A COALITION MODEL FOR EFFICIENT INDEXING IN WIRELESS SENSOR NETWORK WITH RANDOM MOBILITY

HAZEM JIHAD ALI BADARNEH

FACULTY OF COMPUTER SCIENCE AND
INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR

2021

Universiti Malaya

A COALITION MODEL FOR EFFICIENT INDEXING IN WIRELESS SENSOR NETWORK WITH RANDOM MOBILITY

HAZEM JIHAD ALI BADARNEH

THESIS SUBMITTED IN FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF DOCTOR OF PHILOSOPHY

**FACULTY OF COMPUTER SCIENCE AND INFORMATION TECHNOLOGY
UNIVERSITY OF MALAYA
KUALA LUMPUR**

2021

Universiti Malaya

UNIVERSITY OF MALAYA

ORIGINAL LITERARY WORK DECLARATION

Name of Candidate: Hazem Jihad Ali Badarneh

Matric No: WHA150044

Name of Degree: Doctor of Philosophy

Title of Project Paper/Research Report/Dissertation/Thesis ("this Work"): **A coalition model for efficient indexing in wireless sensor network with random mobility**

Field of Study:

I do solemnly and sincerely declare that:

(1) I am the sole author/writer of this Work;
(2) This Work is original;
(3) Any use of any work in which copyright exists was done by way of fair dealing and for permitted purposes and any excerpt or extract from, or reference to or reproduction of any copyright work has been disclosed expressly and sufficiently and the title of the Work and its authorship have been acknowledged in this Work;
(4) I do not have any actual knowledge nor do I ought reasonably to know that the making of this work constitutes an infringement of any copyright work;
(5) I hereby assign all and every rights in the copyright to this Work to the University of Malaya ("UM"), who henceforth shall be owner of the copyright in this Work and that any reproduction or use in any form or by any means whatsoever is prohibited without the written consent of UM having been first had and obtained;
(6) I am fully aware that if in the course of making this Work I have infringed any copyright whether intentionally or otherwise, I may be subject to legal action or any other action as may be determined by UM.

Candidate's Signature　　　　　　　　Date: 15/Oct/2021

Subscribed and solemnly declared before,

Witness's Signature　　　　　　　　Date: 15 Oct 2021

Name:

Designation:

ii

University of Malaya

A COALITION MODEL FOR EFFICIENT INDEXING IN WIRELESS SENSOR NETWORK WITH RANDOM MOBILITY

ABSTRACT

Wireless Sensor Networks grow rapidly due to the sensing and scanning capabilities, in addition to the capabilities of Wireless Sensor Networks nodes to move randomly and send periodic reports. In such environments, the periodic reports of sensors' nodes contain the transmission attributes that represent the transmission locations and time. Because of these attributes, it requires to be efficiently allocated and effectively processed. In the case of random mobile sensors, a huge number of packets are generated, which exacerbated the burden of indexing the mobile sensors' reports, because of dependability on packets' attributes. However, several studies have attempted to improve the Grid-based index-trees by dividing the network area into smaller areas called grids. The fundamental principle underlying the division is to construct the index structure by determining the mobile sensors that belong to each grid, then construct the index according to the attributes of the packet. Nevertheless, it is more practical to improve the indexing procedures of random mobile sensors, because of the following challenges. First, the high dependability on multi-attributes (location and time) of packets in random mobile sensors. Second, in a random mobile environment, it is required for registering the current grid location of each sensor and assign its destination periodically. Specifically, this thesis proposes an efficient model called Dynamic Static Coalition-based (DySta-Coalition for short) to improve the indexing procedures of random mobile sensors. The proposed model consists of Dynamic-Coalition framework, Static-Coalition algorithm, and Coalition-Based Index-Tree framework. Dynamic-Coalition framework establishes a relevance between sensors and gateways to construct dynamic blocks called dynamic-coalitions. The proposed framework aims to improve the indexing packets and mitigate the disseminated data. Static-Coalition algorithm rearranges the packets in each gateway, by

source (location attribute). Accordingly, it arranges the packets, step-by-step before accumulated in the main server for indexing. Coalition-Based Index-Tree framework is an indexing process for random mobile sensors. Finally, the performance of DySta-Coalition model is comprehensively evaluated on various synthetic datasets with various parameters (number of mobile sensors, clusters, and dimensions) and metrics (packets delivery ratio, routing overhead, space cost overhead, index building time, and number of traversed nodes). First, evaluate Dynamic-Coalition framework and Static-Coalition algorithm, in terms of Average Routing Overhead and Packet Delivery Ratio. As well as, evaluate Coalition-Based Index-Tree framework in terms of the number of traversed nodes, space-cost overhead, and index building-time. The second evaluation divides into three scenarios. The first one evaluates Coalition-Based Index-Tree framework independently, without any effect from Dynamic-Coalition framework and Static-Coalition algorithm. The second and third scenarios evaluate the performance of Coalition-Based Index-Tree by examining the effects of Dynamic-Coalition framework and Static-Coalition algorithm on Coalition-Based Index-Tree framework. The evaluation results demonstrate that DySta-Coalition model can mitigate the disseminated packets, which results in minimizing dependability on the attributes of the packet. Furthermore, it is supreme according to the results of index building time and space cost overhead.

Keywords: Coalition, multi-attribute, spatial-temporal index, random mobile data, wireless sensor networks (WSNs), Grid-based indexing.

Universiti Malaya

MODEL PENGUMPULAN UNTUK INDEKS BERKESAN DALAM RANGKAIAN SENSOR WIRELESS DENGAN MOBILITI RANDOM

ABSTRAK

Rangkaian Sensor Tanpa Wayar (WSN) telah menarik minat penyelidik, kerana kemampuan mereka bergerak secara rawak dan menghantar laporan berkala. Nod WSN mampu mengubah lokasi mereka selama ini. Dalam lingkungan seperti itu, sensor diminta untuk melaporkan paket, yang sering mengandungi atribut transmisi. Atribut penghantaran mewakili lokasi penghantaran dan masa paket. Oleh kerana sifat-sifat ini, ia perlu diagihkan dan diproses dengan berkesan. Dalam kes sensor mudah alih rawak, sebilangan besar paket dihasilkan, yang menyebabkan beban pengindeksan laporan sensor bergerak, kerana ketergantungan pada atribut paket. Walau bagaimanapun, beberapa kajian telah berusaha untuk memperbaiki pokok indeks berasaskan Grid dengan membahagikan kawasan rangkaian menjadi grid. Prinsip asas yang mendasari pembahagian adalah membina struktur indeks dengan menentukan sensor bergerak yang berada di dalam setiap grid, kemudian melakukan pengindeksan mengikut atribut paket di setiap grid. Walaupun begitu, adalah lebih praktikal untuk meningkatkan prosedur pengindeksan sensor mudah alih rawak, kerana cabaran berikut. Pertama, kebolehpercayaan tinggi pada pelbagai atribut (lokasi dan masa) paket dalam sensor mudah alih rawak. Kedua, dalam lingkungan rawak, diperlukan untuk mendaftarkan lokasi grid setiap sensor dan menetapkan tujuannya secara berkala. Secara khusus, tesis ini mencadangkan model yang efisien yang disebut Dynamic Static Coalition-based (DySta-Coalition for short) untuk meningkatkan prosedur pengindeksan sensor mudah alih rawak. Model yang dicadangkan terdiri daripada kerangka kerja Dinamik-Koalisi, algoritma Static-Coalition, dan kerangka Indeks-Pohon Berasaskan Koalisi. Kerangka dinamik-koalisi menetapkan perkaitan antara sensor dan gerbang untuk membina blok dinamik yang disebut koalisi dinamik. Ini bertujuan untuk meningkatkan paket pengindeksan dan mengurangkan data yang disebarkan. Algoritma Static-Coalition menyusun semula paket di setiap pintu masuk, sesuai dengan sumbernya. Oleh itu, ia menyusun paket, langkah demi langkah sebelum terkumpul di pelayan utama untuk diindeks. Kerangka Indeks-Pokok Berasaskan Koalisi adalah proses pengindeksan untuk sensor mudah alih rawak. Akhirnya, prestasi model DySta-Coalition dinilai secara komprehensif pada pelbagai set data sintetik dengan pelbagai parameter (bilangan sensor mudah alih, kelompok dan dimensi) dan metrik (Nisbah Penghantaran Paket, Overhead Laluan, overhead kos ruang, masa pembinaan indeks, dan jumlah nod melintasi). Pertama, menilai kerangka kerja Dynamic-Coalition dan Static-Coalition, dari segi Purata Routing Overhead dan Packet Delivery Ratio. Juga, menilai kerangka Indeks-Pohon Berasaskan Koalisi dari segi bilangan nod yang dilalui, overhead kos ruang, dan masa pembuatan indeks. Penilaian kedua terbahagi kepada tiga senario. Yang pertama menilai kerangka Indeks-Pohon Berasaskan Koalisi secara bebas, tanpa kesan dari kerangka kerja Dinamik-Gabungan dan algoritma Static-Coalition. Senario kedua dan ketiga menilai prestasi Indeks-Pokok Berasaskan Koalisi dengan meneliti kesan kerangka kerja Dynamic-Coalition dan algoritma Static-Coalition pada kerangka Coalition-Based Index-Tree. Hasil penilaian menunjukkan bahawa model DySta-Coalition dapat mengurangkan paket yang disebarkan, yang mengakibatkan meminimumkan ketergantungan pada atribut paket. Lebih jauh lagi, ia adalah yang tertinggi menurut hasil overhed masa dan kos ruang indeks.

Keywords: Coalition, multi-attribute, spatial-temporal index, random mobile data, wireless sensor networks (WSNs), Grid-based indexing.

Universiti Malaya

Universiti Malaya

ACKNOWLEDGEMENTS

It is my honor to give thanks to the almighty GOD, Allah, for giving me the strength and ability to complete this thesis. I also owe my deepest gratitude to my mother, *Fatemah*, and my late father, *Jihad* for unlimited love and support. I also want to acknowledge my supervisors, *Dr. Sri Devi Ravana* and *Dr. Ali Mohammed Mansoor* for their invaluable advice, guidance, supervision, encouragement, and overall patience with me throughout my study.

Universiti Malaya

TABLE OF CONTENTS

Universiti Malaya

Universiti Malaya

Universiti Malaya

Universiti Malaya

Universiti Malaya

LIST OF FIGURES

Universiti Malaya

Universiti Malaya

LIST OF TABLES

Universiti Malaya

Universiti Malaya

LIST OF SYMBOLS AND ABBREVIATIONS

CQI	:	Channel Quality Indicator
D-tree	:	Decomposition tree
DySta-Coalition	:	Dynamic-Static Coalition-based
EEDR	:	Energy Efficient Distributed Receiver
GDCF	:	Grid-based DBSCAN with Cluster Forest
PDR	:	Packet Delivery Ratio
WSN	:	Wireless Sensor Network

Universiti Malaya

CHAPTER 1:

INTRODUCTION

Over recent years, Wireless Sensor Networks (WSNs) have attracted researchers, with a wide range of utilities in the era of remote monitoring. WSN is a structure that involves multiple stationary or mobile sensors' nodes deployed geographically. The ubiquitous nature of nodes in WSN results in new environments of communications, which varied from the previous ones. One of those new environments is mobile WSNs that arouses the interest of researchers.

WSN mobile nodes are distributed remotely, such as sensor nodes used in the wild territories, military, and firefighting. In some of these applications, the sensor nodes have random mobility, where the sensor nodes move randomly and transfer packets from different locations and time periods within a specific area. Hence, indexing packets of random mobile WSNs is a critical issue, where the conventional techniques are not applicable to such environments. Indexing random mobile packets per transferring location and time are necessitating. This study addresses the indexing packets of random mobile WSN.

This chapter provides a comprehensive overview of the research. It presents the theoretical background for the research on indexing packets of random mobile WSN. This chapter covers the research background, motivation, problem statement, and specifies the research scope. Moreover, the chapter specifies the objectives and significance of this study.

The remainder of this chapter organizes as follows. Section 1.1 represents the background of the study and Section 1.2 represents the motivation. The statement of the problem presents in Section 1.3. Meanwhile, the research questions and objectives represent in Section 1.4. Research scope and significance in Section 1.5 and 1.6,

Universiti Malaya

respectively. As well as, Section 1.7 shows the summary of the contributions. Last but not least, Section 1.8 shows the outline of the thesis.

1.1 Research Background

The sensor is a thin device that can smoothly measure physical phenomena upon certain conditions, such as heat, light, fire, sound, or any changing environment, and converting the measurement values to digital ones. Real applications use different types of sensors. For example, medical care sensors are used to measure pressure, and weather radars are used to measure humidity, temperature, and wind power.

WSN consists of a large number of sensor nodes. It defines as a network including stationary or mobile sensors geographically distributed for monitoring. These monitoring sensors transfer measurement values as packets periodically to the targeted destination. As presents in Figure 1.1, the components of mobile WSN are sensors and gateways nodes, in addition to the server. The mobile sensor nodes transfer packets to several mobile gateways nodes, and then from the gateways to the server.

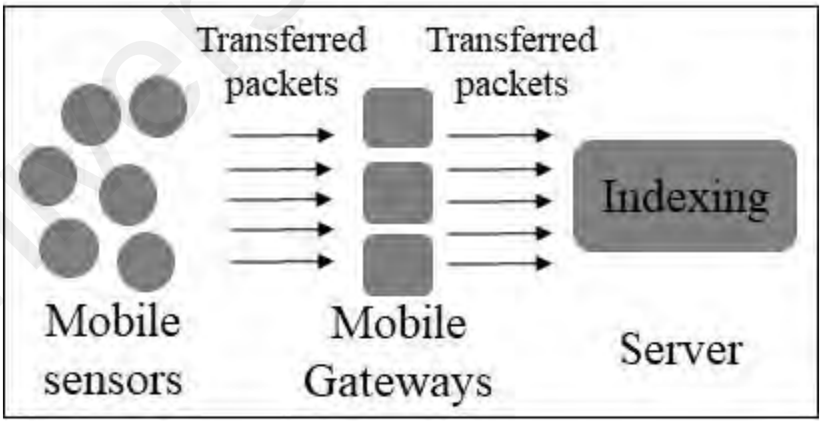

Figure 1.1: Mobile Wireless Sensors Networks components

The mobility of sensors defines as a process of changing the sensor nodes' location throughout time (Chen, N. 2015). The benefits of the sensor node's mobility are the ability to discover new areas by changing locations per time period. In real applications, there are three mobility types of sensors' nodes in WSN. First, if the sensors move within a specific path, this called predefined mobility (Chakrabarti, Sabharwal, and Aazhang 2003) (W. Wang, Srinivasan, and Chua 2007). Second, controlled mobility, where the sensors have an external control to direct the sensors (Somasundara, Ramamoorthy, and Srivastava 2004) (Eylem Ekici, Yaoyao Gu, and Doruk Bozdag 2006). Third, random mobility is contrary to controlled mobility, which applies in diverse areas such as monitoring animals in a wild territory (Shah et al. 2003). In wild territory, each animal carries its sensor that travels randomly within a monitored area.

Sensor nodes of WSN communicate and collaborate to transfer the measurement values remotely to the destination as packets. The transferring packets of mobile sensors require to be stored, processed, and presented in a seamless, efficient, and easily interpretable form. Thereby, it requires the optimal indexing method.

The indexing procedure is a mechanism that collects, analyzes, stores, and arranges the data according to specific circumstances. There are several research efforts have been dedicated to improving the indexing processes in mobile environments (Babenko and Lempitsky 2015) (Puri and Prasad 2015) (Alvarez et al. 2015) (Amato et al. 2015). One of the core methods proposed to index the mobile sensor's data is Grid-based indexing. It is a method dedicated to index sensors' data according to temporal and spatial attributes (Ding et al. 2012). Spatial-indexing is indexing the data based on the location of the sensor. Meanwhile, temporal-indexing indexes the packets upon the time-stamps, which are the transferring time of the packets.

1.2 Research Motivation

The evolutional growth of WSNs extends the boundaries of the digital world that opens up new horizons of knowledge. At present, WSNs embrace different applications, such as those in the military, etc. The innovation of WSNs has resulted in the integration of various types of technologies, like Internet of Things (IoT), big-data, and cloud computing.

Universiti Malaya

In mobile WSNs, the sensors' nature needs to report real-time packets periodically to the server, per small period of time. This requires arranging and manage those packets in a seamless and interpretable form by indexing, based on packets attributes. This is because the indexing process needs to check each transferring packets' location and time-stamp (attributes) periodically. Many works have tried to improve the indexing processes, such as Frequent Updates in R-Trees (FUR-tree) (M. L. Lee et al. 2003), R-trees with Update Memos (RUM-tree) (Xiong and Aref 2006), and RGP-tree (R-tree with dynamic non-uniform grid and sub-R tree) (K. Y. Lee, Kang, and Kim 2014). Unfortunately, those works did not tackle the issues of the indexing for random mobile sensors, because it needs to minimize dependability on the attributes of packets.

In random mobile WSNs, the number of generated packets is large (Guo et al. 2014), because the indexing process considers each packet as a new one. This is because it comes from different locations and time periods. This results in the need to check the packets one by one, not periodically, to provide accurate indexing. Accordingly, the dependability of random mobile packet indexing on the attributes of the packet is exacerbated.

This work dedicates an effort to address one of the most critical research problems. That is indexing packets of random mobile WSN. It is expected that this proposed research will improve the indexing processes of sensors' packets in a random mobile environment, by mitigating the dependability of the packets' attributes.

1.3 Real-time packets indexing scenario

This section discusses the reasons for disseminated packets by drawing a scenario that fits the problem statement (discusses in the next section), to form a comprehensive perception of impact on the performance of indexing.

In WSN the packets are received by the gateways from sensors and then transfers into the server. Moreover, sensors and gateways have random mobility that affects the transferred packets by transferring from different locations and periods of time. In other

Universiti Malaya

words, each gateway may exist packets that are transferred from the same sensor. Within these random mobility characteristics, the disseminated packets throughout destinations are increased. This dissemination considers the main factor that affects negatively the performance of indexing.

According to the aforementioned scenario, the reasons for disseminated packets are as follows. First, the behavior of random mobile sensors, in terms of collecting packets. This is because each packet that is transferred from sensors has attributes (location and time). Second, the packets that belong to the same source (sensor) are accumulated in the server, which increases the burden of the indexing process. Furthermore, without any arrangement and management of the transferred packets before transferring to the next destination, effects adversely indexing performance. More explicitly, the packets are collected directly without any arrangement and management or at least minimize the complexity of packets i.e. removing duplicates, increases the overhead of packets indexing.

Third, the challenges of connectivity between source and destination. This will increase the disseminated packets by considering the transferring of packets by taking into account the routing data. This will increase the disseminated packets throughout destinations.

Forth, the challenge of clustering in the mobile sensor environment. More explanation, current works based on the division of the network area into grids called Minimum Bounding Rectangles (MBR). Further explanation, let say that R-tree index is a technique used to index the sensors in random mobile WSN. R-tree executes indexing by dividing the network area into grids each one called MBR. Each MBR includes sensors, which considers as a parent for each of them. Accordingly, the number of sensors within each MBR will increase the effects on index updating overhead. In addition, when the sensors are moved from one MBR to another, it increases the index building time and updating overhead. Moreover, R-tree requires to check the source of each packet, which affects adversely the performance of an index such as building time.

Based on the above, indexing techniques that conduct structure depends on area division are not suitable for random mobile WSNs. This analysis assumes that the disseminated packets that belong to the same sensors will affect the performance of indexing.

1.4 Limitations of the current works

Grid-based DBSCAN with Cluster Forest (GDCF) (Boonchoo et al. 2019), R-tree (Antonin Guttman 1983) and Decomposition-tree (D-tree) (Chen, N. 2015) try to solve indexing challenges in mobile sensors' nodes – describes in detail in Chapter 2, Section 2.2-. GDCF partitions a space layout into grids by utilizing the HyperGrid Bitmap (HGB) structure. HGB enables indexing grids that include objects and neglects empty ones. It provides neighbor grid merging using the union-find algorithm and cluster forest. D-tree attempts to index multidimensional mobile data to alleviate the high space cost in its inner nodes. It applies a hierarchical index structure that compromises B-tree, and R-tree. The D-tree structure can reduce the time taken for accessing memory by constructing its structure without inner nodes. Furthermore, R-tree bases on partitioning the network area into grids and based on the attributes of the moving sensors' nodes.

Although the referred tree-based index techniques have high popularity in traditional indexing, they fail to efficiently manage the packets of random mobile sensors. This failure is caused by many factors. First, the nature of the reported packets is considered a type of big data. Second, current indexing techniques depend on the partitioning of network areas to build the index structure. Third, they fail to detect unpredictable behaviors, such as random mobility, in which the type of mobility must be considered. These factors increase the indexing overhead because they provide considerable up-

Universiti Malaya

dating (read-write) operations, especially in random environments. The following measures can be performed to avoid the challenges brought by the previous factors. First, the index-tree performance is improved by mitigating the disseminated packets throughout the destinations. Second, the random mobile sensors' packets not only refer to big data but also cover the packet attributes on which Grid-based indexing is based.

So far, all of the aforementioned challenges reach us to highlight and define the statement of problem statement, which will describe in detail in the next section, Section 1.5.

1.5 Problem Statements

Several studies have proposed for indexing of packets transmitted via mobile WSN network nodes, such as Grid-based indexing (Boonchoo et al. 2019). Grid-based indexing indexes the mobile packets based on packets attributes (transferring locations and transferring time), as mentioned in Section 1.2 and Subsection 2.3.2. Such studies base on dividing the network region into grids, as presented in Figure 1.2. As in the figure suppose the network layout –labeled as A in the figure- contains five mobile sensors S_1, S_2, S_3, S_4, and S_5. The current mobile indexing techniques stand on dividing the network layout into grids. As presented in part B of Figure 1.2, the network layout dividing into four grids that are Grid 1, Grid 2, Grid 3, Grid 4. The benefits of dividing the network layout to control mobile sensors for facilitating index structure construction. Accordingly, each node of the constructed index structure is represented by grids, and the sensors (or the transferred packets from each sensor) are added in the appropriate node according to the grid that belongs to, as represented in Figure 1.3.

Universiti Malaya

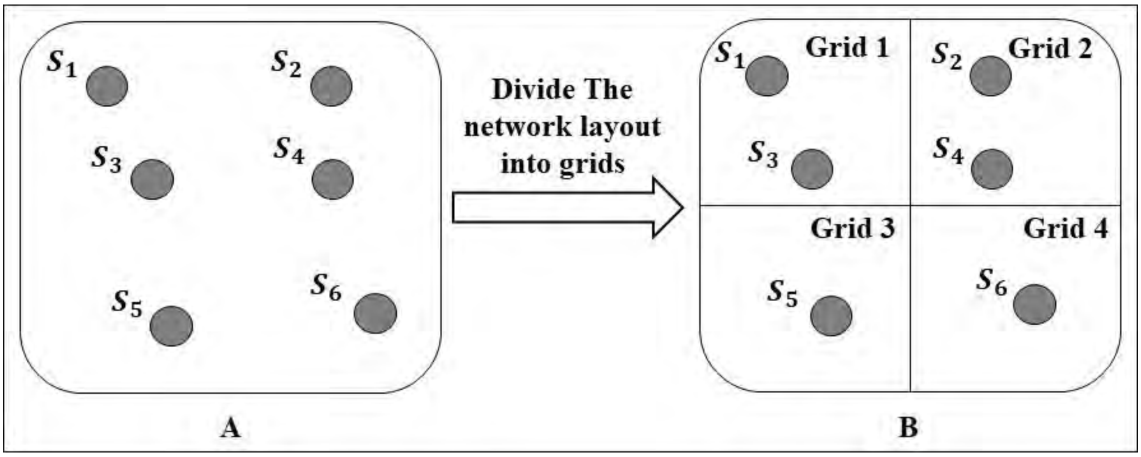

Figure 1.2: Grid-based indexing

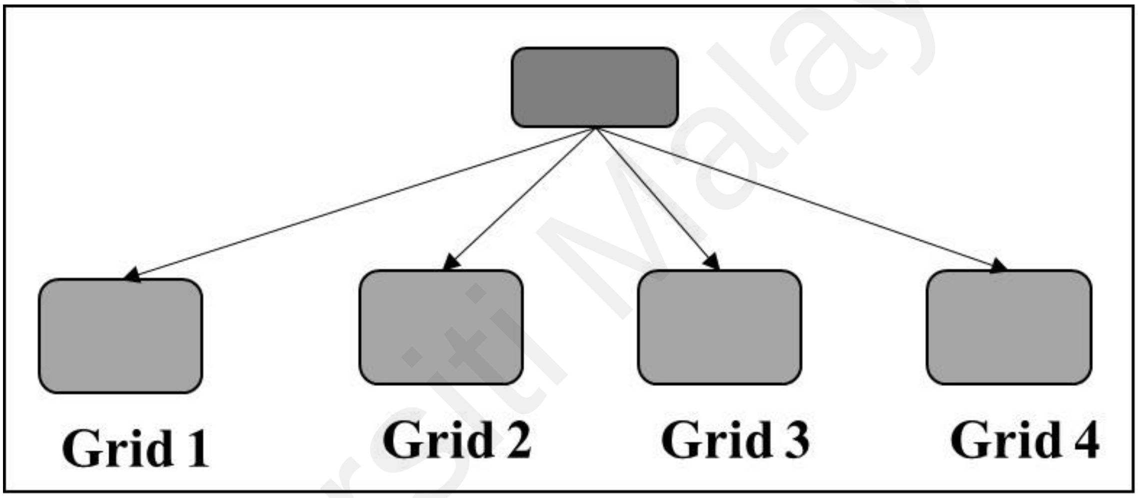

Figure 1.3: The structure of Grid-based indexing

Dividing the network area into grids for indexing mobile packets results in disseminated the packets throughout destinations (Jeongcheol Lee et al. 2018). Packets that belong to the same source (sensor) disseminate throughout the different destinations, as shown in Figure 1.4. I.e. Active data Dissemination (Jeongcheol Lee et al. 2018) transfers packets according to the shortest path algorithm, which increases the disseminated packets, as mentioned. This problem increases indexing overhead because the time for reading the packet's attributes for indexing is increased. This result adversely affect the number of updating operations of the index-structure.

Universiti Malaya

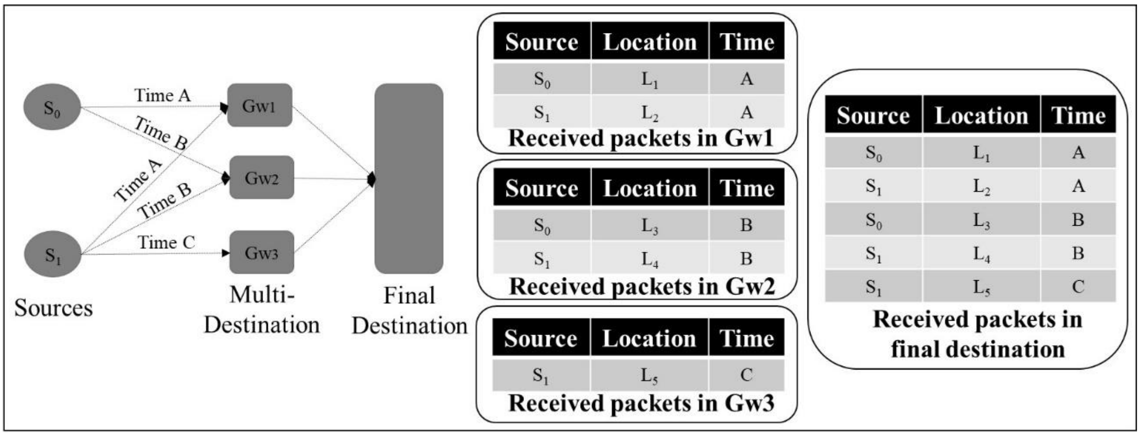

Figure 1.4: Transferred packets from the same source are disseminated throughout the destination

Another problem in indexing random mobile packets is the need to manage and arrange packets (Boonchoo et al. 2019) (Tang et al. 2017). The destination has misarranged packets that belong to the same source, as shown in Figure 1.5. These misarranged packets affect the performance of the index. Hence, the need to collect and arrange the packets that belong to the same source is required to mitigate the overhead of the indexing processes. So, managing and controlling the transmitted packets is required to improve the mobile sensor's indexing. Thereby, it is mitigating the burden and time required to collect and read the received packets, to increase the indexing efficiency.

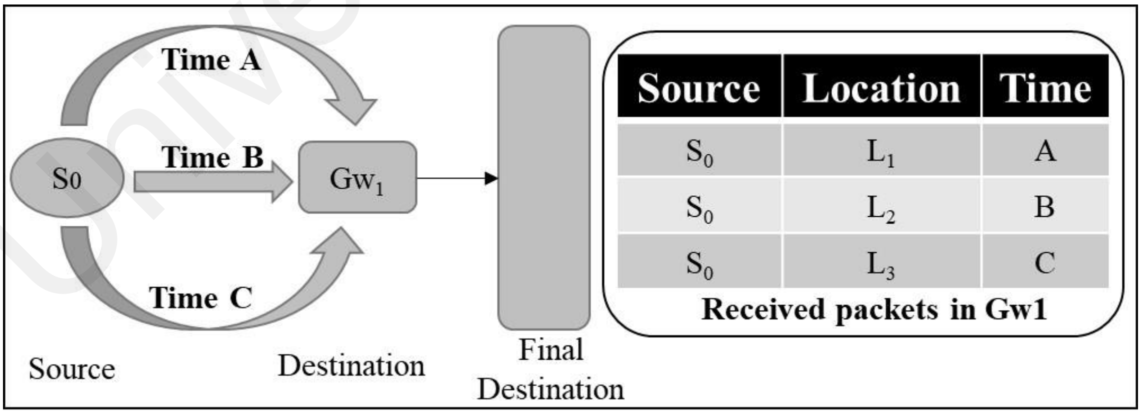

Figure 1.5: Misarranged packets that belong to the same source

In a random mobile environment, each destination requires registering the current network division grids information for each source periodically, and each source needs to assign the destination, in order to transfer packets successfully (Boonchoo et al. 2019)

(Tang et al. 2017). This issue increases the dependability on each attribute of sensors and packets, which results in increasing indexing overhead.

Figures 1.4 and 1.5 represent an example of the resulted packets that receive to the final destination as a consequence of the previous challenges. Figure 1.6 represents the environment of random mobile sensors, labeled as Part a. The environment consists of deployed sensors that move randomly, labeled as S_0, S_1, S_2, and S_3. The sensors transfer packets to random mobile gateways, labeled as Gw_1, Gw_2, and Gw_3. The details of transferring packets for each sensor represents in Part b of Figure 1.6. S_0 transfers packet at Time B and Time C to gateway 1. Also, at Time A and Time C to gateway 2 and gateway 3, respectively. The detail of received packets to each gateway shows in Figure 1.7, Part b. As noted, the problem lies when the destination receives packets from different time periods and locations but belong to the same source. I.e. S_0 in gateway 1, and S_1 in gateway 2. This considers as a source of the problem because it affects negatively the performance of indexing when it accumulated on the final destinations for indexing.

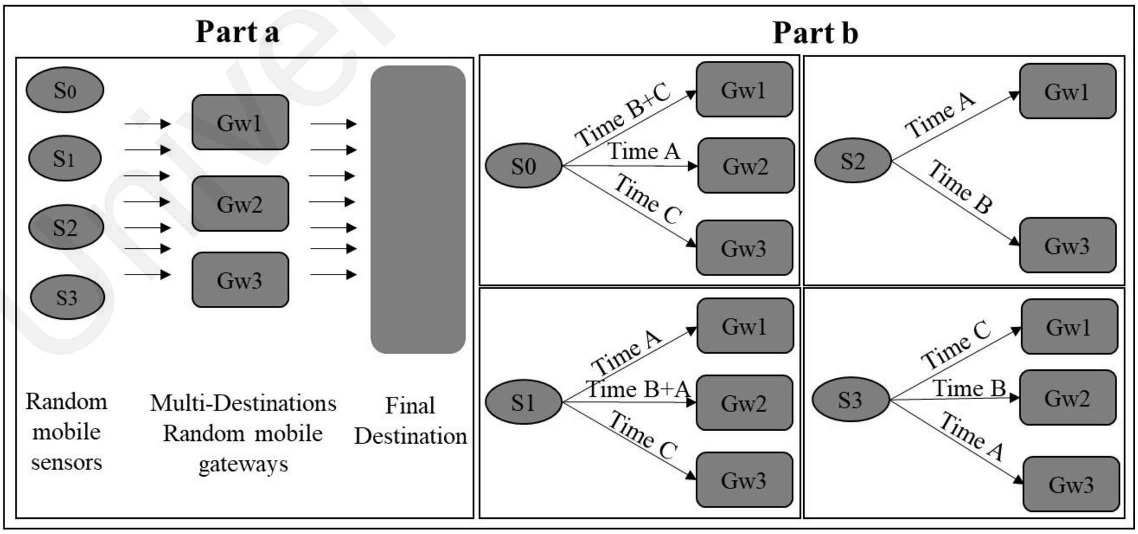

Figure 1.6: Part a: Random mobile sensors environment. Part b: The transferred packets for each sensor to the gateways

Universiti Malaya

Part a

S_{id}	Time	Location
S_0	B	(X_1,Y_1)
S_1	A	(X_2,Y_2)
S_2	A	(X_3,Y_3)
S_3	C	(X_4,Y_4)
S_0	C	(X_2,Y_{25})

Received Packets in Gateway 1

S_{id}	Time	Location
S_0	A	(X_5,Y_5)
S_1	A	(X_{41},Y_{11})
S_1	B	(X_2,Y_2)
S_3	B	(X_3,Y_3)

Received Packets in Gateway 2

S_{id}	Time	Location
S_0	C	(X_{13},Y_{11})
S_1	C	(X_2,Y_2)
S_2	B	(X_3,Y_3)
S_3	A	(X_3,Y_3)

Received Packets in Gateway 3

Part b

S_{id}	Time	Location
S_0	B	(X_1,Y_1)
S_1	A	(X_2,Y_2)
S_2	A	(X_3,Y_3)
S_3	C	(X_4,Y_4)
S_0	C	(X_2,Y_{25})
S_0	A	(X_5,Y_5)
S_1	A	(X_{41},Y_{11})
S_1	B	(X_2,Y_2)
S_3	B	(X_3,Y_3)
S_0	C	(X_{13},Y_{11})
S_1	C	(X_2,Y_2)
S_2	B	(X_3,Y_3)
S_3	A	(X_3,Y_3)

The accumulated Packets in Server

Figure 1.7: Part a: The received packets in each gateway. Part b: The accumulated packets in the final destination (server)

Universiti Malaya

As a summary, the problem statement can be identified as the following:

- Problem statement 1: Dividing the network area into grids for indexing mobile packets results in disseminated the packets throughout destinations (Jeongcheol Lee et al. 2018).
- Problem statement 2: The need to manage and arrange packets that belong to the same source in the destinations (Boonchoo et al. 2019) (Tang et al. 2017).
- Problem statement 3: Random mobile environment increases the dependability on each attribute of sensors and packets, which results in increasing indexing overhead (Boonchoo et al. 2019) (Tang et al. 2017).

1.6 Research Questions and Objectives

This study was undertaken with the aim to improve the indexing procedures by managing and controlling the random mobile sensors' nodes in addition to packets. More specifically, this study identifies the relationship between the behavior of mobile sensors and the performance of packets' indexing of random mobile WSN. In order to address the role of alleviation of the disseminated packets in improving the indexing of random mobile environments. This section presents the Research Objectives *(ROs)* of this research to achieve the aim. Furthermore, it addresses Research Questions *(RQs)*, which uses as a guide at various stages and to fulfill the research objectives. The set of questions (*RQs*) that answered by this thesis is as follows:

RQ1: What are the issues that adversely affect the indexing of random mobile sensors in WSN?

RQ2: How to minimize the sensors Routing Overhead by transmitting to less number of directions?

RQ3: How to mitigate the packets of the sensors from scrambled?

RQ4: How to minimize the number of indexing operations, which affected by the disseminated packets?

RQ5: How to reduce indexing operations overhead (number of traversed nodes) for data transmitted from gateways?

RQ6: What is the performance of the proposed model against others?

This study is achieved through the following set of Research Objectives (ROs):

RO1: To study and identify the current Grid-based indexing techniques and how there have affected by the behavior of random mobile sensors.

RO2: To minimize random mobile sensor overhead, by initiating a relationship between sources and destinations**.**

RO3: To improve the index operations of the system, by maximizing the number of transferred packets from each sensor to a specific destination.

RO4: To reduce the number of indexing operations, in order to minimize the number of traversed index nodes during index construction.

RO5: To effectively build-up static-coalitions in each gateway to reduce the number of indexing operation overhead in terms of space-cost and index building time.

RO6: To validate the proposed model using practical analysis and compare its performance with the best-known competitors.

1.7 Scope of the research

This study addresses the scope of Grid-based index structure, in order to improve packets' indexing in random mobile WSN. It focuses on improving the tree-based index structure, as shown in Figure 1.6. Specifically, the effects of dissemination packets throughout destinations are addressed (Jeongcheol Lee et al. 2018), which leads to adverse effects on the packets indexing. In other words, this work tried to improve the tree-based indexing process in random mobile WSN by mitigating and arranging the disseminated packets throughout gateways. For this purpose, a novel model is proposed

Universiti Malaya

aiming to mitigate the disseminated packets and arrange the packets step-by-step before they accumulated in the main server for indexing.

The proposed model is applied to a synthetic dataset, according to the number of mobile sensors, clusters, and dimensions parameters. Accordingly, the ability of the proposed model is examined to mitigate the disseminated packets. Consequently, it examines the effects of mitigation and arranging packets on the performance of indexing.

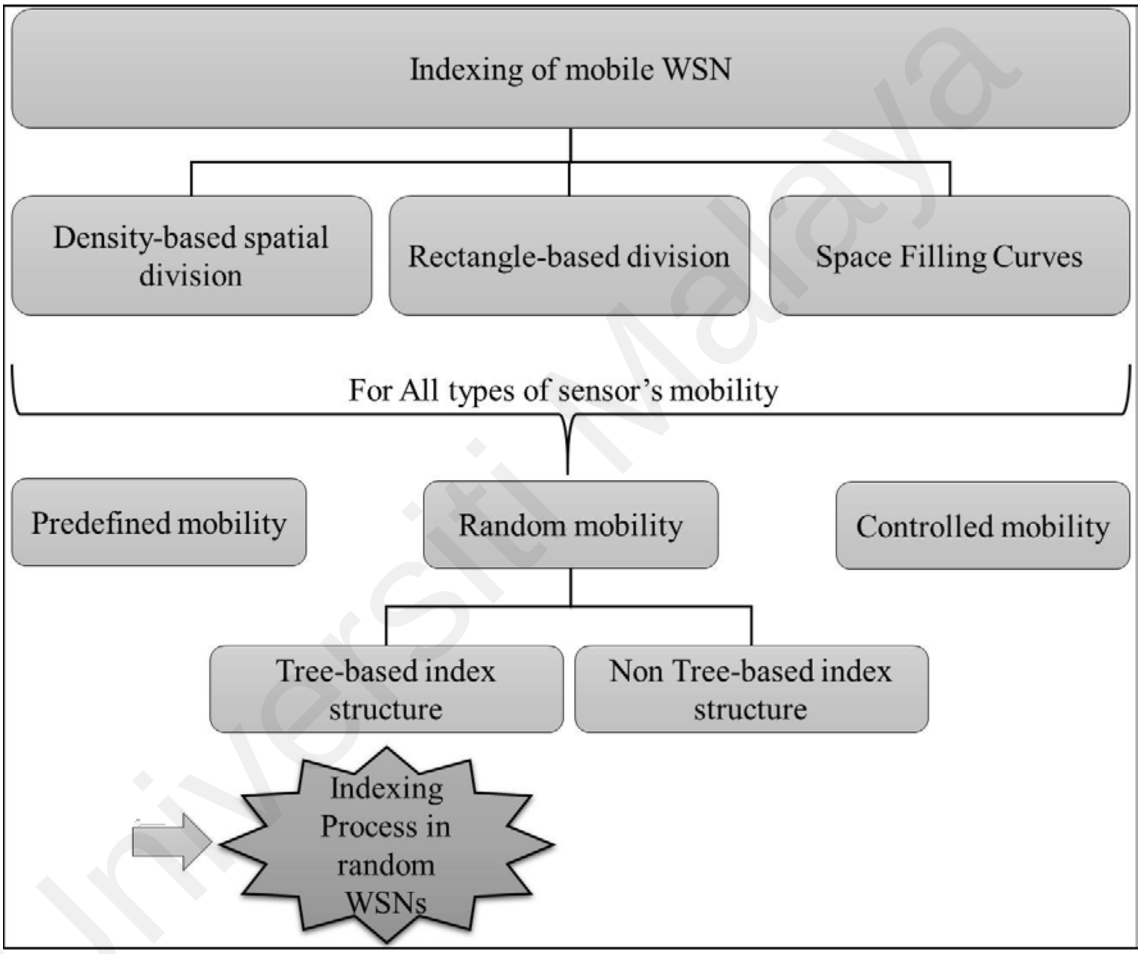

Figure 1.8: Scope of the research

1.8 Research significance

This thesis offers contributions that would have significant benefits in improving indexing processes in random mobile WSNs, as follows:

- Proposing a method that mitigates the disseminated packets throughout destinations, in order to minimize the indexing overhead in the final destination (server).

- Proposing a method to arrange packets step-by-step in the same way of indexing procedures, before the packets arrive at the final destination for indexing. This results in alleviating the index overhead and removing the redundancy of packets.
- Proposing an indexing process that does not depend on the attributes of the packet. This results in its suitability for random environments.

1.9 Thesis Contribution

To solve the aforementioned problem, in Section 1.3, this work proposes a model called DySta-Coalition, which consists of Dynamic-Coalition framework, Static-Coalition Algorithm, and Coalition-Based Index-Tree framework. Accordingly, the contributions of this thesis to the current literature are:

- **Dynamic-Coalition** framework is proposed to minimize the number of disseminated packets throughout gateways, by constructing dynamic blocks based on the coalition, called dynamic-coalitions.
- **Static-Coalition** algorithm is proposed by initiating **static-coalition** in each gateway that alleviates the effects of the Grid-based index structure and removing the redundancy of packets, which results in arranging and preparing data step-by-step before it accumulated in the final server for indexing.
- A variant indexing-tree, called **Coalition-Based Index-Tree** is proposed, which replaces the partitioning of the network area by dynamic-coalitions.

Applying the aforementioned contributions, the effectiveness of the indexing processes will increase, especially in applications whose data are transferred from different locations, and thus the speed of data retrieval increases with a lower cost and greater effectiveness.

Universiti Malaya

1.10 Outline of the Thesis

This thesis consists of six chapters that systematically explain the objectives of this research. Each chapter achieves a portion of the research study. Therefore, this section states the structure and organization of this thesis. Table 1.1 demonstrates the structure of this thesis.

Chapter 1 provides a comprehensive view of this thesis. It presents a motivation behind this work, problem statement, research questions, research aim and objectives, and a brief description of the contributions and significance of this work.

Chapter 2 is a literature review, in which the current studies and related works for mobile WSN are discussed.

A roadmap of the research is drawn in **Chapter 3**, by describing the phases of research methodology that have been followed. As well, it explains the problem formulation and the proposed model and its structure, and the relationships among the proposed model parts of the structure, in order to solve the aforementioned problem.

Chapter 4 and Chapter 5 describe the implementation details for the evaluation of the proposed model. Also, it explains the tools used for testing the proposed model. Furthermore, the experimental results are presented. It also provides comparisons of the results across the experiments and the other baseline methods.

Chapter 6, represents the conclusion of this thesis. A summarize of the research work is accomplished in this chapter. Furthermore, the chapter summarizes the contributions of the thesis and future research direction.

Table 1.1: Thesis Layout

Chapter name	Importance of the Chapter	Results
Chapter 1: Introduction	1. Stating and highlighting the general indexing techniques, and gives some background. 2. Identifying the research problems, questions, and pre-setting the objectives.	a) Identifying the research problem. b) Writing the research questions and objectives.

Universiti Malaya

	3. Highlighting the significance of the research. 4. Providing the thesis layout.	
Chapter 2: Literature Review	1. Reviewing related literature review. 2. Figuring out the strength and defects of the related work. 3. Detecting the open research problem.	a) Criticizing the existing work. b) Linking the current work with the literature.
Chapter 3: Research Methodology	1. Drawing a roadmap of this thesis. 2. Providing the model perspective of the proposed framework. 3. Highlight the evaluation works. 4. Generating the model that is proposed in the research methodology. 5. Explaining the model functions.	a) Outlining the research methodology. b) Formulating the problem. c) DySta-Coalition Model.
Chapter 4 and Chapter 5: Implementation and Evaluation	1. Presenting the experimental evaluation setup. 2. Explaining the tools used for testing the proposed model. 3. Highlighting the strength of the proposed framework. 4. Analyzing the experimental results.	a) Describing the implementation of the proposed model. b) Comparing the proposed model with other competitor baselines.
Chapter 6: Conclusion	1. Stating the contribution of the research work. 2. Explaining the work limitations. 3. Providing a sneak-peek into future work.	a) The research questions are already answered. b) Examining to check if the research objectives were achieved.

Universiti Malaya

CHAPTER 2:

LITERATURE REVIEW

This chapter investigates the current indexing techniques and reviews the detailed background of the traditional solutions in mobile WSNs. Accordingly, this chapter starts with the limitations of the current indexing techniques by investigating the challenges that negatively affect the effectiveness of those techniques from the mobility perspective. Furthermore, this chapter continues by identifying the existing approaches of mobile sensor environments to understand and analyze the existing challenges of such environments that adversely affect the indexing process. To simplify that, the approaches are classified into three categories, (i) meta-heuristic approaches, (ii) Grid-based indexing approaches, and (iii) prediction model approaches. Additionally, this chapter describes each indexing technique by discussing its ability to overcome the challenges in each category. Eventually, this chapter ends with summarising and highlighting the limitations and future works. To simplify the flow of this chapter's structure, see Figure 2.1.

The rest of this chapter organizes into seven sections, as follows. Section 2.1 and Section 2.2 highlights backgrounds and the current indexing techniques, respectively. Section 2.3 represents the classification of mobile sensors techniques in WSN. The challenges of the techniques from the mobility perspectives are investigated in Section 2.4. A comparison of indexing techniques based on the gaps in mobile sensors is proposed in Section 2.5. The relationship between the behavior of the sensors and the performance of indexing techniques is presented in open research issues and future directions, Section 2.6. Finally, this chapter is summarized in Section 2.6.

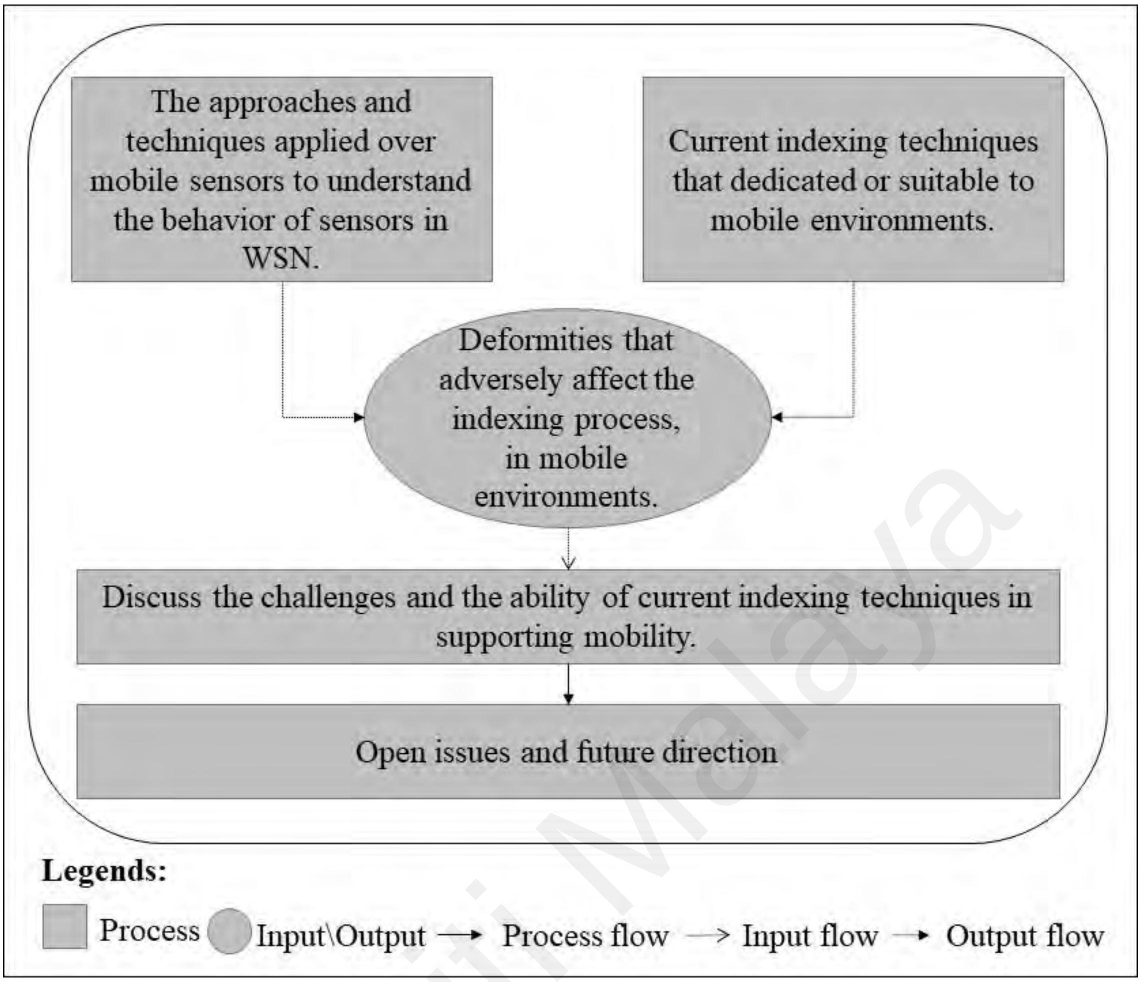

Figure 2.1: Structure of Chapter 2

2.1 Background

This section describes the core terms that recur in this thesis. The first term is indexing. Indexing is a structure that improves the speed of retrieval operations at the cost of additional writes and storage space. It is used to do quickly locate the packets in the index structure without having to search every received packet every time. Thereby, indexing supports a fast lookup. Generally, a good index structure has to have high performance, which defines as how well the index is meeting its defined processes. The index processes are inserting, updating, and deletion. Also, high index performance fulfills by efficient index operations. Efficiency signifies a peak level of the index performance that uses the least amount of inputs to achieve the highest amount of output. In other words, efficient index processes represent by many metrics, such as low space-cost, low index-building time, and low number of traversed nodes. These metrics define in detail in Section 5.2.

Another term uses to identify the performance is overhead. Overhead signifies any combination of excess or indirect computation time, memory, or other resources that are required to perform a specific index operation. The small overhead value is better than the high. Accordingly, this thesis uses space-cost overhead, index-building time overhead, and the number of traversed nodes overhead, to evaluate the performance of the index.

According to the structure of WSN, the need for defining the population of WSN and its concepts appeared. Hence, this subsection represents the definitions that are frequently used in this chapter and the remainder of this thesis.

- **Source:** is a node that generates packets and transfers them to the appropriate destinations. In the structure of WSN, the sources are either sensors or gateways. Hence, when the sensors are transferring packets to the gateways the sensors are considered as a source. Meanwhile, when gateways are transferring packets to the server the gateways are considered as a source.
- **Destination**: this considers the second part of the WSN population, which is nodes that received packets from the source. In the structure of WSN, the destination is either gateways or server.
- **Server:** is a final destination in the WSN, where the overall packets are accumulated and the final indexing process is conducted in.

Identification of terms and concepts, which are frequently used as follows.

- **Tree-based index:** is a structure that consists of a set of linked nodes. The tree-based index handles data by its operations, which are (i) insert, (ii) update, and (iii) delete. In other words, it stores data and helps search, access, insertions, and deletions. Furthermore, it consists of a root node and a sub-tree. Sub-tree

Universiti Malaya

consists of children, which corresponding to the parent node. The perfect tree-structure that has no inner nodes, as discussed in Subsection 4.3.3.

- **Movement (mobile) nodes**: nodes are the population of the network, which moves from one location to another. Mobility is a process of changing the node's location over time (He et al. 2015). The nodes' mobility has three types, mentioned in Section 1.1, and this work focuses on random mobile movement. Hence, WSN consists of a set of mobile nodes that travel from one location to another. Those nodes are connected via Wi-Fi connections. They are capable of requesting and receiving the required packets. According to the structure represented in Subsection 3.2.1, the mobile nodes are the sensors and gateways nodes of the network.
- **Random Waypoint Model**: is a mobility model that the direction of the mobile sensors is chosen randomly without and dependency on previous values.
- **Spatial-temporal indexing:** is an advanced index structure, where the indexing key is the location and time-stamp (attributes) of an object, mentioned earlier in Section 1.1.
- **Coalitions:** a coalition is a set of agents, which cooperate to complete a complex task. Each coalition is associated with a task. A valid coalition should satisfy the situation that the resources of agents in the coalition should cover the required resources of the associated task (Ye, Zhang, and Sutanto 2013).
- **Disseminated packets:** defined as received packets by destinations that transferred from the same source, or packets received by the specific destination that transferred from different sources.

2.2 Current Indexing techniques

This section presents an in-depth investigation of the existing indexing techniques to grasp challenges from the mobility perspective. To simplify that, the investigated techniques in this section are classified into (i) tree-based or (ii) non-tree based, as in Figure 2.2.

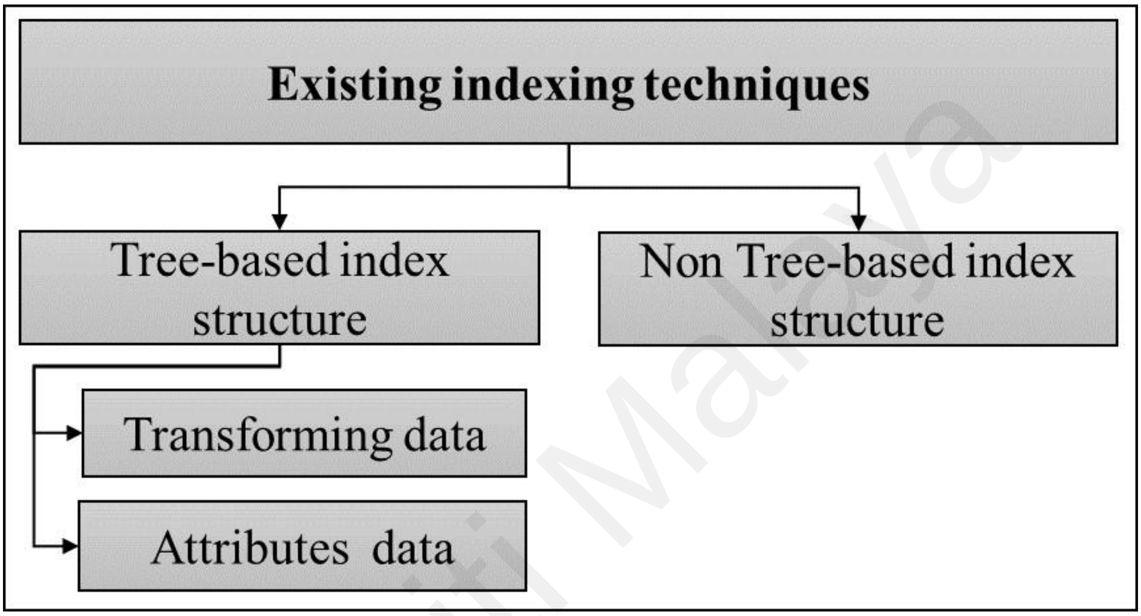

Figure 2.2: Existing indexing techniques classification

2.2.1 Tree-based Indexing

Figure 2.2 depicts the classification of the existing indexing techniques, in accordance with the indexing structure construction. They are constructed either by Transformed-data or by Attributes-data (Badarneh, Ravana, and Mansoor 2020). Transforming data means converting the reported data into a new form such as a bit, to use it in insertion or updating of the tree-index structure. As examples of this kind of indexing are Decomposition-tree (D-tree) (Chen, N. 2015) and Sparse Hashing (SH) (X. Zhu et al. 2013).

The transmitted packets have attributes like transferring time and location. Accordingly, the indexing techniques based on packets' attributes are divided into three parts. (i) Spatial-indexing is based on the locations of sensors that transfer packets to

Universiti Malaya

construct an index structure. (ii) Temporal-indexing constructs the index structure based on the time of transferred packets. Meanwhile, spatial-temporal indexing is a mix of them.

2.2.1.1 Transforming data

There are indexing techniques suitable for indexing according to specific conditions. For example, Hash-based indexing techniques fulfill a fast similarity search by using binary-codes to represent high-dimensional data, as Sparse-Hashing (SH) (X. Zhu et al. 2013). Hash-based indexing techniques have proposed to improve search efficiency in the context of high dimensional data. Hash-based indexing techniques are still suffering to effectively and efficiently generate binary codes and encode binary data. To overcome the mentioned challenges, Sparse-Hashing (SH) is proposed (X. Zhu et al. 2013). The proposed technique follows three steps to perform a search. First, SH transforms the data into low-dimensional data. Second, it converts data into Hamming-space, i.e. binary encoding low-dimensional data. After that, it generates a binary code to represent data. SH suffers from the existence of high encoding operations especially in complex environments like IoT.

The most efficient indexing technique for range-queries on append-only data is Bitmap indexes. I.e. the conversion of B-tree to Bitmap indexing reduces the query response time and disk space usage if the data is append-only (Wu, Shoshani, and Stockinger 2010). The proposed work developed an accurate closed-form formula to predict the size of the index and the cost of the query processing for compressed Bitmap indexes. Since these formulas are so accurate, they are helpful for query planning and cost estimation. Furthermore, this work found optimal parameters for multi-level indexes. It is based on storing bulk index data in the form of a sequence of bits. The sequences of bits are utilized in bit-wise logical operations to answer queries. Bitmap indexes are classified into two groups: multi-level and multi-component compressed bitmap indexes. The most important challenge that the bitmap index suffers from is the high updating cost. A bit-

Universiti Malaya

sliced index is formed from binary encoding schemes through decomposing a group of identifiers (bin numbers) into multiple components (MacNicol and French 2004). Although it exploits binary encoding to initiate the minimum number of bitmaps, it fails to respond to queries in minimum time. On the opposite side, the Bit-sliced index takes less query processing costs and space.

The mentioned bitmap methods suffer from slowness in large-range query processing, require high storage space, and the flexibility to build the index-structure. Thus, the need to solve these challenges is appearing, to support a compromise between the performance of indexing and the requirements of storage. To fulfill that, (Sinha and Winslett 2007) proposed multi-resolution bitmap indexes that solve the mentioned problems by using multi-resolution and parallelizable bitmap indexes. As a multi-resolution, the proposed work used three levels of resolution that provide worthy performance for queries and large datasets. Unfortunately, the proposed method suffers from high operational overhead because of the behavior of a processor of bitmap indexing. Furthermore, it is time-consuming, because it is not supporting self-tuning indexing.

Another work of bitmap indexing techniques was proposed in (Wei, Dutta, and Shen 2018). The mentioned work proposed two approaches. Firstly, information guided stratified sampling IGStS for short that creates a compact sampled dataset, to maintain the significant characteristics of the data. Secondly, a novel data recoveries approach by using Hungarian algorithm that solves the problem of data recovery by converting it into the optimal assignment problem. The advantages of this technique; it reduces the storage space and the flexibility to meet the requirements of the application. On the opposite side, it is not applicable to mobile environments due to increased index updating operations. Besides, it suffers from high computation overhead. The summary of the mentioned techniques and challenges are represented in Table 2.1, B.

2.2.1.2 Attribute-data tree-based index structure

Attribute-data indexing techniques have intended to adapt to specific requirements; thus they are proper for particular situations. In this subsection, an in-depth investigation of the current attribute-based indexing techniques is presented, to discuss and extract the pros and cons of each one.

R-tree (Antonin Guttman 1983) is effective for multi-dimensions indexing, which is extended from B-tree. It is divided into two parts, non-leaf, and leaf node. The main challenge in the structure of R-tree is the node overflow and underflow (Al-Badarneh, Yaseen, and Hmeidi 2010). Node overflow causes by an insertion, which means inserting more than a maximum number of specifying entries for indexing. This affects negatively the R-tree index because it forces the height of the tree structure to increase. Node underflow causes by deletion operations that cause it to reinsert number of objects. The main disadvantage of R-tree is inefficient for updating operations. Another example of spatial indexing is B-tree index (Bayer and McCreight 1972), which complies with the review of large multidimensional data related to a set of rules and operators, which provide specialized plans for search-related data. It is suitable for dealing with records of different lengths that are commonly observed in large data. However, B-tree faces a high query cost because of the complexity of tree structure and wastage of computing resources, especially in real-time data indexing.

To mitigate the shortcomings of the previous techniques, hybrid indexing techniques are proposed. Furthermore, spatial-temporal indexing techniques have also been proposed. B+-tree proposed to mitigate the challenges in B-tree (Sandu Popa et al. 2011). B+-tree is based on combining the graph partitioning and a set of composite local indexes in B+-tree. The advantage of B+-tree appears in the indexing trajectories data. On the opposite side, B+-tree suffers from an index updating overhead, and it works very well in static environments. In order to improve B+- tree, Bx is proposed (Bassi 2004). The proposed index tried to minimize the updating overhead that has resulted from indexing based on minimum bounding rectangles (MBR). To achieve that, Bx tree is proposed based on the location and time of objects. It considers each object's location as a vector. Bx tree performs the index in two ways; first, it represents the moving objects by a linear function of time, which enables the prediction of the next object moving. Secondly, the partitioning of the index is based on the updating time. The defect of Bx-tree is the way to manage the mobile objects which base on timestamps that represent the transferring time, resulting in not being suitable for all kinds of queries which increases the overhead of updating the process in mobile WSN.

Additional work is proposed to enhance R-tree using Probabilistic Threshold Queries index (PTI-Index) (Cheng et al. 2004). This technique depends on improving one dimensional R-tree. PTI-Index tried to enhance the R-tree index based on the technique called: variance-based clustering, in which the objects having the same level of uncertainty are clustered together. The benefit of applying the enhancements on R-tree, enables it to retrieve nodes whose MBR does not overlap the mentioned child (R-tree problem mentioned before), where each internal node has MBR and corresponding pointers. Also, it increases the updating overhead because of the complexity of pointers. Unfortunately, it is not suitable in WSNat all because of the complexity of pointers, which pops-up from the need to keep the parent's pointers in each node in order to perform updates that start from the leaf node until reaching to the root.

Another tree-based indexing is aCN-RB-Tree (D. W. Lee, Baek, and Bae 2009). It is efficient spatial-temporal indexing of the trajectories, and it provides an efficient search for traffic zone of the time interval. The updating operations of the index are based on dividing the mobile object's data according to location and transferring time. It is useful for diverse service applications in the ubiquitous environment. Since aCN-RB-Tree suffers from inefficiency in the updating process, it is unsuitable for indexing real-time data. In mobile environments, the locations of the moving objects are certain at the time of indexing-update. Meanwhile, uncertain to predict the next location of the sensors. To solve this problem, U-Grid is proposed (Cheng, Kalashnikov, and Prabhakar 2004). U-Grid tried to solve the problem of incorrect results of the query based on the previous uncertain data. This challenge arose because of the need to bring the up-to-date location of the object frequently to keep the information updated. It tried to increase the control of uncertainty, to reduce the error in resulting queries. The key idea of U-Grid tree is to estimate probabilistic of the location of each object by considering the independence of each object to another and execute nearest-neighbor queries. The main deformities of U-Grid tree as follows. The time required to answer the query is not optimal, because it requires high computation overhead to execute probabilities due to the frequently updated location of the objects. Also, it is inefficient to answer continuous queries in the term of accuracy. Furthermore, it is based on probability computations, which results in less accuracy and is unsuitable for complex environments, i.e. mobile WSN because of high computation overhead.

It is an important issue for indexing the up-to-date location of the mobile objects that increases the retrieval efficiency. Unfortunately, the changing of objects' location affects the indexing effectiveness, because it increases the updating overhead. To achieve that, Lazy Group Update (LGU) technique is proposed (B. Lin and Su 2005). LGU consists of two structures: Insertion-buffer (I-buffer) and deletion-table (D-table). I-buffer is a disk-based insertion that enhances the insertion operations at the specified internal node. D-table is a memory-based deletion table that supposes for the entire tree. The main disadvantage, it is used a hash lookup table that increases the computation overhead. Furthermore, using the mentioned buffers is good at minimizing the update operation, but it negatively affects the index building-time. History Time-Parameterized R-tree (HTPR*-tree) is based on the creation or updating time of the moving objects (Teixeira,

Universiti Malaya

Ralf, and Hagen 2005). In HTPR*-tree structure, the creation and updating time is located on leaf-node. To update the index, it uses R-tree update operations but applies the bottom-up approach to improve updating efficiency.

Time-parameterized tree (TP-tree) is an extension of R-tree, which aims to answer the queries by representing it as query-window of the mobile objects (Tao and Papadias 2002). As in R-tree, TP-tree constructed indexing structure according to MBR that is based on the object's attributes. Frequent Update R-tree (FUR-tree) (M. L. Lee et al. 2003) is based on T-tree (Lehman and Carey 1986). Since R-tree is the choice to index multi-dimensional data with low dimensionality, it suffers from updating operations. This is because of the up-down updating strategy. So, FUR-Tree supposes to enhance R-tree by presenting a bottom-up updating strategy, to solve the problem of frequent updates. The bottom-up update strategy is mainly based on hash-table that points directly to the leaf-node, to accelerate the updating time. Moving Objects on Networks MON-tree is based on R-tree that efficiently stores and retrieves the current location of the mobile objects (de Almeida and Güting 2005). MON-tree structure assumes that the objects are moving along polylines that connects with two models, edges and route. Since R-tree suffers from updating overhead, R-Tree Updates Memos (RUM-Tree) (Xiong and Aref 2006) is proposed. The basic idea of RUM-tree is avoiding accessing old entries during the updating process. The main advantage of this technique is reducing the cost of insertion. It supports location-based frequent updates and range-based queries.

MSMON resulted from adjusting MON-tree to enable it to deal with mobile objects (Z. Zhu, Yang, and Pi 2012). This tree is based on two main models, edge-based and the route-based model. This technique aims to minimize the number of indexed data and to index the proposed predicted positions. In more detail, it is a two-tier structure and consists of multi-grids R*-tree that indexes the paths of the mobile objects. U-tree is based on index mobile objects by trimming sub-trees that do not contain any results and insert the required details in the leaf nodes (Tao et al. 2005). The updating process is based on R*-tree strategy. The main disadvantage is I/O overhead, and it is unsuitable for a huge amount of data. Grid Partition R-tree (GPR-tree) indexing tree is based on indexing paths of mobile objects. It divides the area into grids of di_erent sizes and performs indexing based on those grids (Huang, Hu, and Xia 2010). The main advantage is improved updating performance.

Self-Tunable Spatio-Temporal B+- Tree (ST2B-Tree) aims to solve the challenge of mobile objects database (MOD) in which the location changes through both space and time, and answering the query may also be changed through time (S. Chen et al. 2008). It is a self-updating B+-tree for mobile objects, which is based on partitioning the area into grids. The main characteristic of this technique is the updating process which is conducted on-line without any user interventions. Unfortunately, in the case of mobile sensors, the scalability challenge appears especially for updating the tree structure. Furthermore, it is based on a clustering technique that makes it unsuitable for mobile environments, especially random mobility. BBx (D. Lin et al. 2005) is the representation of the location and time of a mobile object by a linear function to find the past, current, and future queries. It is based on the B+ technique to perform an index by storing the linear function location of the mobile objects, in which support queries are based on spatial and temporal constraints. Note that BBx is the inherent way of finding current and future locations on Bx-tree.

Past, Present, and Future Index (PPFI) technique (Ying et al. 2008), indexes the Past, Present, and future values of moving objects, with the aim to improve on the update mechanism. PPFI uses the hash-table to improve the updating mechanism in addition to 2DR*-tree and 1DR*-tree. In this strategy, PPFN*-tree (Ying Fang et al. 2013) index mobile objects are grouped into two sets, past information, and future data prediction. This is to ensure that it stores past paths, present and future paths of the mobile objects.

In other words, it is a hybrid structure that contains managing road indexing and predicts the future position of the current sensors. To improve update operations, PPFN* uses hash table.

There is a technique that indexes the past, current, and future information of mobile objects. This technique is named Past-Current-Future+-Index (PCFI+-Index) (Liu and Liu 2005) that enhances the PCFI-index, which is based on SETI-tree and TPR*-tree. Dealing with mobility has partitioned the regions into grids which are used to index those grids based on a spatial access method. The main advantage of PCFI+-index is the ability to handle the query process efficiently. This indexing technique was proposed to solve the mobile objects by overcoming the problem of slower I/O speeds. With this enhancement, RppF (Jidong Chen 2007) became a good technique for big data. Additionally, RppF-tree technique proposed the capability of indexing online location (all objects position in time) from the mobile users. It has stored all the positions as a linear function to increase the I/Ospeed.

So far, tree-based indexing techniques that build the structure of the tree according to attributes have been intensely discussed. This investigation aims to extract a conclusion of how the mentioned techniques tried to support mobility according to benefits. Furthermore, this investigation enables extracting the remaining challenges in mentioned techniques in terms of mobility perspectives, as summarized in Table 2.1, A.

2.2.2 Non-tree based Indexing

To minimize query retrieval time, it is important to implement an index to analyze and retrieve results over heterogeneous data (Siddiqa et al. 2017). Hence, to implement the appropriate index in Hadoop, it is necessary to know the scheme and anticipated Map-Reduce jobs. Trojan index fulfills this condition (Dittrich et al. 2010). Hadoop is a standard for big-data processing. It does not provide index access when lacking priority knowledge of the scheme and Map-Reduce jobs being executed. In contrast, DBMS

Universiti Malaya

requires users to specify the scheme, where indices may be added in demand. Trojan index approach clustered a static index by offering a single attribute to be indexed. In contrast, it offers a different number of attributes called HAIL (Dittrich and Quiané-Ruiz 2012). One attribute is indexed and stored on all duplicates. Indexes are initiated at the time of data uploading. However, the selection of an attribute to be indexed has to select carefully, because it can not be updated later. Actually, one particular index is not sufficient, this considers the first drawbacks of the proposed index. Furthermore, indexing the initial costs is higher than running a full scan query. The initiated index may be unused, and it increases index overhead.

Richter et al. proposed Lazy Indexing (LIAH) to minimize I/O cost of indexing (Richter et al. 2014). To achieve that, LIAH uses offer-rate and initiates many indexes as suggested according to incoming queries. However, the disadvantage of LIAH is the ability to minimize the index building overhead in the case of low offer-rate. More explicitly, there is a trade-off between index building overhead and Map-Reduce jobs. With a view to minimizing index overhead, the value of the offer rate is set to low. The low offer rate increases more Map-Reduce jobs. Because there is no replication factor in LIAH the Adaptive indexing-replace indexing is proposed (Schuh and Dittrich 2015). The replication factor removes the unused index by considering the replicated data for each new index attribute. Data blocks are still replicated for new indexes and consumed disk-space.

In indexing development, many indexing techniques have been proposed for improving Multi-Dimensional Range Query (MDRQ) performance. Nevertheless, the existing indexing techniques are not effectively achieving the required performance of the insertion and flexible MDRQ synchronously. To solve this issue, a novel indexing technique named LCIndex is proposed (Feng et al. 2015). LCIndex is standing for Local and Clustering Index. It is based on global-index and local index, where the global index has high network traffic, but it is easy to implement, while the local index has low network traffic but is difficult to implement. Accordingly, LCIndex does the test of performance on all those indices. It minimizes the network traffic during insertion operation and querying. Furthermore, it supports dynamic indexing operations (insert, update, and delete). Also, LCIndex suffers from high storage-cost.

To alleviate the challenges of inefficient implementations of range queries over existing Distributed Ordered Tables (DOTs), and highly index updating overhead over the index. To solve the mentioned challenges, (Gugnani et al. 2018) characterized the performance of indexing on DOTs from the network perspectives. Then, the RDMA-based high-performance communication framework is proposed. The proposed work designs a parallel insert operation to reduce index creation network performance characteristics of index techniques on overhead.

Another work is proposed in (Cao et al. 2017) that tried to solve the challenge of the ine_ciency of multidimensional queries. The proposed work is called CFIDM short for Column Family Indexed Data Model. It is a novel data-model, partitions the values of the queried column. Each partition is established by column family, by adapting column family into index with no additional cost. After that, (Cao et al. 2017) provided instructions to build CFIDM as a data model. One more work is DISTIL (Patrou et al. 2018), which is a distributed spatial-temporal data processing system, which tried to address high-velocity location data. Distributed in-memory index and storage infrastructure built on a distributed in-memory programming paradigm called APGAS (Asynchronous Partitioned Global Address Space). The location records are distributed across a cluster of nodes, using the producer-consumer model.

As a result, non-tree based indexing techniques are unsuitable for indexing mobile environments at all, as summarized in Table 2.1, 2. There are many reasons. First, it bases on clustering the data based on specific circumstances that increase the scalability

Universiti Malaya

challenge. Second, the high I/O costs for indexing. These challenges are exacerbated in the case of random mobile environments. Thereby, this work does not consider the comparisons of the non-tree based indexing techniques due to the reasons addressed earlier.

In sum, Table 2.1 summarized the current indexing techniques and how these techniques have been improved to support mobility. Furthermore, the challenges of indexing techniques are extracted and summarized according to the mobility environment's perspectives.

Table 2.1: Summary of index techniques and the challenges from the mobility perspectives

#	Index Technique	Properties	How to support Mobility	Challenges
1. Tree-Based Indexing				
A. Attribute data				
1	R-tree	- Effective for multidimensions indexing. - Its structure is hierarchical, dynamic, and height-balanced.	- It supports different types of spatial-queries. - It represents mobile data objects by intervals in several dimensions. - It performs indexing according to MBR.	- Node overflow and underflow. - It uses the up-down update strategy. - The index structure takes more space. - Query response depends upon buffer size.
2	B-tree	It complies with the review of large multidimensional data related to a set of rules and operators, which provide specialized plans for search-related data.	It is suitable for dealing with records of different lengths that are commonly observed in large data.	- The complexity of the tree-structure. - High query cost.
3	B+-tree	- It is based on combining graph partitioning and a set of composite	- Its ability to index trajectories data. - Its flexibility in adjusting the index structure for better	- It suffers from index updating overhead. - It requires addressing queries

		local indexes in B+-tree.	performance of historical-data and in a dynamic context. - It provides good query and update performance, compared with R-tree.	about the current and future location of mobile objects. -It requires improving continuous spatial-temporal queries for constrained mobile sensors. - It is not suitable for mobile environments at all.
4	Bx-tree	- It is based on B+ tree. - It performs the index in two ways: o It represents the moving objects by a linear function of time. o It partitions the index according to updating time.	- It tries to minimize the updating operations. - As in R-tree.	- High dependability on attributes of the objects to index. - Inherent challenges of R-tree. - It is not suitable for all kinds of queries. - The overhead of updating process in mobile WSN.
5	PTI-Index	- It improves one dimensional R-tree and its structure able to answer the queries for different types of uncertainty data, by variance-based clustering.	It enhances R-tree by enabling to retrieve nodes whose MBR does not overlap the mentioned child.	It increases the updating overhead because of the complexity of pointers.
6	aCN-RB-Tree	- It is spatial-temporal indexing of the trajectories. - It provides an efficient search for the traffic zone of the time interval.	Its updating operation is based on dividing moving object data by area and time.	The applicability of WSN.
7	U-Grid	It supports probabilistic of the location of each object that considers independent from others.	It brings an up-to-date location of the objects frequently.	- It bases on probability which results in less accuracy. - It is unsuitable for WSN environment

Universiti Malaya

				because of computation overhead.
8	LGU-tree	- It indexes an up to data location of the mobile objects. - It uses hash-table.	It uses an Insertion buffer (I-buffer) and deletion table (D-table).	-It uses a hash lookup table that increases the computation overhead. -Using the buffers is good in minimizing the update operation but it negatively affects the index building time.
9	History Time-Parameterized R-tree (HTPR*-tree)	- It constructs an index structure based on the creation or updating time of the moving objects. - Creation and updating time in the constructed tree located on leaf-node. - It applies the bottom-up approach to improve updating efficiency.	By minimizing updating overhead.	Still, suffering from gaps that inherited from R-tree.
10	TP-tree	- It is an extension of R-tree. - It aims to answer the query, which represented query-windows" of the mobile objects.	- It performs an index by constructing MBR based on the object's velocity and the current time. - It uses the nearest neighbor search for queries.	It is based on the MBR.
11	FUR-Tree	- It stands on T-tree. - It presenting a bottom-up-update strategy. - It solves the problem of	Same as R-tree (the problem of R-tree is inefficient updating operations).	It does not support dynamic updates, because it inherits the high computation overhead from Hash-table.

Universiti Malaya

		frequent updates that R-tree suffers from.		
12	MON-tree	- It is based on R-tree that has proposed to efficiently store and retrieve the current location of the mobile objects. - Its structure assumes that the objects move along polylines that connects with two models, edges and route.	Same as R-tree	Same as R-tree
13	RUM-Tree	It enhances R-tree to support updating operations.	It handles sensors' updates efficiently.	Its delete operation is not suitable in a complex environment.
14	MSMON	- Resulted from adjusting FNR and MON tree. - It aims to minimize the number of indexed data. - It indexes the predicted position.	By indexing the trajectories of mobile objects.	It is not suitable for WSN because it suffers from low accuracy.
15	U-tree	It bases on R*-tree strategy.	Trimming subtrees that do not contain any results and insert in leaf nodes the required details.	- I/O overhead. - It is not suitable for a huge amount of data.
16	Grid Partition R-tree (GPR-tree)	- It is based on indexing the paths of mobile objects. - It is partitioning the network area into different sizes of grids.	It is improving updating performance.	It is not suitable for mobile environments, especially in the case of random mobile sensors.
17	ST2B-Tree	It aims to solve the challenge of mobile objects by self-	It conducts the updating process	- In the case of mobile sensors, the scalability

		updating B+-tree for mobile objects, which based on partitioning the area into grids.	online without user interventions.	challenge appears especially for updating the tree structure. - It is based on a clustering technique that makes it unsuitable for mobile environments, especially random mobility.
18	BBx	It is based on B+-tree to index by storing the linear function location of the mobile objects, which support queries based on spatial and temporal constraints. It is inherent way of finding current and future locations from Bx-tree.	It stands on representing the object's location and time by a linear function to find the past, current, and future queries.	Same as B-tree.
19	PPFI	- It indexes and retrieves the Past, Present, and future information about moving objects. - It aims to improve the update mechanism.	It uses hash-table to improve the updating mechanism, in addition to 2DR*-tree and 1DR*-tree.	It suffers from computation overhead, which is inherited from Hash-table indexing.
20	PPFN*	- It indexes past and future data. - It performs updating using Hash-table.	It performs indexing according to the object's paths and considers each path as a node.	- Inherited computation overhead from the hash table. - Accuracy problem.
21	PCFI+-Index	It indexes the past, current, and future mobile object's information.	It partitions the area into grids, used to index those grids based on a spatial access method.	It introduces parametric bounding rectangles in R-tree.

Universiti Malaya

22	RPPF-tree	- It bases on R-tree, but enhances MBR. - It overcomes the problem of slower I/O speeds. - It has the capability of index online locations.	It indexes mobile objects based on all locations and periods of time.	Can not track the different characteristics of the sensors, such as temperature, humidity, etc.
B. Transform data				
1	Sparse Hashing (SH)	It follows three steps to perform a search: 1.Transforms data to low-dimensional data. 2. Converts data to Hamming space. 3. It generates a binary code to represent data.	SH builds the tree structure based on converting the indexing data.	The scalability challenge.
2	Bitmap indexes	Developed an accurate closed-form formula to predict the size of the index and the cost of the query processing.	It reduces query response time and space costs.	-Its features are valuable just if the data is append-only. -It suffers from highly updating operations processes.
3	Bit-sliced index	Presents a multicomponent Bitmap index constructed from three basic encoding schemes.	-It takes less space. -Less query-processing cost.	It takes a long time to respond to the query.
4	Multi-resolution and bitmap indexes	Used three levels of a resolution that provides worthy performance for queries and large datasets.	Provide worthy performance for queries and large datasets.	-The proposed method suffers from high operational overhead because of the behavior of a processor of bitmap indexing. -It is time-consuming because it is not supporting self-tuning indexing.

Universiti Malaya

5	Data Sampling and Recovery using Bitmap Indexing	Proposed two approaches: - Information guided stratified sampling, IGStS for short creates a compact sampled dataset, to maintain the significant characteristics of the data. - Novel data recoveries approach by using Hungarian algorithm that solves the problem of data recovery by converting it into the optimal assignment problem.	-It reduces storage space. -Flexibility to meet the application's requirements.	-It is not applicable to mobile environments because of increasing the index updating operations. -It suffers from high computation overhead.
2. Non-tree based indexing				
1	Trojan index	Clustered static index which offers an index on the single attribute to be indexed.	Unable to support mobility	It is unsuitable for mobility at all, reasons: 1. One particular index is not sufficient. 2. Indexing the initial costs is higher than running a full scan query. 3. The initiated index may be unused, and it increases index overhead.
2	Lazy Indexing (LIAH)	It uses the offer rate and initiates many indexes as suggested by incoming queries.	Unable to support mobility	- The ability to minimize the index building overhead in case the offer rate is low. - There is a trade-off between index building overhead and Map-Reduce

Universiti Malaya

				jobs. With a view minimizing index overhead, the value of the offer rate sets to low. The low offer rate increases more Map-Reduce jobs.
3	Adaptive indexing-replace indexing	The replication factor removes the unused index by considering the replicated data for each new index attribute.	Unable to support mobility	Data blocks are still replicated for new indexes and consume disk space.

2.3 Classification of mobile sensor techniques in WSN

This section aims to analyze the current approaches applied to mobile environments to extract the deformities that affect indexing performance. Accordingly, understanding and highlighting the nature of mobile objects concludes the major reasons that adversely affect the indexing process in such an environment. Furthermore, this section is finding the relation between the sensor's mobility behavior and the performance of indexing. To simplify that, the investigation is conducted over three approaches: meta-heuristic approaches, Grid-based indexing approaches, and prediction model approaches, as represented in Figure 2.3.

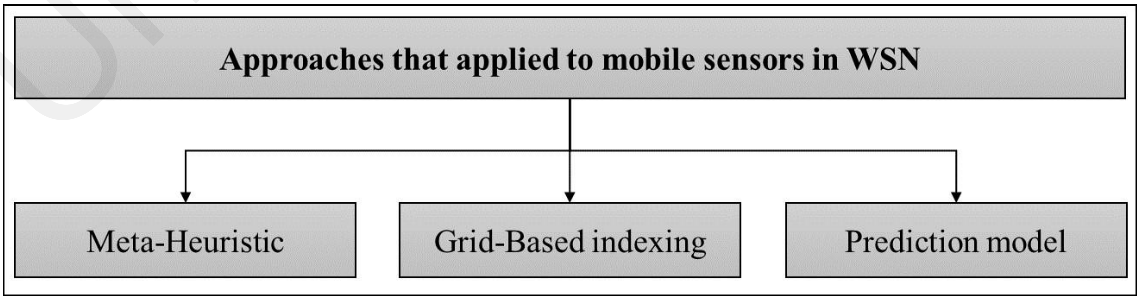

Figure 2.3: Classification of approaches that applied to mobile sensors in WSN

2.3.1 Meta-heuristic Approaches

A meta-heuristic is a group of algorithmic ideas, which utilized to define heuristic techniques. A group of variables that are bounded by a group of constraints is defined as the optimization problem. Meta-heuristic is appropriate for a wide group of various obstacles. I.e. it can be considered as a general-purpose heuristic strategy toward encouraging areas of the search space. Meta-heuristic have been proposed to solve optimization problems. Ant colony optimization (Marco Dorigo, Mauro Birattari 2006), Tabu-search method (Glover 1990), simulated annealing method (Kirkpatrick, Gelatt, and Vecchi 1983), etc. are examples of meta-heuristics techniques. Meta-heuristic approaches are applied to control mobile nodes, i.e. clustering sensors (Ebrahimi et al. 2016), or for sensors coverage (Loscrí, Natalizio, and Guerriero 2012).

M. Ebrahimi et al. tried to reduce the search process of the sensors by proposing an adaptive strategy called Sensor Semantic Overlay Networks (SSONs) (Ebrahimi et al. 2016). SSON is a method aimed to cluster mobile sensors according to the context of information similarities. Such as weather, temperature, etc. More explicitly, sensors with the same information context are embedded into the same cluster called SSON. After SSON creates, the meta-heuristic algorithm called AntClust. AntClust is clustering sensors based on the similarity of information context. The proposed method is unsuitable for the mobile nodes because it did not take into account the source of the transmitted location. Indeed, the mentioned challenge is exacerbated in the case of random mobility. However, the proposed work is unsuitable for mobile environments at all because it is based on classification. Furthermore, Ant colony algorithm suffers in mobile environments, because it has the following drawbacks. (i) It is hard to develop an optimization problem. (ii) It suffers from slow convergence speed. (iii) In the case of a high dimensional optimization problem, it is hard to achieve the ideal time and memory-space.

In mobile environments, it is essential for allocating the locations of the moving sensors. To fulfill that, Ni et al. used Multi-Objective Particle Swarm Optimization (MOPSO) to move the sensors to the proper location (Ni et al. 2015). Furthermore, the mentioned work tries to find the optimal location for moving sensors to improve Quality of Service (QoS). To achieve that, they considered two factors: mobile nodes, and QoS. Mobile nodes consider as an optimal solution, which defines computing distance between them using discrete PSO. Using multi-swarm PSO they achieved QoS. The proposed method improves the QoS of the sensors by providing high coverage of appropriate sensor locations. It also supports the sensor's energy efficiency. However, it did not take into account the status and characteristics of data. The sensors' data is periodically transferring, which considers as big-data. The resulted big-data will cause difficulties by exacerbating the scalability challenge. Also, each mobile sensor is represented by a set of locations; this increases the complexity of dealing with the status of reported data. In addition to that, the consistency between nodes and the communication design topology leads to low connectivity among them.

Rao et al. proposed clustering using particle swarm optimization (PSO) to enable electing the ideal cluster-head by using a head-selection algorithm called PSO-ECHS (Rao, Jana, and Banka 2016). The main aim of this method is to support the efficiency of the sensor's energy by taking into account the sink distance, intra-cluster distance, and residual energy as parameters. The summation of the sensor's energy consumed is minimized, which is considered the first advantage of using the proposed algorithms. However, this will eventually break-down, when the network size and the loop's number are increased. In terms of network lifetime, the algorithm is working well with the remaining energy in the sensors. As a result, the received packet's number depends on energy consumption and the sensor's lifetime. PSO-ECHS does not take into account the routing issues. Also, it proposed to reduce energy consumption, but it failed to balance

Universiti Malaya

the energy among the sensors. Balancing means that when a sensor has low energy it will finish. Also, it is not suitable for homogeneous networks, which will affect fault tolerance. PSO-ECHS algorithm is not suitable in the case of mobility. For example, the sensor's lifetime is decreased when the location of base stations is changed, especially when the base stations move to the corner of the specified area. Another work aimed to elect the optimal cluster head proposed in (Elhabyan and Yagoub 2015). It tries to solve the problem of the cluster-heads ability to send to the base station. To achieve that, they proposed a two-tier routing algorithm that finds the optimal route. It connects the cluster-head with the base-station, in order to maximize energy efficiency, cluster quality, and network coverage. In the proposed work, the classification of packets' without taking into account the packets' arrangement negatively affects the performance of the index.

Another technique applied to mobile sensors has proposed by Kim et al. (Yang G. Kim and Lee 2014). They adopted two meta-heuristic algorithms, Simulated Annealing (SA) and Particle Swarm Optimization (PSO). They try tackling the ideal channel scheduling by periodically scheduling active timing that called slots for all sensors in the network. The proposed scheduling has given better results in terms of end-to-end delay compared with SA-based scheduling. The most important strength of this work is the existence of a clear routing path that is fulfilled from time slot scheduling. On the other hand, the proposed work has some deformities. It is not suitable when the number of sensors is increasing, because this increases computation complexity. Furthermore, the capability of searching for local sensors is not effective. Those deformities exacerbate in the case of random mobility.

There are some works that make improvements over (Yang G. Kim and Lee 2014). Kim et al. exploited the explicit routing path discussed in (Yang G. Kim and Lee 2014) but determined the time slot scheduling based on a routing path not based on nodes (Y.G. Kim et al. 2015)(Yang G Kim et al. 2016). This improvement increases the performance

Universiti Malaya

of end-to-end delay. Also, Lee et al. extended (Yang G. Kim and Lee 2014) by exploiting unoccupied time-slots to establish substitution paths (Junhee Lee, Jeong, and Choi 2016). This way enables increasing the number of time slots and exploits all of them.

Khoshkangini et al. try to solve the problem of selecting the best path in WSN (Khoshkangini and Zaboli 2014). To achieve that, Ant-Colony algorithm was established. The main aim is to save the sensor's energy throughout the multi-constrained QoS technique (IAMQER). The benefits of this algorithm are increasing the Packet Delivery Ratio and support the prolonging of sensor lifetime. To test how the proposed technique performs, it compared with NQoS AODV (Ad-Hoc On-Demand Distance Vector), which is considered as the shortest path algorithm. The outcomes showed that this technique can enhance network throughput in the case of end-to-end delay and packet loss ratio. On the opposite side, solving optimization problems using an ant colony algorithm is not a perfect solution. It is hard to develop the optimization problem, suffers from slow convergence speed, and in the case of a high dimensional optimization problem, it is hard to accept the supreme time and memory space. Also, energy consumption is unsteady, when number of rounds are changing, which affects quality performance.

Another work tried to control the sensor's movement using particle swarm optimization (PSO) to improve network coverage. V. Loscrí et al. proposed a consensus algorithm that updated the sensor's speed (Loscrí, Natalizio, and Guerriero 2012). The consensus algorithm has two versions: global and local versions. The global version used the sensor's information of all sensors in the network. A local version based on the information of the neighborhood information. To improve the proposed algorithms, the authors add the pioneer's concepts, which is responsible to detect the interesting areas before activated the sensors. The drawback of the proposed work is the need highly location update of the moving sensors, especially if the movement of the nodes is random.

Universiti Malaya

Efficient IoT applications must have the ability to manage and utilize a large amount of produced data. This is because IoT applications need widely accepted software technology that works with heterogeneous systems and high-performance software platforms (Psannis, Xinogalos, and Sifaleras 2014). To achieve these IoT needs, TSOIA algorithm has proposed (Luo, Lu, and Cheng 2013) as a new optimization algorithm for sensors in IoT. This algorithm has intended to support applications that do not require separating the solution space for finding optimal solutions. Accordingly, TSOIA group nodes into three sets, and then perform a random search, local search, and orientation search to check set the size and the step length. The results obtained from simulations showed that the proposed algorithm outperforms the existing Genetic Algorithm (GA) and FLAGA in terms of searchability, energy consumption, and network delay. However, the routes of the solution are not always optimal.

Resource management and allocation consider crucial problems in heterogeneous network environments. The authors Li et al. proposed a platform called SACHSEN, that manages executable applications and scheduling tasks by allocating them to the appropriate sensor (W. Li et al. 2014). The idea of SACHSEN is to prolong the network lifetime by exploiting sensors that consume the least amount of energy among all current sensors to perform the task. The key factor that affects the performance of the proposed system is the physical system properties. The physical system may have different properties, such as continuous or discrete resources, reusable or non-reusable resources, etc. All these properties must be considered. Unfortunately, the proposed work did not take into consideration the differences among systems. As a result, it is not suitable to be implemented in random mobile IoT nodes, because the possibility of packet loss in mobile environments is high, which negatively affects the performance of SACHSEN.

Saleem et al. used the swarm method to find an on-demand multi-available path without the need for and geographical information (Saleem, Ullah, and Farooq 2012). The

Universiti Malaya

proposed protocol has inspired by the principle of a bee colony and how the bee is looking for food. The experimental results showed that Bee Sensor performs better than the other Swarm Intelligence-based routing protocols in terms of the Packet Delivery Ratio, latency, and energy efficiency, but it is unable to solve the route coupling problem because bee-colony algorithm considers as agent-to-agent communication to discover optimal paths. In addition, the scalability challenge is increased because the amount of transferred data increased.

2.3.2 Grid-based Indexing approaches

Object movement is a continuous process that operates on both spatial and temporal domains. Indeed, the best representation of moving objects is by considering the transferring location and time. Based on that, this approach performs indexing of moving data, in accordance with the division of the network area into grids. In this subsection, the Grid-based techniques are categorized as spatial-temporal, regular-decomposition, object-directed decomposition, and density-based clustering.

2.3.2.1 Regular Decomposition techniques

Regular decomposition is also known as a space-driven index. Space-driven index means partitioning a network area in a regular or semiregular manner. It is indirectly related to the objects in the network area. Objects will be addressed in the new structure by mapping them to cells, in accordance with geometric criteria. An example of a mapping method in this approach is space-filling curves (OOSTEROM 1999). A space-filling curve is a mean of converting specific points in certain dimensional spaces into a set of numbers. It is also called tile indexing because it stands for converting a two-dimensional space into a one-dimensional space. Many ordering methods, such as row ordering or prime, column and column prime ordering, Morton key (also known as Peano key), N or Z-order, Hilbert ordering, Gray ordering, Cantor diagonal ordering, spiral ordering, and Sierpinski curve, are used in this approach (Abdul, Begum, and Supreethi

2018). The Hilbert ordering method is a space-filling curve (Hilbert 1891), which is suitable in any dimensional space. The Hilbert ordering arranges specific points of a high-dimensional space with integer coordinates along the Hilbert curve using the classical Hilbert–Peano curve. The Hilbert method plots the Hilbert curve to separate all points connects integers called Hilbert indices that represent requests along the Hilbert curve with points, and sorts pairs of points and indices. Nevertheless, the Hilbert curve suffers from sorting selected objects in a high-dimensional space with the formatting of integer values along the Hilbert curves.

Another function of space-driven indexing is double transformation (Bozkaya and Ozsoyoglu 1997). It performs an index by applying a double transformation. The first transformation divides a network area into k-dimensional space. The divided areas are then mapped into one-dimensional space using Hilbert or Peano mapping. Quadtree (Yianilos 1999) is based on dividing an area into squares, and each node in the tree structure corresponds to a square. Each internal node in the quadtree is divided into four children.

2.3.2.2 Object-Oriented Decomposition techniques

Object-oriented decomposition is also known as a data-driven index. This approach divides a network space based on object coordinates. R-tree (Antonin Guttman 1983) and R*-tree (Beckmann et al. 1990) are examples of this indexing type. R-tree (Antonin Guttman 1983) is extended from B-tree, which is a multidimensional data index. It is based on partitioning a network area into grids called MBRs. Each MBR is considered a node structure of index-tree. B-tree (Bayer and McCreight 1972) is a one-dimensional index that aims to index static features. B-tree complies with the review of large multidimensional data related to a set of rules and operators, which provides specialized plans for search-related data. It is suitable for dealing with records of different lengths that are commonly observed in large data. However, B-tree faces a high query cost because of the tree structure complexity. It also suffers from the wastage of computing resources, especially in real-time data indexing.

Another work is proposed, called a Quadrant-based Minimum Bounding Rectangle tree (QbMBR) (Jo and Jung 2018). QbMBR is an index that has a hierarchical structure dedicated to the processing of spatial-query in HBase. It stands for grouping mobile objects according to spatial attributes using QbMBR. Furthermore, it is based on two query processing algorithms, one for range query and the other for kNN query. QbMBR increases the accuracy of grouping increases the accuracy of query processing and reduces the index storage cost of the index structure.

2.3.2.3 Density-Based Clustering techniques

For the density-based clustering, its most famous example is DBSCAN. DBSCAN is designed to discover clusters of arbitrary shape while effectively handling noise or outliers. The main idea of DBSCAN is that, for each object in a cluster, the neighborhood of a given radius must have at least a minimal number of objects.

The traditional DBSCAN suffers from high I/O costs and main memory requirements because it executes clustering directly on databases. This problem is exacerbated in large databases. Sampling-based DBSCAN algorithms are proposed to mitigate the bottleneck of main memory and I/O costs and improve the time-consuming object neighbor query operation (A. Zhou et al. 2000). Such methods expand clusters by representing a few numbers of objects as seeds, instead of all neighbor objects. However, they still suffer from a representative selection problem that adversely affects the accuracy and performance of DBSCAN.

An additional solution for mobile objects is GDCF (Boonchoo et al. 2019). It depends on partitioning data layout to minimize neighbor explosion and merge redundancies. It utilizes HGB structure to index grids that have objects and neglect empty ones. Furthermore, it provides neighbor grid merging by using the union-find algorithm and

Universiti Malaya

cluster forest. This characteristic derives the benefits of keeping cluster information and preventing unnecessary merging computations. Nonetheless, GDCF is insufficient in random mobile environments for the following reasons. First, it suffers from high complexity overhead because it uses bitmap indexing. Second, it suffers from the high updating frequency in the cluster forest.

Another sampling-based DBSCAN method proposed is GF-DBSCAN (Altti Ilari Maarala, Xiang Su 2016). GF-DBSCAN is based on a well-known approach named FDBSCAN. It reduces the number of searches and redefines cluster cohesion merging. Further work is the Grid-based DBSCAN algorithm (Zhao, Zhang, and Shen 2004), which clusters high-dimensional data with low-order neighbors. It partitions the space layout into intervals and hyper-rectangular grids. However, the algorithm will only compute the density of a given object with the objects in its neighboring grids. Several objects can thus be omitted.

2.3.2.4 Spatial-temporal techniques

As mentioned in Subsection 2.2.1, spatial-temporal indexing is an indexing technique dedicated to index mobile objects according to transferring time and location attributes. An example of spatial-temporal indexing is D-tree (Chen, N. 2015). It has proposed for indexing heterogeneous big-data and it is dedicated to index multi-dimensional movement objects. This ability is because it has a high space-cost in its structure. Furthermore, it answers a range of queries efficiently. Indeed, the hierarchical structure of D-tree is applicable for big-data. D-tree technique is based on making the balance between B and R-tree, which minimizes the number of access to the memory because the index structure is without inner-nodes. On the opposite side, it never considers whether the same sensor is moving from one location to another in case of random mobility.

One more proposed work is an energy-efficient index-tree (EGF-tree). It tries to save the energy utilized in collecting, querying, and aggregating data of sensors located in

Universiti Malaya

multiple regions in IoT (Zhangbing Zhou et al. 2014). Building the proposed index structure is based on the division of the network area into grids, in which each grid is considered as a parent of the tree structure. As well, all sensors that belong to each grid are considered as a child of it. The proposed method is based on grids. The grid's division is not optimal in the case of random mobile sensors.

To support real-time retrieval in mobile IoT environments "IoT-SVK Search Engine" is proposed (Ding et al. 2012) (León, Hernández-Serrano, and Soriano 2015). Real-time retrieval in mobile environments requires dealing with and managing a large number of heterogeneous sensors that frequently change locations and transfer packets periodically. The proposed framework has the ability to deal with all types of data in IoT according to spatial-temporal, value-based, and keyword-based conditions. Further explanation, through the distributed global indices built on the extracted full-text keywords, the spatial-temporal attributes, and the sampling values, IoT-SVK Search Engine can support multi-modal search conditions including keyword-based, spatial-temporal, and value-based constraints. Besides, through the grid cell based index updating mechanism, the indices can be updated in real-time with greatly reduced index updating frequencies so that not only the historical but also the latest states of sensors can be retrieved in real-time. The proposed architecture is proper in heterogeneous IoT environment to improve real-time retrieval. Unfortunately, "IoT-SVK Search Engine" is unsuitable for all kinds of searches, i.e. clue-based and event-based searches.

Another work was proposed for the IoT domain by Du et al. (Du et al. 2013). It was proposed to organize sensing devices by indexing them. The proposed multiple indexing is to cluster IoT sensor nodes based on nodes' functionality and also according to spatial and temporal sensor nodes. Also, Du et al.enabled the dealing with a large amount of IoT data, and enable supporting the multi-dimensional access in cloud-based databases, which proposed an update and query efficient index framework (UQE-Index) based on a key-

Universiti Malaya

value store that enables to support high throughput and provide a multi-dimensional query (Ma et al. 2012). The key advantage of the proposed approach, it performs well in the real-time monitoring system. Unfortunately, it is not applicable in reality.

Park et al. decided on the optimal solution for grid structure and handling the highly skewed distribution of mobile objects (Park, Liu, and Yoo 2013). Also, Park et al. developed a query algorithm that can utilize mentioned (OCG) cells to accelerate the query processing. To perform that, Park et al. proposed the concept of OCG that design the index based on the grid. The proposed method is not suitable if the sensors are mobile because it did not take into account the probability of the sensor moving outside of the grid area classification problem, as mentioned.

Lin et al. proposed a scheme that supports moving objects in multi-dimensional space (H. Y. Lin 2012). Compressed B+-tree based indexing scheme is to perform spatial and temporal index based on the attributes of trajectories, which were preserved and organized into a compact index structure. The benefits of this approach are minimizing the index updating overhead and the query results become more accurate. The first defect of this approach did not take into account the time as an essential member when building the index. The second defects are that the approaches have to identify critical locations before starting to build the index.

Tang et al. proposed a method to monitor the region of interest (Tang et al. 2014). The proposed algorithm based on dividing the region into small areas is called grids, which performs the clustering index tree to index these grid cells. For answering queries, Tang et al. developed a time-correlated region query technique. The defects of the proposed solution that is not applicable for IoT applications like environmental monitoring and the suitability for random mobile sensors.

In real applications, sensors that exist in the same location possibly have a similar reading. This happens in the case of a group of mobility sensors moving to the same area since they deployed high density. In addition to the spatial relationship, the sensor reading also has a temporal relationship in which the data reading collects within the same period of time. Both the spatial and temporal sensor reading is causing overhead and uncertainty because of the redundancy of the readings. To alleviate that, Li et al. proposed Kd-tree to detect the nearest neighbor missing data (Y. Li and Parker 2012). To fulfill that, they considered weighted variances and weighted Euclidean distances obtained from measured percentages of missing data. Then the missing data will use in NNs to impute the lost value. The main advantage is that it achieved a high level of accuracy with minimum computation and faster search time, and define the strong relationship among sensor nodes, which give accurate spatial and temporal data readings. On the other hand, it does not give accurate results in the case of the distribution of data.

Top-k Frequent Spatiotemporal Terms (kFST) is proposed (Ahmed et al. 2017). It is an index structure dedicated to answering top-k spatial-temporal range queries. kFST-index structure extracted from R-tree uses top-k sorted term lists (STLs), to achieve faster execution and smaller space requirements. Unfortunately, the proposed index is cannot settle high throughput streaming data and a large volume of data.

In order to proficiently search real-time sensors, Zhang et al. proposed a technique to upgrade the performance of search by estimating newer sensors and classifying the predicted sensors (Zhang et al. 2015). To accomplish that, a model was offered to search sensors by performing an architectural high-productivity content-based sensor search system and then applying estimation prediction. The defects of this work include a lack of applicability and accuracy.

Wang et al. proposed sensors retrieved strategy based on retrieving nodes whose location range intersects the location of the query (W. Wang et al. 2015). The range that includes the location of the node is called MBR. The indexing strategy was performed over MBR by using an R-tree. The proposed indexing method suffers from a set of

Universiti Malaya

drawbacks, which resulted from an R-tree. This method has limited scalability, in case a large number of locations need to be indexed. To solve the mentioned drawbacks, they tried eliminating the need for frequent index updates. To perform indexing based on gateways instead of individual sensors. This minimizes the complexity and heterogeneity of sensors indexing and presents an easy way for sensor search and retrieval. Furthermore, it minimizes the coupling between applications and resources. Unfortunately, it is not suitable for mobile gateways, it is only suitable for predefined and fixed location gateway.

Abbasifard et al. proposed efficient index techniques called Past, Current Index (PCI) to index the past and the current position of the mobile objects on the road network (Abbasifard, Naderi, and Isfahani Alamdari 2017). The main advantages of the proposed techniques are the accuracy and reliability of the query results. Also, index building time is minimized because the tree structure minimizes the amount of indexed data, and minimizes memory consumption.

2.3.3 Prediction-Model approaches

Prediction-model is a process that uses data mining and probability to predict the outcomes. Predicting outcomes constitute the models, and each model has one or more predictors (Classifier). Predictors are variables that influence the predicted results.

The prediction model is widely used in the moving sensor's environment to predict values or events (Zhuo Zhou et al. 2013) (Zhang et al. 2016). The following works explain in detail the applicability of the prediction model in the mobility environment.

Zhou et al.proposed a data processing framework to deal with a large amount of heterogeneous data in IoT that is able to predict the failure of sensors (Zhuo Zhou et al. 2013). The framework focuses on protocol conversion, data processing methods, and integration with applications in the upper layer. The proposed framework is divided into layers; the device layer, physical layer, agent layer, data processing layer, device layer (repeated), and application layer. The framework mainly deals with problems like data

Universiti Malaya

collection, protocol conversion, congestion control, data filtering, data processing, data integration, and application, etc. The main advantage of the proposed framework, it performs well in the online monitoring system. Unfortunately, it is not applicable in reality and lacks prediction accuracy in the case of complex sensor movement such as random mobility.

In order to support the efficiency of the sensors search, Zhang et al. proposed LHPM which stands for "low overhead, high precision prediction model", which predicts the sensor output at the time of query to compare it with probability predicting, to achieve a high-precision (Zhang et al. 2016). To ensure high accuracy of determination of the sensor state, the multi-step prediction model is implemented; by using the sensors ranking method that assesses the matching probability of the sensors. The main advantages of the using prediction technique are saving more storage space and minimizing computing overhead on resources. The best advantage is supporting the search processes by minimizing the communication overhead during the search process. If the prediction model lacks practicability in a real application the results of the research will suffer from very low accuracy.

Qiu et al. proposed a protocol, which has the ability to minimize the paralysis rate and improve the power consumption of the sensors in IoT (Qiu et al. 2013). They used the estimation algorithm to predict the number of nodes that determined the size and shape of the monitoring region. This estimation enables it to determine the available multi-paths and to choose the optimal path that saves the sensor's energy. In the case of packets loss rate, it is good when the sink node is close to the sending sensors; otherwise, it is suffering from a high packet loss rate. And the packet loss rate is coming to failure when the number of nodes is high.

2.4 Summary and Discussion

The research on mobile nodes of WSN lasts for one decade and still draws much attention nowadays. Despite the great research effort, certain crucial issues remain unexplored and could become potential research hotspots. However, there is a conclusion, touched in the mentioned techniques in Section 2.3, which have a high impact on improving the index technique. Furthermore, the behavior of sensors such as mobility, coverage, etc. has a core relation to that improvement. However, an in-depth investigation of those methods enables us to determine the challenges of mobile nodes in WSN. Accordingly, this section investigates the challenges of the mentioned techniques from the mobility perspective. Thereby, it enables us to understand the behavior of mobile sensors to improve the indexing process.

According to the mentioned techniques, identifying the constraints of mobile nodes are not very clear and still generally speak. For example, the constraint of managing data of mobile objects to improve the indexing process. Unfortunately, the current works deal with mobile objects by dividing the space layout into grids, as mentioned before. For a clear understanding, Table 2.2 presents the approaches applied to mobile sensors and discusses how the proposed method may enhance and support the indexing process.

Table 2.2: Ability of the mentioned techniques effect on the indexing performance

	Ref.	Approach	Objective	How to enhance indexing?
Meta-heuristic clustering				
1	(Ebrahimi et al. 2016)	SSONs	It clusters mobile sensors based on the similarity in the context of information.	It minimizes data loss to increase indexing accuracy.
2	(Loscrí, Natalizio, and Guerriero 2012)	PSO	It controls the sensor's movement using PSO to improve network coverage, by determining an	It needs highly locations' update of the moving sensors.

Universiti Malaya

			interesting area using pioneers.	
3	(Ni et al. 2015)	Multi-objective optimization problem, and particle swarm optimization (PSO)	It improves the quality of service of the network by adjusting the location of mobile entities.	It needs to focus on data management.
4	(Rao, Jana, and Banka 2016)	PSO-ECHS	It saves the node's power.	It needs to focus on data management.
5	(Elhabyan and Yagoub 2015)	Two-tier particle swarm optimization protocol	It finds the optimal cluster head, in order to find the optimal route.	Challenge of the classification of data that affect negatively on the performance of the index.
6	(Yang G. Kim and Lee 2014)	Simulated annealing (SA) and particle swarm optimization (PSO)	It exploits multiple channels and time slots to enable multi-hop sensor networks	It minimizes the index overhead using scheduling.
7	(Khoshkangini and Zaboli 2014)	Ant Colony Optimization (ACO) combined with Breadth-First Search (BFS)	It improves the sensor's energy consumption.	It Increases indexing accuracy because it minimizes data loss.
8	(Luo, Lu, and Cheng 2013)	TSOIA	It intends to support applications that do not require separating the space for finding optimal solutions.	The routes to the solution are not always optimal.
9	(W. Li et al. 2014)	SACHSEN	It manages executable applications and scheduling tasks	High possibility of packet loss.

Universiti Malaya

			by allocating them to the appropriate sensor.	
10	(Saleem, Ullah, and Farooq 2012)	Bee-colony Algorithm	It uses the swarm method to find an on-demand multi-available path without the need for and geographical information.	Unable to solve the route coupling problem, because bee-colony algorithm considers as agent-to-agent communication to discover optimal paths. The scalability challenge is increased because the amount of transferred data increased.
Grid-based Indexing approaches				
1	(Chen, N. 2015)	Decomposition Tree (D-tree)	For handling multi- dimensional movement data	It divides the area into a partition called Rectangle. The partition is executed by an encoding method based on the bit-shifting operator.
2	(Ding et al. 2012)	IoT-SVK Search Engine)	Hybrid Search Engine Technique for the Internet of Things based on Spatial-Temporal, Value-based, and Keyword-based Conditions.	It did not take mobility as an issue.
3	(Ma et al. 2012)	UQE-Index	It solves problems resulting from the frequent update of moving sensor node data.	It proposes a key-value store, the same as the primary key in a database table.
4	(H. Y. Lin 2012)	B+-tree based indexing	It supports the processing of moving objects multi-dimensional spaces.	It performs indexing for the trajectory of moving objects but bases on the predefined temporal attribute.
5	(Tang et al. 2014)	Hierarchical clustering indexing	It divides the region into small areas called grids and performs clustering to index these grid cells, for answering queries.	- It is not applicable for IoT applications like environmental monitoring. - It is not suitable for random mobile sensors.

Universiti Malaya

6	(Y. Li and Parker 2012)	Kd-tree	It detects the nearest neighbor by considered weighted variances and weighted Euclidean distances from measured percentages of missing data.	- It does not give accurate results in the case of the distribution of data.
7	(Zhang et al. 2015)	Estimating newer sensors	It performs the architecture of a high-productivity content-based sensor search system and then applying estimation prediction.	- Lack of applicability and accuracy.
8	(Rao, Jana, and Banka 2016)	ECH-tree construction Hierarchical clustering indexing	- It facilitates time-correlated region queries in the IoT. - It divides the region into small areas called grids, and performs clustering to index these grids cells, for answering queries.	- It is not applicable for IoT applications like environmental monitoring. - It is not suitable for random mobile sensors.
9	(Zhangbing Zhou et al. 2014)	EGF-tree construction	It facilitates multi-region query aggregation to support energy efficiency.	It designs an index-tree based on grid division and minimum energy merging principle in the skewness distribution of sensor nodes.
10	(OOSTEROM 1999) (Butz 1969) and (Hilbert 1891)	A space-filling curve and Hilbert ordering	It converts the specific points in certain dimensional spaces to a set of numbers.	It converts two-dimensional into one-dimensional space.

Universiti Malaya

11	(Bozkaya and Ozsoyoglu 1997)	DOuble Transformation (DOT)	It applies double-transformation.	It converts two-dimensional into one-dimensional space.
12	(Yianilos 1999)	Quad-tree	It divides the area into squares to build a tree-structure. Furthermore, each internal node in Quad-tree is divided into four children.	It divides the area into squares.
13	(Antonin Guttman 1983) and (Beckmann et al. 1990)	R-tree and R*-tree	It represents mobile data objects by intervals in several dimensions.	It partitions the network area into grids called MBR.
14	(Jo and Jung 2018)	Quadrant-based Minimum Bounding Rectangle (QbMBR)	It provides more selective query processing and reduces the storage space.	It stands on grouping the spatial objects precisely using QbMBR.
15	(A. Zhou et al. 2000)	Sample-based DBSCAN	Support mobility	It expands the clusters by representing a few numbers of objects as seeds, instead of all the neighbor objects.
16	(Boonchoo et al. 2019)	Grid-based DBSCAN with Cluster Forest (GDCF)	It minimizes the neighbor explosion and merging redundancies. It utilizes HyperGrid Bitmap (HGB) structure to index the grids by neglecting the empty ones.	It depends on partitioning the data layout.
17	(Altti Ilari Maarala, Xiang Su 2016) and (Zhao,	GF-DBSCAN and Grid-based	They reduce the number of searches and redefines the cluster cohesion	They depend on partitioning the data layout.

Universiti Malaya

	Zhang, and Shen 2004)	DBSCAN algorithm	merging. Furthermore, they partition the space layout into intervals and hyper-rectangular grids.	
Prediction Model Clustering				
1	(Zhuo Zhou et al. 2013)	A framework deals with a large amount of heterogeneous data in IoT.	Sensor failure prediction	It is not applicable in reality. Lake of prediction accuracy in case of complex sensor movement such as random mobility.
2	(Zhang et al. 2016)	Low-overhead and high-precision prediction model (LHPM).	It improves the sensor search efficiency.	It reduces the communication overhead of the search process that affects positively the index.
3	(Qiu et al. 2013)	Multipath routing organizing protocol.	It improves multipath.	Lack of accuracy in the case of complex environments such as IoT, and mobile environments.

As mentioned before, meta-heuristic approaches are used in mobile sensors. It mainly concentrates on the node's coverage, data routing, node power consumption, etc. Therefore there is no work concentrated on how the data are collected and arranged, which negatively affects the performance of indexing. The meta-heuristic approach should not ignore the network topology design and homogeneity of nodes. Also, the dependence of meta-heuristic approaches on the nodes' location affects directly by the performance of dealing with data. So, the need to develop meta-heuristic approach dealing with heterogeneous data without the need for the location coordinates appears.

Grid-based indexing reveals a way of dealing with mobile objects, based on location and time, as mentioned before. Unfortunately, this way does not solve the problem of the mobility environment in the case of indexing. There is a conclusion, touched in Table 2.2,

Universiti Malaya

which the current techniques are not taking into account the way of collecting data, and how to arrange it. In other cases, all of the approaches focusing on enhancing the indexing technique to be suitable for the huge amount of data. I.e. Quad-trees and Octrees to compose the approach D-tree.

As shown in Table 2.2, there are few works related to prediction-model, because those methods do not prove competence in heterogeneous environments, such as IoT. The prediction model works to predict outlier data clustering based on a predefined variable, but the compatibility of this for all types of data is not useful since it neglected the semantics of data. Also, it suffers from high computational overhead because of more complex clustering methods.

Table 2.3 summarizes approaches that are used to investigate mobility and indexing. Furthermore, it summarises the challenges that resulted from those approaches that negatively affect indexing performance. The main reasons to discuss the mobility approach are: (i) to understand the current behavior of mobile sensors and its effects on the indexing, (ii) to understand the current gaps and deformities that the mobile nodes in WSN are suffering, and the effects on indexing and (iii) to understand the challenge of mobility in increasing the severity of the amount of sensed data, that effect on the performance of indexing process.

Table 2.3: Approaches used to deal with mobility sensors

	Classification	Definition	Characteristics	Challenges
1	Meta-heuristic based	Use meta-heuristic algorithms (PSO, ant colony, etc.) to enable the sensor's routing.	Apply control of mobility sensors, and it is suitable for random mobility.	- It did not consider the presented data status and its properties. - It ignores the consistency amongst nodes and the topology design, which results in low

Universiti Malaya

				connectivity between them. - The search capability is poor.
2	Grid-based indexing approaches	Apply indexing clustering on sensors or data by taking into account time and location information.	Suitable for multi-dimensional movement data.	Overhead on dealing with data in case of complex movement data.
3	Prediction model-based approaches	Estimate the future state of the sensors based on data generated.	Reduce the overhead of communication, especially in the case of movement data.	Low accuracy, especially in the huge amount of data.

2.5 Comparison of indexing techniques based on the challenges in mobile sensor nodes

So far, the existing challenges of the mobile environments are deduced, as summarized in Table 2.3. Thereby, this section presents a review of the state-of-the-art index techniques that have proposed for confronting the existing challenges in the mobile sensor nodes (mentioned in Section 2.3). Accordingly, the ability of each technique to solve the problems that appear in each category is described (mentioned in Section 2.4). Figure 2.4 represents a list of indexing techniques, which try to solve the challenges in each classified approach. The details discuss as follows.

Universiti Malaya

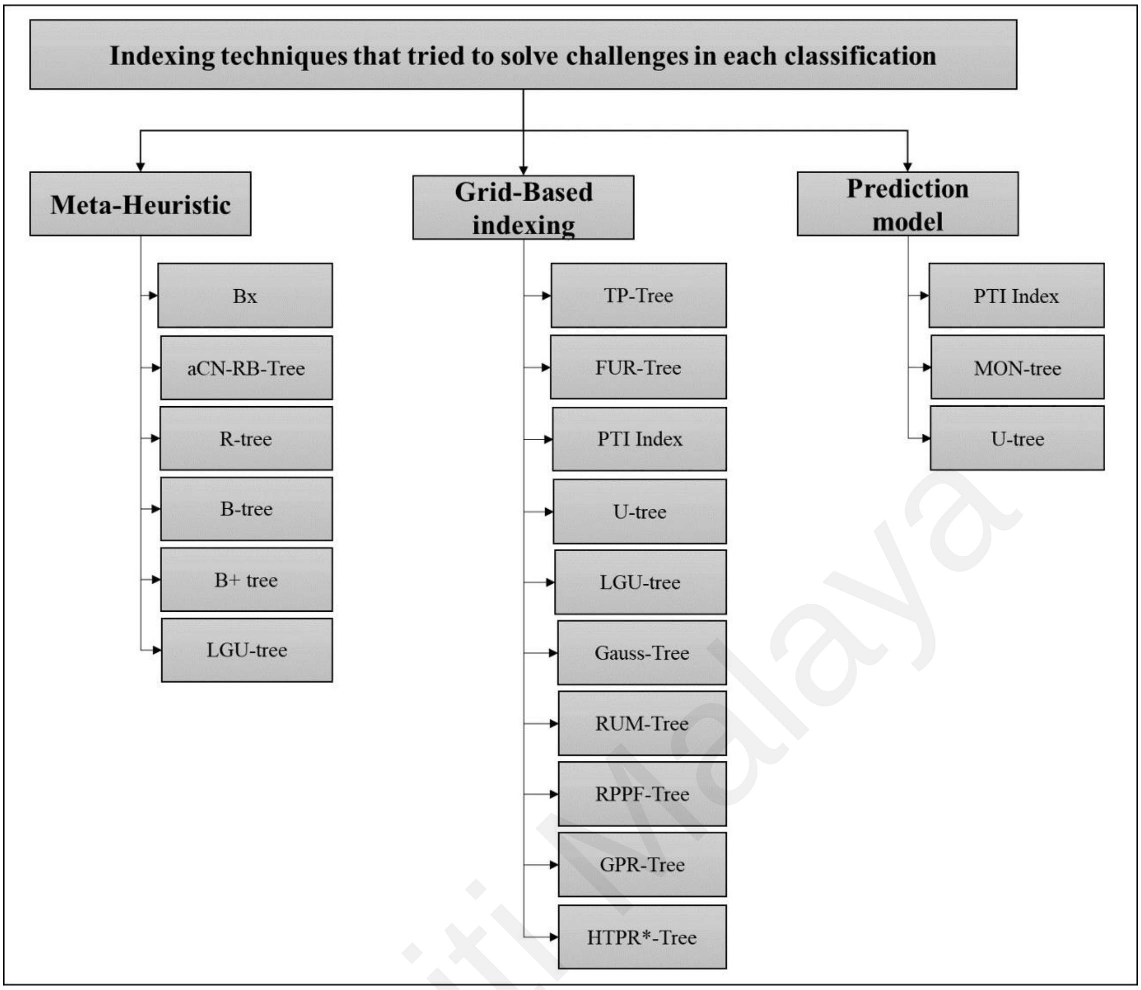

Figure 2.4: Indexing techniques that tried to solve challenges in each classification

2.5.1 Meta-Heuristics approaches

As mentioned in Section 2.4, meta-heuristics techniques applied to mobile nodes in WSN effects the performance of indexing techniques. There are some requirements that the index techniques need to cope to index the mobile data efficiently. First, the need to prepare transferred packets before accumulated them in the final destination for indexing, to minimize the indexing overhead. Additionally, the efficient index structure has to consider the attributes of packets (location and time). B_x-tree tried minimizing the index updating overhead based on MBR. B_x-tree structure tried to group the packets based on transferring time (time attribute). Another work that tries to fulfill this requirement is aCN-RB-Tree, which is spatial-temporal index technique that tried to manage the indexed data in accordance with the location of the data on transferring time. Furthermore, R-tree, B-tree, and B+-tree are also tried to achieve this requirement.

Second, the packets that transfer from different sources increase the burden of indexing. Because it increases the accuracy of indexing by minimizing data loss. Data loss results in ignoring the consistency amongst nodes. To achieve that, U-Grid indexing technique is proposed to achieve this requirement by applying the frequent update of the locations of the sensors. LGU-tree also tried to fulfill this challenge by performing an index for up-to-date locations of the mobile sensor nodes.

Third, the need for arranging the transferred data that belong to the same source together. Actually, till now no real indexing techniques focus on this requirement. Most of the proposed indexing techniques base on classifying the received data based on the data attributes, such as transferring time, location, and so on. I.e. R-tree bases on MBR classification, as mentioned before.

2.5.2 Grid-based indexing approach

In line with the nature of the mobile sensor's packets, the core factor that improves the indexing process is the ability to manage and arrange the generated packets. Namely, the packets should be managed and arranged before they accumulated in the final destination for indexing. There are many existing indexing techniques able to overcome this challenge. Such as TP-tree, PTI Index, TP-tree, FUR-tree, PTI Index, U-tree, LGU-tree, Gauss-tree, RUM-tree, RPPF-tree, GPR-tree, HTPR*-tree. Unfortunately, all of the mentioned techniques are not able to arrange and manage packets in WSN, especially in random mobility.

In more detail, TP-tree, PTI Index, TP-tree, FUR-tree, PTI Index, U-tree, LGU-tree, Gauss-tree, RUM-tree, RPPF-tree, GPR-tree, HTPR*-tree are all indexing techniques tries to solve the challenge of dealing with packets in complex mobility environments i.e. random mobility, to minimize index overhead. As mentioned before, the aforementioned

Universiti Malaya

techniques bases on classifying packets according to specific circumstances. I.e. TP-tree bases on a query window. FUR-tree bases on classifying based on locations, as mentioned in Section 2.2.

Another challenge is the index dependability on classifying the sensed data -mentioned before-. For Example, let say that R-tree index (Antonin Guttman 1983) is the technique used to index the data in mobile WSN. R-tree performs indexing by dividing the network area into grids each one called MBR. Each MBR contains sensors, which consider as a parent for each of them. This technique is optimal in the case of stationary sensors, but in the case of mobile sensors, the overhead of indexing is exacerbated because each sensor may leave the grid and re-enter again.

In sum, most of the Grid-based indexing techniques tried to achieve the requirement by enhancing the way of the divided network area, without taking into account the preparation and arranging of data. Unfortunately, they still suffer from a set of gaps and problems and did not achieve the division requirement. This case will discuss in detail like an open issue in Section 2.6.

2.5.3 Prediction model approach

The main challenge in this approach is the lack of accuracy. This challenge increases the burden on indexing techniques. This burden is also exacerbated in the case of mobile sensor objects. In reality, all the indexing techniques aim to achieve high indexing accuracy. The techniques based on the prediction model negatively affect index accuracy. This is because this approach is based on predicting the mobile objects, which increases the fault percent that negatively affect the index accuracy.

PTI-Index tried to solve this challenge. This technique tries to achieve high accuracy by classifying mobile objects according to the level of uncertainty. Level of uncertainty means classifying the mobile objects that have the same level of uncertain classification together. MON-tree represents the moving objects by polylines that represent edges and

Universiti Malaya

route. There is another technique called U-tree, which means trimming sub-tree that does not have any results.

2.5.4 Summary and Discussions

The research on index mobile WSN lasts for at least one decade and still draws much attention nowadays. Despite the existence of a lot of techniques, certain crucial issues remain unexplored and could potentially become research hot topics.

The current mobile indexing techniques that try to improve the indexing in mobile WSN, can be carried out by analyzing in detail the gaps and challenges that exist in the mobile IoT. Table 2.4 summarizes the problems and challenges that are inherited from the mobile node's environment and relating them with an index that tried to solve this kind of problem.

Table 2.4: Proposed mobile indexing techniques that solve the mobility problems

	Challenge	Index strategy that tried solving it
Meta-heuristic approaches		
1	- It did not consider the presented data status and its properties.	Bx, aCN-RB-tree
2	- It ignores the consistency amongst nodes and the topology design, which results in low connectivity between them.	U-Grid, LGU-tree
3	- Research capability in mobile environments is poor.	R-tree, PTI Index, TP-tree, FUR-tree, U-Grid, MON-tree, Gauss-tree, RUM-tree, PCFI+-Index, HTPR*-tree, MSMON, PCFI+-Index
Grid-based indexing approaches		
1	Overhead on dealing with data in case of complex movement data.	TP-tree, PTI Index, TP-tree, FUR-tree, PTI Index, U-tree, LGU-tree, Gauss-tree, RUM-tree, RPPF-tree, GPR-tree, HTPR*-tree,
Prediction model approaches		
1	Low accuracy, especially in the huge amount of data.	PTI Index, Bx, MON-tree, U-tree, Gauss-tree, ST2B-tree, BBx, RPPF-tree, VPMR-tree, PPFI, GPR-tree, MSMON, PPFN*, FT –tree

As concluded from the previous section, each classification category has a set of problems that require to deal with to improve the indexing efficiency. In meta-heuristic

Universiti Malaya

category, i.e. B_x is able to deal with any kind of data. This is because it constructs the index accuracy according to time-stamp and location. R-tree is still the most important indexing technique, where its structure can support mobility, based on the MBR.

As a result, existing mobile indexing techniques still suffer from a set of gaps and challenges. These challenges require much attention and effort, especially for complex environments such as random mobile WSNs.

2.6 Open research issues and Future Direction

So far, this chapter presents a deep investigation of the current indexing techniques and analyzed indexing challenges from the mobility perspectives. Also, this chapter established a relationship between the behavior of mobile nodes and index performance. Indeed, an in-depth investigation of the current indexing techniques' ability to solve the deformities is conducted. However, the current indexing techniques still face some challenges, namely, when the mobility of the sensors is random, as mentioned earlier in previous sections. Accordingly, besides the issues addressed earlier, some open issues still need further exploration. The following open issues describe in Subsection 4.1.3.

(a) *Mobile sensor's behavior in terms of indexing collected packets*

The deformities caused by the mobile sensor's behavior directly affect the performance of the indexing in terms of collecting packets. The mobile sensor's packets have attributes because each one is generated from different locations and times, as mentioned. The indexing process in the mobile environment is based on those attributes. Thus, the need for checking each attribute's packets one-by-one increases the burden of index performance. Furthermore, the need to find a way able to arrange the received packets step-by-step before accumulated in the final destination for indexing. Accordingly, the packets accumulated in the server increase the burden of the indexing process.

Although there are great efforts devoted, the way of collecting data of mobile sensors for indexing is still missing. I.e. in (Chen, N. 2015), the moving objects exceed the

Universiti Malaya

currently determined rectangle, which results in losing this sensor by missing its location. Unfortunately, this issue results in a large number of index updating operations. The mentioned works only provided a basic understanding of certain problems. Different open issues are left and need further exploration.

- Mobility sensors may move-out of the division grid, which results in losing packets. Losing packets make the collected packets ineffective and exacerbating the index updating average. I.e. all the techniques based on MBR, such as LUGrid (Xiong, Mokbel, and Aref 2006), EGF-tree (Zhangbing Zhou et al. 2014), suffer from this gap because they inherited it from R-tree indexing technique (Antonin Guttman 1983).
- The size of packets that requires indexing is duplicated because each sensor has to send its current location for different periods of time. More explanation, the packets that transfer from the same sensor is generated from different locations and time periods. In this case, indexing all the packets that belong to the same sensor consider as new values, which results in increasing indexing overhead.
- Because of mobile network entities, the network topology is changing randomly. So, it increases the index overhead. Index overhead increases because of the need for determining the source of packets and the destinations, each time the sensor wants to transmit the packets to the sink. Grid-based indexing techniques still suffer from this gap.

(b) *Challenges of connectivity between source and destination*

Currently, the interactivity among sources and gateways destinations in the mobile environment is a real challenge. Researchers rely on exploiting nodes' mobility as a way of optimizing the performance of a network. In a WSN environment, sensors have to be able to transmit the collected packets to the gateway. Therefore, the sensors have to know all the transferring information about the neighborhood gateways.

Universiti Malaya

The following open issues are still missing in the current works:

- A way that enables sensors to know sufficient information related to the gateways. Knowing the pre-knowledge of the generated packets' attributes minimizes the indexing overhead.
- The hardness of finding more than one optimal gateway as the destinations, in the case of mitigation disseminated packets that belong to the same sensor. The disseminated packets increase the indexing overhead because it requires looking for packets and finding the appropriate position in the index node during building the index structure.
- The need to find a reference for each received packet that indicates the location and time of transmission and is transmitted from which source.

(c) ***Preparation of the packets before sending to next destination***

A duty of sensors is scanning the environments based on certain circumstances, then transfer the packets to the appropriate destination. Without any control and management related to those packets, it negatively affects the indexing performance. More explicitly, the packets are collected directly without any enhancement or at least minimize the complexity of it, i.e. removing duplicates. This results in increasing the overhead of indexing those packets. However, to the best of our knowledge, there is little research focus on preparing data to become ready to index.

(d) ***The challenge of clustering in the mobile sensor nodes' environment***

Sensors may be clustered to form a hierarchical topology and determine cluster-head for each cluster. In each cluster, the source nodes transfer packets to the cluster-head, which in turn gathering and dropping the redundant packets, then transmits them to the final destination. Although the failure of the cluster-head often requires re-clustering.

Universiti Malaya

However, to the best of our knowledge, little works are focusing on the problem of clustering in sensor environments. The following represent the issues that still missing in current works:

- The hardness causes by determining cluster-head.
- The packets disseminate throughout clusters, which increase the difficulty of indexing and index overhead.
- Ignore the consistency amongst nodes, which affects connectivity between them, to increase the index clustering accuracy.
- The performance of indexing is decreased in the case of large clusters.

(e) ***Challenge of dealing with Big-Data***

This issue still considers an open issue that needs to be solved, because of the amount of data that transferes per small period of time. This difficulty pops-up because of the nature of reported packets of mobile sensors that have the following characteristics, volume, velocity, scalability, and variability. According to these characteristics, reported packets in random mobile sensors consider a type of big-data (Gärtner, Rauber, and Berger 2014). Velocity, defined as the high speed of the sensor's reporting packets (Jinchuan Chen et al. 2013). Velocity increases the number of indexing operations (update, delete) within a time period to manipulate the speed of the reported packets. Scalability, the data are continuously generated from different sources (sensors), which means the same sensor transfers data from different locations (Kaisler et al. 2013). Scalability increases the index building time and index space cost. Variability, the data are the results from mobile nodes, where the generated data is changing dynamically, according to time periods and locations of the sensors. Fragile data, which means the data, will change continuously over time (Kaisler et al. 2013). According to the requirements of big-data, the effective indexing technique requires satisfying and dealing with these

Universiti Malaya

big-data requirements (Gani et al. 2016). Hence, this type of data exacerbates the need to promote the indexing technique able to deal with the random mobility of the sensors.

So far, the aforementioned open issues consider as mobile WSN indexing problems. Meanwhile, the database indexing problem suffers from a set of problems, such as it does not take into account the status of the indexed objects, where consider each stable or mobile object as a stable one. Additionally, suffering from low accuracy since its dependability on the fixed factor such as the primary key. It suffering from high overhead complexity in dealing with data for indexing, since high updating frequency. Last but not least, the research capability is poor especially in large and complicated environments, such as IoT, heterogeneous mobile WSN, and random mobile WSN environments.

2.7 Chapter Summary

This chapter presented a literature review of state-of-art indexing techniques. Accordingly, an in-depth investigation of the challenges of these techniques from the mobility perspectives is presented. Furthermore, the techniques and methods, which applied to mobile environments were investigated. This investigation enables initiating the relationship between mobility behavior and the efficiency of the indexing. To simplify that, these techniques are classified into three categories: meta-heuristic, Grid-based indexing, and prediction model approaches. The current techniques are analyzed in each category from the mobility and indexing perspectives. Finally, besides some issues addressed earlier, some open issues were identified for future research directions. As previously stated, this chapter aims to identify and establish a problem that has a significant impact on indexing.

According to the relationship between the mobile sensor nodes' behavior and the efficiency of indexing, a novel solution will be introduced, which addresses the highlighted issues. So, developing an efficient framework that mitigates the effects of

Universiti Malaya

random mobile sensors on indexing efficiency is vital. Hence, this work proposes DySta-Coalition framework dedicated to random mobile sensors in WSN, in order to improve the efficiency of indexing. Thus, the subsequent chapter is a detailed description of the research methodology.

Universiti Malaya

CHAPTER 3:

RESEARCH METHODOLOGY

This chapter lays out in-depth the strategy phase of the proposed developed model to accomplish the aim of this research, which is to improve the indexing method for random mobile WSNs. Before attempting to clarify the proposed DySta-Coalition model as a solution, it is critical to formulate and define the statement of the problem, mentioned in Section 1.3.

The analysis presented in Chapter 2 (literature review) reveals that existing mobile environments' indexing techniques still involve challenges. These significant challenges are (i) the challenge of indexing the packets, due to the mobile nodes’ nature, (ii) the lack of cooperation between sources and destinations, (iii) the lack of packets’ preparation and arranging before transferring them to the next destinations, and (iv) the challenge of clustering. The main reason for these challenges is the random nature of the sensor’s nodes, as detailed in Section 2.6. In the face of such issues, the random mobility of sensors' nodes emerges as a substantial problem. Consequently, these challenges contribute to the problem statement, which is mentioned in Section 1.3 as follows: the existence of disseminated packets throughout destinations that need to manage and arrange. As well as, the need to regularly register the existing network division grid (Grid-based indexing) details for each source. Hence, the purpose of this chapter is to formulate the mentioned problem. Additionally, present the proposed model as a solution to overcome the stated challenges.

This thesis proposes a solution called DySta-Coalition model. The first contribution of this proposed solution is to minimize the number of disseminated packets throughout gateways, via constructing dynamic blocks based on the coalition. The second contribution is initiating static-coalition in each gateway that alleviates the effects of the Grid-based index structure and removing the redundancy of packets, which results in

Universiti Malaya

arranging and preparing packets step-by-step before they accumulated in the final server for indexing. The third contribution proposes a variant indexing-tree, called Coalition-Based Index-Tree, which replaces the division of the network area by dynamic-coalitions.

The problem analysis in this chapter and the simulation results in Chapter 4, and Chapter 5 show that the proposed model can efficiently overcome the mentioned problem and create a novel model to deal with random environments comparing with previous works.

The structure of this chapter is as the following. Section 3.1 presents the methodology, that solves the problem of this work. Problem formulation and Empirical analysis present in Section 3.2 and Section 3.3, respectively. The overview of the proposed model to solve the problems presents in Section 3.4. Furthermore, the role of each part of the proposed model to solve the indexing problems in random mobile WSNs are presented in Section 3.5, 3.6, and Section 3.7. Finally, the chapter is summarized in Section 3.8.

3.1 Research Methodology

The methodology of this thesis is constructed using four main stages to achieve the objectives in Chapter 1.

- **Literature review:** In this stage, a summary and critical analysis are performed on the current indexing techniques. As well, the limitations and challenges of these techniques are reviewed and represented in Chapter 2. The indexing techniques' challenges are analyzed from the mobility perspective, as detailed in Section 2.2. To understand the nature and effects of mobile environments on the indexing process, deep investigations of mobile environments are classified into three categories: (i) meta-heuristic, (ii) Grid-based indexing, and (iii) prediction model approaches, as represented in Section 2.3. This helps in establishing a relationship between mobility behavior and the efficiency of the indexing. Also, in this stage, it is shown to what extent the existing indexing techniques can overcome the challenges in random mobile WSNs, as presented in Section 2.4. Furthermore, the issues of existing indexing techniques for indexing random mobile WSNs presents in Section 2.5. Accordingly, the analysis of the existing techniques is drawn a wider picture of the problems in indexing packets of random mobile sensors, as presented in Section 2.6.

Thus far, there are huge research initiatives, which dedicated to boosting the indexing of mobile nodes in WSN, (Babenko and Lempitsky 2015) (Puri and Prasad 2015) (Alvarez et al. 2015) (Amato et al. 2015) (Chen, N. 2015) (Jidong Chen 2007) (Y. Fang et al. 2012) (Liu and Liu 2005) (Ying et al. 2008) (Huang, Hu, and Xia 2010) (Tang et al. 2017) (Jo and Jung 2018). Unfortunately, these works still struggling from a host of challenges. One of the main reasons for those challenges is that some of the literature utilized the sensor's mobility as a solution. More explicitly, using mobility as a solution to solve specific network challenges, i.e. sensor's coverage (Y.-C. Wang, Wu, and Tseng 2012) (Hubaux 2010). Meanwhile, nodes' mobility considers as a substantial challenge in WSN and results in great challenges. Thus, it is required to consider the sensor's mobility as a major issue, which causes challenges and still needs to be solved. On another hand, to the best of our knowledge, few studies address the problem of indexing in random mobile WSNs. This leads to the emergence of the need to manage and control packets, in order to improve indexing in random mobile WSNs.

- **Modeling**: After the existing research is reviewed and the literature review is analyzed, the features and challenges of random mobile WSNs are identified and categorized. The proposed model considers the major problems of the indexing packets of random mobile environments, which are (i) the existence of disseminated packets throughout destinations. (ii) the need to manage and arrange packets in the destinations. (iii) each destination required to register the current division of Grid-based. In this stage, all the required components of the proposed model are discussed in detail, in Section 3.4, Section 3.5, Section 3.7, and Section 3.8.

- **Development:** The detail of this stage presents in Chapter 4, and Chapter 5, in which the proposed model is developed to simulate indexing packets in random mobile WSNs. In this stage, the dependent and independent variables are determined. The independent variables are Routing Overhead, Packets Delivery Ratio, the number of traversed nodes, index space cost overhead, and index building time overhead. Meanwhile, dependent variables are the number of dedicated sensors, and the number of sensors, and the radius. Furthermore, this stage presents the formulation of hypotheses, to predict the relation between the aforementioned independent and dependent variables. As well as, it presents the comparison measurements that use to identify and validate the evaluated results, followed by the specification of (Jeongcheol Lee et al. 2018) and (Boonchoo et al. 2019).
- **Testing and Evaluation:** This stage discusses the evaluations and validations for the framework and explains the evaluated results. To achieve that, the experimental processes divide into two stages. First, evaluate Dynamic-Coalition framework, to examine that it achieves the minimum overhead by comparing it with Active data Dissemination (Jeongcheol Lee et al. 2018), presents in Chapter 4. Second, evaluate Coalition-Based Index-Tree compared with Grid-based DBSCAN with Cluster Forest (GDCF) (Boonchoo et al. 2019), R-tree, and D-tree present in Chapter 5. This stage has accomplished according to three scenarios, (i) evaluate the performance of Coalition-Based Index-Tree, (ii) evaluate the effects of Dynamic-Coalition framework on

Universiti Malaya

Coalition-Based Index-Tree, and (iii) the effects of Static-Coalition algorithm on Coalition-Based Index-Tree. The proposed framework is implemented using a simulator developed by Visual C# 2017 (Ghaleb et al. 2017), as detailed in Subsection 4.1.3.1, and Subsection 4.1.4.

The experimental implementation conducts from sensors and gateways that are deployed randomly. The implementation is designed for random mobile environments; hence the sensors and gateways possess dynamic behavior. The sensors regenerated packets during each iteration by establishing new relevancy between sensors and gateways, where the sensors and gateways locations are changing. Additionally, the details of the datasets and implementation environment are defined, which are applied to show the effectiveness of the proposed model. Firstly, the conducted evaluations of the proposed framework are performed on synthetic datasets, which are generated using a program written in C#. Different synthetic datasets are generated with the various number of sensors, and simulation time, in addition to the stationary number of gateways.

3.2 Problem formulation

Before attempting to solve the research problem, it is important to precisely formulate and analyze the problem, that this thesis intends to solve. To fulfill that, this section starts with preliminaries, in Subsection 3.2.1, to define the definitions and concepts used frequently in this work. Furthermore, the observations are mentioned in Subsection 3.2.2 to pave the way to formulate the problem, which mentions in Subsection 3.2.3.

3.2.1 Preliminaries

This section presents the details of the random mobile WSNs structure. As well as, it identifies the definitions and concepts used frequently in this thesis. Furthermore, the assumptions are identified to simplify problem formulation.

3.2.1.1 Structure of Wireless Sensors Network (WSN)

The structure of WSN composes of random mobile sensor nodes, gateway nodes, and a stationary server, formal definition of WSN and its population, as follows:

Definition 3.1 (WSN): WSN is a structure that contains geographically deployed stationary or mobile nodes that interact together to achieve some tasks under certain circumstances. The included nodes in WSN called the population of the network N. N is modeled by $N = \{n_1, n_2, \dots, n_x\}$, where n represents the population node and x represents the number of nodes in N.

Definition 3.2 (Sensors and Gateways): For each population node n_x in N, n_x can be either sensors or gateways. For network N in a set $N = \{n_1, n_2, \dots, n_x\}$, the n_x is either sensors and gateways, which represents as $S = \{s_1, s_2, \dots, s_a\}$ and $GW = \{gw_1, gw_2, \dots, gw_g\}$, respectively. Where $S \cup GW = N$, $S \cap GW = \emptyset$ and $a + g = x$.

The interaction among the distributed nodes is handled via the traditional architecture, in which all sensors and gateways are connected to a common server. The sensors transfer packets to appropriate gateways, then the gateways forward them to the corresponding server, which is formally represented in Definition 3.3. Figure 3.1 depicts the general overview of the WSN structure.

Universiti Malaya

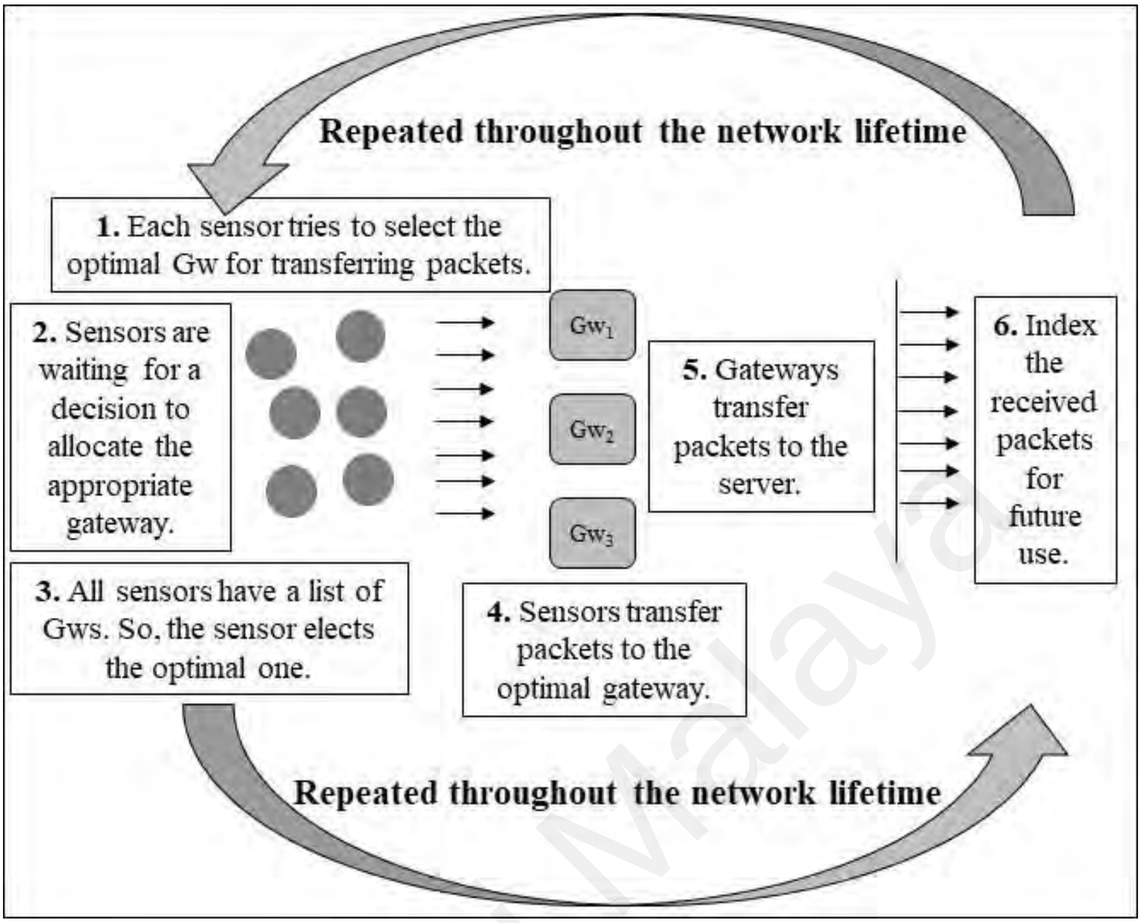

Figure 3.1: Overview of WSN structure

Definition 3.3 (Transferring Packets): Based on Definition 2, $\forall\, s_a in\, S$ transfers packets $Pack = \left\{p_{<ID_{s_a},T_1,L_1>}, p_{<ID_{s_a},T_2,L_2>}, \dots, p_{<ID_{s_a},T_t,L_l>}\right\}$ to the appropriate gateways in GW, by depending on the specific routing algorithm. Where: p is the transferred packet, ID_{s_a} is an identification of sensor s_a, T_t is a transferring time for s_a, and L_l is a transferring location of the sensor s_a. Then, $\forall\, gw_a$ in GW, transfers the received packets to the final destinations (server).

With reference again to Figure 3.1. In the sensor's part, each sensor search for an appropriate gateway as a destination. Hence, the information of gateways has to be shared between them. During this step, sensors are waiting for the decision to choose the appropriate gateway. Each sensor has a list of candidate destinations. An optimal gateway is chosen on the basis of certain conditions, such as relevancy-measure, represents in detail in Subsection 3.3.3. The sensors transfer packets to the selected gateway. Then, Gateways transfer the received packets to the server, where the indexing process starts. These steps iterate during the network lifetime.

Since this research is dedicated to mobility environments, particularly random environments, it is necessary to define the mobile nodes and random mobility. The definitions are as follows:

Definition 3.4 (Mobile nodes): Mobile nodes M_p is the process of changing the node's location (L_l) through time (T_t), where $M_p \epsilon$ N and $S \cup GW \epsilon M_p$. The mobile node called spatial-temporal nodes represents $< ID_{s_a}, T_t, L_l >$, where $ID_{s_a}$is the node ID_{s_a}, T_t is the transferring time, L_l is the transferring location. The movement of the nodes represents by paths, i.e. for sensor s_a in N, its path represents as $Path_{s_a} = \{< ID_{s_a}, T_1, L_1 >, < ID_{s_a}, T_2, L_2 >, \dots, < ID_{s_a}, T_t, L_l >\}$, where, $T_1, T_2, \dots$ and T_t is different time periods (not equals).

Definition 3.5 (Random mobile nodes): $\forall\ Path_{s_a}$ of mobile nodes $\{< ID_{s_a}, T_1, L_1 >, < ID_{s_a}, T_2, L_2 >, \dots, < ID_{s_a}, T_t, L_l >\}$, the inability to predict the next location $< ID_{s_a}, T_t, L_l >$ in the path (Random Waypoint Model) is called Random mobility.

3.2.1.2 Definition of network area division

To index packets in mobile WSN, the index structure is constructed based on grid-dependent indexing, i.e. the division of the network area into grids. A deep discussion is presented in Subsection 3.2.3. The formal definition of the mentioned concepts is as follows:

Definition 3.6 (Grid-based division): The region of the network N divides into grids G, where $G = \{g_1, g_2, \dots, g_n\}$, such as $g_1 \subseteq \{s_1, s_2, \dots, s_a\}, g_2 \subseteq \{s_3, s_4, \dots, s_b\}, \dots, g_n \subseteq \{s_5, s_6, \dots, s_c\}$, where n is the number of grids.

Definition 3.7 (Grid-based indexing): in Grid-based indexing, all sets in G (g_n) converts to parent nodes $N_{parent} = \{g_1, g_2, \dots, g_n\}$, where N_{parent} is a parent node, and

Universiti Malaya

n is a number of parents nodes in the index structure. The children of each parent's node in the index structure are the sensors in each grid.

3.2.1.3 Assumptions

In this subsection, the assumptions that have to consider in this thesis are listed, as follows:

1. The structure of WSN contains sensors and gateways. Sensors consider as a source that transfers packets to the gateways that connect through Wi-Fi.
2. The sensors directly connect with gateways. In other words, the sensors transfer packets to the appropriate gateway, then the gateway transfers them to the server, as described in Subsection 4.1.1.
3. WSN's nodes are all mobile (Random Waypoint Model, Definition 3.5 Page 77) –except server-, where nodes of sensors and gateways have random mobility. So, sensors transfer packets from different locations through time to the most suitable random mobile gateways.
4. The final index structure constructs in the server. It builds based on the packets received from each gateway.
5. The transferred packets from the sources are all of the same types, i.e. they are homogeneous.
6. Each sensor in the WSN has a unique id.
7. All network sensors' nodes are periodically reporting information.
8. The constructed coalitions –will present in detail in Section 3.4, 3.5,3.6,3.7- are defined as circles.

3.2.1.4 Illustrative example

Before continue to the next section, it is important to provide an example described in detail the proposed method. Accordingly, this subsection provides an example with

Universiti Malaya

simple sample data to describe the proposed method and describe its ability to alleviate the problem mentioned in this thesis.

Suppose WSN contains ten GWs and twenty-five S, and the network lifetime is five seconds. The sensors transfer ten packets to the gateways per one second. Then, gateways transfer the packets to the final destination represented as the server, as in Table 3.1.

Table 3.1: Summary of the parameters

#	Parameters	Number
1	GWs	10
2	Sensors	25
3	Network-lifetime	5 Seconds
4	Server	1
5	Packets	10 per one second
6	Total number of received packets by the server	(25*10)*5=1250

Grid-based Indexing starts indexing according to waiting for all packets received by the server, then starts dividing the network structure into grids. After that, the structure of the index starts building by converting each dividing grid to the node in the index tree. To add each packet to the appropriate location in the constructed nodes, required reading all transferring time and location (attributes) of each packet. Transferring time and location means checking the location of the senor that transfers this packet to ensure to which grid it belongs, and at what time.

The proposed method does not wait until the packets are accumulated in the server for indexing. It starts building the index structure since the first packet is transferred. This value is because the indexing method starts arranging the packets when transfers from sensors using coalition, and exploit the existence of gateways to arrange the packets as blocks. Furthermore, transferring packets as blocks alleviate the need to read the attributes

Universiti Malaya

of all packets as mentioned. The structure of the index is constructed according to the constructed coalitions when the packets are transferring from sensors to gateways. Additionally, the packets are added to the node for indexing as blocks, which alleviate the time and the cost for reading the packets' attribute one by one.

For more explanation, the current works require reading the transferring time and location for 1250 packets, meanwhile, the proposed method, arranges the packets into blocks by coalition and adds them as blocks. Suppose each block contains ten packets that mean it deals with 125 (1250/10). But without reading on the attributes.

3.2.2 Observations

According to the definitions introduced above, let's shedding a light on the issues mentioned in Section 2.6.

First, the behavior of random mobile sensors, in the term of collecting packets. This is because each packet has its attributes (location and time) in the mobile environment.

Second, the packets that belong to the same source (sensor) are accumulated in the server, which increases the overhead of the indexing process. The overhead increases because it needs to check each packet attributes one by one even if they transferred from the same source. Furthermore, without any arrangement and management of the transferred packets before transferring to the next destination, it adversely affects indexing performance. More explicitly, the directly collected packets lack arranging and management to minimize complexity. Additionally, the collected packet requires arranging the transferred packets from the same source together. Whether that does not happen, the index performance decreases. This is because it minimizes index-building time overhead and minimizes the number of traversed nodes overhead. Furthermore, one of the benefits of packets' managing and arranging is removing packets' duplicates, which minimizes the overhead of indexing, in terms of space cost overhead.

Universiti Malaya

Third, the challenges of connectivity between sources and destinations. This increases the disseminated packets by considering the transferring of packets by considering the routing data. This increases the disseminated packets throughout destinations.

Forth, the challenge of clustering in the mobile sensor environment. More explanation, current works based on the division of the network area into grids, mentioned in Subsection 2.5.2. Further explanation, let say that R-tree is a technique used to index the sensors in random mobile WSN. R-tree executes indexing by dividing the network area into grids, each one called MBR. Each MBR includes sensors, which considers as a parent for each of them. Accordingly, the number of generating packets within each MBR will increase the burden of the index overhead. In addition, when the sensors are moved from one MBR to another, it increases the index building time and updating overhead. Moreover, R-tree requires checking the source of each packet, which adversely affects the performance of an index, by increasing index building-time.

Accordingly, the following observations must be taken into account.

Observation 3.1: Suppose s_a is a sensor that moves randomly and belongs to a grid g_n. During the network lifetime $T = \{T_1, \ldots, T_t\}$ the times' number of changing its grid is $ch_{s_a} \geq 0$, where the optimal case when $ch_{s_1} = 0$ (does not change), but it is rare.

The nodes that move randomly are frequently changing grid during the network lifetime. This results in increasing dependability on the attributes of the packet, to check the transferring location and time for each packet.

Observation 3.2: According to Definition 3.3, Page 79, there is no requirement that the sources (sensors) are focused on, in order to transfer the packets to a particular gateway. The consequence is that each gateway comprises packets transmitted from the same source.

Observation 3.3: The gateways received repetitive (duplicated) and misarranged packets.

Example 3.1 and Example 3.2 intend to clarify Observation 3.1, Observation 3.2, and Observation 3.3.

Example 3.1: Figure 3.2 presents the way of Grid-based index arranges sensors, in order to index packets. As mentioned, Grid-based indexing bases on partitioning the network area into grids. Accordingly, it divides the network area into four grids $(0,0), (0,1), (1,0), (1,1)$. The sensors' nodes that move randomly are moving from grid to grid (change locations, labeled as L) during the time period, labeled as T_1, T_2, and T_3. Accordingly, it needs to assign each transferred packet from which sensor and belong to which grid, throughout time periods. For example, Grid (0,1) at time period T_1 transfers two packets from two sensors, and so on. This requires periodically register the current group area information (grid) of each sensor and each sensor need to assign the destination, in order to transfer packets successfully. Unfortunately, this technique did not minimize the dependability of the attributes in random mobile environments. Furthermore, the packets that belong to the same sensor nodes are disseminated throughout grids, such as the packets that transferred from S_1 are disseminated throughout Grid $(0,1)$, Grid $(1,1)$, and return to Grid $(0,1)$.

Universiti Malaya

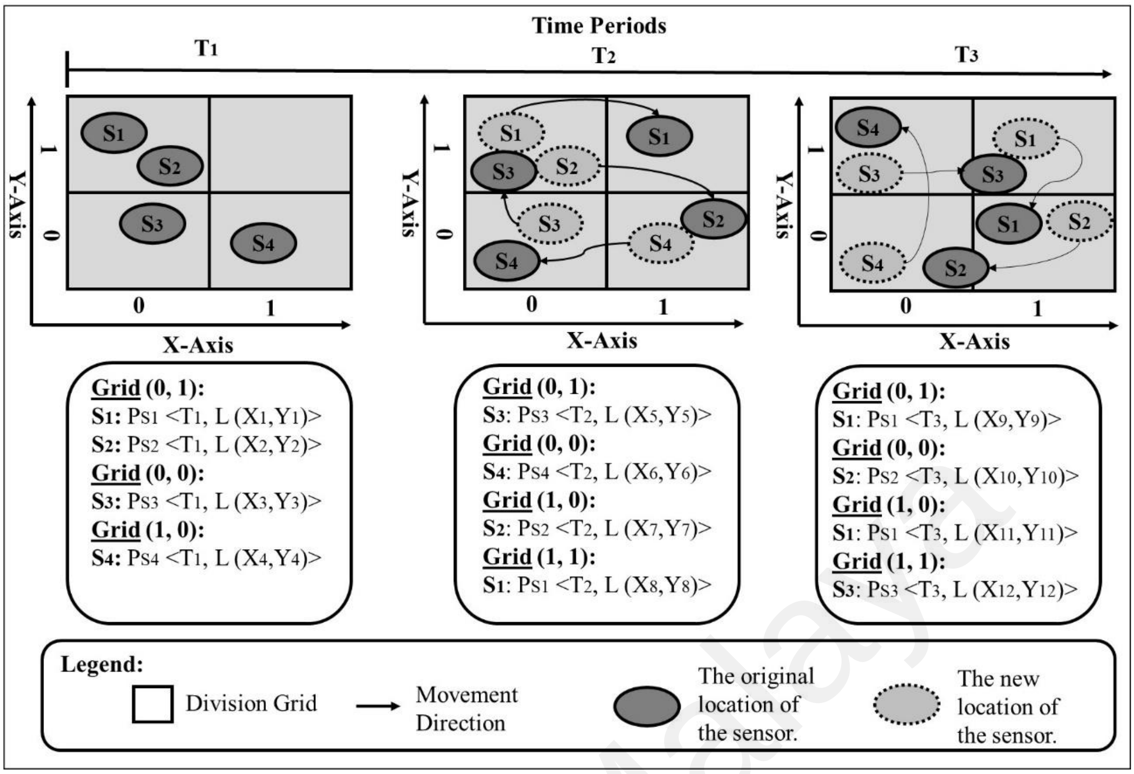

Figure 3.2: Arranging sensors in Grid-based indexing

The mobility behavior of random mobile sensors' nodes is a challenging problem because the packets are referred to each grid according to the random movement, which needs to locate the grids frequently because of this movement. To avoid that, it requires constructing dynamic blocks suitable to the random mobile nature of the sensors, using a coalition. This is fulfilled by a proposed framework called Dynamic-Coalition, presents in Section 3.4, and Section 3.5. It mitigates the disseminated packets throughout gateways to improve indexing.

Example 3.2: As represented in Figure 3.3, Part A, the Grid-based index the packets transfer to the final destination for indexing as in Part A. This increases the overhead of indexing because it needs to check each grid, then the source of the packet in each grid, and after that, it needs to check the period time and location of each transferring packet. To solve that, the gateways are exploited to manage and arrange packets in the same way as the indexing process. In other words, the arranging of packets considers as a partial part of indexing. The arranging performs according to the source (sensor node) and does

Universiti Malaya

not consider the grid, Figure 3.3, Part B. This minimizes the dependability on the attributes indexing, which will describe in the next subsection. Additionally, when the packets are arranging it transfers the packets as blocks called "static-coalitions" to the final destination for indexing.

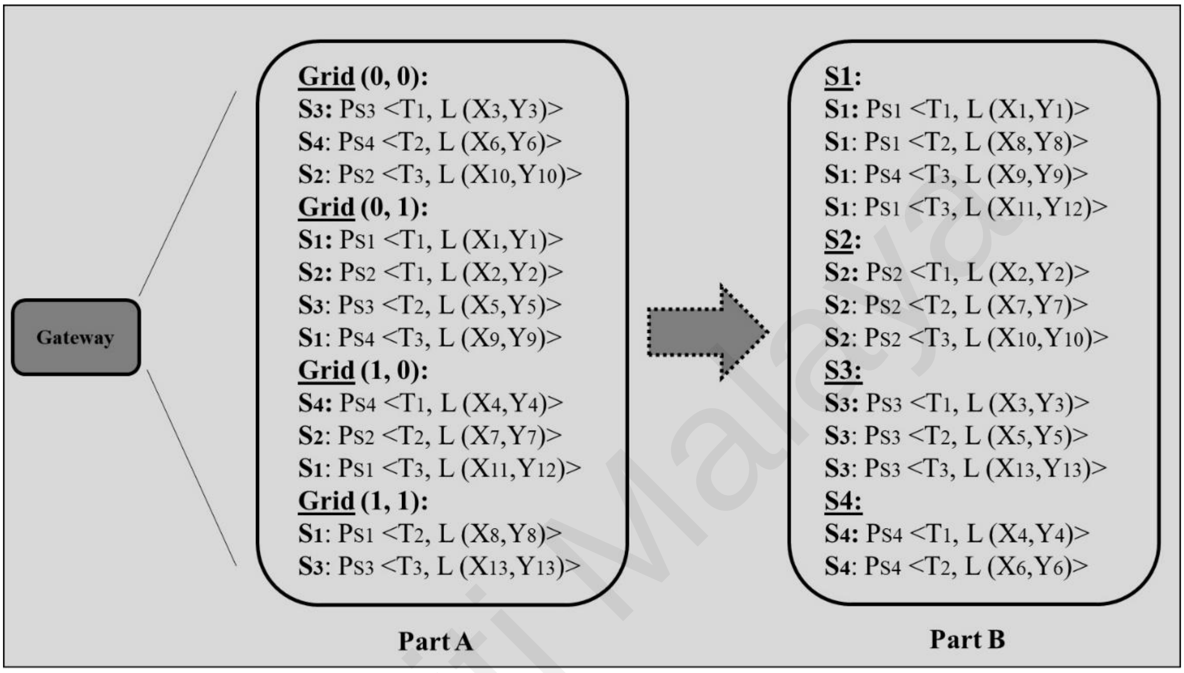

Figure 3.3: The arranges of packets in each gateway

Observation 3.4: Need to replace the dependability on the packets' attribute to build the index structure.

Example 3.3: According to Observation 3.3, the packets received by the final destination are misarranged, as mentioned previously in (b). Furthermore, these misarranged packets are large and need to read each packet attribute one by one to perform the indexing. So, the need to replace the dependability on the attributes of each packet is critical to improving the indexing procedures.

3.2.3 Demonstration of the Problem

So far, a deep analysis of the indexing problem in random mobile WSN is discussed. Furthermore, the challenges of Grid-based indexing are developed and described in the observations form, in order to prepare the way for formulating the research problem.

Universiti Malaya

Accordingly, constructing an indexing structure according to the division of the network area is unsuitable for random mobile WSNs.

This subsection sets out the relationship between each observation and the problem it caused. The summary of the aforementioned observations is as follows: (i) Frequently changing grid during the time period (Observation 3.1). (ii) Required to check the packets from the same sensors from different gateways (Observation 3.2). (iii) Repetitive in packets that transfer from the same sensor (Observation 3.3). (iv) The need to replace the dependability on the packets' attribute to build the index structure (Observation 3.4).

The first observation, in which sensor nodes frequently change the grid during the time period, leads to maximize the indexing overhead by increasing the dependability of the packet attributes. This is because each received packet is anonymous of its source, due to the continuous movement of the sensors' nodes. So, to overcome this problem, the mitigation of the disseminated packets throughout gateways is important. This results in minimizing the number of required packets that need to check the packets' attributes. This is achieved by increased the number of packets that are transferred from the same sensor to the same gateway. In addition, mitigate the number of received packets from different sensors by gateways.

The second observation, if the packets receive the server for indexing is repetitive from the same sensor, it requires from index to read the attributes of packets one by one, which increases the indexing overhead. Furthermore, indexing the packets that are transferred from different sources and from the same time period, and indexing the packets from the same source but different time periods results in repetition in packets that transfer from the same sensor (Observation 3.3). To overcome Observation 3.2 and Observation 3.3, the need to manage and arrange packets in the destinations is required. By achieving this, the received packets by the server in an arranged way minimizes the number of reading packets attributes.

To index random mobile packets, it is required to read the attribute of packets one by one, since the random nature of sensors. Accordingly, the need to replace the dependability on the packets' attribute to build the index structure is required (Observation 3.4) to improve indexing.

According to the aforementioned observations and definitions, the indexing packets of random mobile sensor nodes in WSN is a critical challenge. This leads to the problem that the packets of random mobile sensors that transfer from different sources (sensors) are disseminated throughout destinations (gateways and server), which affects negatively the performance of the indexing. The representation of this problem, as follows:

Problem 3.1: Given a grid g_n is a set of a sensors s_a, and $g_n{}'$ denotes a grid $g_n{}'$ of b sensors $s_b{}'$, from different T_t periods, where $a \cong$ b. Given a difference measure $d(a, f(m))$, determine the function f: $g_n \rightarrow g_n{}'$ that minimize the Sum Square Error (SSE)

$$SSE = \sum_{1=1}^{T_t} d(a, f(m))^2 \tag{1}$$

Where: Sum Square Error (SSE) uses as a measure of variation within a block. If all sensors s_a within a block are identical the SSE would then be equal to 0. More explanation, if the value of SSE is close to zero, this means the grids' changing of random mobile sensors is not too much (the optimum case when SSE =0), as represented in Section 1.3. When the grids' changing of random mobile sensors is low, this indicates less disseminated packets in the server for indexing.

Problem 3.2: For a given destination gw_g or/and server, it contains packets $p_{<ID_{s_a}, T_t, L_l>}$ that transferred from the sensor s_a in a $Path_{s_a} = \{< ID_{s_a}, T_1, L_1 >, < ID_{s_a}, T_2, L_2 >, \dots, < ID_{s_a}, T_t, L_l >\}$, and $p_{<ID_{s_a}, T_t, L_l>}{}'$ that transferred from the sensor $s_a{}'$ (the same sensor but change its location). Given a difference measure $d(a, f(m))$,

Universiti Malaya

determine the function $f: p_{<ID_{s_a},T_t,L_l>} \rightarrow p_{<ID_{s_a},T_t,L_l>}'$ that maximize the Sum Square Error (SSE), as represented in Section 1.3.

Problem 3.3: During the index structure construction, the leaves of parent nodes N_{parent} are frequent changes since the random movement of sensors, which requires registering the current division of Grid-based.

Thus far, the problem statement of this work is formulated. Hence, the need to find the solution to overcome it is required. Accordingly, this work proposes DySta-Coalition model, that divides into (i) Dynamic coalition framework, (ii) Static-Coalition scheme, and (iii) Coalition-Based Index-Tree framework. Dynamic-Coalition framework is responsible for solving Problem 3.1, and Static-Coalition Algorithm is responsible for solving Problem 3.2, and Coalition-Based Index-Tree is responsible for solving Problem 3.3. As a reminder, the existence of disseminated packets that transferred from the same sensor from different grids (locations), which affects adversely on the indexing is Problem 3.1. The need to manage and arrange the transferred packets in the same way of indexing is Problem 3.2. Problem 3.3 summarizes as minimizing the dependability on the attributes of the packet. The proposed model aims to improve indexing procedures in random mobile WSN. As represents in Figure 3.4.

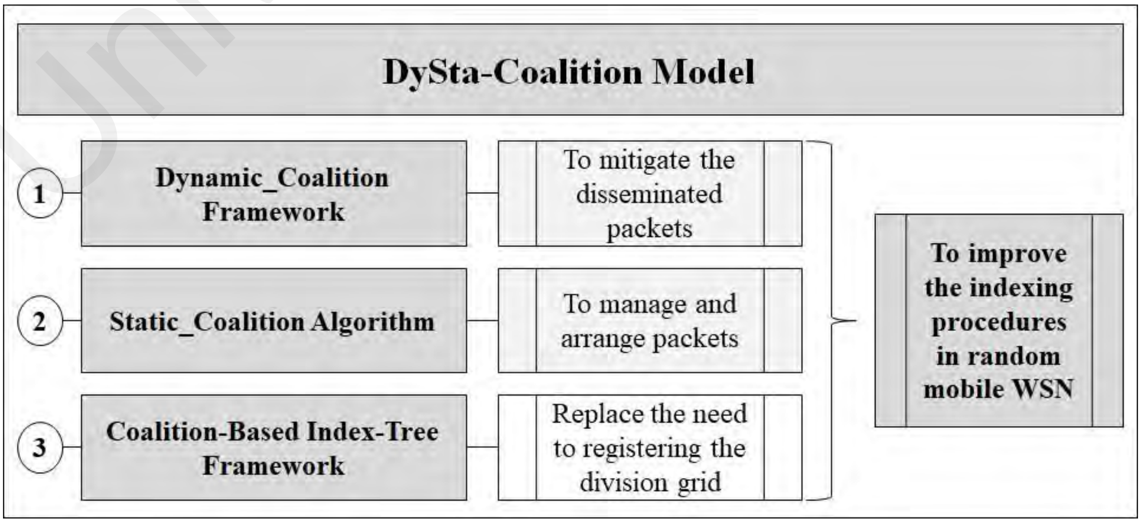

Figure 3.4: DySta-Coalition model and its role to solve the formulated problem

Universiti Malaya

3.3 Empirical analysis

This section presents an empirical analysis, which attempts to highlight the relationship between the random mobility of sensor nodes and indexing performance. Furthermore, this section tries to address the effect of random mobility of sensor nodes on the stability of the grids being constructed. The un-stability of the constructed grids pops-up from the packets that transferred from the same sensor (as source) are disseminated throughout the gateways (as destinations), and each gateway has misarranged packets that belong to the same sensor. Moreover, extending the analysis by modeling a random environment that contains random deployed sensors and gateways. Moreover, establishing a research problem by analyzing the observed output. The next subsections present detailed explanations.

3.3.1 Preliminaries

Before achieving the aim of this section, it is necessary to understand the concept of the coalition. So, the subsequent subsection of this section presents the basic concept of the coalition by defining the terms. This provides a better insight which allows it possible for the user to grasp how this concept was included in the proposed model, as presents in Subsection 3.3.2, Subsection 3.3.3, and Subsection 3.3.4.

3.3.1.1 Coalition Concepts

This section presents the concept of the coalition, which is used to represent the interaction and cooperation among WSNs' nodes, which composes of random mobile sensors, gateways, and a stationary server.

The network area of WSN represents by N, such that (N) is a non-empty set $N_{s+g} = \{1,2, \dots \dots,n\}$ of agents, where agents are the set of players (sensors and gateways (N_{s+g}) in this case). Simply, a coalition is a block that constructs to achieve cooperation between players (agents) within the same coalition (Ye, Zhang, and Sutanto 2013) (Wooldridge 2011). It represents a subset of the players of N. The constructed coalitions are denoted as $C^1, C^2, C^3, \dots$., etc. The grand coalition is the set N of all players, where $C^1 \cup C^2 \cup C^3 = N$. The formal representation of the coalition concept is in the following definitions.

Definition 3.8: A characteristic function game G is given by a pair (N, v), where N is a finite, and non-empty set of agents. Meanwhile, v is a characteristic function, which maps each coalition $C \subseteq N$ to a real number $v(C)$.

The number v(C) is usually referred to as the value of the coalition C. Characteristic function is very important in coalition construction since it has a critical role in the development of sensors and gateways. This is performed by assigning locations of sensors according to gateways, to enable sensors and gateways to be in the same constructed coalition C. Furthermore, the role of $v(C)$ is to assign an intra-relationship, between sensors and gateways, in order to assign them in a coalition.

To construct a coalition, it must consider what form the outcomes of a cooperative game will take. An outcome of a characteristic function game consists of two parts: (i) a partition of network area into coalitions, called a coalition structure, and (ii) a payoff vector, which distributes the value of each coalition among its members (sensors). Formally,

Definition 3.9: Given a characteristic function game $G = (N, v)$, a coalition structure over the network area (N) is a collection of non-empty subsets $CS = \{C1, \dots, Ck\}$, such that $\cup_{j=1}^{k} C^j = N$.

According to Definition 3.8, coalition structure that represented as $G = (N, v)$ means partition of the characteristic function (v) of N into coalitions, where all sensors and gateways that have the perfect value of (v) candidate to be partitioned into a coalition (its name in this work Relevancy-Measure, represents in detail in Section 3.3.3).

Universiti Malaya

Definition 3.10: A vector $x = (x_1, \dots \dots \dots, x_n) \in R^n$ is a payoff vector for a coalition structure $CS = \{C^1, \dots., C^k\}$ over $N = \{1, \dots, n\}$ if (i) $x^i \geq 0$ for all $i \in N$, and $\sum_{i \in C_j} xi \leq v(C^j)$ for any $j \in \{1, \dots, k\}$.

An outcome of G is a pair $(CS, \boldsymbol{x})$, where CS is a coalition structure over G, and x is a payoff vector for CS.

3.3.2 Stability-Metric for coalition formation strategy

This subsection defines the core metric that is used to form coalitions, which called stability-metric. It indicates the stability of sensors to belong to the formed blocks. Accordingly, the blocks result in the network area division. The network area is divided in accordance with the location of the gateways. Since all network population (Definition 3.1) reports information periodically, it is important to be aware of nodes' velocity, position sensing information at any time.

The candidate sensors of each gateway in the formed blocks are determined in accordance with Channel Quality Indicator (CQI). CQI is regarded as a base of Energy Efficient Distributed Receiver (EEDR) routing protocol (Mutthigarahalli Shankarappa and Shankar 2017). EEDR considers a receiver node routing to discover paths between the sender (source) and the receiver (destination) nodes. This condition reduces the transmission of packets in the network. CQI is represented in Equation (2) (Mutthigarahalli Shankarappa and Shankar 2017).

In a network N, assume that clusters (G_C) are constructed from gateways (G), and the existing sensors S in N are belonging to G_C blocks $S_1, S_2, \dots.., S_{G_c} \in R^S$. Each block that shows cluster I is assigned with the measurement $CQI^I: R^S \rightarrow R^{S_I}$. Each measurement of sub-matrix CQI^I represents the measurement in each cluster I (Mutthigarahalli Shankarappa and Shankar 2017).

Universiti Malaya

$$CQI = \left\{ \begin{array}{c|c} 0 & SNIR \leq -16 \\ 16.62 + \frac{SNIR}{1.02} & -16 < SNIR < 14 \\ 30 & 14 \leq SNIR \end{array} \right\} \tag{2}$$

The Signal-to-Noise Interference Ratio (SNIR) is given by

$$SNIR = \frac{\frac{P_T}{L_P}}{N_0 WF} \times P_G \tag{3}$$

Where:

$\mathrm{P_T} = \text{Transmitted Power}$

$\mathrm{L_P} = \text{Path Loss}$

$\mathrm{W} = \text{Signal BandWidth}$

$\mathrm{F} = \text{Noise Figure} = \ 4.8 \times 10^{-2}$(Mann and Singh 2016) (Mutthigarahalli Shankarappa and Shankar 2017)

$\mathrm{N_0}$ is a Thermal noise $= 1.38 * 10^{-23} * 290$ (Meadows 1976)

$\mathrm{P_G} = \text{Processing Gain, given by:}$

$$P_G = \frac{B_s}{R_b} \tag{4}$$

Where:

$\mathrm{B_s} = \text{Spreading Bandwidth}$

$\mathrm{R_b} = \text{Bit\_Rate}$

Each measurement CQI^I is assigned to cluster matrix C, which permutates the entities of the sensors assigned for specific cluster I. This aims to assign each cluster I through its measurement matrix with the permutated sensors S, where S refers to the sensors that belong to C. Hence, multiply C with S, which produces $CS = S^C = \left|S_1^C, S_2^C, \dots \dots, S_{G_C}^C\right|$. Finally, have

$$Y = CQI \times CS \tag{5}$$

Where,

$$\text{CQI}=\begin{pmatrix} CQI^1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & CQI^{Gc} \end{pmatrix} \tag{6}$$

The resulting matrix $CQI\ CS$ represents the distribution of sensor nodes (Y) inside each cluster, and CS is a cluster matrix.

In a practical environment, it is considered that sensors become candidates of cluster CS, when each sensor has the maximum value of CQI depending on gateways. From Equation (5) and the sparse representation of the sensors, obtain

$$Y = CQI \times CS = CQI \tag{7}$$

The CQI in Equation (7) represents the maximum value of the CQI of the sensors.

In addition to CQI, the relative velocity metric (RVM) and relative position metric (RPM) are also used. Gateway i and sensor j are assumed to be in the same cluster, and V_i and V_j are the velocities of gateway i and sensor j, respectively. RVM is employed to indicate the relative mobility of gateway i as

$$\text{RVM(i)} = \frac{1}{(n-1)} \sum_{k=0}^{n} \left| V_i - V_j \right| \tag{8}$$

Similarly, let (X_i, Y_i) and (X_j, Y_j) represent the positions of gateway i and sensor j, respectively. The RPM of gateway i is calculated as follows:

$$RPM(i) = \frac{1}{n-1} \sum_{i \neq j} \left((Xi - Xj)^2 + (Yi - Yj)^2\right)^{\frac{1}{2}} \tag{9}$$

Hence, a sensor with a small RVM is likely to stay in its current cluster for a long time, whereas a sensor with a small RPM can have a good transmission quality with the gateway in its cluster because of the short communication distance.

As proposed in (Huo et al. 2016), a stability-metric is defined to indicate the stability of a node in one cluster

$$M(i) = k1 * RVM(i) + k2 * RPM(i) \tag{10}$$

where $k1$ and $k2$ are the weights of RVM and RPM, respectively. The significance of k1 and k2 values lies in obtaining a similar influence using the position and velocity between gateways and sensors. Excellent clustering performance is achieved by selecting the appropriate values of $k1$ and $k2$.

The preceding analysis indicates that small $M(i)$ indicates that i is highly stable with its formed cluster. Equation (10) describes as a stability-metric that has been relied upon to propose relevancy-measure, as elucidated in the next subsection.

3.3.3 Relevancy-Measure

So far, $M(i)$, that represented as Equation 9 in Subsection 3.3.2, is a basic equation used as a metric to construct a stable coalition structure in the current works, i.e. (Huo et al. 2016). As noted, $M(i)$ depends on velocity (RVM) and position (RPM) of mobile sensors. This dependability is not beneficial for random mobility WSNs, as detailed in Problem 3.1, Problem 3.2, and Problem 3.3. Therefore, the need to minimize dependability on the position and velocity is required. To fulfill that, this subsection defines a relevancy-measure, which is a metric used to evaluate the efficiency of dynamic-coalitions formation.

From $M(i)$, a coalition with a small stability-metric indicates that the nodes in it have similar mobility and a high resource utility rate. A coalition with a head h and n members is assumed to exist. The velocity, position, and efficiency functions will be used to represent the coalition utility. The three important sub-functions are stability-related, velocity, and efficiency-related functions.

The stability-related function of a block is defined as the distance and relative velocity among the nodes in the same block. In other words, a stable block can be formed if the node's velocity and position are similar. Therefore, the definitions of the velocity and position functions are as follows:

$$V_{hCoalition} = \frac{\sum_{n=1}^{n} RVM(i)}{n} \tag{11}$$

The position function is defined as follows:

$$P_{hCoalition} = \frac{\sum_{n=1}^{n} PVM(i)}{n} \tag{12}$$

The bandwidth of nodes may differ due to the random mobile movement of the network nodes. However, the stability-metric (Equation (10)) does not introduce bandwidth. This limitation will result in congestion and the degradation of communication quality in the network. The bandwidth advantages must be utilized by introducing another function called efficiency-related function. The efficiency-related function is given by

$$E_{hCoalition} = \frac{\sum_{n=1}^{n} bandwidth(i)}{n} \tag{13}$$

The stability and efficiency metrics are considered, and the stability-metric (Equation [10]) is defined as

$$U_{hCoalition} = k_1 * V_{hCoalition} + k_2 * P_{hCoalition} + a * E_{hCoalition} \tag{14}$$

where a is an adjustment factor, which represents different transmission efficiency classifications.

On the basis of Equations (2) and (3), the bandwidth is

$$BanW = \frac{\left(\frac{P_T}{L_P}\right) * P_G}{N_0 \, F \, (CQI - 16.62)} \tag{15}$$

Equation (15) is substituted with $U_{hCoalition}$

$$U_{hcoalition} = k_1 * V_{hcluster} + k_2 * P_{hcluster} + a * \frac{\left(\frac{P_T}{L_P}\right) * P_G}{N_0 \, F \, (CQI - 16.62)} \tag{16}$$

Universiti Malaya

CQI (Equation (2)) enables packet transmission without the need to base on the velocity and location. Hence, they were not taken into consideration. The resulting equation is

$$U_{hcoaltition} = a * \frac{\left(\frac{P_T}{L_P}\right) * Bs}{N_0\, F\, R_b((1.02)CQI - 16.9524)} \tag{17}$$

where a is an adjustment factor, and $a \neq 0$. The optimum value of the adjustment factor (a) described in Subsection 3.3.4.1. Equation (17) represents a relevancy-measure, which defines responsibility for making decisions to join the sensors to dynamic-coalitions. The relevancy-measure is defined as the degree of stability of each sensor in its dynamic-coalition. Accordingly, relevancy-measure can utilize the bandwidth among the members of each dynamic-coalition. Proving that the proposed relevancy-measure (Equation [17]) is more superior compared with stability-metric ((Equation [10])) presented in Subsection 3.3.4.2.

In sum, a relevancy-measure is used as a metric to achieve a stable relationship between sensors and gateways. The decisions affected by the relevancy-measure enable the sensors to transfer packets to the optimum gateway, which increases the number of packets received by the gateway from the same sensors to minimize the disseminated packets throughout destinations. The performance of relevancy-measure (Equation [17]) is evaluated in the next subsection.

3.3.4 Numerical Evaluations

Here, relevancy-measure performance is evaluated in accordance with the coalition-stability metric perspectives. To achieve that, it is important to select the optimum adjustment factor (a) value, presented in Equation 16, Subsection 3.3.3. The optimum adjustment factor (a) value helps to achieve excellent coalition constructing performance. Furthermore, the experimental setup is presented, in order to achieve this evaluation. The details as the following.

Universiti Malaya

3.3.4.1 Experimental setup

In addition to the experiments setup mentions in Section 4.1, the optimum adjustment factor (a) is selected on the basis of the following scenario. The sensors and gateways are randomly deployed. The number of random mobile sensors changes in the set {100, 200, 300, 400, 500, 600, 700, 800, 900}. The number of gateways is 10. The sensor and gateway velocity are selected randomly between 0 and 15 m/s. The kind of packets used in the text data of 50Kb and the data-rate is 500 Kbps. The communication radius (the maximum possible distance between any two nodes) is 100 m. Transmitting and receiving power consumption rates of the sensors are 21 mW and 15 mW. The aforementioned parameters are summarized in Table 3.1. This evaluation is implemented using a simulator developed by Visual C# 2017 (Ghaleb et al. 2017).

Table 3.2: Summary of parameters

#	Parameters	Values
1	Sensors' number	{100, 200, 300, 400, 500, 600, 700, 800, 900}
2	Gateways' number	10
3	Velocity	0-15m/s
4	Size of text data packets	50 KB
5	Data-Rate	500Kbps
6	Radius	100m
7	Transmitting and receiving power consumption	21 mW and 15 mW

The constructed coalition performance is evaluated using a coalition stability-metric. The coalition stability-metric represents the average number of the changes in the constructed coalitions per second. Coalition stability is counted in accordance with the number of coalition head changes. The constructed coalitions will be unstable if the value is large.

3.3.4.2 Selection of the optimum value of adjustment factor (a)

To select the optimal value of (a), the evaluation starts with the value = 1.5, as in (Huo et al. 2018). The value is kept minimizing until achieving the optimal one. Figure 3.4

Universiti Malaya

represents the average number of coalition head changes in accordance with the number of sensors. The comparisons in the figure are based on = 0.05, 0.1, 0.025. The judgment of the optimal (a) value is based on two factors. The suitability to a large number of packets and the number of coalition changes is low.

The average number of the coalition is changing per second in Figure 3.5 to illustrate coalition stability. The cost of poor stability is compared with a=0.1 and 0.025 to achieve efficient coalition communication. As for the proposed scenario, the stability of a=0.05 is the case due to the great contributors to efficiency. These numerical evaluations and mentioned analyses help to conclude that the proposed strategy performs well on stability if the adjustment factor is 0.05. By looking at the figure, it is obvious that the curve for a=0.050 goes down at a higher load when the number of sensors is 800 and 900. This indicates the suitability of Equation 16 to the higher value. More explanation, the average number of constructed coalitions changing per second is the lowest when the number of sensors is the highest.

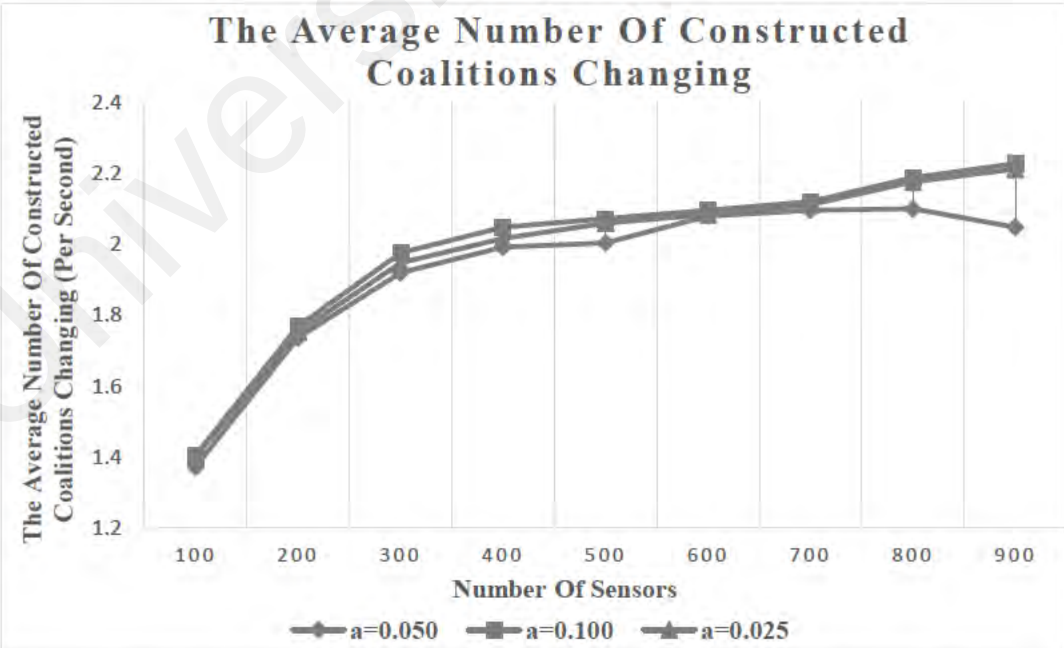

Figure 3.5: Average number of coalitions changing to select the optimum adjustment factor value

3.3.4.3 Proving the superiority of relevancy_measure

As mentioned, the performance of the coalition was assessed by coalition-stability. Coalition-stability is represented by the average number of coalitions changing per second. Coalition changing counted according to the coalition-head changing number. Obviously, the constructed coalitions will not be stable if the value of coalition-stability is large. The coalition-stability metric is used to prove that relevancy-measure (Equation 17) is more superior than the stability-metric (Equation 10).

Figure 3.6 represents the average number of constructed coalitions changing, to prove the superiority of relevancy-measure. The comparison was conducted according to the changing number of constructed coalitions through the number of sensors. Number of sensors ranges in {100,200,...,900}. As in the figure, the change in relevancy-measure values according to the number of sensors is approximately the same. This indicates its suitability to random mobile sensors, whatever its number. Meanwhile, stability-metric values increase according to the number of sensors. Therefore, the figure indicates that relevancy-measure is better than the stability-metrics, because of two reasons. First, the difference between values is small compared with the same metric. I.e. the difference between 100 and 200 of the relevancy-measure is minimum than the difference between 100 and 200 of the stability-metrics. Second, the superiority of the relevancy-measure because it has smaller values.

Universiti Malaya

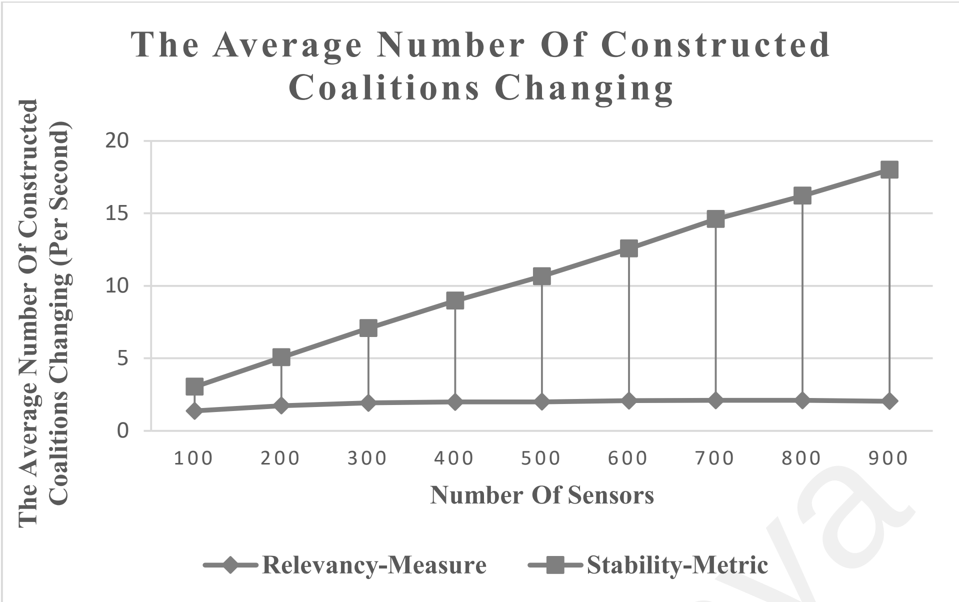

Figure 3.6: Average number of coalition changes to prove the superiority of relevancy-measure

3.4 Overview of the proposed model

This section represents a general overview of the proposed DySta-Coalition model and its parts. The next sections describe each part of the proposed model and its role to solve the aforementioned problems.

A general overview of the proposed model depicts in Figure 3.7. The proposed model called DySta-Coalition that exploits the existence of mobile sensors, gateways, and stationary server for improving the indexing. The proposed model aims to minimize the management and arrange packets to improve the indexing process in random WSN. Accordingly, the overall structure of DySta-Coalition model composes of the following: (i) Dynamic-Coalition framework, (ii) Static-Coalition algorithm, and (iii) Coalition-Based Index-Tree framework, which implements in sensors, gateways, and server, respectively.

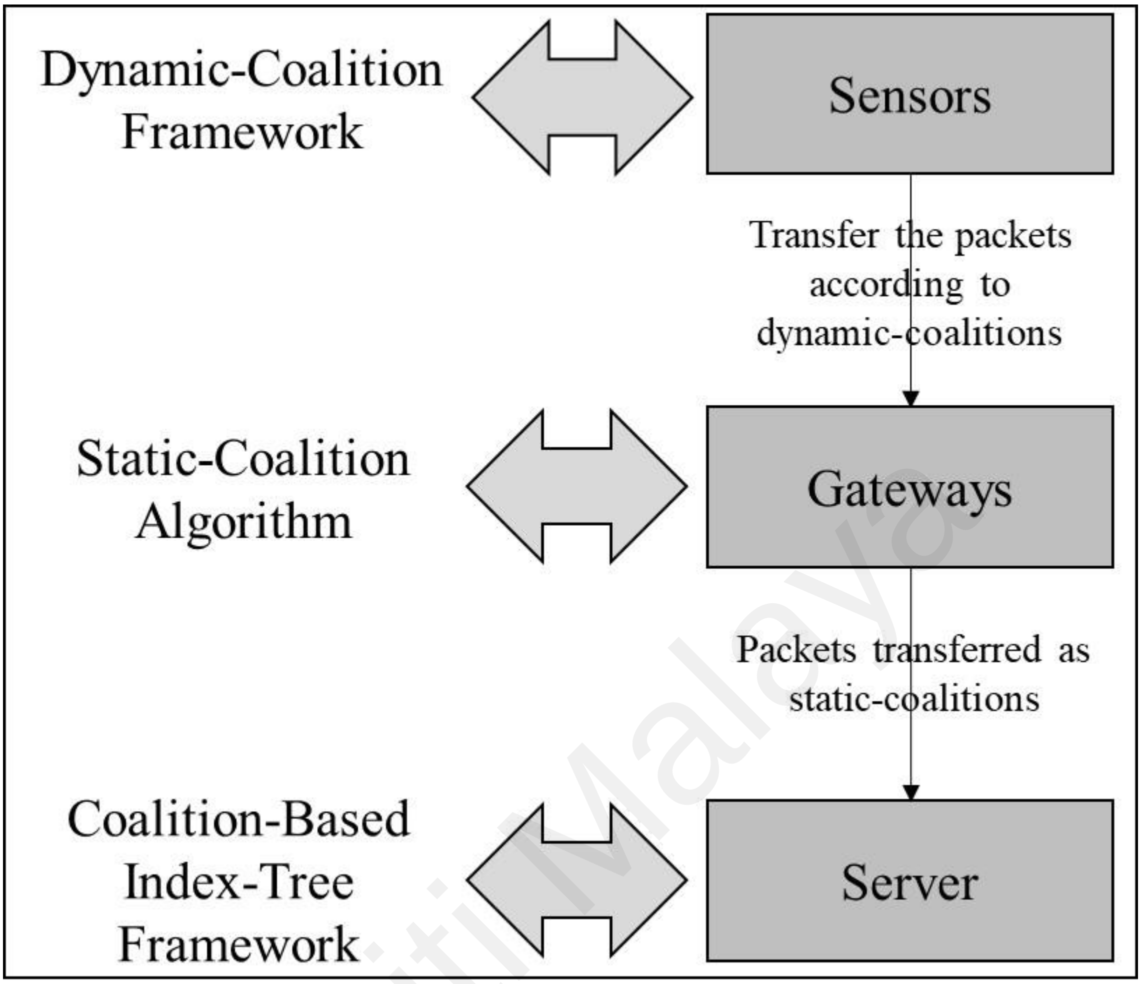

Figure 3.7: General overview of DySta-Coalition model

The first part of the proposed model is Dynamic-Coalition framework. It aims to minimize the number of disseminated packets throughout gateways, by constructing dynamic blocks, called dynamic-coalitions. Dynamic-Coalition indicates that the blocks are keeping constructed as long as sensors and gateways move. Furthermore, by implementing Dynamic-Coalition framework, the number of packets that belong to the different sensors, which are received by gateways will be alleviated. The alleviation of packets means reducing the number of transferred packets from different sensors to a specific gateway as can as possible, in order to alleviate the packets' dissemination. More explicitly, increasing the number of receiving packets that are transferred from the same sensor in each gateway, affects positively improving the indexing. Accordingly, dynamic-coalitions reduce the index dependability on the attributes of packets, the need for periodically registering the current group area information of each sensor, and the need to allocate destination for each sensor. Thus, the goal of dynamic-coalitions is to compromise between the increasing number of transmitted packets to each gateway and the number of disseminated packets in each one, by using the proposed equation (Equation 16) presented in Section 3.3.3.

The second part of the proposed model is Static-Coalition algorithm, which constructs in each gateway. It aims to prepare the received packets of random mobile sensors for indexing, by arranging them together as blocks called static-coalition. Based on constructed dynamic-coalitions, the packets are received by the appropriate gateways. Then, an intra-relevancy allocates between each packet according to its location. Accordingly, the steps of initiating coalitions inside each gateway are started. Static-Coalition algorithm rearranges the packets in each gateway based on the sensors' source. In other words, it arranges the packets step-by-step before they accumulated in the main server for indexing. Accordingly, Static-Coalition algorithm represents the initial arranging of packets to be suitable for indexing.

The third part of the proposed model is Coalition-Based Index-Tree framework. It is responsible for collecting packets that transfer from gateways for indexing. Hence, Coalition-Based Index-Tree is an index-tree structure, with no inner-nodes, dedicated to index packets of random mobile sensors. It constructs nodes based on a coalition, describes in detail in Section 3.7. Since each packet belongs to the sensor, the nodes of the index construct bases on the number of sensors, and the packets will add to each node based on the sensors' source.

The interaction among major components of DySta-Coalition model illustrates in Figure 3.8, as follows:

Universiti Malaya

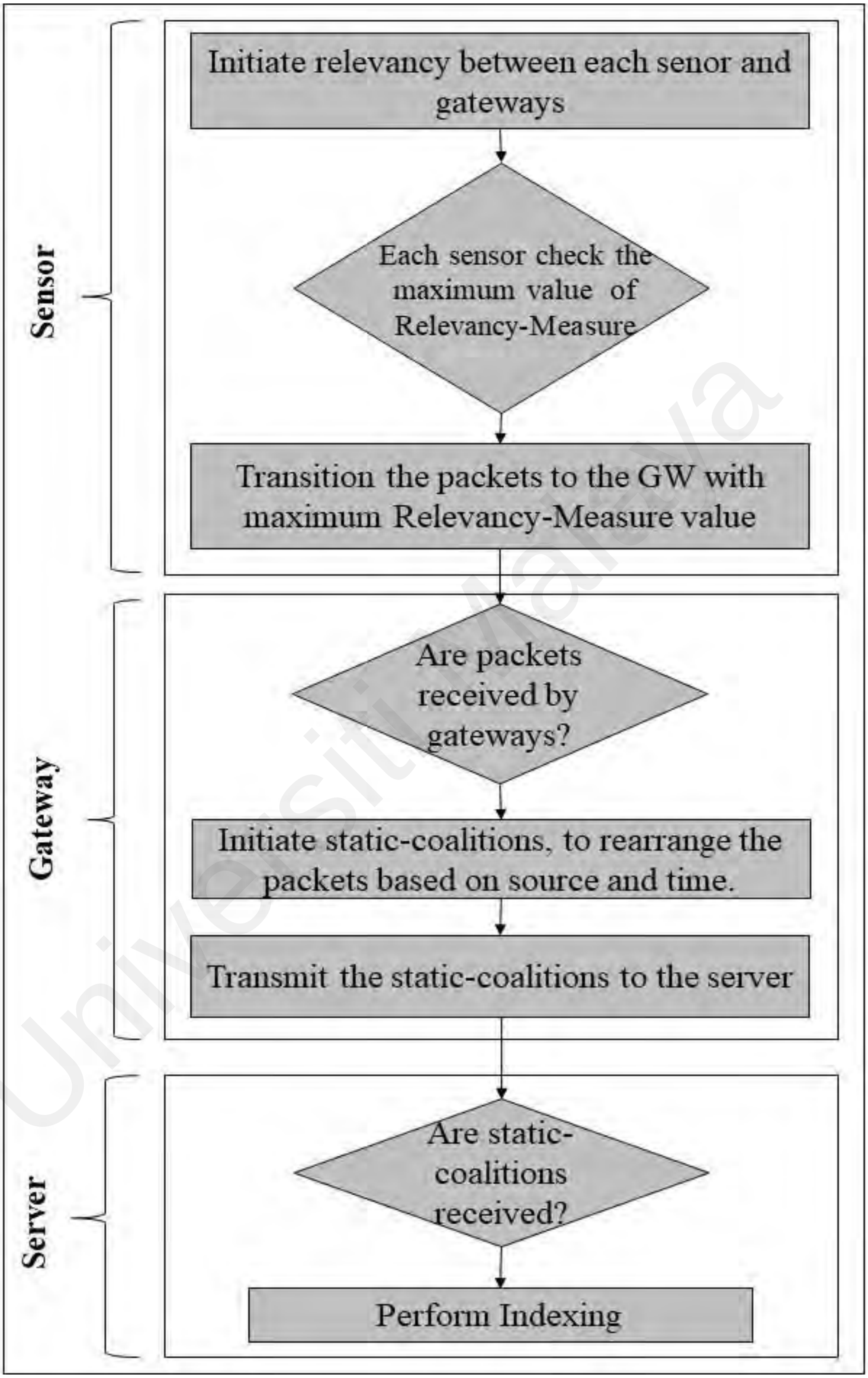

Figure 3.8: Interaction among major components of the proposed model

Phase 1: Dynamic-Coalition framework, which conducted in sensors:

Dynamic-Coalition framework aims to minimize the number of disseminated packets throughout gateways, by replacing Grid-based indexing (mentioned in Chapter 2, to index random mobile sensors) with the coalition. Formed coalitions in this phase construct dynamically, since the random mobile nature of sensors. Therefore, the constructed coalitions called dynamic-coalitions. Dynamic-Coalition framework describes in detail in Section 3.5. The steps of this phase are briefly explained as follows.

Step 1: each gateway checks the candidate sensors.

This step is preliminary for establishing dynamic-coalitions. In this step, the gateways (destinations) ensures which sensors (sources) are able to transfer packets to it. This is performed by establishing a characteristic function (Definition 3.8). The characteristics function used in this case is Channel Quality Indicator (CQI) (Mutthigarahalli Shankarappa and Shankar 2017), represented in subsection 3.3.2. Accordingly, an intra-relationship between gateways and sensors is established. As a result of this phase, each gateway has a set of candidate sensors that the gateways are able to receive packet.

According to Definition 3.8, the formal representation of this phase, as follows:

Given a characteristic function game $G = (N, CQI)$, a coalition structure over the network area (N) is a collection of non-empty subsets, as follows:

$$CS_{gw_1} = \{C_{S_1}, C_{S_2}, C_{S_3} \dots, C_{S_s}\}$$

$$CS_{gw_2} = \{C_{S_3}, C_{S_2}, C_{S_4} \dots, C_{S_s}\}$$

.

.

.

$$CS_{gw_g} = \{C_{S_5}, C_{S_6}, C_{S_7} \dots, C_{S_s}\}$$

Step 2: Each sensor calculates relevancy-measure (payoff vector) for all current gateways.

For each constructed subsets CS_{gw_g}, Dynamic-Coalition framework calculates a payoff vector between sensors and gateways. The payoff is represented here as relevancy-measure, which evaluates the coalition constructing performance. In other words, if a specific sensor in CS_{gw_g} has a small value of relevancy-measure, which means this sensor is more stable to be a member of this coalition than others.

According to Definition 3.9 and Definition 3.10, the formal representation of Step 2 as follows:

For each subset CS_{gw_g}, calculate payoff vector x, where $x = (x_1, \dots \dots \dots, x_n) \in R^n$, as follows:

$$x_{C_{S_1}} = (x_{payoff_1}, x_{payoff_2}, \dots \dots, x_{payoff_{C_{S_1}}})$$

$$x_{C_{S_2}} = (x_{payoff_3}, x_{payoff_4}, \dots \dots, x_{payoff_{C_{S_2}}})$$

$$x_{C_{S_3}} = (x_{payoff_5}, x_{payoff_6}, \dots \dots, x_{payoff_{C_{S_3}}})$$

.

.

.

$$x_{C_{S_s}} = (x_{payoff_7}, x_{payoff_8}, \dots \dots, x_{payoff_{C_{S_s}}})$$

$$x_{C_{S_3}} = (x_{payoff_1}, x_{payoff_2}, \dots \dots, x_{payoff_{C_{S_3}}})$$

$$x_{C_{S_4}} = (x_{payoff_3}, x_{payoff_4}, \dots \dots, x_{payoff_{C_{S_4}}})$$

$$x_{C_{S_5}} = (x_{payoff_5}, x_{payoff_6}, \dots \dots, x_{payoff_{C_{S_5}}})$$

.

.

Universiti Malaya

.

$$x_{C_{S_S}} = (x_{payoff_7}, x_{payoff_8}, \ldots\ldots, x_{payoff_{C_{S_S}}})$$

.

.

.

$$x_{C_{S_5}} = (x_{payoff_1}, x_{payoff_2}, \ldots\ldots, x_{payoff_{C_{S_5}}})$$

$$x_{C_{S_6}} = (x_{payoff_3}, x_{payoff_4}, \ldots\ldots, x_{payoff_{C_{S_6}}})$$

$$x_{C_{S_7}} = (x_{payoff_5}, x_{payoff_6}, \ldots\ldots, x_{payoff_{C_{S_7}}})$$

.

.

.

$$x_{C_{S_S}} = (x_{payoff_7}, x_{payoff_8}, \ldots\ldots, x_{payoff_{C_{S_S}}})$$

Step 3: Each sensor decides to find the minimum value of the payoff vector (relevancy-measure).

According to the calculated values of relevancy-measure, each sensor allocates the best coalition to belong to according to the minimum value of the relevancy-measure. For example, in the set

$$x_{C_{S_2}} = (x_{payoff_3}, x_{payoff_4}, \ldots\ldots, x_{payoff_{C_{S_2}}})$$

Suppose the payoff vector $x_{payoff_4}$is the smallest value, this means the sensor C_{S_2} is a member of coalition CS_{gw_1}.

Step 4: Each sensor in the constructed coalitions transfers packets to the gateway.

Universiti Malaya

After the sensor become a member of a coalition, it has the ability to transfer packets to the gateway in the same coalition. This improves the inter-relationship between sensors and gateway located in the same coalition.

Phase 2: Static-Coalition algorithm, which conducts in each gateway.

Based on the constructed dynamic-coalitions, which extract from the previous phase, the packets receive to the appropriate gateways. Furthermore, an intra-relevancy between each packet is initiated according to its location by Static-Coalition algorithm. Accordingly, the steps of initiating static-coalitions inside each gateway are started, as follows.

Step 1: In each gateway, the static-coalitions are initiated.

As a start point of this step, each gateway initiates its static-coalition, to be ready for receiving packets. Accordingly, when the packets are transferred from sensors to gateways, the packets are ready for adding to the appropriate location of static-coalition, based on the source of each packet. It is important to mention that the Static-Coalition algorithm is preliminary indexing. Hence, the formal representation of a coalition in this phase is similar to the representation of Coalition-Based Index-Tree framework.

Step 2: Each gateway transfers static-coalitions to the server.

Each gateway transfers its static-coalitions when its buffer becomes full.

Phase 3: Coalition-Based Index-Tree framework, which conducts in the server.

The indexing process is started based on the received static-coalition, the details in the next step.

Step 1: Perform indexing.

Coalition-Based Index-Tree is dedicated to random mobile sensors in WSNs with no inner-nodes. It bases on managing and controlling the random mobile sensors packets, to be fitting for random mobile sensors. The nodes of Coalition-Based Index-Tree structure constructs base on the number of sensors. The number of constructed coalitions is equal

Universiti Malaya

to the number of sensors in the network. The division is based on the dynamic grouping of sensors, where each coalition has the same volume. Each constructed node will include the packets that belong to the same sensor.

To understand deeply the relationship between the coalition and the construction of the coalition-based index tree structure, let us see the following example. Suppose the server receives the following static-coalitions from gateways:

$$Static\_coalition_1 = (P_{S_1}(L_1, T_1), P_{S_1}(L_2, T_1), P_{S_2}(L_3, T_1), P_{S_2}(L_4, T_2), P_{S_1}(L_1, T_4))$$

$$Static\_coalition_2 = (P_{S_2}(L_1, T_1), P_{S_3}(L_2, T_1), P_{S_3}(L_3, T_1), P_{S_4}(L_4, T_2), P_{S_4}(L_1, T_4))$$

$$Static\_coalition_3 = (P_{S_1}(L_5, T_4), P_{S_1}(L_6, T_5), P_{S_5}(L_3, T_1), P_{S_2}(L_4, T_2), P_{S_4}(L_5, T_6))$$

$$Static\_coalition_4 = (P_{S_5}(L_6, T_1), P_{S_4}(L_2, T_1), P_{S_2}(L_6, T_7), P_{S_2}(L_5, T_2), P_{S_1}(L_7, T_9))$$

Where: P is a packet, S is a sensor, L is a transferred location, T is a transferring time, and $P_{S_1}(L_1, T_1)$ is a packet that transfers from sensor one at location $L_1$and time T_1.

According to the aforementioned example, the role of the coalition is to arrange the received packets according to sources (sensors). This is performed by forming subsets and each subset labeled as the source. Furthermore, each constructed coalition consider as a node of the index nodes.

The subsets that form according to characteristics functions are not represented here, because according to the static-coalition the subsets are ready, and no need to re-represent them according to characteristic function.

Now, the constructed coalitions base on the payoff vector. The payoff vector is represented as a binary function in this phase, represents in detail in Subsection 3.7.1.1. Binary function determines if this packet belongs to this node or not. Accordingly, coalition-based index tree framework reads each packet from the received static-coalition and builds a coalition according to its source using the binary function (payoff vector). The resulted coalition is as follows:

$$CS_{S_1} = \{P_{S_1}(L_1, T_1), P_{S_1}(L_2, T_1), P_{S_1}(L_1, T_4), P_{S_1}(L_5, T_4), P_{S_1}(L_6, T_5), P_{S_1}(L_7, T_9)\}$$

$$CS_{S_2} = \{P_{S_2}(L_3, T_1), P_{S_2}(L_4, T_2), P_{S_2}(L_1, T_1), P_{S_2}(L_4, T_2), P_{S_2}(L_6, T_7), P_{S_2}(L_5, T_2)\}$$

$$CS_{S_3} = \{P_{S_3}(L_2, T_1), P_{S_3}(L_3, T_1)\}$$

$$CS_{S_4} = \{P_{S_4}(L_2, T_1), P_{S_4}(L_4, T_2), P_{S_4}(L_1, T_4), P_{S_4}(L_5, T_6)\}$$

$$CS_{S_5} = \{P_{S_5}(L_3, T_1), P_{S_5}(L_6, T_1)\}$$

Each constructed coalition converts into the node in the coalition-based index tree structure. This step will be deeply described in Section 3.7.

So far, a full analysis and demonstration of the problem statement is introduced. Furthermore, a brief description of DySta_Coalition model and its parts are presented. Accordingly, the subsequent sections describe in detail the full description of the proposed model and the role of each part of it to solve the mentioned problems. More specifically, Section 3.5 discusses Dynamic_Coalition framework and its role to mitigate the disseminated packets throughout destinations (Problem 3.1). Section 3.6 presents the role of Static-Coalition algorithm to arrange and manage the packets in gateways (Problem 3.2). As well, in Section 3.7, the Coalition_Based Index_Tree framework is presented and mentions its ability to replace the dependability on division of the network area (Problem 3.3).

3.5 Framework for mitigating the disseminated packets throughout destinations

The indexing procedures are highly affected by the number of random mobile sensors' nodes in WSN. Thus, the number of transferred packets to destinations increases. In such a case, the disseminated packets throughout destinations are increased during the network lifetime. However, in the existence of the disseminated packets, the performance of indexing is negatively affected, since it requires to read the received packets from different sources, in order to build or update the index structure.

Universiti Malaya

To address this problem, this thesis presents s framework, called Dynamic-Coalition. It exploits the existence of sensors to improve indexing. Dynamic-Coalition framework is a coalition-based (Ye, Zhang, and Sutanto 2013) (Masoum, Meratnia, and Havinga 2018), which replaces the division of the network area into grids (Definition 3.7) with constructed coalitions. Dynamic-Coalition framework assembles random mobile sensors and gateways based on the coalition by forming blocks, called dynamic-coalitions. The benefit of dynamic-coalition is to establish a relationship between sensors and gateways, to alleviate the disseminated packets. This is achieved by allowing the sensors to transfer packets to the gateway that belongs to the same dynamic-coalition. Relevancy-measure calculated according to equation (16), represented in Subsection 3.3.3. Namely, relevancy-measure is denoted the degree of stability of each sensor in its formed dynamic-coalition. Thereby, relevancy-measure can achieve the full use of the bandwidth among members of each dynamic-coalition. The dynamic-coalitions are forming according to the location of each gateway, and members represented as sensors. More explanation, each dynamic-coalition consists of one gateway, considers as coalition-head and its sensors, consider as candidate-members.

Forming dynamic-coalitions require following a set of steps. Figure 3.9 represents Dynamic-Coalition framework. It has a set of steps. First, the gateways allocate the current locations of the sensors, in accordance with each gateway to initiate a relevance between each sensor and gateways using CQI (Mutthigarahalli Shankarappa and Shankar 2017). In this step, each sensor is considered a candidate for more than one gateway. Each gateway has a list of candidate sensors, represents in detail in Subsection 3.5.1. Second, each gateway identifies relevancy-measure for each candidate sensor. The relevancy-measure evaluates the efficiency of the formation process of dynamic-coalitions and indicates the degree of stability of the sensors in dynamic-coalitions. Third, in accordance with relevancy-measure, each sensor has an indication to associate with blocks. In other words, the sensors with a small relevancy-measure indicate that they are a stable member of this dynamic-coalition. The sensors that belong to a specific dynamic-coalition are called candidate sensors. When the sensors belong to a dynamic-coalition, the status must change from free to busy. They are busy if the sensors belong to a certain dynamic-coalition (candidate sensors), otherwise free. Fourth, each sensor is motivated to transfer packets to the gateway that belongs to the same dynamic-coalition. Finally, each gateway transfers packets to the server. Hence, the indexing process starts on the server, that describes in detail in Section 3.7. The formed dynamic-coalitions are changed dynamically on the basis of gateway movement. In other words, the mentioned steps keep iterating until the network lifetime is finished.

As a result, Dynamic-Coalition framework aims to make the best trade-off between received and disseminated packets. In other words, a trade-off between increasing the numbers of received packets by gateways and alleviating the number of disseminated packets throughout the gateways. To do so, a relevancy-measure is introduced to evaluate the efficiency of the coalition formation process.

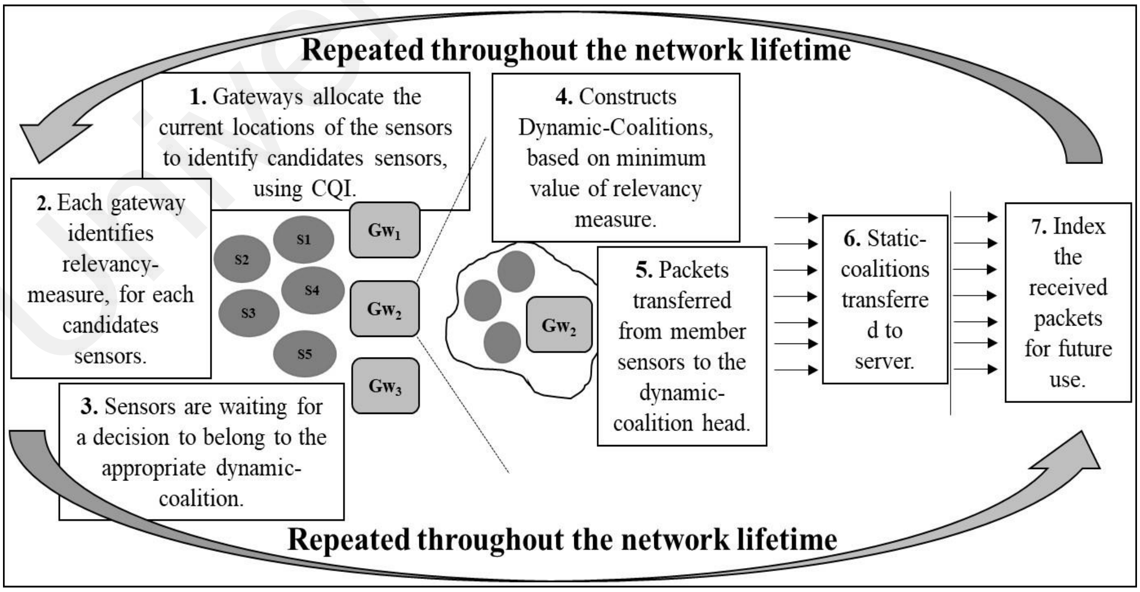

Figure 3.9: Dynamic-Coalition Framework

3.5.1 Coalition-head and Coalition-Members elections

Dynamic-coalitions are constructed according to the location of each gateway at a specific time, where each gateway considers as a coalition-head. The gateways consider as a core node that the dynamic-coalition is based on to formulate. Since there exist many gateways that satisfy to be a coalition-head, dynamic-coalition has to choose one optimal one to be treated as a coalition-head, by relevancy-measure.

Relevancy-measure indicates the stability of sensors in the formed coalition, represented by equation (16). Coalition-members are sensors based on relevancy-measure to determine if they belong to the formed dynamic-coalitions. The gateway that elected as coalition-head has the lowest value of relevancy-measure. As mentioned before, dynamic-coalitions are keeping constructed as long as sensors and gateways are moving. Accordingly, it is usual to change the coalition-head frequently. When the coalition-head wants to reform the dynamic-coalition, it recalculates relevancy-measure.

The details of the dynamic-coalition formation are shown in Algorithm 3.1. It starts by defining gateways and sensors, represented by $GW \text{ and } S$, respectively. Gateways represent the coalition head, and sensors are the members of each gateway. From these definitions, line 3 defines the coalitions, labeled by the gateway name that is represented by G_C. Line 4 defines the candidature set C_k, which represents the gateways and the candidature sensors by CQI. Line 5 starts a For-loop by determining all gateways elected as a coalition head. For-loop counter starts from 0 to the number of gateways elected as a coalition head, which is represented as $(G_C - GW)$. Line 6 to line 8 describe the way to allocate the sensors as a candidate for the gateway based on CQI. Each candidate sensor in C_k calculates a relevancy-measure to ensure the stability of that sensor to belong to a dynamic-coalition as a member. Based on the values of a relevancy-measure, line 9, the negotiation to insert sensor S_{ij} to G_C as a coalition member is represented in line 10.

Algorithm 3.1: Gateways Coalition Formation

Universiti Malaya

Define $GW = gw_1, gw_2, \ldots, gw_n$ as a set of all current Gateways

Define $S = s_{ij}$ as a set of all current Sensors

Define G_C coalitions

Define C_k, by CQI [Equation (1)]

$\boldsymbol{FOR}\ (I = 1; I <= (G_C - GW); I++)\ \boldsymbol{do}$

    $\boldsymbol{FOR}\ (R = 1; R <= |S|; R++)\ \boldsymbol{do}$

        $Find\ the\ position\ of\ s_{ij}; i \in C_k, j \in GW$

    $\boldsymbol{END\ FOR}$

    $Calculate\ relevancy - measure\ U(S_{ij}, GW_C)$ by equation (16)

    $Negotiate(S_{ij}, GW_C)$

$\boldsymbol{END\ FOR}$

3.5.2 Packets transmission inside coalitions

It is logically recalled that the sensors and coalition-head that set within the same dynamic-coalition fall within the same bandwidth region. Accordingly, instead of dissemination packets of sensors throughout gateways in the constructed grids, the proposed framework forms the blocks that enabled the transferring the packets from sensors to the gateway, which belong to the same dynamic-coalitions. Furthermore, since dynamic-coalition is designed for random movement, it does not base on the location of each sensor to alleviate the need for registering the current group area information of each sensor, and the need to allocate a destination for each sensor.

To this end, the main benefit of Dynamic-Coalition framework is initiating blocks without relying on the attributes of the packets, i.e. locations and time. Furthermore, Dynamic-Coalition framework assumes that the network region is categorized into dynamic-coalitions, instead of dividing the network area into grids, as in current works (Jeongcheol Lee et al. 2018).

Universiti Malaya

3.6 Proposed algorithm for managing and arranging packets in the destinations

When the packets are received by different gateways (destinations), each gateway contains packets that belong to the same source, without any arranges. In such a case, the indexing performance is affected, due to the need to check the source of each packet many times in each destination. Accordingly, the need to arrange and manage packets in each gateway is required, before the final indexing starts.

To address this problem a Static-Coalition algorithm is proposed. Figure 3.10 represents the overview of Static-Coalition algorithm. As an initial step, each gateway initiates its static-coalition, in order to be ready for receiving packets. Accordingly, when the packets are transferred from sensors to gateways, the packets are ready for adding to the appropriate static-coalition based on relevancy-measure, based on the source of each packet. After that, determining the appropriate location of each packet in static-coalition based on its location. Finally, rearrange the packets in each node based on transferring time. The gateway transfers its static-coalition to the server for indexing. Finally, the indexing process starts based on the received static-coalition. The indexing phase will describe in detail in Section 3.7.

Static-coalition plays an important role to initiate Coalition-Based Index-Tree. The significance of static-coalition comes from the ability to rearrange the transferred sensor's packets in the gateway to minimize indexing overhead. Furthermore, it arranges the packets in the same way as Coalition-Based Index-Tree builds its structure. Static-coalition consists of sensor Id (S_{id}) that represents the source of packets, transferring time (Time), and the location of the sensors (Location) when transferred the data. Each gateway transfers the static-coalition while the buffer becomes full. I.e., suppose WSN has five gateways and its lifetime is 30 seconds and the period of the time slot is two. This means there are (30/2)*5 number of dynamic-coalitions that will arrive at the server, instead of (30*2)*5.

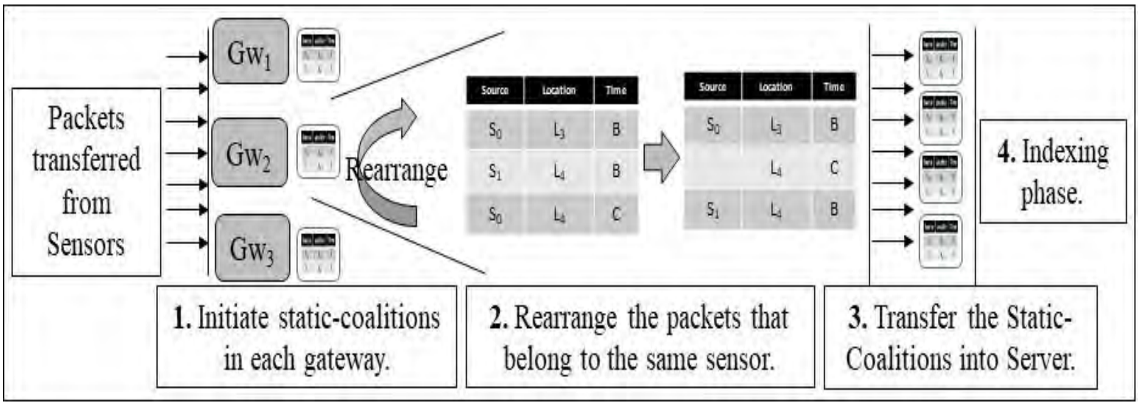

Figure 3.10: The overview of Static-Coalition algorithm

So far, the gateways contain packets that belong to the same source with different locations and time. Therefore, the role of static-coalition appears, it arranges the packets that belong to the same sensor based on the value of Sensor Id (S_{id}). In other words, all the packets that belong to the same source are arranged together, to minimize the index overhead.

3.6.1 Static-Coalition Algorithm

So far, Static-Coalition Algorithm aims to prepare the packets of random mobile sensors, to mitigate the effects of randomly received packets by arranging them together. This subsection presents the algorithms of Static-Coalition algorithm. To describe the algorithm, Static-Coalition consists of two main phases: Elaboration and Processing phase.

3.6.1.1 Phase 1: Elaboration phase

This phase is an inception phase, where the required information to start communication between sources and destinations is determined. The algorithm is based on receiver-based routing (Mutthigarahalli Shankarappa and Shankar 2017), where the destinations are responsible for asking the required information to initiate communication such as network area, the total number of sensors, sensors ids.

Universiti Malaya

3.6.1.2 Phase 2: Processing phase

The processing phase is the core one of this algorithm. The details present in the following sub-phases.

(a) ***Map***

When packets received by the appropriate gateways, Static-Coalition algorithm duty is starting. First, the initial static-coalitions are established. Then, each static-coalition starts arranging the packets that belong to the same sensor together. This phase is considering as pre-initial indexing, in order to improve the indexing process. The output of this phase is $Pi \leftarrow PiU\{<si,gwj>\}$ which P: is considered as a group that represents a set of pairs and each pair represents a gateway and its appropriate sensors that enable it to send packets to it based on the value of relevancy-measure.

The main flow in this sub-phase, as the following: First, each gateway calls Algorithm 3.2. This algorithm helps to construct static-coalition. Then, each gateway is waiting to receive the sensed data. Second, the sensor sends the sensed data and the time of transmission and the location simultaneously. This data sorted in static-coalition. Third, if static-coalition is becoming full the gateway is transmitting static-coalition to the server. Then empty static-coalition to stay ready for receiving new data.

Algorithm 3.2: Map process algorithm.
Input: <Time-Slot, Source Node, Destination Node> **01: Foreach** Time-Slot **02: If** Neighbor contains destination**,** Use the route with maximum CQI**;** **03: Else** Compute CQI values for different links of a node **04:** Find the node with the maximum value of CQI and forward gw_j. **05: For Each** s_i, $s_i \in S$ **06: If** $\exists\ gw_j \in Gw{:}<S_i,gw_i> \in CQI$ **then** **07:** $P_i \leftarrow P_i U\{<s_i,gw_j>\}$ **08:** If can't find destination then use shortest path algorithm,

Universiti Malaya

09: Else

10: Recalculate CQI.

(b) *Aggregate and join*

In this step, the aggregations of the data that belong to the same sources are conducted and join them in one static-coalitions. The output of Algorithm 3.1 is a list of pairs stored in each gateway within a specific time slot. When gateways send the static-coalition to the server, it results in accumulated data received from different sources.

The existence of static-coalition that contains shared information from different resources, requires the need to aggregate and join. This sub-phase aims to rearrange the static-coalition data by the group the data that belong to the same source and group them as Coalitions. In other words, this step aims to generate Coalitions and arrange all data attributes that belong to the same Coalition, inside each static-coalition of the gateway.

As in Algorithm 3.3, arranges the data attributes of the sensors that belong to the same sensor. This process performes inside the gateway and negotiates to put the right Coalition to the appropriate packets.

Universiti Malaya

Algorithm 3.3: Aggregate and Join process algorithm.

Input: $< P_i >$

01: ForEach gw_j, $gw_j \in GW$, in sequential order

02: Randomly select and IDLE s_i, $s_i \in A$

03: State $(s_i) \leftarrow$ BUSY

04: While $t<DL(gw_i)$ do

05: For Each $s_i \in S : <s_i,gw_j> \in P_i$

06: IF $\exists\ pack_{gw} \in PACK(S_i)$: $pack_{gw} = pack_{sen}$ **then**

07: negotiate $(S_i,pack_i)$

08: end if

09: end for

10: if $\forall\ pack_{gw} \in PACK(GW)$: $pack_{gw}$ is satisfied then

11: break;

12: else

```
repeat;
end if
end while;
end for
end
```

3.7 Indexing framework that replaces the dependability on division network area

Grid-based indexing is not suitable for random mobile sensors, since it increases the indexing overhead. The indexing overhead is increased because the insert and update procedure is increased when the mobile sensor changing its grid.

To address this problem, Coalition-Based Index-Tree framework is proposed. It is a tree-based index structure dedicated to random mobile sensors in WSNs with no inner-nodes. It exploits the benefits of D-tree (Chen, N. 2015) that has low-level space, and it treated directly with the actual time and spatial location of the random mobile sensors. The mobility of sensors is most commonly represented as a spatial location and transferring time as attributes. The basic idea of Coalition-Based Index-Tree framework is to manage mobile sensors by Coalition-Based Index-Tree nodes' structure that relies on multiple coalition methods. Furthermore, the basic idea is to initiate the basic coalitions as basic nodes of the index structure. Accordingly, the nodes of the proposed index-structure are based on the dynamic-coalition, where each node (coalition) has the same volume. Each constructed node includes the packets that belong to the same sensor.

Figure 3.11 depicts Coalition-Based Index-Tree framework that describes the basic steps to index packets of random mobile sensors. The steps start when the packets are received by the server, which is received continuously during the lifetime of the network. As in the figure, the layout of the received packets, which represents the time of received packets (t1, t2, ..., tn). Also, the layout of the received packets represents each received

Universiti Malaya

packet i.e. (P1, P2, …) and sources i.e. (S1, S2, …, Sn). Consequently, the initial tree-structure nodes are constructed based on the number of sensors, where, the number of nodes is equal to the number of sensors. Then, the packets are added to each node based on the location and its source. Each node arranges the packets in decedent order based on transferring time. For further details, the next subsections explain in detail Coalition-Based Index-Tree.

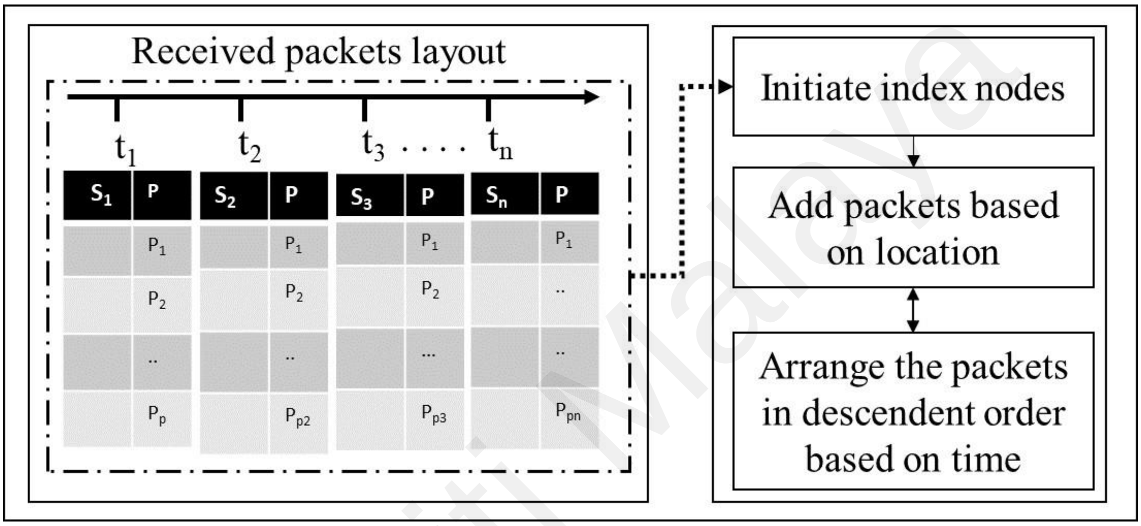

Figure 3.11: Coalition-Based Index-Tree framework

3.7.1 Static-coalition fetching and stored

As mentioned, Coalition-Based Index-Tree performs indexing of the received packets in the server, once packets are received. More details, static-coalitions are received by the server as blocks, and each block arranges the packets in the same coalition-based index tree arranges.

The static-coalitions are fetched when they are transferred from gateways to the server using CQI. The indexing process is starting, where each static-coalition block is arranged and labeled as a sensor. By the existence of the static-coalition it is no need to read each packet alone, it just read the label of the static coalition and adds it in the appropriate index node as a block (describes in detail in Subsection 3.7.2).

As in Figure 3.12, suppose there are three sensors labelled as S_1, S_2, and S_3, transfer static-coalitions labelled as P, represented as Part A. Part B, represents the constructed index structure that contains just one node labeled as S_1. Accordingly, the final index structure represents the following: the index reads the first static coalition S_1 if there is a node in the tree labelled in the same name, if yes no need to rebuild the new node otherwise it needs to build one as in S_2 and S_3. Due to the existence of S_1 node, the index just takes its packets (P_1, P_2, and P_3 in this example) and adds them to the node.

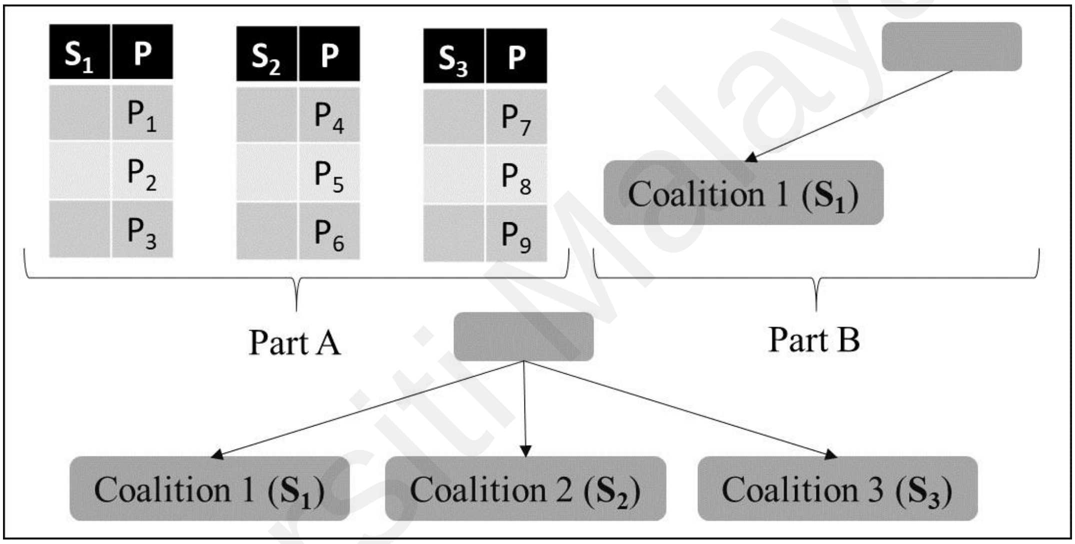

Figure 3.12: Coalition-Based Index-Tree framework

3.7.2 Coalition-Based Index-Tree structure

In Coalition-Based Index-Tree structure, mobile sensors' nodes are represented as a collection of spatial and temporal attributes (Long and Nelson 2013). Namely, they represent in the form S_{data} <S_{id}, T_{stamp}, Position, Attribute>, where S_{id} is a unique sensor identifier, T_{stamp} is the time of data transfer, Position represents the location of the sensor at the transferring time, and Attributes are the attributes of S_{data}. Coalition-Based Index-Tree based on managing and controlling the random mobile sensors packets, to be fitting for random mobile sensors.

The nodes of Coalition-Based Index-Tree structure constructs base on the number of sensors. Figure 3.13 represents the structure of Coalition-Based Index-Tree. To build

Coalition-Based Index-Tree index structure, it constructs the nodes based on two priorities Coalition (C) and then the sensor's Location. Coalition (C) represents the name of the group that contains the details of packets that represent the sensor that belongs to the same source. The number of constructed Coalitions is equal to the number of sensors in the network. By way of explanation, suppose the sensor network has twenty random mobile sensors, which means the number of constructed C will be twenty groups. The content of each C is the actual indexing nodes that represent the real indexing structure. The border of this content is determined by the location, that inserts packets on them, and the sequential arranges of packets in the group C represent the time of the packet transition.

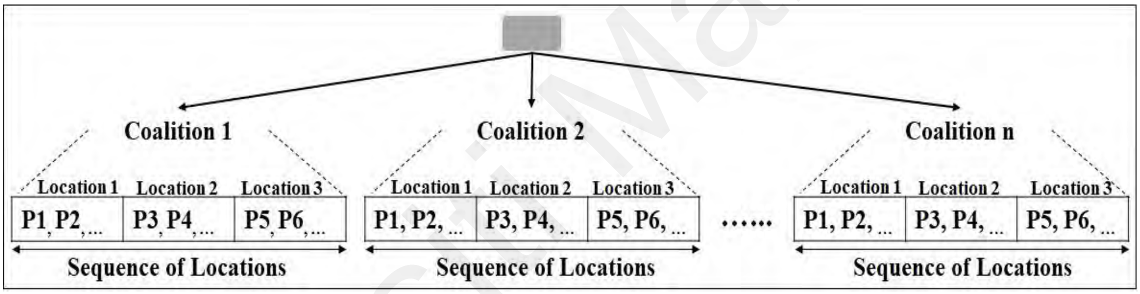

Figure 3.13: Structure of Coalition-Based Index-Tree

3.7.2.1 Spatial and temporal coalitions metrics

The goal of Coalition-Based Index-Tree is to improve the structure index operations, by indexing the packets based on spatial and temporal attributes. In particular, to effectively allocate each packet in its appropriate position in the index structure, it needs to take into account temporal and spatial attributes. As mentioned, the index structure formation is obeyed with the source of each packet, transferring location, and time.

Suppose the agent of the coalition is represented as sensors S. Consequently, to specify the completion of tasks, the binary function B_f is identified. More specifically, the tasks are presenting the packets of gateways, static-coalition in this work S_c. The relation of the coalition defines as a packet's attributes, either time t or spatial S_P. The binary function represents the fact that the task is completed or not, according to the following equation

$$B_f(t) = \begin{cases} 1, & t > current\_time \\ 0, & otherwise \end{cases} \tag{17}$$

Similar to t, the binary function is identified for the location S_P

$$B_f(sp) = \begin{cases} 1, & S_P \text{ } is \text{ } constructed \text{ } inside \text{ } coalition \\ 0, & otherwise \end{cases} \tag{18}$$

Thus, B_f returns 1 only if the given task is completed. Otherwise, it refers to 0. This is considered satisfied or unsatisfied in line 11 of Algorithm 3.5, which accurately describes in the subsequent section.

3.7.3 Coalition-Based Index-Tree algorithm

This subsection explains the details of Coalition-Based Index-Tree framework. It starts from tree nodes construction to add the packets to the appropriate location based on a coalition.

In order to build a structure of Coalition-Based Index-Tree, it starts with initiating its nodes. Hence, Algorithm 4 represents the flow of index nodes construction. As a first step, it starts by defining A which represents the name of a current sensor without duplication. This means, if the actual number of sensors is five, the set A members equal to five. The elements of set A represent the label of nodes, that indicate the source of the received packets. Line 2 to line 4 aim to construct partitions P_i, that means to determine the candidate packets for each constructed sensor based on a relation (R). R indicates the source of packets. In other words, from which sensors are transferred. Accordingly, the constructed P_i is a set contains packets belong to the same source, as candidates. After that, the actual node construction is starting. In each sensor in P_i, if the sensors already build as a node, restart a loop. Else, initiates the non-existing node label.

Based on the output of Algorithm 3.4, the results in a set of partitions, and the nodes are constructed, it starts to add the packets to the appropriate node. Algorithm 3.5 describes this in detail. As a source of each packet, if the node exists add it directly, else return to Algorithm 3.4 to build the node, then add.

Algorithm 3.4: Coalition-Based Index-Tree nodes construction

01: **For each** S_a; $S_a \in A$ in Sequential order;
02: **If** $\exists\, S_a \in A: < S_a, Pack_i > \in R$ **then**
03: $P_i \leftarrow P_i \cup \{< S_a, Pack_i >\}$; // To create partition
04: **End**
05: **For each** partition in P_i
06: If S_a not exist in the tree nodes
07: Build it
08: **Else** break
09: **End** for

Algorithm 3.5: Adding packets on the constructed nodes

01: Calling Algorithm 4 to construct partitions
02: For each packet received by the server
03: If $S_a\ of\ Pack_i \in P_i$ exists
04: Break;
05: Else Call Algorithm 4
06: End for;
07: For each $Pack_i \in \emptyset; in\ sequential\ order$
08: Randomly select an idle agent, $L_i \in A$ as initiator;

$$State(L_i) \leftarrow BUZY$$

09:While $t < DL(Pack_i)\ do$ /* t is the real-time*/
10:For each $L_i \in A: < L_i, Pack_i > \in P$
11:If $\exists r_{\emptyset}^{l} \in R(\emptyset): r_{\emptyset}^{l} = r_{aj}$
And $r_{\emptyset}^{l}$ is unsatisfied as equation (17) and (18) then
12:Negotiate ()\\put the packets and arrange based on time
13:End if

Universiti Malaya

3.8 Chapter summary

This chapter presented the details of the proposed model, which is dedicated to improving the indexing of random mobile sensors in WSN. Furthermore, this chapter presented an empirical analysis to demonstrate that the mentioned problem is non-trivial. Indeed, the impact of the randomness of the sensor nodes' mobility on indexing performance is investigated. Also, this chapter experimentally and analytically demonstrated that randomness mobility of sensor nodes exacerbated the challenges of indexing. The aforementioned results touched the conclusion that there is a relationship between the disseminated packets of random mobile sensors and the performance of indexing.

DySta-Coalition model bases on exploiting the sensors and gateways, in order to prepare packets for indexing in the server. Furthermore, it improves indexing by taking into account the behavior of mobility sensors. The proposed model consists of Dynamic-Coalition framework, Static-Coalition algorithm, and Coalition-Based Index-Tree framework. Dynamic-Coalition alleviates the effect of distribution packets of random mobile sensors, by constructing dynamic blocks based on the coalition structure, called dynamic-coalition. Static-Coalition algorithm is able to rearrange the packets in each gateway based on the source of them, in order to arrange the packets, step-by-step, before it accumulated in the main server for indexing. Coalition-Based Index-Tree framework is a tree-based index structure dedicated to random mobile sensors. The subsequent chapters (Chapter 4 and Chapter 5) present the details of the empirical setup and model the solution setup.

Universiti Malaya

CHAPTER 4:

FRAMEWORK FOR THE DISSEMINATED PACKETS THROUGHOUT DESTINATIONS: IMPLEMENTATION AND EVALUATION

So far, the problem of indexing packets in random mobile WSN has identified and investigated. As well, a novel model called DySta-Coalition is proposed as a solution. The importance of this chapter and the subsequent one comes from the need to examine DySta-Coalition model's ability to solve the formulated problem in Chapter 3. The evaluation conducts in two phases, the first phase presents in this chapter, and the second presents in Chapter 5. The First phase evaluates Dynamic-Coalition framework and comparing them with Active data Dissemination (Jeongcheol Lee et al. 2018). This evaluation phase examines the ability of the proposed Dynamic-Coalition framework (the first part of DySta-Coalition model) to mitigate the disseminated packets. The second phase evaluates Coalition-Based Index-Tree framework, which examines the effects of mitigation of the disseminated packets (the first part of DySta-Coalition model) and arranging packets (the second part of DySta-Coalition model, that is Static-Coalition Algorithm) on the performance of indexing. The evaluation results compared with R-tree (Antonin Guttman 1983), as well as a recent algorithms Decomposition-tree (D-tree) (Chen, N. 2015), and Grid-based DBSCAN with Cluster Forest (GDCF) (Boonchoo et al. 2019).

This chapter presents an evaluation platform that examines the efficiency of the Dynamic-Coalition framework to mitigate the disseminated packets throughout destinations. The chapter organizes as follows. Section 4.1 explains the experimental design. Section 4.2 presents the metrics used in the evaluation phase. The evaluation results and discussion of Dynamic-Coalition framework presents in Section 4.3. The summary of this chapter presents in Section 4.4.

4.1 Experimental Design

This section demonstrates the followed processes to evaluate DySta-Coalition model. The contents of this section are for the purposes of this chapter and Chapter 5. Hence, the observed effects are obtained as evaluation results.

To deeply explain the experimental design, this section presents and explains the dependent and independent variables. Indeed, this section presents the formulation of hypotheses, to predict the relation between the independent and dependent variables. Furthermore, it presents the comparison standard that uses to detect and measure the evaluation results. Last but not least, this section mentions the used simulator and its detailed procedures.

4.1.1 Formulation of hypotheses

An important aspect of the evaluation is to formally and clearly state what is going to be evaluated. Therefore, the hypotheses – denoted as H in this thesis- are identified to predict the relationship between the independent and dependent variables. The hypotheses are identified according to the Research Questions (RQs), which are mentioned in Section 1.4.

RQ2: How to minimize the sensors Routing Overhead by transmitting to a fewer number of directions?

H1: Overhead and the efficiency of the indexing procedures are related, such that the minimization of Average Routing Overhead for all sources (sensors and gateways) increases the index efficiency.

This hypothesis focuses on mitigating the number of transferred packets (Routing Overhead) from each sensor. Hence, by establishing a relationship between sources and destinations using the proposed coalition-based model, it mitigates the disseminated packets for improving indexing efficiency. Namely, improving indexing in terms of number traversed nodes, building-time overhead, and space-cost overhead, as depicted in Figure 4.1, H1.

RQ3: How to mitigate the packets of the sensors from scrambled?

H2: Packet Delivery Ratio and throughput and the efficiency of the index are related, such that the ability of each destination receives packets from a dedicated number of sources increases the efficiency of the index.

This hypothesis focuses on maximizing the number of transferred packets from each sensor to a specific destination. Hence, by establishing a relationship between source and destination using coalition, in order to mitigate the disseminated packets for improving indexing efficiency. Namely, improving indexing in terms of traversed nodes, building time, and space cost, as depicted in Figure 4.1, H2.

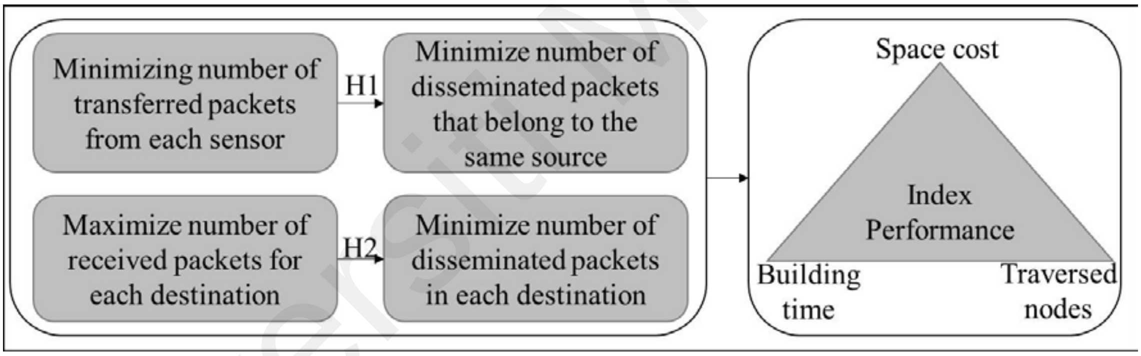

Figure 4.1: Hypothesis 1 and 2 and the effects on the index performance

RQ4: How to minimize the number of indexing procedures, which affected by the disseminated packets?

H3: Number of traversed nodes in order to build index structure and the number of index operations are related, such that minimizing the number of traversed nodes minimizes the index operations (insert, update, and delete).

This hypothesis focuses on minimizing the number of passed nodes to build the index structure, in order to mitigate the insert, update, delete operations overhead for improving indexing efficiency. Namely, improving indexing in terms of traversed nodes, as depicted in Figure 4.2, H3.

RQ5: How to reduce indexing procedures overhead for data transmitted from gateways?

H4: Index operations overhead and the packets received by destination are related, such that minimizing the number of received packets by the server using static-coalitions, which constructed by Static-Coalition Algorithm, alleviates the index overhead in terms of index building time and space cost overhead.

This hypothesis focuses on minimizing the index building time and space cost overhead by initiating static-coalitions, in order to increase the indexing efficiency. Namely, improving indexing in terms of building time, and space cost, as depicted in Figure 4.2, H4.

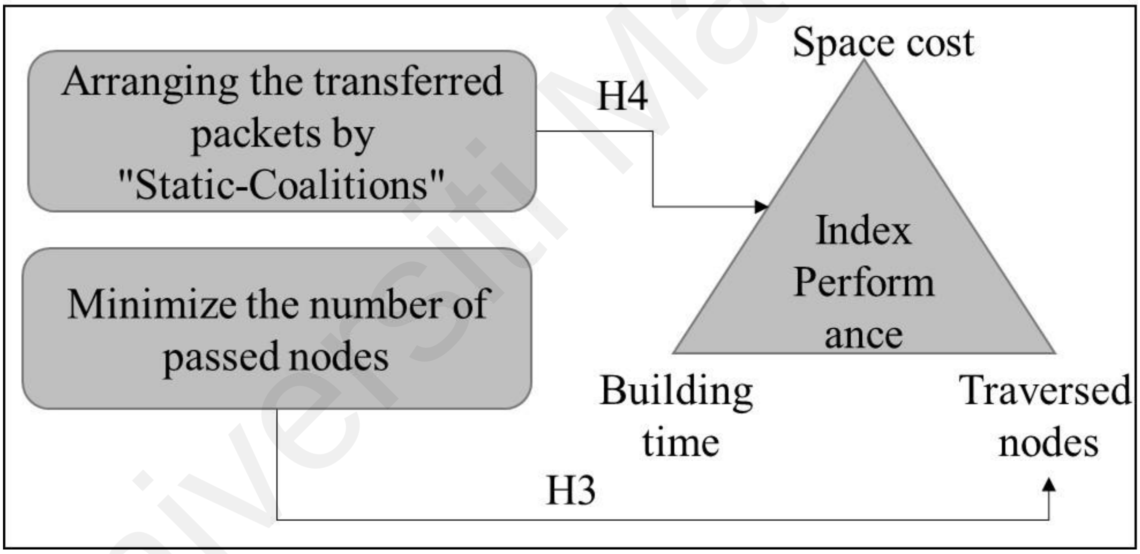

Figure 4.2: Hypothesis 3 and 4 and the effects on the index performance

For more clarification, the relation among RQs, Os, and the constructed Hypotheses are summarized in Table 4.1.

Table 4.1: The relation among RQs, Os, and the constructed Hypothesis

#	Question	Objective	Hypothesis
1	**RQ2**: How to minimize the sensors Routing Overhead by transmitting to a	**RO2**: To minimize random mobile sensor overhead, by initiating a relationship between source and destination.	**H1:** O*verhead* and the efficiency of the index are related, such that the minimization of Routing Overhead for each source

	fewer number of directions?		(sensors and gateways) increases the index efficiency.
2	**RQ3**: How to mitigate the packets of the sensors from scrambled?	**RO3**: To improve the index operations of the system, by maximizing the number of transferred packets from each sensor to a specific destination.	**H2:** Packets Delivery Ratio and throughput and the efficiency of the index are related, such that the ability of each destination receives packets from a dedicated number of sources increases the efficiency of the index.
3	**RQ4**: How to minimize the number of indexing operations, which affected by the disseminated packets?	**RO4**: To reduce the number of indexing operations in order to minimize the number of index nodes traversed during index construction.	**H3:** a number of traversed nodes in order to build index structure and the number of index operations are related, such that minimizing the number of traversed nodes minimizes the index operations (insert, update, and delete).
4	**RQ5**: How to reduce indexing operations overhead for data transmitted from gateways?	**RO5**: To effectively build-up static-coalitions in each gateway to reduce the number of indexing operation overhead in terms of space-cost and index building time.	**H4:** index operations overhead and the packets received by destination are related, such that minimizing the number of received packets by the server using static-coalitions alleviates the index overhead in terms of index building time and space cost overhead.

4.1.2 Variables

According to the hypotheses explained in the previous subsection, noted that there are a set of variables that should be listed. The variables are defined as factors that can be changed in the evaluation. Any variable is either a dependent or independent variable. Independent variables are variables whose values are intentionally modified or manipulated, which are measured by units. Dependent variables are variables whose response values change, due to the dependability on the inputted values of independent variables.

Universiti Malaya

In this thesis, the independent variables are the number of dedicated sensors, the number of sensors, and the radius of the constructed group. The constructed groups, which construct by the gateways and the sensors that fall within the same bandwidth region, are defined as circles (Assumption 8 in Subsection 3.2.1.3). Likewise, the dependent variables are the Average Routing Overhead, the packets delivery ratio, throughput, the number of traversed nodes, the space cost overhead, and the indexing building time overhead. Table 4.2 summarizes the hypotheses, dependent, independent variables, and in which chapter they were used. The details of the dependent variables are defined in detail in Section 4.2.

Table 4.2: Hypotheses and derived dependent and independent variables

#	Hypothesis	Independent Variables	Dependent Variables	In which chapter they used?
1	H1	Routing Overhead	Number of dedicated sensors and Number of sensors	Chapter 4
2	H2	Packets delivery ratio and Throughput	Number of dedicated sensors and Number of sensors	Chapter 4
3	H3	The number of traversed nodes	Radius and Number of sensors	Chapter 5
4	H4	Index space-cost and index building time overhead	Radius and Number of sensors	Chapter 5

4.1.3 Experimental setting

This subsection defines the comparison standard based on performing evaluation. So, the comparison standard of the proposed model is performed according to baselines, R-tree, D-tree, GDCF, and Active data Dissemination. The standard of comparison is followed by the specification of Active data Dissemination (Jeongcheol Lee et al. 2018) and GDCF (Boonchoo et al. 2019). Table 4.3 summarizes the used baselines, the followed

Universiti Malaya

comparison standard, and in which framework and chapter they are used. The detail of each baseline and the rationale for why choosing them as a standard baseline explains in the following subsection.

Table 4.3: Summary of used baselines, and the followed comparison standard, and in which framework and chapter they are used

#	Baseline	Proposed used framework	The followed comparison standard	In which chapter they used?
1	Active data Dissemination	Dynamic-Coalition framework	(Jeongcheol Lee et al. 2018)	Chapter 4
2	GDCF, D-tree, and R-tree	Coalition-Based Index-Tree framework	(Boonchoo et al. 2019)	Chapter 5

4.1.3.1 Brief description of Baselines

This research has reported an evaluation with two phases. In order to assess the effectiveness of these experiments, the experiment compares with other baselines. The baselines are D-tree, R-tree, Active data Dissemination, and GDCF that are presented in (Chen, N. 2015), (Antonin Guttman 1983), (Jeongcheol Lee et al. 2018) and (Boonchoo et al. 2019), respectively. Although the mentioned baselines are mentioned in detail in Chapter 2, it does not matter if re-sheds light on them again. A brief description of each one is explained below.

(a) ***Active data Dissemination***

Active data Dissemination is introduced in (Jeongcheol Lee et al. 2018), which uses in this chapter. It reduces the number of transmitted packets in the network, by moving near to the sensor nodes to collect data. Thus, reducing the probability of collisions, and transmissions.

The premise behind the proposition is that all sensors know the moving direction and pattern of the sinks, which is informed by a sinking leader. The network establishes Grid-based local data areas, where it is predicted that the sinks will go through. All the sensor nodes send the readings to those local data areas, which will be picked up by the sinks in one-hop communication, while they are passing those localities. Given the short transmissions between the nodes and sinks, energy in the network can be substantially saved. Nevertheless, the sinks are expected to move slowly, so that the moving directions can be accurately estimated. For more explanation, Active data Dissemination stands on the following stages: (i) Sink group registration. (ii) The sensors (sources) disseminate packets to the current region. In the first stage, the network nodes (sensors and gateways) define as mobile groups, in which each group consists of a source, mobile sink group, and leader sink. Accordingly, when the packets are received by gateways from sensors, the mobile sink group needs to register its current region location information to the source. Furthermore, it needs to elect the leader to sink for the sink group. The leader sink is elected by choosing the most stable gateway. As well, in the second stage, each mobile sink could get the packets from sensors that move in the area. To fulfill that, a closed Grid-based local data area is constructed.

(b) *Grid-based DBSCAN with Cluster Forest*

GDCF tried to solve two problems: (i) neighbor explosion and (ii) redundancies in merging. It utilizes bitmap indexing to support efficient neighbor grid queries. Second, based on the concept of a union-find algorithm, a forest-like structure is devised, called cluster forest. Cluster forest alleviates the redundancies in the merging. Moreover, finding that running the cluster forest in different orders can lead to a different number of merging operations needed to perform in the merging step. The merging step in a uniformly random order is performed to optimize the number of merging operations.

(c) *Decomposition-tree*

D-tree (Chen, N. 2015) is spatial-temporal indexing dedicated to multi-dimensional mobile big-data. D-tree's structure is without inner-nodes, which occupies less memory space-size. Indeed, it has a discrete linear structure, which makes parallel implementation easy. In brief, D-tree is a space division access method. It recursively decomposes the basic space into 2-d smaller sub-grids. Each sub-grid has the same volume and has a unique integer.

(d) *R-tree*

The last baseline is R-tree (Antonin Guttman 1983) is a height-balanced structure for n-dimensional spatial-objects. In order to handle spatial data efficiently, as required in spatial-temporal applications, it requires an indexing mechanism that helps to retrieve data items quickly according to spatial locations. So, R-tree is a dynamic index structure, which meets the mentioned requirement.

In Summary, the conducted evaluation in this thesis is applied to assess the ability to minimize the number of disseminated packets throughout destinations and increasing the number of dedicated sensors arrived at each destination (presents in this chapter). Accordingly, Active data Dissemination considers recommending work to be qualified as a baseline. Furthermore, the conducted evaluation (explains in detail in Chapter 5) applies to assess the performance of Coalition-Based Index-Tree. Since D-tree, R-tree, and GDCF are based on the division of the network area, it considers recommending work to be qualified as a baseline.

4.1.3.2 Evaluation Environment and model

The proposed DySta-Coalition model is evaluated using a simulator implemented using Visual C# 2017 (Ghaleb et al. 2017). The implementation is not just conducted for the proposed framework, but also to Active data Dissemination, R-tree, D-tee, and GDCF. The implementation is carried out on Windows 10 Home 64-bit, i7 CPU, 8192 RAM, and 2.20 GHz.

Universiti Malaya

The implementation is designed for random mobile environments. Hence, the sensors and gateways nodes possess dynamic behavior. The sensors regenerated packets during movement by establishing new relevancy between sensors and gateways.

In order to examine the validity of the proposed DySta-Coalition model, two evaluations are carried out, each of which is executed in a different environment. The first evaluation environment plans to evaluate Dynamic-Coalition framework by comparing it with Active data Dissemination. The main aim of this evaluation phase is to ensure that the proposed framework succeeds in mitigating disseminated packets. Thereby, examining the effects of the disseminated packet on the performance of the index, in the next evaluation phase. The second evaluation environment examines the proposed Coalition-Based Index-Tree framework by comparing it with R-tree, D-tree, and GDCF. The summary of the evaluation is summarized in Table 4.4. The detail of each evaluation is mentioned in the subsequent subsections.

Table 4.4: Summary of evaluation steps

Proposed Solutions	Dynamic-Coalition framework	Static-Coalition Algorithm	Coalition-Based Index-Tree framework
Evaluation	Evaluate them, to measure that achieving the minimum Routing Overhead, throughput and Packet Delivery Ratio compared with Active data Dissemination.		Evaluate Coalition-Based Index-Tree compared with R-tree, D-tree, and GDCF.
Chapter	Chapter 4		Chapter 5

(a) ***The First Evaluation Environment***

This evaluation phase evaluates the performances of Dynamic-Coalition framework by comparing it with Active data Dissemination. Active data Dissemination is the most recent group-casting framework. In this evaluation phase, the model of sensor nodes is followed by the specification of (Jeongcheol Lee et al. 2018).

The evaluation implementation conducts from sensors and gateways that deploy randomly. The implementation designs for random mobile environments. Hence the sensors and gateways possess dynamic behavior. Accordingly, the transmission range of sensor nodes is 50 m, where the transmission range is the distance required to transfer packets from source to destination. Transmitting and receiving power consumption rates of the sensors are 21 mW and 15 mW, respectively. In evaluation scenarios, 4000 sensor nodes are randomly distributed in 3000 m × 3000 m sensor fields. The number of gateways is 10 that move randomly. The gateways have a circle region and the radius (R) of the circle region is 100 m. The mobile sink group follows Random Waypoint Model with an average of 10 m/s within the mobile sink group region. The location of a source is randomly selected, and the source generates a data packet every two seconds. The total simulation time lasts for 100s. For each scenario, the results presented here are the average of 10 separate simulation runs.

Another implementation design conducts in this phase according to the number of sensors. The sensors regenerated packets during each iteration by establishing new relevancy between sensors and gateways, where the sensors and gateways locations are changing. Furthermore, the number of deployed gateways is 10, and the number of deployed sensors is changed in the set {100, 200, 300, 400, 500, 600, 700, 800, 900}.

All the nodes are assumed to have a certain amount of energy before the routing process, the residual energy of each node depends on the number of transmissions that the node is involved in. The sensors are responsible to transmit the packets to neighbor's gateways, thus it should communicate with nodes that are in the range in order to find the best route between receivers (represent gateways if the data is transferred from sensors and server if data transferred from gateways) and the sender (from gateways to server or from sensors to gateways). The kind of packets used in the text data of 50Kb and the data-rate is 500 Kbps. The simulation execution time is a user control that is equal to 5 seconds.

The packets are generated from sensors. Each sensor transfers packets from different locations and time periods. Hence each sensor creates a set of packets during the execution period from different locations.

(b) ***The Second Evaluation Environment***

This phase evaluates Coalition-Based Index-Tree framework compared with GDCF. This phase splits into three scenarios. The first scenario is evaluating the performance of Coalition-Based Index-Tree without interventions. The second and third ones examine the effects of Dynamic-Coalition in addition to Static-Coalition on Coalition-Based Index-Tree. In addition to the previous parameters in the first stage, this stage followed the specification of (Boonchoo et al. 2019). The synthetic dataset is generated using C# (Ghaleb et al. 2017), according to the following parameters: the number of mobile sensors, the number of clusters, the number of dimensions. The number of dimensions is equal to 3. And the number of clusters is equal to the number of gateways which equals 10. According to the mentioned parameters, the size of the generated dataset is equal to 3000,000. In addition to the previous experiment setup, another experiment is conducted according to the number of sensors. This experiment setup follows the mentioned parameters. The Summary of parameters is in Table 4.5.

Table 4.5: Summary of parameters

#	Parameters	Values
1	Number of mobile sensors	4000
2	Number of clusters	10
3	Number of dimensions	3

4.2 Evaluation Metrics

The evaluation (routing evaluation) conducts in this chapter used Routing Overhead and packets delivery ratio metrics. A brief description of used metrics as the following.

Universiti Malaya

- **Packets delivery ratio**: the ratio of the number of packets sent by source nodes and the number of packets received by destinations.

$$PDR = \frac{\text{Received Packets}}{\text{Generated Packets}} \times 100\%$$

- **Throughput**: is the number of packets that transferred from the sensors to the destination over a given period of time. The destination in this work is represented by gateways and server.

$$Throughput = \frac{\text{Number of Transferred Packets}}{\text{Period of time}}$$

- Average Routing Overhead: is the number of routing packets required for network communication.

$$\text{Routing } Overhead = Number\ of\ source\ routing\ packets$$

Table 4.6 provides the list of metrics evaluated along with the units.

Table 4.6: Evaluation Metrics and units

#	**Metric**	**Unit**
1	Routing Overhead	Packets
2	Packets delivery ratio	Percent (%)
3	Throughput	Packets\second

4.3 Dynamic-Coalition framework evaluations

The purpose of this phase is to examine the ability of the proposed Dynamic-Coalition framework in mitigating the disseminated packets throughout gateways. The disseminated packets are mitigated when minimizing the number of transferred packets from each sensor and increasing the number of transferred packets from the sensor to a specific gateway. Accordingly, Dynamic-Coalition framework is evaluated in accordance with the Routing Overhead, throughput and packets delivery ratio.

Universiti Malaya

4.3.1.1 Packet Delivery Ratio

Packet Delivery Ratio is the ratio of the number of packets sent by the source to the number of packets received by the destination. Maximizing the received packets by gateways increases the efficiency of the index, wherefore minimizes the disseminated packets. Accordingly, ensuring its ability to increase the number of received packets by each gateway that transferred from the same sensor. Consequently, it results in the mitigation of the disseminated packets to improve indexing performance.

Figure 4.3 shows the average Packet Delivery Ratio values resulting in executing 4000 sensors that transferred to 10 gateways. The X-axis represents the number of dedicated sensors, ranging from {1,2, …,9}. Dedicated sensors mean concentrating the packets of sensors around the gateways, based only on the power of the sensor. The comparison is conducted by comparing Dynamic-Coalition framework with Active data Dissemination baseline. As noted in the figure, the proposed framework successes to maximize the transferred packets to a specific gateway without affecting the routing performance. The reason behind that, Dynamic-Coalition framework is based on increasing the packets that belong to the same sensor, regardless of the location of the destination. Furthermore, Dynamic-Coalition framework has the ability to initiate relevance between sensors and gateways, to minimize the disseminated packets. Moreover, Dynamic-Coalition framework does not base on the location of the sensors, whatever if there are dedicated sensors or not. More explanation, it is based on relevancy-measure, as mentioned before.

The ability of Active data Dissemination to cope with the proposed framework is unsatisfactory. In this result, the probability of collisions for concentrated packets around the gateway is increased, which drops the packets delivery ratio. Specifically, Active data Dissemination bases on flooding packets in the network area, so many nodes in the network cause high-probability collisions due to the concentrated packets around the sink. Additionally, the tree structure of Active data Dissemination is sparsely constructed over

Universiti Malaya

the entire network, but in the case of supporting mobile gateways groups, the tree is densely constructed in the group area. Thus, it negatively affects the packets delivery ratio as increasing packets' transmission of respective sources. Meanwhile, the proposed Dynamic-Coalition framework shows a better delivery ratio of packets, regardless of the dedicated sensors.

It is obvious that the average packets delivery ratio value for Active data Dissemination is closed to the proposed framework. This closeness does not refute the validity of the proposed framework, due to it overcomes the Active data Dissemination in maximizing the number of transferred packets from sensors to gateways. In these results, the proposed framework satisfies the mitigation of the disseminated packets throughout gateways, as mentioned before.

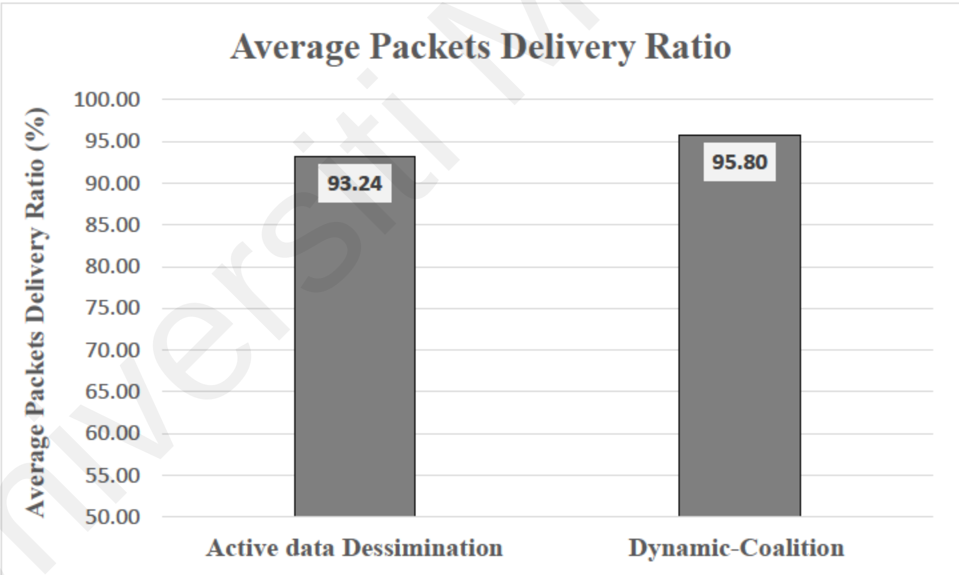

Figure 4.3: Average Packet Delivery Ratio of Dynamic-Coalition compared with Active data Dissemination (Baseline)

Universiti Malaya

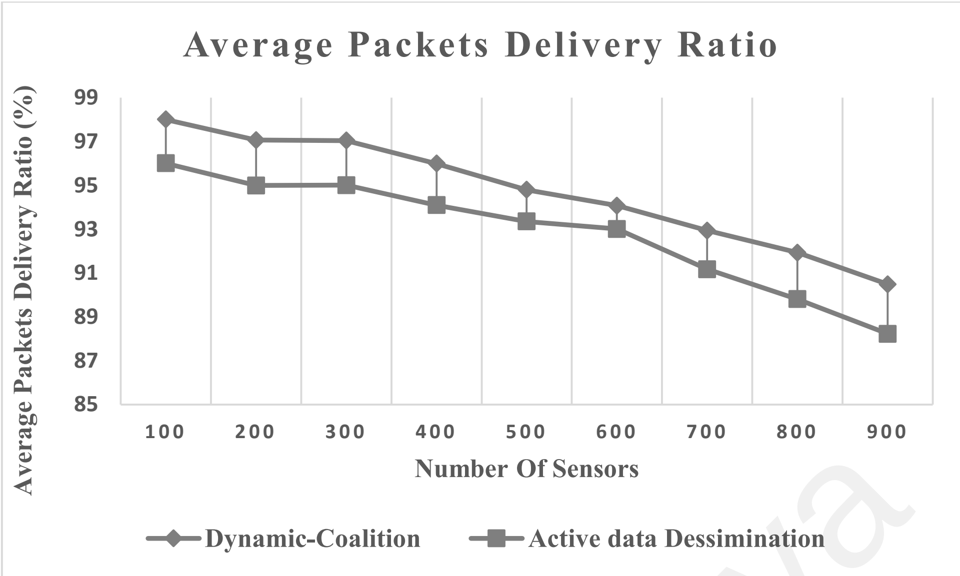

Figure 4.4: Average Packet Delivery Ratio of Dynamic-Coalition framework in accordance with the number of sensors compared with Active data Dissemination (Baseline)

Figure 4.4 shows the packets delivery ratio for the number of sensors. The figure represents packets delivery ratio –represented by Y-axis- of the number of sensors that change in the set {100, 200, 300, 400, 500, 600, 700, 800, 900}-represented in X-axis-. The proposed framework shows a high Packet Delivery Ratio. This is because the number of transferred packets from the same sensor is high, regardless of the location of the destination, as mentioned. However, Active data Dissemination reduces the Packet Delivery Ratio, when the number of sensors increases.

Looking at the figure again, it is obvious that the Packet Delivery Ratio values for both frameworks are declining when the number of sensors increases. This is normal for Packets Delivery Ratio values that minimize when the number of objects increases. However, the difference between the Packet Delivery Ratio values in the figure is not fixed. Since it bases on the movement of sensors and gateways. In other words, the differences are based on the location of transferring packets of sensors and the location of gateways as a receiver. Whatever the differences were, the proposed framework satisfies to mitigate the disseminated packets, due to it has better Packet Delivery Ratio values.

Universiti Malaya

As a result, Packet Delivery Ratio values for the proposed Dynamic-Coalition framework are more superior to Active data Dissemination. The superiority of the proposed framework is because it is able to deal with mobile sensors regardless of locations. More explanation, it constructs coalitions using the existing sensors wherever they come from. Meanwhile, the Packet Delivery Ratio values of Active data Dissemination are dropping, when the dedicated sensors' number is increased.

4.3.1.2 Throughput

Throughput is the number of successfully received packets in a unit time. Maximizing the number of received packets from each sensor to each gateway within a unit time, it increases the efficiency of the index. This is because enabling to alleviate the number of packets that transferred from the same sensor. In other words, the maximum number of accumulated packets in each gateway that transferred from each sensor consider better in improving the performance of an index, by increasing the number of packets that belong to the same source in each destination.

Figure 4.5 shows the average throughput values resulting in executing 4000 sensors that transferred to 10 gateways. The X-axis represents the number of dedicated sensors, ranging from {1,2, …,9}. Dedicated sensors mean concentrating the packets of sensors around the gateways, based only on the power of the sensor. The comparison is conducted by comparing Dynamic-Coalition framework with Active data Dissemination baseline. As noted in the figure, the proposed framework successes to maximize the transferred packets to a specific gateway without affecting the routing performance. The reason behind that, Dynamic-Coalition framework is based on increasing the packets that belong to the same sensor, regardless of the location of the destination. Furthermore, Dynamic-Coalition framework has the ability to initiate relevance between sensors and gateways, to minimize the disseminated packets. Moreover, Dynamic-Coalition framework does not

Universiti Malaya

base on the location of the sensors, whatever if there are dedicated sensors or not. More explanation, it is based on relevancy-measure, as mentioned before.

The ability of Active data Dissemination to cope with the proposed framework is unsatisfactory. In this result, the probability of collisions for concentrated packets around the gateway is increased, which drops the packets delivery ratio. Specifically, Active data Dissemination bases on flooding packets in the network area, so many nodes in the network cause high-probability collisions due to the concentrated packets around the sink. Additionally, the tree structure of Active data Dissemination is sparsely constructed over the entire network, but in the case of supporting mobile gateways groups, the tree is densely constructed in the group area. Thus, it negatively affects the packets delivery ratio as increasing packets' transmission of respective sources. Meanwhile, the proposed Dynamic-Coalition framework shows a better throughput of packets, regardless of the dedicated sensors.

It is obvious that the average throughput value for Active data Dissemination is closed to the proposed framework. This closeness does not refute the validity of the proposed framework, due to it overcomes the Active data Dissemination in maximizing the number of transferred packets from sensors to gateways. In these results, the proposed framework satisfies the mitigation of the disseminated packets throughout gateways, as mentioned before.

Universiti Malaya

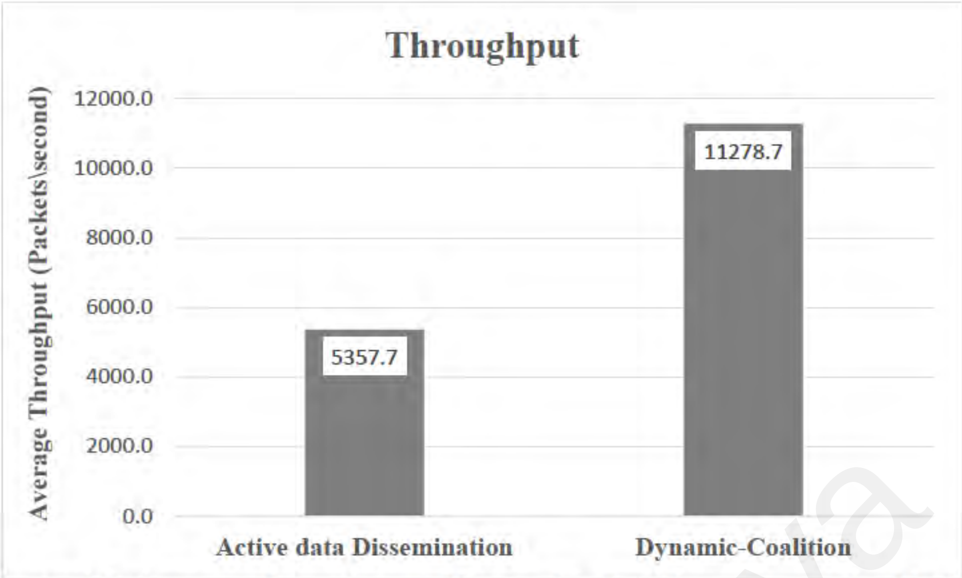

Figure 4.5: Average throughput of Dynamic-Coalition compared with Active data Dissemination (Baseline)

Figure 4.6 shows the average throughput for the number of sensors. The figure represents throughput –represented by Y-axis- of the number of sensors that change in the set {100, 200, 300, 400, 500, 600, 700, 800, 900}-represented in X-axis-. The proposed framework shows a high throughput. This is because the number of transferred packets from the same sensor is high, regardless of the location of the destination, as mentioned. However, Active data Dissemination reduces the throughput, when the number of sensors increases.

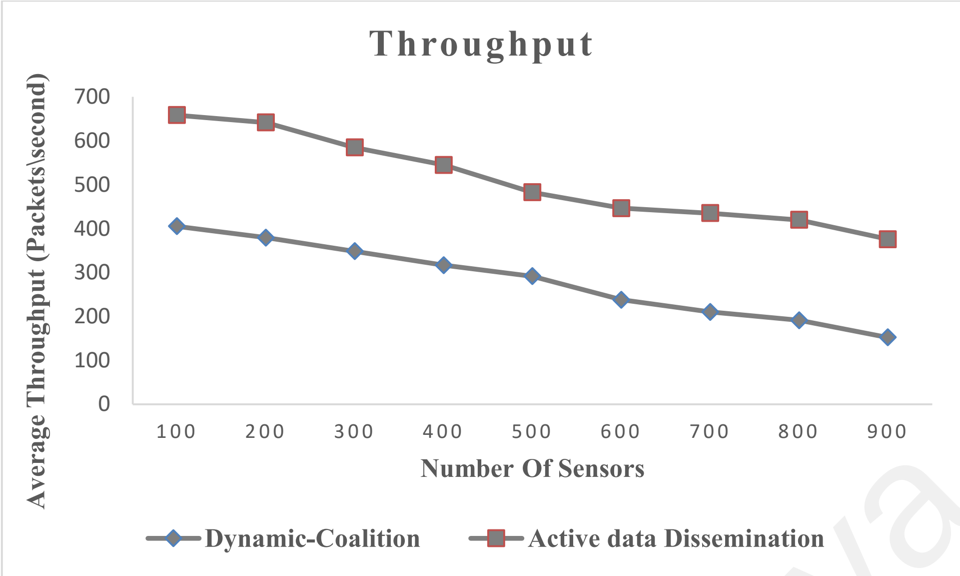

Figure 4.6: Average throughput of Dynamic-Coalition framework in accordance with the number of sensors compared with Active data Dissemination (Baseline)

Looking at the figure again, it is obvious that the throughput values for both frameworks are declining when the number of sensors increases. This is normal for throughput values that minimize when the number of objects increases. However, the difference between the throughput values in the figure is not fixed. Since it bases on the movement of sensors and gateways. In other words, the differences are based on the location of transferring packets of sensors and the location of gateways as a receiver. Whatever the differences were, the proposed framework satisfies to mitigate the disseminated packets, due to it has throughput values.

As a result, throughput values for the proposed Dynamic-Coalition framework are more superior to Active data Dissemination. The superiority of the proposed framework is because it is able to deal with mobile sensors regardless of locations. More explanation, it constructs coalitions using the existing sensors wherever they come from. Meanwhile, the throughput values of Active data Dissemination are dropping, when the dedicated sensors' number is increased.

4.3.1.3 Average Routing Overhead

Average Routing Overhead is the number of required transferred packets for network communication. The small values of Routing Overhead indicate that the number of transferred packets that belong to the same source is mitigated. This leads to mitigate the dissemination packets. The values of Routing Overhead extracted from this evaluation phase represent in Figure 4.7.

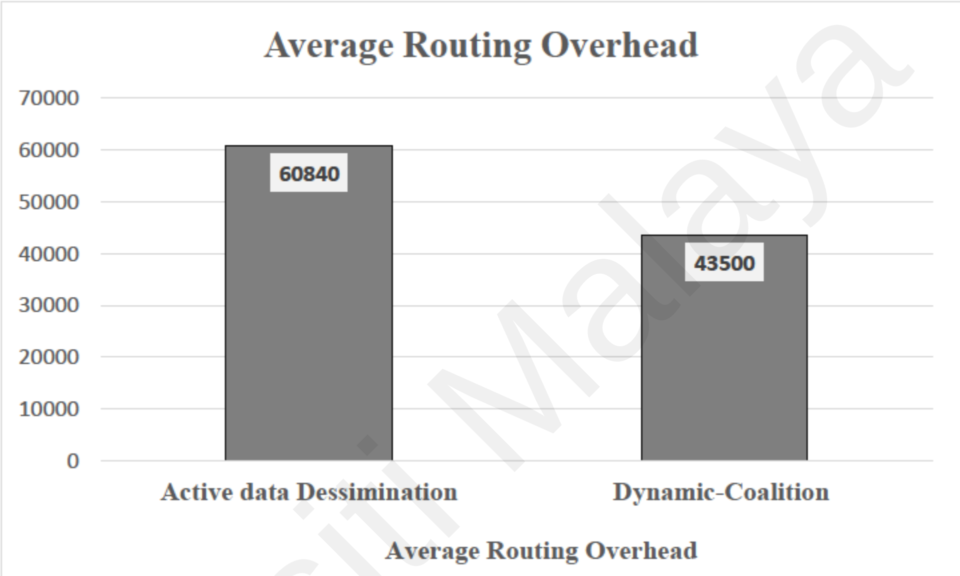

Figure 4.7: Average Routing Overhead of Dynamic-Coalition framework compared with Active data Dissemination (Baseline)

Figure 4.7 plots the Average Routing Overhead of Active data Dissemination and Dynamic-Coalition framework. As shown in the figure, Routing Overhead values of Dynamic-Coalition framework are supreme compared with Active data Dissemination. This superiority is caused by many factors. First, Dynamic-Coalition framework is just dealing with the number of mobile sensors, regardless of transferring locations. Specifically, the proposed framework does not be affected by the number of packets reported from sensors, even in the existence of a large number of them. This is because the proposed framework bases on the number of existing sensors that belong to each coalition. Accordingly, all packets that will report from the same source are treated as a single unit. Second, Dynamic-Coalition framework minimizes the dependability on the

Universiti Malaya

attributes of each packet and substitutes it by depending on the sensors' members that exist in each coalition.

As observed in the figure, the represented behavior of the overhead values with respect to Active data Dissemination is the minimum. This refers that each experiment has its own separated behavior because of the random mobility of the deployed sensors. For more explanation, Active data Dissemination transfers packets according to the shortest-path algorithm, where it is not suitable in random environments, because of the dynamic nature of sensors. Meanwhile, the proposed framework does not be affected because of its dependability on relevancy-measure, as mentioned before.

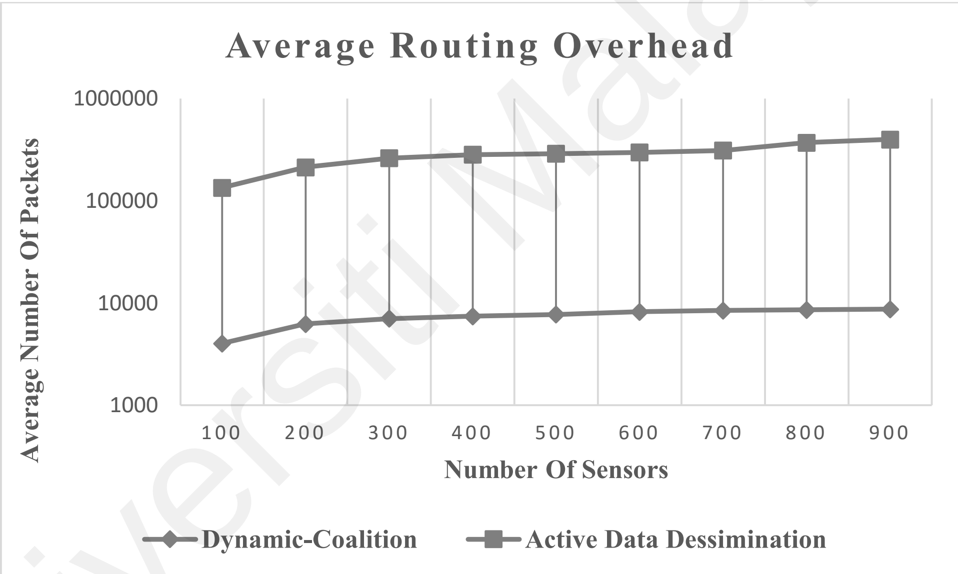

Figure 4.8: Average Routing Overhead in accordance with the number of mobile sensors compared with Active data Dissemination (Baseline)

Figure 4.8 represents the Average Routing Overhead –represented by Y-axis- of the number of sensors by comparing the proposed framework with Active data Dissemination. The number of sensors changes in the set {100, 200, 300, 400, 500, 600, 700, 800, 900}-represented in X-axis-. As shown in the figure, the proposed Dynamic-Coalition framework is superior compared with Active data Dissemination. The superiority of Dynamic-Coalition framework is caused by dealing with the number of mobile sensors, regardless of transferring location, not like Active data Dissemination that deals with all values of the coming packets. More explicitly, if the same sensor transfers packets from different locations, Active data Dissemination considers each packet as a new value, where negatively affects the final indexing.

The maximum value of Routing Overhead is 8727.5204 and 399712.8 for Dynamic-Coalition and Active data Dissemination, respectively. The showed values are reasonable because each sensor transfers packets from a different location and a different period of time. That means the transferred data consider as big-data in the case of random mobile sensors.

Routing Overhead values of Dynamic-Coalition as the following {4034.69, 6251.226, 7048.96, 7431.778, 7733.6, 8230.2205, 8475.2, 8599.338, 8727.5204} for {100, 200, 300, 400, 500, 600, 700, 800, 900}, respectively, as presents in Table 4.7. The differences between these values are smaller than the baseline. This is because the probability of the same sensor for transferring packets from the same location is high.

Table 4.7: Average Routing Overhead values of Dynamic-Coalition

#	**Number of sensors**	Routing Overhead values
1	100	4034.69
2	200	6251.226
3	300	7048.96
4	400	7431.778
5	500	7733.6
6	600	8230.2205
7	700	8475.2
8	800	8599.338
9	900	8727.5204

Universiti Malaya

By returning again to Figure 4.7 the evaluation has conducted for 4000 sensors, and within 100 seconds, as mentioned in Section 4.1.3.2 (a). Meanwhile, Figure 4.8 is conducted for 100 to 900 sensors and within 5 seconds for each one. Both figures are not comparable because they have executed in different environments.

So far, the evaluation has proved that transferred packets from each sensor to different gateways are minimized, and received packets by each gateway from a specific sensor are increased. This results in alleviating the disseminated packets throughout gateways. Accordingly, evaluation of its effects on the Coalition-Based Index-Tree processes, in terms of index building time overhead, space-cost overhead, and the number of traversed nodes. As describes in the subsequent chapter.

4.4 Chapter Discussion and Summary

This chapter presented the evaluation processes, which followed to obtain results. It demonstrated the hypotheses of the research and extracted the independent and dependent variables from them. Moreover, the comparison standards are mentioned to compare the conducted experiments with them. Furthermore, the procedures of evaluation are explained in detail.

The main concept of Dynamic-Coalition framework is utilizing the coalition concept instead of grids (division the network area into grids). This added value to improve indexing because of the following reasons. First, minimize the number of transferred packets from each sensor. Second, Increase the number of transferred packets from the specific sensor to a specific gateway. Third, it is based on gateway (as a coalition head) to construct a coalition, in which the constructed coalition is more stable compared with the grids. This stability gives the ability to deal with random mobile environments. Forth, the proposed framework is able to deal with random mobile sensors, regardless of locations.

Universiti Malaya

The extensive simulation results of Dynamic-Coalition framework showed that it is the superior state-of-the-art framework in reducing disseminated packets, in terms of (i) Routing Overhead and (ii) packets delivery ratio. The values of the packets delivery ratio of the proposed model are the highest. This indicates that the number of packets that are transferred from the same sensor to a specific gateway is high. Also, the proposed model has minimum Routing Overhead values. This indicates that the number of transferred packets from sensors to destinations is mitigated. Accordingly, this evaluation phase proved that the disseminated packets are mitigated.

So far, this chapter presents the evaluation of the disseminated packets problem. The next chapter presents the effect of dynamic-Coalition framework and Static-Coalition algorithm on the Coalition-Based Index-tree framework.

Universiti Malaya

CHAPTER 5:

INDEXING FRAMEWORK FOR REPLACING THE DEPENDABILITY ON GRID: IMPLEMENTATION AND EVALUATION

The evaluation process started in Chapter 4, in which evaluated the dynamic-Coalition framework. This chapter is a continuation of the previous one, which represents the evaluation Coalition-Based Index-Tree framework, by examining the role of replacing grids of network area with the coalition on the indexing performance. Furthermore, this chapter examines the effects of mitigation of the disseminated packets and arranging packets on the performance of indexing.

As mentioned, the experiment design of this chapter is represented in Section 4.1. The sequence of this chapter arranges as follows. Section 5.1 represents the evaluation metrics. Section 5.2 presents the evaluation results of Coalition-Based Index-Tree. The summary and discussion of this chapter present in Section 5.3.

5.1 Evaluation Metrics

The evaluation metrics used in this evaluation phase are space-cost overhead, index-building time, and the number of traversed nodes. A brief description of the used metrics is as follows.

- **Space-cost overhead**: is represented as a total size that occupied by the index after completing the building. Thus, the total Space-Cost (Sc) can be calculated by the following formula (Chen, N. 2015) (He et al. 2015):

  $$S_c = \mathrm{nLN} \times (\mathrm{Ns} + \mathrm{Ps}) \times \mathrm{Cn} + \mathrm{LN} \times (\mathrm{Ns} + 3 \times \mathrm{Ps}) \times \mathrm{Cn} + 4 \times \mathrm{S} \times (\mathrm{Ns} + \mathrm{Ps})$$

  Where: nLN is a number of non-leaf nodes, Ns is Size of Double float number, Ps is the size of pointers, Cn is the capacity of each node, LN is the number of Leaf nodes, S is the total number of moving sensors.

Universiti Malaya

- **Index building time Overhead**: is the total time required to build the index in the server.
- **The number of traversed nodes:** defined as a number of nodes that need to pass in order to build the tree structure.

Table 5.1 provides the list of metrics evaluated along with the units.

Table 5.1: Evaluation Metrics and units

#	Metric	Unit
3	Space cost overhead	MByte
4	Index building time	Seconds
5	The number of traversed nodes	Number of passed node

5.2 Coalition-Based Index-Tree evaluation

This section displays Coalition-Based Index-Tree evaluation and discusses its benefits that are added by the proposed framework. Thereby, the evaluation of Coalition-Based Index-Tree is constructed based on three scenarios. First, evaluate the performance of Coalition-Based Index-Tree by improving it with Dynamic-Coalition framework. Second, evaluate the performance of Coalition-Based Index-Tree by improving it with Static-Coalition algorithm. Third, evaluate the performance of Coalition-Based Index-Tree alone as an indexing technique, independently. However, the mentioned scenarios are conducted in terms of (i) index building time overhead, (ii) space-cost overhead, and (iii) number of traversed nodes.

5.2.1 Scenario 1: Effect of Dynamic-Coalition framework on Coalition-Based Index-Tree

This scenario examines the performance of Coalition-Based Index-Tree, by examining the effect of Dynamic-Coalition on it. Furthermore, the results of this subsection reflect the effects of minimizing the disseminated packets throughout gateways to improve Coalition-Based Index-Tree.

Universiti Malaya

5.2.1.1 Index building time overhead

Figure 5.1 shows index-building time overhead according to the radius. As defined before, index building time overhead is the time required for indexing the received packets. The figure shows the required time to build the proposed index structure is less than the building time of GDCF, R-tree, and D-tree structures. The superiority of the proposed index refers to the following reasons. It enables to group all packets that transferred from the same sensor together, which enables minimizing the time of reading each packet for indexing. Further, the proposed index-tree does not need further computation, to find the appropriate position of each packet in the tree-structure, which results in minimizing index building time. Moreover, the proposed index-tree is completely based on the source of packets. Besides, it has the ability to arrange and manage the received packets based on the source.

The failure of the proposed competitors refers to many reasons. First, the dependability of partitioning the network area, in order to build an index structure. This adversely affects the index building time. These adverse effects are caused by the nature of random mobile sensors, which change location dynamically. Meanwhile, other comparators consider each sensor a new one when it is changed its grid (move out of its grid). This results in increasing the burden of reading the attributes of the same sensor many times, which results in further time consumption.

Universiti Malaya

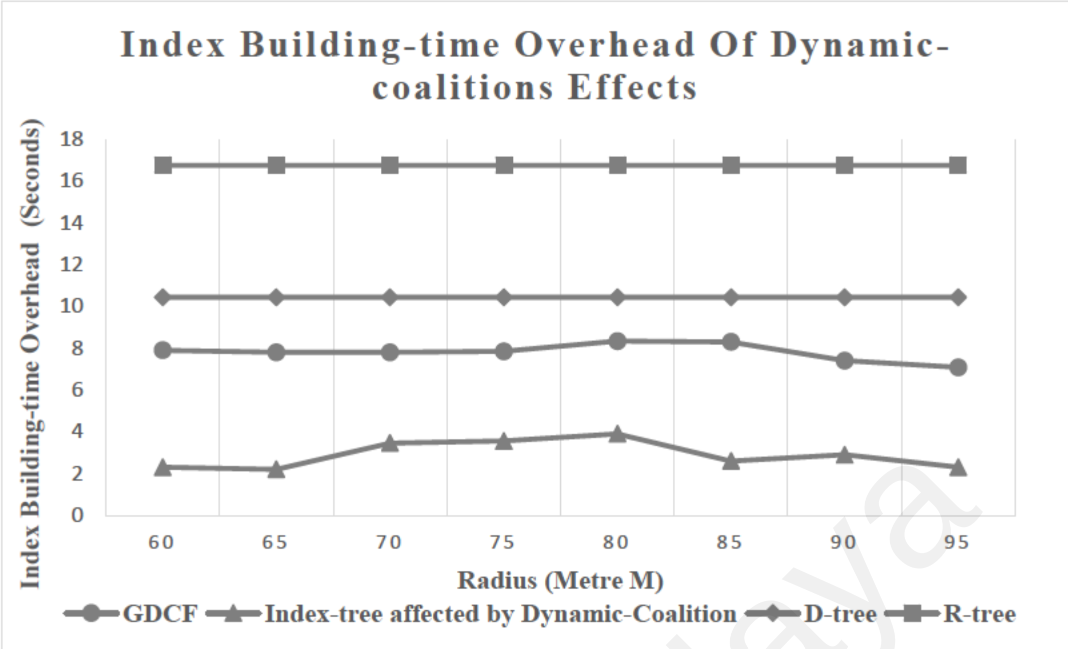

Figure 5.1: Index-building time overhead of Coalition-Based Index-Tree affected by dynamic-coalitions compared with the baselines (GDCF, R-tree, and D-tree)

As noted in the figure, the building time values of the index-tree are the smallest, when the radius is 60 and 65. This refers to the probability of the same sensor that transferred from the same location is high. Also, the values are increased in 70, 75, and 80. This indicates the probability is low. Further explanation, the probability of the redundancy of sensors that transfer from the same location and time (redundancy in time and location attributes) when the radius is 60, and 65 are higher than when the radius is 70. However, the proposed index-tree is performing the best when the similarity of the transmitted location is high. On the other side, the values of GDCF are the highest in the case of radius are 80 and 85. This indicates that sensors that change grids are high, which affects adversely indexing building time. Furthermore, the merging overlapping in GDCF is high, because of the redundancies. The index building time for R-tree and D-tree is not affected by the radius. The worst index-building time values are R-tree, because of high dependability on the attributes of the packets.

Further experiments are constructed according to the number of sensors, represents in Figure 5.2. The values are approximately similar when the number of sensors is 100.

Universiti Malaya

Meanwhile, the superiority of the proposed tree appears when the number of sensors is 900. This indicates the suitability of the proposed index to big-data.

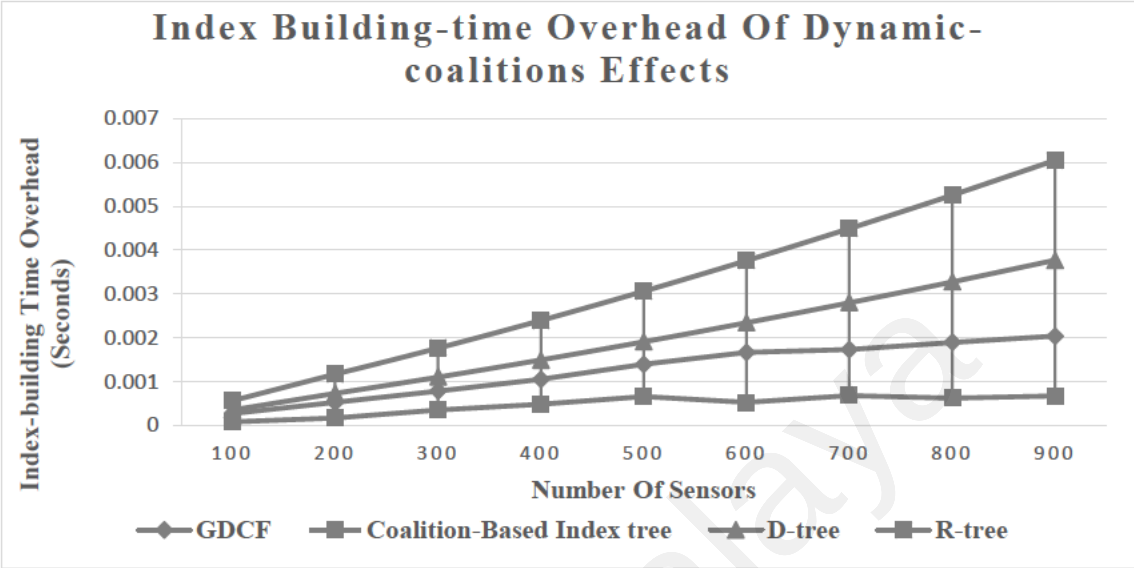

Figure 5.2: Index building time overhead of Coalition-Based Index-Tree in accordance with the number of sensors compared with the baselines (GDCF, R-tree, and D-tree)

5.2.1.2 Space-Cost Overhead

Space-cost overhead defines as the total amount of space occupied by the index structure after its construction. In order to evaluate space-cost of the proposed tree, the following parameters are used: memory size; the number of the moving objects; the number of the non-leaf nodes in a tree index; the number of the leaf nodes in a tree index; maximum capacity of each node; depth of the tree; the size of a double float number; the size of a pointer; the size of an integer; dimension of movement data (He et al. 2015).

The proposed tree has no inner-nodes, therefore, the number of non-leaf nodes are equal to zero. Each leaf node represents a packet value and the pointer that represents the maximum capacity of each node. Also, in visual C# 2017, the size of a float is equal to the size of pointers that equal to 4-bytes in a 64-bit process.

Figure 5.3 shows the average space-cost overhead –Y-axis-, according to different radius values. As demonstrated in the figure, the proposed index-tree is not much sensitive to add value and comparable compared to GDCF. As noted, GDCF values are the highest, because it contains inner-nodes that increase the volume storage of the index in memory, in addition to that it considers each sensor in a new location as a new one.

Furthermore, R-tree values are the worst, because it contains an inner-nodes. Indeed, it does not have the ability to remove redundancies. Although D-tree does not have inner-nodes, it occupies a high space volume compared with index-tree. This is because it is based on the moving sensors, which affect the index space. Furthermore, the capacity of each node is fixed for all D-tree that affects negatively the constructed index-space.

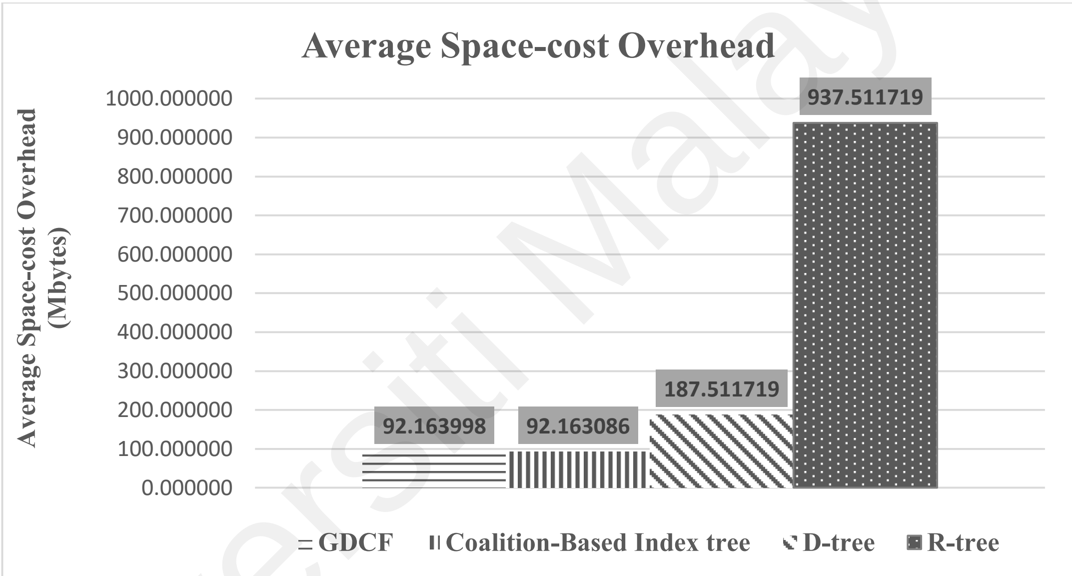

Figure 5.3: Average Space-Cost of Coalition-Based Index-Tree compared with the baselines (GDCF, R-tree, and D-tree)

Figure 5.4 represents the space-cost overhead according to the number of sensors. The index-tree is compared with GDCF (Boonchoo et al. 2019), R-tree (Antonin Guttman 1983), and D-tree (Chen, N. 2015). The analysis of this figure is as mentioned.

Universiti Malaya

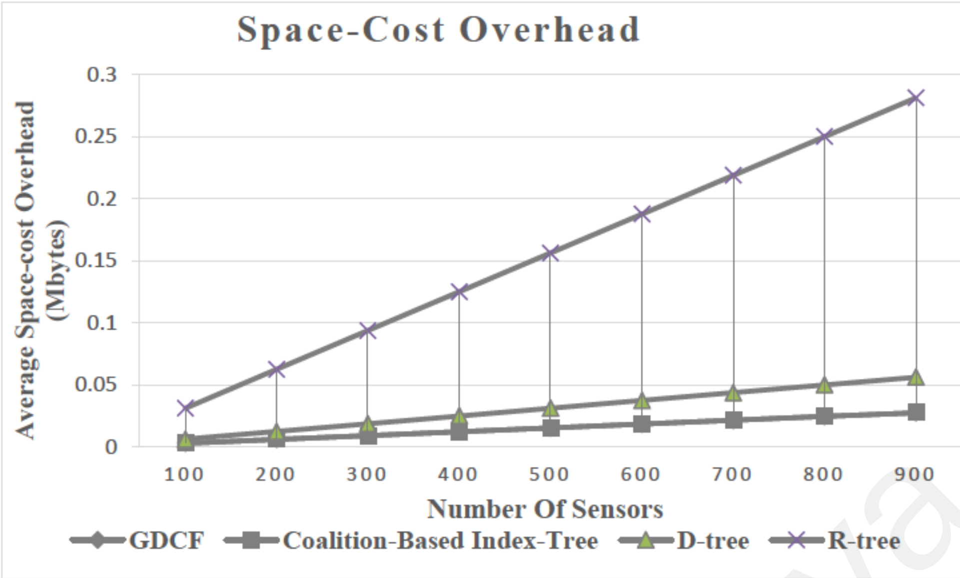

Figure 5.4: Space-cost overhead of the Coalition-Based Index-Tree compared with the baselines (GDCF, R-tree, and D-tree), in accordance with the number of sensors

5.2.1.3 Number of traversed nodes

As mentioned before, this metric defines as a number of passed index-tree nodes in order to build the structure of the tree according to the coming packets. The small value of traversed nodes would optimize the performance of the index.

Figure 5.5 shows the average of traversed node's number of the index-tree, GDCF, R-tree, and D-tree. As in the figure, the vulnerability in GDCF pops-up from the dependability on the classification of packets. GDCF considers each packet as a new value, which results in increasing the burden of reading and updating. Furthermore, GDCF needs to identify the received packets based on non-arranged packets. In other words, it requires reading from a large number of received packets to insert the packets to the specific location in the index structure. All the mentioned lead to increasing the traversed nodes required to build or update the index structure.

The vulnerability in R-tree and D-tree pops-up from the dependability on the classification of packets, i.e. MBR in the case of R-tree. These techniques consider each packet as a new value, which results in increasing the burden of reading and updating because the generated packets are large. Furthermore, R-tree and D-tree index techniques need to identify the received packets based on non-arranged packets. In other words, it requires reading packets from a large number of received packets to insert the packets to the specific location in the index structure, which affects the number of touched nodes in order to build or update the index.

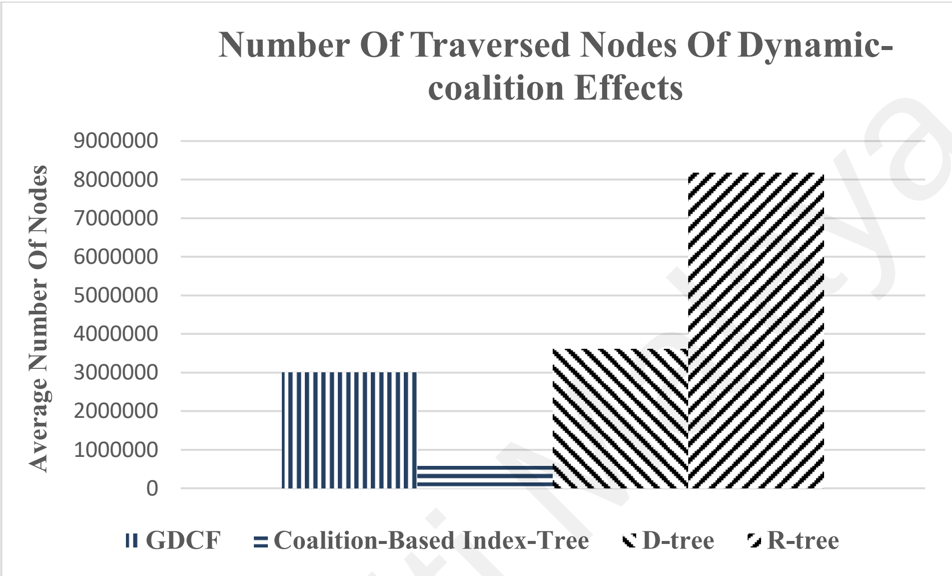

Figure 5.5: Average number of traversed nodes of Coalition-Based Index-Tree improved by Dynamic-Coalition compared with the baselines (GDCF, R-tree, and D-tree)

More evaluation is conducted according to the number of sensors, presents in Figure 5.6. The figure indicates the suitability of the proposed index tree to the big-data. The explanation for this superiority is mentioned before.

Universiti Malaya

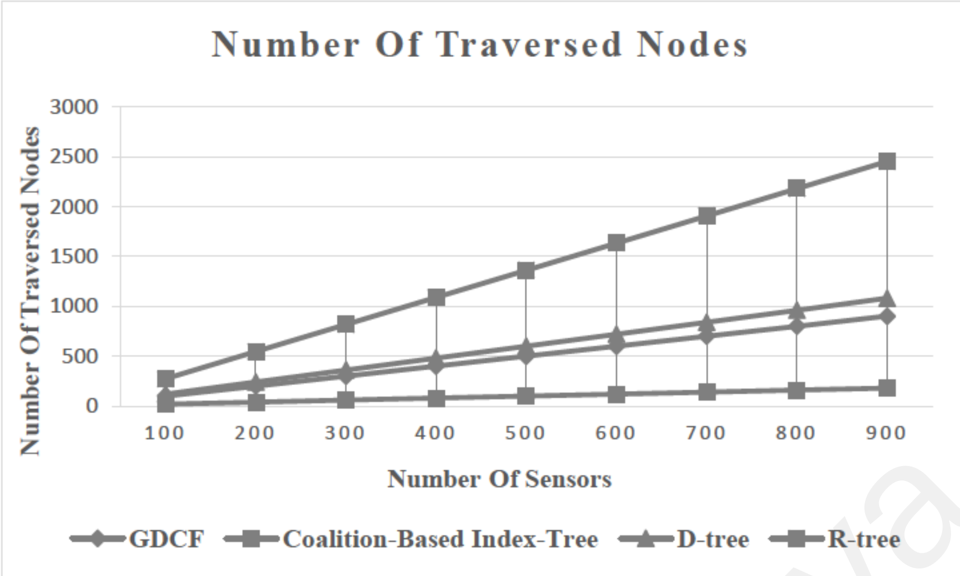

Figure 5.6: Number of traversed nodes of Coalition-Based Index-Tree compared with the baselines (GDCF, R-tree, and D-tree) in accordance with the number of sensors

5.2.2 Scenario 2: The effect of Static-Coalition on indexing

This scenario presents the evaluation platform that assesses the performance of Coalition-Based Index-Tree by improving it with Static-Coalition algorithm. The importance of this comparison is to evaluate the role of Static-Coalition algorithm in improving Coalition-Based Index-Tree. Thereby, Coalition-Based Index-Tree is evaluated by comparing with GDCF in terms of (i) index-building time overhead and (ii) traversed nodes. Space-cost overhead is not represented here, because there are no additional effects, by comparing with the previous scenario.

5.2.2.1 Index building time overhead

Figure 5.7 shows the time taken for building Coalition-Based Index-Tree improved by Static-Coalition, and compared with GDCF. As defined before, Index building time overhead is the time required for building the node's structure, in order to index the received packets. The results shown in the figure indicate that constructing Coalition-Based Index-Tree structure takes less time than constructing GDCF. The superiority of the proposed index-tree is referring to the existence of Static-Coalition. It minimizes the index building time because of the following reasons. First, it minimizes the number of received packets to the server. Because in the existence of Static-Coalition the packets are transferred as blocks called static-coalitions. This will minimize the time required to read the packet's attributes, in order to build the index structure. Second, Static-Coalition groups the packet's attributes that belong to the same sensor together. Thereby, it minimizes the time of reading each packet for indexing. Moreover, Coalition-Based Index-Tree does not need further computation as Hash-Table, to find the appropriate position in the tree-structure. Accordingly, it minimizes the overhead of index building time.

Again, taking a look at Figure 5.7 noted the following. First, the value of the index time of Coalition-Based Index-Tree when the value of the radius is 65 is the lowest. This indicates that the number of similarities between the locations of the sensors is less than others. Namely, the framework is performing the best in case of the similarity of the transmitted location is high.

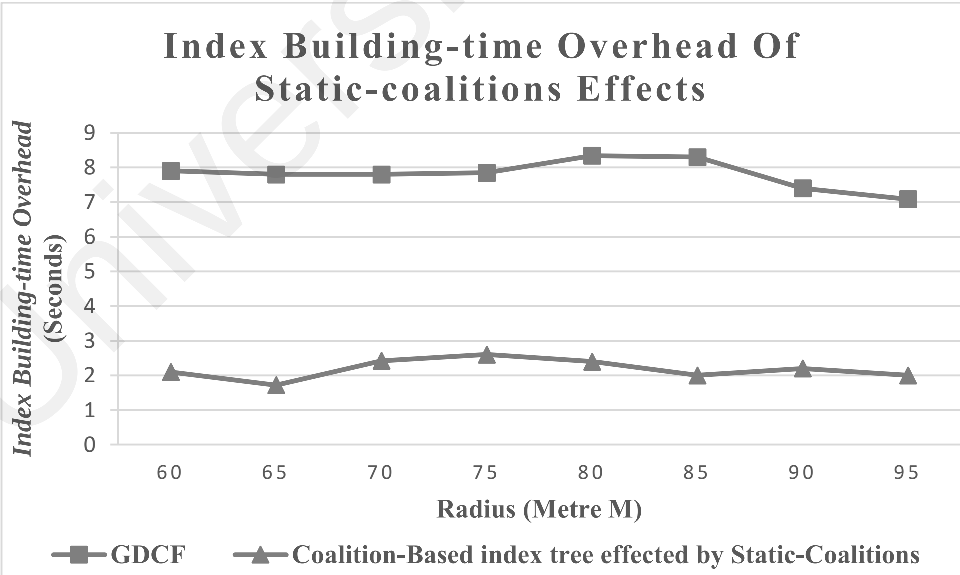

Figure 5.7: Index-building time overhead of Coalition-Based Index-Tree affected by Static-Coalitions compared with GDCF (Baseline)

5.2.2.2 Number of traversed nodes

One of the objectives of the proposed model is to minimize the number of index insertion operations by minimizing the number of traversed nodes while building the index structure. This is achieved by calculating the number of traversed nodes, which is used to estimate the number of passed nodes in order to insert a specific node for index building.

Figure 5.8 represents the number of traversed nodes of Coalition-Based Index-Tree compared with GDCF. Coalition-Based Index-Tree outperforms GDCF. The superiority of Coalition-Based Index-Tree emanates from the way of the building and updating of the indexing process. Accordingly, there are many factors that contributed to the superiority of Coalition-Based Index-Tree. First, the packets that the server received for indexing have already been arranged. The packets arrangement is adaptive in the same way of building index-structure, according to the Static-Coalition. Second, to start the indexing process, it is required to read all the received packets. The required readings help to identify the packets' position in the index-tree structure. Accordingly, instead of looking for the packet in a large group of received packets, Coalition-Based Index-Tree just reads from the small number of packets. This is because each one is labeled as $CoalitionId$ and all the packets that belong to the same $CoalitionId$ will be included in this node. Third, inside each node of the Coalition-Based Index-Tree, it goes directly to the appropriate position. Because it is based on Coalition, as described before. So, based on these factors, the number of passed nodes of Coalition-Based Index-Tree is minimized.

The vulnerability in GDCF pops-up from the dependability on the classification of the packets. Because of the random mobility nature of sensors, GDCF considers each packet as a new value which results in increasing the burden of reading and updating because the generated packets are large. Furthermore, GDCF index technique needs to identify the received packets based on non-arranged packets. More explicitly, it requires reading packets from a large number of received packets to insert the packets to the specific location in the index structure, which adversely affects the number of passed nodes.

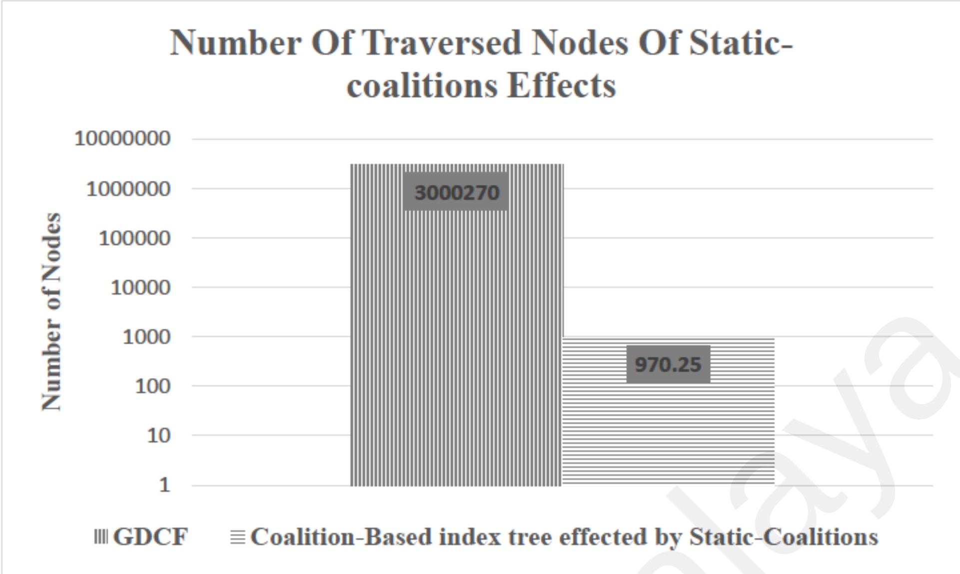

Figure 5.8: Average number of traversed nodes of Coalition-Based Index-Tree effects by Static-Coalition compared with GDCF (Baseline)

5.2.3 Scenario 3: Coalition-Based Index-Tree Effectiveness

This scenario examines the performance of the proposed index alone without any intervention. The evaluation carried on Coalition-Based Index-Tree to evaluate its performance as an indexing technique. Coalition-Based Index-Tree was evaluated in terms of index-building time overhead and the number of traversed nodes. Space-cost overhead is not represented here, because there are no additional effects, by comparing with the previous scenario.

5.2.3.1 Index Building Time Overhead

This part of the evaluation aims to prove that arranging packets by Static-Coalition algorithm and Dynamic-Coalition framework improve Coalition-Based Index-Tree with respect to index building time. Figure 5.9 plots the index building time overhead in second under the radius values {60, 65, 70, 75, 80, 85, 95, 95}. Furthermore, the figure compares Coalition-Based Index-Tree affected by Static-Coalition algorithm, Coalition-Based Index-Tree affected by Dynamic-Coalition framework, Coalition-Based Index-Tree alone

Universiti Malaya

without any intervention, and off-coalition. Off-coalition means the indexing process without the existence of the coalition.

As observed in the figure, static-coalitions have added tangible effects on the effectiveness of tree-based indexing. These tangible effects have referred to many reasons. First, arranging the packets in each gateway leads to minimizing the time of building the index structure. Second, the index processes are based on Static-Coalitions, because they stand for adapting the packets according to the way of tree-based indexing, which results in minimizing index-building time.

The effects of static-coalition on the proposed tree are not valuable when the radius value is 80. This is because of the increasing number of static-coalitions because of the minimum location of sensors redundancies. On the other side, Static-Coalition added the value when the value is 65, this refers to high redundancies in the sensor's locations.

The off-coalition has the highest values, because it deals with each packet independently, which results in increasing the index building time. Also, Coalition-Based Index-Tree affected by dynamic-coalition has added tangible effects on the effectiveness of the proposed index-tree. These tangible effects have referred to many reasons. First, the alleviation of the disseminated packets leads to minimizing the time of building the index structure. Second, the index processes are based on dynamic-coalitions, because they stand for adapting the packets according to the way of tree-based indexing, which results in minimizing index-building time. Third, it is completely based on the source of getaways and it is arranged and manage the transferred packets based on the source of the gateway, according to dynamic-coalition.

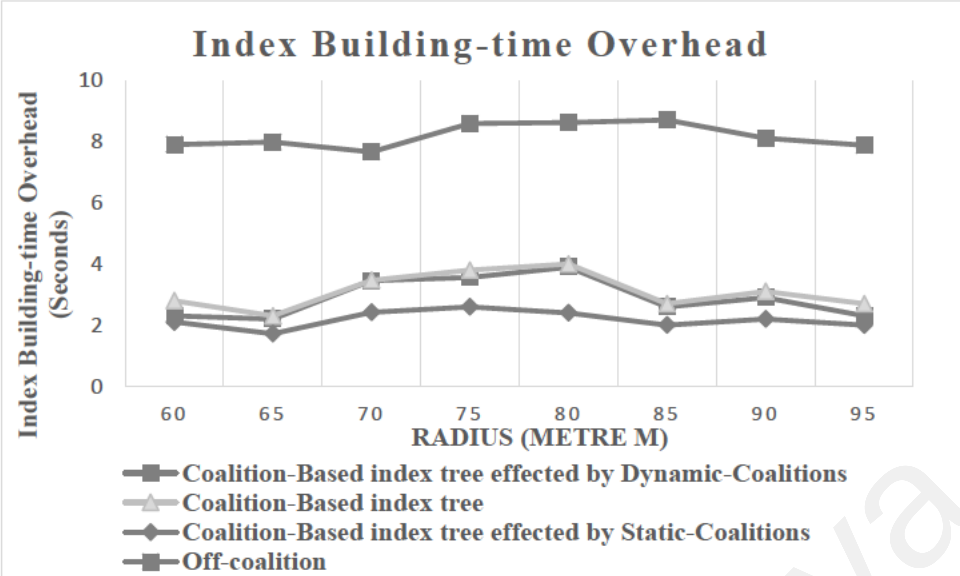

Figure 5.9: Index building time overhead of Coalition-Based Index-Tree

5.2.3.2 Number of traversed nodes

Figure 5.10, represents the number of traversed nodes of Coalition-Based Index-Tree compared with the role of Dynamic-Coalition and Static-Coalition in improving indexing in terms of the number of traversed nodes. Accordingly, mitigation of the disseminated packets improves the indexing with respect to traversed nodes. Furthermore, the proposed framework is based on Dynamic-Coalitions constructed based on the location of each gateway with a different period of time. This will minimize the dependability of the packets' attributes for each packet.

The main effects of static-coalition as the following. Static-coalitions arrange the packets according to transferring source, this leads to mitigate the dependability on the packets' attributes. One more superiority, static-coalitions are transferred to the server as blocks, where arranges the packets in the same way Coalition-Based Index-Tree constructs its nodes. Further explanation, instead of reading each packet alone in order to build the tree structure, it reading from blocks that contain all the packets that transferred from the same source.

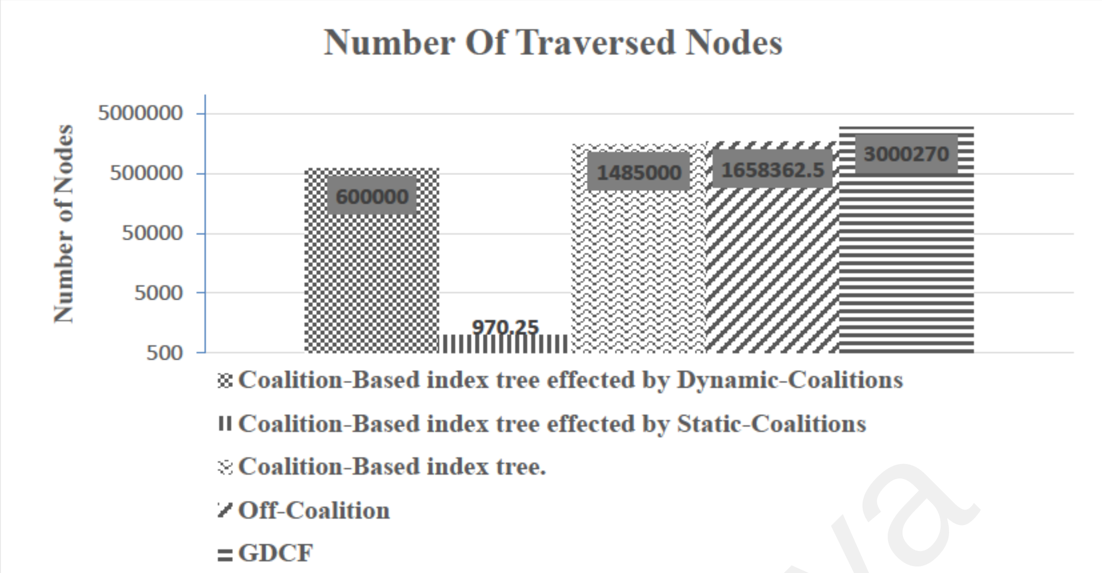

Figure 5.10: Average number of traversed nodes of Coalition-Based Index-Tree affected by Dynamic-Coalitions and Static-Coalitions, pure Coalition-Based Index-Tree, Off-Coalition Coalition-Based Index-Tree, and GDCF (Baseline)

5.3 Chapter Discussion and Summary

This chapter presented Coalition-Based Index-Tree framework, to replace the dependability on packets attributes and grids for building index structure. The proposed framework can replace this dependability by utilizing the coalition concept to build the indexing structure. Furthermore, this framework is improved by Dynamic-Coalition framework and Static-Coalition algorithm, as mentioned in Chapter 3 and Chapter 4. The extensive evaluation results of Coalition-Based Index-Tree framework showed that it is superior in replacing the dependability on packets attributes and grids, which results in improving the packets' indexing performance in random mobile WSN, in terms of space-cost overhead, index-building time, and the number of traversed nodes.

In all evaluation scenarios, the index-building time is comparable. This refers to the dependability of Coalition-Based Index-Tree framework on the coalition, in which it considers each coalition as a parent node in the index structure, and the children are its member sensors. Accordingly, there is no need to check the attributes of each packet to

Universiti Malaya

builds the index structure. One more reason is the role of Static-Coalition Algorithm that arranges the packets in the same way of indexing and transfers packets as blocks.

The number of the traversed nodes is reduced because of the existence of Static-Coalition Algorithm that arranges the packets according to its source as a coalition, and transfers them as blocks. This results in consider each block as one value according to the source of these packets. Furthermore, the space cost overhead is superior because the redundant (duplicated) value is removed in Static-Coalition Algorithm.

To this point in this work, the overall ideas of this thesis were included. The subsequent chapter will present the overall conclusion of this thesis.

Universiti Malaya

CHAPTER 6:

CONCLUSION AND FUTURE WORK

This chapter presents a comprehensive conclusion of this thesis and highlights future research trends. Thereby, this chapter starts with the thesis summary to connect the ideas of the thesis together. Furthermore, this chapter re-examines and illuminates the research questions and objectives by revisiting them, to ensure the mentioned work of thesis achieved them. Moreover, this chapter presents the importance of this study by shedding a light on its contributions. Eventually, this chapter ends with a discussion of research limitations and possible future directions.

The rest of this chapter organizes as follows. First, the summarization of this thesis presents in Section 6.1. Second, taking a tour to revisit the research objectives presents in Section 6.2. Third, the contributions are presented in Section 6.3. The fourth section discusses the limitations of this research. Finally, some suggestions on the potential enhancement to the proposed framework are provided for future research work presents in Section 6.4.

6.1 Summary of this thesis

A comprehensive literature review is investigated to understand the indexing techniques of random mobile packets. An in-depth investigation of the literature review has successfully identified and analyzed the techniques in two folds. First, it analyzed and discussed the current techniques applied to mobile sensors, to identify the effects on the indexing process. Second, a deep investigation of the current indexing techniques dedicated to mobile environments, to find challenges and gaps. Accordingly, during the investigation, the role of disseminated packets in random mobile environments in increasing the burden of the indexing process is highlighted. Furthermore, the challenges

Universiti Malaya

of Grid-based indexing were identified and discussed. As a result, to tackle the existing challenges, a novel model is proposed called DySta-Coalition.

The main aim of DySta-Coalition model is to improve the indexing process in random mobile environments. It addresses the negative role of the packets' dissemination on the performance of indexing in random mobile WSN. More explanation, DySta-Coalition model improves the indexing process by mitigating the disseminated packets. Thereby, DySta-Coalition model bases on exploiting WSN's nodes, sensors, gateways, and server. Accordingly, DySta-Coalition model is divided into three complementary phases. Dynamic-Coalition framework, Static-Coalition algorithm, and Coalition-Based Index-Tree framework, which implemented in sensors, gateways, and server, respectively.

Dynamic-Coalition framework indicates that dynamic-coalitions keep constructed as long as sensors and gateways are moving. Furthermore, by implementing it, the number of transferred packets by sensors will be alleviated. Indeed, the number of packets received by gateways from a specific source increased.

Static-Coalition algorithm constructed in each gateway according to the dynamic-coalitions. It stands to manage and arrange the received packets step-by-step in the gateway before accumulated in the main server for indexing. Meanwhile, Coalition-Based Index-Tree framework is a tree-based indexing technique, dedicated to random mobility, and its structure is without inner-nodes.

This thesis conducted an analysis of the problem. The problem analysis helps to simplify the understanding of the problem statement. Furthermore, it proves the relationship between the index effectiveness and the random behavior of the sensors. The metric used in the empirical analysis is relevancy-measure. It has the ability to test the stability of each mobile sensor in the constructed coalition, in order to refute the existing works of Grid-based indexing.

Universiti Malaya

The evaluation results showed DySta-Coalition model is superior, compared with the best-known competitors. Evaluating the proposed model conducted by implementing it using a simulator, implemented using Visual C# 2017 (Ghaleb et al. 2017). The evaluation setup consists of a randomly deployed number of sensors and gateways, with random movement. Subsequently, DySta-Coalition model was evaluated according to the following phases. First, examining the validity of Dynamic-Coalition framework and Static-Coalition algorithm. Namely, in terms of Routing Overhead and Packet Delivery Ratio. Second, examining the validity of Coalition-Based Index-Tree framework. Namely, in terms of index building time overhead, space-cost overhead, and the number of traversed nodes. Thereby, an evaluation of the proposed index tree is conducted based on two scenarios. The first one evaluated Coalition-Based Index-Tree as an indexing technique to ensure its performance. The second scenario examined the effect of Dynamic-Coalition framework and Static-Coalition algorithm on Coalition-Based Index-Tree.

6.2 Research objectives revisited

As mentioned, this work aims to improve the efficiency of indexing in random mobile WSN. In this section, the research question and objectives, which are stated in Chapter 1, Section 1.4, and Chapter 3, Subsection 3.2.3 will be reassessed after the research journey. Accordingly, this section reviews the questions to confirm if the evaluation results answered the research questions. Furthermore, it reviews the research objectives, also mentioned in Chapter 1 and Chapter 3, to check if this work achieved the targeted objectives. In the following, the detail of re-examining the research aim and objectives.

Question#1: What are the issues that adversely affect the indexing of random mobile sensors in WSN.

Objective#1: To study and identify the current Grid-based indexing techniques and how there have been affected by the behavior of random mobile sensors.

Universiti Malaya

The first objective was identifying and investigating the lately developed Grid-based indexing techniques. Furthermore, it was obtaining insight into the role of random behavior on the Grid-based indexing effectiveness, to identify the existing problems. This research was constructed according to a rigorous and thorough literature review, by considering major researches from well-established scholarly published databases worldwide; i.e. Springer publications, Web of Science, IEEE Explorer.

According to the literature review, a set of research issues have been specified. The extracted research issues elucidated that there are critical issues that obstruct the Grid-based indexing. The random mobility of the WSN's nodes caused disseminated packets throughout gateways, that negatively impact the Grid-based indexing performance. This problem was lately addressed through the elaboration of the proposed framework, Chapter 4.

Question#2: How to minimize the sensors Routing Overhead by transmitting to less number of directions?

Objective#2: To minimize random mobile sensor overhead, by initiating a relationship between source and destination.

This objective was analyzing and investigating the effect of minimizing disseminated packets throughout gateways by mitigating the number of transferred packets. The evaluation of Dynamic-Coalition framework is conducted. The evaluation helps a better understanding of the problem of disseminated packets is alleviated. The experiments help us better understand the problem, by establishing a relation between the transferred packet's number and the disseminated packets throughout destinations.

Question#3: How to mitigate the packets of the sensors from scrambled?

Universiti Malaya

Objective#3: To improve the index operations of the system, by maximizing the number of transferred packets from each sensor to a specific destination.

The third objective was finding the effects of maximizing the number of transferred packets from each sensor to each gateway, to mitigate the disseminated packets throughout gateways. As a solution to the identified problem, Dynamic-Coalition framework and Static-Coalition model are proposed. To examine that this objective is achieved, the evaluation was conducted according to the Packet Delivery Ratio.

Question#4: How to minimize the number of indexing operations, which affected by the disseminated packets?

Objective#4: To reduce the number of indexing operations, in order to minimize the number of traversed index nodes during index construction.

This objective and objective number five are dedicated to assessing the performance of indexing. The fourth object reducing the number of index operations by minimizing the number of traversed nodes of the index structure.

Question#5: How to reduce indexing operations overhead for data transmitted from gateways?

Objective#5: To effectively build-up static-coalitions in each gateway to reduce the number of indexing operation overhead in terms of space-cost and index building time.

Question#6: What is the performance of the proposed model against others.

Objective#6: To validate the proposed model using practical analysis and compare its performance with the best-known competitors.

The last objective was evaluating the performance of the proposed solution, by considering several metrics. Namely, Packet Delivery Ratio, Routing Overhead, number of traversed nodes, index building-time, and space-cost overhead. To this end, a simulator that implemented using Visual C# 2017 (Ghaleb et al. 2017). The results were used to validate the proposed model, which showed superiority compared with

Universiti Malaya

the results of competitors. The performance of DySta-Coalition model was evaluated, and it can significantly outperform the classical techniques R-tree and D-tree and the recent one GDCF in terms of index building-time overhead, a number of traversed nodes, and space-cost overhead. This superiority is due to its ability to deal with received packets as arranged blocks regardless of transferring time and location.

6.3 Contributions

The illuminating contributions of this work are summarized as follows:

First, **Dynamic-Coalition** framework is proposed to minimize the number of disseminated packets throughout gateways, by constructing dynamic clusters based on the coalition, called **dynamic-coalitions**.

Second, **Static-Coalition** algorithm is proposed, by initiating **static-coalitions** in each gateway that alleviates the effects of the Grid-based index structure and removing the redundancy of packets, which results in arranging and preparing packets step-by-step before accumulated in the final server.

Third, a variant Indexing-tree, called **Coalition-Based Index-Tree** is proposed, which replaces the Grid-based index structure by dynamic-coalition. Table 6.1 summarizes the research mapping for this work.

Table 6.1: Research mapping of this thesis

#	Research Question	Research Objective	Methodology	Outcome
1	RQ1: What are the issues that adversely affect the indexing of random mobile sensors in WSN.	O1: To study and identify the current Grid-based indexing techniques and how there have affected by the behavior of random mobile sensors.	Conducting a paper that presents deformities of indexing processes in mobile WSN	Review paper

Universiti Malaya

2	RQ2. How to minimize the sensors Routing Overhead by transmitting to less number of directions?	O2. To minimize random mobile sensor overhead, by initiating a relationship between source and destination.	Developing a framework that minimize the disseminated packets based on the coalition.	Dynamic-Coalition framework.
	RQ3. How to mitigate the packets of the sensors from scrambled?	O3. To maximize the number of transferred data to each destination, by mitigating sensor's position collision problem.	Developing Coalition-based algorithm that arranges packets.	Static-Coalition algorithm.
2	RQ4. How to reduce indexing overhead for data transmitted from gateways?	O4. To reduce the number of indexing operations, in order to minimize the number of traversed index nodes during index construction.	Developing a novel Coalition-Based Index-Tree framework.	Proposed a variant Indexing-tree, called Coalition-Based Index-Tree, which replaces the Grid-based with dynamic-coalition.
	RQ5. How to minimize the number of indexing operations, that affected by the disseminated packets?	O5. To effectively build-up static-coalitions in each gateway to reduce the number of indexing operation overhead in terms of space-cost and index building time.		
3	RQ6: What is the performance of the proposed model against others?	O6: To validate the proposed model using practical analysis and compare its performance with	Evaluating DySta-Coalition framework	Proposed a simulator, used to evaluate the proposed model.

Universiti Malaya

		the best-known competitors.		

6.4 Limitations

Although the work of this thesis has managed to achieve its objectives, there are some inevitable limitations in this research. First, the optimum case of the proposed model is when the repetition of the sensor's locations is high. Second, DySta-Coalition model is dedicated to homogeneous data, where it can not deal with heterogeneous. In future work, we will discuss the proposed index and the efficiency of retrieving data from it, in addition to the limitations of disk size.

6.5 Recommendations for Future Research Work

Many Ph.D. studies have been performed to improve indexing processes in random mobile WSN. However, going beyond the boundaries of any research topic is not sufficient for a single Ph.D. study. Consequently, this research highlights ideas on a number of potential guidelines upon which supplementary research can be conducted on the basis of current research outcomes. The effectiveness of the indexing process was achieved through intensive simulation. The effect of significant parameters was measured, and the result was presented in gaps to simplify the addressing of gaps between comparable baselines.

The motivation of this research is to make a balance by reducing the number of transferred packets that are received by gateways from different sensors and increase the received packets by gateways from the same sensor. This balance improves the indexing processes of sensors' packets in a random mobile environment, by mitigating the dependability of the packets' attributes. Hence, the future direction of this research includes extending and improving the outcome of this research.

The direction of future research can be considered by including the proposed model with respect to the efficiency of query retrievals, such as precision, recall, and F-measure. Furthermore, improving the way of collecting data by applying machine-learning, to improve the index performance. Additional work can be considered by applying the proposed model over more complex environments, such as IoT, cloud computing, etc. Besides that, further work to connect the enhancement of the proposed work over environments that produce the heterogeneous data.

Universiti Malaya

REFERENCES

Abbasifard, Mohammad Reza, Hassan Naderi, and Omid Isfahani Alamdari. 2017. "Efficient Indexing For Past and Current Position of Moving Objects on Road Networks." *IEEE Transactions on Intelligent Transportation Systems* 19: 1–12.

Abdul, Shaik, Nusrath Begum, and K P Supreethi. 2018. "A Survey on Spatial Indexing." *Journal of Web Development and Web Designing* 3(1): 1–25.

Ahmed, Pritom et al. 2017. "Efficient Computation of Top-k Frequent Terms over Spatio-Temporal Ranges." In *SIGMOD*, , 1227–41.

Al-Badarneh, Amer F., Qussai Yaseen, and Ismail Hmeidi. 2010. "A New Enhancement to the R-Tree Node Splitting." *Journal of Information Science* 36(1): 3–18.

de Almeida, Victor Teixeira, and Ralf Hartmut Güting. 2005. "Indexing the Trajectories of Moving Objects in Networks." *GeoInformatica* 9(1): 33–60.

Altti Ilari Maarala, Xiang Su, Jukka Riekki. 2016. "Semantic Reasoning for Context-Aware Internet of Things Applications." *IEEE INTERNET OF THINGS JOURNAL* 3(4): 1–13.

Alvarez, Victor, Stefan Richter, Xiao Chen, and Jens Dittrich. 2015. "A Comparison of Adaptive Radix Trees and Hash Tables." In *31st IEEE Int. Conf. on Data Engineering*, , 1227–38.

Amato, Flora et al. 2015. "SemTree: An Index for Supporting Semantic Retrieval of Documents." In *Proceedings - International Conference on Data Engineering*, , 62–67.

Antonin Guttman, Michael Stonebraker. 1983. "R-Trees: A Dynamic Index Structure for

Spatial Searching.” In *Encyclopedia of GIS*, , 1805–17.

Babenko, Artem, and Victor Lempitsky. 2015. “The Inverted Multi-Index.” *IEEE Transactions on Pattern Analysis and Machine Intelligence* 37(6): 1247–60.

Badarneh, Hazem Jihad, Sri Devi Ravana, and Ali Mohammed Mansoor. 2020. “A Survey on Indexing Techniques for Mobility in Internet of Things’: Challenges, Performances, and Perspectives.” *International Journal of Network Management* (e2097): 1–28.

Bassi, Corina. 2004. “Query and Update Efficient B + -Tree Based Indexing of Moving Objects.” : 1–11.

Bayer, R., and E. M. McCreight. 1972. “Organization and Maintenance of Large Ordered Indexes.” *Acta Informatica* 1(3): 173–89.

Beckmann, Norbert, Hans-Peter Kriegel, Ralf Schneider, and Bernhard Seeger. 1990. “The R*-Tree: An Efficient and Robust Access Method for Points and Rectangles.” *ACM SIGMOD Record* 19(2): 322–31.

Boonchoo, Thapana et al. 2019. “Grid-Based DBSCAN: Indexing and Inference.” *Pattern Recognition* 90: 271–84.

Bozkaya, Tolga, and Meral Ozsoyoglu. 1997. “Distance-Based Indexing for High-Dimensional Metric Spaces.” In *Proceedings of the 1997 ACM SIGMOD International Conference on Management of Data - SIGMOD ’97*, , 357–68. http://portal.acm.org/citation.cfm?doid=253260.253345.

Butz, Arthur R. 1969. “Convergence with Hilbert’s Space Filling Curve.” *Journal of Computer and System Sciences* 3(2): 128–46.

Universiti Malaya

Cao, Chun, Weiyi Wang, Ying Zhang, and Xiaoxing Ma. 2017. "Leveraging Column Family to Improve Multidimensional Query Performance in HBase." In *IEEE International Conference on Cloud Computing, CLOUD*, , 106–13.

Chakrabarti, Arnab, Ashutosh Sabharwal, and Behnaam Aazhang. 2003. "Using Predictable Observer Mobility for Power Efficient Design of Sensor Networks." In *Information Processing in Sensor Networks*, , 129–45.

Chen, N., et al. 2015. "Decomposition Tree: A Spatio-Temporal Indexing Method for Movement Big Data." *Cluster Computing* 18(4): 1481–92.

Chen, Jidong. 2007. "Indexing the Past, Present and Future Positions of Moving Objects in Urban Traffic Networks." *Tods* 44(212): 1008. jidong.

Chen, Jinchuan et al. 2013. "Big Data Challenge: A Data Management Perspective." *Frontiers of Computer Science* 7(2): 157–64. http://www.scopus.com/inward/record.url?eid=2-s2.0-84876005101&partnerID=40&md5=8bc0d2c00d407e15a6d37d1df34e5f42.

Chen, Su, Beng Chin Ooi, Kian-Lee Tan, and Mario A. Nascimento. 2008. "ST 2 B-Tree: A Self-Tunable Spatio-Temporal B+-Tree Index for Moving Objects." In *Proceedings of the 2008 ACM SIGMOD International Conference on Management of Data - SIGMOD '08*, , 29. http://portal.acm.org/citation.cfm?doid=1376616.1376622.

Cheng, Reynold et al. 2004. "Efficient Indexing Methods for Probabilistic Threshold Queries over Uncertain Data." *Vldb*: 876–87.

Cheng, Reynold, Dmitri V. Kalashnikov, and Sunil Prabhakar. 2004. "Querying Imprecise Data in Moving Object Environments." *IEEE Transactions on Knowledge* *and Data Engineering* 16(9): 1112–27.

Ding, Zhiming, Jian Dai, Xu Gao, and Qi Yang. 2012. “A Hybrid Search Engine Framework for the Internet of Things.” In *Proceedings - 9th Web Information Systems and Applications Conference, WISA 2012*, , 57–60.

Dittrich, Jens et al. 2010. “Hadoop++: Making a Yellow Elephant Run Like a Cheetah.” In *Proceedings of the VLDB Endowment*, , 515–29. http://dl.acm.org/citation.cfm?doid=1920841.1920908.

Dittrich, Jens, and Jorge-Arnulfo Quiané-Ruiz. 2012. “Only Aggressive Elephants Are Fast Elephants.” In *Proceedings Ofthe VLDB Endowment*, , 1591–1602. http://arxiv.org/abs/1208.0287.

Du, Chu et al. 2013. “An Effective IoT Services Indexing and Query Technique.” In *Proceedings - 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing, GreenCom-IThings-CPSCom 2013*, , 777–82.

Ebrahimi, Mohammad et al. 2016. “An Adaptive Meta-Heuristic Search for the Internet of Things.” *Future Generation Computer Systems* 76: 486–94.

Elhabyan, Riham S Y, and Mustapha C E Yagoub. 2015. “Two-Tier Particle Swarm Optimization Protocol for Clustering and Routing in Wireless Sensor Network.” *Journal of Network and Computer Applications* 52: 116–28. http://dx.doi.org/10.1016/j.jnca.2015.02.004.

Eylem Ekici, Yaoyao Gu, and Doruk Bozdag, Ohio State University. 2006. “Mobility-Based Communication in Wireless Sensor Networks.” In *IEEE Communications Magazine*, , 56–62.

Universiti Malaya

Fang, Y., J. Cao, C. Zeng, and N. Chen. 2012. “Indexing the Past, Present and Future Positions of Moving Objects Using PPFI.” *Proceedings - 2012 8th International Conference on Computing Technology and Information Management, ICCM 2012* 1(June): 223–35.

Fang, Ying et al. 2013. “Efficient Indexing of the Past, Present and Future Positions of Moving Objects on Road Network.” *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)* 7901 LNCS: 223–35.

Feng, Chen et al. 2015. “LCIndex: A Local and Clustering Index on Distributed Ordered Tables for Flexible Multi-Dimensional Range Queries.” In *Proceedings of the International Conference on Parallel Processing*, IEEE, 719–28.

Gani, Abdullah, Aisha Siddiqa, Shahaboddin Shamshirband, and Fariza Hanum. 2016. “A Survey on Indexing Techniques for Big Data: Taxonomy and Performance Evaluation.” *Knowledge and Information Systems* 46(2): 241–84.

Gärtner, Markus, Andreas Rauber, and Helmut Berger. 2014. “Bridging Structured and Unstructured Data via Hybrid Semantic Search and Interactive Ontology-Enhanced Query Formulation.” *Knowledge and Information Systems* 41(3): 761–92.

Ghaleb, Mukhtar et al. 2017. “A Performance Simulation Tool for the Analysis of Data Gathering in Both Terrestrial and Underwater Sensor Networks.” *IEEE Access* 5: 4190–4208.

Glover, Fred. 1990. “Paper_Glover_Ts_2.Pdf.” *ORSA Journal on Computing* 2(1): 4–32.

Gugnani, Shashank et al. 2018. “Characterizing and Accelerating Indexing Techniques on Distributed Ordered Tables.” In *Proceedings - 2017 IEEE International*

Universiti Malaya

Conference on Big Data, Big Data 2017, , 173–82.

Guo, Huadong, Lizhe Wang, Fang Chen, and Dong Liang. 2014. "Scientific Big Data and Digital Earth." *Chinese Science Bulletin* 59(35): 5066–73.

He, Zhenwen et al. 2015. "Decomposition Tree: A Spatio-Temporal Indexing Method for Movement Big Data." *Springer Cluster Computing* 18(4): 1481–92.

Hilbert, D. 1891. "Uber Die Stetige Abbildung Einer Linie Auf Ein Flachenstuck." *Math. Ann.* 38: 459– 460.

Huang, Menglong, Peng Hu, and Lanfang Xia. 2010. "A Grid Based Trajectory Indexing Method for Moving Objects on Fixed Network." In *18th International Conference on Geoinformatics, Geoinformatics 2010*, , 1–4.

Hubaux, Jun Luo and Jean-Pierre. 2010. "Joint Sink Mobility and Routing to Increase the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility." *IEEE/ACM Transactions on Networking* 18(3): 871—884.

Huo, Yan et al. 2016. "An Enhanced Low Overhead and Stable Clustering Scheme for Crossroads in VANETs." *Eurasip Journal on Wireless Communications and Networking* 2016(1). http://dx.doi.org/10.1186/s13638-016-0573-9.

Huo, Yan, Wei Dong, Jin Qian, and Tao Jing. 2018. "Coalition Game-Based Clustering Strategy for Crossroads in VANETs." In *Proceedings - 2016 International Conference on Identification, Information and Knowledge in the Internet of Things, IIKI 2016*, IEEE, 317–24.

Jo, Bumjoon, and Sungwon Jung. 2018. "Quadrant-Based Minimum Bounding Rectangle-Tree Indexing Method for Similarity Queries over Big Spatial Data in

Universiti Malaya

HBase." *Sensors* 18(9): 3032. http://www.mdpi.com/1424-8220/18/9/3032.

Kaisler, Stephen, Frank Armour, J. Alberto Espinosa, and William Money. 2013. "Big Data: Issues and Challenges Moving Forward." In *2013 46th Hawaii International Conference on System Sciences*, , 995–1004. http://ieeexplore.ieee.org/document/6479953/.

Khoshkangini, Reza, and Syroos Zaboli. 2014. "Efficient Routing Protocol via Ant Colony Optimization (ACO) and Breadth First Search (BFS)." In *International Conference on Internet of Things (IThings 2014)*, , 375–81.

Kim, Y.G., Yu Wang, B.S. Park, and H.H. Choi. 2015. "Heuristic Algorithms for Scheduling Resources in Time-Constrained Wireless Sensor Networks." In *2015 International Conference on Platform Technology and Service*, , 33–34. http://ieeexplore.ieee.org/document/7079628/.

Kim, Yang G., and Myung J. Lee. 2014. "Scheduling Multi-Channel and Multi-Timeslot in Time Constrained Wireless Sensor Networks via Simulated Annealing and Particle Swarm Optimization." *IEEE Communications Magazine* 52(1): 122–29.

Kim, Yang G, Yu Wang, ByoungSeob Park, and Hyo Hyun Choi. 2016. "A Heuristic Resource Scheduling Scheme in Time-Constrained Networks." *Computers & Electrical Engineering* 54: 1–15. http://www.sciencedirect.com/science/article/pii/S0045790616301562.

Kirkpatrick, S., C. D. Gelatt, and M. P. Vecchi. 1983. "Optimization by Simulated Annealing." *Science* 220(4598): 671–80. http://www.sciencemag.org/cgi/doi/10.1126/science.220.4598.671.

Lee, Dong Wook, Sung Ha Baek, and Hae Young Bae. 2009. "ACN-RB-Tree: Update

Universiti Malaya

Method for Spatio-Temporal Aggregation of Moving Object Trajectory in Ubiquitous Environment." *Proceedings of the 2009 International Conference on Computational Science and Its Applications, ICCSA 2009*: 177–82.

Lee, Jeongcheol et al. 2018. "Active Data Dissemination for Mobile Sink Groups in Wireless Sensor Networks." *Ad Hoc Networks* 72: 56–67.

Lee, Junhee, Wun Cheol Jeong, and Byeong Cheol Choi. 2016. "A Multi-Channel Timeslot Scheduling Algorithm for Link Recovery in Wireless Multi-Hop Sensor Networks." In *2016 International Conference on Information and Communication Technology Convergence, ICTC 2016*, , 871–76.

Lee, Ki Young, Jeong Jin Kang, and Jeong Joon Kim. 2014. "Indexing Method for Moving Sensor Node Retrieval." *Int. J. Sen. Netw.* 15(4): 238–45. http://dx.doi.org/10.1504/IJSNET.2014.064434.

Lee, Mong Li et al. 2003. "Supporting Frequent Updates in R-Trees." In *Proceedings 2003 VLDB Conference*, , 608–19. http://linkinghub.elsevier.com/retrieve/pii/B9780127224428500604.

Lehman, Tobin J., and Michael J. Carey. 1986. "Query Processing in Main Memory Database Management Systems." In *Proceedings of the 1986 ACM SIGMOD International Conference on Management of Data - SIGMOD '86*, , 239–50. http://portal.acm.org/citation.cfm?doid=16894.16878.

León, O., J. Hernández-Serrano, and M. Soriano. 2015. "IoT-SVKSearch: A Real-Time Multimodal Search Engine Mechanism for the Internet of Things." *International Journal of Communication Systems* 23(5): 633–52.

Li, Wei et al. 2014. "Efficient Allocation of Resources in Multiple Heterogeneous

Universiti Malaya

Wireless Sensor Networks." *Journal of Parallel and Distributed Computing* 74(1): 1775–78. http://dx.doi.org/10.1016/j.jpdc.2013.09.012.

Li, Yuanyuan, and Lynne E. Parker. 2012. "Nearest Neighbor Imputation Using Spatial-Temporal Correlations in Wireless Sensor Networks." *Information Fusion* 15(1): 64–79. http://dx.doi.org/10.1016/j.inffus.2012.08.007.

Lin, Bin, and Jianwen Su. 2005. "Handling Frequent Updates of Moving Objects." In *Proceedings of the 14th ACM International Conference on Information and Knowledge Management - CIKM '05*, , 493. http://portal.acm.org/citation.cfm?doid=1099554.1099691.

Lin, Dan, Christian S. Jensen, Beng Chin Ooi, and Simonas Šaltenis. 2005. "Efficient Indexing of the Historical, Present, and Future Positions of Moving Objects." In *Proceedings of the 6th International Conference on Mobile Data Management - MEM '05*, , 59. http://portal.acm.org/citation.cfm?doid=1071246.1071256.

Lin, Hung Yi. 2012. "Using Compressed Index Structures for Processing Moving Objects in Large Spatio-Temporal Databases." *Journal of Systems and Software* 85(1): 167–77. http://dx.doi.org/10.1016/j.jss.2011.08.005.

Liu, Zhao-hong, and Xiao-li Liu. 2005. "Indexing Large Moving Objects from Past to Future with PCFI+-Index." In *ADVANCES IN DATA MANAGEMENT 2005*, , 131–37.

Long, Jed A., and Trisalyn A. Nelson. 2013. "A Review of Quantitative Methods for Movement Data." *International Journal of Geographical Information Science* 27(2): 292–318.

Loscrí, Valeria, Enrico Natalizio, and Francesca Guerriero. 2012. "Particle Swarm

Universiti Malaya

Optimization Schemes Based on Consensus for Wireless Sensor Networks Categories and Subject Descriptors." In *MSWiM'12*,.

Luo, Shiliang, Xu Lu, and Lianglun Cheng. 2013. "TSOIA: An Efficient Node Selection Algorithm Facing the Uncertain Process for Internet of Things." *Journal of Network and Computer Applications* 36(2): 738–43.

Ma, Youzhong et al. 2012. "An Efficient Index for Massive IOT Data in Cloud Environment." In *CIKM '12: Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, , 2129–33. http://portal.acm.org/citation.cfm?id=2396761.2398587&coll=DL&dl=ACM&CFID=398282159&CFTOKEN=30246837%5Cnpapers3://publication/doi/10.1145/2396761.2398587.

MacNicol, Roger, and Blaine French. 2004. "Sybase IQ Multiplex - Designed For Analytics." In *Pvldb*, , 1227–30. http://dl.acm.org/citation.cfm?id=1316689.1316798.

Mann, Palvinder Singh, and Satvir Singh. 2016. "Performance Analysis of Energy-Efficient Routing Protocol for Wireless Sensor Networks." *IET Wireless Sensor Systems*: 126–29.

Marco Dorigo, Mauro Birattari, Thomas Stützle. 2006. "Ant Colony Optimization . A Computational Intelligence Technique." *IEEE Computational Intelligence Magazine* 1(4): 28–39.

Masoum, Alireza, Nirvana Meratnia, and Paul J.M. Havinga. 2018. "Coalition Formation Based Compressive Sensing in Wireless Sensor Networks." *Sensors* 18(7): 2331.

Meadows, R. G. 1976. Electrical Communications. Palgrave, London *Noise*.

Universiti Malaya

Mutthigarahalli Shankarappa, Pramod, and Shiva Shankar. 2017. "Performance Analysis of EEDR Routing Protocol for WSNs." *IET Wireless Sensor Systems* 7(1): 21–26. http://digital-library.theiet.org/content/journals/10.1049/iet-wss.2016.0029.

Ni, Qingjian et al. 2015. "An Improved Dynamic Deployment Method for Wireless Sensor Network Based on Multi-Swarm Particle Swarm Optimization." *Natural Computing*.

OOSTEROM, P VAN. 1999. "Spatial Access Methods." In *Geographical Information Systems*, , 385–400.

Park, Yonghun, Ling Liu, and Jaesoo Yoo. 2013. "A Fast and Compact Indexing Technique for Moving Objects." *Proceedings of the 2013 IEEE 14th International Conference on Information Reuse and Integration, IEEE IRI 2013*: 562–69.

Patrou, Maria et al. 2018. "DISTIL: A Distributed in-Memory Data Processing System for Location-Based Services." In *Proceedings of the 26th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems*, , 496–99.

Psannis, K.E., S. Xinogalos, and A. Sifaleras. 2014. "Convergence of Internet of Things and Mobile Cloud Computing." *Systems Science & Control Engineering* 2(1): 476–83. http://www.tandfonline.com/doi/abs/10.1080/21642583.2014.913213.

Puri, Satish, and Sushil K. Prasad. 2015. "A Parallel Algorithm for Clipping Polygons with Improved Bounds and a Distributed Overlay Processing System Using MPI." *Proceedings - 2015 IEEE/ACM 15th International Symposium on Cluster, Cloud, and Grid Computing, CCGrid 2015*: 576–85.

Qiu, Tie, Weifeng Sun, Yuanchao Bai, and Yu Zhou. 2013. "An Efficient Multi-Path

Universiti Malaya

Self-Organizing Strategy in Internet of Things." *Wireless Personal Communications* 73(4): 1613–29.

Rao, P. C Srinivasa, Prasanta K. Jana, and Haider Banka. 2016. "A Particle Swarm Optimization Based Energy Efficient Cluster Head Selection Algorithm for Wireless Sensor Networks." *Wireless Networks*: 1–16.

Richter, Stefan, Jorge Arnulfo Quiané-Ruiz, Stefan Schuh, and Jens Dittrich. 2014. "Towards Zero-Overhead Static and Adaptive Indexing in Hadoop." *VLDB Journal* 23(3): 469–94.

Saleem, Muhammad, Israr Ullah, and Muddassar Farooq. 2012. "BeeSensor: An Energy-Efficient and Scalable Routing Protocol for Wireless Sensor Networks." *Information Sciences* 200: 38–56. http://dx.doi.org/10.1016/j.ins.2012.02.024.

Sandu Popa, Iulian et al. 2011. "Indexing In-Network Trajectory Flows." *VLDB Journal* 20(5): 643–69.

Schuh, Stefan, and Jens Dittrich. 2015. "AIR: Adaptive Index Replacement in Hadoop." *Proceedings - International Conference on Data Engineering* 2015-June: 22–29.

Shah, R. C., S. Roy, S. Jain, and W. Brunette. 2003. "Data MULEs: Modeling a Three-Tier Architecture for Sparse Sensor Networks." In *Proceedings of the 1st IEEE International Workshop on Sensor Network Protocols and Applications, SNPA*, , 30–41.

Siddiqa, Aisha, Ahmad Karim, Tanzila Saba, and Victor Chang. 2017. "On the Analysis of Big Data Indexing Execution Strategies." *Journal of Intelligent and Fuzzy Systems* 32(5): 3259–71.

Universiti Malaya

Sinha, Rishi Rakesh, and Marianne Winslett. 2007. "Multi-Resolution Bitmap Indexes for Scientific Data." *ACM Transactions on Database Systems* 32(3): 16-es.

Somasundara, Arun A., Aditya Ramamoorthy, and Mani B. Srivastava. 2004. "Mobile Element Scheduling for Efficient Data Collection in Wireless Sensor Networks with Dynamic Deadlines." In *25th IEEE International Real-Time Systems Symposium*, , 296–305.

Tang, Jine, Bo Zhang, You Zhou, and Liangmin Wang. 2017. "An Energy-Aware Spatial Index Tree for Multi-Region Attribute Query Aggregation Processing in Wireless Sensor Networks." *IEEE Access* 5: 2080–95.

Tang, Jine, Zhangbing Zhou, Jianwei Niu, and Qun Wang. 2014. "An Energy Efficient Hierarchical Clustering Index Tree for Facilitating Time-Correlated Region Queries in the Internet of Things." *Journal of Network and Computer Applications* 40(1): 1–11. http://dx.doi.org/10.1016/j.jnca.2013.07.009.

Tao, Yufei et al. 2005. "Indexing Multi-Dimensional Uncertain Data with Arbitrary Probability Density Functions." In *Vldb*, , 922–33.

Tao, Yufei, and Dimitris Papadias. 2002. "Time-Parameterized Queries in Spatio-Temporal Databases." In *Proceedings of the 2002 ACM SIGMOD International Conference on Management of Data - SIGMOD '02*, , 334. http://portal.acm.org/citation.cfm?doid=564691.564730.

Teixeira, Victor, De Almeida Ralf, and D- Hagen. 2005. "Indexing the Trajectories of Moving Objects in Network." In *Proceedings Frontiers in Education 35th Annual Conference*, , 67–174. http://ieeexplore.ieee.org/document/1611877/.

Wang, Wei, Suparna De, Gilbert Cassar, and Klaus Moessner. 2015. "An Experimental

Universiti Malaya

Study on Geospatial Indexing for Sensor Service Discovery.” *Expert Systems with Applications* 42(7): 3528–38. http://dx.doi.org/10.1016/j.eswa.2014.11.058.

Wang, Wei, Vikram Srinivasan, and Kee-Chaing Chua. 2007. “Trade-Offs between Mobility and Density for Coverage in Wireless Sensor Networks.” In *MobiCom ’07*, , 39–50.

Wang, You-Chiun, Fang-Jing Wu, and Yu-Chee Tseng. 2012. “Mobility Management Algorithms and Applications for Mobile Sensor Networks.” *Commun. Mob. Comput* 12(4): 7–21. http://doi.wiley.com/10.1002/wcm.886.

Wei, Tzu Hsuan, Soumya Dutta, and Han Wei Shen. 2018. “Information Guided Data Sampling and Recovery Using Bitmap Indexing.” In *IEEE Pacific Visualization Symposium*, IEEE, 56–65.

Wooldridge, Michael. 2011. *Computational Aspects of Cooperative Game Theory*.

Wu, Kesheng, Arie Shoshani, and Kurt Stockinger. 2010. “Analyses of Multi-Level and Multi-Component Compressed Bitmap Indexes.” *ACM Transactions on Database Systems* 35(1): 1–52.

Xiong, Xiaopeng, and Walid G. Aref. 2006. “R-Trees with Update Memos.” In *22nd International Conference on Data Engineering (ICDE’06)*, , 22.

Xiong, Xiaopeng, Mohamed F. Mokbel, and Walid G. Aref. 2006. “LUGrid: Update-Tolerant Grid-Based Indexing for Moving Objects.” In *Proceedings - IEEE International Conference on Mobile Data Management*,.

Ye, Dayong, Minjie Zhang, and Danny Sutanto. 2013. “Self-Adaptation-Based Dynamic Coalition Formation in a Distributed Agent Network: A Mechanism and a Brief

Universiti Malaya

Survey." *IEEE Transactions on Parallel and Distributed Systems* 24(5): 1042–51.

Yianilos, PN. 1999. "Excluded Middle Vantage Point Forests for Nearest Neighbor Search." In *In DIMACS Implementation Challenge, ALENEX'99*, http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.31.2373.

Ying, Fang, Cao Jiaheng, Peng Yuwei, and Wang Liwei. 2008. "Indexing the Past, Present and Future Positions of Moving Objects on Fixed Networks." In *Proceedings - International Conference on Computer Science and Software Engineering, CSSE 2008*, , 524–27.

Zhang, Puning et al. 2016. "Low-Overhead and High-Precision Prediction Model for Content-Based Sensor Search in the Internet of Things." *IEEE Communications Letters* 20(4): 720–23.

Zhang, Puning, Yuan An Liu, Fan Wu, and Bihua Tang. 2015. "Matching State Estimation Scheme for Content-Based Sensor Search in the Web of Things." *International Journal of Distributed Sensor Networks* 2015.

Zhao, Y., C. Zhang, and Y.-D. Shen. 2004. "Clustering High-Dimensional Data with Low-Order Neighbors." In *IEEE/WIC/ACM International Conference on Web Intelligence (WI'04)*,.

Zhou, A. et al. 2000. "Approaches for Scaling DBSCAN Algorithm to Large Spatial Databases." *Journal of Computer Science and Technology* 15(6): 509–26.

Zhou, Zhangbing et al. 2014. "EGF-Tree: An Energy-Efficient Index Tree for Facilitating Multi-Region Query Aggregation in the Internet of Things." *Personal and Ubiquitous Computing* 18(4): 951–66.

Universiti Malaya

Zhou, Zhuo et al. 2013. “A Data Processing Framework for IoT Based Online Monitoring System.” In *Proceedings of the 2013 IEEE 17th International Conference on Computer Supported Cooperative Work in Design*, , 686–91.

Zhu, Xiaofeng et al. 2013. “Sparse Hashing for Fast Multimedia Search.” *ACM Transactions on Information Systems* 31(2): 1–24. http://dl.acm.org/citation.cfm?doid=2457465.2457469.

Zhu, Zhanyu, Qun Yang, and Dechang Pi. 2012. “Past, Current and Future Positions Index of Moving Objects in Networks.” In *Lecture Notes in Electrical Engineering*, , 335–42.

Universiti Malaya