above work, but differs by adopting micro-services is the approach proposed in [79]. The architectural style contains various patterns, including client-server, peer-to-peer and cloud computing patterns. The framework proposes the adoption of micro-services. Micro-services adopts a simple Application Programming Interface (APIs), which are thinly layered (light weighted compared to Service Oriented Architecture (SOA)). Some might argue that micro-services are similar to SOA. However, both apply service based architecture that enables service use and reuse. The differentiation is in the way where processing functionalities are triggered, where data is processed, architectural style, architectural charac-

teristics, service characteristics and capabilities. The proposed approach is energy efficient and interoperable. However, scalability that will accommodate the growth of IoT devices is not well supported. Furthermore, other challenges such as privacy, offline capability and security are not considered.

2.4.5 Hybrid DI

A hybrid approach combines the functionality of various algorithms from DI categories. The main aim of the hybrid approaches is to reduce the effect of the limitations of the aforementioned DI categories.

Most of the introduced hybrid approaches are mainly concerned with issues related to the management of data, processing of data in a timely manner and privacy, by mixing different algorithms from various technologies to achieve the required goal. In early studies, distributed intelligence was achieved by integrating the architecture of WSNs at the various level of IoT in support of distributed intelligence [13]. In such approaches, the aim is that wireless sensor network architecture is to be connected to the Internet, and the intelligence should be distributed at several layers. These approaches are considered efficient in regard to energy usage. This is because data processing is distributed among all layers and they provide flexibility. However, they lack scalability, privacy and offline capability, which are considered fundamental challenges of the IoT domain. They also lack support for interoperability.

In [80], the authors applied fog computing as a means to support distributed intelligence by setting up an architecture that is made up of three layers. The sensory layer is concerned with the transmissions of data to the upper layer. A fog layer plays the role of data processing transferred from the sensor nodes. The cloud computing layer is used for heavy processing of data. The system is suitable for timely response applications and is energy efficient since processing is performed near the data source. However, the approach lacks support for other IoT technical challenges such as scalability, offline capability and privacy.

Recently, a computing paradigm called Edge Mesh is being suggested in [81] to enable distributed intelligence in IoT. Decision-making task is distributed through the network among devices, instead of data being transmitted directly to a central location for processing. Combining the use of both computation and data, tasks are exchanged with Edge Mesh through a network of routers and edge devices. The architecture of Edge Mesh comprises of several devices. First of all, the end devices are concerned with actuation and sensory purposes. Secondly, edge devices can be used to process and connect end devices. Thirdly, routers are being utilised to transmit data. Finally, the cloud is increasingly being used to perform advanced analysis of data. The incorporation of Edge Mesh could bring various benefits such as improved security and privacy. Nevertheless, some will be concerned about privacy and security, but how privacy can be accomplished is not taken into account. Furthermore, how scalability is ensured at the end devices is not provided.

In comparison to the above, the research in [82] suggested an AI-based distributed intelligence solution. The solution incorporates the use of both cloud based and edge controller to enable distribute intelligence. To be specific, it has been shown that the cloud-based controller is capable of providing intelligence at high level. The edge controller is designed to support intelligence at low level. The advantages of their research are reducing response time and loosening rules requirements. However, the approach lacks a mechanism that allows offline capability and privacy preserving.

A hybrid distributed intelligence approach is proposed in [83]. The approach comprises of several layers, including IoT layer, fog layer, and cloud layer. The IoT layer contains of WSNs that are mainly concerned with data collection from the environment, then data is transmitted to the fog layer, which is responsible for the processing. The cloud layer is responsible for heavy computation. The architecture is energy efficient and scalable. However, it lacks an offline capability mechanism and privacy.

A hybrid distributed data flow is introduced in [84, 85]. All levels of the architecture comprise of fog nodes and these fog nodes operate based on their computing resources. It has edge input/output and nodes for computing data. The input nodes are used to communicate and transmit data to the compute nodes. The computing nodes are mainly concerned with data processing. The proposed system takes into account the scalability. However, offline capability and privacy remains unsolved.

A novel tiered architecture to allow distributed intelligence is presented in [86]. The three-tier architecture manages gathered data from sensors. The regional tier contains fog nodes that are mainly concerned with data combination and pre-processing. The cloud data centre is hosted to deal with heavily computations of data. The proposed system reduces energy usage by utilising fog nodes to process data. However, scalability is not well maintained within the system and privacy is not considered. Furthermore, it applies static orchestration, leading to system failure.

Most recently, the authors in [29] propose a novel approach in support of distributed intelligence. The approach consists of (1) IoT nodes; (2) tangle to manage transactions; (3) Proof of Work (PoW) server and mobile agents to gather transactions data. All of the IoT nodes in the system are linked with neighbouring nodes to exchange information. IoT devices deployed to sense data. A PoW server contains high power resources and deals with heavy computations. The mobile Agents are triggered to gather transactions data along their identified routes. The mobile-Agent Distributed Intelligence Tangle-Based approach MADIT [29] is scalable and shares information between sensor nodes. Furthermore, the system is energy efficient and decrease the data that needs to be collected. However, offline-capability is not considered, but outlined as future work.

In [87], a distributed Internet-like architecture is introduced. The system has three main layers. The first layer is mainly concerned with real world objects such as sensor devices. The second layer communicates and coordinates the tasks coming from the first layer. The

final layer is mainly concerned with user requests and services. The system is scalable and enables interoperability. However, dealing with other challenges such as offline capability is not introduced and does not consider resource constraints posed by IoT devices.

The authors in [88] introduced an approach that relies on Mobile Cloud Computing. It was proposed that sensing and processing be merged in the architecture of the network and it requires that the application workload to be shared among server side and nodes. This proposed approach allows the data to be analysed and monitored in real time. However, the approach in [88] is not scalable and is not designed to cope with time-critical applications. Furthermore, offline capability is not considered but outlined as future work.

Another work is introduced in [89]. The work describes a system architecture that consists of data collection, self-organisation and reasoning. The data collection is to be used for gathering and communicating data to a gateway for further processing, while self-organisation is responsible for proper management such as configuration, discovery and duplicated identification checks. The publish-subscribe is responsible for disseminating/acquiring data, which can be handled by the MQ Telemetry Transport (MQTT) protocol. Finally, the reasoning plays the role of extracting knowledge based on context using a Naïve Bayes method. The approach is scalable and delay is avoided due to the use of bayesian reasoning. However, the ability to work in critical cases is not supported.

In [90], the authors present a hierarchical distributed computing architecture. Layer one is largely used for computations. It is designed to provide centralised control purposes and wide city monitoring. The second layer consists of the intermediate computing nodes that recognise and respond to potentially dangerous activities and act upon the risks identified. The third layer consists of high-performance edge devices, which are low-powered linked to a group of sensors that manages raw data from sensors and analyses data promptly. The fourth layer is comosed of sensor nodes to track environmental changes . The benefits include low

latency, efficient responses in real-time and energy efficient. Nonetheless, issues related to the IoT, such as security and scalability, are not taken into account in the proposed approach.

2.4.6 Intelligence Levels

Intelligence-levels in DI approaches aim to indicate where raw data processing occurs and where processing functionalities are triggered. In each distributed intelligence approach, the level of supported intelligence can be either low-level, high-level or both levels supported. For example, the works reported in [49–54] focuses on high-level intelligence by enabling data processing and processing functionality to occur in the cloud. Other research efforts that primarily support high-level intelligence are proposed in [67, 70, 25, 76] in which nodes with advanced computational resources i.e., energy is responsible for heavy computations and data processing.

Low-level intelligence is mainly concerned with enabling data processing to occur at the edge of the network. For example, the DI approaches described in [57–59, 61, 26] in which the main idea is to provide low-level intelligence to the data at the edge. In addition, cooperation among physical IoT devices by means of data sharing. Such research efforts would lead to a significant decrease in energy consumption since data computation is performed near the IoT devices and faster response time is obtained.

Both high-level and low-level intelligence are supported in all levels of the IoT system to ensure minimum resource usage. For example, the DI approaches proposed in [77, 13, 81, 82, 86, 29] focuses on enabling high-level and low-level intelligence. In these approaches, low-level intelligence is supported by enabling data to be shared among various IoT devices and perform computation to provide usefulness insight out of the data. On the other hand, high-level intelligence is supported by making use of the cloud to perform big data analytics [82].

2.4.7 Similar Approaches and Algorithms

This part describes distributed intelligence approaches and mobile agent itinerary planning algorithms that are similar to the research in this thesis. These approaches and algorithms are described as follows:

Distributed Intelligence Approaches

There are several distributed intelligence solutions proposed in the literature [51, 59, 70, 77, 81]. These solutions are based on cloud computing, mist computing, distributed ledger technology, service-oriented computing and hybrid as described in 2.4. For example, the work reported in [51] the Cloud-Assisted Real-time Methods for Autonomy (CARMA), which aims to design, develop, and test cooperative automated driving technology, based on a distributed control system. The system consists of three-tier distributed computing architecture namely: the CARMA Vehicle, the CARMA edge and the CARMA core. The executions of autonomous functions are distributed between the on-board system and the cloud-based high performance shared back-end system.

The authors in [59] proposed a framework that compromises of four layers, namely: IoT physical devices layer, mist nodes layer, fog nodes layer, and cloud nodes layer. The IoT layer is responsible for transmitting data to the upper layer, which consists of mist nodes. The mist nodes are mainly responsible for data processing. The cloud node layer is used for heavy computations task. The proposed framework is energy efficient because of the use of mist nodes and has less latency. However, it lacks scalability at the physical layer and does not support security, privacy and offline capability.

A distributed intelligence approach that adopts the IOTA protocol is proposed [70]. The introduced approach develops an infrastructure network for smart homes. The IoT nodes are connected with neighbouring nodes to exchange information and the data is eventually sent to the tangle. The approach is only suitable for small scale applications and would lead to

higher energy consumption since the Proof of Work (PoW) computation is performed on local IoT nodes. In [77] a system called LEONORE to support distributed intelligence is introduced. LEONORE is made up of components from the service-oriented architecture. It supports several application components in large-scale IoT deployments. The proposed framework is energy-efficient and scalable. However, offline-capability, energy-efficiency, security and privacy are not well supported.

A hybrid paradigm called edge mesh is proposed in [81]. The architecture of edge mesh consists of four types of devices including: end devices, edge devices, routers, and cloud each of which performs a particular task. The end devices are responsible for sensing the surrounding area. The edge devices are responsible for decision making and facilitate interaction between end devices. The routers are responsible for routing data between edge devices, while the cloud provides computing resources such as storage and processing. The framework has several advantages such improved security and privacy. Nevertheless, how privacy can be accomplished is not taken into account. Furthermore, how scalability is ensured at the end devices is not provided.

Mobile Agent Itinerary Planning Algorithms

There are several multi-mobile agent itinerary planning algorithms in the literature [19–21]. For example, a spawn multi-mobile agent itinerary planning (SMIP) approach is introduced in [19] to mitigate the substantial increase in cost of energy and time used in the data collection processes. The SMIP works by allowing the main mobile agent to spawn other mobile agents with different tasks assigned from the main mobile agent. Similarly, a multi-mobile agent itinerary planning approach called GIGM-MIP approach is proposed in [20] . In GIGM-MIP each group in the network will be visited by more than mobile agent. The data size of the source nodes in each group decides how many mobile agents will be sent to that particular group. The approach balances the accumulated data between mobile agents.

However, having more than one mobile agent dispatched to visit one group would result in an increase of the mobile agent migration hops. In addition, multiple mobile agents carrying the same aggregation code within a single partition would result in an increase in energy consumption.

Another multi-mobile agent itinerary planning approach is introduced in [21]. The approach consists of four main phases including: visiting central location (VCL) selection algorithm, source-grouping algorithm, single itinerary planning (SIP) algorithm and its iterative algorithm. The VCL selection algorithm is responsible for calculating a high source node density. The source-grouping algorithm is mainly concerned with grouping nodes in the network and assign mobile agents to a specific group. The itinerary planning of the mobile agent is determined by using a SIP. The iterative algorithm is used to ensure that all sensor nodes are assigned to the allocated mobile agents. The proposed algorithm considers a cluster-based technique in which the source nodes are arranged geographically and distributed in several clusters. This limits the use of proposed algorithm when the nodes are sparsely deployed.

2.4.8 Evaluation of Distributed Intelligence Approaches

The evaluation of distributed intelligence approaches covers 30 representative approaches. As described in Table 2.1, the evaluation aims to assess existing research using the categorisation presented in Section 2.4 and the identified challenges in Section 2.2.

According to Table 2.1, the least implemented challenges of distributed intelligence are offline capability, security and scalability. Many research efforts support only two or three of the IoT DI challenges, which are potentially critical for many IoT applications. In terms of offline capability, it has not been implemented in most of the research efforts according to Table 2.1.

DI Categories	AP	RC	SC	SE	PR	OC	IO	IL
Cloud Computing	[49]	✓	✓	X	✓	X	X	H
	[50]	✓	✓	X	✓	X	X	H
	[51]	✓	X	X	✓	X	✓	H
	[52]	✓	X	✓	✓	X	X	H
	[54]	X	X	✓	✓	X	✓	H
	[53]	X	✓	X	✓	✓	X	H
Mist Computing	[57]	✓	X	✓	X	X	X	L
	[26]	✓	X	✓	X	X	X	L
	[65]	✓	X	✓	X	X	X	L
	[58]	✓	X	X	✓	X	✓	L
	[59]	✓	✓	X	X	X	✓	L
	[60]	✓	✓	X	✓	✓	X	L
	[61]	✓	✓	X	X	X	X	L
Distributed Ledger Technology	[67]	X	X	✓	✓	X	X	H
	[68]	✓	✓	✓	X	X	X	H
	[70]	X	✓	✓	✓	X	X	H
	[25]	✓	✓	X	X	✓	X	H
	[76]	X	X	X	✓	✓	X	H
Service Oriented Computing	[77]	X	X	✓	✓	X	X	H&L
	[78]	✓	✓	✓	X	X	X	H&L
	[79]	X	✓	✓	✓	X	X	H&L
Hybrid	[13]	✓	✓	X	X	✓	X	H&L
	[80]	✓	✓	✓	X	X	✓	H&L
	[81]	✓	✓	✓	X	X	X	H&L
	[82]	X	✓	X	X	X	✓	H&L
	[83]	✓	✓	✓	X	X	✓	H&L
	[84]	X	✓	X	✓	X	✓	H&L
	[85]	✓	X	✓	X	X	X	H&L
	[86]	✓	✓	✓	X	X	X	H&L
	[29]	✓	✓	X	X	✓	✓	H&L
	[87]	X	✓	✓	✓	X	✓	H&L
	[88]	✓	X	✓	X	X	X	H&L
	[89]	✓	X	✓	X	X	X	H&L
	[90]	✓	X	X	✓	X	X	H&L

AP: Approach
RC: Resource Constraints
SC: Scalability
SE: Security
PR: Privacy
OC: Offline Capability
IO: Interoperability
IL: Intelligence Levels
L: Low-level
H: High-level
H&L: Both High-level and Low-level

Table 2.1 Evaluation of distributed intelligence approaches

Among these research efforts, the work proposed in [26] provides an interesting case study on applying distributed intelligence in smart factory applications. When applying the fog computing technology, fog nodes are described by the hardware and software architecture. Therefore, real time analysis is supported and low-latency is minimised. It has been indicated that the fog computing approach is able to reduce bandwidth because processing is occurring within the network. The work proposed by the authors in [52] relates to the security issue in which a cloud based approach is used to deal with attacks (e.g, data theft) where two additional security features are added. Consequently, better security can be achieved through the proposed built in features in addition to existing cloud security features. In regards to the offline capability issue, the authors in [13] introduced offline capability in their architectural design, but without giving details about the implementation and evaluation.

2.4.9 A Summary of Shortcomings of Existing Distributed Intelligence Approaches

From the above discussion, it can be seen that most of the current approaches to enabling distributed intelligence in IoT are subject to all the problems inherent in distributed systems. Firstly, the approaches suggested usually depend on centralised architecture for processing data [64] that offers high cost and unacceptable delays for many distributed applications. These include health monitoring, autonomous driving, emergency response etc. Furthermore, transferring data to a central location requires high network bandwidth [91, 8].

Bottlenecks and delays are expected from the communications between the devices and the centralised system [88]. Tracing data stored in the cloud is very difficult and lacks accountability. IoT that is based on central infrastructure requires trusting third party for dealing with data and the storage of data in the cloud has the possibility of that data to be deleted or tampered with [92]. Besides, solutions that fully relies on fog computing is considered

to have problems with security and privacy [62, 28]. They also lack interoperability and interaction models [28].

Previous research recommends that IoT needs to shift away from a central point of control [93]. Approaches based on Blockchains introduce overhead and performance issues [94, 95]. Therefore, developing a standardised approach is required to define IoT data. For example, the one provided in the IOTA Identification of Items (IDoT) [96] that aims to protect the network, too. Also, blockchains require transferring a large portion of data, which is the header block, leading to a wastage of resources [97].

2.5 Mobile Agent and Distributed Intelligence

Mobile agents (MAs) are software abstractions that perform data processing autonomously while physically migrating between nodes in the network to enable the sharing of data amongst participants' nodes [98]. MAs facilitates the flexibility and scalability problems of centralised models [99], and are commonly deployed in Wireless Sensor Networks (WSN) for data collection and in-network processing. Many MAs approaches dispatch agents to collect data from the network rather than sending the data back to a gateway. The benefits of using MAs as stated in [100] include: reduced task redundancy, lower network bandwidth and reduced network load.

2.5.1 Single Itinerary Planning (SIP)

Over the last few years, mobile agent itinerary planning has drawn many researchers' attention in the field of WSNs. The interested readers are referred to the recent survey [101] and the references therein for a comprehensive review of the mobile agent itinerary planning approaches in WSNs. It is noticed that many of these research efforts are towards optimising and constructing an energy efficient itinerary planning mechanism [102–109]. One piece

of the early work on Single Itinerary Planning (SIP) is proposed in [102], in which the authors have developed two heuristic algorithms to calculate the itinerary of the single mobile agent. Two algorithms are named local closest first (LCF) and global closest first (GCF) are proposed. LCF operates by finding the next node in the shortest distance to the current node while GCF aims to find the centre's closest node. The proposed algorithms are static and can save energy as the itinerary planning needs calculated only once.

However, the approach does not scale well if a single MA has to visit thousands or millions of sensor nodes. It also leads to big delays in reporting the data because of using only a single MA, which has to move between all sensor nodes in the network.

The authors in [106] proposed a mobile agent directed diffusion (MADD) approach, which is based on a directed diffusion algorithm. The approach works by making the sink to initially get diffused with an interest for notifications of low-rate exploratory events that are intended for path setup and repair. The proposed approach reduces energy consumption because it relies on directed diffusion agent trip and eliminates data redundancy. However, it introduces a delay since a single mobile agent is routed among sensor nodes and is not suitable for large scale sensor networks.

The mechanism introduced in [107] is an improvement over the MADD approach. There are three phases of the proposed approach: First, the MA action phase; second, the dissemination phase of exploratory data; and third, the controlled setup phase of gradients. In the first phase, the sink node floods its neighbor with interest messages in the controlled setup phase of gradients. It sets up an itinerary to the next hop based on two metrics: (1) the remaining energy threshold and (2) the minimum hop count. In the exploratory data dissemination phase, the main aim is to discover the source nodes and to establish the TargetSrcTable (containing targets and source node information) in each target node. Sensory data is stored in the cache of each source node, waiting for collection in the next phase. The

approach is energy efficient. However, due to the same limitations as above research efforts, the solution lacks scalability due to the use of a single MA.

The work proposed in [110] is called Itinerary Energy Minimum Algorithm (IEMA). IEMA extends LCF by considering estimated communication costs. The aim of IEMA is to achieve better energy efficiency. It focuses on choosing the first visiting node among the remaining set of source nodes as well as an optimal source node as the next source node to be visited. The algorithm estimates the energy costs of the alternative choices of the first node. The proposed schema is energy efficient. However, it does not scale to a large number of sensor nodes since only a single mobile agent is used. In addition, it does not take into consideration the growing size of collected data of the mobile agent when visiting a sequence of nodes.

In [111] an event-driven adaptive method is proposed, which implements a semi-dynamic routing strategy based on a two-level genetic algorithm. A fitness function is constructed to meet the desired detection accuracy while minimising energy consumption and path losses in a global sense. The sink node has necessary predetermined knowledge for performing the global optimisation, such as the geographical locations of all sensor nodes. The sink node is responsible for computing the routes for mobile agent. The mobile agent follows the route computed by the genetic algorithm according to the fitness function. The proposed approach is energy efficient. However, it lacks scalability since a single mobile agent is used, which is not suitable for time critical applications that require real time processing.

2.5.2 Multiple Itinerary Planning (MIP)

To alleviate the inherent problem caused by the use of a single mobile agent, a number of Multiple Itinerary Planning (MIP) approaches have been developed.

Authors in [103] investigate the role of multiple mobile agents and propose a novel a routing itinerary algorithm called DMAIP. The idea is to group all sensor nodes into multiple

itineraries for a mobile agent. The approach consists of three main components, including: remote user, sink node and sensor node. The remote user assigns a task to a sink node. When the sink node receives a task it traverses the network topology to generate a spanning tree, and assigns each path to one of the mobile agents.

A new immune inspired algorithm, called the Clonal Selection Algorithm for Multi-agent Itinerary Planning (CSA-MIP), is proposed by the authors in [105], in order to solve the MIP problem in WSNs. The important components of CSA-MIP includes: encoding, mutation operators, cloning of antibodies and affinity function. CSA-MIP is based on two computational stages called Stage I and Stage II. Each stage involves a different mutation operator. The proposed approach has less computational complexity and is energy efficient.

A novel central location-based MIP (CL-MIP) framework is presented in [21]. The framework consists of four parts including: visiting central location (VCL) selection algorithm, source-grouping algorithm, SIP algorithm and its iterative algorithm. The VCL selection algorithm is used to calculate a high source node density. The source-grouping algorithm is responsible for grouping nodes and assigning mobile agents to particular groups. The SIP algorithm is adopted to determine the itinerary of mobile agents. Finally, the iterative algorithm is mainly concerned with ensuring that all source nodes are assigned to the allocated MAs. The CL-MIP algorithm considers a cluster-based technique in which the source nodes are arranged geographically and distributed into several clusters. This indicates that the CL-MIP is not applicable to be used if the nodes are sparsely deployed.

The authors in [112] proposed a new itinerary planning strategy, which consists of three phases. First, the network is partitioned into clusters according to the distance between the sensor nodes using the k-means algorithm. Second, the number of MAs is determined for each partition based on the volume of data from each source node and the geographical distance. Third, an optimised itinerary plan is produced for each partition group, identifying the source nodes to be visited according to a greedy randomised adaptive search procedure

(GRASP). This approach is scalable and delay is minimised due to dispatching of multiple agents for each group. However, this algorithm is not sufficiently robust as the data volume increases. Furthermore, the number of partitions must be manually identified by the user, which can result in sub-optimal partitions of the network.

In [108], an energy efficient itinerary planning approach is proposed. The algorithm is based on Iterated Local Search (ILS), a metaheuristic method commonly used for solving discrete optimisation problems. ILS iteratively applies a simple modification to a local search routine, each time starting from a different initial configuration, in search of an improved solution. The ILS algorithm is executed centrally at the sink which statically determines the number of MAs that should be used and the itineraries these MAs should follow. The proposed algorithm is energy efficient and avoids delays. However, the itinerary planning is deterministic and pre-defined at the sink node. Therefore, if a sensor node depletes in energy, it would result in re-constructing the paths for each mobile agent.

The authors in [109] proposed a system, which employs both mobile agents and mobile servers to collect data from sensor nodes deployed in a sensing field. Mobile agents migrate from node to node autonomously and return to the mobile server after data collection. The migration process relies on a geographic routing approach to route mobile agents. Upon collecting data mobile agents find the current location of the mobile server and returns to it with the aggregated data. The system focuses on a effective and intelligent gathering mechanism.

The authors in [113] proposed directional source grouping algorithms (DSG-MIP). The main idea is to divide the network area into sector zones with specific angles, the centres of which are the immediate neighbors of the sink node. After this, the source nodes are allocated to an itinerary within each sector zone. The route to the sink node inevitably converges on each MA's round trip and increasingly extends as the MAs travel further from the sink node [104].

A multi-mobile agent itinerary planning-based energy and fault aware data aggregation (MAEF) method is proposed in [114] to plan itineraries for MAs. The approach comprises of three main phases, including: 1) cluster head selection and construction, 2) cluster head based itinerary planning and 3) mobile agent migration and data collection. In the cluster head selection phase, the idea is to distribute the density impact factor of each node to the other sensor node, then the sensor node with the highest accumulated impact factor will be selected as a cluster head node. In the cluster head itinerary planning phase, mobile agent itineraries among cluster head nodes are constructed based on a minimum spanning tree (MST). In the final phase, the sink dispatches mobile agents to gather data from cluster head nodes.

In [20], a multiple mobile agent itinerary planning approach named as GIGM-MIP is proposed, which works in three phases. In the first phase, the network is partitioned using the$k$-means method and based on geographical information in which a set of partitions is generated and each partition can have several mobile agents. In the second phase, the number of mobile agents is determined and groups of nodes are defined for each mobile agent. Finally, the third phase is concerned with defining the itinerary that passes throughout the source nodes grouping of each mobile agent. Several mobile agents can be allocated to each partition.

The authors in [115] proposed a Scalable and Load-balanced Mobile Agents-based Data Aggregation (SLMADA) protocol, in which the itinerary of a mobile agent is dynamically decided at each hop. The whole monitoring area is divided into centric zones and it is assumed that the sink node knows the location coordinates of source nodes. A zone coordinator is selected in each concentric zone, which assists the MAs in the dispatching and receiving to and from the network. The sink node will create a set of MAs, one for each zone coordinator and dispatch mobile agents to the centric zones. This approach allows mobile agents to decide their visit sequence dynamically based on information provided by the zone coordinators.

Energy-Aware Mobile Agent Based (EAMB) is proposed in [116]. The network is divided into multiple clusters and a group of sensor nodes is assigned to one cluster. The MAs moves between cluster head nodes only, which is defined by using a minimum spanning tree (MST).

In [117], a dynamic and distributed migration protocol, called energy and trust aware mobile agent migration (ETMAM), is proposed. The main idea is to identify and bypass faulty or malicious nodes during the mobile agent migration process. The sink node dispatches mobile agents concurrently to the coordinator nodes of each wedge region in the network. Coordinator nodes are the nodes of the innermost concentric ring. Due to the need to detect malicious nodes, the whole approach is considered complicated and requires heavy computation.

In [19], the authors proposed a spawn multi-mobile agent itinerary planning (SMIP) to reduce significant rises in energy costs and time spent on data collection. This is based on the spawning agents, which allows the primary MA to spawn another MA into a single fraction. The proposal uses k-means algorithm to calculate the number of clusters based on Bayesian ratings. The sink node specifies the number of MAs and their itineraries for each partition when the partitioning is complete. The approach is energy efficient and scalable.

To summarise, most of the above approaches do not take into consideration that when the mobile agent moves along the routes, the size of collected data from sensor nodes increases rapidly, leading to higher consumption of network bandwidth.

The evaluation of the recent mobile agent itinerary planning literature covers 18 representative approaches. Table 2.2 presents the comparisons of existing research from a number of angles, including: scalability, energy-efficiency, grouping strategy, and delay. From Table 2.2, it can be seen that all SIP approaches lack scalability and suffer from delays in reporting data back to a sink node during mobile agent migration. Meanwhile, all MIP approaches can help address the issues of scalability, and avoid excessive delays.

Table 2.2 Comparison Among Mobile Agent (MA) Approaches

Categories	Approach	Scalability	Energy-Efficiency	Grouping	Delay
SIP	[102]	X	✓	X	✓
	[111]	X	✓	X	✓
	[106]	X	✓	X	✓
	[107]	X	✓	X	✓
	[110]	X	✓	X	✓
MIP	[103]	✓	✓	✓	X
	[104]	✓	✓	✓	X
	[105]	✓	✓	✓	X
	[108]	✓	✓	✓	X
	[109]	✓	✓	✓	X
	[114]	✓	✓	✓	X
	[113]	✓	✓	✓	X
	[20]	✓	✓	✓	X
	[115]	✓	✓	✓	X
	[116]	✓	✓	✓	X
	[117]	✓	✓	✓	X
	[19]	✓	✓	✓	X

2.6 Hardware-based Security Primitives for IoT

The inherently distributed nature of distributed intelligence approaches provides several vulnerable points to compromise security. Consequently, it is a fundamental challenge to ensure authenticity, integrity, confidentiality and availability among various integrated devices [118, 119].

Hardware security primitives have been put forward as a promising security primitive to achieve security. This refers to hardware devices that are used as fundamental building blocks to create security solutions [119]. It consists of Physically Unclonable Functions (PUFs) and True Random Number Generator (TRNG). On one hand, PUFs are an integrated circuit, which has the ability to generate secret responses and cryptographic keys by applying inherent physical variations from manufacturing [120]. PUFs can be implemented on programming micro-controllers. In PUFs, inputs are referred to as “challenges” and outputs are “responses.” A challenge and its associated response are known as a challenge-response pair (CRP). On the

other hand, TRNG are hardware components that is responsible for producing random bits according to the outcome of unpredictable physical processes such as the device's internal thermal noise [121].

Recent research focused on exploiting the benefits of hardware security primitives for Cyber-Physical Systems (CPS) [122, 120, 123, 124]. The authors in [122] have proposed a novel integrated TRNG-PUF architecture based on Photo-voltaic (PV) solar cells. The proposed architecture works according to two phases including: Training phase and Run phase. The training phase is mainly concerned with learning the entropic nature of PV solar cells and sets an optimal sampling interval, which is a vital step to set optimum TRNG throughput. The run phase is mainly responsible for obtaining sensor response in either a dynamic (large variation) response to produce TRNG output or static (stable) response to generate PUF output. The proposed integrated architecture could be beneficial in space-limited CPS.

Another PUF design that is specifically targeted for use in IoT applications is proposed in [124]. The architecture of the proposed PUF consists of a microcontroller, eight piezo sensors, eight 100 K resistors and an ac voltage source, each of which is responsible for performing a specific task. The proposed PUF should be considered a weak PUF as it is designed to have only one possible challenge-response pair (CRP). The reason there should only be one pair is because the response generated by the PUF is a result of comparing the intrinsic characteristics of the piezo sensors. The proposed PUF approach can be incorporated into IoT devices as a cybersecurity solution.

A novel solar cell based PUF that leverages the intrinsic variations present in solar cells is introduced in [120]. The proposed design utilises a microcontroller to read the open-circuit voltages (Voc) of a selection of solar cells and generate an associated response. The proposed design was implemented using amorphous silicon solar cells, monocrystalline solar cells, and polycrystalline solar cells. A microcontroller is responsible for capturing voltages output and

converting them to digital values. The PUF uses these values to generate a 128 bit response by comparing the voltages in a pre-determined pattern. Each bit in the generated response is a direct result of a comparison made between the output voltages of two different groups of solar cells. The proposed approach ensures the security of IoT devices without adding hardware.

A novel weak PUF design using thermistor temperature sensors is proposed by the authors in [123]. The design uses the differences in resistance variation between thermistors in response to temperature change. The approach is based on 8 thermistor temperature sensors. Each sensor is connected to a microcontroller in the configuration. A microcontroller is used to compare readings from groups of thermistor temperature sensors to generate a weak response. An algorithm is used to process the individual voltage data and construct a 128-bit response. It produces a response by making a series of comparisons between total output readings for predetermined groups of a given component. It assumes that each component should have the same reading and any differences are solely due to their intrinsic variations. The approach has shown an improved overall reliability with regards to changes in temperature.

2.7 Challenges and Opportunities

This chapter discussed several issues and challenges that are important in a distributed intelligence approach. Also, it examined more than 30 research efforts to evaluate distributed intelligence approaches. these challenges are research fields that needs to be further investigated. The related references for each challenge is provided in which the interested reader can use to further look into a specific challenge.

The challenges in developing a distributed intelligence approach/platform that could potentially support all of the challenges is summarised as follows:

– Both DI and DLT are still in their early stages and require further experimentation. Distributed intelligence is considered to be very critical in determining the success of almost any IoT applications such as smart parking. The main reason behind it is that it requires the placement of where methods should be invoked/triggered and where data should be processed. This needs an effort to where distributed intelligence should be put and activated. This process requires several stages such as business logic, energy efficiency and computation efficient. Distributed ledger technology (DLT), will change the infrastructure for the Internet of Things into collaborative distributed participants. The IOTA technology will tackle many of the IoT issues by enabling scalablity, efficient processing of data, security and privacy [29, 25].

– There is an overall lack of a DI platform and approach that can provide an efficient framework for other researchers to test alternative approaches and distributed algorithms. For example, in order to design and develop a *new hybrid* DI approach that combines various technologies such as Network Function Virtualization (NFV) [125], Mobile Cloud Computing [126], Multi-Agent system [127, 128], Distributed ledger technology [129], is yet to be developed.

– Ensuring where to place the intelligence is a major research question that should be taken into consideration when developing a distributed intelligence approach. Other related research questions that should be investigated including: How IoT devices should cooperate with each other to support low-level intelligence? Where should heavy computations be invoked? Who is responsible for the distribution of tasks between devices? When to trigger data processing and where? These are all research questions that need to be investigated further. The computation tasks should be distributed among various devices depending on the available resources of each device. For example, IoT devices with higher resources can perform data processing on behalf of constrained IoT devices. Therefore, the tasks should be distributed among different

devices depending on the application requirements and resources available on the devices. Mist computing can work on the edge of the network and can deal with issues related to security, data processing and access control, etc. This requires mist devices to be robust and flexible. It is a challenge to manage all the tasks concurrently when developing distributed intelligence approaches.

2.8 Proposed Solution and Theories

This section briefly describes why distributed ledger technology is suitable for IoT. In addition, it describes the theories of the proposed distributed intelligence approach and the multi-mobile agent itinerary planning approach.

DLT can be used to eliminate the requirement of a central entity from the IoT architectures. IOTA technology is a decentralised public ledger that is used for organising the data and executing the transactions. IOTA technology offers a significant benefit including: scalability, zero-fees transactions, resiliency and decentralization. In addition, the different types of nodes developed in the IOTA technology e.g., full nodes and light nodes are suitable for the resource constraints IoT devices, such as power consumption and limited storage. Chapter 3 describes the features of the IOTA technology in detail.

2.8.1 A New Enabling Approach

The concept of IOTA Full Node [96], which can be defined as a node within the tangle architecture that is capable of finding neighbours and communicating with them, attaching transactions to the tangle, bundling and signing, tip selection, validation, and performing the Proof of Work (PoW). IOTA full nodes hold a complete copy of the tangle.

The concept of light nodes [96] that participates in the network and can be defined as a node within the tangle architecture that relies on the full node to interact with the tangle; it

distinguishes itself from other nodes in the sense that it does not store a copy of the tangle and does not validate transactions. It has been specifically designed as a lightweight node for resource constrained IoT nodes.

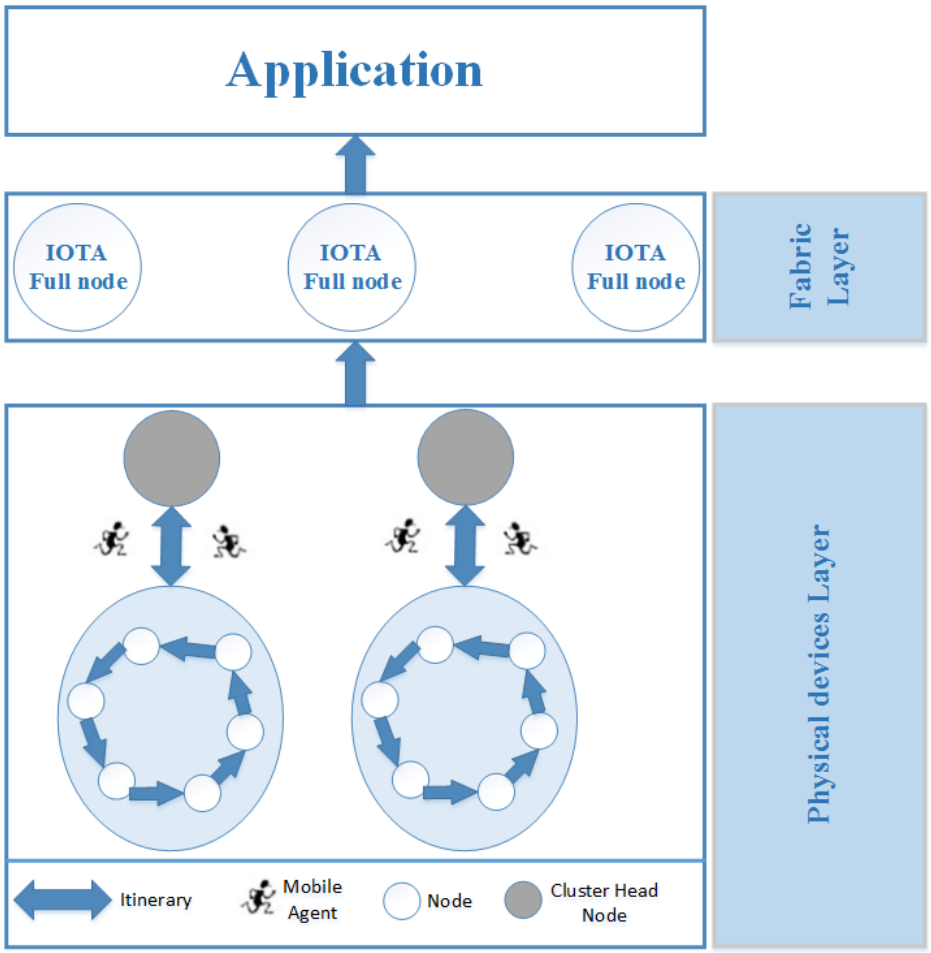

Fig. 2.2 A New Distributed Intelligence Approach for IoT.

The proposed approach consists of three layers. The first layer consists of end nodes running an IOTA light client and will act as an end points to the IOTA network. The second layer, comprises of a less unconstrained IoT devices running the IOTA full node. Finally, the application layer consists of logic and enables the developer to create decentralised applications.

Mobile agent (MA) technology can provide cooperation and information sharing among different types of nodes. A mobile agent can be defined as a piece of software that performs

data processing autonomously while moving from node to node in the network [130]. The agent can collect local data and perform any necessary data aggregation. Mobile agents can make decisions autonomously without user input. They provide flexibility in terms of decision making and reliability in terms of node failure.

2.9 Summary

This chapter presented an overview and classifications of distributed intelligence approaches in IoT. It also described the need for distributed intelligence in the IoT domain. It described distributed intelligence approaches according to the following categorisation: Cloud Computing, Mist Computing, Distributed Ledger Technology, Service Oriented Computing and Hybrid. This chapter also evaluated distributed intelligence approaches according to the identified challenges, followed by a summary of shortcomings of existing distributed intelligence approaches. In addition, it presented hardware-based security primitives with recent research effort in IoT. Finally, it described future research directions towards developing a scalable and energy efficient distributed intelligence approaches.

Chapter 3

IOTA Distributed Ledger Technology

3.1 Introduction

The Internet of Things (IoT) is considered to be as an enabling technologies for several applications. It connects physical objects together with the aim of exchanging data with other systems over the internet to enable communications between these objects [1], also referred to as Cyber-Physical Systems (CPS) [7].

Distributed ledger Technology (DLT) is emerging technology that has been developed to share data among different participants deployed over various locations all over the world. This technology provides several benefits to various IoT applications. DLT is being investigated by many researchers across the world as a promising solution to IoT. It can tackle many of the challenges imposed by the IoT systems such as the scalability, energy-efficiency, security and privacy [29, 131].

IOTA is an open source distributed public ledger technology, which records and executes transactions between machines and devices in the IoT. In 2015, IOTA foundation [132] introduced the concept of IOTA technology that has attracted much attention over the past years as an emerging peer-to-peer (P2P) technology for distributed computing and decentralised data sharing. The IOTA can avoid the attacks that want to take control over

the system. Interestingly, due to its unique and attractive features such as: scalability, zero-fees transaction, transactional privacy, security, the immutability of data, integrity and fast transaction confirmation, IOTA has been applied in various sectors beyond the cryptocurrencies. Some of the areas, digital healthcare [133], access and rights management system [75], and internet of electric vehicles [134].

3.2 Distributed Ledger Technology

Distributed Ledger Technology (DLT) can be divided into three main types based on the differentiation of the data structure used for the ledger, including: BlockChain (BC) [135], IOTA tangle (DAG) [136], and Hashgraph [137]. BC is a distributed, decentralised, and immutable ledger for storing transactions and sharing data among all network participants [138]. Hashgraph, is considered as an alternative to BC and is used to replicate state machines, which guarantees Byzantine fault tolerance by specifying asynchrony and decentralisation, as well as no need for proof-of-work (PoW), eventual consensus with probability of one and high speed in the consensus process [139]. BC has been criticised for its cost, energy consumption and lack of scalability.

To overcome these limitations, the IOTA tangle technology has been introduced as a decentralised data storage architecture and a consensus protocol, based on a Directed Acyclic Graph (DAG). Each node in the DAG represents a transaction, and the connections between transactions represent the transaction validators [136].

Blockchain technology recently started to receive attention from both academic and industry, since it offers a wide range of potential benefits to areas beyond cryptocurrency (in particular the IoT), as it has unique characteristics such as immutability, reliability and fault-tolerance [140]. It is predicted that BC will transform the IoT ecosystems by enabling them to be smart and more efficient. According to the International Data Cooperation (IDC), it is stated that 20 per cent of IoT deployments will employ a basic level of BC enabled

services [141]. This number will continue to increase for the adoption of BC in the IoT since it is in the early stages of innovation. Overall, BC is considered as an effective solution to be integrated with the IoT to achieve some of the IoT technical challenges [142].

BC is potentially able to overcome some of the IoT issues such as privacy and security [143]. However, building an energy-efficient and scalable IoT applications remains a challenge. Firstly, all BC consensus mechanisms in either private or public BC, require all fully participating nodes to retain copies of all transactions recorded in the history of BC, which comes at the cost of scalability [140]. Furthermore, IoT devices have limited processing capabilities, and memory storage which brings an issue when using BC-based architectures. Some of the IoT devices will not be able to engage in performing the Proof of Work (PoW) consensus operations due to their limited computational power and battery life. Also, IoT devices do not always come with the required storage space to place a complete copy of the BC [144].

With the IOTA tangle, transactions are directly attached to the tangle without the need to wait as they need to approve two previous transactions called tips. Hence, the tangle is more efficient than traditional BC under the well-designed architectures [70].

3.3 IOTA Platform: An Overview

Currently, IOTA is scheduled to undergo a two-part protocol upgrade, IOTA 1.5 (Chrysalis) and IOTA 2.0 (Coordicide), aimed at implementing a series of major DLT technology advancements to improve network functionality and achieve greater decentralisation. The IOTA 1.5 introduces a protocol enhancements that enable smart contract functionality, tokenized assets and stable coins, which could enable new use cases for consumer and enterprise IoT applications, an implementation of product features including: reusable addresses, UTXO, new Firefly wallet, and new libraries and Application Programming Interface (APIs) for an improved developer experience. The IOTA 2.0, implements a new consensus mechanism that

aims to improve IOTA's scalability, security, and decentralisation by removing the centralised Coordinator node. The architecture of the IOTA tangle is an evolving DLT platform aimed at addressing transaction costs, mining and scalability issues (in the context of Blockchain technology) [66], that are related to IoT. The architecture of a tangle [136], which is central to IOTA, a Directed Acyclic Graph (DAG) that offers a potentially scalable IoT-enabled applications. The tangle can be used to build IoT applications. However, the tangle has the advantages of being intuitively understandable. IOTA technology offers the necessary scalability and versatility for IoT. In the context of transactions, IOTA may promote IoT interactions. This approach radically changes the overall design, development, implementation and management process of the IoT systems.

3.3.1 The Tangle

The IOTA tangle was developed to cope with the requirements of IoT applications such as scalability, energy-efficiency and security. The tangle is built upon a Directed Acyclic Graph (DAG), which is considered to be as the ledger that stores transactions. The tangle is the data structure which consists of a collection of sites and edges [136]. In order to issue a transaction by a node, the node should work to approve two previous transactions. Choosing the two previous transaction is done by using the tip selection technique where by default is the Markov Chain Monte Carlo (MCMC) technique [136]. The main aim of the tangle network is to make all the transactions to be confirmed and to make all the unconfirmed transactions to confirmed transactions, the MCMC technique is executed n number of times. Genesis is the first transaction of the network, which is approved directly or indirectly by the other transactions.

IOTA tangle is designed in a way to enable transactional settlement to be more scalable, more the transactions made more secure and efficient the tangle gets [136].

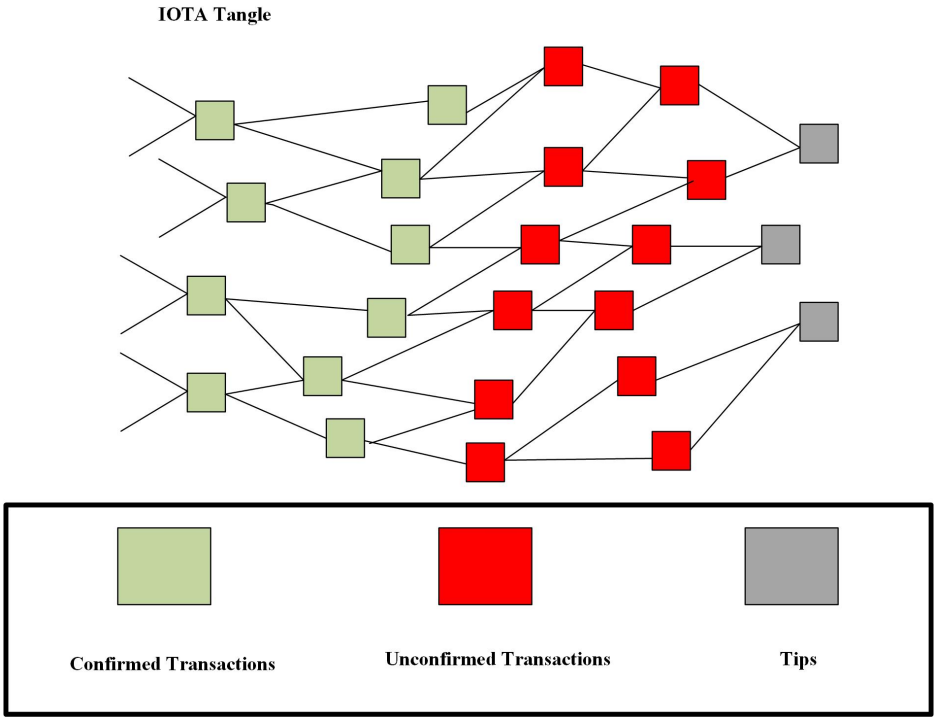

Fig. 3.1 IOTA Tangle is based on a Directed Acyclic Graph (DAG)

IOTA is based on ternary messages and it uses this system because, compared to binary, ternary computing is considered to be more efficient as it can represent data in three states rather then just two. The messages are represented by a base-3 number system where each digit is -1, 0, 1. There is also the concept of tryte, which consists of 3 trits that can be represented by 27 states. Consequently, these states are required to be with uppercase letters e.g., A-Z and the number “9”.

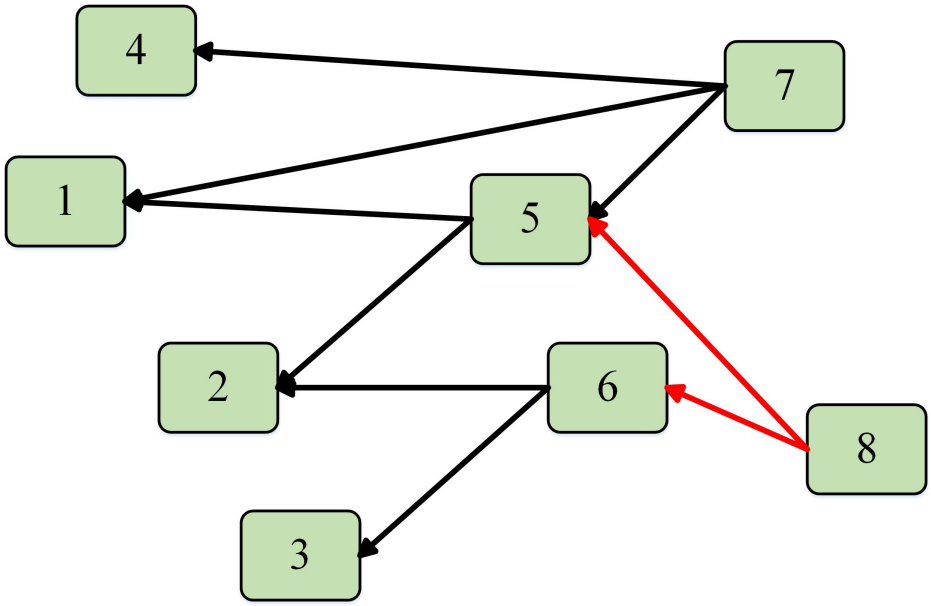

Fig. 3.2 Transaction 8 directly approves 5 and 6. It indirectly approves 1, 2 and 3. It does not approve 4 and 7

Fig 3.2 illustrates how transactions are approved in the IOTA tangle.

The possible advantages and reasons for incorporating the IOTA technology into the IoT infrastructure are as follows:

- **Scalability:** Scalable infrastructure is required by all of the IoT applications to handle the growth in the number of the IoT devices. The natural uniqueness of the tangle-based architecture in terms of decentralized consensus that ensures the participants also validate transactions in the network. Scalability is ensured within The IOTA tangle and does not have a scaling limitations.

- **Decentralisation:** Data exchange is validated and approved in centralised network architectures by central third-party authorities. In comparison to centralised server maintenance, this results in a more significant expense. Nodes exchange transactions with one another in the IOTA tangle-based architecture without depending on a centralised authority. Accordingly, all participants wishing to share transactions on the tangle must also take an active part in consensus processes.

- **Security and privacy:** Security and privacy remains a crucial challenge in the IoT domain. To make sure that data stays confidential and protected, IOTA developed a protocol called IOTA streams, which is a cryptographic framework that is designed as a secure message verification and protection protocol for transferring data over a given transport layer.

- **Zero-fees transactions:** As IOTA participants themselves perform the Proof of Work (PoW), IOTA does not involve miners. The transaction costs shall be considered the electricity sufficient to verify a working mechanism for two previous unconfirmed transactions. This implies that to sustain the tangle network, all participants of the network should use their power usage, thus eliminating transaction charges. Through the use of the tangle system, IOTA would operate free of charge, making the system much more distributed.

- **Energy-efficiency:** To be able to save power usage of the restricted IoT devices, IOTA allows the Proof of Work (PoW) to be moved to a devices with higher resources. IoT nodes have limited resources in terms of power consumption, therefore, IoT systems need to be built to optimise energy efficiency to enhance the life of the network and the devices. IOTA technology enables Proof of Work (PoW) to be outsourced to a more powerful device for efficacy with regard to energy efficiency and transaction throughput.

- **Resiliency:** The integrity of the data transmitted and analysed is required for IoT applications, hence the IoT infrastructure must be resilient to breakage and data leak (e.g. offline capacity). The IOTA network has replicas of the data kept by peers of IOTA. This helps maintain data integrity and provides extra flexibility for the IoT infrastructure in combination with the offline tangle capability.

Table 3.1 Node Types in IOTA Network

Node Type	Storage	Validation
Full Node	Full Tangle last Snapshot	Yes
Light Node	None	No
Hornet FullNode	Full Tangle	Yes
GoShimmer	Full Tangle	Yes
Wasp	Full Tangle	Yes
Chronicle Permanode	Full Tangle Permanently	Yes

The IOTA networks consist of interconnected nodes running the same node software. This software enables read and write access to the tangle, validation of transactions and storage of transactions in their local ledgers. Table 3.1 Describes the features of the participant's nodes of IOTA network.

3.3.2 Anatomy of IOTA Transaction

The IOTA version 1.0 was managed and enhanced by the IOTA Reference Implementation (IRI) network of nodes that implements the network of IOTA specifications and communicates via the JSON-REST HTTP1 interface [1]. A transaction is a functional unit of the IOTA tangle. The transaction performs local interactions between nodes in the network. The anatomy of the transaction is given in [145].

The transaction comprises of 11 elements as shown in Table 3.2. The field of a transaction object has a single purpose, apart from the signatureMessageFragment. This field may contain up to 2187 trytes, and includes either a user's digital signature for a value-based transaction or user-defined data for a zero-value transaction take place on the IOTA network. Therefore, the ability to effectively to store user-defined data in this field leaves the door open for the tangle to continue serving as a tamper-proof, unauthorised data repository. The IOTA transaction consists of 2673 trytes (if encoded). When decoding the trytes, there will be a transaction object with the following values:

[1] https://github.com/iotaledger/iri/releases/tag/v1.8.2-RELEASE

Table 3.2 IOTA Transaction Structure

Field	**Data types and values**
Hash	*String 81-trytes*
SignatureMessageFragment	*String 2187-trytes*
Address	*String 81-trytes*
value	*Int*
Timestamp	*Int*
CurrentIndex	*Int*
LastIndex	*Int*
Bundle	*String 81-trytes*
TrunkTransaction	String 81-trytes
BranchTransaction	String 81-trytes
Nonce	String

- Hash: is a *string* that consists of 81-trytes, which is unique hash of this transaction. IOTA replaced Curl-P-27 with a hash function based on Keccak-384, which is called Kerl. Keccak is a function that went on to become SHA-3. Kerl encodes the input bytes into ternary (0, 1 -> -1, 0, 1) before hashing.

- SignatureMessageFragment: is a *string* that consists of 2187-trytes signature message fragment. In case there is a spent input, the signature of the private key is stored here. If no signature is required, it is empty (all 9's) and can be used for storing the message value when making a transfer. IOTA uses the Winternitz one-time signature scheme (W-OTS) to generate digital signatures. This signature scheme is quantum robust, which means that signatures are resistant to attacks from quantum computers. W-OTS uses relatively small key and signature sizes. As a result, it is a one-time signature scheme, it can only be used to securely sign one message.

- Address: is a *string* that consists of 81-trytes address. In case this is an *output*, then this is the address of the recipient. In case it is an *input*, then it is the address of the input, which is used to send the tokens from (i.e., address generated from the private key).

- value: is an *int* value transferred in this transaction.
- Timestamp: is an *int* of the transaction. It is important to know that timestamps in IOTA are not enforced.
- CurrentIndex: is an *int* of the transaction and refers to the index of this transaction in the bundle.
- LastIndex: is an *int* that refers to the total number of transactions in this bundle.
- Bundle: Is a *string* that consist of 81-tryte bundle hash, which is used for grouping transactions of the bundle together. With the bundle hash you can identify transactions which were in the same bundle.
- TrunkTransaction: Is a *string* that consists of 81-trytes hash of the first transaction that was approved with this transaction.
- BranchTransaction: Is a *string* that consists of 81-trytes hash of the second transaction that was approved with this transaction.
- Nonce: Is a *string* that consists of 81-trytes hash. The nonce is required for the transaction to be accepted by the network. It is generated by doing Proof of Work (either in IRI via the attachToTangle API call, or with one of the libraries such as, ccurl or python).

3.3.3 IOTA Streams

The IOTA Streams is a cryptographic framework that is designed as a secure message verification and protection protocol for transferring data over a given transport layer [146]. The channels protocol is specifically developed as a replacement for the previously used Masked Authentication Messaging (MAM) [147] library for transferring data using the tangle

as the primary transportation mechanism. The channels in IOTA streams are structured in a number of different ways with any arbitrary combination of Publishers and Readers.

Streams Channels Protocol

There are several types of channels provided for the IOTA streams, which are necessarily for Authors and Subscribers to be produced in order to interact with the tangle. The channels are described as follows:

Authors: A channel called author is mainly concerned with producing a new channel along with the configuration of the structure of that particular channel. It can be either a single branch or a multibranch. The Author of a channel is potentially capable of setting the access restrictions to branches within a channel structure. In addition, the author of a channel accepts and manages user subscription messages.

Subscribers: A channel called subscriber is simply any user in a particular channel, but not the author. A subscriber is produced in an independent way without requiring verification by an author. To be able to write to a specific branch or to process a private stream, a subscription to the channel is required and the author will accept and process that subscription. Consequently, a subscriber can use a Pre-Shared keys as an alternative way of subscription to interact with a stream without the need to conduct a subscription process.

Branching: Branches are defined as a sequential that contains a set of messages, which is spawned and connected to the announcement message. These branches will typically be produced in two ways including: a signed packet or keyload message for public and private streaming respectively. A channel consists of two different types including: single branch and multi branch. The single branch is a linear sequencing of messages where every message connected to the previous one, while the multi branch is a sequence of messages that does not depend on sequential linking of messages. Now, when producing a channel, the author decides what the channel should use, which can be either a single branching or

multi-branching. Then, this will notify the Streams instance in which way to proceed in order to perform the sequencing. The subscribers will ultimately be notified as they process the root message (Announcement). Therefore, the instances know the sequencing order eventually.

Keyloads: A keyload message controls access rights, which enables the author to identify who will be able to decrypt any messages that are attached following it. There are two different ways in order to identify access when producing a keyload including: subscriber public keys and pre-shared keys. The subscriber public keys is adopted in the processing of subscription messages, where public keys are masked and giving to the author to be stored on their instance. That Author will have the ability to identify who can gain access the subsequent messages by including that public key in the keyload. The pre-shared keys are shared between users offline. consequently, these keys are used for granular access restrictions to a stream without requiring the subscription process.

3.3.4 IOTA Smart Contract

The concept of IOTA Smart Contract Protocol (ISCP) was originally coming from the Qubic project [148] and inherits most of its useful features. It is considered as a second layer protocol built on top of the core protocol and executed by GoShimmer nodes [149]. The protocol is developed in a way to be fully decoupled into a separate node called Wasp. Consequently, IOTA's smart contracts are run through the network of Wasp nodes, all of which are connected to the tangle. IOTA Smart Contract (SC) is an effective solution that shifts priorities towards a pragmatic solution. Fundamentally, IOTA SC are "Quorum-based Computations", which does not change from the original vision of the Qubic protocol. Fig 3.3 describes the structure of the IOTA smart contract.

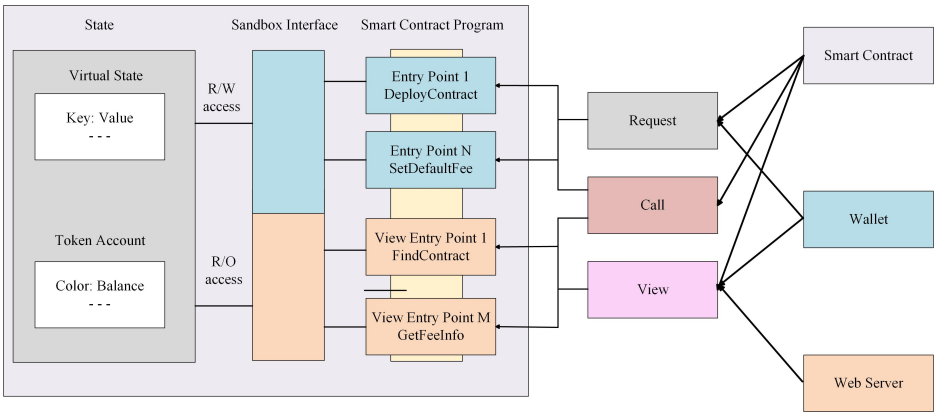

Fig. 3.3 Structure of IOTA Smart Contract

IOTA SC adopts the logic of UTXO Digital Assets (also known as Colored Coins). In order to create a smart contract, a new digital asset should be created and sent to the address of the SC. This transaction is essentially the Genesis transaction of the SC, which becomes the origin transaction (Genesis) of the SC. It does not require any additional fess for this process to be completed. This created digital asset remains at the address of the SC for the duration of the SC‘s term. Hence, the property of the SC.

In order to issue a request to the SCs, a digital asset is required to be created in this request transaction and sent to the address of the SC. Then, the request coin is usually sent to the SC and remains the property of the SC (at its address). This only temporary digital asset will be converted back to the original IOTA token when the request is processed (settled, confirmed) and remains in the SC. Since the original transaction usually includes an initialisation request to create a SC, it requires two IOTA tokens to create a SC.

IOTA SCs are defined as immutable state machines and works according to the following two steps: State Machine: Every Smart Contract contains a state that is connected to the tangle. The state consists of data, such as account balances, input conditions and consequences over time. Each state update represents a state transition on the tangle. The Smart Contract’s state and program code are both unchangeable due to the fact that they are stored on the tangle. The state can be updated incrementally by appending new transactions to the tangle. The

tangle supports a verifiable audit trail of state transitions. This creates confidence that the state transitions are valid and cannot be corrupted by malicious or faulty nodes.

A multi-chain environment has been integrated into IOTA SC and completely secured by the tangle (base layer 1): Subnets consists of Wasp nodes (committees), which is mainly designed to run many blockchains in parallel on it, without losing sight of the tangle environment that secures IOTA's digital assets. Each of these chains are a fundamental and functional equivalent of an Ethereum blockchain, which has the ability to host several Smart Contracts.

IOTA smart contracts does not require all nodes to participate in the network to execute all smart contracts as its designed to be more flexible. This meets the needs of the smart contract owner. Therefore, it will reduce cost and power while increasing flexibility.

3.3.5 Relationship between Coordicide and Coordinator

The IOTA network currently relies on a coordinator, which has been implemented and operated by the IOTA foundation as a third parity to manage transactions confirmation. The purpose of the coordinator is to prevent attacks such as parasite chains [150]. The coordinator is responsible for issuing milestone transaction every two minutes to validate the transactions. Each transaction attached to the IOTA tangle has its own parameter values [150] that the coordinator uses to determine its path.

In order for the IOTA network to be fully decentralised, Coordicide has been introduced [151], which is a proposal for the removal of the coordinator. When this happens, nodes will be able to reach a consensus without milestones, making IOTA networks decentralised. Coordicide is focused on the removal of the coordinator through the implementation of several network components. At a high level it consists of several modules to achieve the vision including: node accountability, auto-peering, node discovery, rate control, consensus, voting and tip-selection. Coordicide anticipates that all honest participants of the network

would agree on which transactions should be considered valid [151]. It is important to remark that one should not be afraid of the probabilistic nature of it if something occurs with strictly positive probability, this doesn't yet mean it would ever occur in practice. Another important idea is that, while there is no need for total consensus on what is really important (transactions' validity), the total consensus on everything is not required. Therefore, it may use an approximate consensus to achieve the total one with high probability [151].

3.3.6 Auto-Peering

In IOTA network, a node is the machine that holds all of the information about the tangle. In order for the network to work efficiently, nodes exchange information with each other to be kept up-to-date about the new ledger state. In the current IOTA implementation, a manual peering process is adopted for nodes to mutually register as neighbors. This manual peering leads to attacks, which affects the network topology. In order to prevent these attacks and to make the setup process of new nodes simpler, an auto-peering mechanism has been introduced. The mechanism enables nodes to select their neighbors automatically. The process of enabling nodes to select their neighbors should not require manual intervention by the node operator. Hence, it is called auto-peering. Fig 3.4 shows the IOTA auto-peering mechanism.

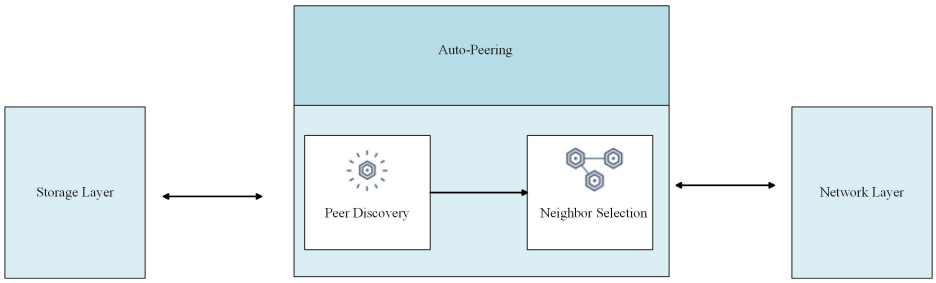

Fig. 3.4 IOTA Auto-peering mechanism

The auto-peering aims to achieve two important goals. First, it creates an infrastructure where new nodes can easily join the network; second, it ensures that an attacker cannot

target specific nodes during the peering process, i.e., it ensures the network to be secure against eclipse attacks. The auto-peering works according to five phases including: peer discovery, neighbor selection, network reorganisation, eclipse protection, choosing salts and sybil protection all of which has a specific functionality. The interested reader is referred to the auto-peering section for full details about each phase [151].

3.3.7 Snapshotting

Since there is a huge amount of transactions and data is exchanged among IOTA participants nodes, the network will ultimately grow big, in particular for zero value data transaction. In order to keep IOTA in accordance with the requirements of embedded devices, which will permeate and constitute the Internet of Things. It applies the concept of snapshotting. Snapshotting is defined as a method that reduces the size of the tangle database by eliminating all transactions from the tangle [96]. It leaves only a record of address with corresponding balances. Simply speaking a snapshot is a list of every address with corresponding non zero balance. Snapshotting is similar to Blockchain pruning, except snapshotting has the significant advantage of grouping several transfers to the same address into 1 record, which leads to a smaller storage requirement [96].

3.4 Integration of IOTA and IoT

Centralised cloud models have made a considerable contributions in the growth of IoT regarding data processing and storage, but in data transparency, there is an inherent need of trust and a lack of absolute confidence. Centralised cloud models are considered much like a black box for IoT applications and IoT users do not have control of the data since they are stored in a third party. Furthermore, centralised cloud models are vulnerable to faults. In the evolution of IoT, the fog computing paradigm, which works at the edge

of the network is getting more functionality since data is processed near the IoT devices as compared to the cloud [8]. Therefore. The IoT can benefit from the decentralised network architecture offered by IOTA tangle, where further deployments of IoT applications will continue while eliminating the need for trust in centralised cloud models. However, IOTA tangle is still in its early stages of research and development, and there are still several research challenges towards integrating IoT and IOTA tangle in a seamless manner. Achieving absolute decentralisation, scalability, energy-efficiency, security and privacy in the IoT using the IOTA tangle is needed, considering the differences and heterogeneity in various devices involved in the IoT domain. Most of the IoT devices have limitations in terms of energy consumption, limited processing and limited storage and incapable of holding a complete copy of the tangle.

3.5 Application Scenarios

In this part, we describe several application scenarios. The application scenarios provided in this part are on 1) Smart Parking, 2) Smart Campus, and 3) Self-Driving Cars. These generic application scenarios were modified with minor alterations to support the use of IOTA platform in an efficient manner. These scenarios are referred to throughout this chapter and have applied them to describe the functionality of IOTA protocol and to demonstrate how IOTA protocol can be used in several IoT applications.

3.5.1 Smart Parking

Smart Parking is a crucial component of smart cities with the primary goal of finding, allocating, reserving, and providing parking spaces for individual vehicle drivers in a particular area. It provides vehicle drivers with the ability to find available spaces in congested areas in the city. This smart parking system is considered to be based on real world projects that have

been implemented in several cities in Europe and the United Kingdom, particularly in London and Cardiff [152]. According to [153] existing smart parking solutions are complicated and transdisciplinary. However, smart parking enables drivers to efficiently find parking spaces via information and communications technology [153]. It presents the opportunity for smart cities to take on an effort to efficiently optimize the use of their parking resources. In this scenario, the system deploys smart sensors around the parking area for the purpose of monitoring and reporting occupancy and for data processing to obtain useful insights from the gathered sensor data. To be specific,

Fig 3.5 describes a smart parking system, in which sensors are deployed to gather information. The procedure requires the end user to perform several steps to be able to reserve a parking slot. These steps are described as follows 1) the use of a mobile application to search for a parking space near the required destination 2) select the zone and navigate through the parking slots to check for availability in that selected zone 3) finally pay the charges in the preferred way. To develop an intelligent parking system that takes into consideration energy efficiency, including power consumption and scalability, the adoption of an IOTA platform can help to perform tasks more efficiently.

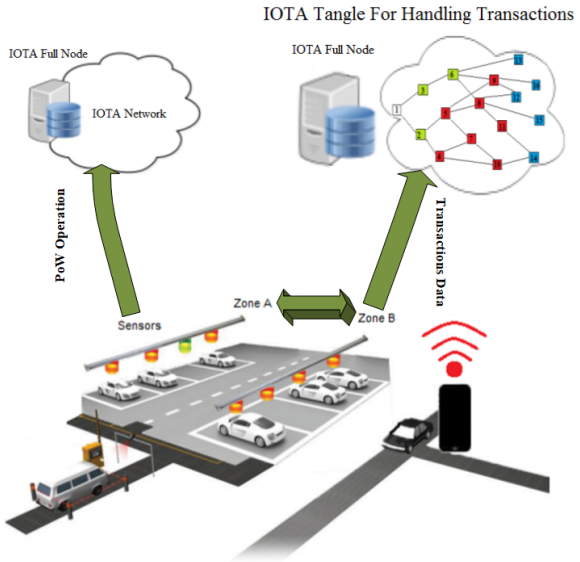

Fig. 3.5 A Smart Parking Application.

3.5.2 Smart Campus

Smart campus extracts, applies the same principles and operation of the smart city to the operation of the campus. It enables the connectivity between students, staff and their surrounding environments. It has been adopted by many universities worldwide. Smart Campus requires universities to use advanced technologies, e.g., smart cards and sensing technology, in order to be able to control and monitor various facilities on campus automatically. The benefits of building a smart campus are to significantly cut down operational costs, automate the process of maintenance, reduce energy consumption, and improve the learning experience[154]. The smart campus is usually based on a three-layer architecture: layer of physical things, layer of network and layer of application. The layer of physical things requires the deployment of sensors to gather data in regards to the monitored environment. In the network layer, data is fused and aggregated, while, the application layer, uses cloud systems to store and retrieve data.

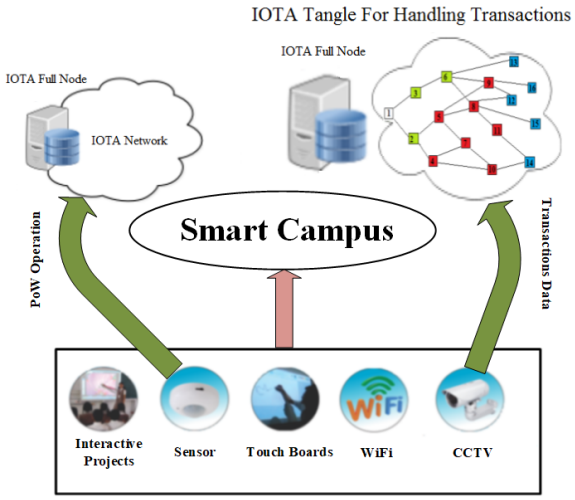

Fig. 3.6 A Smart Campus Application.

Fig 3.6 illustrates and provides the building blocks of the smart campus domain. In order to make smart campus energy efficient and scalable, it's recommended to use the IOTA platform. This is due to the fact that it can store data in the tangle in an efficient way

and accommodate the scalability growth of the domain. Furthermore, a reward system can be built within the smart campus and tokens could be given to students who study at the library. These tokens can then be used to make a discount for students on their tuition fees. Consequently, motivating students to study as well as keeping the campus sustainable.

3.5.3 Self-Driving Vehicles

A self-driving vehicle is an emerging industry and advanced technological development in the field of the automobile with several solutions already adopted by many companies such as Google and IBM. A self-driving vehicle refers to the capability of any vehicle with features, which enables it to accelerate, brake and steer with limited or no driver interaction. Self-driving cars will be part of smart cities and require the improvement of the city infrastructure to support and protect them. This self-driving case study is based on a real world project as described in [155]. In this scenario, the component of the architecture is logically partitioned into three tier and these include CARMA vehicle, CARMA edge, and CARMA core, each of which has its own functionality. CARMA vehicle is responsible for connecting all the sensors and other equipment's. CARMA edge processes the information gathered from the vehicle. It allows collaboration with roadside and is capable of providing further computation. CARMA core is the cloud systems that are responsible for storing information coming from multiple vehicles and other supporting services.

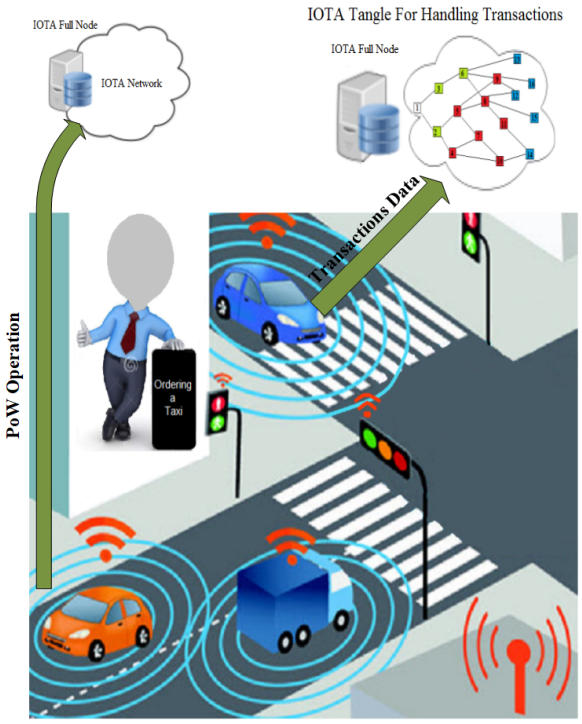

Fig. 3.7 A Self-Driving Vehicles Application.

Fig 3.7, shows an example of self-driving vehicles, e.g., taxi equipped with sensors and a person on the road ordering a taxi through his/her mobile application. To support an energy-efficient and scalable infrastructure for such application scenario. The IOTA platform brings benefits by efficiently enabling secure communications and information sharing e.g., in the form of transactions among vehicles. It allows the payment to be made through the crypto-wallet. Further, vehicles are able to gather sensor and actuator data then store them securely and efficiently in the IOTA tangle. Consequently, data would be offered in virtual market places to, e.g., other traffic participants, researchers or city planners.

3.6 Lessons Learned

In this part, we take into account the main characteristics of IOTA described throughout Section 3.3 and discuss how they would help to build safe, scalable and energy-efficient

sensing infrastructures for numerous IoT applications, e.g., smart parking. Creating smart living spaces for people is a result of high demand and competition. The government and industry sectors have been pushed to lunch smart city programs. Investments in smart cities aim to find ICT-based sustainable solutions to rising challenges [156].

Smart parking is an important aspect of smart cities because it offers opportunities for vehicle drivers to find available parking in congested areas of the city. Smart Parking makes it possible to park more vehicles with a limited number of resources. Smart parking should also ensure that limited resources are not exhausted [157]. The best approach is to use the most efficient and optimal use of resources. In this respect, the components and mechanisms of a smart car park and how these applications handle and consume resources (such as energy consumption, practices, payments through crypto wallet, etc.) must be understood. In the context of IoT, all of the information needed to understand smart parking and it's parking slots are concealed in IoT data. It requires collecting, aggregating and analysing them efficiently on a large scale to obtain knowledge and useful information from IoT data. The scalable sensing network is a crucial component in the processing and analysis of IoT data.

– It is important to ensure that data analysis is applied in the right time and the right place i.e., distributed intelligence. The process of data in the smart parking system moves from sensor to the cloud infrastructure. It is always recommended to process data in the network before sending it to the cloud. This ensures a decrease in transferring data to the cloud, which reduces storage costs and computational costs. Large amount of data is aggregated and generates only a summarised results. The summarised results indicates that less data is to be sent to the cloud, leading to lower costs in energy consumption and bandwidth.

– Communication protocols are beyond the scope of this chapter. However, communication protocols are required in establishing the IoT infrastructure. This requires a careful consideration in using the suitable protocol such as MQTT. All protocols are

characterised by their own advantages and disadvantages. Most of the protocols are designed to reduce the overall energy consumption. In addition, IoT gateways can apply the needed communication protocol at run time to further reduce IoT resources. For example, the authors in [28, 8] shows that a lot of energy is reduced by utilising the MQTT rather than HTTP. Therefore, which protocol should be used depends on the application deployed. The interested reader is referred to the surveys in [28, 8] to determine which communication protocol is suitable for which scenario.

– IOTA is considered as an important part in decentralising the network. IOTA is gaining a particular attention to ensure scalability and privacy of IoT systems. In IOTA there is no need for centralised repositories to allow for IoT services. Moreover, it can potentially solve the scalability and energy-efficiency in IoT systems [29].

– According to [158, 159], context-awareness plays a critical role in decreasing energy consumption in the described application scenarios. For example, It can decide (1) when nodes are required to sense and when to go offline, i.e., event-driven (2) when to apply the required protocol, (3) when to transfer data, and (4) when to lower or speed rates of sampling i.e., time-driven.

3.7 Summary

This chapter presented a solid background of IOTA tangle and the working principle of the tangle; it also described the benefits of integrating the IOTA tangle in the IoT domain. It described several application scenarios showing the benefits of IOTA tangle to IoT applications to achieve better scalability and energy-efficiency. In addition, it described the lessons learned in order to build a scalable and energy-efficient sensing infrastructures for numerous IoT applications, e.g., smart parking.

Chapter 4

A Distributed Intelligence Framework for the IoT with IOTA and Mobile Agents

Several studies have demonstrated the benefits of using distributed intelligence (DI) to overcome these challenges. In this chapter, a Mobile-Agent Distributed Intelligence Tangle-Based approach (MADIT) is proposed as a potential solution based on IOTA (Tangle), where Tangle is a distributed ledger platform that enables scalable and transaction-based data exchange in large P2P networks. MADIT enables distributed intelligence at two levels. First, multiple mobile agents are employed to cater for node level communications and collect transactions data at a low level. Second, high level intelligence uses a Tangle based architecture to handle transactions. The Proof-of-Work offloading computation mechanism improves efficiency and speed of processing, while saving energy consumption. Extensive experiments show that transaction processing speed is improved by using mobile agents, thereby providing increased scalability. The research presented in this chapter was published in [6, 160].

4.1 Introduction

IoT applications connect everyday objects to the Internet and enable the gathering and exchange of data to increase the overall efficiency of a common objective [1]. It is estimated that there will be approximately 125 billion devices connected to the Internet in 2030 [2, 161, 162]. Consequently, most IoT applications are required to be highly scalable and energy efficient, so that they are capable of dynamically responding to a growing number of IoT devices [163].

Distributed Intelligence (DI) is an approach that could address the challenges presented by the proliferation of IoT applications. DI is a sub-discipline of artificial intelligence that distributes processing, enabling collaboration between smart objects and mediating communications, thus supporting IoT system optimisation and the achievement of goals [164].

The local interactions among IoT devices will be finally attached to the IOTA tangle. We propose an integration of the IOTA tangle [136] and mobile agents [165] techniques, in order to realise a scalable DI approach by providing low-level and high-level intelligence. Functionalities are distributed to both low-level and high-level layers. The proposed Mobile-Agent Distributed Intelligence Tangle-Based approach (MADIT) specifically recognises resource-constrained devices, which might not be able to perform the required computation at low-level. High-level computation is performed by more advanced computational devices.

The work in this chapter outlines the design of a scalable system that can be used in various IoT applications. The IOTA tangle is used to achieve scalability and a higher level node is responsible for performing the Proof of Work (PoW) for efficacy with regard to energy efficiency and transaction throughput [163].

IOTA technology focuses on Machine-2-Machine (M2M) transactions. Due to its high scalability and zero transaction fees, IOTA's network facilitates data exchange with IoT-connected devices worldwide. The different types of nodes, which are the core of the IOTA network, such as the full node and the light client nodes are suitable for resource constrained

IoT devices. The light client can run on IoT devices with limited resources e.g., power and memory, while the full node can run on devices with higher resources. This justifies why IOTA technology was chosen over other technologies.

4.2 SDIT: Scalable Distributed Intelligence Tangle-Based Approach

This section presents the proposed Scalable Distributed Intelligence Tangle-based approach (SDIT) that aims at tackling the scalability, energy-efficiency and decentralisation by adopting the IOTA tangle technology.

4.2.1 SDIT: System Architecture

Figure 4.1 illustrates an abstract view picture of the proposed system architecture. The architecture is divided into three main parts including: IoT devices, Tangle to process transactions(txs), and PoW enabled server. Each IoT device is connected with neighbouring nodes via TCP/IP protocols for communication, and interaction with the tangle is in the form of transactions. The tangle is responsible for managing, collecting and processing the transactions. A PoW-enabled server has rich resources and is mostly responsible for performing all of the computations on behalf of the IoT devices. This is a critical task for efficacy with regard to energy efficiency and transaction throughput. The tangle can act as a data management layer for processing and storing data in an efficient way.

The green boxes in Figure 4.1 represent fully confirmed transactions, which means that they are approved by all of the current tips, whereas the red boxes are not confirmed transactions. The blue boxes are the tips.

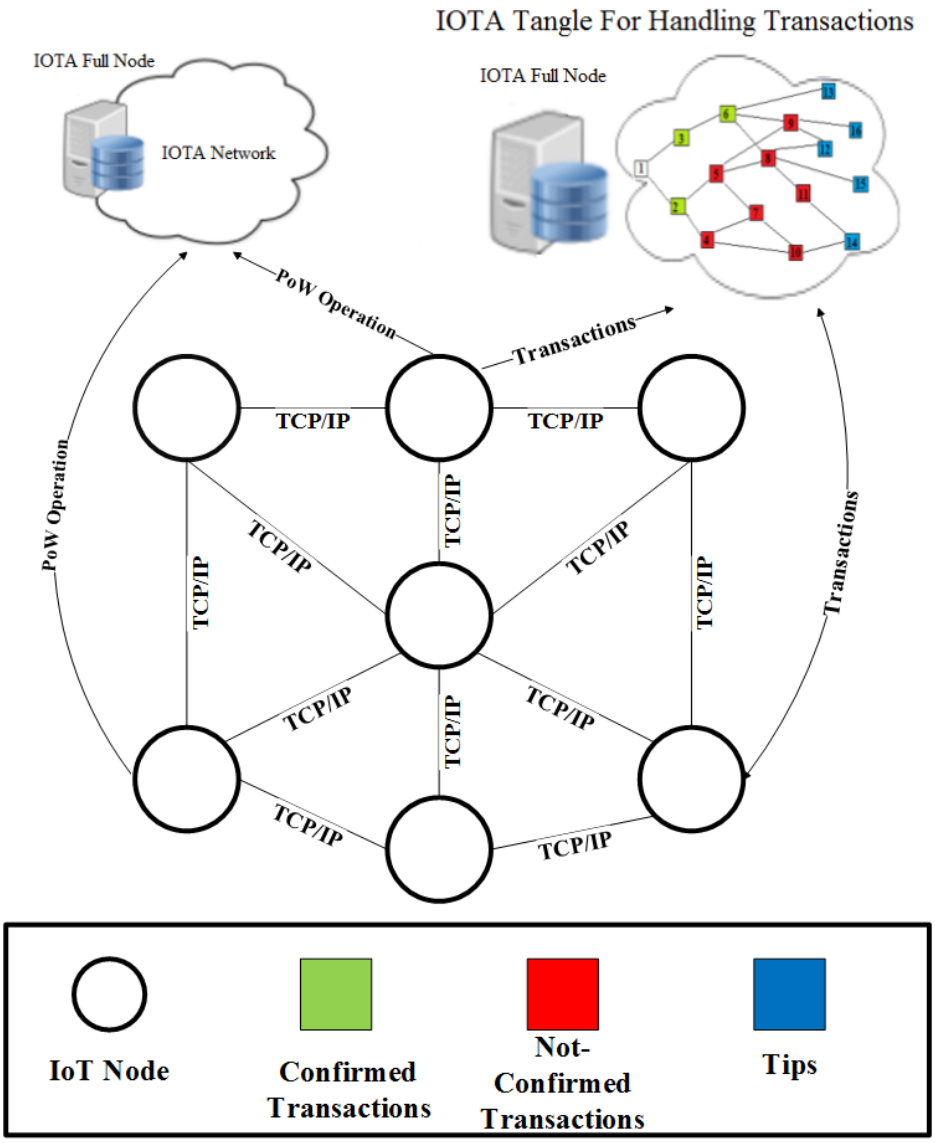

Fig. 4.1 The Scalable Distributed Intelligence Tangle-based approach (SDIT)

4.2.2 Consensus Mechanism Employed

Since we are utilising the IOTA tangle to deal with transactions. The same working principals is followed in which a new transaction should choose two previous unapproved transactions, which are called Tips, to approve based on the tip selection algorithm. After tips are selected, the IOTA nodes are able to publish their new transactions to the Tangle. In the advanced Markov Chain Monte Carlo Tip Selection Algorithm (MCMC) N independent random walks

are generated on the *tangle*; the walks begin at the genesis or at a random node and keep moving along the edges of the tangle based on a probability function.

The MCMC Algorithm ensures that the tips are selected non-deterministically along the path of the largest cumulative weight for a reasonable amount of time. The probability from transaction walking from the genesis L_x to a tip L_y is proportional to $P(-\alpha(L_x - L_y))$, where P_{xy} is an increasing function (generally an exponential), α is a constant and c_i represents the Cumulative Weight of transaction i. The process ends when the walker reaches a tip, which is then selected for approval.

Typically, the first tip is usually selected for approval. For further details on the working mechanism of the MCMC algorithm, the interested reader is referred to the IOTA white paper in [136]. To support the advanced tip selection process, the MCMC technique [136], applies a set of rules for deciding the probability of each step in a random walk, and works as follows:

1. Run the MCMC algorithm N times to choose 100 new transactions (tips). The probability of the transaction being accepted is therefore M / N (M is the number of times a tip is reached that has a direct path to the transaction);

2. Calculate how many tips that are directly or indirectly connected to a particular transaction and decide with what probability transactions will be accepted as follows:

   (a) if it is less than 50%, the transaction is not approved as yet (not confirmed);

   (b) if it is above 50%, the transaction has a fair chance to be approved (awaiting to be confirmed);

   (c) if it reaches the level of 98% or 100%, the transaction is considered approved (fully confirmed).

4.2.3 Proof of Work Offloading

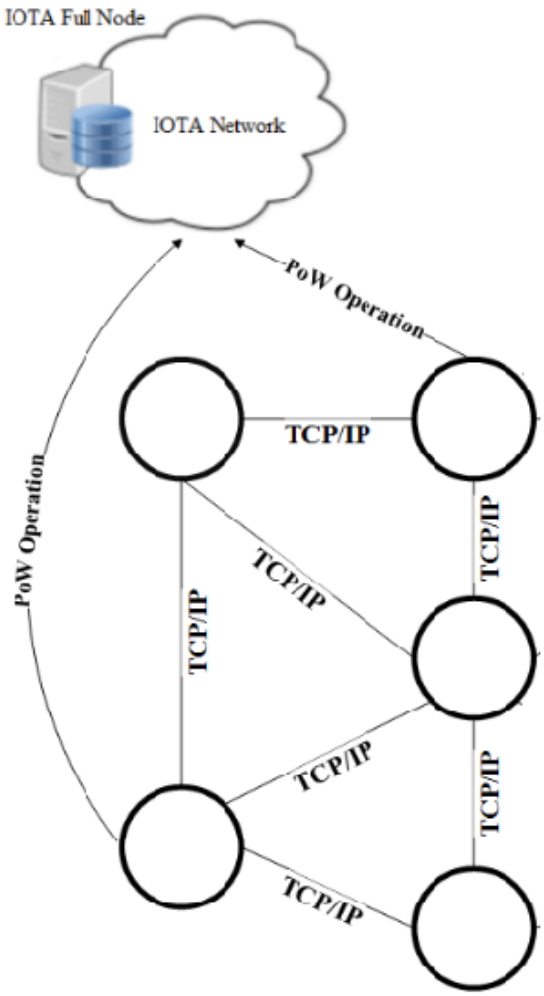

Fig. 4.2 Computation offloading in SDIT approach

Offloading can be divided into two categories including: data offloading and computation offloading. The former refers to the use of novel network techniques to transmit mobile data originally planned for transport via cellular networks. The latter refers to the offloading of heavy computation tasks to conserve resources [166]. It is commonly assumed that the implementation of computation offloading depends heavily on the design of the network architecture. The main goal of offloading is to save energy consumption or overall task execution time, or both of them. It was suggested by the authors in [32] to conserve energy

of IoT devices by performing the proof-of-work operation on a device with rich resources, thus achieving improved energy-efficiency.

Figure 4.2 illustrates the computation offloading mechanism used in the SDIT approach in which the computation operation of performing the PoW is offloaded to a device with higher resources. This saves energy consumption of constrained IoT devices.

In particular, we achieve scalability and decentralisation by adapting the IOTA tangle and their consensus mechanism. The proposed approach is presented in view of the architecture, consensus mechanism and the computation offloading technique employed.

4.3 MADIT: Mobile-Agent Distributed Intelligence Tangle-Based approach

4.3.1 MADIT: System Architecture

The envisioned framework, Mobile-Agent Distributed Intelligence Tangle-Based approach (MADIT), represents the novel contribution of the work and is depicted in Fig. 4.3. One of the key contributions of this work is the attempt to establish a baseline for a reference framework for Tangle-based MADIT that can be used to support various IoT applications.

The framework is divided into four main parts: (1) IoT devices; (2) Tangle to process transactions(txs); (3) PoW enabled server, and; (4) Mobile Agent to carry a list of transactions data. Each IoT device is connected with neighbouring nodes via TCP/IP protocols for communication, and interactions with the Tangle are in the form of transactions. IoT devices are responsible for collecting data from the environment. The Tangle is responsible for managing and processing the transactions. A PoW-enabled server is an IoT device that has rich resources and is responsible for performing costly computations on behalf of IoT devices. Mobile Agents are responsible for transporting a list of transactions when visiting nodes on

their routes. This is an important task that supports inter-node communications. The Tangle can act as a data management layer for processing and storing data in an efficient way.

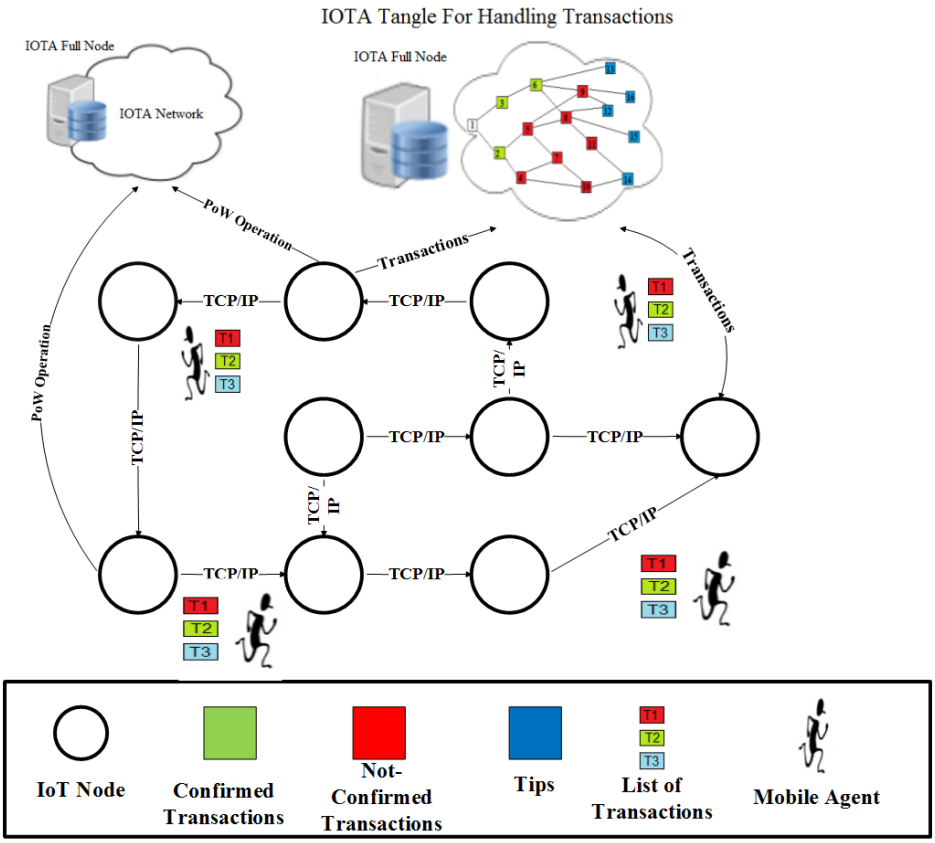

Fig. 4.3 The Mobile Agent Distributed Intelligence Tangle-based Approach (MADIT)

4.3.2 Algorithms Design

An algorithm 2 that efficiently partitions the network into several groups, which can aid in the mobile agent itinerary planning is proposed in order to improve energy-efficiency and scalability. This algorithm leverages three key factors: identifying a set of routes covering all nodes in the network, comparing the routes to select the route with the least number of nodes and source nodes grouping. The set of routes are generated to identify the shared nodes and private nodes. The shared nodes are used for grouping purposes, while private nodes are considered as nodes the belongs to a particular route. The routes are compared based on

the number of nodes in order to select the route that has a smaller number of nodes. Then, the source grouping technique takes place to generate several groups based on the identified shared nodes.

Another algorithm 3 that is used for dispatching mobile agents to collect data is proposed. The algorithm intelligently schedules mobile agents to visit a set of nodes in a particular group. This algorithm leverages three key factors: visiting a specific group, one mobile agent to each group and threshold value. The mobile agents are dispatched to a specific group to collect data. This algorithm ensures that there are not two agents visiting the same group. Since the mobile agent size increases when visiting nodes in the network, the algorithm initializes a threshold value to ensure that the mobile agent data buffer is not overloaded.

4.3.3 Mobile Agent Transactions for Local Interactions

The framework employs multiple mobile agents to avoid delays in reporting transaction data and to support local interactions (i.e., low-level intelligence). The framework considers that nodes in close proximity of each other will most likely generate similar data; therefore applying data aggregation techniques to eliminate redundancy is required. A data aggregation technique similar to [167, 168] is used to calculate the size of transaction data accumulated by the MA. Transaction data results are fused with an aggregation ratio ($\rho, 0 \leqslant \rho \leqslant 1$). Consider L^i_{ma} to be the amount of accumulated transactions data result after the MA finishes from source i, where A_i is the amount of transactions data to be aggregated by ρ, which is the fusion factor. Then:

$$L^i_{ma} = A_i$$

$$L^2 = A_1 + (1-\rho) \times A_2 \tag{4.1}$$

$$L^i = L^{i-1} + (1-\rho) \times A_i \tag{4.2}$$

$$L^i = A_1 + \sum_{g=2}^{i} (1-\rho) \times A_g \tag{4.3}$$

In equation (4.3) there will be no data aggregation in the first node and the value of p depends upon the type of deployed application.

The packet message format of the proposed MADIT is described in Fig. 4.4. The pair of *Itinerary Planning* and *List of transactions* are the payload of the agents. *Dispatcher ID* is used to identify the root node that creates and dispatches MA. *FirstNode*, denotes the first node that the MA will visit. *Static Routes*, denotes the computed routes for MAs with all of the assigned nodes to be visited. *ToVisitFlag*, is set to indicate that whether the node has been visited by an agent or not.

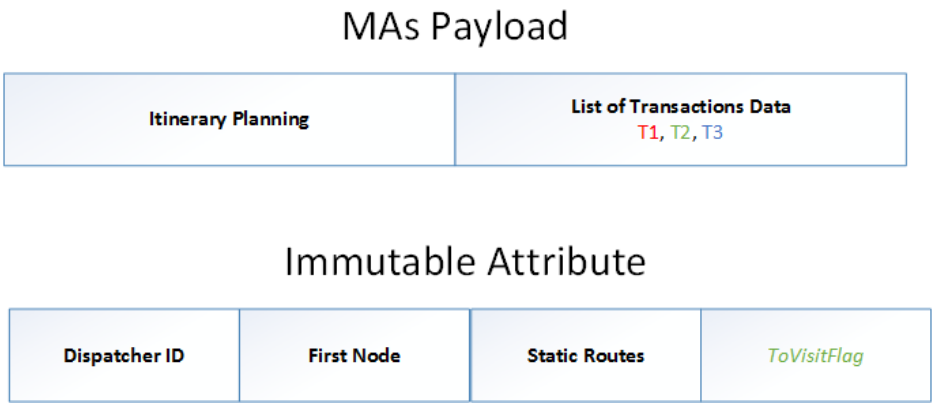

Fig. 4.4 Message format of the proposed (MADIT) approach

The reason for applying mobile agents in the work is not just to support low-level intelligence. It was stated in [169] that one of the most power hungry operations is radio communication; therefore, dispatching agents to collect transactions data rather than sending it is more efficient.

Algorithm 1 presents the pseudocode of establishing a random DAG G. Algorithm 2 presents the pseudocode of computing the routes and generating groups for all mobile agents.

Algorithm 3 presents the pseudocode of dispatching multi-mobile agents to start collecting transactions data.

Initially, the establishment of a random Directed Acyclic Graph (DAG) is proposed as described in (Algorithm 1), which is designed to build a graph with a random number of Nodes and Edges. The algorithm iterates to add the required number of nodes *nodeNum*. Then, it performs a check to ensure that the graph *G* is a directed acyclic graph. Then, the algorithm applies the Depth-first search (DFS) technique to ensure that all nodes can be traversed back to the root.

The Algorithm source grouping of sensor nodes (Algorithm 2) uses *G* Algorithm 1 as input and is designed specifically for making a group of source nodes for all mobile agents. It checks If the node has at least one connection or more in-degree connection and if the node has two or more connection out-degree connection. Hence, this is a shared node. It plays an important role within the network because it has multiple routes and is considered as a building block in generating groups. In addition, it acts as a main hub for connecting the nodes within the DAG network.

The algorithm finds a set of routes covering all nodes in the network by identifying the roots and the leaves. Then, it finds all possible routes between leaves and roots. For finding the least route, the source grouping algorithm sorts out all routes and compares them to identify the route with the least number of nodes. We define a shared node as the node that belongs to multiple routes. This means shared nodes can be reached by multiple routes. Shared nodes among different routes will be allocated as follows:(1) Initially each route is assigned a group of nodes, which belong to each particular route only (or we can call these nodes as private nodes on the route); (2) A shared node will be allocated to the group currently with the least number of nodes among the associated routes. Each generated group defines a set of nodes for the dispatched MAs to collect data from.

Algorithm 1: Generate a random directed acylic graph G

Input: $nodeNum, edgeNum$
Output: G

Initialize G to a directed acylic graph (DAG) with $nodeNum$ nodes but without any edges, and nodes range from 0 to $nodeNum - 1$
while $edgeNum \geq 0$ **do**
    $node_a \leftarrow$ randint(0, $nodeNum$)
    $node_b \leftarrow node_a$
    while $node_b == node_a$ **do**
        $node_b \leftarrow$ randint(0, $nodeNum$)
        Add edge($node_a$, $node_b$) to G
        if *G is still DAG* **then**
            $edgeNum \leftarrow edgeNum - 1$
        else
            Remove edge($node_a, node_b$) from G
    Get DFS post-route (G)
    Return DFS post-order
Return G

Algorithm 2: Source grouping of sensor nodes

Input: DAG G
Output: List of routes, Groups of source nodes

if *in-degree connection of node* $>= 1$ *and out-degree connection of node* $>= 2$ **then**
    Shared Node in G
for *root in Roots in DAG* **do**
    for *leaf in Leaves in DAG* **do**
        for *route in nx.all simple paths (Dag, root, leaf)* **do**
            Find all routes in G
for *each shared node routes list in DAG* **do**
    if *route* $>= 2$ **then**
        Select the route with the least number of nodes
for *each selected least route* **do**
    Add all least route nodes into a group.
Return list of routes, Groups of source nodes

Algorithm 3: Dispatch a mobile agent *MA* to collect transactions

Input: DAG G, list of routes R, Groups of source nodes gs
Output: Transactions T collected by *MA*

```
1  Initialize T as an empty set of transactions collected by MA
2  while MA has not completed the allocated tasks do
3      Move to visit the next node n according to the given route r
4      if n has been visited by any other mobile agent then
5          Repeat Step 3, until all nodes in r have been visited
6      if all nodes in r have been visited then
7          MA completes the allocated tasks
8      else
9          Dispatch MA to visit node n
10         Collect transactions T' (not exceeding limitation d in total) from node n
11         Add transactions in T' to T
12         Set visited flag of node n to true
13     if T contains d transactions then
14         MA completes the allocated tasks and publish the transaction to the ledger
15 Return T
```

The algorithm that dispatches mobile agents is described in Algorithm 3. It begins with input (1) DAG G, List of routes R $r \in R$ and a groups of source nodes gs, given through algorithm 2. Then, it initializes T as an empty set of transactions collected by *MA*. It starts dispatching mobile agents on a particular group in gs and ensures that no two agents will follow the same route. During the trip, each MA will visit nodes according to the given group gs. It will first check whether the current visiting node has been visited by any of the mobile agents or not. If the flag $visited$ of the node is $true$, the MA will move on to visit the next node on the route. Otherwise, if the current node is not visited during the same mission, the MA collects transactions data up to its data load d, and sets the flag $visited$ of the node as $true$. The MA completes the allocated tasks and returns either when all nodes on the given route have been visited, or when the MA has collected d transactions on the trip. The data load d threshold for each agent ensures that the agent buffer is not overloaded with transactions data during one single trip.

4.3.4 Proof of Work Offloading

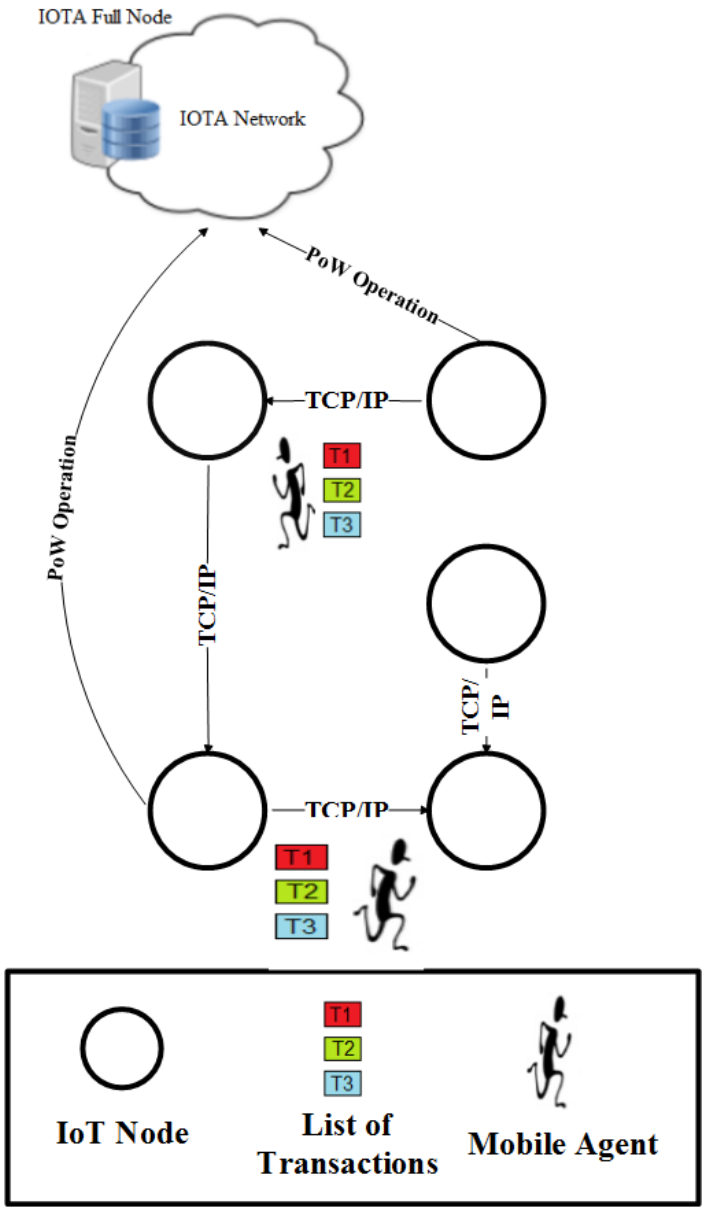

Fig. 4.5 Computation Offloading in MADIT Approach.

Fig. 4.5 illustrates the computation offloading mechanism used in the MADIT approach. It shows that the Proof of Work (PoW) operation is outsourced to an IOTA full node with unlimited power.

In particular, the framework addresses the problem of scalability and decentralisation without loss of efficiency by adapting and integrating the IOTA tangle and mobile agents. the work describes the proposed framework in view of the architecture, the consensus mechanism, the role of mobile agents and the Proof of Work computation offloading technique employed.

4.4 Experimental Results, Evaluation and Analysis

In this section, a description of the implementation of the experimental results, followed by an evaluation of the proposed solution in regards to the scalability, throughput and decentralisation are provided. Then, an analysis and discussion of the results obtained is also provided, to highlight the useful characteristics of IOTA tangle for IoT.

4.4.1 Environment Setup

The latest release of IOTA Reference Implementation (IRI 1.8.1) is deployed, which is the official Java build embodying the IOTA network specifications, [1] on the DigitalOcean cloud platform[2], and another IOTA Reference Implementation (IRI 1.8.1) on a local server dedicated for performing the operation of the Proof of Work (PoW).

The functionality related to IOTA addresses, transactions, broadcasting, routing, and multi-signatures have been implemented using iota.lib.py [170], the official Python library of IOTA Distributed ledger. In total, a large number of nodes with the specifications of medium size virtual machines (4GB RAM, 2 VCPU and 60.0GB Disk) are used to create the network. A medium size nodes and nodes with rich resources are used because this is more representative of real-world IoT scenarios. Nodes with rich resources enhance the performance by reducing the time it takes to perform the PoW.

[1] https://github.com/iotaledger/iri/releases/tag/v1.8.1-RELEASE
[2] https://www.digitalocean.com

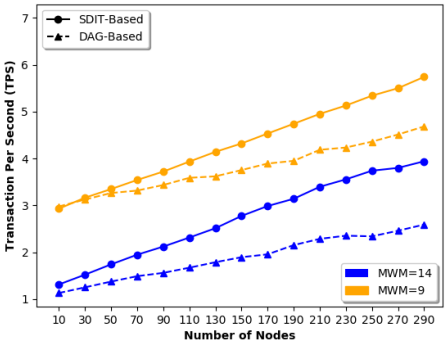

Fig. 4.6 Scalability in Tangle with 290 Nodes

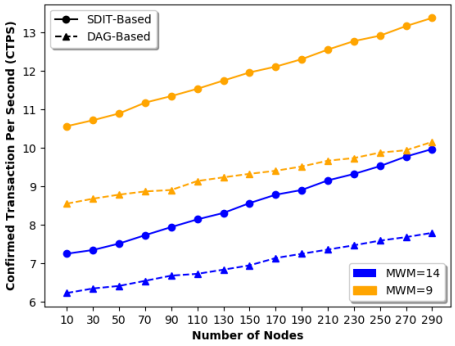

Fig. 4.7 Scalability in Tangle with 290 Nodes

In order to measure transaction speed and scalability, the sending nodes are configured to initiate a fixed number of transactions = 5. The experiments uses a set of different Minimum Weight Magnitude (MWM) (9,11,13,14). These transactions are broadcasted among all nodes through TCP/IP.

In order to validate the scalability, the Transaction Per Second (TPS) and Confirmed Transaction Per Second (CTPS) are tested under different numbers of nodes (50, 100, 150, up to 290[3]) with different Minimum Weight Magnitude (MWM) configurations as presented above, as shown in Figure 4.6, and 4.7 respectively. TPS is defined as the number of transactions published to the network per second and CTPS is defined as the number of transactions that move from pending to confirmed per second. The results obtained is based on a real deployment of IOTA nodes.

4.4.2 Results and Analysis

In this part, the performance of the scalability and throughput is presented, which was evaluated over several runs to obtain accurate results. Then, it was compared against one of the recent approaches in the literature, namely, DAG-based smart communities [131]. Their publications gave their full specifications, making it possible for researchers to implement

[3]Due to resource constraints, the experiments considers only up to 290 nodes.

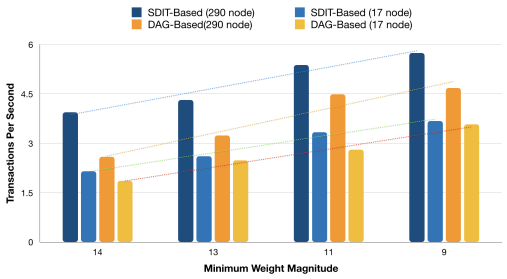

Fig. 4.8 Performance of TPS under different MWM

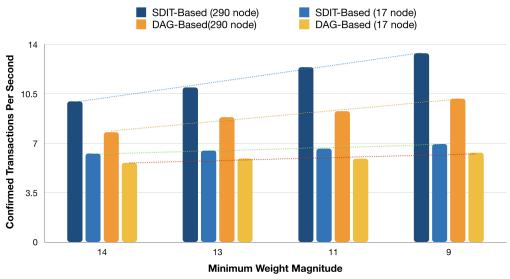

Fig. 4.9 Performance of CTPS under different MWM

and reproduce the published results. Finally, they achieved promising results for smart homes and are planning to extend their work with further comparisons.

Scalability: The results can be seen from Fig. 4.6 to Fig. 4.9. As shown in Fig. 4.6 and Fig. 4.7, when the MWM is set to 14, the TPS/CTPS results with a different number of nodes, it is clear that as the number of nodes increases, the TPS/CTPS transaction speed approximately increases linearly. Therefore, the transaction speed has good linear scalability when the number of nodes increases. For example, when 50 nodes are sending transactions, the STDI-Based approach TPS reaches 1.376 tx/s and CTPS 6.418 tx/s respectively, whereas with a DAG-Based approach the TPS reaches 1.743 tx/s and CTPS reaches 7.519 tx/s respectively. This demonstrates that the proposed STDI-Based approach outperforms the DAG-Based approach and performs well when the number of nodes increases.

Throughput: As shown in Fig. 4.6 and Fig. 4.7, it is clear that the proposed STDI-Based approach outperforms the DAG-Based approach in terms of efficiency in processing transactions. For example, in the situation in which 10 nodes are sending, the average TPS reaches 1.132 tx/s and CTPS 6.234 tx/s, respectively in STDI-Based approach. Whereas in the DAG-Based approach the TPS reaches 1.314 tx/s and CTPS reaches 7.256 tx/s respectively. This is due to the computation offloading mechanism used in STDI-Based approach.

The results in Fig. 4.8 and Fig. 4.9 are conducted to test the effect of MWM on the TPS and CTPS. In these experiments, we set the MWM to 9,11,13,14 to measure the effect on the

TPS/CTPS. In Fig. 4.8, it is clear that the TPS is affected by the use of different MWM, as when it is set to 13, it almost reaches 5.321 tx/s and when it is set to 14, it almost reaches 6.591 tx/s. From Fig. 4.9, the changes in MWM have almost no influence on the CTPS.

Decentralisation: The proposed SDIT-Based approach outperforms the DAG-Based approach in terms of decentralisation.

4.5 MADIT Experiments, Evaluation and Analysis

In this section, an experimental results and an evaluation of the proposed MADIT solution in terms of scalability and decentralization is presented. In addition, it provides analysis and discussion of the results, to establish important insights that illustrate the usefulness of IOTA tangle integrated with mobile agents for the IoT domain.

4.5.1 Environment Setup

The latest release of IOTA Reference Implementation (IRI 1.8.2) is deployed (IRI 1.8.2)[4], which is the official Java build embodying the IOTA network specifications, on the DigitalOcean cloud platform[5], and another IOTA Reference Implementation (IRI 1.8.2) on a local server dedicated for performing Proof of Work (PoW) operations.

The functionality related to IOTA addresses, transactions, broadcasting, routing, and multi-signatures have been implemented using iota.lib.py [170], the official Python library of the IOTA Distributed Ledger. Different numbers of IOTA participant nodes were used to create the network in order to simulate real life scenarios.

[4]https://github.com/iotaledger/iri/releases/tag/v1.8.2-RELEASE
[5]https://www.digitalocean.com

4.5.2 Results and Analysis

The following two performance metrics are used in the experiments: Transaction Per Second (TPS) and Throughput.

Table 4.1 Performance metrics for experimental work.

Performance Metrics	
Evaluation Metrics	**Definition**
Transaction Per Second (TPS)	refers to the number of transactions published to the Tangle network per second.
Throughput	refers to the efficiency in processing transactions in a given amount of time.

Scalability: The obtained results can be seen in Fig. 4.10. As shown in Fig. 4.10, it is clear that as the number of nodes increases, the TPS transaction speed increases linearly. For example, when the MWM is 9 and 50 nodes are engaged, with one mobile agent dispatched, as shown by the green line, the TPS of MADIT (WA denotes with mobile agents dispatched) reaches 3.749 tx/s (i.e., transactions per second) compared to the baseline (NA denotes no mobile agents dispatched) TPS, which is 2.942 tx/s. Hence, MADIT is 1.27 times faster than the baseline method. Still when the MWM is 9, and the number of nodes is 150, in this case, the average TPS with MA reaches 5.422 tx/s whereas in the baseline, TPS reaches 3.997 tx/s. This time, MADIT is 1.36 times faster than the baseline method. This demonstrates that the proposed MADIT approach is more scalable than the baseline method.

Throughput: As shown in Fig. 4.10, it is clear that the proposed MADIT approach brings an improvement over the baseline approach in terms of efficiency in processing transactions. For example, in the situation in which 150 nodes are engaged, and the MWM is set to 14, the average TPS of baseline reaches 4.176 tx/s (shown by the red line), whereas when employing MAs, the average TPS reaches 2.776 tx/s, as shown by the green line. This is due to two

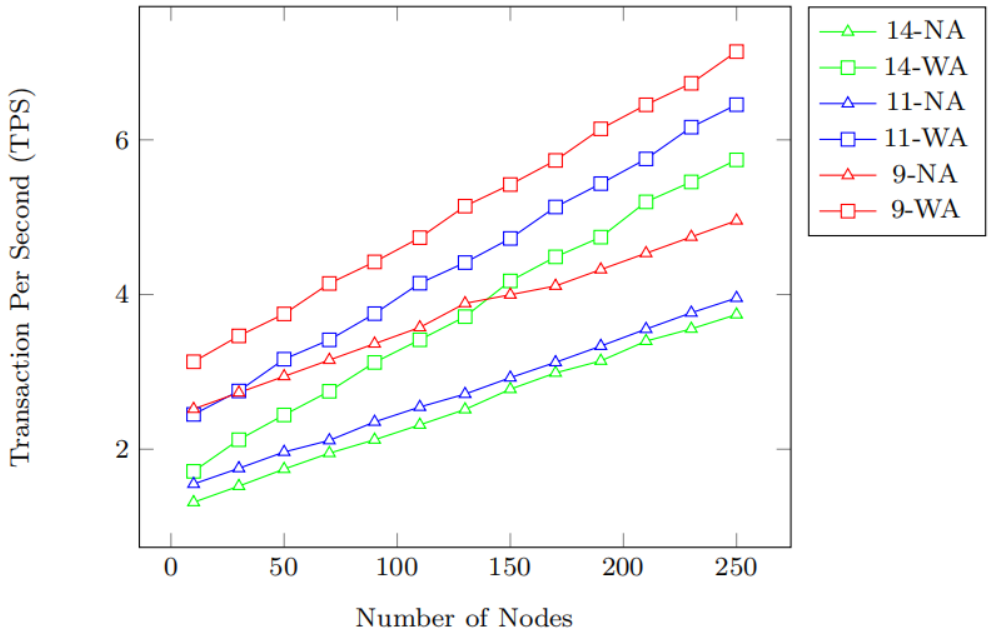

Fig. 4.10 Scalability in Tangle with/without mobile agents

factors: (1) the computation offloading mechanism, and (2) the inclusion of mobile agents in the MADIT approach.

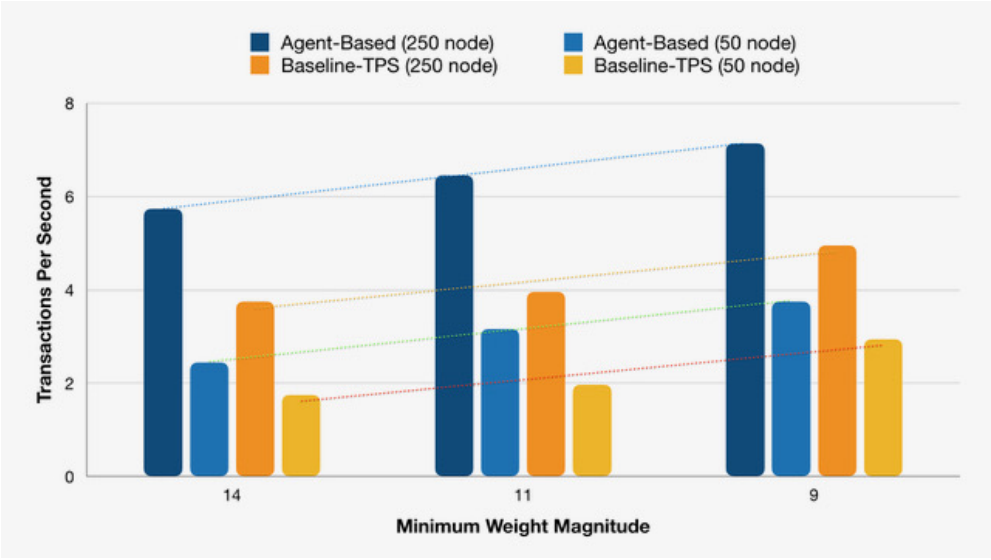

Fig. 4.11 Performance of Baseline-TPS and Agent-Based under different MWM.

Fig. 4.11 demonstrates the effect of the different Minimum Weight Magnitude (MWM) on the TPS. In this experiment, the MWM is set to 9,11,14 to measure the effect on the TPS. In Fig. 4.11, it is clear that the TPS is affected by the use of different MWM configurations as when it is set to 11, it reaches 6.455 tx/s, and when it is set to 14, it reaches 7.141 tx/s.

Decentralisation: The proposed MADIT approach is fully decentralised as the use of the consensus mechanism is adopted.

4.6 Summary

This chapter presented an important step towards the integration of IOTA tangle with the IoT. It described a scalable IOTA Tangle-Based Distributed Intelligence Approach. The results indicate that an IOTA tangle can scale to a large number of IoT devices, thus addressing the scalability challenge in the IoT domain. Compared to existing work, SDIT enables high-scalability and decentralisation possibilities for building large-scale IoT applications. Also, it described the proposed framework, which is called Mobile-Agent Distributed Intelligence Tangle-Based approach (MADIT). The framework advocates IOTA tangle and mobile agents for supporting distributed intelligence in IoT. It presents an IOTA tangle and mobile agents based approaches as a solution to the problem of the limitations of traditional distributed intelligence systems. Mobile agents deliver an efficient way of collecting transactions. The advantages of MADIT include: scalability; decentralisation and the facilitation of node level communications (low level intelligence).

Chapter 5

An Energy Efficient Multi-Mobile Agent Itinerary Planning Approach

Mobile Agent (MA) technology brings many benefits into Wireless Sensor Networks (WSNs), such as saving network bandwidth and enabling energy efficient mechanisms for collecting sensor data. Nowadays, itinerary planning for MAs is one of the most important features of the WSN. However, the way in which all dispatched MAs are routed inside the sensor networks must be intelligently planned to reduce energy consumption and improve information accuracy. There have been many research efforts designing itinerary planning algorithms to deploy multiple MAs in a given sensor network, where routes are generated so that MAs can follow different routes to collect data from sensor nodes efficiently and effectively. This chapter proposes a new energy efficient Graph-based Static Mutli-Mobile Agent Itinerary Planning approach (GSMIP). GSMIP applies Directed Acyclic Graph (DAG) related techniques and divide sensor nodes into different groups based on the routes defined by MAs itineraries. MAs follow the predefined routes and only collect data from the groups they are responsible for. The experimental findings demonstrate the effectiveness and superiority of the proposed approach compared to the existing approaches in terms of energy consumption and task delay (time). The research presented in this chapter was published in [171]

5.1 Introduction

A pervasive interconnected network, including a wireless sensor network (WSN), is defined by its capacity to perform basic tasks through exchanging resources that are in network or in node domains.

One primary aim of WSNs is to allow users to access information of interest from data obtained through spatially distributed sensors. Sensors are generally installed in large numbers to gain full visibility of the controlled physical environment. Such sensor network systems are designed in a way that immense amounts of data will be produced [172]. Mobile agent techniques have been widely used to enable efficient collaborative data collection from a WSN. In these techniques, mobile agents (MAs) will be dispatched, which will traverse the sensor along predefined routes, generated by itinerary planning, to collect data from sensor nodes on the way. The need to locate and handle mobile agents in energy-efficient WSN applications is primarily characterised by approaches used to design the itinerary planning of MAs.

Practical constraints on the implementation of sensor nodes, such as computational capacity and battery-limited sensor nodes makes itinerary planning a challenging [16] task. The critical issues while dispatching a mobile agent include the migration cost of mobile agent, itinerary planning and the approaches to establishing such a plan.

The principal objective of the mobile agent is to collect and process data in a network. Without user interaction, they can combine and make local decisions autonomously. The main reason why mobile agents are used is that radio communication is one of the most effective hungry operations [169]. To avoid long distance radio communication, mobile agents are dispatched to gather data instead of sending it back to a sink node. In such scenarios, planning mobile agent itinerary in order to optimise energy consumption for sensor nodes is critical. However, it has been challenging to solve the problem, which is NP-hard, of finding an ideal sequence of sensor nodes to be visited by a mobile agent [173, 174].

Hence, one main challenge is how to create an appropriate itinerary for MAs to collect data [16]. Itinerary planning refers to identifying a route of a MA which the MA should follow when traversing the sensor network and visiting sensor nodes. Each route contains a sequence of source nodes to be visited through the MA migration trip. Current techniques for the development of MA itineraries can be generally classified into three types: Static itinerary, Dynamic itinerary and Hybrid itinerary [16], which will be discussed in more detail below.

This chapter is based on the data structure used in IOTA tangle, which is a Directed Acyclic Graph (DAG) and applies related techniques to generate an efficient itinerary planning for MAs. It divides sensor nodes into different groups based on the routes defined by MAs itineraries. MAs follow the predefined routes and only collect data from the groups they are responsible for.

MAs have been put forward as an efficient technique for collecting data in WSNs. It was stated in [169] that one of the most power hungry operations is radio communication. MAs facilitate the flexibility and scalability problems of centralised models [99]. MAs can autonomously move among sensor nodes to collect data without requiring human inputs, leading to reductions in energy consumption and network bandwidth usage. This justifies why mobile agent is efficient in collecting data from sensor nodes.

5.2 Components of Mobile Agent

In WSNs, Mobile Agents (MAs) are referred to as software abstractions performing information-rich data collection and autonomous data processing whilst dynamically migrating between network nodes so that data is exchanged between participant nodes [98].

MAs have also recently been suggested to address the limitations of centralised models' scalability and the flexibility problems of static hierarchical frameworks. MAs comprise of four components as shown in Fig. 5.1.

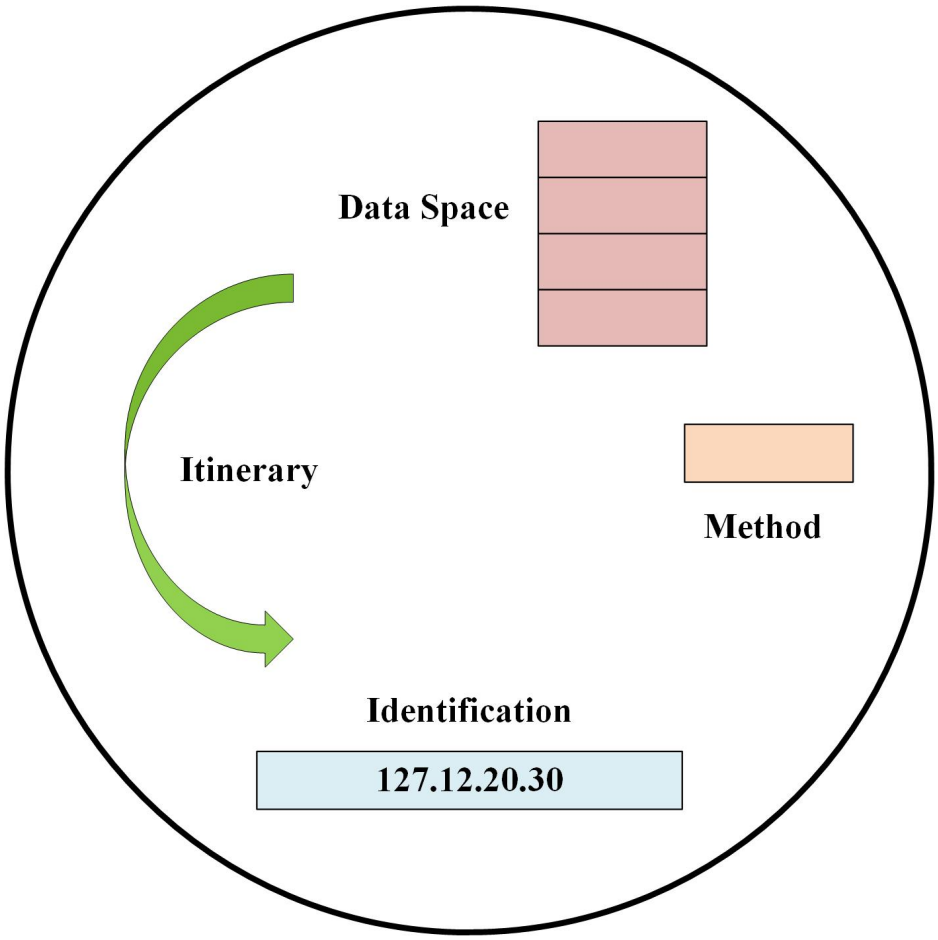

Fig. 5.1 Components of Mobile Agent

- **Itinerary:** It can be identified as the mobile agent trip route for visiting source nodes. Itinerary planning is usually divided into three categories: static, dynamic and hybrid. In a static itinerary, the route is computed at the dispatcher prior to the MA migration. In a dynamic itinerary, the route of MA is determined by the MA on the fly. In a hybrid itinerary, the sensor nodes to be visited by the MA are selected by the dispatcher, but the visiting sequence is determined by the MA on the fly.

- **Data space:** This is the data buffer of MA, and is primarily capable of producing data integration. The findings should have incremental precision as the agent moves from one node to another.

• **Identification:** This is a unique number that identifies the mobile agent and the dispatcher. Typically presented in a 2-tuple (i : j) format, where i denotes the dispatcher's IP address, and j is a serial number assigned to each MA by the dispatcher.

• **Method:** This is the execution code that each MA executes.

5.3 Proposed GSMIP Itinerary Planning Approach

5.3.1 GSMIP Architecture

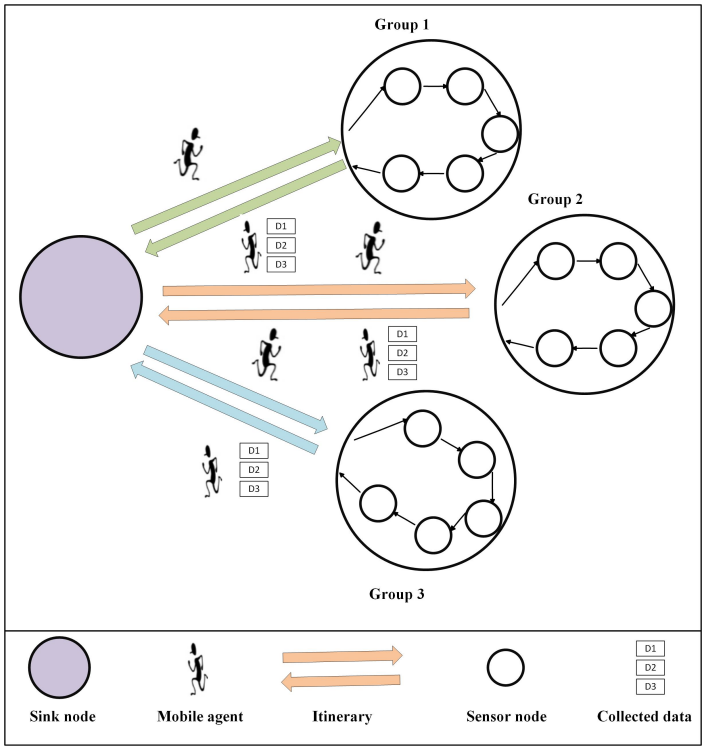

Fig. 5.2 The Proposed Mobile Agent Itinerary Planning Approach

Fig. 5.2 presents an abstract view of the proposed Graph-based Static Mutli-Mobile Agent Itinerary Planning approach (GSMIP). It shows a set of sensor nodes in each route and how nodes are grouped together. The routes are generated to cover all nodes in the network. The sink node is responsible for dispatching MAs to each group in order to collect data from.

The MAs collect data from the groups that they are assigned to. For example, the itinerary (e.g., orange lines) represents the routes to the assigned group.

Fig. 5.3 describes an example of the working principles of the algorithms. It shows a set of routes (route 1, route 2 and route 3 in this example) that cover all nodes, including private nodes and shared nodes, in the network. Private nodes are nodes that belong to a particular route only. Shared nodes are source nodes that are on multiple routes. There is only one shared node in this example, and it is on both route 1 and route 2. In addition, a group is a collection of private nodes and allocated shared nodes in a particular route. The groups are generated based on allocating shared nodes to the group of a route with the least number of nodes. Take Fig. 5.3 as an example. Since the group of nodes for route 2 has only two private nodes, while that of route 1 has three private nodes, the shared node is allocated to the group for route 2. Note that, the source sink (e.g., dispatcher) and the sink node (e.g., destination) in practice can be the same (sink) node. Here, because the network is modeled as a DAG, it virtually divide it into two nodes, where each node will make use part of the links to source nodes in the network.

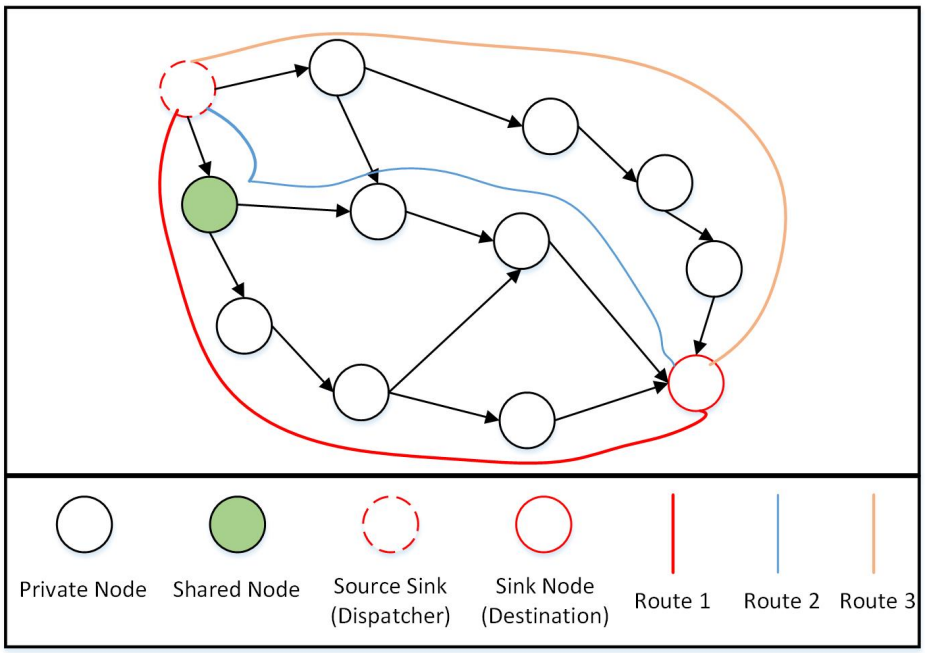

Fig. 5.3 An Example of the Working Principles of the Algorithms

5.4 Experiments, Evaluation and Analysis

5.4.1 Simulation Setup

The proposed GSMIP is implemented and tested and is compared with existing approaches from the literature, namely SMIP [19], GIGM-MIP [20], and CL-MIP [21], using the Pymote simulator [175].

Pymote focuses on WSNs, which generally are networks of low power embedded devices. It is widely used by many researchers and developers to test algorithms. The network model is adopted from [21] where 100 sensor nodes are uniformly deployed in an area and the sink node is placed at the centre of the area.

5.4.2 Simulation Parameters

The sensor nodes are static and uniformly deployed within a *1000 m* × *500 m* network size. The number of sensor nodes is set to *100* and sensor nodes are randomly distributed in the network. The sink node is located at the center of the network, and it has unlimited energy supply and higher computational capability. All the sensor nodes have the same initial energy. The radio transmission range is set to *60 m*, while the raw data size is *2048 bits*. The mobile agent code size is*1024 bits*, the data processing rate is *50 Mbps*, the raw data reduction ratio is *0.8* and the mobile agent accessing delay is *10 ms*. The experiments consider all types of energy consumption in the simulations, including transmission, idling and sensing. Table 5.1 lists all of the mobile agent parameters used during simulation.

Table 5.1 Simulation Parameters of the Proposed GSMIP Approach.

Network Size	1000 m x 500 m
Number of Sensor Nodes	100
Raw Data Size	2048 bits
MA Code Size	1024 bits
Data Processing Rate	50 Mbps
Raw Data Reduction Ratio	0.8
Aggregation Ratio	0.9
Radio Transmission Range	60 m
MA Accessing Delay	10 ms

Table 5.2 Performance metrics for experimental work.

Performance Metrics	
Evaluation Metrics	**Definition**
Energy Consumption	refers to the energy spent for transmitting, and receiving messages by mobile agent from all sensor nodes.
Task Duration	refers to the average time when the mobile agents are dispatched by the sink to the time when the last mobile agent returns back to the sink.
Dispatched Mobile Agent	refers to the number of dispatched mobile agent to collect data from all sensor nodes.

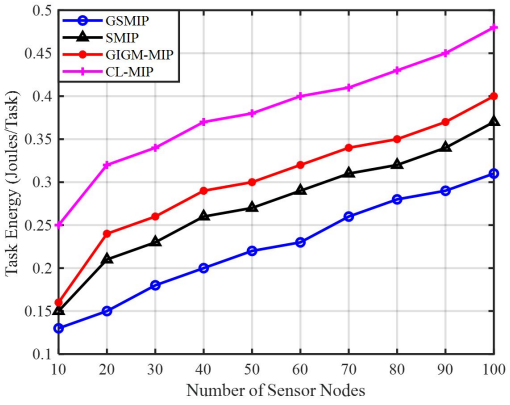

Fig. 5.4 The Impact of Number of Sensor Nodes on Consumed Energy

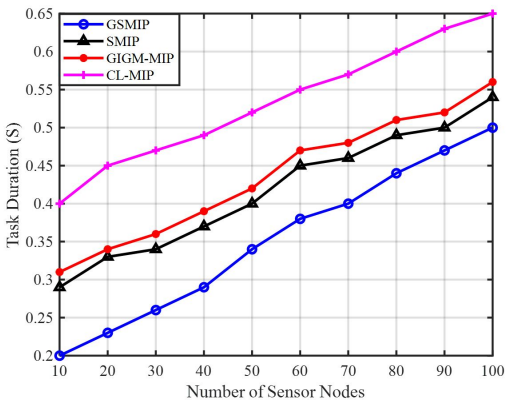

Fig. 5.5 The Impact of Number of Sensor Nodes on Task Duration

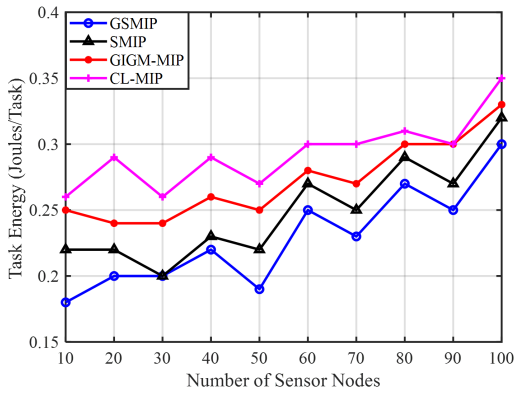

Fig. 5.6 The Impact of Number of Sensor Nodes on Consumed Energy

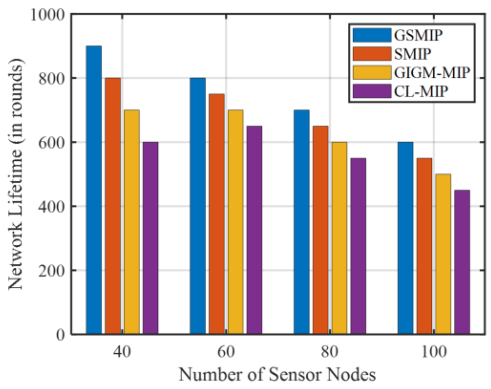

Fig. 5.7 The Impact of Number of Sensor Nodes on the Network Lifetime

5.4.3 Evaluation and Analysis

To evaluate the performance of different approaches, the following three performance metrics are considered: Task Duration, Energy Efficiency and Number of Dispatched Mobile Agents. Table 5.2 describes these three performance metrics.

Energy Efficiency: As shown in Fig. 5.4, it is clear that the proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption. More energy is required for more agents to perform tasks in all of the four approaches. But it can be observed that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 31.2% and 13.3% energy decrease when compared

to SMIP 36.4% and 15.2%, GIGM-MIP 47.4% and 17.1%, and CL-MIP 48.5% and 25.6% when the number of nodes decreases from 100 to 10. The CL-MIP algorithm consumes the highest energy, and this is because of the distribution of a large number of mobile agents to the sensor network leading to an increase in the number of mobile agents hops. The GIGM-MIP algorithm has better energy consumptions compared to the CL-MIP when the number of sensor nodes increases. This is due to the fact that the GIGM-MIP algorithm distributes the data collection between mobile agents in each region of the network. The proposed GSMIP achieves better energy consumption compared to SMIP, GIGM-MIP and CL-MIP respectively. This achievement is obtained for two main reasons including: efficient partitioning i.e., grouping technique and the shortest itinerary of the mobile agents in each group. The efficient partitioning of the network constructs less itineraries for the mobile agents in each group. This minimises the number of hops for each mobile agent in each group.

Task Duration: Fig. 5.5 compares the four approaches in terms of task duration. It is observed that the proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. It can be observed that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 50.8% and 20.4% task duration decrease when compared to SMIP 54.9% and 28.7%, GIGM-MIP 56.3% and 32.5%, and CL-MIP 65.9% and 40.2%, which has the highest delay when the number of nodes decreases from 100 to 10. In SMIP, GIGM-MIP and CL-MIP algorithms, each mobile agent is scheduled to visit all sensor nodes with the static routes determined by the sink node to collect data from sensor nodes. The process increases the number of mobile agents hops, leading to a considerable delays. Meanwhile, The proposed GSMIP has better task duration because of less nodes that are required to be visited by the mobile agents i.e., shortest itineraries.

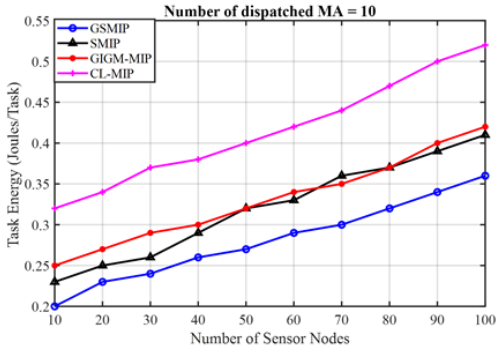

Fig. 5.8 The Impact of Number of Dispatched Mobile Agent's on energy consumption (10 MAs)

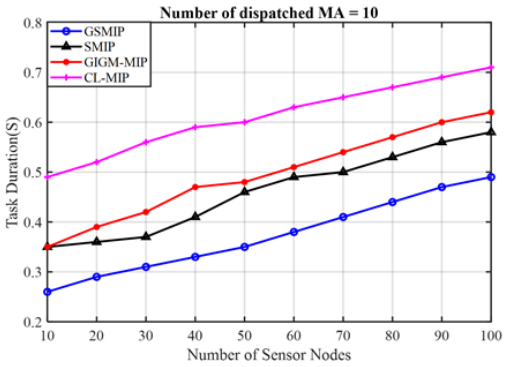

Fig. 5.9 The Impact of Number of Dispatched Mobile Agent's on Task duration (10 MAs)

Energy Efficiency: As shown in Fig. 5.6, which shows the effect of energy consumption by considering different parameters. It is clear that the proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption. It can observed that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 33.3% and 27.1% energy decrease when compared to SMIP 38.1% and 31.2%, GIGM-MIP 44.7% and 35.3%, and CL-MIP 53.5% and 40.9% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is because the number of hops for each MA is minimised within the groups.

Network Lifetime Fig. 5.7 shows the impact of the number of sensor nodes on the network lifetime. It can be observed that as the number of nodes increases, the network lifetime for the proposed GSMIP is almost the same as compared to SMIP, GIGM-MIP, and CL-MIP that shows a noticeable decrease. This is due to the fact that the proposed GSMIP approach applies data aggregation and therefore carries a smaller size of the data packets, which leads to less energy consumption thereby increasing network lifetime.

Energy Efficiency: As shown in Fig. 5.8, which shows the effect of the number of dispatched MAs on energy consumption. It is clear that the proposed GSMIP approach

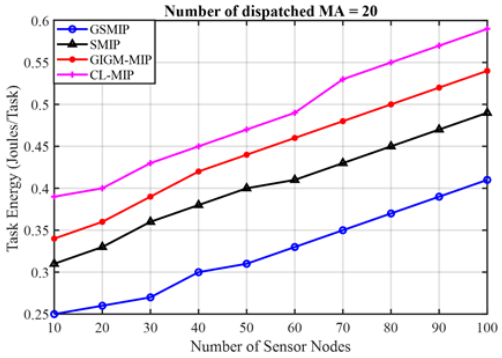

Fig. 5.10 The Impact of Number of Dispatched Mobile Agent's on energy consumption (20 MAs)

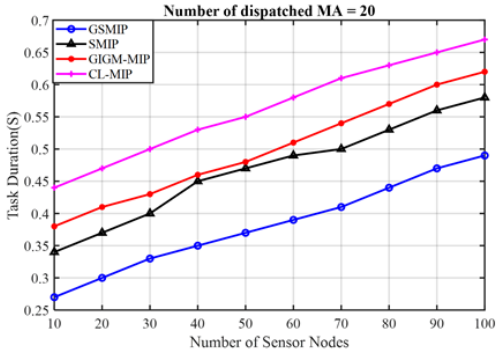

Fig. 5.11 The Impact of Number of Dispatched Mobile Agent's on Task duration (20 MAs)

outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption while varying the number of dispatched MAs. This experiment considers the number of dispatched MAs as 10. It can be observed that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 36.3% and 20.1% energy decrease when compared to SMIP 41.2% and 24.1%, GIGM-MIP 43.4% and 25.1%, and CL-MIP 53.4% and 32.7% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is due to the fact that the number of hops for each MA is minimised within the groups.

Task Duration: Fig. 5.9 compares the four approaches in terms of task duration while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 10. It is observed that the proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. It can be observed that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 49.2% and 22.6% task duration decrease when compared to SMIP 59.8% and 32.9%, GIGM-MIP 61.2% and 32.5%, and CL-MIP 71.5% and 49.3%, which has the highest delay when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach for the task duration is due to the fact that the shortest itineraries are constructed for each MA in each group.

Energy Efficiency: As shown in Fig. 5.10, which shows the effect of the number of dispatched MAs on energy consumption. It is clear that the proposed GSMIP approach outperforms SMIP, GIGM-MIP, and CL-MIP approaches in terms of energy consumption while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 20. It can observed that the proposed GSMIP exhibits better energy saving over other approaches. The proposed GSMIP approach achieves 41.4% and 25.2% energy decrease when compared to SMIP 49.6% and 31.3%, GIGM-MIP 57.5% and 34.2%, and CL-MIP 59.6% and 40.8% when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach on energy consumption is due to the fact that the number of hops for each MA is minimised within the groups.

Task Duration: Fig. 5.11 compares the four approaches in terms of task duration while varying the number of dispatched MAs. The experiment considers the number of dispatched MAs as 20. It is observed that the proposed GSMIP approach outperforms the three existing approaches, including SMIP, GIGM-MIP and CL-MIP. It can be observed that the proposed GSMIP achieves the best task duration of all approaches. The proposed GSMIP achieves 50.3% and 26.5% task duration decrease when compared to SMIP 56.6% and 34.8%, GIGM-MIP 63.3% and 37.4%, and CL-MIP 67.3% and 45.4%, which has the highest delay when the number of nodes decreases from 100 to 10. The improvement of the GSMIP approach for the task duration is due to the fact that the shortest itineraries are constructed for each MA in each group.

The Effect of the Number of Dispatched Mobile Agents on Task Duration: Fig. 5.13 demonstrates the impact of the number of dispatched MAs on task duration for the proposed GSMIP. From Fig. 5.13, it is observed that when 20 mobile agents are dispatched and the number of nodes is 100, the task duration reaches 0.34/S; whereas when 30 mobile agents are dispatched and the number of sensor nodes is 100 the task duration reaches 0.27/S. Also, It is clear that when 40 mobile agents are dispatched and the number of sensor nodes is 100, the

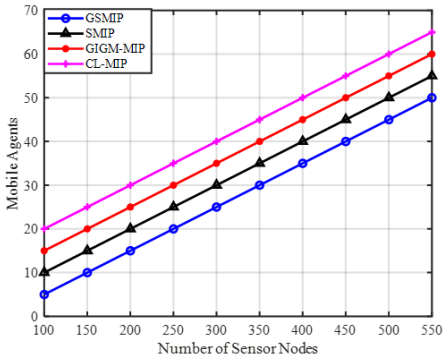

Fig. 5.12 The number of dispatched mobile agent of the proposed GSMIP and alternative approaches

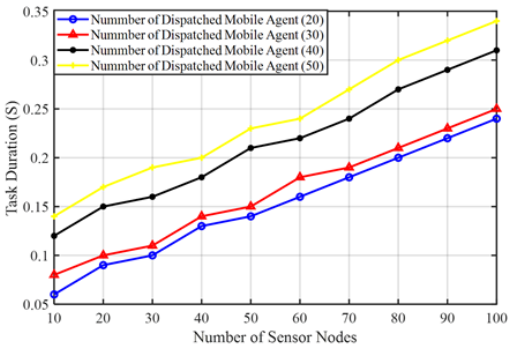

Fig. 5.13 The Impact of Number of Dispatched Mobile Agent's on Task duration (MAs 20, 30, 40, 50)

task duration reaches 0.24/S whereas when 50 mobile agents are dispatched and the number of sensor nodes is 100 the task duration reaches 0.22/S.

Number of Dispatched Mobile Agent: As shown in Fig. 5.12, which shows the number of dispatched mobile agent. It is clear that the CL-MIP dispatches the highest number of mobile agents, which is more than 60 MAs whereas GIGM-MIP dispatches 60 MAs followed by SMIP that dispatches more than 50 MAs. On the other side, the proposed GSMIP dispatches the minimum MAs, which is 50 MAs.

5.5 Summary

This chapter addressed the issue of efficient multi-mobile agents itinerary planning. In order to achieve scalability and to reduce energy consumption, an energy-efficient itinerary planning approach, called GSMIP, has been proposed. A grouping strategy is developed where nodes in the network are assigned into smaller sets, making energy depletion less of a problem. Experimental evaluation shows that the proposed itinerary planning approach is scalable, energy-efficient, and reduces delays while increasing network lifetime.

Chapter 6

Conclusion and Future Work

6.1 Introduction

This chapter concludes the study and presents a detailed conclusion of each chapter, along with the aim, objectives and achievements. In addition, it provides complete future research directions and open research questions that require further investigation.

6.2 Conclusion

The work in this thesis proposes a distributed intelligence framework that can be useful in various IoT application scenarios. In addition, the thesis proposes an energy efficient multi-mobile agent itinerary planning approach. The results obtained for the scalability and energy-efficiency are promising, and every algorithm proposed is implemented so that experimental comparison can be made against the existing state of the art approaches. Over the last few years, the Internet of Things (IoT) have been widely adopted for delivering services across various domains, from smart cities, smart campus and intelligent transportation systems to industry. IoT interconnects heterogeneous devices with diverse functionalities to meet the evolving requirements of the earlier mentioned domains. IoT devices are characterised

by limited resources, such as power consumption, memory and processing. Distributed Intelligence is defined as a sub-discipline of artificial intelligence, which allows processing functionality to be distributed, enables collaboration between smart objects and mediates data exchange to optimise communication for IoT applications. This concept may be spilt into two levels: intelligence at a high level and intelligence at a low level.

Chapter 2 describes distributed intelligence in IoT that underpins this research i.e., the existing distributed intelligence approaches in IoT; the concept of distributed intelligence; the motivation and challenges of distributed intelligence; and an overview of hardware-based security primitives techniques for IoT. For the first part, the taxonomy of distributed intelligence is illustrated in details including: distributed intelligence challenges, intelligence levels and classifications of distributed intelligence approaches. For the second part, the mobile agent and an overview of the current techniques are presented in details including single itinerary planning and multiple itinerary planning. It's shown that while distributed intelligence is an active research area, there are challenges in ensuring scalability and decentralisation. Finally, a description of the theories of the contributions is presented followed by a discussion of the challenges and future research directions that requires further investigation.

Chapter 3 presents the theories and background of the IOTA distributed ledger technology. The fundamental working principles of IOTA technology and the components of IOTA including: masked authenticating messaging, IOTA smart contract, relationship between coordicide and coordinator, auto-peering, and snapshotting are described in details. The suitability of IOTA technology to IoT followed by the characteristics of IOTA. The similar distributed intelligence approaches and mobile agent itinerary planning algorithms that are closely related to the work in this thesis are also presented. Finally, a three different IoT use case scenarios that would benefit from the IOTA technology e.g, smart parking, smart

campus, and self-driving vehicles are described in details. This chapter concluded that IOTA is an efficient and suitable technology to support distributed intelligence.

Chapter 4 focuses on the design and development of the proposed distributed intelligence framework and its related components including: IoT devices to collect data, tangle to process transactions, mobile agent to collect transaction and Proof of Work (PoW) enabled server for performing heavy computations tasks. Most current distributed intelligence approaches are configured to transmit data to a central location for further processing. However, such approaches do not focus on the scalability at the physical layer and consume energy due to transmission as well as network bandwidth. By considering the limited resources of IoT devices such as power consumption, enabling distributed intelligence was achieved through two levels including: high and low. In the high level, a tangle-based architecture is used to deal with transactions, while low level adopts mobile agents to cater for node level communications. The advantages include: scalability; decentralisation, elimination of redundant transaction data and the facilitation of node level communications. A simulation is conducted in order to evaluate the advantages of the proposed framework in a larger-scale network.

Chapter 5 focuses on the design and development of the proposed multi-mobile agent itinerary planning approach. The approach describes all relevant components including: sensor nodes mobile agents, mobile agent itinerary and collected data. The sensor nodes are deployed to sense the environment. The mobile agents are dispatched to collect data from a particular group. The itinerary is the route followed during mobile agent migration. It has been identified that most current multi-mobile agent approaches are designed to be scalable and energy-efficient. However, such approaches have several issues including: lack of determining the optimal number of mobile agents and efficiently partitioning the sensor network into groups. By considering these issues, a grouping strategy is introduced

where nodes in the network are assigned into smaller sets, making energy depletion less of a problem. A simulation is conducted to validate the performance of the proposed approach.

6.3 Aim, Objectives, and Achievements

This research project is aiming to develop a scalable and energy-efficient distributed intelligence framework for the IoT. The framework adopts the IOTA tangle architecture and mobile agents in order to enable distributed intelligence whilst minimising energy-consumption and ensuring scalability. Also, the framework develops a new multi-mobile agent itinerary planning approach that is scalable and energy-efficient.

Objective One: To examine and identify common requirements of existing distributed intelligence approaches in IoT.

Achievement One: The first objective was met via a comprehensive review that was carried out covering existing distributed intelligence approaches and classified them into five categories namely: cloud computing, mist computing, distributed ledger technology, service-oriented computing and hybrid. In addition, a review of mobile agent itinerary planning approaches have been presented in Chapter Two.

Objective Two: To develop a distributed intelligence framework using multi-mobile agents and IOTA technology as an efficient architectural technique to facilitate local interaction, collection, aggregation of transactions data and it enables the deployment of IoT applications that are scalable and energy efficient.

Achievement Two: The second objective was satisfied through the development of a new Mobile-Agent Distributed Intelligence Tangle-Based framework (MADIT) that adopts the IOTA tangle to support high-level intelligence and multi-mobile agents to support

low-level intelligence i.e, cater for node level communications. The framework is discussed in more details in Chapter Four.

Objective Three: To evaluate an existing Proof of Work (PoW) offloading mechanism for efficacy with regard to energy efficiency and transaction throughput.

Achievement Three: The third objective was met through the evaluation of the IOTA Proof of Work (PoW) offloading mechanism in which the PoW computation was offloaded to devices with higher resources for efficacy with regard to energy efficiency and transaction throughput. The PoW computation offloading is discussed in Chapter Four.

Objective Four: To develop a new energy-efficient multi-mobile agent itinerary planning mechanism by partitioning Directed Acyclic Graph (DAG) into groups and allowing mobile agents to visit a particular group.

Achievement Four: The fourth objective was satisfied through the proposed Graph-based Static Multi-Mobile Agent Itinerary Planning approach (GSMIP). It applies Directed Acyclic Graph (DAG) related techniques to divide the network into several groups, which will eventually allow each mobile agent to visit a particular group. The proposed approach is discussed in more details in Chapter Five.

6.4 Future Work

As this is an emerging research field, there are a number of interesting directions for future work that can be used to extend the work further. Below is a summary of interesting directions for future work

6.4.1 Hybrid and Adaptable Framework

IoT networks consists of heterogeneous devices ranging from low-power devices to high end servers. Therefore, a single solution would not be deployed for all IOTA-based IoT architectures. This is because of the different capabilities provided by IoT networks. Consequently, a possible solution should initially be adaptable and take into consideration all of the IoT constraints. Thus, one of the challenges that require further attention in the future is how to design and develop a hybrid, dynamic and adaptable framework for IOTA-based IoT architectures. The main questions that might arise are as follows:

- Where functionality should be invoked?
- Where heavy computation tasks should be placed?
- How much support control should the framework allow?
- How to support cooperation among IoT devices to cater for node level communications in an efficient way?

6.4.2 Energy-Efficiency

Due to the high power consumption of PoW, computation offloading is a suitable solution for saving energy consumption. Another interesting research direction for reducing energy consumption of IoT devices would be to apply the mechanism of "Compute and Wait" where several proof rounds are needed. For example, any given consensus node must resolve the game in the the current round before participating in the next round. Consequently, a node that solves the game of current round can only move on to the next round if a predetermined number of solutions have been found by other nodes as described in [176]. This would result in a significant reduction in energy-consumption.

6.4.3 Security Against Attack

Security is considered as a crucial challenge that is required by almost all of the IoT applications to avoid cyber attack. For example, IoT devices, deployed IOTA nodes and IoT gateways can be affected by cyber attack, which would lead to unusual behaviour and functioning of these IoT devices. This can lead to provide wrong decisions in response to emergent situations. Furthermore, the security of the network is also important as it is capable of protecting the IoT system from various attacks such as sniffer and jamming. Cyber security [42] to be added as an important enhancement to many IoT applications.

6.4.4 Privacy-Preserving

IOTA Streams has been developed to ensure privacy and authentication when sending IoT data. It can be applied to enable multiparty authentication scenarios [5]. Moreover, location privacy [43] that focuses on how to effectively select reasonable dummy locations and avoid having the real locations. These are considered as an important issues for providing an effective IoT privacy [28, 177]. IOTA streams combined with Dynamic Searchable Symmetric Encryption (DSSE) [178] would lead to an efficient privacy solution for the IoT healthcare application. An important privacy issues arise as follows:

- How to design access control mechanism based on IOTA technology to preserve transaction privacy in IoT healthcare applications?

- How to use IOTA token mixers to guarantee privacy?

- How to design a set of suitable forms as a proof of concept to support a complete privacy solution?

6.4.5 Adaptive Routing Protocol

IOTA tangle can be used to generate an effective routing protocol for IoT networks. A routing protocol should balance the following characteristics: energy-efficiency, scalability, robustness and Quality of Service (QoS) [179]. It has been suggested that clustering is an important factor in determining a successful implementation of any routing protocol. Hierarchical routing consists of two tiers including upper and lower. In the upper-tier, nodes are called cluster head and act as routing backbone, while nodes in the lower-tier are concerned with sensing activities. The main questions that might arise are as follows:

- How to cluster the network in an efficient way?
- How to elect cluster head nodes? What metrics should be considered when selecting a cluster head node?
- When to rotate cluster head nodes, e.g., based on the level of remaining energy?

6.4.6 Offline Capability

The IOTA tangle can be used to solve the problem of offline capability. This task is not simply a network entities configuration problem; the major issue is related to clustering the network. However, it can be achieved by creating offline tangle where a certain number of nodes can effectively go offline and issue transactions among themselves. This means that an active internet connection is not needed, while the tangle is offline. Upon completion, it is possible to simply attach the transactions of the offline tangle back to the online one.

6.4.7 Dynamic Multi-Mobile Agents Itinerary Planning

How to derive a dynamic or a hybrid itinerary plan for mobile agents is a critical task, which allows each mobile agent to decide the visiting sequence on-the-fly. This is particularly

useful for providing fault-tolerance and can be achieved by adopting an efficient clustering method in which nodes will be grouped according to specific criteria and mobile agents will be directed to a particular group as described in [165].

- How to route mobile agent among sensor nodes in an intelligent way?
- How to design a dynamic itinerary planning that enables mobile agents to decide the visiting sequence on the fly?
- How to design and develop an efficient grouping mechanism that would aid in mobile agent itinerary?

6.4.8 IOTA Tangle in Wireless Sensor Networks

The benefits offered by IOTA tangle can be explored in other areas, such as Wireless Sensor Networks (WSN). It will not necessarily be pertinent to the scalability and energy-efficiency issues and undoubtedly these issues will be taken into consideration. In addition, it would be interesting to investigate the possibility of adapting it to suit Information Extraction (IE) techniques in WSNs such as event-driven (Threshold-based), time-driven (periodic) and query-based (request-response) [159]. Therefore, not limiting the benefits of IOTA tangle to a specific problem or problem domain.

Finally, how to design and develop a *new* programming abstraction model [180] that will suit all of the IE techniques. Consequently, it will be used as a building block in establishing an infrastructure for a *new* integrated hybrid Information Extraction framework. It will be made up of a specific, customised components and techniques along with the development of distributed algorithms from several technologies such as Network Function Virtualization (NFV) [125], Coordination Models and Languages [181], Distributed Ledger technology [182] and Micro-services [183], wrapped up with an Application Programming Interface (API).

References

[1] Luigi Atzori, Antonio Iera, and Giacomo Morabito. The internet of things: A survey. *Computer Networks*, 54(15):2787–2805, 2010.

[2] Cisco. Internet of things at a glance. (1), December 2016.

[3] Gartner. Gartner says the internet of things installed base will grow to 26 billion units by 2020. (1), December 2013.

[4] API Research. More than 30 billion devices will wirelessly connect to the internet of everything in 2020. (1), May 2013.

[5] Hussain Al-Aqrabi, Anju Pulikkakudi Johnson, Richard Hill, Philip Lane, and Lu Liu. A multi-layer security model for 5g-enabled industrial internet of things. In *7th International Conference on Smart City and Informatization (iSCI 2019), Guangzhou, China, November 12-15, 2019*, Lecture Notes in Computer Science, Switzerland, 8 2019. Springer International Publishing AG.

[6] Tariq A. A. Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Enabling distributed intelligence for the internet of things with IOTA and mobile agents. *Computing*, 102(6):1345–1363, 2020.

[7] Carlos Cares, Samuel Sepúlveda, and Claudio Navarro. *Agent-Oriented Engineering for Cyber-Physical Systems: Helping Teachers Develop Research Informed Practice*, pages 93–102. 02 2019.

[8] Charith Perera, Yongrui Qin, Julio C. Estrella, Stephan Reiff-Marganiec, and Athanasios V. Vasilakos. Fog computing for sustainable smart cities: A survey. *ACM Comput. Surv.*, 50(3):32:1–32:43, June 2017.

[9] Tam Thanh Doan, Reihaneh Safavi-Naini, Shuai Li, Sepideh Avizheh, Muni Venkateswarlu K., and Philip W. L. Fong. Towards a resilient smart home. In *Proceedings of the 2018 Workshop on IoT Security and Privacy*, IoT S&P '18, pages 15–21, New York, NY, USA, 2018. ACM.

[10] E. De Angelis, A.L.C. Ciribini, L.C. Tagliabue, and M. Paneroni. The brescia smart campus demonstrator. renovation toward a zero energy classroom building. *Procedia Engineering*, 118:735–743, 2015.

[11] Hussain Al-Aqrabi, Anju P Johnson, Richard Hill, Phil Lane, and Tariq Alsboui. Hardware-intrinsic multi-layer security: A new frontier for 5g enabled iiot. *Sensors*, 20(7):1963, 2020.

[12] Michael Armbrust, Armando Fox, Rean Griffith, Anthony D. Joseph, Randy H. Katz, Andy Konwinski, Gunho Lee, David A. Patterson, Ariel Rabkin, Ion Stoica, and Matei Zaharia. A view of cloud computing. *Commun. ACM*, 53(4):50–58, 2010.

[13] Floris Van den Abeele, Jeroen Hoebeke, Girum Ketema Teklemariam, Ingrid Moerman, and Piet Demeester. Sensor function virtualization to support distributed intelligence in the internet of things. *WIRELESS PERSONAL COMMUNICATIONS*, 81(4):1415–1436, 2015.

[14] M. J. M. Chowdhury, M. S. Ferdous, K. Biswas, N. Chowdhury, A. S. M. Kayes, M. Alazab, and P. Watters. A comparative analysis of distributed ledger technology platforms. *IEEE Access*, 7:167930–167943, 2019.

[15] Tariq Alsbou'i, Mohammad Hammoudeh, Zuhair Bandar, and Andy Nisbet. An overview and classification of approaches to information extraction in wireless sensor networks. 08 2011.

[16] Tariq Alsbou'i, Mustafa al Rifaee, Rami Etaywi, and Mohammad Abdul Jawad. Mobile agent itinerary planning approaches in wireless sensor networks-state of the art and current challenges. volume 188, 11 2017.

[17] Q. Wu, N.S.V. Rao, J. Barhen, S.S. Iyenger, V.K. Vaishnavi, H. Qi, and K. Chakrabarty. On computing mobile agent routes for data fusion in distributed sensor networks. *IEEE Transactions on Knowledge and Data Engineering*, 16(6):740–753, 2004.

[18] Gang Chen, Sai Wu, Jingbo Zhou, and Anthony K.H. Tung. Automatic itinerary planning for traveling services. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):514–527, 2014.

[19] Huthiafa Qadori, Zuriati Zukarnain, Mohd Hanapi Zurina, and Shamala Subramaniam. A spawn mobile agent itinerary planning approach for energy-efficient data gathering in wireless sensor networks. *Sensors*, 17:1280, 06 2017.

[20] Imene Aloui, Okba Kazar, Laid Kahloul, and Sylvie Servigne. A new itinerary planning approach among multiple mobile agents in wireless sensor networks (wsn) to reduce energy consumption. *International Journal of Communication Networks and Information Security (IJCNIS)*, 7:116–122, 08 2015.

[21] Min Chen, Sergio González-Valenzuela, and Yan Zhang. Multi-agent itinerary planning for wireless sensor networks. volume 22, pages 584–597, 11 2009.

[22] E. Bright Wilson. An introduction to scientific research. pages 7–104. Dover Publications, 11 1991.

[23] Tariq Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Distributed intelligence in the internet of things: Challenges and opportunities. *SN Comput. Sci.*, 4(1):1, 2021.

[24] Tariq Alsboui, Yongrui Qin, and Richard Hill. Enabling distributed intelligence in the internet of things using the IOTA tangle architecture. In Muthu Ramachandran, Robert John Walters, Gary B. Wills, Víctor Méndez Muñoz, and Victor Chang, editors,

Proceedings of the 4th International Conference on Internet of Things, Big Data and Security, IoTBDS 2019, Heraklion, Crete, Greece, May 2-4, 2019, pages 392–398. SciTePress, 2019.

[25] Tariq Alsboui, Yongrui Qin, and Richard Hill. Towards a scalable iota tangle-based distributed intelligence approach for the internet of things. In *Intelligent Computing*, Advances in Intelligent Systems and Computing. Springer Verlag, 10 2019.

[26] Charles C. Byers and Patrick Wetterwald. Fog computing distributing data and intelligence for resiliency and scale necessary for iot: The internet of things (ubiquity symposium). *Ubiquity*, 2015(November):4:1–4:12, November 2015.

[27] Michael Vögler, Johannes Schleicher, Christian Inzinger, and Schahram Dustdar. A scalable framework for provisioning large-scale iot deployments. *ACM Transactions on Internet Technology*, 16:1–20, 03 2016.

[28] Jasenka Dizdarevic, Francisco Carpio, Admela Jukan, and Xavi Masip. A survey of communication protocols for internet of things and related challenges of fog and cloud computing integration. *ACM Computing Surveys*, 51, 04 2018.

[29] Tariq Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Enabling distributed intelligence in the internet of things with iota and mobile agents. *Computing*, xx, 01 2020.

[30] Eugene Siow, Thanassis Tiropanis, and Wendy Hall. Analytics for the internet of things: A survey. *ACM Comput. Surv.*, 51(4):74:1–74:36, 2018.

[31] Alicia Klinefelter, Nathan E. Roberts, Yousef Shakhsheer, Patricia Gonzalez, Aatmesh Shrivastava, Abhishek Roy, Kyle Craig, Muhammad Faisal, James Boley, Seunghyun Oh, Yanqing Zhang, Divya Akella, David D. Wentzloff, and Benton H. Calhoun. self-powered iot soc with integrated energy-harvesting power management and ulp asymmetric radios. In *2015 IEEE International Solid-State Circuits Conference-(ISSCC) Digest of Technical Papers*, pages 1–3, 2015.

[32] Atis Elsts, Efstathios Mitskas, and George Oikonomou. Distributed ledger technology and the internet of things: A feasibility study. pages 7–12, 11 2018.

[33] Jumana Haimour and Osama Abu-Sharkh. Energy efficient sleep/wake-up techniques for iot: A survey. pages 478–484, 04 2019.

[34] Andre B. Bondi. Characteristics of scalability and their impact on performance. In *Workshop on Software and Performance*, pages 195–203, 2000.

[35] Mardiana binti Mohamad Noor and Wan Haslina Hassan. Current research on internet of things (iot) security: A survey. *Computer Networks*, 148:283–294, 2019.

[36] Kewei Sha, Wei Wei, T. Andrew Yang, Zhiwei Wang, and Weisong Shi. On security challenges and open issues in internet of things. *Future Generation Computer Systems*, 83:326–337, 2018.

[37] J. Granjal, E. Monteiro, and J. Sá Silva. Security for the internet of things: A survey of existing protocols and open research issues. *IEEE Communications Surveys Tutorials*, 17(3):1294–1312, thirdquarter 2015.

[38] L. Nastase. Security in the internet of things: A survey on application layer protocols. In *2017 21st International Conference on Control Systems and Computer Science (CSCS)*, pages 659–666, May 2017.

[39] Kewei Sha, T. Andrew Yang, Wei Wei, and Sadegh Davari. A survey of edge computing based designs for iot security. *Digital Communications and Networks*, 2019.

[40] Stefano Tedeschi, Jörn Mehnen, and Rajkumar Roy. Iot security hardware framework for remote maintenance of legacy machine tools. In *Proceedings of the Second International Conference on Internet of things and Cloud Computing, ICC 2017, Cambridge, United Kingdom, March 22-23, 2017*, pages 43:1–43:4, 2017.

[41] Sibin Mohan, Mikael Asplund, Gedare Bloom, Ahmad-Reza Sadeghi, Ahmad Ibrahim, Negin Salajageh, Paul Griffioen, and Bruno Sinopoli. The future of iot security: special

session. In *Proceedings of the International Conference on Embedded Software, EMSOFT 2018, Torino, Italy, September 30 - October 5, 2018*, page 16, 2018.

[42] Amandeep Singh Sohal, Rajinder Sandhu, Sandeep K. Sood, and Victor Chang. A cybersecurity framework to identify malicious edge device in fog computing and cloud-of-things environments. *Computers and Security*, 74:340 – 354, 2018.

[43] Gang Sun, Victor Chang, Muthu Ramachandran, Zhili Sun, Gangmin Li, Hongfang Yu, and Dan Liao. Efficient location privacy algorithm for internet of things (iot) services and applications. *Journal of Network and Computer Applications*, 89:3 – 13, 2017. Emerging Services for Internet of Things (IoT).

[44] Chong-zhi Gao, Qiong Cheng, Xuan Li, and Shi-bing Xia. Cloud-assisted privacy-preserving profile-matching scheme under multiple keys in mobile social network. *Cluster Computing*, 22, 01 2019.

[45] Mehdi Gheisari, Quoc-Viet Pham, Mamoun Alazab, Xiaobo Zhang, Christian Fernández-Campusano, and Gautam Srivastava. Eca: An edge computing architecture for privacy-preserving in iot-based smart city. *IEEE Access*, 7:155779–155786, 2019.

[46] James Brogan, Immanuel Baskaran, and Navin Ramachandran. Authenticating health activity data using distributed ledger technologies. *Computational and Structural Biotechnology Journal*, 16:257 – 266, 2018.

[47] Jussi Kiljander, Alfredo D'Elia, Francesco Morandi, Pasi Hyttinen, Janne Takalo-Mattila, Arto Ylisaukko-oja, Juha-Pekka Soininen, and Tullio Cinotti. Semantic interoperability architecture for pervasive computing and internet of things. *Access, IEEE*, 2:856–873, 01 2014.

[48] Rajkumar Buyya, Chee Shin Yeo, Srikumar Venugopal, James Broberg, and Ivona Brandic. Cloud computing and emerging it platforms: Vision, hype, and reality for delivering computing as the 5th utility. *Future Generation Computer Systems*, 25(6):599 – 616, 2009.

[49] W. Tärneberg, V. Chandrasekaran, and M. Humphrey. Experiences creating a framework for smart traffic control using aws iot. In *2016 IEEE/ACM 9th International Conference on Utility and Cloud Computing (UCC)*, pages 63–69, Dec 2016.

[50] Sean Seniro, Chris Rec, Hitendra Nishar, and Tom Horton. Aws connected vehicle solution: Aws implemnation guide. 06 2017.

[51] Katsaros Konstantinos, Stevens Alan, Dianati Mehrdad, Han Chong, McCullough, Alexandros Mouzakitis, Maple C, and Fallah Saber. Cooperative automation through the cloud: The carma project. 06 2017.

[52] S. J. Stolfo, M. B. Salem, and A. D. Keromytis. Fog computing: Mitigating insider data theft attacks in the cloud. In *2012 IEEE Symposium on Security and Privacy Workshops*, pages 125–128, May 2012.

[53] Luis Alberto B. Pacheco, Eduardo Adílio Pelinson Alchieri, and Priscila América Solís Mendez Barreto. Device-based security to improve user privacy in the internet of things. In *Sensors*, 2018.

[54] Zhitao Guan, Jing Li, Longfei Wu, Yue Zhang, and Xiaojiang Du. Achieving efficient and secure data acquisition for cloud-supported internet of things in smart grid. *IEEE Internet of Things Journal*, PP:1–1, 04 2017.

[55] Manas Kumar Yogi, K. Chandrasekhar, and G. Vijay Kumar. Mist computing: Principles, trends and future direction. *ArXiv*, abs/1709.06927, 2017.

[56] T. Pratik, R. K. Lenka, G. K. Nayak, and A. Kumar. An architecture to support interoperability in iot devices. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 705–710, Oct 2018.

[57] Md. Asif Rahman, Fariha Afsana, Mufti Mahmud, M. Shamim Kaiser, Muhammad Ahmed, Omprakash Kaiwartya, and Anne James-Taylor. Towards a heterogeneous

mist, fog, and cloud based framework for the internet of healthcare things. *IEEE Internet of Things Journal*, PP:1–1, 10 2018.

[58] T. Pratik, R. K. Lenka, G. K. Nayak, and A. Kumar. An architecture to support interoperability in iot devices. In *2018 International Conference on Advances in Computing, Communication Control and Networking (ICACCCN)*, pages 705–710, Oct 2018.

[59] P. Battistoni, M. Sebillo, and G. Vitiello. Experimenting with a fog-computing architecture for indoor navigation. In *2019 Fourth International Conference on Fog and Mobile Edge Computing (FMEC)*, pages 161–165, June 2019.

[60] Pietro Battistoni, Monica Sebillo, and Giuliana Vitiello. Computation offloading with mqtt protocol on a fog-mist computing framework. 10 2019.

[61] Mohan Liyanage, Chii Chang, and Satish Srirama. Adaptive mobile web server framework for mist computing in the internet of things. *International Journal of Pervasive Computing and Communications*, 14, 11 2018.

[62] C. Esposito, A. Castiglione, F. Pop, and K. R. Choo. Challenges of connecting edge and cloud computing: A security and forensic perspective. *IEEE Cloud Computing*, 4(2):13–17, March 2017.

[63] Shanhe Yi, Cheng Li, and Qun Li. A survey of fog computing: Concepts, applications and issues. In *Proceedings of the 2015 Workshop on Mobile Big Data*, Mobidata 15, pages 37–42, New York, NY, USA, 2015. ACM.

[64] L. Gillam, K. Katsaros, M. Dianati, and A. Mouzakitis. Exploring edges for connected and autonomous driving. In *IEEE INFOCOM 2018 - IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)*, pages 148–153, April 2018.

[65] Quang Duy La, Mao V. Ngo, Thinh Quang Dinh, Tony Q.S. Quek, and Hyundong Shin. Enabling intelligence in fog computing to achieve energy and latency reduction.

Digital Communications and Networks, 5(1):3 – 9, 2019. Artificial Intelligence for Future Wireless Communications and Networking.

[66] K. Zhang and H. Jacobsen. Towards dependable, scalable, and pervasive distributed ledgers with blockchains. In *2018 IEEE 38th International Conference on Distributed Computing Systems (ICDCS)*, pages 1337–1346, July 2018.

[67] Shi-Cho Cha, Jyun-Fu Chen, Chunhua Su, and Kuo-Hui Yeh. A blockchain connected gateway for ble-based devices in the internet of things. *IEEE Access*, PP:1–1, 01 2018.

[68] O. Novo. Blockchain meets iot: An architecture for scalable access management in iot. *IEEE Internet of Things Journal*, 5(2):1184–1195, April 2018.

[69] A. Dorri, S. S. Kanhere, and R. Jurdak. Towards an optimized blockchain for iot. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 173–178, April 2017.

[70] C. Fan, H. Khazaei, Y. Chen, and P. Musilek. Towards a scalable dag-based distributed ledger for smart communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 177–182, April 2019.

[71] Amir Abbaszadeh Sori, Mehdi Golsorkhtabaramiri, and Amir Masoud Rahmani. Cryptocurrency grade of green; iota energy consumption modeling and measurement. In *2020 IEEE Green Technologies Conference(GreenTech)*, pages 80–82, 2020.

[72] Umair Sarfraz, Sherali Zeadally, and Muhammad Alam. Outsourcing iota proof-of-work to volunteer public devices. *Security and Privacy*, 3:e98, 12 2019.

[73] Mohd Majid Akhtar, Danish Raza Rizvi, Mohd Abdul Ahad, Salil S. Kanhere, Mohammad Amjad, and Giuseppe Coviello. Efficient data communication using distributed ledger technology and iota-enabled internet of things for a future machine-to-machine economy. *Sensors*, 21(13), 2021.

[74] Sehrish Shafeeq, Sherali Zeadally, Masoom Alam, and Abid Khan. Curbing address reuse in the iota distributed ledger: A cuckoo-filter-based approach. *IEEE Transactions on Engineering Management*, 67(4):1244–1255, 2020.

[75] Sehrish Shafeeq, Masoom Alam, and Abid Khan. Privacy aware decentralized access control system. *Future Generation Computer Systems*, 101:420–433, 2019.

[76] Odysseas Lamtzidis, Dionisis Pettas, and John Gialelis. A novel combination of distributed ledger technologies on internet of things: Use case on precision agriculture. *Applied System Innovation*, 2:30, 09 2019.

[77] Michael Vögler, Johannes Schleicher, Christian Inzinger, Stefan Nastic, Sanjin Sehic, and Schahram Dustdar. Leonore – large-scale provisioning of resource-constrained iot deployments. 03 2015.

[78] Kashif Dar, Amir Taherkordi, Harun Baraki, Frank Eliassen, and Kurt Geihs. A resource oriented integration architecture for the internet of things: A business process perspective. *Pervasive and Mobile Computing*, 20:145 – 159, 2015.

[79] Onoriode Uviase and Gerald Kotonya. Iot architectural framework: Connection and integration framework for iot systems. In *ALP4IoT@iFM*, 2017.

[80] David C Klonoff. Fog computing and edge computing architectures for processing data from diabetes devices connected to the medical internet of things, 2017.

[81] Y. Sahni, J. Cao, S. Zhang, and L. Yang. Edge mesh: A new paradigm to enable distributed intelligence in internet of things. *IEEE Access*, 5:16441–16458, 2017.

[82] Hasibur Rahman and Rahim Rahmani. Enabling distributed intelligence assisted future internet of things controller (fitc). *Applied Computing and Informatics*, 14(1):73 – 87, 2018.

[83] M. Aazam and E. Huh. Dynamic resource provisioning through fog micro datacenter. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 105–110, March 2015.

[84] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung. Developing iot applications in the fog: A distributed dataflow approach. In *2015 5th International Conference on the Internet of Things (IOT)*, pages 155–162, Oct 2015.

[85] Ammar Muthanna, Abdelhamied A Ateya, Abdukodir Khakimov, Irina Gudkova, Abdelrahman Abuarqoub, Konstantin Samouylov, and Andrey Koucheryavy. Secure iot network structure based on distributed fog computing, with sdn/blockchain. 2019.

[86] Minh-Quang Tran, Duy Tai Nguyen, Van An Le, Duc Hai Nguyen, and Tran Vu Pham. Task placement on fog computing made efficient for iot application provision. *Wireless Communications and Mobile Computing*, 2019, 2019.

[87] C. Sarkar, A. U. Nambi S. N., R. V. Prasad, A. Rahim, R. Neisse, and G. Baldini. Diat: A scalable distributed architecture for iot. *IEEE Internet of Things Journal*, 2(3):230–239, June 2015.

[88] Higinio Mora, Maria Teresa Pont, David Gil, and Magnus Johnsson. Collaborative working architecture for iot-based applications. *Sensors*, 18:1676, 05 2018.

[89] Hasibur Rahman, Rahim Rahmani, and Theo Kanter. *The Role of Mobile Edge Computing Towards Assisting IoT with Distributed Intelligence: A SmartLiving Perspective*, pages 33–45. Springer International Publishing, Cham, 2019.

[90] Bo Tang, Zhen Chen, Gerald Hefferman, Tao Wei, Haibo He, and Qing Yang. A hierarchical distributed fog computing architecture for big data analysis in smart cities. In *Proceedings of the ASE BigData & SocialInformatics 2015*, ASE BD&SI '15, pages 28:1–28:6, New York, NY, USA, 2015. ACM.

[91] U. Bellur, P. Patel, S. Chauhan, and Y. Qin. A semantic-enabled framework for future internet of things applications. In *2017 IEEE World Congress on Services (SERVICES)*, pages 106–113, 2017.

[92] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani. Applications of blockchains in the internet of things: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 21(2):1676–1717, Secondquarter 2019.

[93] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014.

[94] Erikson Júlio De Aguiar, Bruno S. Faiçal, Bhaskar Krishnamachari, and Jó Ueyama. A survey of blockchain-based strategies for healthcare. *ACM Comput. Surv.*, 53(2), March 2020.

[95] Claus Pahl, Nabil El Ioini, and Sven Helmer. A decision framework for blockchain platforms for iot and edge computing. 03 2018.

[96] IOTA Foundation. Iota development roadmap. (1), Dec 2016. (visited on 2-01-2019).

[97] Pietro Danzi, Anders E. Kalør, René B. Sørensen, Alexander K. Hagelskjær, Lam D. Nguyen, Čedomir Stefanović, and Petar Popovski. Communication aspects of the integration of wireless iot devices with distributed ledger technology, 2019.

[98] Chen Min, Kwon Taekyoung, Yong Yuan, and Victor Leung. Mobile agent based wireless sensor networks. *Journal of Computers*, 1, 04 2006.

[99] Daniel Massaguer, Chien-Liang Fok, Nalini Venkatasubramanian, Gruia-Catalin Roman, and Chenyang Lu. Exploring sensor networks using mobile agents. In *Proceedings of the Fifth International Joint Conference on Autonomous Agents and Multiagent Systems*, AAMAS '06, pages 323–325, New York, NY, USA, 2006. ACM.

[100] Danny B. Lange and Mitsuru Oshima. Seven good reasons for mobile agents. *Commun. ACM*, 42(3):88–89, March 1999.

[101] Mohamed El Fissaoui, Abderrahim beni hssane, Slimane Ouhmad, and Khalid El Makkaoui. A survey on mobile agent itinerary planning for information fusion in

wireless sensor networks. *Archives of Computational Methods in Engineering*, 03 2020.

[102] Hairong Qi and Feiyi Wang. Optimal itinerary analysis for mobile agents in ad hoc wireless sensor networks. 02 2004.

[103] Bo Liu, Jiuxin Cao, Jie Yin, Wei Yu, Benyuan Liu, and Xinwen Fu. Disjoint multi mobile agent itinerary planning for big data analytics. *EURASIP Journal on Wireless Communications and Networking*, 2016, 12 2016.

[104] Junfeng Wang, Yin Zhang, Zhuanli Cheng, and Xuan Zhu. Emip: energy-efficient itinerary planning for multiple mobile agents in wireless sensor network. *Telecommunication Systems*, 62, 03 2015.

[105] Yu-Cheng Chou and Madoka Nakajima. A clonal selection algorithm for energy-efficient mobile agent itinerary planning in wireless sensor networks. *Mobile Networks and Applications*, 23, 01 2017.

[106] Chen Min, Kwon Taekyoung, Yong Yuan, and Choi Yanghee. Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007, 01 2007.

[107] Fei Jiang, Haoshan Shi, Zhiyan Xu, and Xiangjun Dong. Improved directed diffusion-based mobile agent mechanism for wireless sensor networks. *2009 4th International Conference on Communications and Networking in China, CHINACOM 2009*, 08 2009.

[108] Damianos Gavalas, Ioannis Venetis, Charalampos Konstantopoulos, and Grammati Pantziou. Energy-efficient multiple itinerary planning for mobile agents-based data aggregation in wsns. *Telecommunication Systems*, 63, 02 2016.

[109] Mianxiong Dong, Kaoru Ota, Laurence T. Yang, Shan Chang, Hongzi Zhu, and Zhenyu Zhou. Mobile agent-based energy-aware and user-centric data collection in

wireless sensor networks. *Computer Networks*, 74:58 – 70, 2014. Special Issue on Mobile Computing for Content/Service-Oriented Networking Architecture.

[110] Min Chen, Laurence Yang, Ted Kwon, Liang Zhou, and Minho Jo. Itinerary planning for energy-efficient agent communications in wireless sensor networks. *Vehicular Technology, IEEE Transactions on*, 60:3290 – 3299, 10 2011.

[111] Chase Wu, Nageswara Rao, Jacob Barhen, Sundararaj Iyengar, Vijay Vaishnavi, Hairong Qi, and Krishnendu Chakrabarty. On computing mobile agent routes for data fusion in distributed sensor networks. *Knowledge and Data Engineering, IEEE Transactions on*, 16:740 – 753, 07 2004.

[112] Imene Aloui, Okba Kazar, Laid Kahloul, Azeddine Aissaoui, and sylvie servigne. A new "data size" based algorithm for itinerary planning among mobile agents in wireless sensor networks. In *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, BDAW '16, pages 36:1–36:9, New York, NY, USA, 2016. ACM.

[113] Min Chen and Sergio Gonzalez-Valenzuela. Directional source grouping for multi-agent itinerary planning in wireless sensor networks. pages 207 – 212, 12 2010.

[114] Mohamed El Fissaoui, Abderrahim beni hssane, and Mostafa Saadi. Multi-mobile agent itinerary planning-based energy and fault aware data aggregation in wireless sensor networks. *EURASIP Journal on Wireless Communications and Networking*, 2018:92, 12 2018.

[115] Govind Gupta, Manoj Misra, and Kumkum Garg. Towards scalable and load-balanced mobile agents-based data aggregation for wireless sensor networks. *Computers and Electrical Engineering*, 64, 10 2017.

[116] O. S. Egwuche, O. S. Adewale, S. A. Oluwadare, and O. A. Daramola. Enhancing network life-time of wireless sensor networks through itinerary definition and mobile agents for routing among sensor nodes. In *2020 International Conference in*

Mathematics, Computer Engineering and Computer Science (ICMCECS), pages 1–7, 2020.

[117] Govind Gupta, Manoj Misra, and Kumkum Garg. Energy and trust aware mobile agent migration protocol for data aggregation in wireless sensor networks. *Journal of Network and Computer Applications*, 41, 05 2014.

[118] Milos Grujic, Vladimir Rozic, David Johnston, John Kelsey, and Ingrid Verbauwhede. Invited: Design principles for true random number generators for security applications. In *2019 56th ACM/IEEE Design Automation Conference (DAC)*, pages 1–3, 2019.

[119] Carson Labrado and Himanshu Thapliyal. Hardware security primitives for vehicles. *IEEE Consumer Electron. Mag.*, 8(6):99–103, 2019.

[120] Carson Labrado, S. Dinesh Kumar, Riasad Badhan, Himanshu Thapliyal, and Vijay Singh. Exploration of solar cell materials for developing novel pufs in cyber-physical systems. *SN Comput. Sci.*, 1(6):313, 2020.

[121] Hao Jiang, Daniel Belkin, Sergey Savel'ev, Siyan Lin, Zhongrui Wang, Yunning Li, Saumil Joshi, Rivu Midya, Can Li, Mingyi Rao, Mark Barnell, Qing wu, Jianhua Joshua Yang, and Qiangfei Xia. A novel true random number generator based on a stochastic diffusive memristor. *Nature Communications*, 8, 10 2017.

[122] Amit Degada and Himanshu Thapliyal. An integrated trng-puf architecture based on photovoltaic solar cells. *IEEE Consumer Electronics Magazine*, pages 1–1, 2020.

[123] Carson Labrado, Himanshu Thapliyal, Stacy J. Prowell, and P. Teja Kuruganti. Use of thermistor temperature sensors for cyber-physical system security. *Sensors*, 19(18):3905, 2019.

[124] Carson Labrado and Himanshu Thapliyal. Design of a piezoelectric-based physically unclonable function for iot security. *IEEE Internet Things J.*, 6(2):2770–2777, 2019.

[125] D. Zhao, J. Ren, R. Lin, S. Xu, and V. Chang. On orchestrating service function chains in 5g mobile network. *IEEE Access*, 7:39402–39416, 2019.

[126] Talal Noor, Sherali Zeadally, Abdullah Alfazi, and Quan Sheng. Mobile cloud computing: Challenges and future research directions. *Journal of Network and Computer Applications*, 115, 05 2018.

[127] Ioannis E. Venetis, Damianos Gavalas, Grammati E. Pantziou, and Charalampos Konstantopoulos. Mobile agents-based data aggregation in wsns: Benchmarking itinerary planning approaches. *Wirel. Netw.*, 24(6):2111–2132, August 2018.

[128] Tariq Alsboui, Mustafa Alrifaee, Rami Etaywi, and Mohammad Abdul Jawad. Mobile agent itinerary planning approaches in wireless sensor networks- state of the art and current challenges. In Leandros A. Maglaras, Helge Janicke, and Kevin Jones, editors, *Industrial Networks and Intelligent Systems*, pages 143–153, Cham, 2017. Springer International Publishing.

[129] Michel Rauchs, Andrew Glidden, Brian Gordon, Gina Pieters, Martino Recanatini, François Rostand, Kathryn Vagneur, and Bryan Zhang. Distributed ledger technology systems: A conceptual framework. *SSRN Electronic Journal*, 01 2018.

[130] Tariq Alsboui, Mustafa Alrifaee, Rami Etaywi, and Mohammad Abdul Jawad. Mobile agent itinerary planning approaches in wireless sensor networks- state of the art and current challenges. In Leandros A. Maglaras, Helge Janicke, and Kevin Jones, editors, *Industrial Networks and Intelligent Systems*, pages 143–153, Cham, 2017. Springer International Publishing.

[131] C. Fan, H. Khazaei, Y. Chen, and P. Musilek. Towards a scalable dag-based distributed ledger for smart communities. In *2019 IEEE 5th World Forum on Internet of Things (WF-IoT)*, pages 177–182, 2019.

[132] Popov Serguei. Iota distributed ledger technology. (1), October 2015.

[133] James Brogan, Immanuel Baskaran, and Navin Ramachandran. Authenticating health activity data using distributed ledger technologies. *Computational and Structural Biotechnology Journal*, 16:257 – 266, 2018.

[134] Mudassir Ali, Adeel Anjum, Adnan Anjum, and Muazzam A. Khan. Efficient and secure energy trading in internet of electric vehicles using iota blockchain. In *2020 IEEE 17th International Conference on Smart Communities: Improving Quality of Life Using ICT, IoT and AI (HONET)*, pages 87–91, 2020.

[135] Satoshi Nakamoto et al. Bitcoin: A peer-to-peer electronic cash system. 2008.

[136] Popov Serguei. The tangle. (1), October 2017.

[137] Nabil El Ioini and Claus Pahl. A review of distributed ledger technologies. In Hervé Panetto, Christophe Debruyne, Henderik A. Proper, Claudio Agostino Ardagna, Dumitru Roman, and Robert Meersman, editors, *On the Move to Meaningful Internet Systems. OTM 2018 Conferences*, pages 277–288, Cham, 2018. Springer International Publishing.

[138] Andreas M. Antonopoulos. *Mastering Bitcoin: Unlocking Digital Crypto-Currencies*. O'Reilly Media, Inc., 1st edition, 2014.

[139] B. Cao, Y. Li, L. Zhang, L. Zhang, S. Mumtaz, Z. Zhou, and M. Peng. When internet of things meets blockchain: Challenges in distributed consensus. *IEEE Network*, pages 1–7, 2019.

[140] M. S. Ali, M. Vecchio, M. Pincheira, K. Dolui, F. Antonelli, and M. H. Rehmani. Applications of blockchains in the internet of things: A comprehensive survey. *IEEE Communications Surveys Tutorials*, 21(2):1676–1717, Secondquarter 2019.

[141] I-SCOOP. Blockchain and the internet of things: the iot blockchain opportunity and challenge. (1), Feb 2018. (visited on 19-09-2019).

[142] A. Dorri, S. S. Kanhere, and R. Jurdak. Towards an optimized blockchain for iot. In *2017 IEEE/ACM Second International Conference on Internet-of-Things Design and Implementation (IoTDI)*, pages 173–178, April 2017.

[143] K. Christidis and M. Devetsikiotis. Blockchains and smart contracts for the internet of things. *IEEE Access*, 4:2292–2303, 2016.

[144] K. Biswas and V. Muthukkumarasamy. Securing smart cities using blockchain technology. In *2016 IEEE 18th International Conference on High Performance Computing and Communications; IEEE 14th International Conference on Smart City; IEEE 2nd International Conference on Data Science and Systems (HPCC/SmartCity/DSS)*, pages 1392–1393, Dec 2016.

[145] IOTA Foundation. The anatomy of a transaction. Feb 2018. (visited on 12-02-2020).

[146] IOTA Foundation. Overview of iota streams. (1), Jan 2020. (visited on 3-12-2021).

[147] Paul Handy. Introducing masked authenticated messaging. (1), Nov 2017. (visited on 6-01-2019).

[148] IOTA Foundation. The qubic protocol. (1), Dec 2016. (visited on 2-1-2019).

[149] IOTA. An introduction to iota smart contracts. (1), May 2020. (visited on 16-09-2021).

[150] Kristy Yau and Nagaveni Biradar. Iota-next generation block chain. *International Journal Of Engineering And Computer Science*, 7:23823–23826, 04 2018.

[151] Hans Moog Serguei Popov. The coordicide. (1), Jan 2020. (visited on 16-09-2021).

[152] Stephan Wuffli. Smart parking systems. (1), December 2015.

[153] T. Lin, H. Rivano, and F. Le Mouël. A survey of smart parking solutions. *IEEE Transactions on Intelligent Transportation Systems*, 18(12):3229–3253, 2017.

[154] Xin Dong, Xiangjie Kong, Fulin Zhang, Zhen Chen, and Jialiang Kang. Oncampus: a mobile platform towards a smart campus. *SpringerPlus*, 5(1):974, Jul 2016.

[155] Katsaros Konstantinos, Stevens Alan, Dianati Mehrdad, Han Chong, McCullough, Mouzakitis Alexandros, Maple C, and Fallah Saber. Cooperative automation through the cloud: The carma project. 06 2017.

[156] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014.

[157] M. Aazam and E. Huh. E-hamc: Leveraging fog computing for emergency alert service. In *2015 IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, pages 518–523, March 2015.

[158] C. Perera, A. Zaslavsky, P. Christen, and D. Georgakopoulos. Context aware computing for the internet of things: A survey. *IEEE Communications Surveys Tutorials*, 16(1):414–454, First 2014.

[159] Tariq Alsboui, Abdelrahman Abuarqoub, Mohammad Hammoudeh, Zuhair Bandar, and Andy Nisbet. Information extraction from wireless sensor networks: System and approaches. *Sensors & Transducers*, 14(2):1, 2012.

[160] Tariq A. A. Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. Towards a scalable IOTA tangle-based distributed intelligence approach for the internet of things. In Kohei Arai, Supriya Kapoor, and Rahul Bhatia, editors, *Intelligent Computing - Proceedings of the 2020 Computing Conference, Volume 2, AI 2020, London, UK, 16-17 July 2020*, volume 1229 of *Advances in Intelligent Systems and Computing*, pages 487–501. Springer, 2020.

[161] Gartner. Gartner says the internet of things installed base will grow to 26 billion units by 2020. (1), December 2013.

[162] API Research. More than 30 billion devices will wirelessly connect to the internet of everything in 2020. (1), May 2013.

[163] Tariq A. A. Alsboui, Yongrui Qin, and Richard Hill. Enabling distributed intelligence in the internet of things using the iota tangle architecture. In *IoTBDS*, 2019.

[164] Parker Lynne. Distributed intelligence: Overview of the field and its application in multi-robot systems. In *The AAAI Fall Symposium Series*. AAAI Digital Library, 2007.

[165] Tariq Alsboui, Mustafa Alrifaee, Rami Etaywi, and Mohammad Abdul Jawad. Mobile agent itinerary planning approaches in wireless sensor networks- state of the art and

current challenges. In Leandros A. Maglaras, Helge Janicke, and Kevin Jones, editors, *Industrial Networks and Intelligent Systems*, pages 143–153, Cham, 2017. Springer International Publishing.

[166] Kai Peng, Victor Leung, Xiaolong Xu, Lixin Zheng, Jiabin Wang, and Qingjia Huang. A survey on mobile edge computing: Focusing on service adoption and provision. *Wireless Communications and Mobile Computing*, 2018, 10 2018.

[167] Yu-Chee Tseng, Sheng-Po Kuo, Hung-Wei Lee, and Chi-Fu Huang. Location tracking in a wireless sensor network by mobile agents and its data fusion strategies. In Feng Zhao and Leonidas Guibas, editors, *Information Processing in Sensor Networks*, pages 625–641, Berlin, Heidelberg, 2003. Springer Berlin Heidelberg.

[168] Min Chen, Taekyoung Kwon, Yong Yuan, Yanghee Choi, and Victor C.M. Leung. Mobile agent-based directed diffusion in wireless sensor networks. *EURASIP Journal on Advances in Signal Processing*, 2007(1):036871, Dec 2006.

[169] G. J. Pottie and W. J. Kaiser. Wireless integrated network sensors. *Commun. ACM*, 43(5):51–58, May 2000.

[170] IOTA Foundation. Pyota: The iota python api library. (1), Feb 2018. (visited on 8-08-2019).

[171] Tariq Alsboui, Yongrui Qin, Richard Hill, and Hussain Al-Aqrabi. An energy efficient multi-mobile agent itinerary planning approach in wireless sensor networks. *Computing*, 103(9):2093–2113, 2021.

[172] Tariq Alsbou'i, Abdelrahman Abuarqoub, Mohammad Hammoudeh, Zuhair Bandar, and Andy Nisbet. Information extraction from wireless sensor networks: System and approaches. *Sensors and Transducers*, 14:1–17, 03 2012.

[173] Chase Wu, Nageswara Rao, Jacob Barhen, Sundararaj Iyengar, Vijay Vaishnavi, Hairong Qi, and Krishnendu Chakrabarty. On computing mobile agent routes for

data fusion in distributed sensor networks. *Knowledge and Data Engineering, IEEE Transactions on*, 16:740 – 753, 07 2004.

[174] G. Chen, S. Wu, J. Zhou, and A. K. H. Tung. Automatic itinerary planning for traveling services. *IEEE Transactions on Knowledge and Data Engineering*, 26(3):514–527, 2014.

[175] Damir Arbula and Kristijan Lenac. Pymote: High level python library for event-based simulation and evaluation of distributed algorithms. *International Journal of Distributed Sensor Networks*, 9(3):797354, 2013.

[176] Mostefa Kara, Abdelkader Laouid, Muath AlShaikh, Mohammad Hammoudeh, Ahcene Bounceur, Reinhardt Euler, Abdelfattah Amamra, and Brahim Laouid. A compute and wait in pow (cw-pow) consensus algorithm for preserving energy consumption. *Applied Sciences*, 11(15), 2021.

[177] ChaoZhou, afialho, and Felandil. Untangle care. (1), October 2019.

[178] Seny Kamara, Charalampos Papamanthou, and Tom Roeder. Dynamic searchable symmetric encryption. In *Proceedings of the 2012 ACM Conference on Computer and Communications Security*, CCS '12, page 965–976, New York, NY, USA, 2012. Association for Computing Machinery.

[179] Mohammad Hammoudeh and Robert Newman. Adaptive routing in wireless sensor networks: Qos optimisation for enhanced application performance. *Information Fusion*, 22:3–15, 2015.

[180] Luca Mottola and Gian Pietro Picco. Programming wireless sensor networks: Fundamental concepts and state of the art. *ACM Comput. Surv.*, 43(3):19:1–19:51, April 2011.

[181] George A. Papadopoulos and Farhad Arbab. Coordination models and languages. volume 46 of *Advances in Computers*, pages 329 – 400. Elsevier, 1998.

[182] Paul Klimos. The distributed ledger technology: a potential revamp for financial markets? *Capital Markets Law Journal*, 13(2):194–222, 03 2018.

[183] Dharmendra Shadija, Mo Rezai, and Richard Hill. Microservices: Granularity vs. performance. In *UCC 2017 Companion - Companion Proceedings of the 10th International Conference on Utility and Cloud Computing*, pages 215–220. Association for Computing Machinery, Inc, 12 2017.