Universitat Autònoma de Barcelona

# MULTI-CHORD P2P APPROACH FOR INTEGRATED SATELLITE IOT NETWORKS

**Ahmed Ismail Abdelghafar Ismail**

Universitat Autònoma de Barcelona

# MULTI-CHORD P2P APPROACH FOR INTEGRATED SATELLITE IOT NETWORKS

Author
Ahmed Ismail Abdelghafar Ismail

Co-Directors
Dr. Ángeles Vázquez-Castro
Dr. Mohamed Essam Mahoud Khedr

*A Thesis submitted in fulfillment of the requirements for the degree*

*of*

*Doctor of Philosophy in Electronic and Telecommunication Engineering*

*in the*

*Department of Electronic and Telecommunication Engineering*

*September 2022*

# Abstract

This research focuses on real-time data dependent applications and the urgent need for novel and creative architecture of Internet of Things (IoT) that will extend the fog computing further to be more peer-to-peer (P2P) approach.

Mainly this research identifies two primary challenges:

1. An architecture that supports IoT, and is characterized by low delay, small complexity and high robustness and resilience.
2. Reliable data handling algorithms.

We classify the objectives of our research into three main objectives:

**Objective 1**: develop a generic overlay architecture for peer-peer IoT networks. The architecture should be able to accommodate a large variety of nodes as well as to be able to handle the generated data that is characterized by being high in volume and highly sensitive to time. This first objective is crystalized around the concept of peer-to-peer (P2P) networking and specifically the chord approach. The proposed overlay architecture is called multi-chord peer to peer IoT as it extended the standard chord to be multiple chords that could logically correspond to a group of nodes in a terrestrial network or to a group of satellites in a constellation in space or a mix of both. The multi-chord approach will give IoT the required characteristics of being robust, scalable, resilient and being self-maintaining and on a large scale as well as handling a vast data type.

**Objective 2**: perform the mathematical analysis and formulation of the different algorithms and operations carried on multi-chord architecture. This includes the algorithms of data lookup and manipulation, node joining and leaving as well as the network stabilization process in case of failure of changes in the network topology. The mathematical model is based on discrete Markov model as all operations in multi-chord are probabilistic in nature and depends on the current and next/previous state. This mathematical model will allow us to theoretically model the performance of the multi-chord and find the optimum situation for minimizing the cost of the operation as well as modeling the complexity.

**Objective 3**: evaluate the performance of the multi-chord architecture by providing in-depth analysis of the achievable capacity and the robustness and resilience of the multi-chord in face of multiple of success failures and to provide performance improvements over state-of-the art cloud and/or fog based IoT networks.

Our new proposed architecture based on multi-chord, requires algorithms that will insure the accurate and correct handling of data in the IoT as well as the simplicity of controlling the operation in the multi-chord architecture. This led to three main contributions in this research:

1. The first contribution is the development of a new multi-chord IoT architecture for the IoT networks based on multi-chord P2P overlay topology.
2. The second contribution is the extension of the existing chord functions and algorithms of stabilization and resilience to the multi-chord IoT architecture.
3. The third contribution is the development of a mathematical model and performance evaluation of the multi-chord architecture in terms of its appropriateness of integration of terrestrial and satellite networks. We obtain enhancements with respect to the state of the art of up 20% in the meantime to stabilization. Also, scalability is shown to be enhanced to 30% while the mean of lookup time showed 8% enhancement (while increasing the number of nodes with respect to the state-of-the art of P2P single-cord architectures). The variance of search delay is also enhanced by 2%.

# Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof. Angeles Vazquez Castro and Prof. Mohamed Essam Khedr, for their continuous effort throughout my PhD study. Their guidance and support are true essence of the work ahead in this dissertation.

I would also like to thank my wife, Dina Nakhla, who has always been by my side in both study and life.

My deepest appreciation goes to my parents and brothers. They have always got my back.

And last but not least, this dissertation is dedicated to my beloved son, Yassin, who is my ultimate inspiration.

# Table of Contents

# List of Figures

# List of Tables

# Glossary

| | |
|---|---|
| 6Lo | IPv6 over Networks of Resource Constrained Nodes |
| 6LoWPAN | IPv6 over Low power Wireless Personal Area Network |
| 6TiSCH | IPv6 over Time Slotted Channel Hopping Mode of IEEE 802.15.4e |
| AMQP | The Advanced Message Queuing Protocol |
| CAN | Content Addressable Network |
| CARP | Channel-Aware Routing Protocol |
| CID | Chord ID |
| CoAP | Constrained Application Protocol |
| CoRE | Constrained RESTful Environment |
| CORPL | Cognitive RPL |
| DDS | Data Distribution Service |
| DECT/ULE | Digital Enhanced Cordless Telephone with Ultra Low Energy |
| DHT | Distributed Hash Table |
| DTMC | Discrete Time Markov Chain |
| DtS | Direct to Satellite |
| FCC | Federal Communications Committee |
| FCD | Fog computing domains |
| Gbps | Gigabit per second |
| GEO | Geostationary Earth Orbit |
| ICT | Information and Communication Technologies |
| IEEE | Institute of Electrical and Electronics Engineers |
| IETF | Internet Engineering Task Force |
| IoT | Internet of Things |
| ITU-T | International Telecommunication Union – Telecommunications Standardization section |
| Kbps | Kilobits per second |
| KID | Key ID |
| KPI | Key Performance Indicator |
| LEO | Low Earth Orbit |
| LoRaWAN | Long Range Wide Area Network |
| LPWA | Low Power Wide Area |
| LSB | Least Significant Bits |
| LTE | Long-Term Evolution |
| LTE-A | Long-Term Evolution Advanced |
| MAC | Media Access Control |
| Mbps | Megabits per second |
| MC | Markov Chain |
| MLTS | Multilayer Tabu Search |
| MQTT | Message Queue Telemetry Transport |
| MSB | Most Significant Bits |
| NFC | Near Field Communication |
| NFC | Near Field Communication |
| NID | Node ID |

| | |
|---|---|
| P2P | Peer-to-Peer |
| QoE | Quality of Experience |
| QoS | Quality of Service |
| RFID | Radio Frequency Identification |
| RPL | Routing Protocol for Low-Power and Lossy Networks |
| RPL | Routing Protocol for Low-Power and Lossy Networks |
| RTT | Round Trip Time |
| SCID | Successor Chord ID |
| SHA | Secure Hash Algorithm |
| SIN | Space Information Network |
| SIoT | Satellite Internet of Things |
| SMQTT | Secure MQTT |
| TEG | Time-Evolving Graph |
| WiFi | Wireless Fidelity |
| WSN | Wireless Sensor Network |
| XMPP | Extensible Messaging and Presence Protocol |

# Chapter 1

## 1. Introduction

This chapter provides an introduction and background knowledge of Internet of Things (IoT) and its integration to large scale Low Earth Orbit (LEO) satellite constellation. The motivation and applications of Satellite Internet of Things (SIoT) is highlighted, which inspires this research work. A description of the contributions and up-to-date publications during the author's PhD study is also provided.

The recent development of communication devices and wireless network technologies continues to advance the new era of the Internet and telecommunications. The various "things", which include not only communication devices but also every other physical object on the planet, are also going to be connected to the Internet, and controlled through wireless networks. This concept, which is referred to as the "Internet of Things (IoT)", has attracted much attention from many researchers in recent years, which is considered as era of IoT. IoT enables interconnection of anything, anytime and anywhere [1]. International Telecommunication Union – Telecommunications Standardization section (ITU-T) defines IoT as "global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable Information and Communication Technologies (ICT)." [2]

The IoT has been defined by different authors in many ways. Vermesan et al. [3] define the IoT as simply an interaction between the physical and digital worlds. The digital world interacts with the physical world using a plethora of sensors and actuators. Another definition by Pe˜na-L´opez et al. [4] defines the IoT as a paradigm in which computing, and networking capabilities are embedded in any kind of conceivable object.

IoT is also considered as the main building block of the fourth industrial revolution, known as Industry 4.0 [5,6], as it integrates the Internet of Things (IoT), cloud computing, big data and artificial intelligence to enable transition from automated manufacturing toward an intelligent manufacturing [7]. And one of the key drivers for next-generation mobile communications (5G) is IoT as well due to its support with billions of objects connected to the Internet and very low latency. The 5G technology will support the realization of smart cities, smart environments, and big data applications [8,9&10].

As stated in [11,12] there are 30 billion interconnected nodes, and 75 billion nodes are expected by 2025. Utilizing IoT connected devices can help in monitoring vital, restricted, unsafe places and processes [13].

Internet of Things (IoT) is a coin term recently used in Information and Communication Technology (ICT) research and industrial community to express the involvement of devices of different capabilities and functionalities in the daily activities of people and organizations. Mobile phones, wearable devices, sensors, appliances, and automobiles are examples of such devices in IoT environment while users' tasks, transportation systems, health care systems, smart digital cities and educational systems are examples of applications where IoT will be used. With such vast highly dynamic environment, challenges rise on methods and techniques than will be used to discover, connect, organize such devices from one side and mechanisms to collect, aggregate, filter, process, store and retrieve generated data from other side.

IoT platforms, access technologies, data storage & processing, data analytics, and security [14], these five aspects are growth enabler for future IoT networks. The IoT platform consists of hardware, software, and connectivity [15,16]. The objective of data analytics is to use the data to realize trends and results that can help develop business efficiently and make a positive impact on the market. IoT security can ensure the normal working of all the functionalities in an operational system.

Now IoT has several dimensions for many applications, and it facilitates massive machine type communications. IoT can reach to the remote areas where there is no infrastructure available. Infrastructure is normally available where there are human communities. However, there are many scenarios where we need the information without the presence of the infrastructure. For example, measurements of under surface soil humidity and fertilizer intensity can be brought into the Internet through the IoT. This can be done by deploying appropriate sensors under the surface. Similarly, there are applications in which micro monitoring is required in the territories. In those cases, the supervision of the IoT can be done through some remote communication nodes.

Satellites and high altitude-based communication devices are helpful in these applications. Terrestrial technologies have a common weakness which is they fail to provide global connectivity. Terrestrial networks are also highly vulnerable in case of natural disaster or terrorist attack. Instead, large scale Low Earth Orbit (LEO) satellite constellations have shown their potential to extend terrestrial networks. In this context satellites can be leveraged to support a worldwide expansion of the promising IoT market such as Starlink, Iridium and Orbcomm [17].

LEO satellites can be characterized by an altitude between 160 km to 2000 km which provides a coverage on earth of hundreds of kilometers and less than 100 ms Round Trip Time (RTT) delay while in case of Geostationary Earth Orbit (GEO) the RTT is in order of 600 ms to 700 ms. Currently, the delay from terrestrial communication equipment to its visible Low earth orbit (LEO) satellites can be minimized to 1 to 4 ms, which can meet the user's Quality of Service (QoS) requirement [18].

Recently, the combination of satellite networks and Low Power Wide Area (LPWA) technologies is proposed as a promising hybrid network architecture [19]. In Fig. 1-1, the concept of satellite based IoT network is shown.

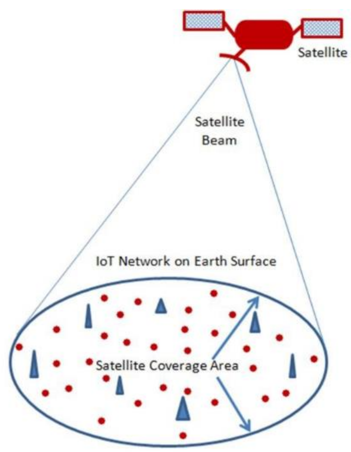



***Figure 1-1 Satellite based IoT network***

## 1.1. Motivation and Objectives

### 1.1.1. Motivation

With the increasing demand of Internet, mobile applications and things providing context awareness about surrounding environment, the current network infrastructure, data processing platforms and computing devices are facing challenges of limiting power and capabilities. This is due to factors such as rigid infrastructure of Internet that cannot cope with the high mobility and high versatility of networking technologies as well as scalability to cope with the high volume of data generated even with the usage of cloud computing and the increasing demand for ubiquitous computing in a global scale. Therefore, major challenges are required in the paradigm of networking to achieve efficient, scalable, reliable, and resilient architecture for the future of networking technologies such as IoT, blockchain and integration of heterogeneous architecture to a homogeneous overlay architecture.

Currently, Cloud computing and mobile networks [20] are used to provide the services required for IoT. This architecture depends on the approach that processing data and handling services are carried out in the cloud platform and the mobile network is for transmission and connectivity. However, this approach gets complicated and less efficient for many reasons:

1. Separation of communication and processing may cause high delay in response to user requests especially if heterogeneous communication networks are invoked like mobile and Satellite networks.
2. Scalability of this approach especially for time-sensitive applications like sensor data, actuators, and audio/video streaming.
3. Resilience of this approach and ability to continue to provide services in case of failure of interruption either in the network or in the computing platform.
4. Data integrity as data is highly dynamic and location of data storage and processing can affect the decision making depending on this data.

Currently most of the communication networks use cloud computing concept for IoT applications [21] where only the cloud nodes are responsible for the data processing. Other nodes in the network are used only for routing and capturing data. However, processing in the cloud domain only is not the optimal solution. It has been established recently that by employing edge nodes to perform some processing capability will achieve more reliability. This approach is referred as fog computing [*18*].

Fog computing extends cloud computing operations from being centralized at the cloud platform to gateways or fog devices that allow for data processing and manipulation at such nodes. Fog devices are middle links where the operational data is processed and stored by small data centers which can be reached at low latency. These middle links (fog devices) connect devices (things) with cloud nodes [22]. Fog computing provides the opportunity to enhance both the scalability and resilience in the existing IoT networks. In 2018, Institute of Electrical and Electronics Engineers (IEEE) adopted OpenFog [23] as the reference architecture for Fog Computing.

This dissertation focuses on real-time data dependent applications and the urgent need for novel and creative architecture of IoT that will extend the fog computing further to be more Peer-to-Peer (P2P) approach.

Mainly this research identifies two primary challenges:

1. Low delay, small complexity and high robustness and resilience for architecture that supports IoT. Although the state-of-the-art in terms of cloud and fog computing can provide higher reliability, but in general they involve higher delay, complexity, and center of failure

concept. Therefore, the IoT networks should be designed by considering these factors to provide robust, scalable, and resilient solution.

2. Reliable data handling algorithms. The current state-of-the-art solutions is based on structured relation-based data handling which is often do not take into consideration the constraints imposed by the big data generated and the unstructured characteristics of this data. Therefore, data handling algorithms should be designed in an innovative and creative way to enhance access to the data and at the same time can allow for accurate data manipulation in the existence of node failure and in the existence of large volume of data generated for the IoT.

### 1.1.2. Objectives

We classify the objectives of our research into three main objectives:

**Objective 1:** develop a generic overlay architecture for P2P IoT networks. The architecture should be able to accommodate large and vast types of nodes as well as to be able to handle the generated data that is characterized by being high in volume and highly sensitive to time. This first objective is crystalized around the concept of P2P networking and specifically the chord approach [24]. The proposed overlay architecture is called multi-chord P2P IoT as it extended the standard chord to be multiple chords that could be a group of nodes in a terrestrial network or a group of satellites in a constellation in space or a mix of both. The multi-chord approach will give IoT the required characteristics of being robust, scalable, resilient and being self-maintaining and on a large scale as well as vast data types.

**Objective 2:** perform the mathematical analysis and formulation of the different algorithms and operations carried on multi-chord architecture. This includes the algorithms of data lookup and manipulation, node joining and leaving as well as the network stabilization process in case of failure of changes in the network topology. The mathematical model is based on discrete markov model [25] as all operations in multi-chord are probabilistic in nature and depends on the current and next/previous state. This mathematical model will allow us to theoretically model the performance of the multi-chord and find the optimum situation for minimizing the cost of the operation in multi-chord as well as modeling the complexity of the multi-chord.

**Objective 3:** evaluate the performance of the multi-chord by providing in-depth analysis of the achievable capacity and the robustness and resilience of the multi-chord in face of multiple of success failures and to provide performance improvements over state-of-the art cloud and/or fog based IoT networks.

## 1.2. Research Approach

Our research follows a well-established work on terrestrial P2P networks, the chord approach, and states the required procedures to identify, analyze, adjust, and contribute to this P2P network that finally should provide solid and efficient terrestrial and satellite P2P networking schemes, the multi-chord approach, for IoT.

**To make advantage of Chord Peer-to-Peer (P2P) approach**: The chord P2P approach is a well-established approach that ensures correctness of data lookup and in an efficient manner. Chord algorithm provides efficient load balancing, decentralization, scalability, and availability. The synchronization mechanism also allows chord to adjust quickly under frequent node arrivals and failures/departures. The use of chord algorithm as the basis for multi-chord approach in IoT provides us algorithmic advantages for data manipulation and network stabilization such that extension to usage with satellite constellation in conjunction with terrestrial P2P network will be realistically achievable.
Accordingly, certain considerations are identified to focus on theoretical aspects of multi-chord while reflecting its applicability to realistic situations. Thus, several considerations have been set as follows:

### 1.2.1. Stability of Underlying Network:

Chord is an overlay approach. This means that there is some network topology that currently exists, such as Internet, and chord is implemented as an application layer algorithm over such topology. This will be used in multi-chord approach as well, but we will develop the process needed such that chord approach will be altered to run over satellite constellation already deployed and bridge with the chord running over the terrestrial network topology to form a global multi-chord overlay network.

### 1.2.2. Focus on One chord over Satellite Constellation and Two chords over terrestrial network:

Our main objective in this research is to provide the architecture of the multi-chord approach as well as the required algorithms for data lookup and network stabilization and resilience. However, we will show the appropriateness of multi-chord for the case of one chord in the satellite constellation with two chords in the terrestrial network. This represents the simplest deployment of multi-chord, but the approach can be systematically extended to any number of chords as the developed algorithms are generic.

### 1.2.3. Targeting Real-Time Data Generated from IoT Nodes:

The research targets only real-time data generated from IoT devices such as sensors, actuators, measuring devices and any other devices that is characterized by limited resources and generating data characterized by high volume, high variety and small. This impacts the design of multi-chord as it must be characterized with high reliability, high resilience, and fast response as well as smaller delay, complexity, and overhead.

## 1.3. Contribution of the Dissertation

We carried in this research a thorough study and investigation of architecture of IoT networks from being based on client-server model to cloud based approach and finally a fog-based approach. This gave us the insight to develop our innovative approach that is based on P2P networks first and on multi-chord extension to well-known chord approach [24].

This new approach –multi-chord approach- will require the algorithms that will insure the accurate and correct handling of data in the IoT as well as the simplicity of controlling the operation in multi-chord approach. This led to three main contributions in this research:

1. The first contribution lies in the development of a new architecture – multi-chord IoT - for the IoT networks based on P2P overlay topology.
2. The second contribution extends chord functions and algorithms of stabilization and resilience to the multi-chord IoT architecture
3. The third contribution focuses on developing the mathematical model and evaluating the performance of the multi-chord in terms of its appropriateness of integration of terrestrial and satellite networks.

As a summary of the research findings, table 1-1 shows the gain in the performance of multi-chord approach in comparison with the single chord approach.

| Indicator | Percentage of gain from Single Chord to Multi-chord |
|---|---|
| Mean time to Stabilization | 20% enhancement |
| Scalability | 30% enhancement |
| Mean of Lookup time | 8% enhancement |
| Variance of Search delay | 2% enhancement |

***Table 1-1 Gain of the Multi-chord Approach in Comparison with the Single Chord Approach***

In chapter 3 we present the architecture of the multi-chord approach for IoT networks which is a novel approach that changes the concept terrestrial as well as satellites networks follow which is merely relay of data over the network and satisfy the first objective of this research. However, upcoming satellites already have big storage capacity, which can be used to reduce the latency and complexity of operations to access and manipulate data. In addition, the integration of the satellites network with the terrestrial networks through the overlay approach of the multi-chord will assure the scalability, global coverage, reliability, and resilience that are highly required in IoT networks. For example, data that require certain real time operations may get processed and stored in the satellite nodes. Hence, to solve the problem of latency and complexity, the multi-chord concept would bring advantage to make logical networks that include some or all satellites in a constellation and this chapter presents the first and second contribution of this research.

Chapter 4 presents the mathematical model developed for multi-chord that shows the theoretical performance of multi-chord in different operation required such as data manipulation, node operations and network operations. This contribution is considered to the best of our knowledge is not clear in chord P2P networks and for sure in out proposed multi-chord network. Thus, this contribution serves the second objective and gives the formal method for evaluation of multi-chord IoT in terms of performance metrics, e.g.:

- logical sub-chords including satellites should have nice trade-off between energy consumed, access delay and failure recovery times.
- Hops/node for data lookup and data storage
- Hops/node for stabilization
- Percentage of failed lookups (robustness)
- Lookup latency

Chapter 5 highlights the third contribution of this research that emphasizes on the performance of IoT multi-chord in terms of organizing nodes in a P2P from the flat approach of the traditional chord to a multi-chord topology with hierarchal structure. Also, nodes in multi-chord IoT will be classified either based on their location or function. In addition, the accessibility mechanism is redesigned in multi-chord IoT to include beside finger table the chord table and multi-successor node tables also which will affect the multi-chord stabilization method. These operations will be evaluated through in-depth simulation to proof the applicability of the multi-chord networks and satisfy the third objective.

## 1.4. Publications

During the PhD study, the following publications have been published:

1. *Abdel Ghafar, A. I., Castro, Á. V., & Khedr, M. E. (2019). Multidimensional Self-Organizing Chord-Based Networking for Internet of Things. 2nd Europe - Middle East - North African Regional Conference of the International Telecommunications Society (ITS): "Leveraging Technologies For Growth", Aswan, Egypt, 18th-21st February, 2019.*

2. *I. Abdel Ghafar, Á. V. Castro and M. E. Khedr, "Satellite IoT services Using Multichord Peer to Peer Networking," 2019 IEEE 2nd 5G World Forum (5GWF), 2019, pp. 566-571, doi: 10.1109/5GWF.2019.8911672.*
3. *I. Abdel Ghafar, A. Vazquez-Castro and M. E. Khedr, "Resilience Analysis of Multichord Peer to Peer IoT Satellite Networks," 2021 23rd International Conference on Advanced Communication Technology (ICACT), 2021, pp. 220-225, doi: 10.23919/ICACT51234.2021.9370775.*

## 1.5. Outline of the Dissertation

The remainder of this dissertation is as follows:

Chapter 2 provides a comprehensive literature review on current IoT protocols and applications. The content in this chapter provides fundamental knowledge and inspiration of the rest of research in this dissertation.
Chapter 3 presents the architecture of the proposed multi-chord IoT as an overlay architecture with the associated protocols to enhance the performance of IoT by improving robustness, Responsiveness, scalability, and resilience.
Chapter 4 introduces the analysis and the mathematical model of the multi-chord IoT.
Chapter 5 presents in depth the performance evaluation of the multi-chord IoT using simulation software and demonstrate the performance based on realistic application scenario through the performance matrices.
Chapter 6 highlights the conclusion of this research with a perspective on the future work that can be extended over the multi-chord Satellite IoT.

# Chapter 2

## 2. Background and Literature Review

In this chapter we provide general background information that is relevant to describe the context within which this research has been conducted.

### 2.1. IoT Architecture

Three- and Five-Layer Architectures are the most basic architectures proposed by many researchers [26] as shown in Fig. 2-1

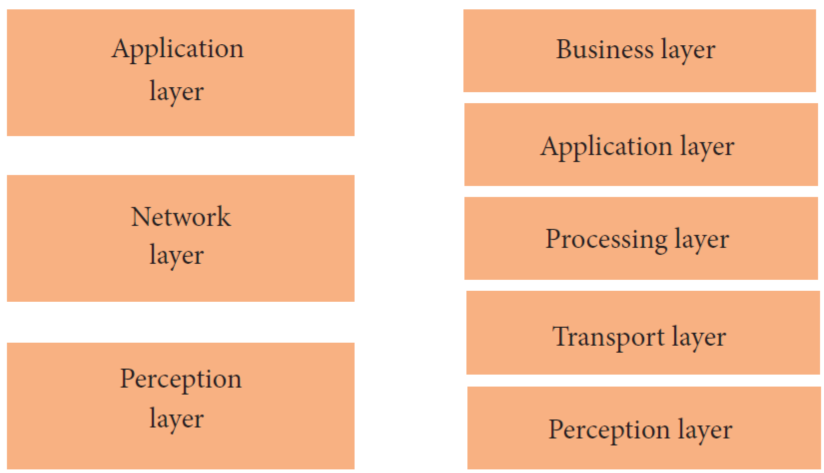



***Figure 2-1 Common IoT Architecture [26]***

The three-layer architecture is a very basic one. The three-layer architecture defines the main idea of the Internet of Things, but it is not enough for research on IoT because research often focuses on more finer aspects of the Internet of Things. Hence, we have many more layered architectures developed. The five-layer architecture added the processing and business layers.

Our research topic introduces an overlay architecture of the basic architecture, and the work can be considered in a middleware layer between the network layer and the application layer.

Also, authors in [26] provided a full survey for the research taxonomy of in IoT technologies as shown in Fig. 2-2

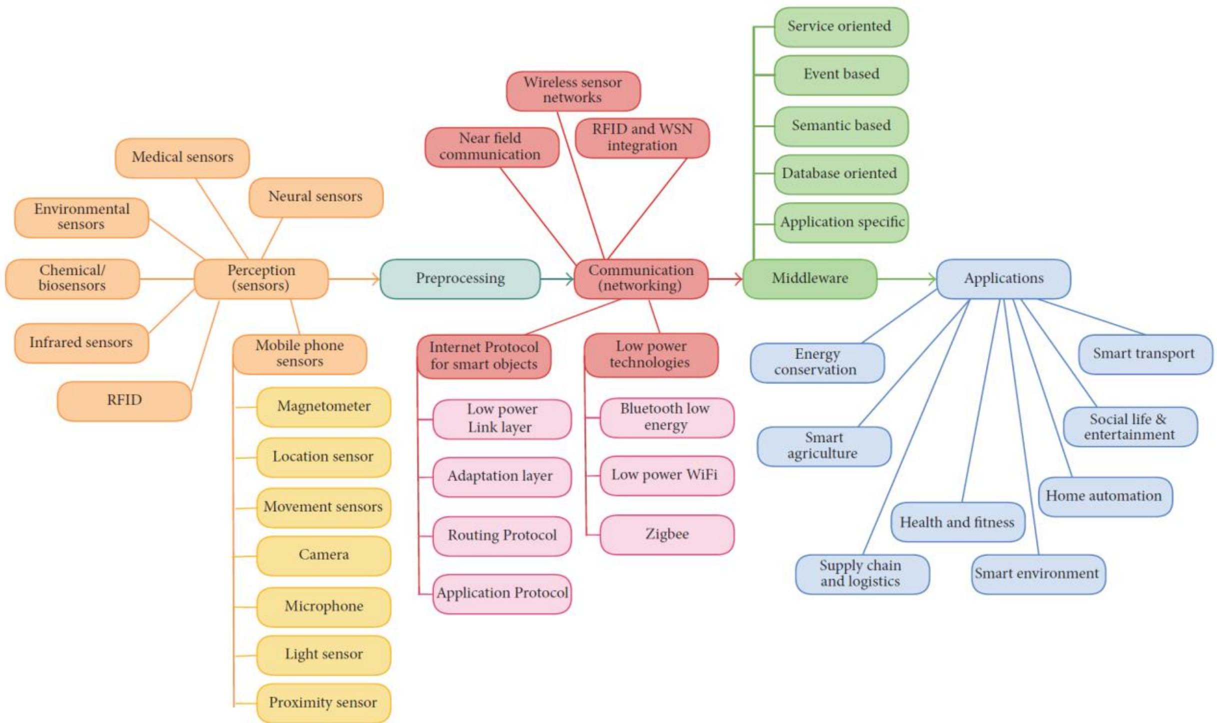



***Figure 2-2 Taxonomy of research [26]***

Based on this taxonomy, the perception layer collects data using sensors. The preprocessing layer filters and summarizes data before sending it to the network layer. Many protocols and standards are used to communicate over the network, the most common communication technologies for short range low power communication protocols are Radio Frequency Identification (RFID) and Near Field Communication (NFC) and for the medium range there are Bluetooth, Zigbee, and Wi-Fi. Communication in the large scale IoT needs specific networking protocols and approaches. Therefore, new protocols and approaches have been proposed and implemented for each layer based on the requirements required by IoT devices.

Authors in [27] provided a survey for IoT protocols as shown in table 2-1. It discusses different standards introduced by IEEE, Internet Engineering Task Force (IETF) and ITU to enable technologies matching the rapid growth in IoT. These standards include communication, routing, network, and session layers of the networking architecture that are being developed just to meet requirements of IoT. IEEE 802.15.4 is the most used IoT standard for Media Access Control (MAC), it defines a frame format, headers including source and destination addresses, and how nodes can communicate with each other. IEEE 802.11ah is a light version low energy of the original IEEE 802.11 (known as Wi-Fi) wireless medium access standard. It has been designed with less overhead to meet IoT requirements. Routing Protocol for Low-Power and Lossy Networks (RPL) is distance-vector protocol that supports a variety of datalink protocols. IPv6 over Low power Wireless Personal Area Network (6LoWPAN) is the first and most used standard.

| **Session** | | MQTT, SMQTT, CoRE, DDS, AMQP, XMPP, CoAP, …… | **Security** | **Managment** |
|---|---|---|---|---|
| **Network** | **Encapsulation** | 6LowPAN, 6TiSCH, 6Lo, Thread, …. | TCG, Oath 2.0, SMACK, SASL, ISASecure, ace, DTLS, Dice, …. | IEEE 1905, IEEE 1451, ……… |
| | **Routing** | RPL, CORPL, CARP, ….. | | |
| **Datalink** | | WiFi, Bluetooth Low Energy, Z-Wave, ZigBee Smart, DECT/ULE, 3G/LTE, NFC, 802.11ah, 802.15.4e, LTE-A, LoRaWAN, ….. | | |

***Table 2-1 Protocols for IoT***

In [28], well known IoT platforms were explained and compared to each other as shown below in table 2-2. The authors concluded that ThingWorx and Microsoft Azure are the most promising platforms for IoT solution compared to the others. Google Cloud platform is popular because of its compute engine, application engine and container engine. IBM BlueMix is common due to its powerful application creation and management support 3rd party API and services that it offers to its users. ThingWorx IoT platform is widespread as it supports 3$^{rd}$ party device cloud facility, open API and Always On features. Microsoft Azure cloud is different from others for its networking capabilities as well as its computing capabilities.

| | **Google Cloud Compute** | **IBM Blue Mix** | **Thing Worx** | **Microsoft Azure** | **Amazon Web Service** |
|---|---|---|---|---|---|
| Scalability | ✓ | ✓ | ✓ | ✓ | ✓ |
| 24*7 availability | ✓ | ✓ | ✓ | ✓ | ✓ |
| Security and Privacy provisioning | ✓ | ✓ | ✓ | ✓ | ✓ |
| Plug and play | ✓ | ✓ | ✓ | ✓ | ✓ |
| Support for millions of devices | ✓ | ✓ | ✓ | ✓ | ✓ |
| Real time data | ✓ | ✓ | ✓ | ✓ | ✓ |
| Storage of data | ✓ | ✓ | ✓ | ✓ | ✓ |
| Solution Type | PaaS | PaaS | Complete IoT | PaaS | IaaS |

***Table 2-2 Comparison of Various IoT Platforms***

## 2.2. Satellite IoT

Due to the limitations of the terrestrial communication network, the application of IoT is restricted to a great extent. The satellite IoT and terrestrial IoT jointly build a space terrestrial integrated IoT architecture, effectively expanding the application fields and scope of IoT.

Analyst firm Omdia forecasted that the global satellite IoT connectivity business will double its current revenues and may be more, going from $233 million in 2019 to reach $544 million in 2025. Cumulative satellite connections are expected to rise fourfold to more than 10 million by 2025 [29]. The installed base of satellite IoT connections is expected to increase by a nearly a factor of four in the coming years, growing at a 25 % compound annual growth rate (CAGR) from 2.7 million units in 2019 to 10.3 million units in 2025.

In this context, US Federal Communications Committee (FCC) made arrangements with SpaceX to build a constellation of 12,000 low Earth orbit communication satellites in a project called "Starlink" that will utilize phased array antennas for up and downlinks which will provide the ecumenical with low latency high bandwidth coverage [30].

A comprehensive survey for satellite IoT is introduced in [31] including the concept of satellite IoT and most of the current challenges were discussed that are facing the satellite IoT vision. As shown in Fig. 2-3, to position the new paradigm of satellite IoT in the arena of long-range IoT technologies, the following Key Performance Indicators (KPI) were considered

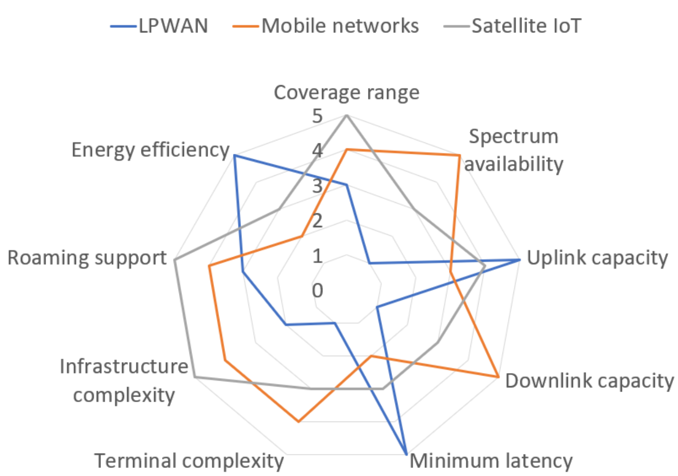



***Figure 2-3 Radar chart to compare the envisioned KPIs across different long-range [31]***

A state-of-the-art of Direct to Satellite (DtS) IoT is presented in [32], two modes of interoperability were illustrated for such a link: direct access and indirect access. The direct access mode allows devices to directly communicate with the satellite, while in the indirect access mode, each sensor and actuator in a network may communicate with the satellite through an intermediate sink node. Also Required bandwidth vs. range capacity for personal, cellular, LPWA and satellite networks was presented as shown in Fig. 2-4.

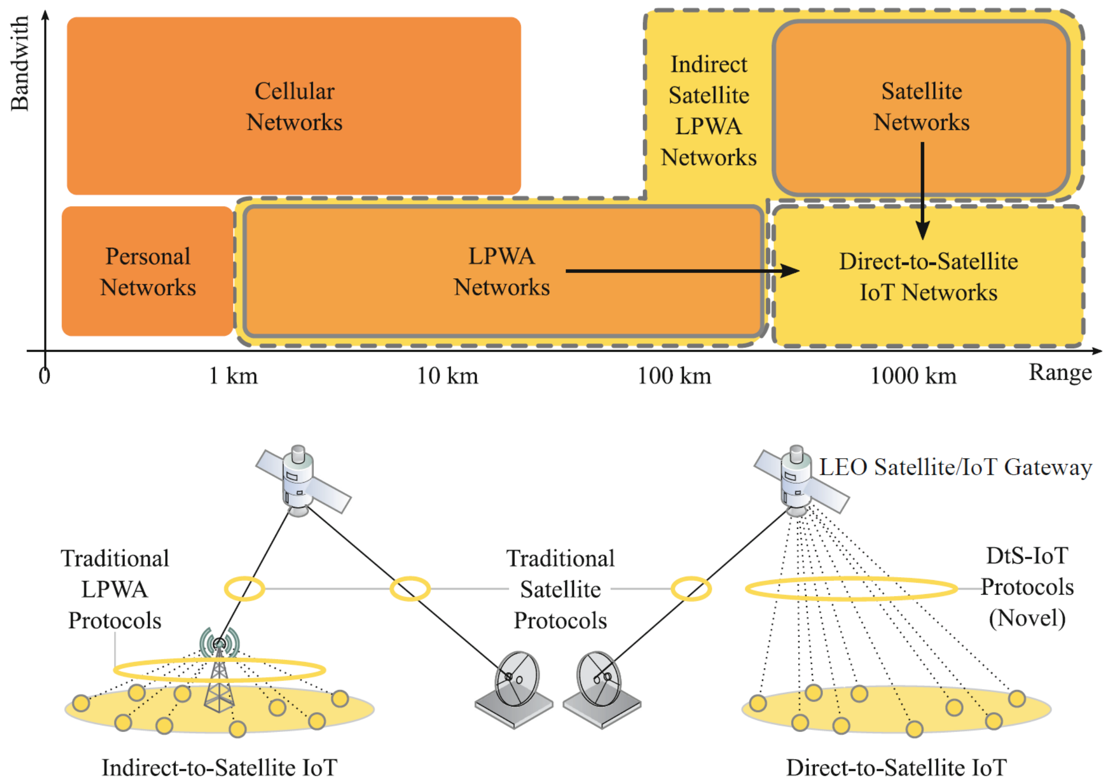



***Figure 2-4 Required bandwidth vs. range [32]***

Quality of Experience (QoE) scheme for satellite IoT is proposed to enhance the user satisfaction by constructing QoE factors mentioned in [33], QoE-aware evaluation scheme is implemented by four QoE factors which are connectivity, rate, blocking rate and charges based on the Multilayer Tabu Search (MLTS) algorithm.

## 2.3. Peer-to-peer (P2P) IoT

Peer-to-peer (P2P) topology is a well-known approach for distributed systems. Authors of [34] propose an architecture to manage IoT based on cognitive P2P networks. The IoT devices will spread highly confidential data, sending this sensitive personal data to a centralized server or cloud will introduce a severe risk to people's privacy, the risk is that whoever can access that point can exercise power over people.

Authors in [35] proposed that the data produced by personal IoT devices are safely stored in a distributed system whose design guarantees privacy by leveraging the use of P2P storage networks in addition to using the blockchain. Focusing on the scalability issue of the blockchain by performing simulations that provides empirical measures on its scalability degree. They also consider the possibility of employing blockchains whose scalability is higher than the Bitcoin blockchain.

In [36] discussed the main problem related to centralized industrial network and third part trust operation. Centralized networking has had issues with flexibility, efficiency, availability, and security. Hence, their main purpose is to present a distributed P2P network architecture that enhances the security and scalability. Their proposed architecture was developed based on blockchain technology which helped the development of a distributed P2P network with high security and scalability.

Authors of [37] addressed the requirement for scalable IoT architectures with the capability of maintaining the security and privacy of collected data especially when considering sensitive data such as medical records. They proposed an architecture model named FogChain, which combines the technologies Blockchain, Fog computing, and the IoT for the healthcare domain. Its concept is overcoming IoT constraints by employing a differential approach, adding an intermediate Fog layer near to the edge to improve their capabilities and resources. Experiments demonstrate that FogChain can achieve a 62.6% faster response time when compared to Cloud-like Blockchain infrastructures.

Lately, there has been much interest in emerging P2P network overlays because they provide a good pillar for creating large-scale data sharing and content distribution applications. These P2P networks attempt to provide many features such as selection of nearby peers, redundant storage, efficient search/location of data items, trust and authentication, and anonymity. Also, the traditional search model is based on centralized architectures that can only handle small IoT platforms' resource discovery. Most IoT systems have many different types of sensors and actuators. Hence, the centralized model is not suitable for them. The more popular model is decentralized, and the P2P architecture is the most typical architecture [38].

Due to the potential benefits of P2P approaches such as scalability, autonomy, and robustness. Authors of [39] introduced a holistic P2P application layer protocol and how it supports the data centric approach in their architecture which can span the Wireless Sensor Networks (WSN) and services over the internet. They used a Distributed Hash Tables (DHT) based on Kademlia [40] to find nodes and allow new nodes to join by knowing only the address of a node in the overlay. This DHT also allows an innovative use of forming groups of data or nodes with an associated identifier, similarly to an info-hash in BitTorrent [41] this P2P approach allows IoT to move beyond isolated islands of data to nodes and services that are more easily deployed, developed, and integrated, e.g., in a healthcare.

In [42], authors consider the appropriateness of P2P approach for fog computing to meet the requirements of IoT networks. P2P networks potentially offer an efficient routing architecture that is self-organizing, massively scalable, and robust in the wide area, combining fault tolerance, load balancing and explicit notion of locality. There are two classes of P2P overlay networks which are structured and unstructured:

- Structured P2P, is that the P2P overlay network topology is firmly controlled, and content are placed not at random peers but at specified locations that will make successive queries more efficient. Such Structured P2P systems use the Distributed Hash Table (DHT) as a substrate, in which data object (or value) location information is placed deterministically, at the peers with identifiers corresponding to the data object's unique key.
- Unstructured P2P system consists of peers joining the network with some loose rules, without any prior knowledge of the topology. The network uses flooding as the mechanism to send queries across the overlay. When a peer receives the flood query, it sends a list of all content matching the query to the originating peer. While flooding-based techniques are effective for locating highly replicated items and are resilient to peers joining and leaving the system. Obviously, this approach is not scalable as the load on each peer grows linearly with the total number of queries and the system size. Therefore, unstructured P2P networks face one basic problem which is the system does not scale when handling a high rate of aggregate queries and sudden increase in system size.

The use of Distributed Hash Tables (DHT) for resource discovery in structured P2P networks has become a highly studied research area [43]. Content Addressable Network (CAN) [44], Pastry [45] and Chord are widely used examples of DHTs.

## 2.4. Chord

By reviewing the features included in recent P2P applications, it results redundant storage, efficient data location and searching [24]. Hence, applications need an efficient method for determining the location of a data item.

Chord is widely known as one of popular DHT algorithms to significantly reduce the number of hops for looking up and updating in P2P network. By using consistent hashing, chord organizes nodes into a ring that can cover the huge number of nodes. Each node maintains routing information of other O(*log* N) nodes. Thus, only O(*log* N) hops are required for each lookup request and O($log^2$ N) hops for updating when a node joins or leaves the ring. Chord has many advantage features such as:

- Load balance: distributed hash function, spreading keys evenly over nodes
- Decentralization: Chord is fully distributed, no node more important than other, improves robustness
- Scalability: logarithmic growth of lookup costs with number of nodes in network, even very large systems are feasible.
- Availability: Chord automatically adjusts its internal tables to ensure that the node responsible for a key can always be found.
- Locality: Nodes close on ring can be far in the network

Chord uses consistent hashing [46] to assign (store) keys to its peers. Consistent hashing is designed to let peers enter and leave the network with minimal interruption. This decentralized scheme tends to balance the load on the system, since each peer receives roughly the same number of keys, and there is little movement of keys when peers join and leave the system. In a steady state, for N peers in the system, each peer maintains routing state information for about only O(*log*N) other peers (N number of peers in the system) as a small routing table called finger table.

The consistent hash functions assign peers and data keys an m-bit identifier using the hash function Secure Hash Algorithm SHA-1 [47]. Each chord node (peer) has a unique identification (ID) by hashing its IP address and each information (key) has also a unique ID by the hashing the key itself and both are uniformly distributed:
Node ID = SHA-1(IP address)
Key ID = SHA-1(key)

Identifiers are ordered on an identifier circle (ring) modulo 2m [48]. Key k is assigned to the first peer whose identifier is equal to or follows k in the identifier space. This peer is called the successor peer of key k, denoted by successor(k). If identifiers are represented as a circle of numbers from 0 to $2^{m-1}$, then successor (k) is the first peer clockwise from k. The identifier circle is termed as the Chord ring. To maintain consistent hashing mapping when a peer n joins the network, certain keys previously assigned to n’s successor now need to be reassigned to n. When peer n leaves the chord system, all its assigned keys are reassigned to n’s successor. Therefore, peers join and leave the system with $(log\mathrm{N})^2$performance.

Each peer in the chord ring needs to know how to contact its current successor peer on the identifier circle. Lookup queries involve the matching of key and NodeID. For a given identifier could be passed around the circle via these successor pointers until they encounter a pair of peers that include the desired identifier; the second peer in the pair is the peer the query maps to. The response is returned along the reverse of the path.

Chord simply is a protocol that solves the lookup problem and a simple system that uses it for storing information, given a key, it will determine the node responsible for storing the key's value. The basic chord system supports five main operations where all operations use the lookup primitive offered by the chord protocol:

1. Storing (assigning) keys
2. Addition of chord nodes
3. Departure of chord nodes
4. Update
5. Lookup

**Storing (assigning) keys:** Key K is assigned to the first node whose identifier is equal to or follows (the identifier of) k in the identifier space.
In Fig. 2-5, the number of bits is three (m=3), so we have $2^m \rightarrow 2^3$ places, three nodes only are available at 0, 1 & 3 places

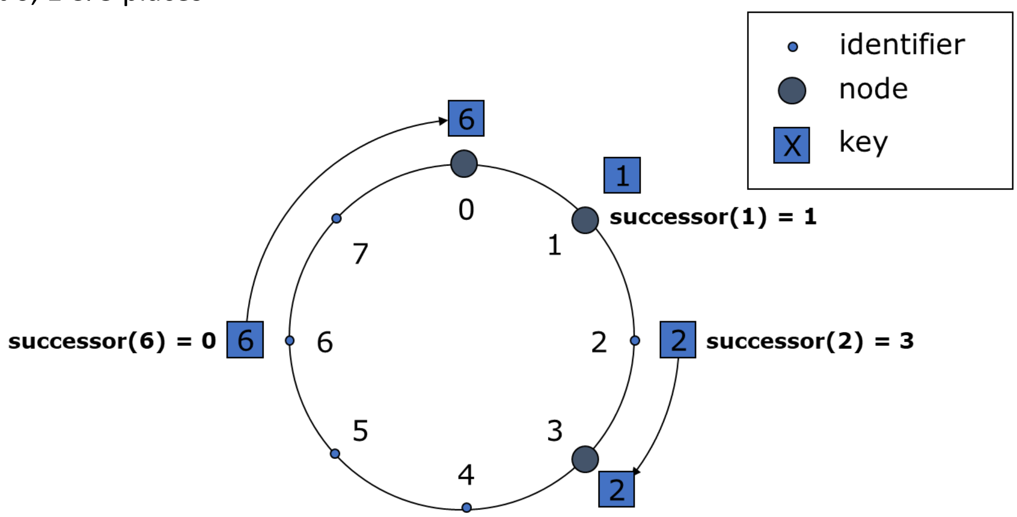



*Figure 2-5 Assigning keys to Chord*

Key 1 stored at node 1 (successor (1) = 1), key 2 stored at node 3 (successor (2) =3) as node 2 doesn't exist in our chord ring, key 6 stored in node 0 (successor (6) = 0) as node 6 doesn't exist in our chord ring.

**Join and departure of nodes:** When a node n joins the network (chord ring), certain keys previously assigned to n's successor now become assigned to n. When node n leaves the network, all its assigned keys are reassigned to n's successor. In Fig. 2-6, Node 6 will join the chord ring. A redistribution for the keys will take place and identifying the successors and predecessors. So key 5 (k5) will be moved from node 0 and stored in new node 6.

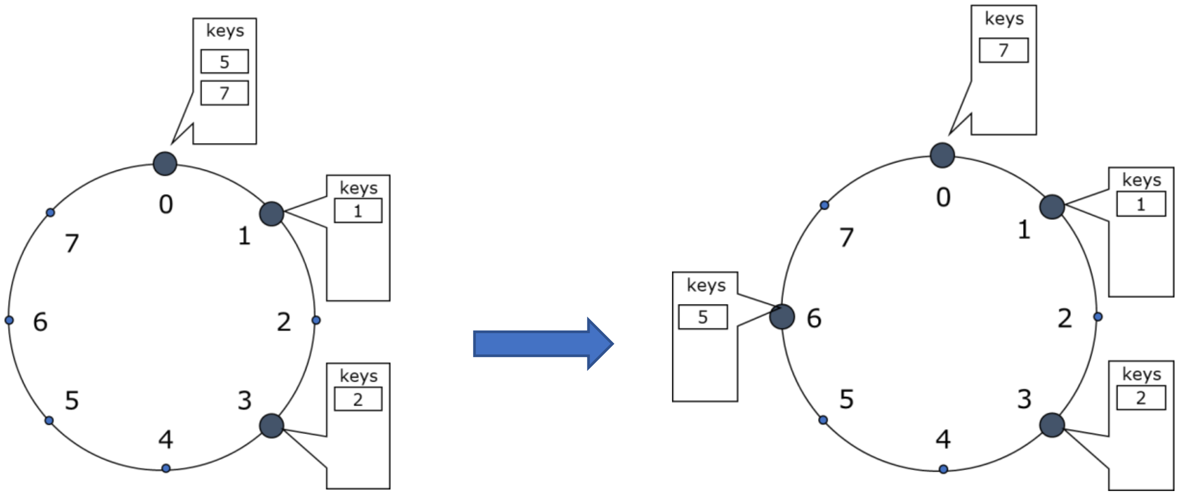



***Figure 2-6 New node joining the Chord***

In Fig. 2-7, node 1 will leave the chord ring. Node 1 was storing key 1 only, and it will be moved to node 3.

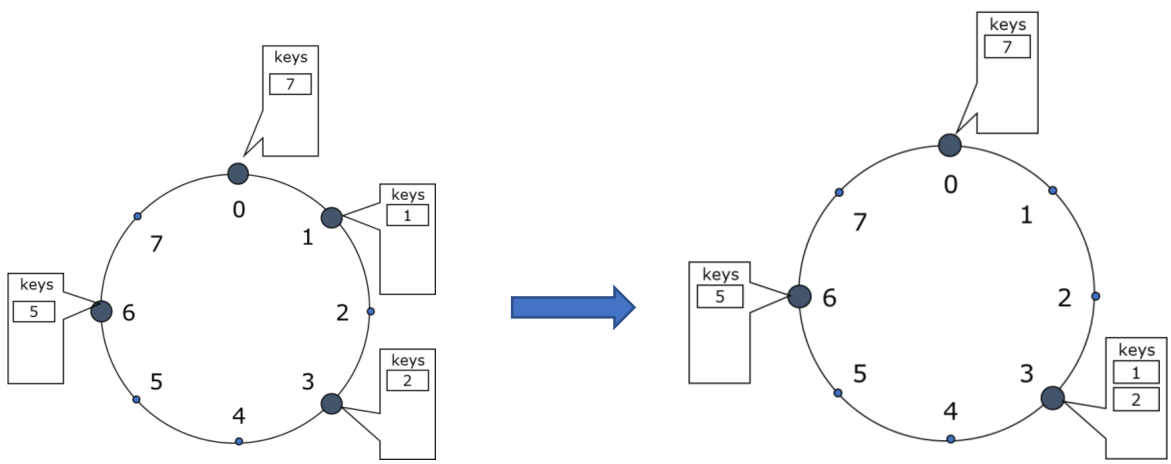



***Figure 2-7 Node leaving the Chord***

Each node n' maintains a lookup table with up to m entries (which is in fact the number of bits in identifiers), called finger table.

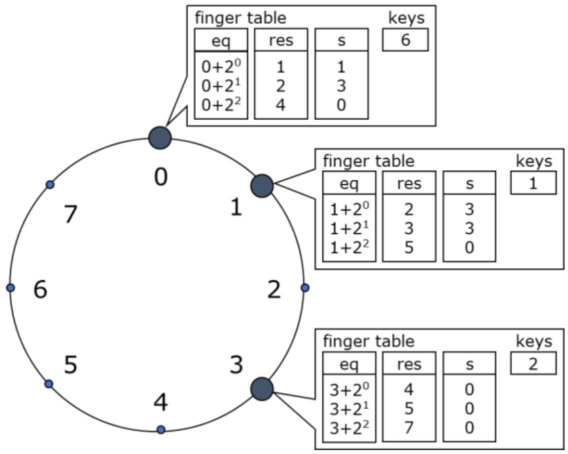


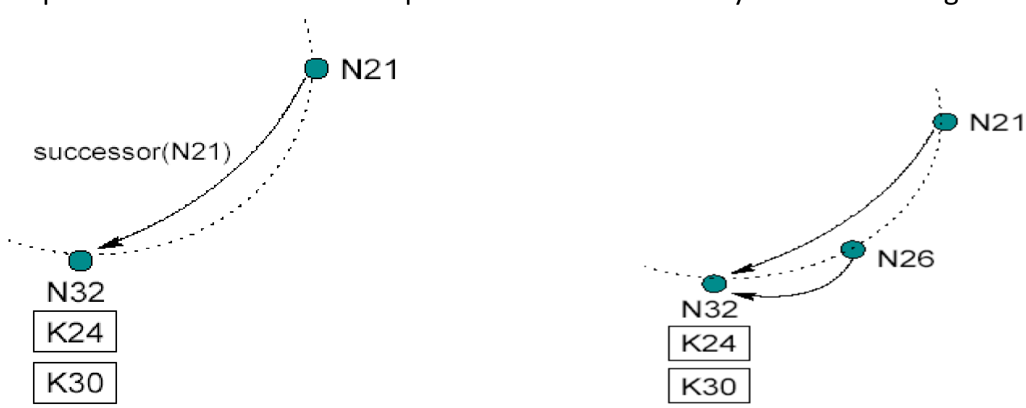
*Figure 2-8 Building up finger tables*

Building finger table is based on using the following formula: s = successor(n+$2^{i-1}$) as shown in Fig. 2-8.

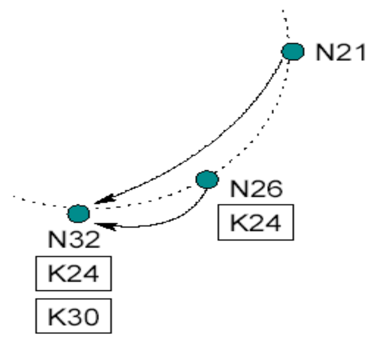
The following applications are examples of how chord could be used:

- Cooperative File System [49], in which multiple providers of content cooperate to store and serve each other's data. Spreading the total load evenly over all participant hosts lowers the total cost of the system, since each participant needs to provide capacity only for the average load, not for the peak load.
- Chord-based DNS [50], provides a lookup service, with host names as keys and IP addresses (and other host information) as values. Chord could provide a DNS-like service by hashing each host name to a key.

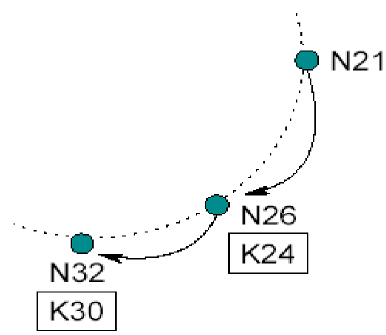
**Stabilization protocol:** The successor/predecessor links are rebuilt by periodic stabilize notification messages sent by each node to its successor to inform it of the (possibly new) identity of the predecessor. The successor pointers are used to verify and correct finger table entries.




- N26 joins the system
- N26 acquires N32 as its successor
- N26 notifies N32
- N32 acquires N26 as its predecessor




- N26 copies keys
- N21 runs stabilize() and asks its successor N32 for its predecessor which is N26.




- N21 aquires N26 as its successor

## 2.5. Multidimensional Approach

The multidimensional approach has been adopted by many researchers because they lack a mechanism for integration of heterogenous different smart Contexts. In this way, the hierarchical architecture at logical management layer such as multidimensional approach for organizing and discovering devices in large IoT environments is the urgent requirement today.
Authors of [43] showed how multi-layered multi-ring architecture cased on chord can support multi-dimensional data lookup. They demonstrated a reduction in messages required to route a query in worst-case scenarios compared with a single-layered chord ring and other multiring architectures.

In [51], an efficient chord-based horizontal architecture. In the proposed architecture, shown in Fig. 2-9, they focused on the fog layer and present a new fog architecture that minimizes lookup and storage complexity. A direct approach to the proposed chord-based architecture would be to consider each node (Fog-level or thing-level) of the environment as an entity. To reduce the number of nodes, the nodes are aggregated into Fog computing domains (FCDs). The aggregation process elects one or more Fog nodes as coordinators for a given FCD. Figure 2-9 shows only one coordinator for each FCD. Because the coordinator is representing the whole FCD, it is held responsible for any violations by the FCD members. The coordinator is expected to manage the member nodes and represent its FCD within the global Fog community. The global Fog community (i.e., Coordinators) relates to a chord ring.

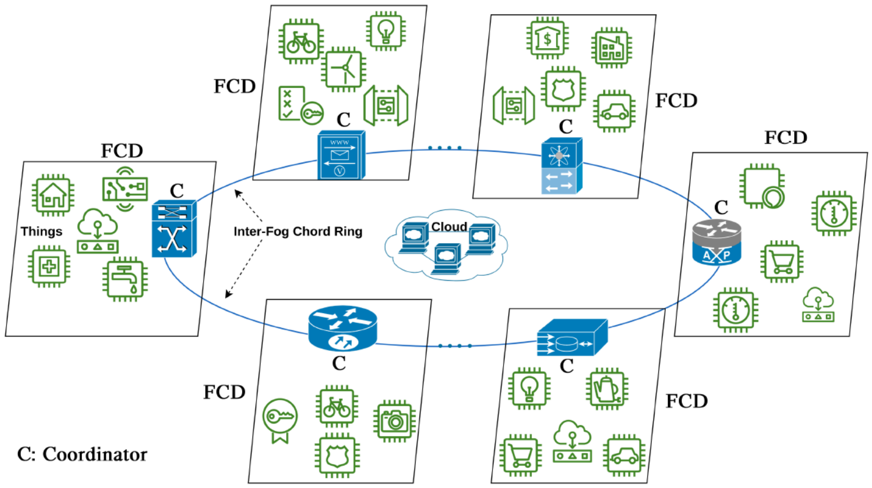



*Figure 2-9 Chord-based Fog Architecture [51]*

In [52], the authors propose a novel approach to manage things in large IoT environments based on chord protocol. Each node on the ring is considered as an IoT gateway and plays the role of transmitter and connector to sensors. In their proposed architecture, by using Distributed Hash Table (DHT), they identify each ring, gateway, and sensor by a unique hashed ID under several features to distinguish each other as in Fig 2-10.

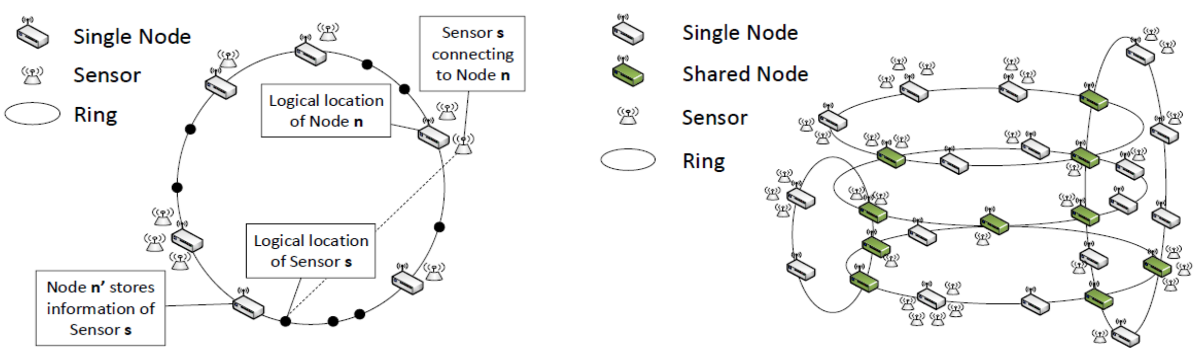



*Figure 2-10 Multiple Peer Chord Rings [52]*

Their architecture is tested by simulating a large-scale environment with many nodes and sensors to evaluate performance and operation costs of the system with various topologies and properties. The achieved results proved that the proposal has the operability and feasibility to be practically applied.

## 2.6. Summary

A summary table including all the related work

| Reference | year | IoT | 5G | P2P | Fog | SIoT | Cloud | Standards | DHT | Chord | Multi-dimensional |
|---|---|---|---|---|---|---|---|---|---|---|---|
| 24 | 2001 | | | | | | | | | ✓ | |
| 26 | 2016 | ✓ | | | ✓ | | ✓ | ✓ | | | |
| 27 | 2017 | ✓ | | | | | | ✓ | | | |
| 28 | 2015 | ✓ | | | | | ✓ | ✓ | | | |
| 29 | 2020 | ✓ | | | | ✓ | | | | | |
| 30 | 2020 | ✓ | ✓ | | | ✓ | | | | | |
| 31 | 2021 | ✓ | ✓ | | | ✓ | | ✓ | | | |
| 32 | 2019 | ✓ | ✓ | | ✓ | ✓ | ✓ | ✓ | | | |
| 33 | 2019 | ✓ | ✓ | | ✓ | | | ✓ | | | |
| 34 | 2019 | ✓ | | ✓ | | | ✓ | | | | |
| 35 | 2017 | ✓ | | ✓ | | | | | | | |
| 36 | 2018 | ✓ | | ✓ | | | ✓ | | | | |
| 37 | 2021 | ✓ | | ✓ | ✓ | | | | | | |
| 38 | 2021 | ✓ | | ✓ | | | | | ✓ | ✓ | ✓ |
| 39 | 2019 | ✓ | | ✓ | | | ✓ | | ✓ | | |
| 40 | 2002 | | | ✓ | | | | | ✓ | | |
| 41 | 2008 | | | ✓ | | | | ✓ | | | |
| 42 | 2019 | ✓ | | ✓ | ✓ | | ✓ | | | | |
| 43 | 2005 | | | ✓ | | | | | | ✓ | ✓ |
| 44 | 2001 | | | ✓ | | | | ✓ | ✓ | | |
| 45 | 2001 | | | ✓ | | | | | ✓ | | |
| 46 | 1997 | | | ✓ | | | | | ✓ | | |
| 47 | 1995 | | | | | | | | ✓ | | |
| 48 | 2005 | | | ✓ | | | | ✓ | ✓ | ✓ | |
| 49 | 2001 | | | ✓ | | | | | ✓ | ✓ | |
| 50 | 2002 | | | ✓ | | | | | ✓ | ✓ | |
| 51 | 2021 | ✓ | | ✓ | ✓ | | ✓ | ✓ | | ✓ | ✓ |
| 52 | 2017 | ✓ | | ✓ | | | | | ✓ | ✓ | ✓ |

***Table 2-3 Related Work Summary***

# Chapter 3

## 3. Satellite IoT Multi-Chord Approach

It is well-known that the IoT solutions require the capabilities of very high scale and dynamic adaptation to be able to expand on large space such as towards the development of smart cities or even smart country soon.

Following this universal adoption, we were motivated to propose an architecture for IoT. Such architecture will be used as an enabler for large scale IoT networks [53]. Our goal is to develop multi-chord architecture for self-organizing P2P integrated networks [54,55] considering the global coverage of satellite network and advantages of fog service.
Multiple chord rings are used to build a new framework that adapts to multi-attribute and range queries. The framework is organized into two parts, the Resources Directory System and the Resources Index and Discovery System [38].
It is three-fold contribution as follows:

1. Multiple Interconnected chords.
2. Fast response between IoT nodes for data manipulation.
3. Extension from the terrestrial to satellite IoT networks.

### 3.1. Approach

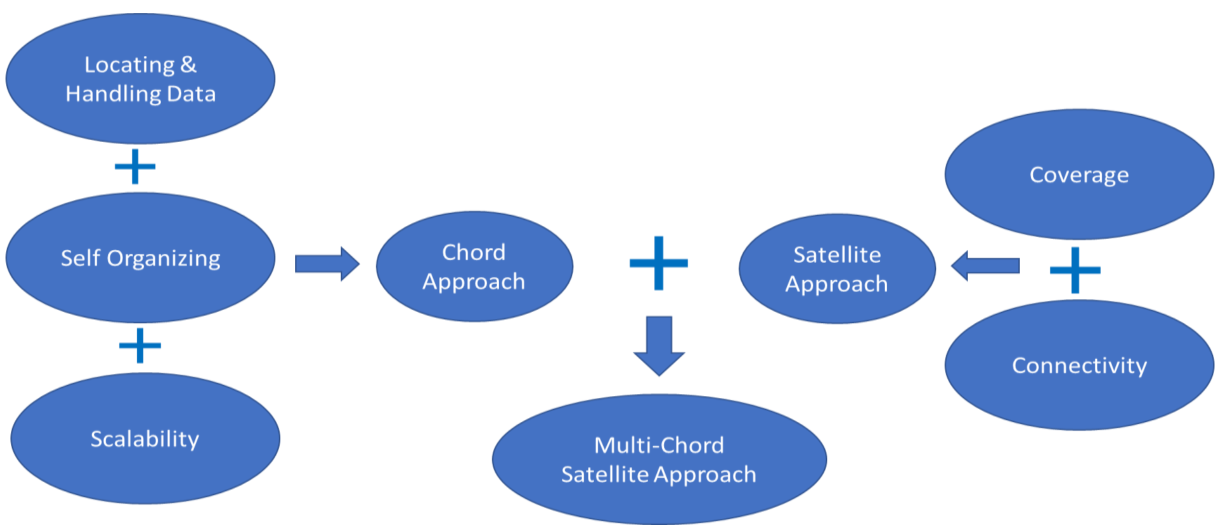



***Figure 3-1 Multi-Chord Satellite Approach***

The adopted approach gathers the benefits and challenges of both Chord and Satellite approaches. The benefits of chord are Locating & handling data, Self-Organizing and Scalability. Simply, locating and handling data is means that if the target of a query is near the originating node, then the originating node should not have to contact distant nodes to resolve the query. The self-organizing is using advanced algorithms in conjunction with high-level human-defined goals and policies, Self-stabilizing and hashing approach are considered as a proposed enhancement at our research. The current Scalability approaches adopted the cloud-based architectures.

On the opposite side, the space network that consists of numerous Low Earth Orbit (LEO) satellites, considered as a promising technique since it is the only solution to provide full global coverage for the whole earth [56]. Nevertheless, compared with Geostationary Earth Orbit (GEO) satellites, the LEO satellites always move very fast to cover an area within only 5-12 minutes per pass, bringing high dynamics to the network access. Furthermore, to reduce the cost, the power and spectrum channel resources of each LEO satellite are very limited, i.e., less than 10% of GEO.

Authors in [57] generalized a techno-economic model to assess the engineering-economics of satellite constellations by simulating the impact on coverage, capacity, and cost, as both the number of satellites and quantity of subscribers increases. Then assessment is visualizing the potential capacity and cost per user via different subscriber scenarios. The results show how limited the capacity will be once resources are spread across users in each satellite coverage area. For example, for 0.1 users per $km^2$ (so 1 user per 10 $km^2$), we estimate a mean per user capacity of 24.94 Mbps, 1.01 Mbps and 10.30 Mbps for Starlink, OneWeb and Kuiper, respectively, in the busiest hour of the day. But if the subscriber density increases to 1 user per $km^2$, then the mean per user capacity drops significantly to 2.49 Mbps, 0.10 Mbps and 1.02 Mbps. LEO broadband will be an essential part of the connectivity toolkit, but the results reveal that these mega-constellations will most likely have to operate below 0.1 users per km 2 to provide a service that out-competes other broadband connectivity options.

The constellation of Starlink has grown to almost reach 1,700 satellites through 2021 and will eventually consist of many thousands of mass-produced small satellites in low Earth orbit (LEO), which communicate with designated ground transceivers.
While the technical possibility of satellite internet service covers most of the global population, actual service can be delivered only in countries that have licensed SpaceX to provide service within any specific national jurisdiction. As of January 2022, the beta service offering is available in 24 countries. The initial engineering parameters from the ITU fillings are reported in [58,59].
Early-stage planning began in 2014, with product development occurring in earnest by 2017. Two prototype test-flight satellites were launched in February 2018. Additional test satellites and 60 operational satellites were deployed in May 2019. SpaceX launches up to 60 satellites at a time, aiming to deploy 1,584 of the 260 kg (570 lb) spacecraft to provide near-global service by late 2021 or 2022.

## 3.2. Case Study System Model

We get inspired by the new LEO constellations which are essential in the large IoT networks. Such as Starlink, OneWeb and Kuiper [57] which are the three main competing LEO constellations provides satellite Internet access coverage with high-speed, low-latency broadband internet across the globe.

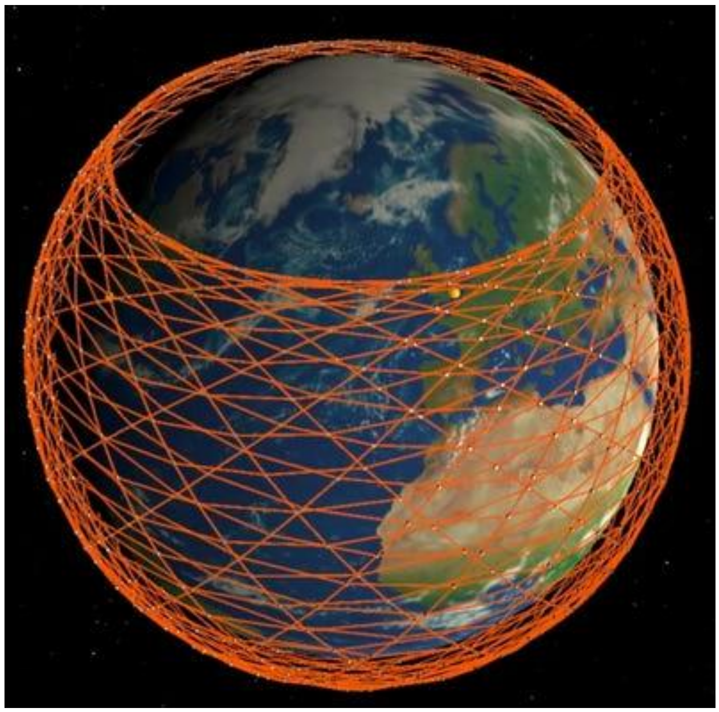


***Figure 3-2 Starlink constellation***

The case study in our research is modelling LEO orbit constellation of as a "main-chord" ring, the LEO satellites are presenting the chord nodes. Each LEO satellite has a unique ID [60] as in StarLink satellites shown in Fig. 3-2. Each main-chord according to its coverage at that time can be linked by one or more (up to the number of LEO satellites per orbit) terrestrial smart context as IoT environment. After certain time, the same smart contexts will be covered/linked by different LEO satellite, this will be explained later in relative motion subsection. The smart context is presented by "sub-chord". Our example shown in Fig. 3-3, we have one main-chord connected by two terrestrial sub-chord.

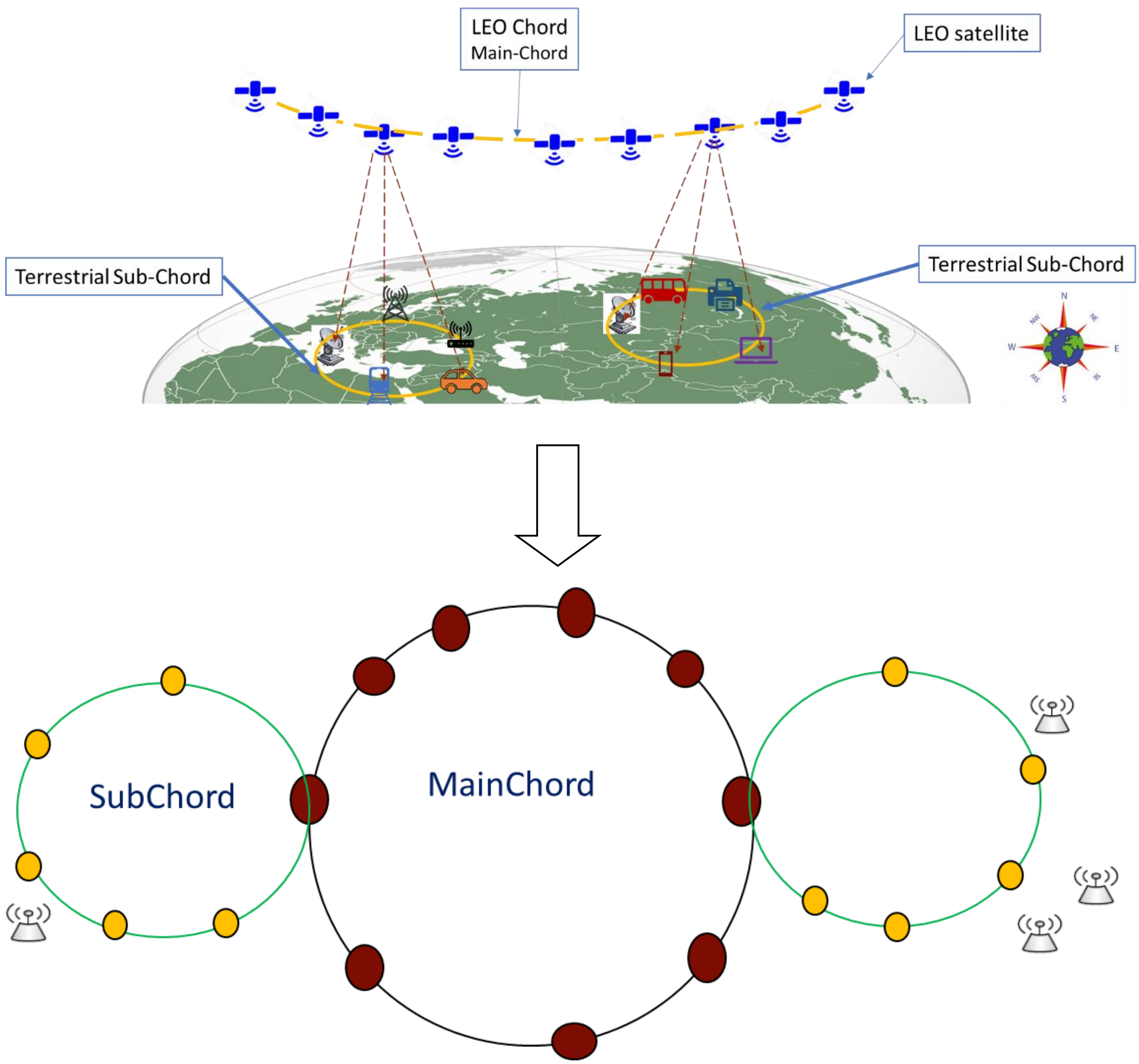



***Figure 3-3 Mapping our case study into logical representation***

We consider the environment as an integration of multiple logical Chord rings, which could be intersected together by one or more shared nodes. Our proposed structure consists of main-chord and sub-chords. Each smart context is considered as a sub-chord.

So, the chord ring will be the logical management network for a single IoT context. Ring uses chord as a base method to organize and discover things. Node is a peer object located on the ring. Node corresponds with a gateway device in fact.

We divide nodes into two types:

- Single node: is a node that is participating in only one chord ring.
- Shared node: is a node that is simultaneously participating in at least two chord rings.

Sensor is an object representing sensors in practice. In our multi-ring architecture, sensors cannot join directly on rings like nodes (i.e gateway) because they obvious characteristics of real devices, which are limited by capabilities of communication, computing, and storage. Therefore, sensors must connect to a certain gateway. Connection information between a sensor and its gateway is distributed on ring nodes.

Figure 3-3 presents the traffic characteristics of deployed massive IoT connected devices in a city scenario [61].

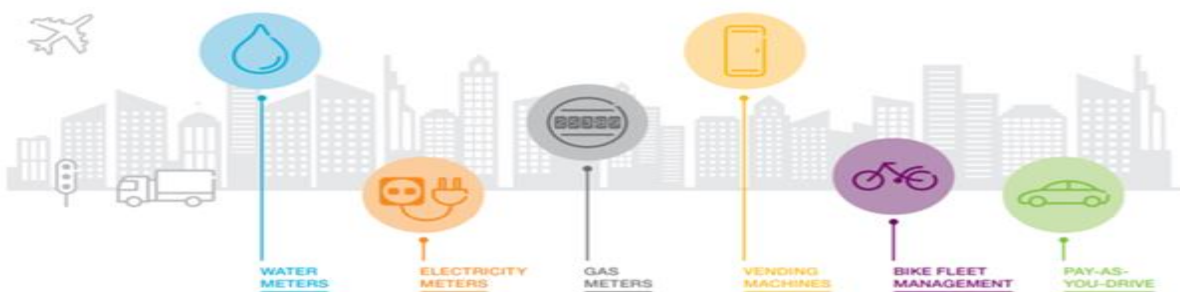



***Figure 3-4 Massive IoT Connected devices in a city scenario [61]***

A dense urban environment with 10,000 households per $km^2$ was used as the base for a massive IoT services scenario. A selection of connected device types was assumed to be deployed in the area, including water, gas and electricity meters, vending machines, rental bike position monitors and accelerometers in cars monitoring driver behavior. Traffic characteristics for each device are summarized below in table 3-1:

| | **Water meters** | **Electricity meters** | **Gas meters** | **Vending machines** | **Bike fleet management** | **Pay as you drive** |
|---|---|---|---|---|---|---|
| Typical Message size | 100 bytes | 100 bytes | 100 bytes | 150 bytes | 150 bytes | 150 bytes |
| Message interval | 12 hours | 24 hours | 30 minutes | 24 hours | 30 minutes | 10 minutes |
| Device density | 10,000/$km^2$ | 10,000/$km^2$ | 10,000/$km^2$ | 150/$km^2$ | 200/$km^2$ | 2,250/$km^2$ |

***Table 3-1 Traffic Characteristics***

The number of connected devices used in this scenario represents a mature, large scale massive IoT scenario and the services represent a realistic range of massive IoT use cases that are expected to be deployed in an urban environment.

Deployment environment and traffic models differ for these services: a remote-controlled meter may face an indoor coverage challenge, while a device mounted on a bike is usually found outside. The traffic intensity from meters may be once per day, whereas other devices may need to transmit every 10 minutes. The data traffic for massive IoT devices is small; the typical data packet for a service is about 100-150 bytes, accounting for a payload of the device ID, time stamp and report data values.

## 3.3. Basic Chord Adaptation

In multi-chord P2P topology, nodes may join or leave a sub-chord as in single chord topology. The main difference in operations is that the nodes joining and/or leaving a sub-chord may be a member of multiple sub-chords and exist in the chord ring and finger tables of nodes residing in multiple sub-chords.

After reviewing all chord characteristics, two aspects so far must be enhanced which are the chord table and the chord operations to be able to integrate the Chord into dynamic IoT environment.

### 3.3.1. Chord Table

A new table is created, Chord table, which is distributed on every node exist in chord ring. Thus the new table provides routing information that enables discovery queries can go from a chord ring to others.
Each Chord table row contains two fields:

- Node ID: indicates a distinct successor node that appears in Finger table.
- Chord ID: lists Chord IDs that the node is participating.

The main reason for creating a new chord table on every node instead of adding an extra field Chord-IDs in Finger table is to avoid checking and updating repeatedly and redundantly Chord-IDs of the same successor node on many rows of Finger table. Otherwise, because a node can simultaneously join in multiple Chord rings, for each Chord rings, the node requires an individual Chord ring data including information of Chord-ID, Successor, Predecessor, Finger table, and Chord table. With our design, things organization and discovery on each ring will be maintained stably and independently.
Hence, Node ID and Chord ID can be generated as below:
Node ID = SHA-1(IP address)
Chord ID = SHA-1(LEO satellite ID)

### 3.3.2. Operations

The overall flow diagram of these operations is shown in Fig. 3-5

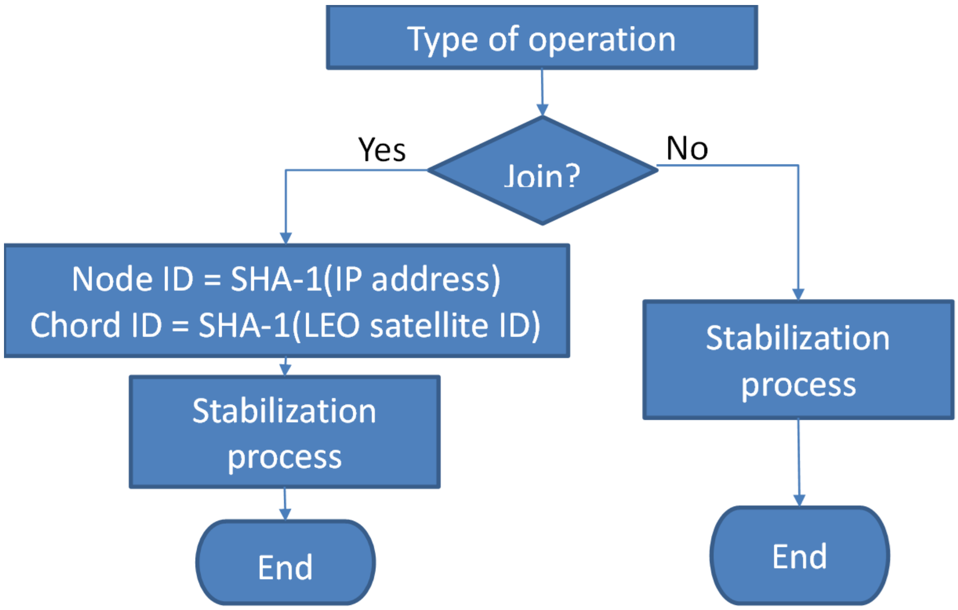



***Figure 3-5 Flow chart of node joining operation***

First a decision is made if a node in joining or leaving. If a node is leaving the stabilization process is executed in the sub-chords of interest as will be explained later. However, if a node is joining, then an ID must be allocated to the new node as well as the ID of the sub-chord that it will join using the consistent hashing of the LEO Satellite ID and hashing the node IP address to get the Node ID. Then a stabilization process is executed in the sub-chords of interest.

**Node Joins and Stabilization :** To ensure that lookups of keys are executed correctly after a node join, nodes member of the same Chord ID will have to change their corresponding finger table as the set of participating nodes changes. In addition, nodes that are existing in such sub-chord and common in other sub-chords will have to change their ring tables.
In the sub-chord of interest, each node's successor pointer must be changed to include the new node that joined and be up to date. Nodes achieve this through the "stabilization" protocol introduced in chord topology. The stabilization protocol runs based on event-driven concept and not periodically as in standard chord topology to update sub-chord nodes' finger tables, successor list and the Chord table.

When node n starts to join a sub-chord, it calls first **n.hash ()** to set its Node ID and its Chord ID. It then calls **n.broadcast()** where it broadcast its Node ID for other nearby nodes to hear and reply with their Node ID accordingly. **n.join( n')**, where n' is sub-chord node that replied to the broadcast and closest to follow the ID of the joined node. The **join ()** function asks n' to find the immediate successor of n. **join ()** does not make the rest of the nodes in the sub-chord aware of new node n that joined the sub-chord. however due to the chain effect of the event of node n joining the sub-chord, every node that learns of the existence of the new node runs **stabilize ()** to update its finger table and successor list and inform nodes in its old finger table and successor list of the changes it made in order that they in return run the stabilize function. Each time node n runs **stabilize ()**, it asks its successor for the successor's predecessor p, and decides whether p should be n's successor instead. This is how new nodes initialize their finger tables, and it is how existing nodes incorporate new nodes into their finger tables. Each node also runs check predecessor, to clear the node's predecessor pointer if **n.predecessor** has failed; this allows it to accept a new predecessor in notify.

We completed the mechanisms of operations including discovery, join in and out in this architecture. With this approach, we can solve the complicated connection problem among things and hence enables IoT systems to manage and control things more efficiently.
The basic chord functions are:

- Insert
- Lookup
- Search
- Delete

**Insert a key/lookup a key:** In standard chord to insert / lookup key is a simple operation as it requires only hashing the data to a key then finding the nodes that data to a key then finding the nodes that has an id similar or directly succeeding the key.
However, in multi-chord the operation is more complicated as the key may be stored in a sub-chord of the multi-chord network. Thus, the insert operation requires two steps. First step is to decide sub-chord ID and the second step is to find the node to store this key.

**Delete a key / update a key:** To delete / update a key, the process is very similar to insert a key as the steps required are first to lookup the sub-chord, then the node inside the sub-chord that holds the keys to be deleted / updated.

<u>Lookup algorithm</u>
*//Step 1: hash the data to a key ID*
*KID = n. hash(data)*
*//step 2: Lookup within same sub-chord for node storing the key*
*B=n. find (FingerTable, KID)*
*If B is true go to step 5*
*Else go to step 3*
*//step3: find shared node in chord table to forward lookup process to*
*SCID=n. find (CID, Chord table)*
*B=SCID. find (FingerTable, KID)*
*If B is true go to step 5*
*Else go to step 4*
*//step 4: find node in CID that will forward the request*
*SCID = SCID. find successor Chord ID (CID, Chord table)*

*Repeat step 4 until SCID is the Chord ID*
*//step 4: find node in CID that holds key ID*
*n. find node (key ID)*
*where SCID is the successor Chord ID and CID is the Chord ID and KID is the Key ID*

<u>Building finger table algorithm</u>

```
// step 1 : for a certain node in a certain sub-chord, find successor
Finger [1]. Node ID = Node ID . Find successor ()
//step 2 : build the rest of finger table
For i= 1 to y (y is number of bits representing the number of nodes in a sub-chord)
        F_NID next = +NID
        If F_NID next doesn't belong to sub-chord
                Finger [i+1].NID = NID. Find closest successor
        Else finger [i+1].NID = F NID next
        End
End
```

<u>Building Chord table algorithm</u>

```
// step 1 : for a certain node in a certain sub-chord, find successor
Nsuccessor = NID find successor
Npredecessor = NID find predecessor
Chord [i] . NID = Chord ID

// step 2: build the rest of ring table
For i = 1 to x (x is number of bits to represent number of sub-chords)
        If Nsuccessor . Chord ID ≠ NID . CID
                Chord [i+1] = Nsuccessor . CID
                Else repeat until new CID is found
                Nsuccessor = NID .Find successor
                Chord [i+1] = Nsuccessor . CID
                Else Chord [i+1] = CID
                End

        If Npredecessor . Chord ID ≠ NID . CID
                Chord [i+1] = Npredecessor . CID
                Else repeat until new CID is found
                Npredecessor = NID .Find predecessor
                Chord [i+1] = Npredecessor . CID
                Else Chord [i+1] = CID
                End
End
```

New proposed chord functions were developed to be able to integrate the chord into dynamic IoT networks instead of the static IP networks according to the different possible scenarios that we may face in our research such as:

**Remove shared node:** In our multi-chord approach, a situation my occur where a node that is a member in the main-chord and a sub-chord fails or disconnected. Accordingly, the nodes in this sub-chord will need to connect to another node in the main-chord. This scenario requires the blow steps as shown in Fig 3-6:

1. Allocating another node in main-chord:

   The criteria of allocating another node in main-chord is to choose a node that is currently not shared in another sub-chord so not to overload existing shared node. To allocate this node, the finger table of the removed node is used to find this new node that is currently free. If all nodes in finger table are shared, then the finger table of successor node is investigated until a free node is allocated.

2. Updating the finger table of the chosen node

   The updating of the finger table is like the algorithm of a node joining the sub-chord

3. Updating the chord tables of nodes within the network

   The chord table of the removed node is copied to the chosen node and a stabilization process is carried to update the chord tables of nodes in concern

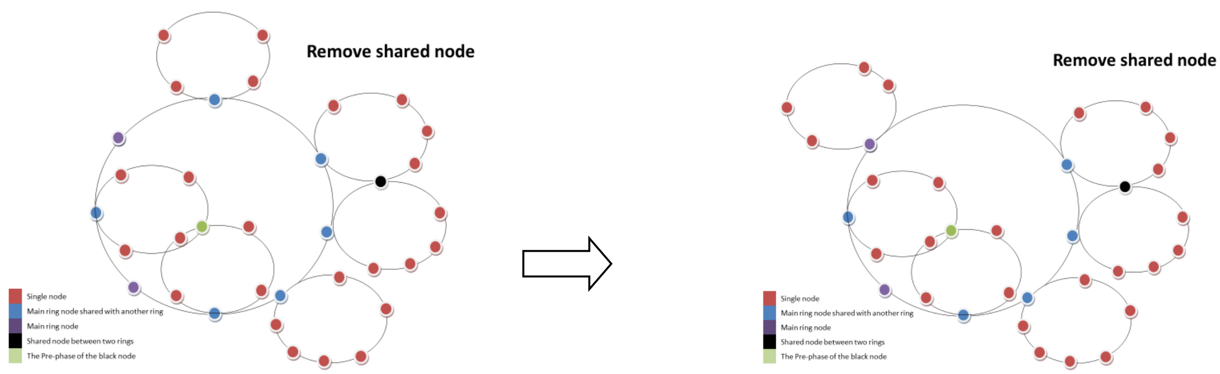



***Figure 3-6 Remove Shared Node***

**Merge shared chords:** In this situation, two sub-chords are merged into one. This may be due to remove of shared nodes of one of those sub-chords and there is no free node in the main chord to take place of the removed node. In this scenario the process is very similar to a removed node with the addition of an analyzing process where the nodes of the sub-chord that will merge to another sub-chord will build an analyzing table that maps the chord ID of the sub-chord to be merged with as shown in Fig. 3-7.

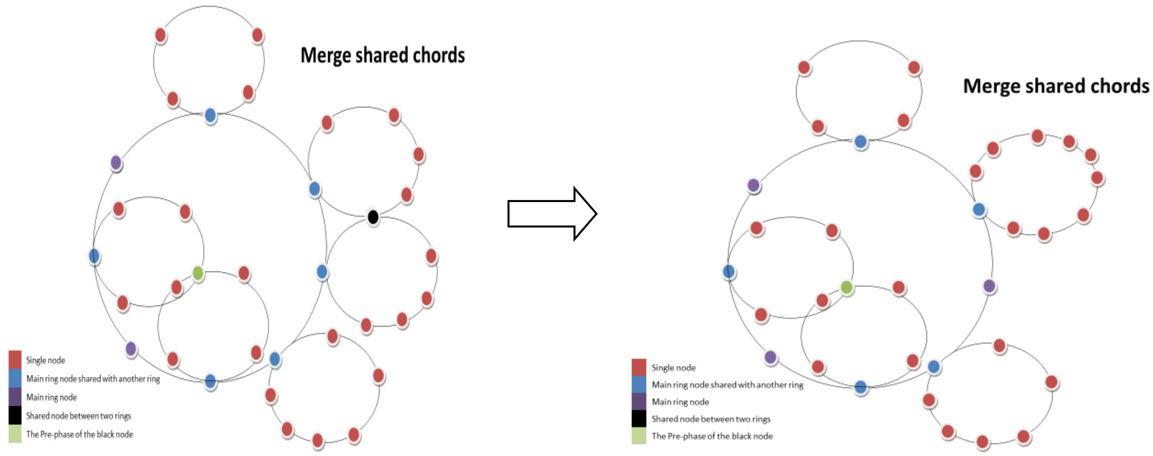



***Figure 3-7 Merge Shared Chords***

## 3.4. Logical Representation

The logical representation for the system model is clearly presented in Fig. 3-8. Accordingly, the Finger tables and chord tables are calculated for each node in the network as shown in Fig. 3-9

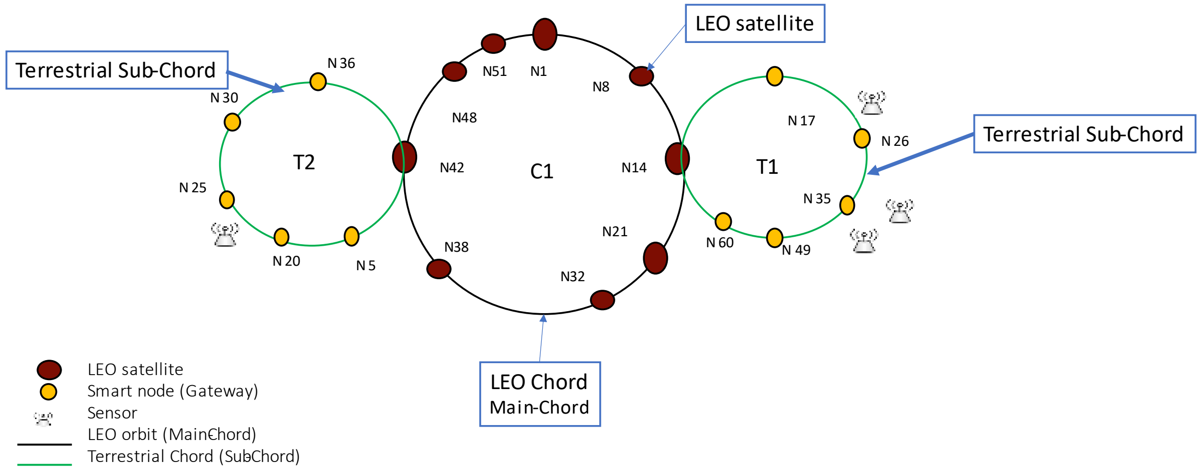



***Figure 3-8 Multi-chord logical representation***

**N1**

Finger table

| | |
|---|---|
| N1+1 | N8 |
| N1+2 | N8 |
| N1+4 | N8 |
| N1+8 | N14 |
| N1+16 | N21 |
| N1+32 | N32 |

Chord Table

| | |
|---|---|
| N8 | C1 |
| N14 | C1,T1 |
| N21 | C1 |
| N38 | C1 |

**N8**

Finger table

| | |
|---|---|
| N8+1 | N14 |
| N8+2 | N14 |
| N8+4 | N14 |
| N8+8 | N21 |
| N8+16 | N32 |
| N8+32 | N42 |

Chord Table

| | |
|---|---|
| N14 | C1,T1 |
| N21 | C1 |
| N32 | C1 |
| N42 | C1,T2 |

**N14**

Finger table

| | |
|---|---|
| N14+1 | N21 |
| N14+2 | N21 |
| N14+4 | N21 |
| N14+8 | N32 |
| N14+16 | N32 |
| N14+32 | N48 |

Chord Table

| | |
|---|---|
| N21 | C1 |
| N32 | C1 |
| N48 | C1 |

**N21**

Finger table

| | |
|---|---|
| N21+1 | N32 |
| N21+2 | N32 |
| N21+4 | N32 |
| N21+8 | N32 |
| N21+16 | N38 |
| N21+32 | N1 |

Chord Table

| | |
|---|---|
| N32 | C1 |
| N38 | C1 |
| N1 | C1 |

**N32**

Finger table

| | |
|---|---|
| N32+1 | N38 |
| N32+2 | N38 |
| N32+4 | N38 |
| N32+8 | N42 |
| N32+16 | N48 |
| N32+32 | N1 |

Chord Table

| | |
|---|---|
| N38 | C1 |
| N42 | C1,T2 |
| N48 | C1 |
| N1 | C1 |

**N38**

Finger table

| | |
|---|---|
| N38+1 | N42 |
| N38+2 | N42 |
| N38+4 | N42 |
| N38+8 | N48 |
| N38+16 | N1 |
| N38+32 | N8 |

Chord Table

| | |
|---|---|
| N42 | C1,T2 |
| N48 | C1 |
| N1 | C1 |
| N8 | C1 |

**N48**

Finger table

| | |
|---|---|
| N48+1 | N51 |
| N48+2 | N51 |
| N48+4 | N1 |
| N48+8 | N1 |
| N48+16 | N1 |
| N48+32 | N21 |

Chord Table

| | |
|---|---|
| N51 | C1 |
| N1 | C1 |
| N21 | C1 |

**N51**

Finger table

| | |
|---|---|
| N51+1 | N1 |
| N51+2 | N1 |
| N51+4 | N1 |
| N51+8 | N1 |
| N51+16 | N8 |
| N51+32 | N21 |

Chord Table

| | |
|---|---|
| N1 | C1 |
| N8 | C1 |
| N21 | C1 |

**N42**

Finger table

| | |
|---|---|
| N42+1 | N5 |
| N42+2 | N5 |
| N42+4 | N5 |
| N42+8 | N5 |
| N42+16 | N5 |
| N42+32 | N20 |

Chord Table

| | |
|---|---|
| N5 | T2 |
| N20 | T2 |

**N49**

Finger table

| | |
|---|---|
| N49+1 | N60 |
| N49+2 | N60 |
| N49+4 | N60 |
| N49+8 | N60 |
| N49+16 | N14 |
| N49+32 | N17 |

Chord Table

| | |
|---|---|
| N60 | T1 |
| N14 | T1,C1 |
| N17 | T1 |

**N17**

Finger table

| | |
|---|---|
| N17+1 | N26 |
| N17+2 | N26 |
| N17+4 | N26 |
| N17+8 | N26 |
| N17+16 | N35 |
| N17+32 | N49 |

Chord Table

| | |
|---|---|
| N26 | T1 |
| N35 | T1 |
| N49 | T1 |

**N20**

Finger table

| | |
|---|---|
| N20+1 | N25 |
| N20+2 | N25 |
| N20+4 | N25 |
| N20+8 | N30 |
| N20+16 | N36 |
| N20+32 | N5 |

Chord Table

| | |
|---|---|
| N25 | T2 |
| N30 | T2 |
| N36 | T2 |
| N5 | T2 |

***Figure 3-9 Finger and Chord tables***

## 3.5. Common Scenarios

Many common scenarios can be repeatedly occurred in satellite multi-chord architecture, in the following section some of the main scenarios will be illustrated.

### 3.5.1. Lookup Scenario

In this scenario the user using the laptop is connected to node 49 (N49) as shown in Fig. 3-10 and is asking about a certain information which is a student' ID. By hashing the required information, we get that the key we need to lookup is key 15 (K15).
N49 is responsible to lookup K15 as shown in the below example:

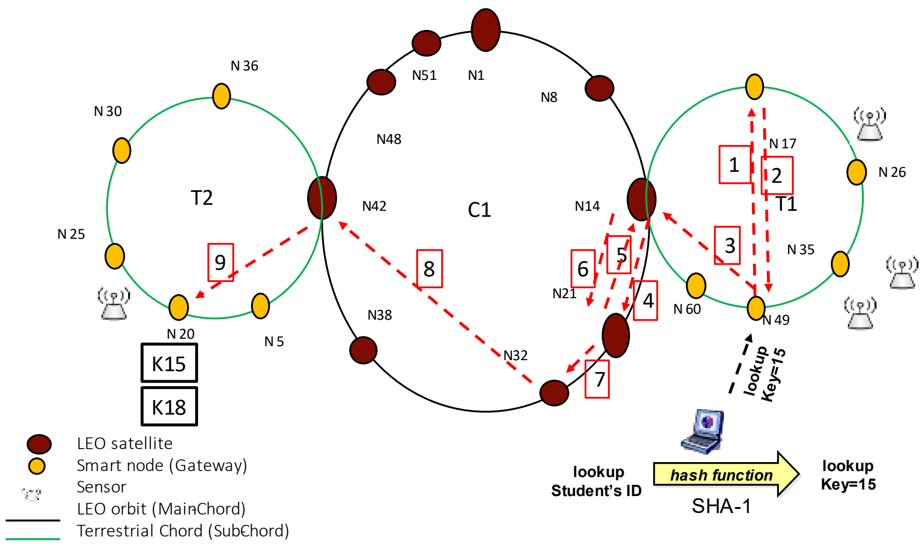



***Figure 3-10 Lookup example***

1. According to the finger table of N49, N49 lookup K15 at N17
2. N17 replied to N49 that it does not have k15
3. N49 will start to lookup K15 outside its sub-chord (T1). From chord table of N49, N49 will ask N14 to search for K15
4. Based on finger table of N14, N14 asks N21 about K15
5. N21 returned to N14 informing it that it doesn't has K15
6. Then N21 will lookup K15 outside C1, but N14 doesn't have another chord in its chord table. N14 asks its successor (N21) to lookup K15 outside chord C1
7. From the chord table of N21, N21 can't communicate with another chords. So N21 will ask its successor (N32)
8. Based on chord table of N32, N32 asks N42 to lookup K15 in T2.
9. Finally, N42 lookup K15 at N20 based on its finger table

### 3.5.2. Node Join

In this scenario presented in Fig. 3-11, a new node N40 will join terrestrial sub-chord T1. The following steps will illustrate the steps of node joining and accordingly the stabilization protocol.

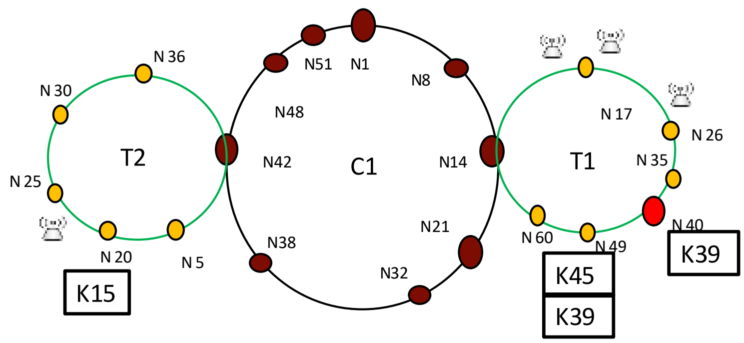



***Figure 3-11 Node join example***

1. N40 joins the system
2. N40 aquires N49 as its successor
3. N40 notifies N49
4. N49 aquires N40 as its predecessor
5. N40 copies keys
6. N35 runs stabilization and asks its successor, N49 for its predecessor which is N40.
7. N35 aquires N40 as its successor
8. N35 notifies N40 of its existence
9. N40 aquires N35 as predecessor

### 3.5.3. Relative Motion Scenario

Before the smart device requests a service, it needs to establish a continuous connection with the LEO satellite [62]. Since the LEO satellite moves at a high-speed relative to the smart device, there are issues of access switching and task delivery.

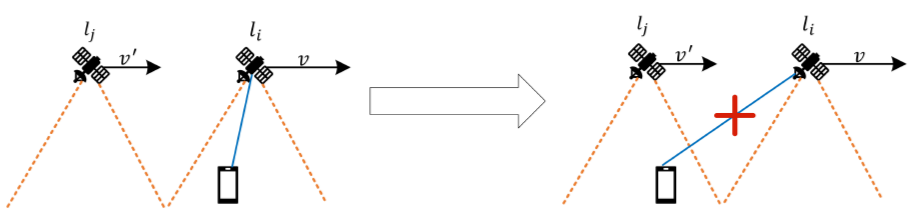


***Figure 3-12 Relative motion example***

As shown in Fig. 3-12, the smart device first establishes connection with the satellite$l_i$, submits a request. However, continuous movement of the satellite $l_i$ at a relative speed v causes the smart device to be disconnected because it is not within the coverage of the satellite signal.

To continuously obtain service, smart devices will involve access handoff, that is, re-access the next LEO satellite $l_j$ whose signal covers the smart devices.

The following Fig. 3-13 and Fig. 3-14 illustrates the relative motion scenario at the multi-chord and how this issue managed in our research.

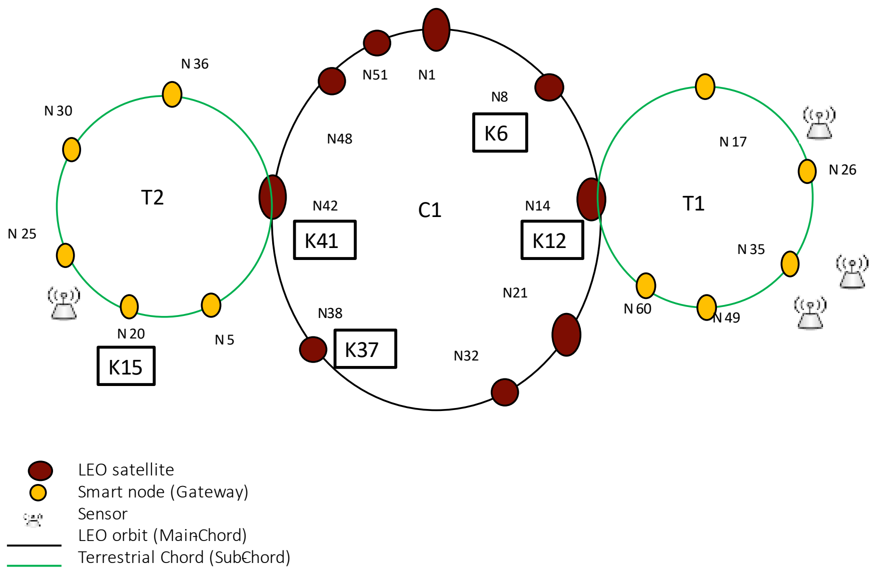



*Figure 3-13 Multi-chord before relative motion scenario*

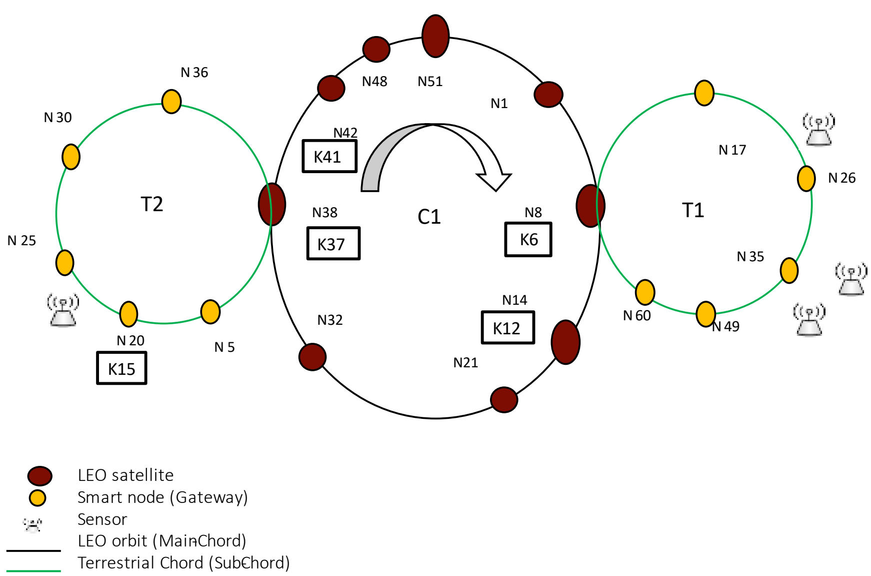



*Figure 3-14 Multi-chord after relative motion scenario*

# Chapter 4

## 4. Mathematical model

We modeled the multi-chord P2P network as a stochastic process where nodes in the sub-chord are modeled as states in a discrete Markov chain [63] and the operations of lookup-delete-add and modify of keys in a sub-chord are modeled as transitions on the discrete Markov chain. A stochastic process is defined to be an indexed collection of random variables $\{X_t\}$, where the index *t* runs through a given set *T*. *T* is taken to be the set of non-negative integers, and $X_t$ represents a measurable characteristic of interest at time *t*.

### 4.1. Markov Chain

Markov chain is a special kind of stochastic process [64] where it has the special property that probabilities involving how the process will evolve in the future depend only on the present state of the process and are independent of events in the past. Thus, it differs from a general stochastic process in that a Markov chain must be "memory-less [65].

A Markov chain is a mathematical model that explains probabilistically the transitions from one state to another according to certain probabilistic rules
Assumptions are necessary regarding the joint distribution of $X_0$, $X_1$, . . . to obtain analytical results. One assumption that leads to analytical tractability is that the stochastic process is a Markov chain, which has the following key property:
A stochastic process $\{X_t\}$ is said to have the Markovian property if P$\{X_{t+1} = j \mid X_0 = k_0$, $X_1$= $k_1$, . . . , $X_{t-1}$= $k_{t-1}$, $X_t$ = *i*}, for t = 0, 1, . . . and every sequence i, j, $k_0$, $k_1$, . . . , $k_{t-1}$.
In words, this Markovian property states that the conditional probability of any future "event," given any past "event" and the present state $X_t = i$, is independent of the past event and depends only upon the present state.

It is worth mentioning that the conditional probabilities P$\{X_{t+1} = j \mid X_t = i\}$ for a Markov chain are called (one-step) transition probabilities if, for each i and j

P$\{X_{t+1} = j \mid X_t = i\}$ = P$\{X_t = j \mid X_0 = i\}$ for all t = 1, 2, . . . ,

*(4-1)*

And thus the (one-step) transition probabilities are said to be stationary implying that the transition probabilities do not change over time. The existence of stationary transition probabilities also implies that, for each i, j, and n (n = 0, 1, 2, . . .),

P$\{X_{t+n} = j \mid X_t = i\}$ = P$\{X_n = j \mid X_0 = i\}$ for all t = 0, 1, . . . .

*(4-2)*

These conditional probabilities are called n-step transition probabilities.
To simplify notation with stationary transition probabilities, let

$$P_{ij} = P\{X_{t+1} = j \mid X_t = i\} \tag{4-3}$$

$$P_{ij}^{(n)} = P\{X_{t+n} = j \mid X_t = i\} \tag{4-4}$$

Thus, the n-step transition probability $P_{ij}^{(n)}$ is just the conditional probability that the system will be in state *j* after exactly n steps (time units), given that it starts in state *i* at any time *t*. When n = 1, note that $P_{ij}^{(n)} = P_{ij}$

Since $P_{ij}^{(n)}$ are conditional probabilities, they must be non-negative, and since the process must make a transition into some state, they must satisfy the properties

$P_{ij}^{(n)} \geq 0$, for all *i* and *j*; *n* = 0, 1, 2, . . . ,

and

$\sum_{j=0}^{M} P_{ij}^{(n)} = 1$ for all i; *n* = 0, 1, 2, . . . .

## 4.2. Classification of Markov Chain Model

Transition probabilities associated with the states is crucial factor in the behavior of Markov chains. To further describe the properties of Markov chains, it is necessary to present some concepts and definitions concerning these states [66].

State *j* is said to be accessible from state *i* if $P_{ij}^{(n)} > 0$ for some $n \geq 0$. Thus, state *j* being accessible from state *i* means that it is possible for the system to enter state *j* eventually when it starts from state *i.* For example, $P_{ij}^{(2)} > 0$ for all *i* and *j*, means every state is accessible from every other state. In general, a sufficient condition for all states to be accessible is that there exists a value of *n* for which $P_{ij}^{(n)} > 0$ for all *i* and *j*.

If state j is accessible from state i and state *i* is accessible from state *j*, then states *i* and *j* are said to communicate.

**Recurrent States and Transient States:** It is often useful to know if a process entering a state will ever return to this state. A state is also said to be a transient state if, upon entering this state, the process may never return to this state again [67]. Therefore, state *i* is transient if and only if there exists a state *j* ($j \neq i$) that is accessible from state *i* but not vice versa, that is, state *i* is not accessible from state j.

Thus, if state *i* is transient and the process visits this state, there is a positive probability (perhaps even a probability of 1) that the process will later move to state *j* and so will never return to state *i.*

A state is said to be a recurrent state if, upon entering this state, the process will return to this state again. Therefore, a state is recurrent if and only if it is not transient. Since a recurrent state will be revisited after each visit, it will be visited infinitely often if the process continues forever. A state is said to be an absorbing state if, upon entering this state, the process never will leave this state again. Therefore, state *i* is an absorbing state if and only if $P_{ij} = 1$.
A Markov transition matrix [68] is a square matrix showing the probabilities of going from one state to another where each row in the matrix represents an initial state and each column represents a terminal state.

$$p_{ij} = \mathrm{Prob}(\text{ State } n+1 \text{ is } S_i \mid \text{State } n \text{ is } S_j),$$

$$P = \begin{bmatrix} p_{11} & p_{12} & \cdots & p_{1r} \\ p_{21} & p_{22} & \cdots & p_{2r} \\ \vdots & & \ddots & \\ p_{r1} & & & p_{rr} \end{bmatrix}$$

The main properties of the transition matrix are:

- It is square matrix
- All entries are between 0 and 1, because all entries represent probabilities.
- The sum of the entries in any row must be 1

Discrete Time Markov chain is commonly used in P2P network topologies. As authors in [69] used Markov to deal with the time dependencies of consecutive time steps to propose Time-Evolving Graph (TEG) to capture the evolution of the connectivity of P2P networks. Also, authors in [70] were concerned by the deployment of blockchain in IoT, a three-layer architecture includes smart devices layer, edge cloud layer and blockchain-based distributed cloud layer. The authors proposed a content selection algorithm of edge cache nodes, and the algorithm adopts Markov chain model.

Markov chain can be used in modelling all chord operations such as node joining, node leaving, lookup and storing because all these operations are probabilistic and Markov chain may be the appropriate model for these operations.
Hence, the state in our work will refer to entry of our finger table in case of lookup a key (data), node joining, node leaving and stabilization process. Our work will be modelled as a Discrete Time Markov Chain (DTMC) $\{t_1, t_2, t_3, \ldots\}$
From the Markov chain, performance of multi-chord approach in terms of the number of hops to finish operations, time to response, load distribution and response to failure will be measured using Markov chain steady state equation.

The term SteadyState probability means that the probability of finding the process in a certain state, say *j*, after many transitions tends to the value *j*, independent of the probability distribution of the initial state. It is important to note that the steady-state probability does not imply that the process settles down into one state. On the contrary, the process continues to make transitions from state to state, and at any step *n* the transition probability from state *i* to state *j* is still $P_{ij}$.

The $\pi_j$ array is be interpreted as stationary probabilities (not to be confused with stationary transition probabilities) in the following sense. If the initial probability of being in state *j* is given by $\pi_j$ (that is, P{$x_0$ = *j*} = $\pi_j$) for all *j*, then the probability of finding the process in state *j* at time *n* = 1, 2, . . . is also given by $\pi_j$ (that is, P{$x_n$ = *j*} = $\pi_j$.

To elaborate more how Markov will be used in chord, let's have an example of m equals to 3 bits shown in Fig. 4-1.

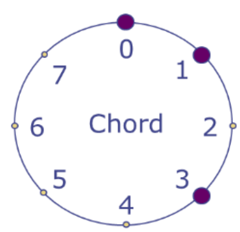



***Figure 4-1 Chord ring with m=3***

Then the finger tables of nodes n0, n1 and n3 will be as shown below in Fig. 4-2:

| Finger Table of n0 |
|---|
| n0+1 → n1 |
| n0+2 → n3 |
| n0+4 → n0 |

| Finger Table of n1 |
|---|
| n1+1 → n3 |
| n1+2 → n3 |
| n1+4 → n0 |

| Finger Table of n3 |
|---|
| n3+1 → n0 |
| n3+2 → n0 |
| n3+4 → n0 |

***Figure 4-2 Finger tables of nodes n0, n1 and n3***

Then we can draw the transition diagram as shown in Fig. 4-3:

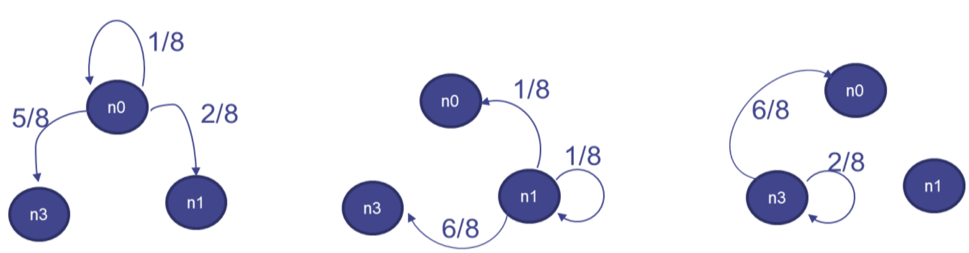
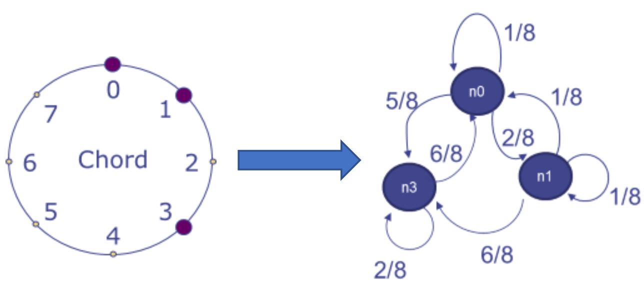



***Figure 4-3 Transition diagrams Chord ring with m=3***

The transition probabilities are calculated from the finger table in terms of frequency of transition from state to another to lookup a certain key.

For example, to lookup a key from node $n_0$ to any other node in the network, the probability will be 0.125 from node $n_0$ to itself as it is only one key that will be looked up stored in node $n_0$ and probability 0.25 from node $n_0$ to node $n_1$ as there will be two keys that can be looked up from node $n_1$ and probability 0.625 from node $n_0$ to node $n_3$ as there will be five keys that can be looked up from node $n_3$

After that, the transition matrix can be easily written as below in Fig. 4-4.




**Transition Matrix:**

$$P = \begin{bmatrix} 1/8 & 2/8 & 5/8 \\ 1/8 & 1/8 & 6/8 \\ 6/8 & 0 & 2/8 \end{bmatrix}$$

***Figure 4-4 Transition Matrix extraction at n=3***

Another example will be illustrated when n6 joined the same chord ring. Then the chord ring will be as below in Fig. 4-5.

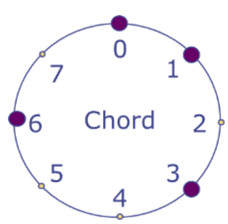



***Figure 4-5 Chord ring with n6***

Then the finger tables and transition diagrams of nodes n0, n1, n3 and n6 will be as shown in Fig. 4-6 and the transition diagrams in Fig. 4-7.

| Finger Table of n0 | Finger Table of n1 | Finger Table of n3 | Finger Table of n6 |
|---|---|---|---|
| n0+1 → n1 | n1+1 → n3 | n3+1 → n6 | n6+1 → n0 |
| n0+2 → n3 | n1+2 → n3 | n3+2 → n6 | n6+2 → n0 |
| n0+4 → n6 | n1+4 → n6 | n3+4 → n0 | n6+4 → n1 |

***Figure 4-6 Finger tables of nodes n0, n1, n3 and n6***

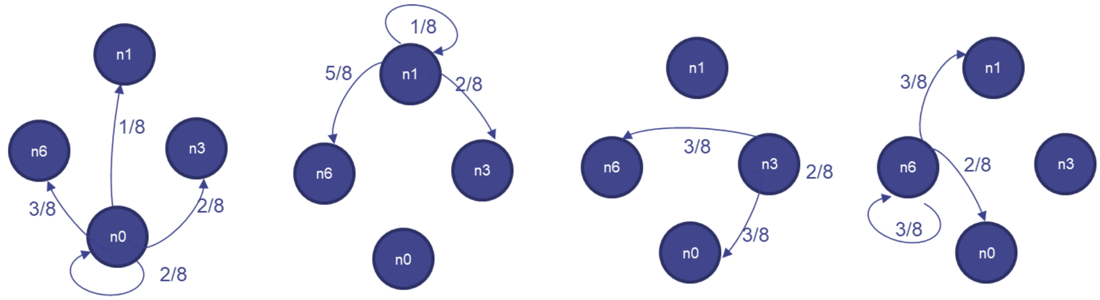



***Figure 4-7 Transition diagrams for Chord ring at n=4***

At last, the transition matrix will be as shown below in Fig. 4-8.

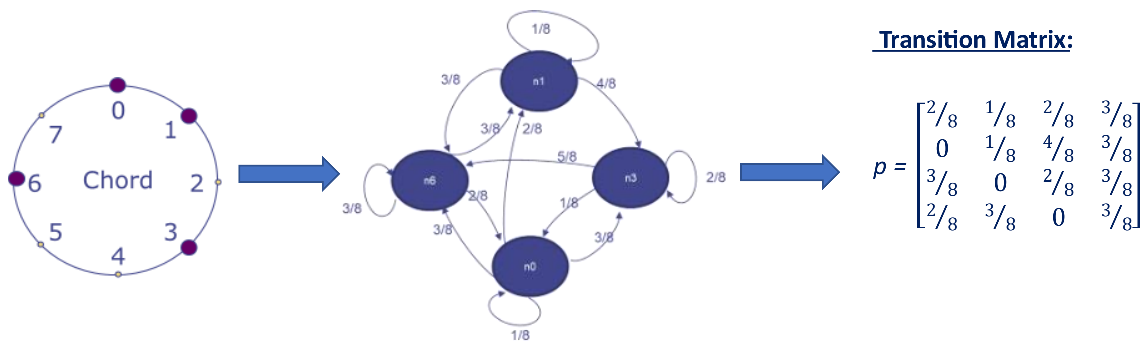



***Figure 4-8 Transition Matrix extraction at n=4***

## 4.3. General Markov Model

To model a saturated (all nodes are existing) fully loaded (all possible keys are considered) chord, we can get the below diagram presented in Fig. 4-9 where u = $n2^{m-1}$.

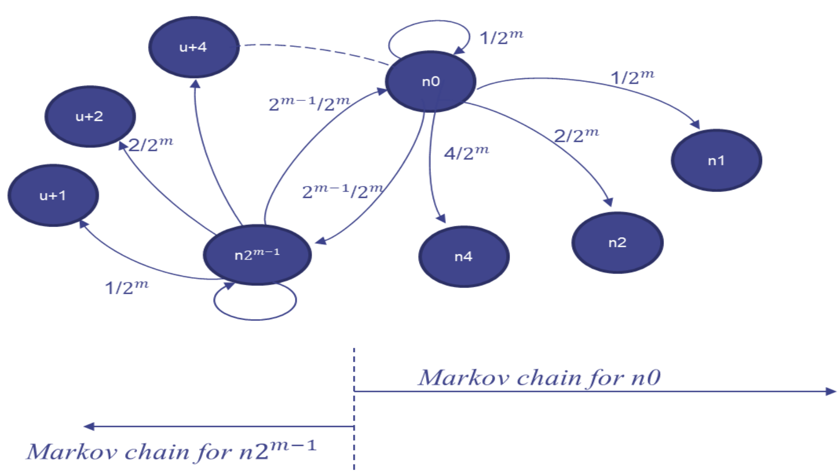



***Figure 4-9 General Markov model***

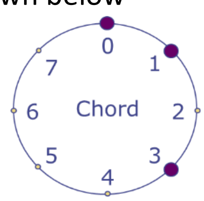
$$p = \begin{bmatrix} 1 & 1 & 2 & 0 & 4 & 0 & 0 & 0 & 8 & \cdots & 2^{m-1} & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots \\ 0 & 1 & 1 & 2 & 0 & 4 & 0 & 0 & 0 & 8 & \cdots & 2^{m-1} & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots \\ 0 & 0 & 1 & 1 & 2 & 0 & 4 & 0 & 0 & 0 & 8 & \cdots & 2^{m-1} & 0 & 0 & 0 & \cdots & \cdots & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\ 0 & 0 & 0 & 0 & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & \cdots & 2^{m-1} \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & 2^{m-1} & \vdots \end{bmatrix} \times \frac{1}{2^m} \tag{4-5}$$

where the matrix dimensions are $2^m \times 2^m$.

## 4.4. Steady State Analysis

Modeling of the steady state analysis s to analytically derive the probability state vector of Satellite IoT consists of multi-chord networks is a prior requirement to do the system analysis [71,72]. This study presents a steady-state Markov chain model to predict the long-term probability of drought conditions.

Suppose a chord ring of m = 3 bit as shown below




U (p - 1) = 0 (using eigen vector method).
Let x stands for n0, y stands for n1, and z stands for n3.

$$[x \quad y \quad z] \times \left( \begin{bmatrix} 1/8 & 2/8 & 5/8 \\ 1/8 & 1/8 & 6/8 \\ 6/8 & 0 & 2/8 \end{bmatrix} - \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \right) = 0 \tag{4-6}$$

$$[x \quad y \quad z] \times \begin{bmatrix} -7/8 & 2/8 & 5/8 \\ 1/8 & -7/8 & 6/8 \\ 6/8 & 0 & -6/8 \end{bmatrix} = 0 \tag{4-7}$$

$$-{}^{7}/_{8}\,x + {}^{1}/_{8}\,y + {}^{6}/_{8}\,z = 0$$

*(4-8)*

$${}^{2}/_{8}\,x - {}^{7}/_{8}\,y = 0$$

*(4-9)*

$${}^{5}/_{8}\,x + {}^{6}/_{8}\,y - {}^{6}/_{8}\,z = 0$$

*(4-10)*

Given that $\boldsymbol{x} + \boldsymbol{y} + \boldsymbol{z} = 1$

By solving those equations, we can get that:
x = ${}^{42}/_{101}$ , y = ${}^{12}/_{101}$ & z= ${}^{47}/_{101}$
Steady state vector: $\begin{bmatrix}\frac{42}{101} & \frac{12}{101} & \frac{47}{101}\end{bmatrix}$

It presents the probability of existence of those nodes. The steady state vector will be used to find the expected number of hops for any key in a multi-chord approach. To find tis expected number of lookups, we must find all possible combinations of nodes in a chord. This can be achieved by binomial distribution.

Binomial Distribution

$$\binom{N}{l} = \frac{N!}{l!\,(N-l)!}$$

$\binom{N}{l}$: number of alternatives representations of k nodes from total n nodes
N: $2^{m}$
l: actual available nodes

| **l** | **0** | **1** | **2** | **3** | **4** | **5** | **6** | **7** | **8** |
|---|---|---|---|---|---|---|---|---|---|
| $\binom{N}{l}$ | 1 | 8 | 28 | 56 | 70 | 56 | 28 | 8 | 1 |

***Table 4-1 Binomial Distribution***

PMF: $\binom{N}{l}\,p^{l}(1-p)^{N-l}$
Mean = N × p
p: probability to choose a node. (Assume p equals half)
q: probability to not choose a node (assume q equals half)

The following cases will clarify the steady state analysis:

**Case 1:**

*m = 3, N=8 & l=8*

*All 8 nodes exist*

$$\begin{bmatrix} 1 & 1 & 2 & 0 & 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 & 1 & 2 & 0 & 4 \\ 4 & 0 & 0 & 0 & 1 & 1 & 2 & 0 \\ 0 & 4 & 0 & 0 & 0 & 1 & 1 & 2 \\ 2 & 0 & 4 & 0 & 0 & 0 & 1 & 1 \\ 1 & 2 & 0 & 4 & 0 & 0 & 0 & 1 \end{bmatrix} \times \frac{1}{8}$$

*(4-11)*

Steady state vector: $[1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1 \quad 1] \times \frac{1}{8}$

**Case 2:**

*m = 3, N=8 & l=7*

*7 nodes exist (n7 doesn't exist)*

$$\begin{bmatrix} 1 & 1 & 2 & 0 & 4 & 0 & 0 \\ 0 & 1 & 1 & 2 & 0 & 4 & 0 \\ 0 & 0 & 1 & 1 & 2 & 0 & 4 \\ 3 & 0 & 0 & 1 & 1 & 3 & 0 \\ 4 & 0 & 0 & 0 & 1 & 1 & 2 \\ 1 & 4 & 0 & 0 & 0 & 1 & 2 \\ 2 & 0 & 4 & 0 & 0 & 0 & 2 \end{bmatrix} \times \frac{1}{8}$$

*(4-12)*

Steady state vector: $[196 \quad 81 \quad 191 \quad 50 \quad 174 \quad 93 \quad 216] \times \frac{1}{1000}$

**Case 3:**

*m = 3, N=8 & l=7*

*7 nodes exist (n3 doesn't exist)*

$$\begin{bmatrix} 1 & 1 & 2 & 4 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 4 & 0 & 0 \\ 0 & 0 & 1 & 2 & 0 & 4 & 0 \\ 3 & 0 & 0 & 1 & 3 & 0 & 0 \\ 4 & 0 & 0 & 1 & 1 & 2 & 2 \\ 1 & 4 & 0 & 0 & 1 & 2 & 1 \\ 2 & 0 & 4 & 0 & 0 & 2 & 1 \end{bmatrix} \times \frac{1}{8}$$

*(4-13)*

Steady state vector: $[174 \quad 93 \quad 216 \quad 196 \quad 81 \quad 191 \quad 50] \times \frac{1}{1000}$

**Case 4:**

*m = 3, N=8 & l=6*

*6 nodes exist (n6 & n7 don't exist)*

$$\begin{bmatrix} 3 & 1 & 2 & 0 & 2 & 0 \\ 0 & 1 & 1 & 2 & 0 & 4 \\ 1 & 0 & 1 & 1 & 4 & 0 \\ 3 & 0 & 0 & 1 & 1 & 3 \\ 4 & 0 & 0 & 0 & 1 & 3 \\ 3 & 4 & 0 & 0 & 0 & 1 \end{bmatrix} \times \frac{1}{8} \quad (4\text{-}14)$$

Steady state vector: $[311 \quad 152 \quad 110 \quad 75 \quad 163 \quad 189] \times \frac{1}{1000}$

**Case 5:**

*m = 3, N=8 & l=6*

*6 nodes exist (n1 & n4 do not exist)*

$$\begin{bmatrix} 1 & 4 & 0 & 3 & 0 & 0 \\ 0 & 2 & 2 & 4 & 3 & 0 \\ 0 & 0 & 1 & 3 & 0 & 4 \\ 0 & 2 & 0 & 2 & 1 & 3 \\ 2 & 4 & 0 & 0 & 1 & 1 \\ 2 & 1 & 4 & 0 & 0 & 1 \end{bmatrix} \times \frac{1}{8} \quad (4\text{-}15)$$

Steady state vector: $[92 \quad 235 \quad 178 \quad 174 \quad 126 \quad 195] \times \frac{1}{1000}$

**Case 6:**

*m = 3, N=8 & l=6*

*6 nodes exist (n2 & n6 don't exist)*

$$\begin{bmatrix} 1 & 2 & 1 & 4 & 0 & 0 \\ 0 & 1 & 3 & 0 & 4 & 0 \\ 0 & 0 & 2 & 1 & 2 & 3 \\ 4 & 0 & 0 & 1 & 2 & 1 \\ 0 & 4 & 0 & 0 & 1 & 3 \\ 1 & 2 & 3 & 0 & 0 & 2 \end{bmatrix} \times \frac{1}{8} \quad (4\text{-}16)$$

Steady state vector: $[185 \quad 193 \quad 222 \quad 80 \quad 197 \quad 223] \times \frac{1}{1000}$

**Case 7:**

*m = 3, N=8 & l=5*

*5 nodes exist (n5, n6 & n7 don't exist)*

$$\begin{bmatrix} 4 & 1 & 2 & 0 & 1 \\ 1 & 1 & 1 & 5 & 0 \\ 2 & 0 & 1 & 1 & 4 \\ 3 & 0 & 0 & 1 & 4 \\ 7 & 0 & 0 & 0 & 1 \end{bmatrix} \times \frac{1}{8} \tag{4-17}$$

Steady state vector: $[501 \quad 72 \quad 153 \quad 73 \quad 201] \times \frac{1}{1000}$

**Case 8:**

*m = 3, N=8 & l=5*

*5 nodes exist (n2, n4 & n7 don't exist)*

$$\begin{bmatrix} 2 & 2 & 2 & 2 & 0 \\ 0 & 2 & 3 & 4 & 0 \\ 2 & 0 & 2 & 3 & 0 \\ 1 & 3 & 0 & 2 & 2 \\ 4 & 0 & 3 & 0 & 1 \end{bmatrix} \times \frac{1}{8} \tag{4-18}$$

Steady state vector: $[167 \quad 226 \quad 201 \quad 341 \quad 65] \times \frac{1}{1000}$

This clearly shows that there is no unified mathematical closed form to find the steady state of nodes in a multi-chord approach and thus numerical analysis will be used in our simulations to find steady state of different situations.

# Chapter 5

## 5. Simulation Results

We will present in this chapter the simulation results that evaluate the performance of the multi-chord P2P architecture. we carried out three simulation scenarios, involving: (1) Key lookup and node joining on multiple chords to evaluate the complexity and prove the scalability of the proposed multi-chord architecture; (2) Node joining operation on multi-chord architecture to demonstrate its operability as well as to examine cost; (3) Key lookup on multi-ring architecture to evaluate cost of key finding in various cases and properties.

As the proposed multi-chord architecture is difficult to evaluate on real testbed with full hardware and software, we performed those tests through simulations on discrete event simulator platform. For each test, we used several scenarios that test certain characteristics in the multi-chord approach such as effect of number of nodes in the system to gain generality. In addition, the number of hops is considered as main metrics in our evaluation, and we also did not consider the communication costs among nodes at the physical layer as it is out of scope of this dissertation.

The main performance indicators that we will be evaluated in this chapter are:

- Response time is defined as the time a query for a lookup will take from the time it is requested by a node until a reply from destination node is received and is measured in seconds and is evaluated according to equation 5-1.
- Mean time of node response is the statistical average time a query takes for looking up a key.
- Variance of time response is the statistical deviation encountered by queries for looking up keys.
- The mean time of node response and the variance of time response are both evaluated numerically as they are based on equation 4-5 that has no closed form.
- Time of network stabilization is the time a chord or multi-chord takes to modify the finger tables and keys in nodes when nodes join or leave a chord or a multi-chord
- Steady state probability of a node in a multichord is the probability of a given node in a chord to be responsible for performing a query in the network and it is evaluated based on equation 4-5
- Outdegree of a multi-chord is defined as the average number of connections going out from a node to other nodes
- Indegree of a multi-chord is defined as the average number of connections going into a node from other nodes in network.
- Both outdegree and indegree are evaluated numerically and their mathematical closed form are shown in section 5.4

- Pdf of shared node is defined as the probability density function of possibility of nodes in a chord to perform a certain query in the network and is evaluated based on the numerical histogram obtained from system simulation.

## **5.1.** Response Time to Key Lookup within a Single Sub-Chord

We modeled and simulated the process in one of the 32 starlink orbits that includes 64 satellite nodes.
Fig. 5-1 shows the constellation comprised of 64 satellite nodes in one of the 32 starlink orbits and the virtual connectivity between the satellites using the chord P2P networks. Each satellite node can reach six other satellites as proposed by the concept of finger table in the chord networks. This will guarantee that all satellites in each orbit will be reachable in a nearly mesh-like topology. For example, the satellite with ID=7 will have the finger table shown in the Fig. 5-1. We then started simulating the response time the satellite will take to respond to request for data stored in the multi-chord approach.

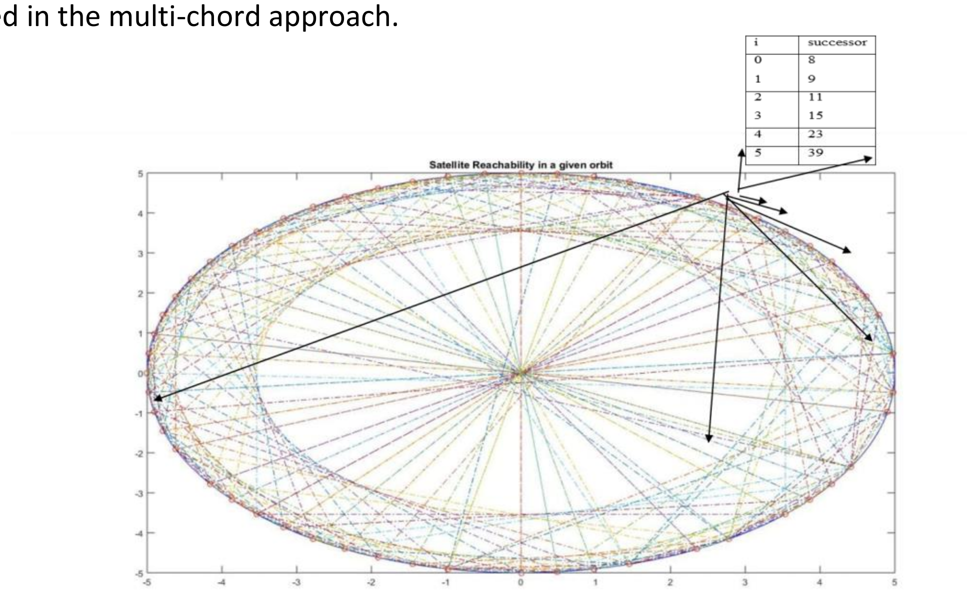



***Figure 5-1 Virtual connectivity between satellites in each orbit***

We investigated the response time in this multi-chord architecture where the response time is measured without taking the propagation time into consideration as this will be function to the orbit distance from the Earth and some physical characteristics that are out of scope of this dissertation. Thus, the represented results are the processing and queuing time within nodes or between nodes within same or different sub-chords.
Figure 5-2 shows the mean time of node response to lookup requests as a function of the number of nodes in a sub-chord. Figure 5-2 also shows that the average response is around 0.5 msec which is considered very adequate to high speed IoT networks for all variations of satellite numbers per orbit. This is contributed to the fact that mean time to access data will follow a complexity in order of log (n) as will be shown in further simulation figures.

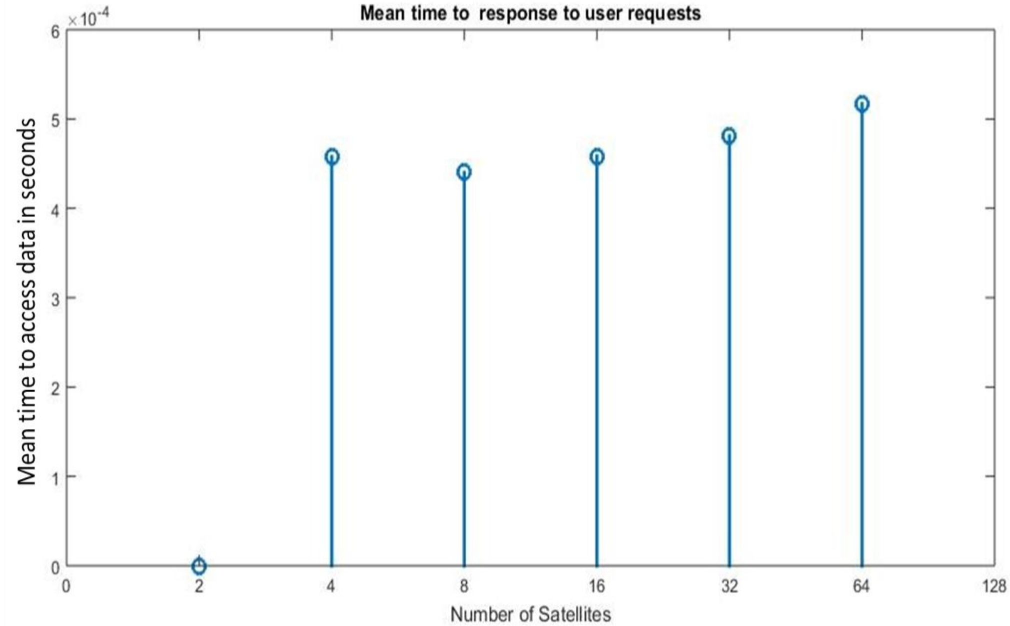



*Figure 5-2 Mean time in sec to access data within an orbit*

To quantify the results in Fig. 5-3, we measured the variance in such time as function of number of nodes in multi-chord approach as shown in Fig. 5-3. Although the average time to access data within a sub-chord is nearly the same as the number of nodes increases in the sub-chord, the variance of such time is linearly dependent on the number of nodes in a sub-chord as seen in Fig. 5-3.

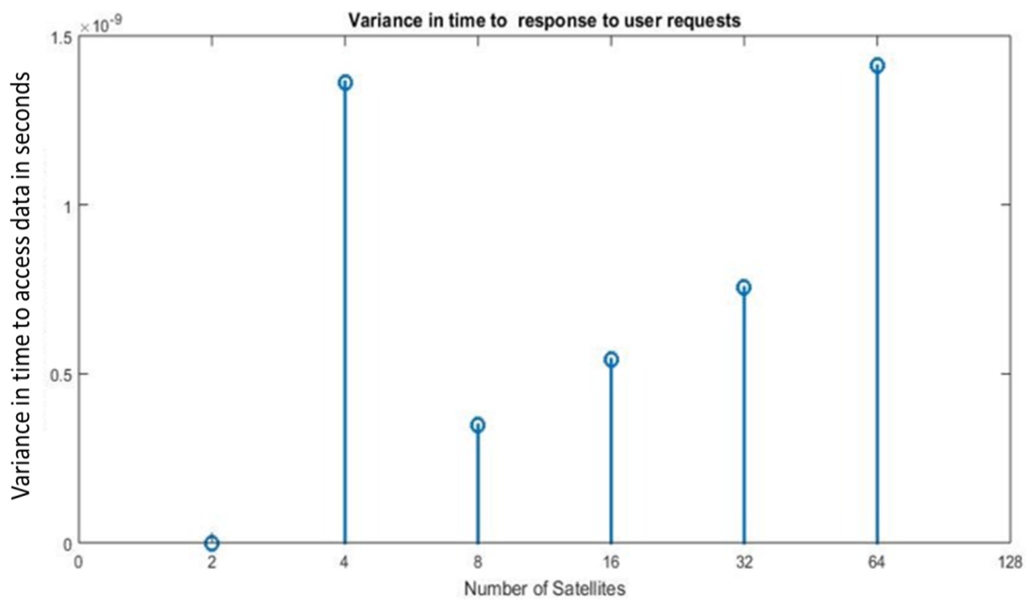



*Figure 5-3 Variance in time to access data within an orbit*

This can be contributed to the fact that data will be distributed on larger number of nodes in sub-chord architecture and thus expected to take more time for lookup and retrieval in some cases.

On the other hand, Fig. 5-4 shows the time taken to stabilize from point of data storage and retrieval when a node or group of nodes fail. It also shows the effect of node failures on the time needed to stabilize when amount of traffic data is varied. The figure shows that a time of nearly 4 to 5 seconds is needed to stabilize the network after a node failure, this time is also considered adequate in high speed IoT networks. However, statistics shown in Fig. 5-5 proves that 90% of the time required to stabilize a network will not exceed 4 seconds in the most vulnerable situation of 10 nodes in failure. The figure also shows that the difference between the 10 percentile and 90 percentiles is within acceptable range and is nearly showing a uniform distribution.

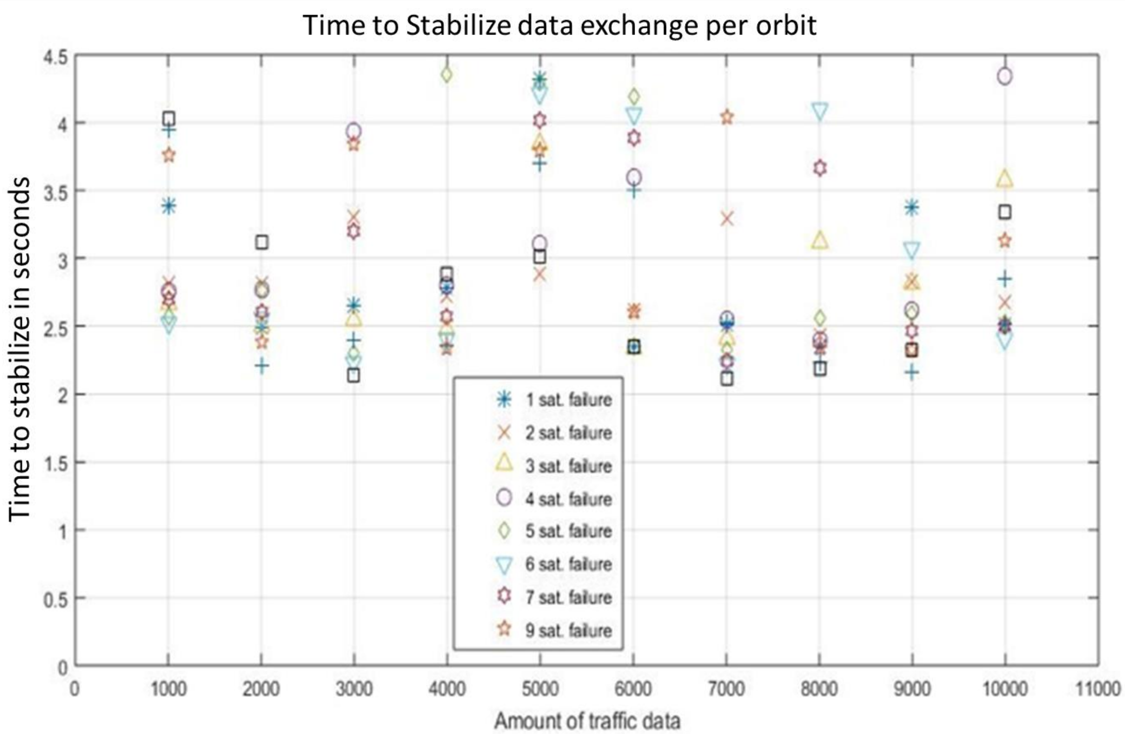



***Figure 5-4 Network stabilization time in sec vs number of satellites failures***

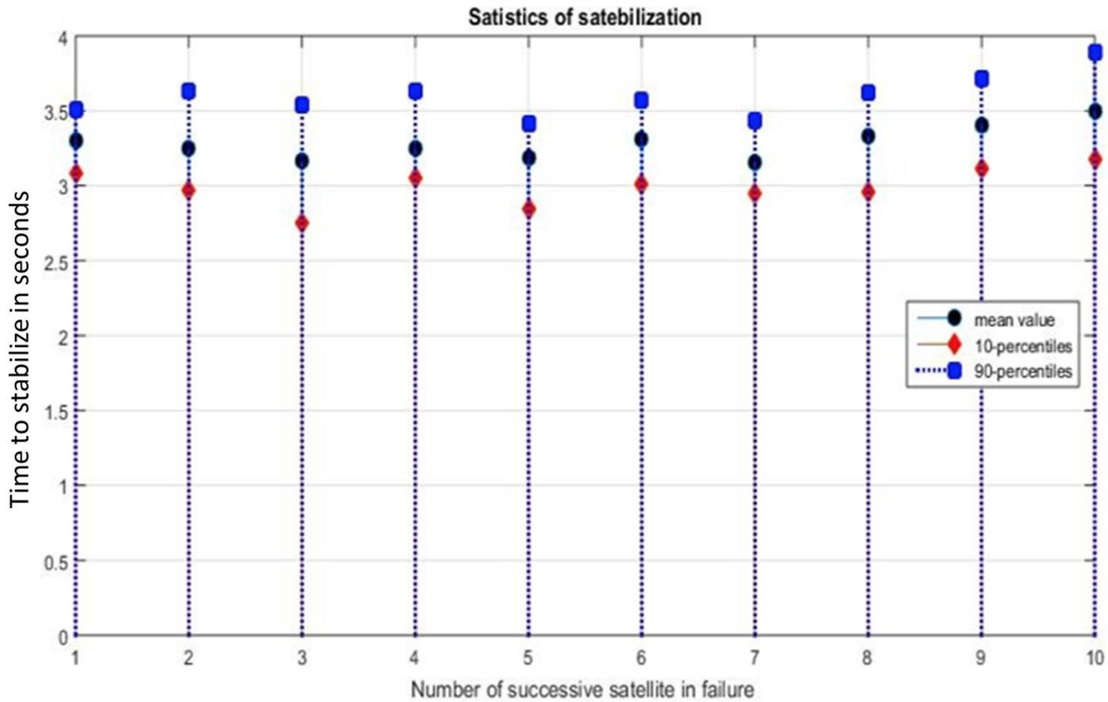



***Figure 5-5 Statistical values of network stabilization***

## 5.2. Response Time to Failure within a Single Sub-Chord

We also carried three tests to evaluate the ability of multi-chord satellite network to respond to failure in nodes listed in the successor list and the response of the multi-chord to failure in a sub-chord and not a node and lastly the ability of the system to keep the connections alive among different sub-chords in the system.
We carried our simulation model based on an abstract time unit. This time unit represents the time a node will need to detect failure in connections with one of its successors. The exact value of this time unit depends on the satellite systems under consideration and differs from one satellite networks to another. Fig. 5-6 shows the mean time to achieve stabilization if the nodes listed as successors suffer from failure. The simulation was carried in two situations. The first situation assumes the lookup in the successor list is done in order and the second situation when the lookup is done randomly.

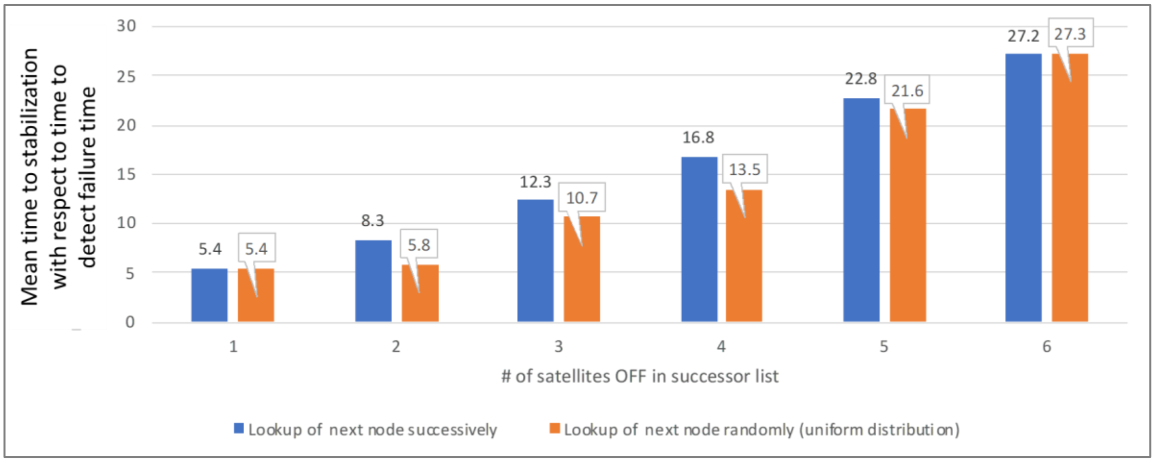



***Figure 5-6 Number of satellites OFF in successor list***

Figure 5-6 shows there is a nearly linear increase in the meantime with the number of nodes in failure in first situation compared to the situation when the lookup is done randomly. It also confirms that random lookup is superior when the number of failure nodes is low.

Figure 5-7 shows that the growth in mean time is nearly exponential in case the failure in the sub-chord as a whole and not in a node or group of nodes in a sub-chord. This due to increase in lookup time for an alternative sub-chord and a node in that alternative sub-chord that will be the merge node as explained in chapter 3.

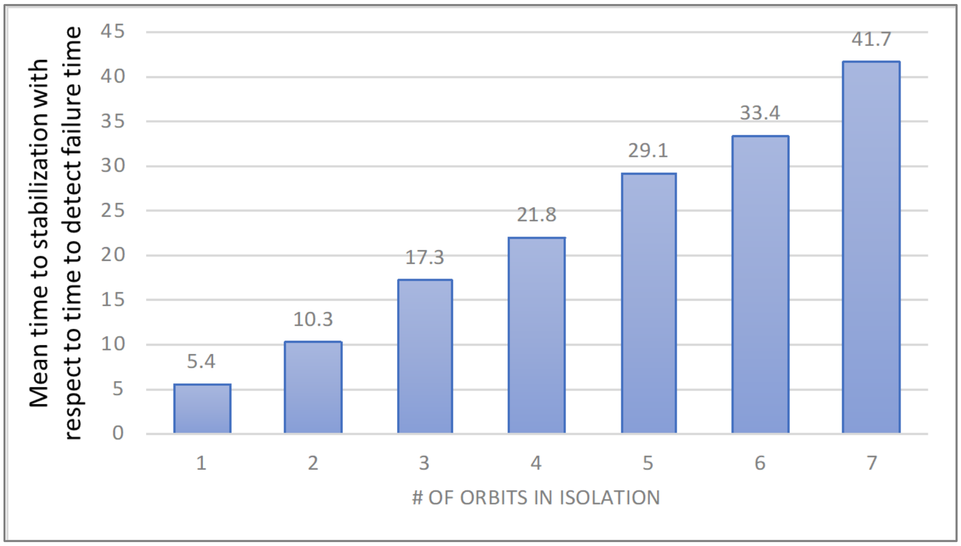



***Figure 5-7 Mean time to stabilization***

The last experiment is shown in Fig. 5-8 that measures the percentage of keeping the link alive vs the remaining time available in the **ON** session of a satellite. The figure shows that there is at least 50% chance of success to keep the link alive if the remaining time of on session is more than half its overall time. R represents the link connectivity live time between satellite ***i*** itself and its successor ***j***, while S represents the remaining time for satellite ***i*** to discover its successor before it goes **OFF**

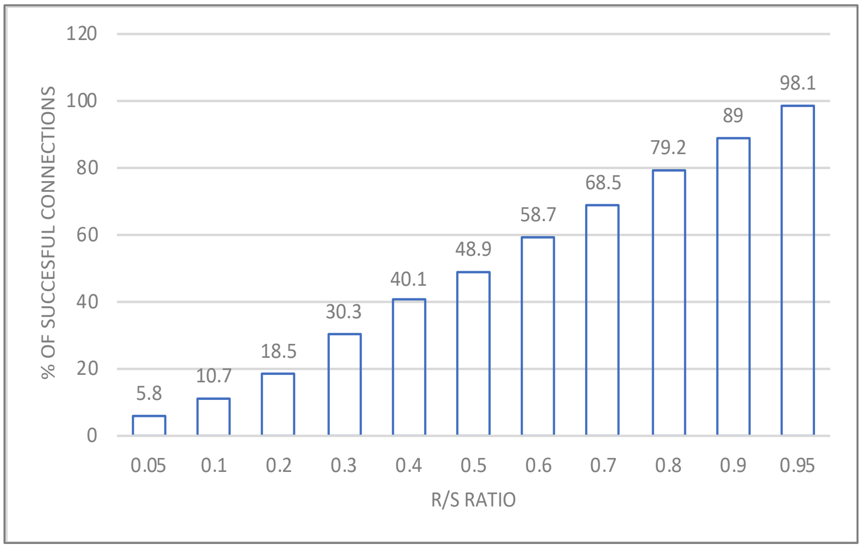



***Figure 5-8 Number of successful connections***

Figure 5-9 shows the time needed to generate a sub-chord along with the number of nodes in the sub-chord. As shown, the time taken becomes quazi-linear with the number of the nodes in the sub-chord starts to be large. This is shown in Fig. 5-9 when the number of nodes in a sub-chord exceeds 4000 nodes. This proves that as the number of nodes exceeds certain value, the time to build a sub-chord may be inappropriate and thus a split of the sub-chord is needed into smaller sub-chords to maintain the time of deploying a sub-chord.

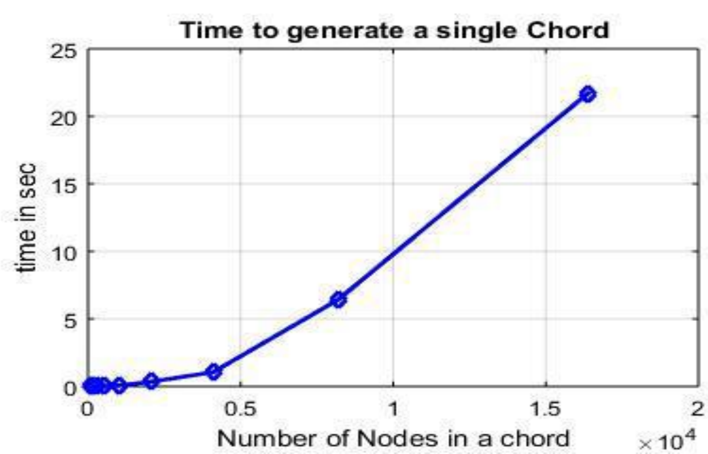



***Figure 5-9 Stablishing a sub-chord***

To show the fast response of the multi-chord network, Fig. 5-10 shows the mean time needed to add new keys in a sub-chord. The figure shows the mean time of adding 500 keys in sub-chords that have nodes ranging from 64 node up to 6000 nodes. The figure illustrates the applicability of the proposed model to accommodate keys with no significant change in time needed as the figure shows nearly flat response to do such operations.

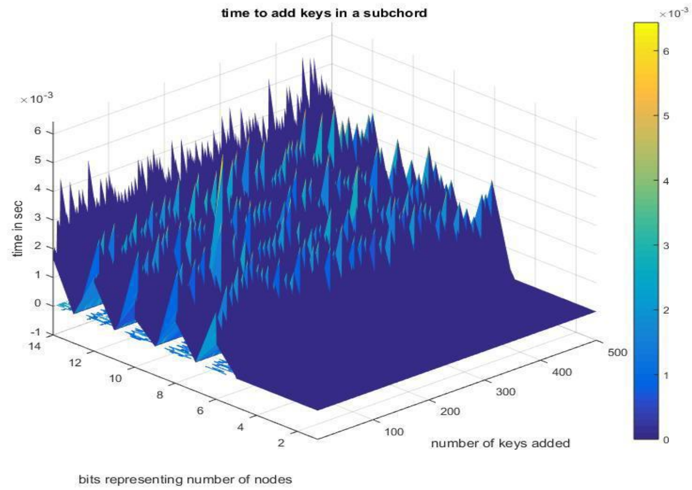



*Figure 5-10 Time to add a key in sub-chord*

## 5.3. Response Time to Adding Keys or Node Joining within a Single Sub-Chord

The time needed for nodes to join a sub-chord is illustrated in Fig. 5-11 where n is the number of bits used in the hashing algorithm to represent the number nodes in the multi-chord approach. As shown in the figure the required time to join a sub-chord of certain capacity is nearly constant. But as the capacity increases the time required to join the sub-chord becomes large which signifies again the importance to have a dynamic multi-chord network instead of a single chord network. This is further proven in Fig. 5-12 that shows the time required to leave a sub-chord in the system.

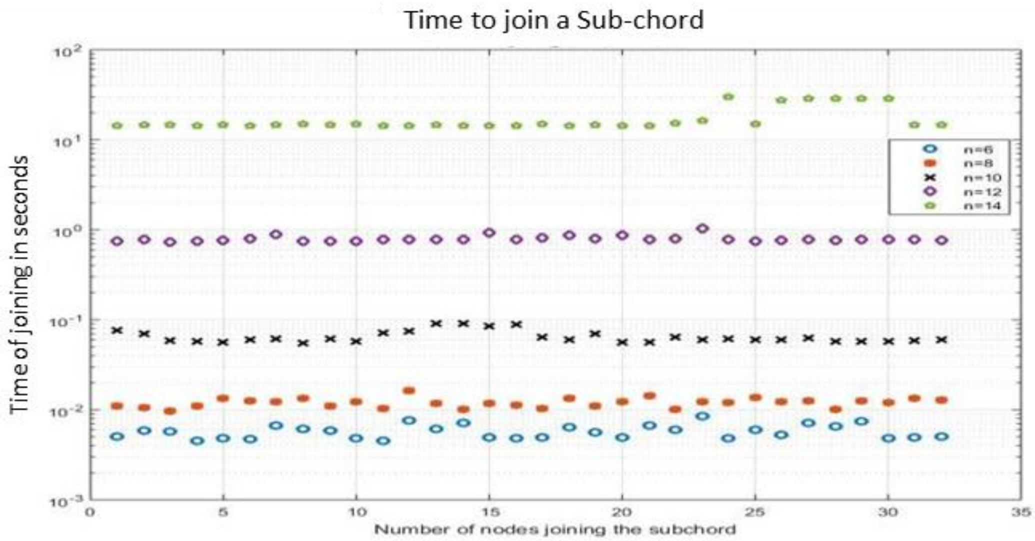



***Figure 5-11 Time to join a sub-chord***

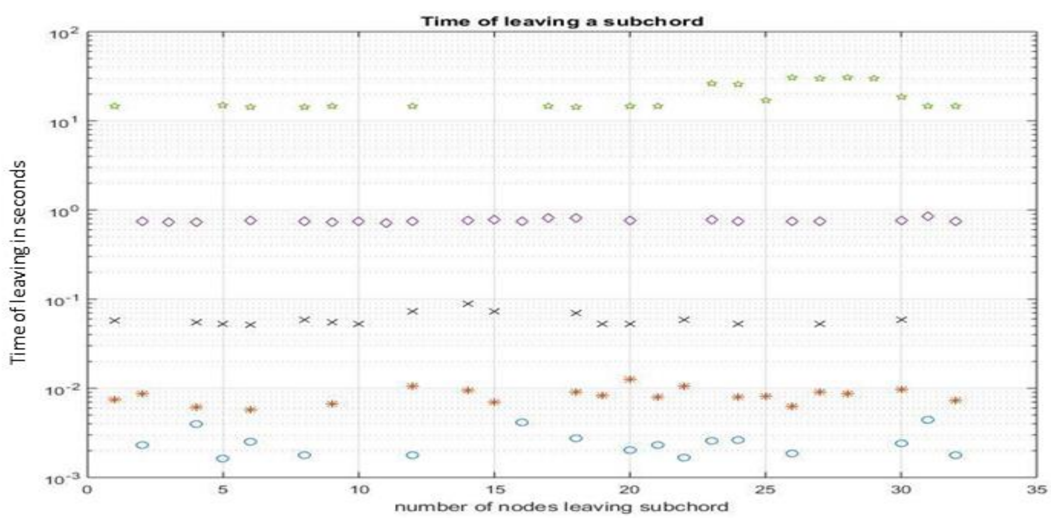



***Figure 5-12 Time to leave a sub-chord***

To evaluate the scalability of from the perspective of number of hops to finish an operation in multi-chord, we carried simulation experiments that evaluate the number of entries required in a finger table vs the number of sub-chords based on Markov chain. Operations considered in these simulations are node joining, node leaving, key lookup and storing because these operations are probabilistic, and depend only on current and previous states and Markov chain will be the appropriate for these operations as explained in chapter three.

A sub-chord with identifier m bits will have $2^m$ nodes, and suppose N nodes exist, at each lookup step for a key k, the next nodes depend on the position of node n holding the key.

Using Markov Chain (MC), we define the ID of the nodes in the form of positions taking values: 0, 1, 2, …. ,$2^m$-1. We also define the destination node which holds the key we are looking up for it as n.
Suppose the lookup algorithm has reached node x (node between the source node and destination node) which is away from node n by a difference in value of $n_{id}$ from $x_{id}$, which will be defined as Z = $|x_{id} - n_{id}|$.
Then, in the next lookup, we will find farthest node y in finger table of x that doesn't pass node n.
Assume d = z - $2^{\lfloor \log_2 z \rfloor}$ , where d is the difference between node $y_{id}$ and node $n_{id}$.
We have two possibilities that may happen:
The first possibility: node y might be closest to node n (n is in finger table of y). This happens when w = $2^{\lfloor \log_2 w \rfloor}$ , where w is the difference in value between of $\mathrm{n_{id}}$ from $\mathrm{y_{id}}$, where w = $|y_{id} - n_{id}|$ with a probability $(\frac{2^m-w}{2^m})^{N-2}$, where this probability is the probability of any node which is not in the region between node y and node n.
The second possibility: next lookup from y brings to a node v which is closer to n, v is located between y and n. This happens with a probability $(\frac{2^m-w+l}{2^m})^{N-2}$ - $(\frac{2^m-w+l-1}{2^m})^{N-2}$ , where L = $|v_{id} - y_{id}|$ and w= j – $2^{\lfloor \log_2 j \rfloor}$ ,

Therefore, the transition matrix can be formulated as

$$t_{ij} = \begin{cases} (\frac{2^m-w}{2^m})^{N-2} & ,i=0 \\ (\frac{2^m-w+l}{2^m})^{N-2} - \left(\frac{2^m-w+l-1}{2^m}\right)^{N-2} & ,0<i<w \\ 0 & ,\ i>w \end{cases}$$

*(5-1)*

We present the following results that represent the scalability, resilience, and responsiveness of multi-chord approach.
We simulated the multi-chord approach for different network sizes and different number of nodes per sub-chord. We present here an instance of the simulation that shows the Markov state diagram, steps to reach steady state of MC and the steady state probabilities for a sub-chord of five existing nodes out of eight nodes.
Figure 5-13 shows the Markov transition probabilities for the simulated sub-chord and it is in line with the expected transition matrix.

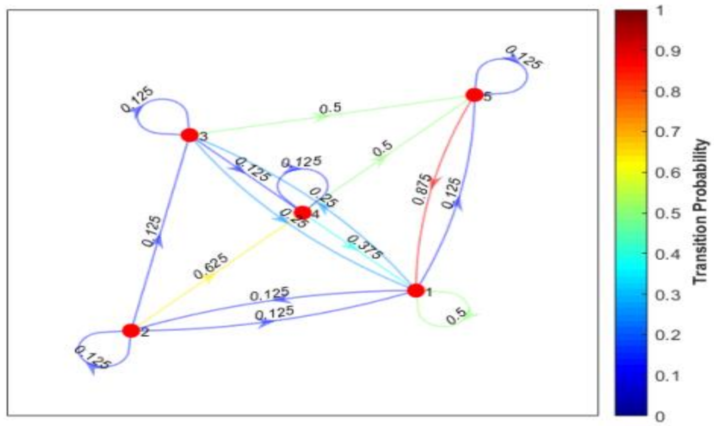



***Figure 5-13 Markov Transition Probabilities***

We then simulated the long-term behavior of the sub-chord and found that the sub-chord nodes reach a steady state after an average number of iterations nearly equal to the number of existing nodes in sub-chord as shown in Fig. 5-14 which means that the steady state responsiveness of the system is in O(n) and the steady state probabilities depend on the position of the existing nodes as shown in Fig. 5-15.

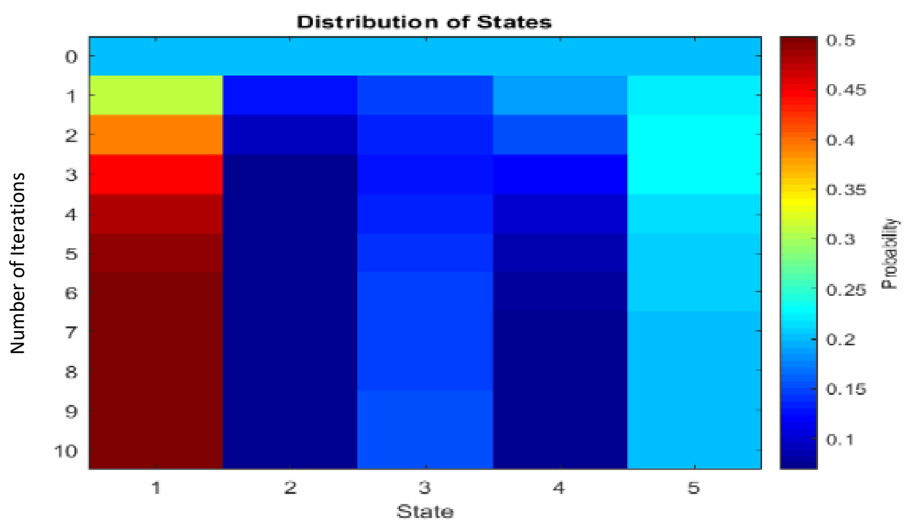



***Figure 5-14 Steady State analysis***

The results in Fig. 5-15 indicate that there are some nodes in the chord of importance than other nodes as their steady state probability is higher and means that they will play a higher priority role in the assurance of stability and fault resistance than other nodes in sub-chord and this will be presented in coming results.

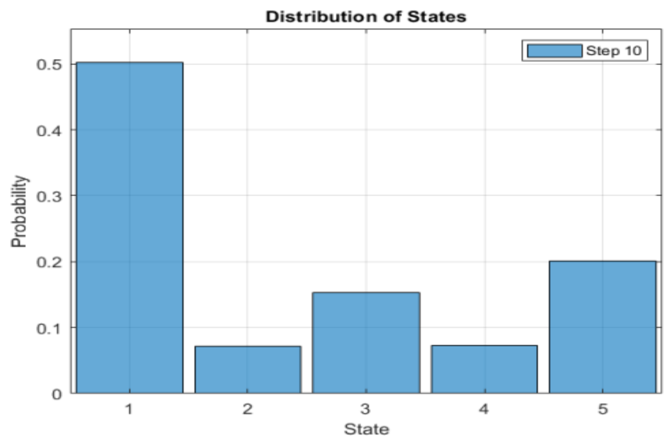



*Figure 5-15 Distribution of States*

## 5.4. Scalability of Multi-Chord Peer to Peer P2P Approach

**Outdegree of multi-chord :**The outdegree measurement is an indicator to scalability of the network as it shows the dependability of any node on other nodes to reach a destination. From the analysis of standard chord and the multi-chord, Fig. 5-16 shows the average number of connections going out from a node to other nodes in network. As Fig. 5-16 shows, the relation is in order of log(N/C), where N represents the number of total nodes in the network and C represents the number of sub-chords.

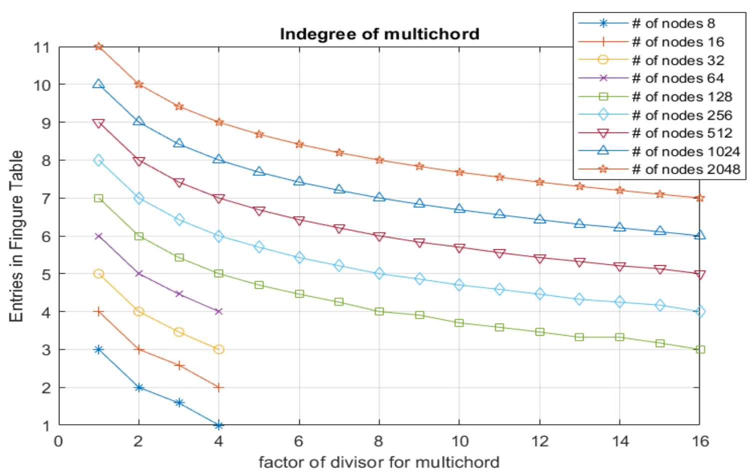



*Figure 5-16 Outdegree of multi-chord*

**Indegree of multi-chord :**The indegree measurement is an indicator to resilience the network to dynamic changes in nodes of the network. It shows the dependability other nodes will have on any other node to maintain the stability of the network. From the analysis of standard chord and the multi-chord, the indegree in Fig. 5-17 shows the average number of connections going into a node from other nodes in network. Also, Fig. 5-17 shows the relation is in order of $\log^2(N/C)$.

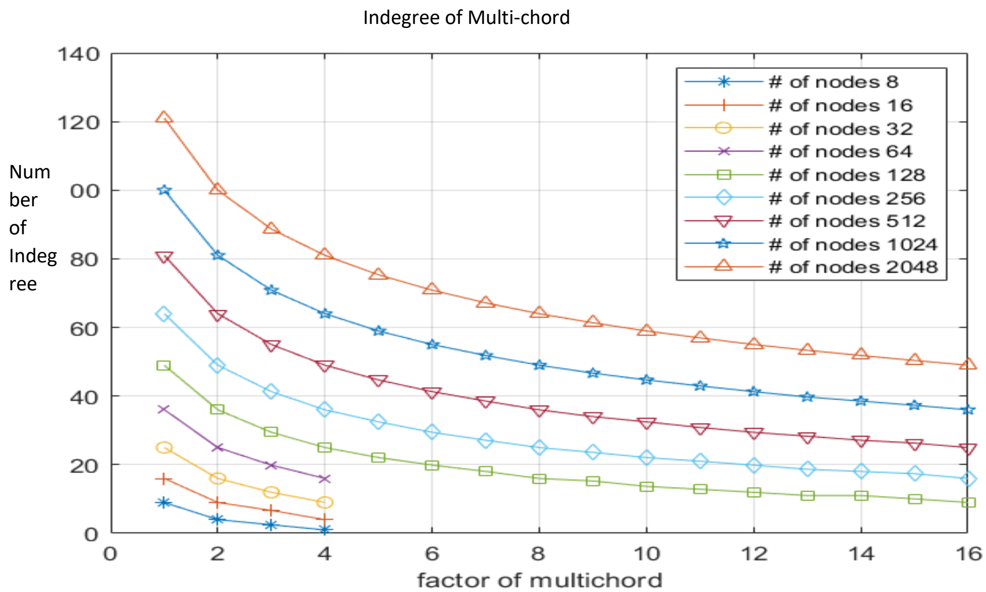



***Figure 5-17 Indegree of multi-chord***

**Diameter of reachability in multi-chord: The** diameter in multi-chord shown in Fig. 5-18 represents the maximum distance that can be covered from a certain node to reach a certain destination. If destination is within the sub-chord then, the diameter will be $O(\log(\frac{N}{C}))$ and if the destination is in another sub-chord, the diameter will be (C-1) + $O(\log(\frac{N}{C}))$ in multi-chord.

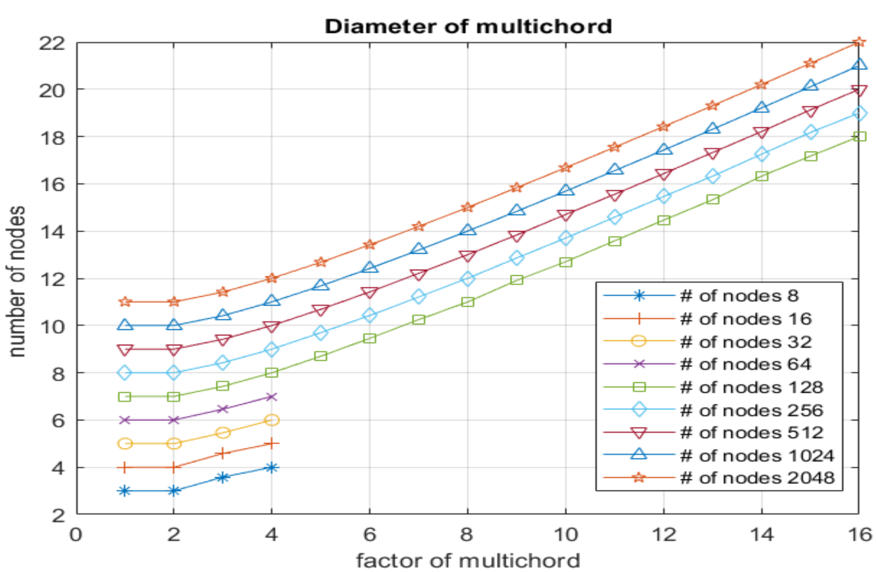



***Figure 5-18 Diameter of Multichord***

As mentioned before, in our multi-chord model, the choice of shared node in any sub-chord was assumed to be random at the beginning. But after thorough investigation on different sub-chord sizes and the number of nodes in each sub-chord, it was found that not all nodes are of equal probability in being a shred node to other sub-chords. It was found that some nodes will exhibit higher probability in being better nodes to act as shared nodes than others. Thus, choosing shred nodes in multi-chord model will be based on the size of each sub-chord, number of actual nodes active and their distribution.

## 5.5. Selection of Nodes as Shared Nodes in Multi-Chord Approach

For example, the following Fig. 5-19 shows the probability distribution function of nodes in a 16-node sub-chord. Each sub-figure shows the pdf of number of active nodes in the sub-chord. First node is the best candidate for being a shared node in a very few active nodes. As the number of active nodes increases to relatively half the size of the sub-chord, the nodes that are best candidates are middle ones. Then for nearly fully loaded sub-chord, the best candidate nodes are equally likely among all nodes.

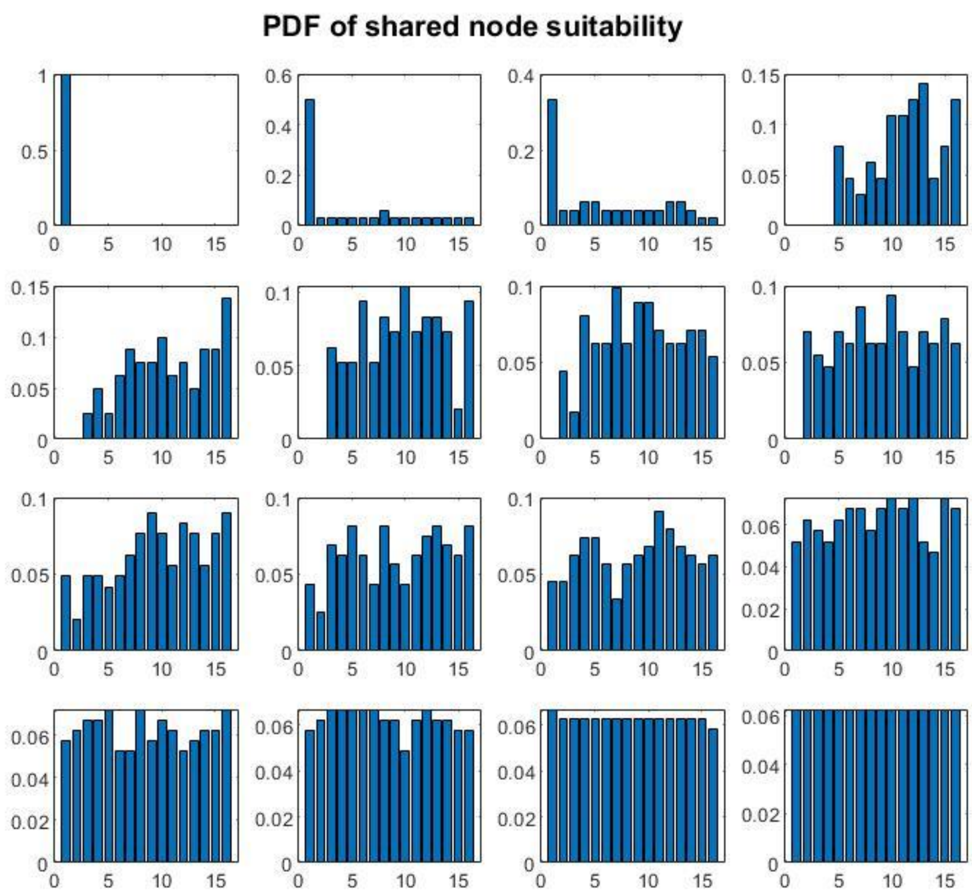



***Figure 5-19 PDF of shared node suitability at n=16***

This result is assured when we run for a 24-node sub-chord as shown in Fig. 5-20 and 64-nodes sub-chord shown in Fig. 5-21

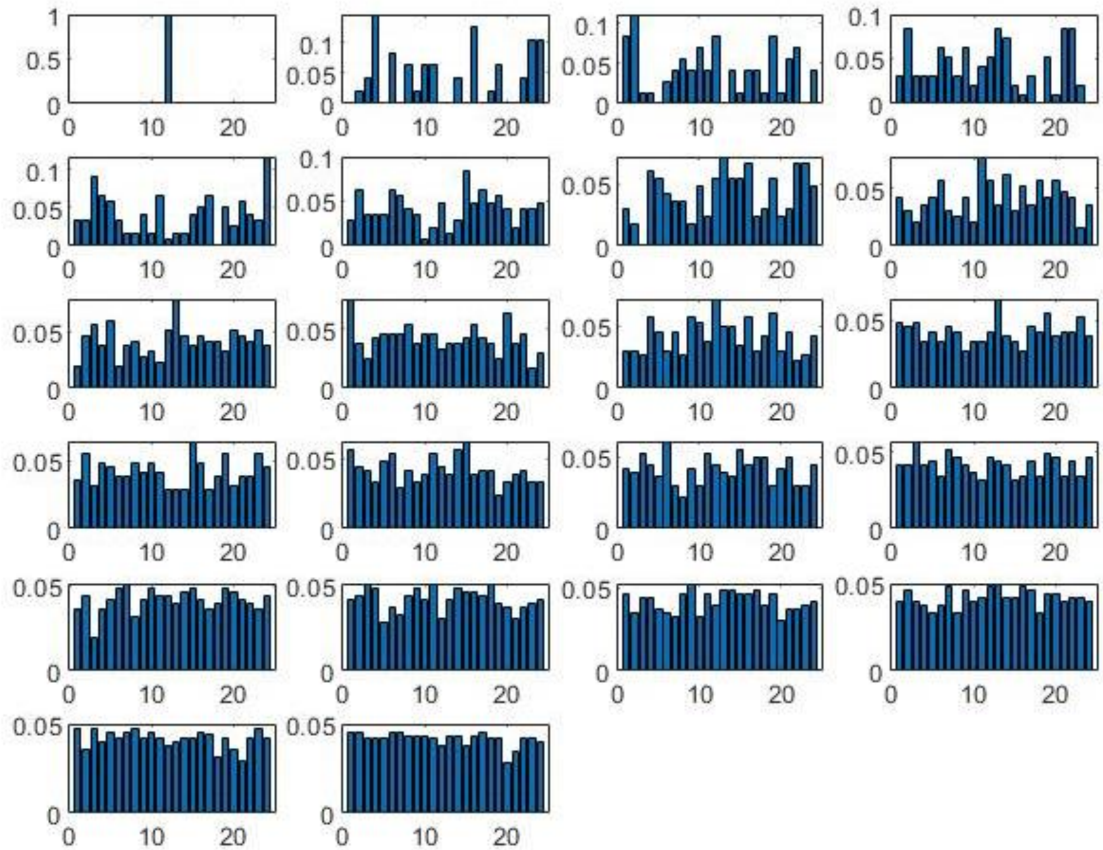


*Figure 5-20 PDF of shared node suitability at n=24*

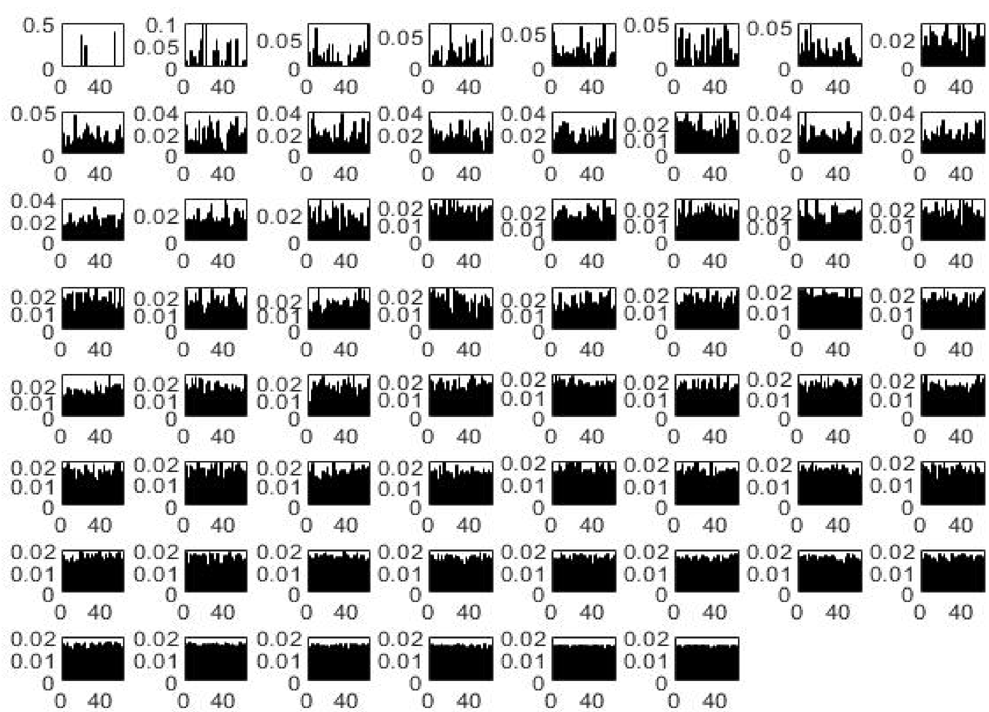


*Figure 5-21 PDF of shared node suitability at n=64*

However, as the number of nodes in sub-chord becomes large, the probability of choosing specific shared nodes tends to be distributed equally among active nodes and no certain nodes will be of higher probability than other as shown in the graphs of Fig. 5-22 of 800 nodes and of 1800 nodes successively.

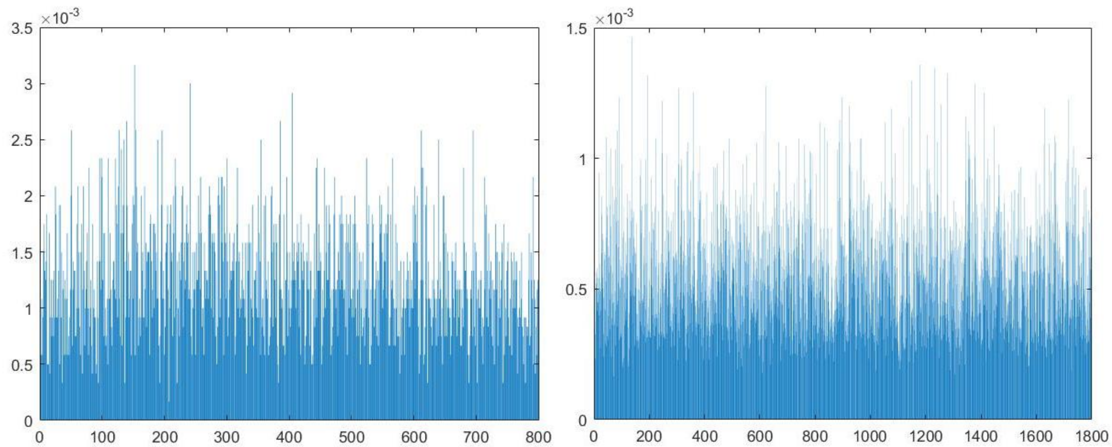



***Figure 5-22 Probability of choosing a node***

## 5.6. Response Time to Chord Operations within a Multi-Chord Approach

To analyze the delay of multi-chord in terms of time needed to complete a search function in multi-chord rings where:

- X: indicates how many times a query must be forwarded until it reaches the peer having the answer. X will be denoted as the peer distance
- H: number of overlay hops needed to complete a search, i.e., the number of forwards of the query plus one hop for the transmission of the answer
- n: size of the Chord-ring
- T: describes the total search duration
- TN: describes the delay of a query packet, which is transferred from one peer to a successor peer
- TA: represents the time needed to transmit the answer from the peer (having the answer) back to the originator

We differentiate between TN and TA, as the size (and hence the delay) of a search packet and an answer packet may be unequal [73]. The answer might, e.g., consist of multiple packets containing a detailed reply to the query

| X | H | P(X=i) | Search Time T |
|---|---|---|---|
| 0 | 0 | $p_0 = \frac{\binom{\log_2(n)}{0}}{n}$ | 0 |
| 1 | 2 | $p_1 = \frac{\binom{\log_2(n)}{1}}{n}$ | $T_A + T_N$ |
| 2 | 3 | $p_2 = \frac{\binom{\log_2(n)}{2}}{n}$ | $T_A + T_N + T_N$ |
| .<br>.<br>. | .<br>.<br>. | .<br>.<br>. | .<br>.<br>. |
| i | i+1 | $p_i = \frac{\binom{\log_2(n)}{i}}{n}$ | $T_A + \sum_{i=1}^{i} T_N$ |
| $\log_2(n)$ | $\log_2(n) + 1$ | $p_{\log_2(n)} = \frac{\binom{\log_2(n)}{\log_2(n)}}{n}$ | $T_A + \sum_{i=1}^{\log_2(n)} T_N$ |

***Table 5-1 Peer Distance Distribution and Search Time***

Table 5-1 shows the probability that searched peer is i hops away from a searching peer in any given sub-chord with symmetric search space and uniformly distributed keys. The table also shows the search time for every case in terms of $T_A$ and $T_N$ where $T_A$ is time needed to transmit the response from the peer having the searched key back to the node looking up for the key and $T_N$ is delay a query takes from one peer to a successor peer

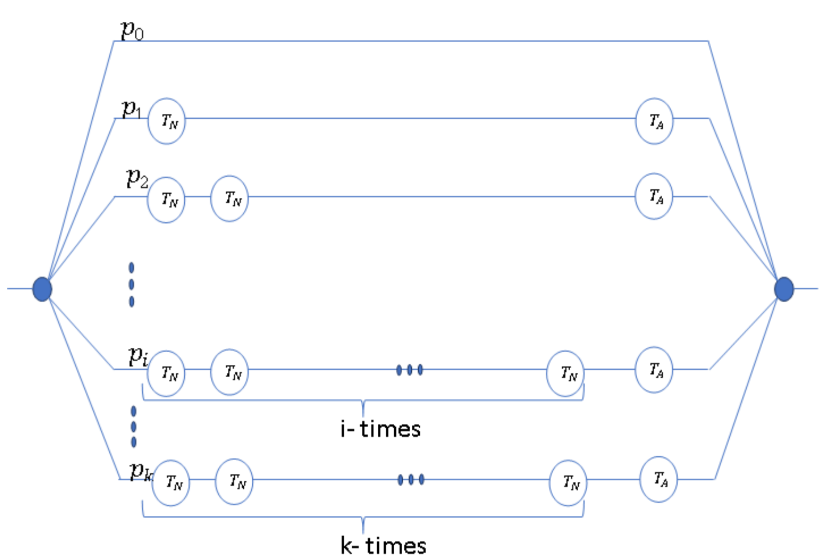



***Figure 5-23 Phase diagram of the search duration T***

The search delay diagram is shown in Fig. 5-23 that shows the search delay for any particular path *i* within a sub-chord with probability $p_i$ depicted in table 5-1

Figure 5-24 shows the normalized mean delay with respect to mean delay of lookup query of single hop and vs the total number of nodes in the system. This means that we referred the mean lookup delay of the multi-chord network to the mean delay required for a single key lookup. The figure compares also the normalized mean delay of single chord to a multi-chord approach.

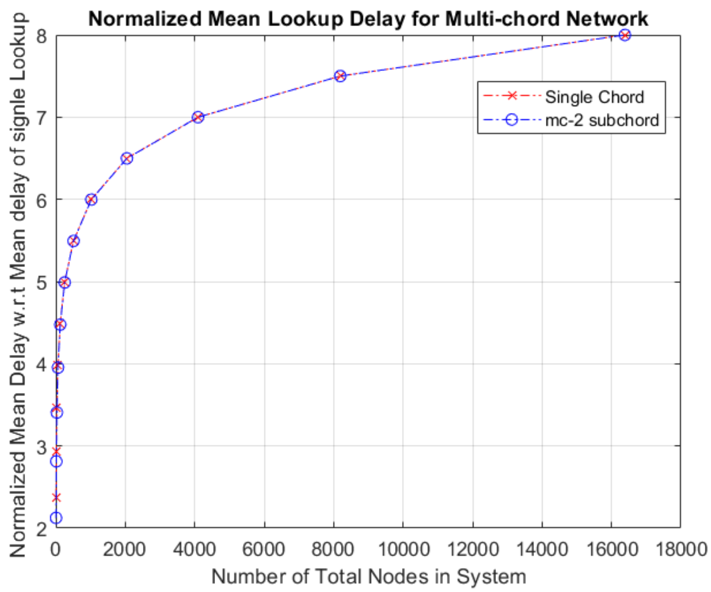



***Figure 5-24 Normalized Mean Lookup Delay of multi-chord network***

We assumed Fig. 5-24 the number of multi-chord to be two multi-chords with equal number of nodes in each sub-chord. As the figure shows the normalized mean delay is identical in both scenarios as the time saved in looking in a single chord is nearly equal to the time needed to lookup between sub-chords. In addition, the values tend to be proportional to number of nodes in the finger table and ring table.

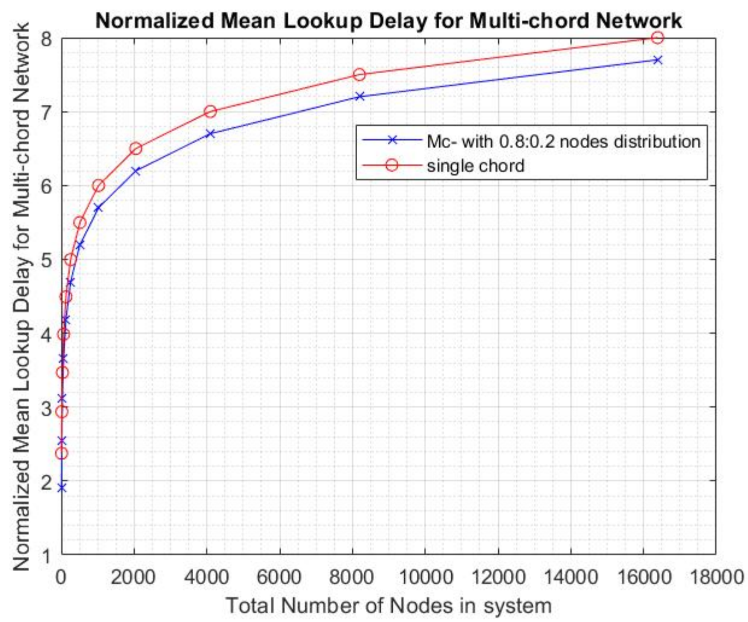



*Figure 5-25 Normalized Mean Lookup Delay for Multi-chord Network*

However, when the number of nodes in a sub-chord becomes unequal as in Fig. 5-25, the multi-chord approach became superior to the single chord. Figure 5-25 shows the comparison between single chord and a multi-chord with two sub-chords with distribution of 80% of the nodes in one chord and 20% in the other chord. The superiority came from the fact that the lookup time inside a sub-chord becomes significantly lower and the number of lookup between sub-chords became significantly lower as well.

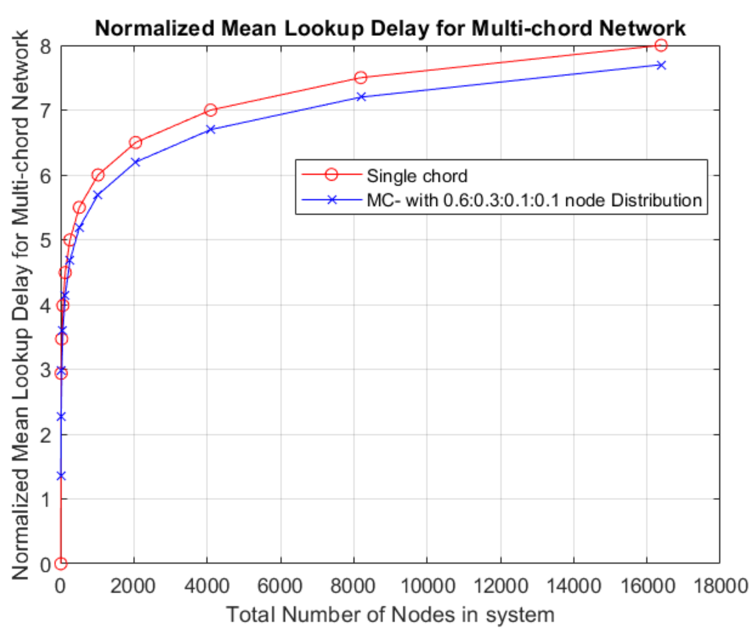



*Figure 5-26 Normalized Mean Lookup Delay for Multi-chord Network*

This is also proven from Fig. 5-26 when we generated four sub-chords with nodes distribution of ration 06:0.3:0.1:0.1. however, the gain from increasing number of sub-chords to reduce the mean lookup delay is not significant.

The coefficient of variation of search delay is shown in Fig. 5-27 vs the total number of nodes in the system for different transmission delay coefficients of variation. We also compared the results with a single chord approach.

The variation in delay is significantly lower in multi-chord approach than in single chord approach in case of having two sub-chords with equal number of nodes. This proves that also the mean delay is identical, the variation in delay in single chord is higher than in multi-chord.

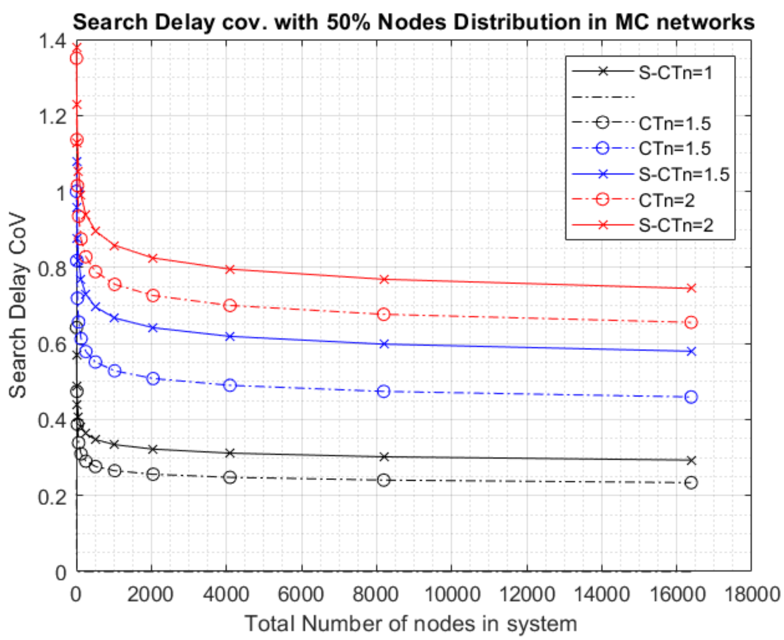



***Figure 5-27 Search delay***

We run the experiment for 80:20 ratio and for the 60:20:10:10 ratio as in Fig 5-28 and Fig. 5-29 and it was found that the difference in variations between single chord and multi-chords tends to get smaller and this was expected as the mean delay presented in figures showed that the mean delay became smaller in multi-chord compared to single chord.

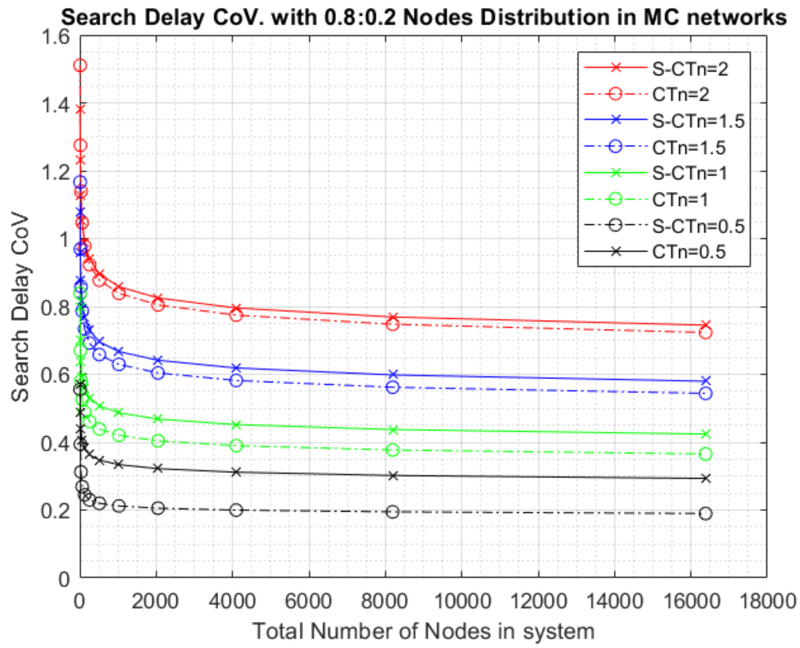



*Figure 5-28 Search Delay at 80:20 ratio*

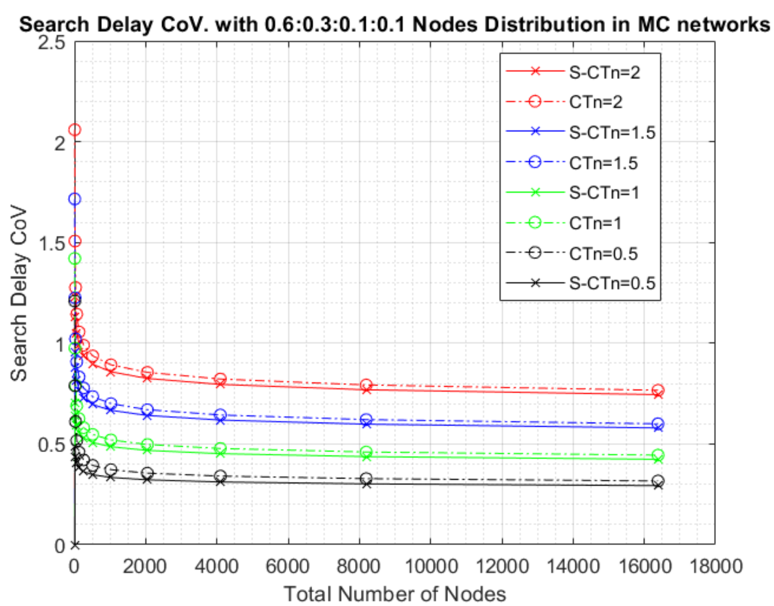



*Figure 5-29 Search Delay at 60:20:10:10 ratio*

Lastly, Fig. 5-30 proves the scalability of multi-chord as it shows the probability of lookup delay taking longer time decreases as function of expected mean delay. And as the number of nodes in the multi-chord increases the probability of lookup delay taking longer time does not worsen exponentially but nearly linear.

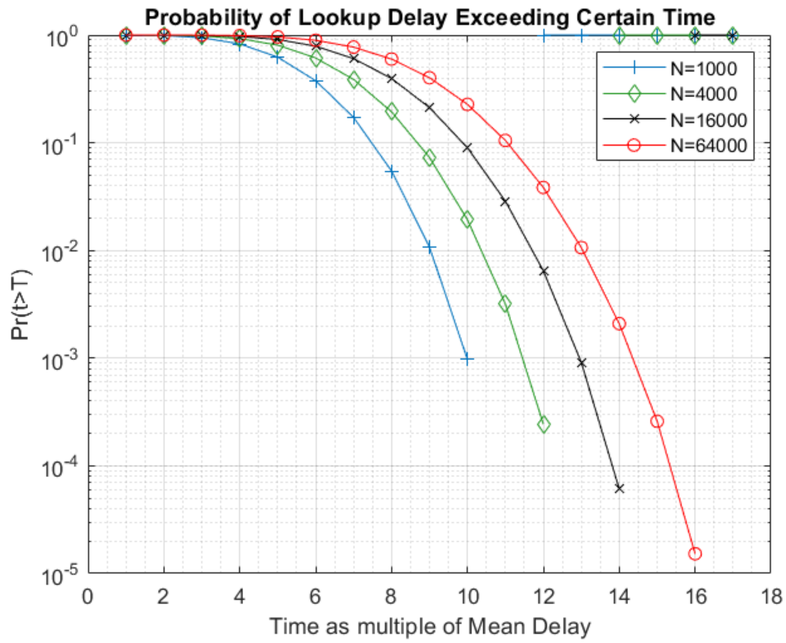



***Figure 5-30 Probability of lookup delay***

## 5.7. Results Summary

The performance of a single chord in the system is presented in section 5-1. Section 5.2 and section 5.3 shows the effectiveness of the multi-chord in case of failure which fulfills the criteria of the third research objective.

Also, section 5.3 presents the theoretical results of Markov chain model of multi-chord to cover the second research objective of the thesis.

For complexity and scalability, analysis was required for the third research objective, section 5.4 proves the applicability of multi-chord as number of nodes and sub-chords increase in the network.

The first research objective mentions the self-organization of maintenance requirement in multi-chord network, and this is proven in section 5.5.

And finally, the performance of the whole architecture is presented in section 5-6 that covers the criteria of the third research objective.

In the case of evaluating the performance of a sub-chord in the multi-chord network:

- Reduction in mean stabilization time of nearly 14% was achieved using the random lookup of next node in successor list compared to the usual method of sequential lookup of next node.
- An average increase of nearly 30% in mean time of stabilization will be experienced if the network faces failure in chord operation in multi-chord architecture.

- The complexity of meantime to lookup keys in multi-chord is reduced an average 20% in multi-chord compared to single chord architecture.

And in the case of evaluating the performance of multiple sub-chords in the multi-chord model:

- Mean time to Stabilization showed 20% enhancement
- Scalability showed an enhancement by 30%
- Mean of Lookup time showed 8% enhancement
- Variance of Search delay showed 2% enhancement

It is worth mentioning that the first objective was presented in chapter three which illustrates the architecture of the multi-chord P2P the modes of operations as well as the behavior of the proposed network.

# Chapter 6

## 6. Conclusion and Future Work

### 6.1. Conclusion

The dissertation describes cutting-edge research on satellites - terrestrial integrated networks, particularly with the LEO mega constellations at the space, to explore its potential and possibility. In detail, following contributions have been made to the related research area:

We introduced an optimized architecture of the multi-chord approach for IoT networks which is a novel approach that changes the concept terrestrial as well as satellites networks follow which is merely relay of data over the network and satisfy the first objective of this research.
However, upcoming satellites already have big storage capacity, which can be used to reduce the latency and complexity of operations to access and manipulate data. In addition, the integration of the satellites network with the terrestrial networks through the overlay approach of the multi-chord will assure the scalability, global coverage, reliability, and resilience that are highly required in IoT networks. For example, data that require certain real time operations may get processed and stored in the satellite nodes. Hence, to solve the problem of latency and complexity, the multi-chord concept would bring advantage to make logical networks that include some or all satellites in a constellation and this chapter presents the first and second contribution of this research.
We also presented the mathematical model developed for multi-chord that shows the theoretical performance of multi-chord in different operation required such as data manipulation, node operations and network operations. This contribution is considered to the best of our knowledge is not clear in chord P2P networks and for sure in out proposed multi-chord network. Thus, this contribution serves the second objective and gives the formal method for evaluation of multi-chord IoT in terms of performance metrics, e.g.:

- Logical sub-chords including satellites should have nice trade-off between energy consumed, access delay and failure recovery times.
- Hops/node for data lookup and data storage
- Hops/node for stabilization
- Percentage of failed lookups (robustness)
- Lookup latency

The performance of IoT multi-chord in terms of organizing nodes in a P2P from the flat approach of the traditional chord to a multi-chord topology with hierarchal structure was presented. Also, nodes in multi-chord IoT will be classified either based on their location or function. In addition, the accessibility mechanism is redesigned in multi-chord IoT to include beside finger table the chord table and multi-successor node tables also which will affect the multi-chord stabilization method. These operations will be evaluated through in-depth simulation to proof the applicability of the multi-chord networks and satisfy the third objective.

The performance of multi-chord IoT network is examined in a simulated environment.

The achieved results prove that our proposal has the operability and feasibility to apply to practice. According to the simulation results, the next generation LEO mega constellation can provide services that meet QoS requirements.

The contributions presented in this thesis hopefully represent a small step forward in this direction.

As a conclusion, multi-chord satellite IoT network can fulfill the basic requirements of many applications such as smart cities and health care monitoring in terms of scalability, resilience, and time responsiveness. However, multi-chord faces issues of real time processing for applications that require media data as storage of high volume is required as well as fast processing times.

Also, multi-chord is constrained by technology of satellite networks and their limitation in power and special devices to communicate through satellite.

The first research objective of achieving self-organization in multi-chord network was presented in section 5.5 and in chapter three we illustrated the architecture of the multi-chord P2P the modes of operations as well as the behavior of the proposed network.

The performance of and effectiveness of the multi-chord presented in chapter 5 fulfilled the requirements of the third research objective and we succeed in analyzing the operations of multi-chord mathematically which fulfilled the requirements of second objective

And finally, the performance of the whole architecture is presented in section 5-6 that covers the criteria of the third research objective.

We end up this dissertation by proposing some directions for future work.

## 6.2. Future Work

In the recent years, as of the great coverage and flexibility, the research topic of the satellites – terrestrial integrated networks have become dramatically popular thanks to the rapid development of the LEO satellite manufacturing and launching industry. More and more enterprises and institutes such as SpaceX, Google, Virgin Group and much more, have joined the commercial satellite market to provide low-cost customer-oriented satellite communication services. The cost of satellite communications has become arguably competitive to traditional terrestrial communication services.

Although this study has brought forward a novel logical architecture in managing things, however there are still challenges regarding this work that can be continued to improve in the near future:

1. In particular, the centralized authentication schemes are no longer suitable for the emerging LEO satellite assisted IoT ecosystem [74]. As Blockchain is a decentralized system, deploying Blockchain in satellite IoT multi-chord networks will enhance the security aspect by introducing an efficient and privacy-preserving blockchain-based authentication scheme.
2. Due to its correctness and simplicity, we may use the subnetting approach in IP network as a method for identifying sub-chords in multi-chord networks. The m bits used in defining the output of hash will be divided into x bits for sub-chord ID and y bits for node ID. Thus, after hashing the data, the most significant x bits in the resulted m bits will be the sub-chord ID and will be looked up from the chord table and the remaining y bits will apply the standard insert a key operation in the chosen sub-chord.
3. Physical layer parameters will be taken into consideration in our future work and study its impact on our results.
4. Investigate the resilience of the networks to satellite failure among other types of intermittent disconnectivity in inter-orbit communication and the robustness to cascaded enormous number of satellites failure in one and multiple orbits.
5. Combining the location information of nodes, a location-aware overlay mechanism for wireless mesh networks, called WILCO, including a new identification (ID) mapping scheme and an improved finger table. By knowing the location of a peer, good quality of service in stretched wireless environment can be achieved. Accurate location information can improve localization accuracy and reduce communication cost.
6. Introduce GEO satellites layer to the system model to get the benefits of Space Information Network (SIN) fog service.

# References


1. S. Boštjančič Rakas, V. Timčenko, M. Kabović and A. Kabović, "Industrial Internet: Architecture, characteristics and implementation challenges," 2021 20th International Symposium INFOTEH-JAHORINA (INFOTEH), 2021, pp. 1-4, doi: 10.1109/INFOTEH51037.2021.9400694.

2. ITU-T Recommendation Y.2060 Overview of the Internet of things, 2012.

3. O. Vermesan, P. Friess, P. Guillemin et al., "Internet of things strategic research roadmap," in Internet of Things: Global Technological and Societal Trends, vol. 1, pp. 9–52, 2011.

4. I. Pe˜na-L´opez, Itu Internet Report 2005: The Internet of Things, 2005.

5. U. Bodkhe et al., "Blockchain for Industry 4.0: A Comprehensive Review," in IEEE Access, vol. 8, pp. 79764-79800, 2020, doi: 10.1109/ACCESS.2020.2988579.

6. S. -F. Su, I. J. Rudas, J. M. Zurada, M. J. Er, J. -H. Chou and D. Kwon, "Industry 4.0: A Special Section in IEEE Access," in *IEEE Access*, vol. 5, pp. 12257-12261, 2017, doi: 10.1109/ACCESS.2017.2704758.
7. Y. Ma, Y. Chen and J. Chen, "SDN-enabled network virtualization for industry 4.0 based on IoTs and cloud computing," 2017 19th International Conference on Advanced Communication Technology (ICACT), 2017, pp. 199-202, doi: 10.23919/ICACT.2017.7890083.

8. Q. K. Ud Din Arshad, A. U. Kashif and I. M. Quershi, "A Review on the Evolution of Cellular Technologies," 2019 16th International Bhurban Conference on Applied Sciences and Technology (IBCAST), 2019, pp. 989-993, doi: 10.1109/IBCAST.2019.8667173.

9. A. Ghosh, A. Maeder, M. Baker and D. Chandramouli, "5G Evolution: A View on 5G Cellular Technology Beyond 3GPP Release 15," in IEEE Access, vol. 7, pp. 127639-127651, 2019, doi: 10.1109/ACCESS.2019.2939938.

10. "Keynote 2: IoT via 5G Satellite Systems," 2021 16th International Conference on Telecommunications (ConTEL), 2021, pp. 2-2, doi: 10.23919/ConTEL52528.2021.9495960.

11. Confais, B.; Lebre, A.; Parrein, B. A Fog storage software architecture for the Internet of Things. Adv. Edge Comput. Massive Parallel Process. Appl. 2020, 35, 61–105.

12. Schwandt, F. Internet of Things (IoT) connected devices installed base worldwide from 2015 to 2025 (in billions). Statista 2018. Available online: https://www.statista.com/statistics/471264/iot-number-of-connected-devices-worldwide/ (accessed on 25 September 2021).

13. Li, Y.; Orgerie, A.C.; Rodero, I.; Amersho, B.L.; Parashar, M.; Menaud, J.M. End-to-end energy models for Edge Cloud-based IoT platforms: Application to data stream analysis in IoT. Future Gener. Comput. Syst. 2018, 87, 667–678.

14. Jurcut, Anca D., Pasika Ranaweera, and Lina Xu. "Introduction to IoT security." IoT security: Advances in authentication (2020): 27-64.

15. Guth, Jasmin, et al. "Comparison of IoT platform architectures: A field study based on a reference architecture." 2016 Cloudification of the Internet of Things (CIoT). IEEE, 2016.

16. Guth, Jasmin, et al. "A detailed analysis of IoT platform architectures: concepts, similarities, and differences." Internet of everything. Springer, Singapore, 2018. 81-101.

17. Fraire, Juan & Céspedes, Sandra & Accettura, Nicola. (2019). Direct-To-Satellite IoT - A Survey of the State of the Art and Future Research Perspectives: Backhauling the IoT Through LEO Satellites. 10.1007/978-3-030-31831-4_17.

18. J. Guo and Y. Du, "Fog Service in Space Information Network: Architecture, Use Case, Security and Challenges," in IEEE Access, vol. 8, pp. 11104-11115, 2020, doi: 10.1109/ACCESS.2020.2964804.

19. O. Kodheli, N. Maturo, S. Chatzinotas, S. Andrenacci and F. Zimmer, "NB-IoT via LEO Satellites: An Efficient Resource Allocation Strategy for Uplink Data Transmission," in IEEE Internet of Things Journal, vol. 9, no. 7, pp. 5094-5107, 1 April1, 2022, doi: 10.1109/JIOT.2021.3109456.

20. J. Rui and S. Danpeng, "Architecture Design of the Internet of Things Based on Cloud Computing," *2015 Seventh International Conference on Measuring Technology and Mechatronics Automation*, 2015, pp. 206-209, doi: 10.1109/ICMTMA.2015.57.

21. Wei, J.; Han, J.; Cao, S. Satellite IoT Edge Intelligent Computing: A Research on Architecture. Electronics 2019, 8, 1247. https://doi.org/10.3390/electronics8111247

22. Zhang, C. Design and application of fog computing and Internet of Things service platform for smart city. Future Gener. Comput. Syst. 2020, 112, 630–640.

23. Association, I.S. IEEE Standard for Adoption of OpenFog Reference Architecture for Fog Computing. IEEE Std 1934-2018 2018, 1–176.

24. Stoica, Ion & Morris, Robert & Karger, David & Kaashoek, M. & Balakrishnan, Hari. (2001). Chord: A Scalable Peer-to-Peer Lookup Service for Internet Applications. ACM SIGCOMM Computer Communication Review, vol. 31. 31. 10.1145/964723.383071.

25. (Bajram) Spaseski, Narela & Ler, Daria. (2013). Application of Discrete-Time Markov Models. SOUTHEAST EUROPE JOURNAL OF SOFT COMPUTING. 2. 37-41. 10.21533/scjournal.v2i1.431

26. Sethi, Pallavi & Sarangi, Smruti. (2017). Internet of Things: Architectures, Protocols, and Applications. Journal of Electrical and Computer Engineering. 2017. 1-25. 10.1155/2017/9324035

27. Salman, Tara & Jain, Raj. (2017). Networking protocols and standards for internet of things. 10.1002/9781119173601.ch13

28. Nakhuva, Bhumi & Champaneria, Tushar. (2015). Study of Various Internet of Things Platforms. International Journal of Computer Science & Engineering Survey. 6. 61-74. 10.5121/ijcses.2015.6605.

29. S. Lucero, “Satellite IoT market repot - 2020,” Omdia, Whitepaper, Mar. 2020

30. H. A. -D. F. Kokez, "On Terrestrial and Satellite communications for telecommunication future," 2020 2nd Annual International Conference on Information and Sciences (AiCIS), 2020, pp. 58-67, doi: 10.1109/AiCIS51645.2020.00019.

31. M. Centenaro, C. E. Costa, F. Granelli, C. Sacchi and L. Vangelista, "A Survey on Technologies, Standards and Open Challenges in Satellite IoT," in IEEE Communications Surveys & Tutorials, vol. 23, no. 3, pp. 1693-1720, thirdquarter 2021, doi: 10.1109/COMST.2021.3078433.

32. Fraire J.A., Céspedes S., Accettura N. (2019) Direct-To-Satellite IoT - A Survey of the State of the Art and Future Research Perspectives. In: Palattella M., Scanzio S., Coleri Ergen S. (eds) Ad-Hoc, Mobile, and Wireless Networks. ADHOC-NOW 2019. Lecture Notes in Computer Science, vol 11803. Springer, Cham. https://doi-org.are.uab.cat/10.1007/978-3-030-31831-4_17

33. C. -Q. Dai, M. Zhang, C. Li, J. Zhao and Q. Chen, "QoE-Aware Intelligent Satellite Constellation Design in Satellite Internet of Things," in IEEE Internet of Things Journal, vol. 8, no. 6, pp. 4855-4867, 15 March15, 2021, doi: 10.1109/JIOT.2020.3030263.

34. M. Ghorbani, M. R. Meybodi and A. Mohammad Saghiri, "An Architecture for Managing Internet of Things based on Cognitive Peer-to-peer Networks," 2019 5th International Conference on Web Research (ICWR), 2019, pp. 111-116, doi: 10.1109/ICWR.2019.8765283.

35. M. Conoscenti, A. Vetrò and J. C. De Martin, "Peer to Peer for Privacy and Decentralization in the Internet of Things," 2017 IEEE/ACM 39th International Conference on Software Engineering Companion (ICSE-C), 2017, pp. 288-290, doi: 10.1109/ICSE-C.2017.60.

36. Z. Li, A.V. Barenji, G.Q. Huang, Toward a blockchain cloud manufacturing system as a peer-to-peer distributed network platform, Robot. Comput. Integr. Manuf. 54 (2018) 133–144, https://doi.org/10.1016/j.rcim.2018.05.011.

37. A. H. Mayer, V. F. Rodrigues, C. A. d. Costa, R. d. R. Righi, A. Roehrs and R. S. Antunes, "FogChain: A Fog Computing Architecture Integrating Blockchain and Internet of Things for Personal Health Records," in IEEE Access, vol. 9, pp. 122723-122737, 2021, doi: 10.1109/ACCESS.2021.3109822.

38. Z. Li, J. Yao and H. Huang, "A CoAP-based Decentralized Resource Discovery for IoT Network," 2021 6th International Conference on Communication, Image and Signal Processing (CCISP), 2021, pp. 398-402, doi: 10.1109/CCISP52774.2021.9639297.

39. D. Tracey and C. Sreenan, "Using a DHT in a Peer to Peer Architecture for the Internet of Things," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 560-565, doi: 10.1109/WF-IoT.2019.8767261.

40. P. Maymounkov and D. MaziШeres, “Kademlia: A Peer-to-peer Information System Based on the XOR Metric,” in First International Workshop on Peer-to-Peer Systems (IPTPS), 2002.

41. B. Cohen, “The BitTorrent Protocol Specification,” 2008. [Online]. Available: https://www.bittorrent.org/beps/bep_0003.html. [Accessed September 2021].

42. D. Tracey and C. Sreenan, "How to see through the Fog? Using Peer to Peer (P2P) for the Internet of Things," 2019 IEEE 5th World Forum on Internet of Things (WF-IoT), 2019, pp. 47-52, doi: 10.1109/WF-IoT.2019.8767275.

43. Antonopoulos, N., Salter, J., & Peel, R. (2005). A Multi-Ring Method for Efficient Multi-Dimensional Data Lookup in P2P Networks. FCS.

44. S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, “A scalable content addressable network,” in Processings of the ACM SIGCOMM, 2001, pp. 161–172.

45. A. Rowstron and P. Druschel, “Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems,” in Proceedings of the Middleware, 2001.

46. Karger, D., Lehman, E., Leighton, T., Panigrahy, R., Levine, M., Lewin, D.. Consistent hashing and random trees: Distributed caching protocols for relieving hot spots on the world wide web. In: Proceedings of the Twenty-ninth Annual ACM Symposium on Theory of Computing. STOC ’97; New York, NY, USA: ACM. ISBN 0-89791-888-6; 1997, p. 654–663. URL:http://doi.acm.org/10.1145/258533.258660. doi:10.1145/258533.258660

47. “Secure hash standard,” NIST, U.S. Dept. of Commerce, National Technical Information Service FIPS 180-1, April 1995.

48. Eng Keong Lua, J. Crowcroft, M. Pias, R. Sharma and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," in *IEEE Communications Surveys & Tutorials*, vol. 7, no. 2, pp. 72-93, Second Quarter 2005, doi: 10.1109/COMST.2005.1610546.

49. F. Dabek, M. F. Kaashoek, D. Karger, R. Morris, and I. Stoica, “Widearea cooperative storage with cfs,” in Proceedings of the eighteenth ACM symposium on Operating systems principles, 2001, pp. 202–215.

50. R. Cox, A. Muthitacharoen, and R. Morris, “Serving dns using chord,” in Proceedings of the First International Workshop on Peer-to-Peer Systems, March 2002.

51. Ghaleb, M.; Azzedin, F. Towards Scalable and Efficient Architecture for Modeling Trust in IoT Environments. Sensors 2021, 21, 2986. https://doi.org/10.3390/s21092986

52. Nguyen, Minh & Hoang, Nhat & Hluchý, Ladislav & Vu, Tuyet & Le, Hieu. (2017). Multiple Peer Chord Rings Approach for Device Discovery in IoT Environment. Procedia Computer Science. 110. 125-134. 10.1016/j.procs.2017.06.133.

53. A. I. Abdel Ghafar, Á. V. Castro and M. E. Khedr, " Multidimensional Self-Organizing Chord-Based Networking for Internet of Things" 2nd Europe – Middle East – North African Regional ITS Conference, Aswan 2019, http://hdl.handle.net/10419/201736.

54. A. I. Abdel Ghafar, Á. V. Castro and M. E. Khedr, "Satellite IoT services Using Multichord Peer to Peer Networking," 2019 IEEE 2nd 5G World Forum (5GWF), 2019, pp. 566-571, doi: 10.1109/5GWF.2019.8911672.

55. A. I. Abdel Ghafar, A. Vazquez-Castro and M. E. Khedr, "Resilience Analysis of Multichord Peer to Peer IoT Satellite Networks," 2021 23rd International Conference on Advanced Communication Technology (ICACT), 2021, pp. 220-225, doi: 10.23919/ICACT51234.2021.9370775.

56. B. Zhao, J. Liu, Z. Wei and I. You, "A Deep Reinforcement Learning Based Approach for Energy-Efficient Channel Allocation in Satellite Internet of Things," in IEEE Access, vol. 8, pp. 62197-62206, 2020, doi: 10.1109/ACCESS.2020.2983437.

57. O. B. Osoro and E. J. Oughton, "A Techno-Economic Framework for Satellite Networks Applied to Low Earth Orbit Constellations: Assessing Starlink, OneWeb and Kuiper," in IEEE Access, vol. 9, pp. 141611-141625, 2021, doi: 10.1109/ACCESS.2021.3119634.

58. J. Hindin, TECHNICAL APPENDIX: Application of Kuiper Systems LLC for Authority to Launch and Operate a Non-Geostationary Satellite Orbit System in Ka-Band Frequencies, Federal Communications Commission, Washington DC, USA, 2019.

59. SpaceX Non-Geostationary Satellite System Attachment a Technical Information to Supplement Schedule S, Federal Communications Commission, Washington DC, USA, 2021. [Online]. Available: https://fcc.report/IBFS/SAT-MOD-20181108-00083/1569860.pdf

60. Starlink satellites. n2yo.com. (n.d.). Retrieved July 1, 2022, from https://www.n2yo.com/satellites/?c=52

61. Ericsson, “Traffic Model for legacy GPRS MTC” (February 15-19, 2016), document GP 160060, 3GPP GERAN meeting #69.

62. S. Kalyanasundaram, E. K. P. Chong and N. B. Shroff, "An efficient scheme to reduce handoff dropping in LEO satellite systems," Proceedings Seventeenth IEEE Symposium on Reliable Distributed Systems (Cat. No.98CB36281), 1998, pp. 431-436, doi: 10.1109/RELDIS.1998.740536.

63. Ching, Wai-Ki, and Michael K. Ng. "Markov chains." Models, algorithms and applications (2006).

64. R. M. Feldman and C. Valdez-Flores, Applied Probability and Stochastic Processes. Heidelberg, Germany: Springer, 2009.

65. N. Li, I. Kolmanovsky, A. Girard and D. Filev, "Fuzzy Encoded Markov Chains: Overview, Observer Theory, and Applications," in IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 51, no. 1, pp. 116-130, Jan. 2021, doi: 10.1109/TSMC.2020.3042960.

66. Ross, Sheldon M. "Markov chains." Introduction to Probability Models, (1996): 185-280.

67. Privault, Nicolas. "Understanding markov chains." Examples and Applications, Publisher Springer-Verlag Singapore 357 (2013): 358.

68. H. Li and K. Li, "A New Process Mining Approach Based on the Markov Transition Matrix," 2014 International Conference on Computational Science and Computational Intelligence, 2014, pp. 81-85, doi: 10.1109/CSCI.2014.99.

69. Q. Cai, J. Niu and Y. Gao, "Capturing the Evolving Properties of Disconnected Mobile P2P Networks," 2011 Fifth FTRA International Conference on Multimedia and Ubiquitous Engineering, 2011, pp. 22-27, doi: 10.1109/MUE.2011.15.

70. H. Wang, Y. Li, X. Zhao and F. Yang, "An algorithm based on markov chain to improve edge cache hit ratio for blockchain-enabled IoT," in China Communications, vol. 17, no. 9, pp. 66-76, Sept. 2020, doi: 10.23919/JCC.2020.09.006.

71. A. Sepasi Zahmati, X. Fernando and A. Grami, "Steady-state Markov chain analysis for heterogeneous cognitive radio networks," 2010 IEEE Sarnoff Symposium, 2010, pp. 1-5, doi: 10.1109/SARNOF.2010.5469751.

72. Azimi, Saeed, et al. "Probabilistic analysis of long-term climate drought using steady-state markov chain approach." Water Resources Management 34.15 (2020): 4703-4724.

73. Binzenhöfer, Andreas, and Phuoc Tran-Gia. "Delay Analysis of a Chord-based Peer-to-Peer File-Sharing System." ATNAC 2004 (2004).

74. B. Wang, Z. Chang, S. Li and T. Hämäläinen, "An Efficient and Privacy-Preserving Blockchain-Based Authentication Scheme for Low Earth Orbit Satellite Assisted Internet of Things," in IEEE Transactions on Aerospace and Electronic Systems, 2022, doi: 10.1109/TAES.2022.3187389.