Data Replication and Collection Framework for Enhanced Data Availability in IoT-based Sensor Systems

by

Waleed Bin Qaim

A Dissertation Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in

Computer Science and Engineering

KOÇ
UNIVERSITY

September 10, 2018

Data Replication and Collection Framework for Enhanced Data Availability in IoT-based Sensor Systems

Koç University
Graduate School of Sciences and Engineering
This is to certify that I have examined this copy of a master's thesis by

Waleed Bin Qaim

and have found that it is complete and satisfactory in all respects,
and that any and all revisions required by the final
examining committee have been made.

Committee Members:

Assoc. Prof. Öznur Özkasap (Advisor - Koç University)

Asst. Prof. Didem Unat (Koç University)

Assoc. Prof. Özlem Durmaz İncel (Galatasaray University)

Date:

To my loving wife Rabia

ABSTRACT

Internet of Things (IoT) is an emerging technology aimed at bridging the gap between physical and digital worlds. IoT envisions to provide a seamless and efficient way of interacting with real-world physical objects through the Internet which highly impacts modern day life. Wireless Sensor Networks (WSNs) are primarily responsible for embedding intelligence and turning physical objects into IoT-enabled communicating entities that can be remotely accessed and controlled, thus, providing the basis for smart solutions to all the human needs. To realize the concept of IoT, efficient data generation, collection, and presentation through WSNs are crucial issues. However, WSNs face numerous challenges mainly because of the resource-constrained and unreliable nature of nodes which may result in loss of valuable sensed data. Data replication is a promising technique to facilitate efficient data management in IoT-based sensor systems. This thesis initially proposes a taxonomy of state-of-the-art data replication protocols for IoT-based sensor systems highlighting their pros and cons by classifying them into sub-categories of data retrieval, query balancing, system robustness, and data availability. Then, we propose a data replication and collection (DRACO) framework, composed of a fully distributed hop-by-hop data replication technique for IoT-based sensor systems to avoid data loss due to node failures and enhance data availability in the network. Additionally, the proposed data replication scheme in DRACO is coupled with an efficient data collection mechanism where a mobile sink node visits a relatively smaller percentage of network nodes to collect most of the network data. Thus, no routing structure is needed to enable communication among network nodes which completely eliminates the routing overhead which is a key feature of the proposed approach. Extensive simulation experiments using network simulator (NS-3) show that DRACO ensures high data availability in the presence of node failures as well as provides maximum data collection efficiency. Comparative simulation results reveal that in comparison to two other state-of-the-art approaches, namely, greedy and random replication techniques, DRACO improves data availability with maximum gains of about 15% and 34%, respectively. Similarly, it improves average replicas created in the network with maximum gains of about 18% and 40%, respectively. Furthermore, DRACO ensures a better replica spread which determines the quality of data dissemination in the network as well as facilitates efficient data collection.

ÖZETÇE

Nesnelerin İnterneti (IoT), fiziksel ve dijital dünya arasındaki bağı güçlendirmeyi amaçlayan ve gelişmekte olan bir teknolojidir. IoT, modern dünyanın önemli bir parçası olan İnternet üzerinden gerçek dünyadaki fiziksel nesnelerle etkileşimin sorunsuz ve verimli bir şekilde sağlanmasını öngörmektedir. Kablosuz Algılayıcı Ağları (WSN) tüm insan gereksinimlerine akıllı çözümler sağlamak amacıyla fiziksel nesneleri akıllı, uzaktan erişilebilen, denetlenebilen ve IoT özellikleriyle iletişim kurabilen nesnelere dönüştürmekten sorumludur. IoT kavramını gerçekleştirmek için, etkili veri üretmek, veri toplamak ve kablosuz algılayıcı ağlar aracılığıyla toplanan verileri sunmak önemli konulardır. Bununla birlikte, WSN, düğümlerin kaynak kısıtlı ve güvenilmez yapısı nedeniyle değerli veri kaybına yol açabilecek birçok zorlukla karşılaşmaktadır. Veri replikasyonu, IoT tabanlı sensör sistemlerinde verimli veri yönetimini kolaylaştırmak için ümit vaat eden bir tekniktir. Bu tezde, öncelikle IoT tabanlı sensör sistemlerinin avantaj ve dezavantajlarını vurgulayarak son teknoloji veri replikasyon protokolleri için veri alma, sorgu dengeleme, sistem sağlamlığı, veri kullanılabilirliği gruplarından oluşan bir sınıflandırma önerilmektedir. Ardından, düğüm hataları nedeniyle oluşan veri kayıplarını önlemek ve ağdaki veri erişilebilirliğini artırmak amacıyla IoT tabanlı sensör sistemlerine özel, tamamen dağıtık ve atlamalı veri replikasyon tekniklerinden oluşan bir veri replikasyon ve toplama (DRACO) sistemi önermekteyiz. DRACO sisteminde önerilen veri replikasyon algoritması, bir mobil evre düğümünün, ağ verilerinin çoğunu toplamak için ağ düğümlerinin az bir kısmını ziyaret etmesini gerektiren verimli bir veri toplama mekanizması ile birleştirilmiştir. DRACO sisteminin en önemli özelliği, yönlendirme yükünü tamamen ortadan kaldırarak ağ düğümleri arasında yönlendirme iletişimine ihtiyaç duymamasıdır. Ağ simülatörü (NS-3) kullanılarak gerçekleştirilen kapsamlı simülasyon deneyleri, DRACO'nun, düğüm

hatalarının varlığında yüksek veri erişilebilirliği ve veri toplama verimliliği sağladığını göstermektedir. Karşılaştırmalı simülasyon sonuçları, literatürdeki öncü veri replikasyon tekniklerine kıyasla, DRACO'nun veri erişilebilirliğini yaklaşık %15 ve %34'lük kazanımlarla artırdığını ortaya koymaktadır. Benzer şekilde, ağda oluşturulan veri kopyaları da yaklaşık %18 ve %40 oranında azaltılarak sistem verimliliği geliştirilmiştir. Ayrıca, DRACO, ağdaki veri dağıtım kalitesini belirleyen ve verimli veri toplanma işlemini kolaylaştıran etkin replika yayılımı sağlamaktadır.

ACKNOWLEDGMENTS

First and foremost I thank Allah, the Almighty. After that, I would like to express my sincere gratitude to my advisor Prof. Öznur Özkasap for guiding, supporting, trusting, and encouraging me throughout my graduate studies at Koç University. All the useful discussions, advices, and suggestions by her contributed a lot towards completion of this research work. Thank you Prof. for everything.

Next, I would like to thank the committee members Prof. Özlem Durmaz İncel and Prof. Didem Unat for their help, support, and participation.

A very special thanks to my dearest friend Seyhan Uçar for his invaluable support and help throughout. He was always there as my best friend and brother to answer all my questions, encourage, motivate, and appreciate me. I owe you big time.

I want to thank all the past and present members of the Distributed Systems and Reliable Networks (DISNET) research lab for their friendly and helpful attitude. Thank you all Seyhan, Umur, Karim, Ahsan, Berkin, Dzenaida, Yahya, Shahid, Ipek, Beakal, and Sanaz for your support. Your company has been a source of great relief and entertainment in this challenging journey. I wish you all the very best in life.

I am thankful to the whole Pakistani community here at Koç University for being so helpful and supportive to all the fellow Pakistanis. I am also thankful to Koç and Higher Education Commission (HEC) Pakistan for supporting me throughout in an uninterrupted manner.

I remain highly indebted to my family who always spared me from my responsibilities and encouraged me to focus on my studies. Thank you Abu, Ami, Mona, Ramsha, and Osama for being a source of continuous support and encouragement. I feel blessed to have you all in my life.

Finally, loving thanks to my wife, Rabia, my biggest source of motivation who was always there to listen to me, believe in me, encourage me in tough days, and cheer for me in my achievements. Nobody has been more important to me in the pursuit of this degree than her. Words can not do justice for her unconditional and continuous support that always kept me going. I was always amazed by her willingness to discuss my research work and proofread countless pages of my writings. It would never have been possible without you. My love and gratitude for always being by my side.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

INTRODUCTION

Internet of Things (IoT) is an emerging technology aimed at bridging the gap between physical and digital worlds. IoT envisions to automate and improve urban life by bringing physical objects and devices closer to the consumers to provide better access and control [1]. The increasing reliance of users on technology and the Internet, demands smart solutions for all the human needs. Thus, IoT-based systems are gaining increasing popularity to satisfy such requirements which greatly reduce human effort and time. Any device that can be connected to and controlled through the internet can be termed as an IoT-enabled device. Examples include computers, cell phones, tablets, and IP-enabled cameras [2]. However, not every device and object around us can be connected to the internet. Such as our home appliances, vehicles, buildings and so on. Therefore, the goal of the IoT technology to connect everything to the internet can be realized by turning physical objects and devices around us into IoT-enabled devices. Smartness in physical objects and non-communicating entities is induced through embedding tiny electronic equipment into them, such as sensors and actuators. Sensors are small devices that are capable to sense, record, and communicate the current state of any object they are attached to. Whereas, actuators make it possible to control the objects of interest by making them operate in the desired manner through issuing instructions [3]. These devices may range from personal gadgets to every day home and office appliances, vehicles or simply anything of interest.

With the aim of turning objects into IoT-enabled devices and connecting them to the Internet, IoT technology has a very wide scope spanning several wired and wireless networks involving fixed and ad hoc infrastructures [4]. These may include wired networks, cellular networks, mobile ad hoc networks (MANETs), vehicular ad hoc networks (VANETs), the recently proposed flying ad hoc networks (FANETs), and the backbone internet infrastructure [5, 6]. However, Wireless Sensor Networks (WSNs) remain at the core of IoT to provide various IoT-based services as depicted in figure 3.1 [7, 8].

A WSN is formed of small sensor nodes wirelessly connected to each other and to a relatively more powerful sensor node called the gateway node. Sensor nodes are capable of sensing different physical phenomena in their vicinities, such as temperature, humidity, noise, and pressure [9]. The sensor nodes are also capable of storing and carrying out some basic processing on the sensed information. A sensor node can either be a source node or an intermediate router to relay information to other nodes in the network. The recorded observations are communicated to the gateway node which is responsible to collect the observations from all the sensor nodes in the network. Finally, an end user may connect to and query the gateway node to retrieve the information from the network [10]. In an IoT-based WSN, the gateway node can be connected to the internet, thus enabling users to access and control the devices from anywhere around the world through the internet [11, 12]. In typical WSN applications, sensor nodes are assumed to be static, whereas the sink node can be either static or mobile depending upon the application. Sensor nodes are resource constrained devices and can be homogeneous in characteristics such as processing power, available memory, energy and communication capabilities. However, a WSN can also consist of heterogeneous nodes with some powerful nodes with higher capabilities as compared to the rest of the nodes. These powerful nodes are usually devoted to performing some heavy duty tasks in the network, such as scheduling, clustering, and data relaying [13]. Each sensor node is assumed to periodically generate data upon its sensing interval. The generated data by sensor nodes is collected by the sink node to be further processed and relayed to the end user. Data relaying towards the gateway can be done either proactively at regular intervals or it can be reactive where sensor nodes relay the sensed data as a reply to a query initiated by the sink node or some other node in the network. The gateway node can be either a static and always present node in the network or it can be a mobile node that occasionally visits the sensor nodes to collect the sensed information. In the case of a fixed gateway node, all the sensed information needs to be routed towards the gateway which involves the routing overhead as well as may cause the energy hole problem in nodes close to the sink. Whereas, in the case of a mobile sink node, the information flow towards a fixed node is not required and the mobile gateway can visit the desired nodes in the network and connect to each device to retrieve information. The use of a mobile gateway alleviates the routing overhead which allows conserving power of sensor nodes which are battery-powered devices with limited power resources [14]. Thus, enabling the network to operate for a longer period of time [15]. The number of gateway nodes also referred to as sink nodes, may also vary depending upon the application and the monitored area. Applications that require a real-time response, such as battlefield monitoring and surveillance, need to deploy static sink nodes [16]. Whereas, delay tolerant applications, such as environmental and habitat monitoring may implement mobile sink nodes to collect data occasionally [17].

Embedding smartness into objects and devices around us to turn them into communicating entities, opens the horizon to many possible IoT applications such as intelligent transportation system, smart home, smart city, eHealth, surveillance, environmental, and habitat monitoring, to name a few [17–25]. With such a wide range of applications, comes the need for efficient data management [26]. In the IoT context, sensor nodes generate an enormous amount of valuable sensed observations that need to be efficiently retrieved, processed, and stored [27, 28]. However, WSNs suffer from some intrinsic limitations, such as limited storage capacity and unreliable nature of the devices involved, which may result in loss or mismanagement of the sensor data [29].

Several data management techniques have been proposed for WSNs over the years [30–32]. One such technique is data replication, that allows sensor nodes to utilize the available memory space of peers in the network to save redundant copies of sensed data [33, 34]. It allows preserving sensed data at different locations in the network which benefits the overall network performance in a number of ways. The benefits of data replication include fault tolerance through redundancy, query balancing, enhanced data availability, efficient memory utilization, and data gathering [35].

1.1 Contributions

In this thesis, we address the benefits of data replication in a homogeneous WSN for an IoT-based sensor system. We propose DRACO, a data replication and collection framework for enhanced data availability and efficient data collection in the network. We then evaluate the performance of the proposed framework based on the metrics of interest. The main contributions of this thesis are as follows:

- We present a chronological and consolidated review of the state-of-the-art data replication protocols proposed for WSNs in the IoT context.
- We propose a taxonomy of data replication solutions in IoT-based WSNs and classify them into subcategories of data retrieval, query balancing, system robustness, and data availability based on the purpose served.
- We present a detailed discussion on key characteristics of state-of-the-art solutions while comparing them in terms of their pros and cons.
- We propose a fully distributed hop-by-hop data replication technique to avoid data loss due to node failures.
- We analyze the effect of data replication on data availability in the presence of node failures to preserve data as well as the effect of node density on the average number of replicas created in the network.

- We propose an intelligent data collection mechanism to benefit from data replication where a mobile sink node visits a relatively smaller percentage of network nodes to collect most of the network data.

- We analyze the effects of data replication, node density, and node failures on the proposed data collection mechanism.

- We perform extensive simulations in Network Simulator-3 (NS-3) to validate and compare the performance of the proposed framework to other state-of-the-art solutions.

- Comparative simulation results show that, in comparison to two other state-of-the-art approaches, namely greedy and random replication techniques, the proposed framework improves data availability with maximum gains of about 15% and 34%, respectively. Similarly, it improves average replicas created in the network with maximum gains of about 18% and 40%, respectively.

- Moreover, our proposed framework also improves the quality of data dissemination by achieving a better replica spread in the network in terms of distance traveled by successive replicas from the source node. Hence, with the proposed intelligent data collection mechanism, we are able to achieve higher data collection efficiency in comparison to two most commonly used and state-of-the-art data collection approaches with uncontrolled sink mobility namely, random walk and self-avoiding random walk.

1.2 Organization

The rest of the thesis is organized as follows. Chapter 2 presents a detailed survey of the related works. In chapter 3, we present the system overview. Chapter 4 provides algorithmic details of the proposed data replication technique along with its performance analysis. Chapter 5 details the data collection algorithm and the experiments performed to validate and compare its performance. Finally, chapter 6 concludes our major findings.

Chapter 2

RELATED WORK

In this chapter, we provide the literature review on data replication and collection techniques for IoT-based sensor systems. We present our proposed classification scheme for state-of-the-art techniques and based on the purpose data replication serves, the existing solutions are classified into (i) data retrieval, (ii) query balancing, (iii) system robustness, and (iv) data availability solutions, as depicted in figure 2.1 [36]. The strategies listed in each category are discussed in more detail under corresponding sections.

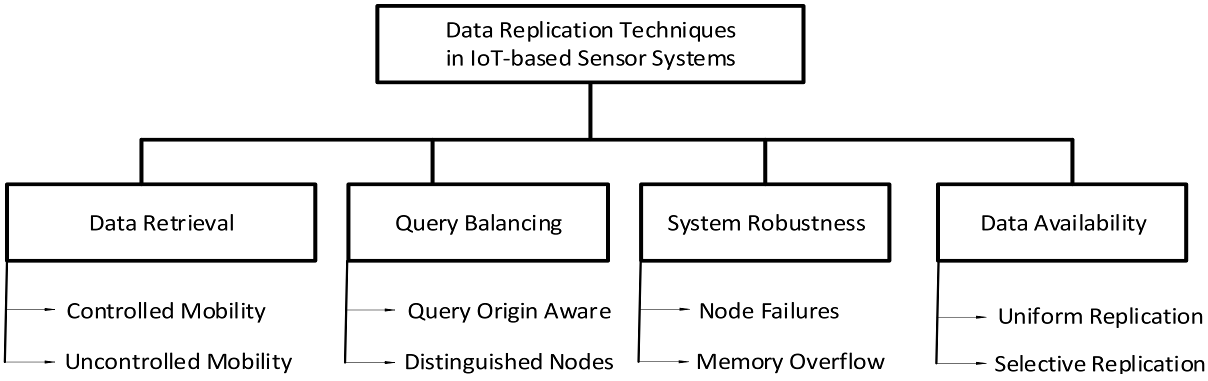


Figure 2.1: Taxonomy of Data Replication Techniques in IoT-based Sensor Systems

2.1 Data Retrieval

One of the benefits of data replication is that it facilitates the use of mobile sink for efficient data collection from the network. In case of no data replication or aggregation, a mobile sink visiting each node for data collection in a large network is considered to be inefficient [37]. Thus, effectively distributing network data among the nodes allows efficient data collection. Through data replication, each node in the network holds data from multiple other nodes, besides its own data items. Thus, it allows a mobile sink to visit a small percentage of nodes to collect a reasonable amount of network data. There are two main approaches regarding the mobility of the sink node. First, the sink node may have a *controlled mobility*, in which case the sink node always follows a predetermined path in the network for data collection. However, this approach has a path control overhead as well as it requires network data to be located mostly on or around that path [38, 39]. Second, the mobile sink node can have *uncontrolled mobility*. In this case, there is no path restriction and thus no path control overhead. However, it requires the network data to be distributed in such a manner that it allows maximum data collection by visiting only a small fraction of randomly chosen nodes. Hence, all the existing solutions that leverage data replication for efficient data collection as their main objective are combined under this category.

2.1.1 Methods

A proactive data dissemination and storage scheme for WSNs called DEEP has been given in [39] that distributes and replicates data equally in the network nodes. A mobile sink then collects network data by visiting a certain number of nodes at a reduced communication overhead. DEEP incorporates uncontrolled sink mobility where the sink node may choose to randomly visit a small percentage of nodes as compared to the total number of nodes to gather almost 90% of data items generated by the network. DEEP provides a probabilistic forwarding of data items in the network coupled with probabilistic storage for replication purposes. Performance of DEEP has been measured both analytically and through simulations for several metrics including data gathering efficiency, message overhead and quality of data distribution. Comparative results reveal that DEEP is a viable solution as it offers a reasonable data gathering efficiency at a cost of reduced communication overhead as compared to other similar schemes.

Another flexible probabilistic data dissemination approach for WSNs called Supple is given in [40]. Supple is composed of three phases, namely: (i) tree construction, (ii) weight distribution, and (iii) data replication. In the tree construction phase, a central node in the network initiates a tree construction mechanism where every node in the network knows its path to the sink node and its distance from the central node in terms of the number of hops. In the weight distribution phase, nodes are assigned weights to represent their probability to store data based on their distance from the sink. Finally, in the data replication phase, all the nodes send their data to the central node. The central node eventually decides the number and position of replicas for each data item it receives from the network; based on distances and weights of nodes. However, the positioning of the central node in the network and the problem of traffic overload on nodes close to the central node is not addressed.

A proactive data replication scheme to decouple data dissemination phase from data retrieval in WSNs with mobile sinks is given in [41]. The authors propose a controlled Random Walk (RW) based approach to uniformly distribute sensor data in the network. The data generating node initiates a RW by randomly selecting the next neighbor and this process continues until the RW reaches its destination. Nodes on the path replicate data locally as the RW continues, thus each node has a set of storage nodes in the network. Consequently, a mobile sink node may visit any set of m nodes to collect a significant amount of data in a network composed of n nodes where $m << n$. Performance results show that if every node replicates data from $\sqrt{n}$ nodes in the network, a mobile sink can retrieve 90% network data by visiting any $2.3\sqrt{n}$ nodes in the network.

2.2 Query Balancing

Balancing query overload in WSNs is another purpose served by data replication. In query-based systems, nodes respond to queries by communicating the requested data items. If a specific node receives a large number of queries, it might exhaust its power resources quickly by constantly engaging in data transmissions, as well as the responses might get delayed. Therefore, data replication can enable sharing the burden of queries among nodes, thus resulting in better network performance. Such solutions mainly follow two approaches. The first approach referred to as *query origin aware*, involves placing replicas of highly requested data items close to querying nodes such that the queries are shared as well as the query response time is reduced. Such methods generally require some sort of query localization. The second approach is to replicate data items at some *distinguished nodes* in several neighborhoods of the entire network, e.g., in cluster heads. Thus, requests arising from each cluster get a response through the respective cluster head. Solutions using data replication as a means to balance query overload in the network are presented in this section.

2.2.1 Methods

An adjustable grid-based data replication scheme for wireless sensor networks called ADR has been given in [42]. ADR makes use of a virtual grid to control traffic overload for some specific nodes. ADR proposes that in response to a large number of queries, nodes spend more energy in communication as compared to energy spent in holding data items. Such a situation is called query hotspot problem. Thus, when a node receives a large number of queries, it marks itself as a query hotspot. Consequently, a replica node is created in its vicinity to share the burden of responding to the queries. When the queries reduce below a certain threshold, that replica node is removed. Comparative simulation results for ADR show that the proposed scheme outperforms similar approaches in terms of the number of replicas, total packets, energy consumption and query latency.

A service-oriented approach (SOA) has been given in [43] for wireless sensor networks. The authors argue that the unreliable nature of a WSN may cause a service failure in which case the collected or computed data for the desired service may get lost. Therefore, a redundancy infrastructure is suggested for such networks which is composed of two main parts namely data replication and service recovery. SOA proposes to replicate service data among suitable sensor nodes selected through a scoring function based on node properties such as connectivity, available storage, and remaining energy. The node with the highest score is selected as a replica node. The second phase attempts to resume a failed service by collecting the service data from replica nodes and resuming computation for service delivery. Performance of SOA was compared with other similar approaches namely bounded flooding and randomized replica distribution. Comparative simulation results reveal that SOA provides better results as compared to other techniques in terms of reliability, cost of service recovery, storage efficiency, and energy consumption.

A replica arrangement scheme called Dynamic Replica Arrangement on Concentric Circular Arcs (DRACA) for WSNs is given in [44]. DRACA proposes to dynamically create replicas geographically close to the area generating frequent queries along virtual concentric circles. Nodes on the same circular arc are made to point the position of a replica node. In such a way, a query may get completed by a replica node on its way to the target node which reduces the overall communication cost for queries in the network. DRACA also provides a replica update mechanism, such that if a data item is updated at any point in time, all the replica nodes are also updated to ensure consistency. Replicas are dynamically created and removed based on the query rate, i.e., if queries fall below a certain threshold for a particular data item, its replicas are removed.

A geographical hash table (GHT) system for data-centric storage in sensor networks is proposed in [45] to geographically distribute sensed data in the network and ensure data persistence in the presence of node failures. GHT uses a geographic routing algorithm, called GPSR [46], as the underlying routing structure to maintain persistence of data if network nodes move or fail. GHT proposes to hash keys into geographical coordinates and replicate data on a set of perimeter nodes geographically located close to the hashed coordinates in a key-value form. GHT ensures to always select the same node with high consistency for a given key called the home node. Thus, queries for required data can be routed to a specific location in the network instead of flooding the queries in the network. Moreover, GHT distributes storage and communication overhead evenly among multiple keys. If query hotspots are created as a result of hashing multiple keys to the same home node, GHT handles it through structured replication where events hashing to a common home node are distributed among several mirror nodes.

An extension of geographical hash tables (GHTs), called Double rulings, has been proposed in [47]. In double rulings, replicas of data items generated by sensor nodes are placed on circular geographical trajectory, called the replication curve; instead of creating replicas on separate isolated sensor nodes as in the case of GHTs. Data consumers reach the closest replica of the desired data item by initiating their queries along another trajectory, called the retrieval curve. It is guaranteed that the retrieval curve always intersects the replication curve to reach the nearest replica of the desired data item to ensure successful retrieval. In such a way, it reduces the consumer cost to reach the desired data item as well as it prevents query overload on some specific nodes in the network. However, the limitation is that it requires structured networks following a uniform deployment scheme such as grids.

A data replication technique for top-k query processing in mobile wireless sensor networks (MWSN) is given in [48]. The authors argue that in a query-driven MWSN, it is not efficient to flood the entire network with a query where all the sensor nodes respond to that query, resulting in unnecessary network traffic. Rather, retrieving selective and necessary data can significantly reduce network traffic as well as conserve energy. Top-k query processing is a promising solution which allows a node to retrieve only k data items with the best score (based on some scoring function). Furthermore, in a mobile environment, network partitioning is very common due to which some query issuing node might not be able to retrieve data from nodes in a different partition. Thus, data replication with top-k query processing (with and without data update) is proposed to keep high accuracy for the top-k query results in the presence of network partitioning. In the proposed schemes, data items with high scores have a higher probability of being replicated as well as nodes exchange their list of data items to reduce replica duplication. In case of data update, higher priority is given to data items with longer update time so that data remains valid for a longer period.

The simulation results show higher accuracy in query results.

2.3 System Robustness

System robustness is another benefit achieved by data replication. Node failures and resource overflow can never be overruled in any WSN application. These problems may become severe in unattended WSNs deployed in harsh environments resulting in high rate of node failures causing high data loss. Thus, in order to combat data loss, data replication is a promising solution that increases system resilience by creating redundant copies of the sensed data items at various nodes in the network. There are two main sources of data loss in WSNs namely, *node failures* and *memory overflow*. Hence, all the data replication techniques that incorporate data replication as a means to preserve data are combined under this category.

2.3.1 Methods

A distributed data storage scheme through in-network data replication for large-scale IoT-based surveillance systems has been proposed in [49]. The study argues that applications that do not require real-time response, WSNs observe environmental data through sensing and relaying this data to a mobile sink node that wanders across the monitored area occasionally without any defined period. In such a case, if data retrieval becomes rare, valuable sensed data might be lost as a result of node failures or memory overflow. A low complexity data replication mechanism has been proposed that selects replicas based on the available memory space of neighboring nodes. The performance of the proposed scheme has been tested both experimentally as well as through simulations. The performance metrics include system storage capacity, the time required to reach this limit and robustness. Performance results show that the proposed scheme guarantees a reasonable spread of replicas and ensures robustness of the scheme against node failures.

A distributed data storage protocol called ProFlex for heterogeneous WSNs with mobile sinks has been proposed in [37]. The assumed WSN consists of two types of nodes namely H(High) sensors and L(Low) sensors (according to their capabilities). ProFlex suggests incorporating distributed data replication on carefully selected replica nodes such that a sink can get a representative sample of the entire network by visiting only a few nodes. ProFlex is composed of three phases including tree construction, importance factor distribution, and data distribution. Experiments are performed to find an optimal number of H sensor nodes and L sensor nodes in the network as well as their communication radius for improved data gathering efficiency and message overhead. Performance of ProFlex was compared with a number of similar approaches available in the literature and was found presenting minimum overhead in terms of total messages exchanged with comparable data gathering efficiency for varying node failure and message loss rates. Furthermore, as an improvement to ProFlex, it was suggested to leverage the data correlation and redundancy feature that is inherent to WSNs which resulted in a considerable decrease in overall replication overhead.

2.4 Data Availability

Data replication can also be used to enhance data availability in the network. WSNs deploy various schemes to conserve energy of the nodes. For instance, duty cycling is one approach where sensor nodes temporarily turn off their major energy consuming components (e.g., transceivers, processors, memory) and enter sleep mode to conserve energy. Deploying such techniques may adversely affect the overall data availability in the network. Therefore, through data replication, data items are redundantly maintained at various nodes in the network, ensuring that some copies remain available in the network to enhance data availability. There are usually two approaches followed under this context. First, deploying a *uniform replication* where data is distributed equally throughout the network. Second, *selective replication* can be carried out at some distinguished nodes that might have higher resources and availability.

2.4.1 Methods

tinyDSM provides a reactive replication scheme for WSNs to enhance data availability [50]. The authors argue that since sensor nodes are resource constrained devices, deploying duty cycling to conserve energy is a common practice. This makes some nodes unavailable for short periods of time. Also, sensor nodes are prone to failures. Such nodes are referred to as disappearing nodes. Disappearing behavior of the sensor nodes highly degrades overall data availability in the network. tinyDSM proposes a configurable replication approach in terms of replication area where replicas are randomly distributed based on replica number and density.

USEE provides a uniform data replication strategy in WSNs for large-scale communicating materials (e.g., concrete in smart buildings) to observe physical transformations in the material lifecycle [51]. It makes use of limited neighborhood information to replicate and save data in the network. USEE is composed of three strategies. First, it incorporates a counter-based flooding mechanism to disseminate data in the network. Second, it uses a neighbor storage information approach to control replication and avoid duplication in a single neighborhood. Third, it provides a probabilistic storage based on data importance. USEE attempts to optimize energy and memory utilization of nodes in large-scale communicating materials network.

Multiple replica allocation schemes for correlated data items in ad hoc sensor networks are given in [52] namely, C(Correlation)-SAF(Static Access Frequency), C-DAFN(Dynamic Access Frequency), and C-DCG(Dynamic Connectivity Based Grouping). The authors are of the view that generally, sensor nodes access data items with correlation among the requested data sets. If nodes are unable to access the whole data set composed of multiple correlated data items located at different nodes, it adversely affects the overall data accessibility in the network. All the proposed techniques consider different criteria for replica allocation besides, data correlation. C-SAF considers access frequency to each data item in the network, whereas C-DAFN and C-DCG consider neighborhood information among nodes and the whole network topology, respectively, along with access frequency. Comparative performance results

Category	Methods	Replication Strategy	Topology	Sink Node	Sensor Nodes	Replica node	Replica Update
Data Retrieval	DEEP [39]	Proactive	Random	Mobile	Homogeneous	Any	No
	Supple [40]	Proactive	Tree	Mobile	Homogeneous	Any	No
	Random Walks [41]	Proactive	Random	Mobile	Homogeneous	Any	No
Query Balancing	ADR [42]	Reactive	Grid	No	Homogeneous	Any	No
	SOA [43]	Proactive	Random	No	Homogeneous	Any	No
	DRACA [44]	Reactive	Grid	No	Homogeneous	Distinguished	Yes
	GHT [46]	Proactive	Grid	Static	Homogeneous	Distinguished	No
	Double Rulings [47]	Proactive	Grid	Static	Homogeneous	Distinguished	No
	Top-k Query [48]	Reactive	Random	No	Homogeneous	Any	Yes
System Robustness	Greedy [49]	Reactive	Grid	No	Homogeneous	Any	No
	ProFlex [37]	Reactive	Random	Mobile	Heterogeneous	Any	No
Data Availability	tinyDSM [50]	Reactive	Grid	No	Homogeneous	Any	No
	USEE [51]	Proactive	Random	No	Homogeneous	Any	No
	C-SAF/DAFN/DCG [52]	Reactive	Random	No	Homogeneous	Any	No

Table 2.1: Comparison of Data Replication Techniques in IoT-based Sensor Systems

show that considering data correlation improves overall network data accessibility. However, network traffic also increases under C-DAFN and C-DCG methods.

Overall, Table 2.1 presents a comparison of the existing data replication techniques for IoT-based sensor systems. The solutions are compared based on some generic system features including replication strategy, network topology, presence and type of the sink node, nature of sensor nodes, selection of replica node, and deployment of any replica update mechanism. It can be observed that the replication strategy may be proactive or reactive depending upon the application and purpose. In a proactive approach, data replication is carried out at regular intervals. Whereas, a reactive replication is triggered due to the occurrence of some event such as data generation and query overload. WSNs may be deployed with different network topologies which can be broadly classified as structured or unstructured. In structured network topology, deployment of network nodes follows some predetermined scheme such as grid or tree. Whereas, in unstructured network topologies sensor nodes are randomly deployed to get a uniform random observation of the monitored area. Sink node can be of two types, i.e, either static or mobile. Sensor nodes in the network may be homogeneous in characteristics, such as processing power, energy, memory, and communication capability or they can be heterogeneous with different capabilities. Replicas may be created at any node in the network or some distinguished nodes can be selected based on node properties or locations. Finally, some solutions might incorporate replica update mechanism where outdated replicas of data items may be replaced with updated values or removed altogether.

After the advent of IoT technology, numerous smart solutions have been developed that are transforming the modern way of life. Considering the importance of efficient data management, we provided a comprehensive survey and taxonomy of state-of-the-art data replication protocols in IoT-based sensor systems. To the best of our knowledge, our study is the first one that presents a detailed discussion on the existing solutions by classifying them into subcategories of data retrieval, query balancing, system robustness, and data availability methods. Benefits and drawbacks of each method have been highlighted under all subcategories along with a comparison of solutions based on generic system properties.

Chapter 3

SYSTEM OVERVIEW

In this chapter, we present our system preliminaries including system model, details of the node failure model, and simulation setup. We also describe the performance metrics used in this thesis to evaluate and compare the performance of the proposed DRACO framework.

3.1 System Model

The system model consists of a set of N static sensor nodes deployed in a sensing field of area A as depicted in figure 3.1. The deployment of sensor nodes follows a uniform random distribution to uniformly distribute the sensor nodes across the field. Each sensor node in the system is uniquely identified by its node ID i where $i = \{1, 2, ...N\}$. All sensor nodes in the system regularly sense their environment and generate data items with a fixed interval called the sensing interval. The sensor nodes are assumed to be homogeneous in characteristics such as processing power, memory, and communication capabilities. Each sensor node has a fixed memory space called the buffer to hold data items. Thus, a sensor node can hold data items generated by itself as well as serve as a replica node to some other nodes in the system but not for all of them due to limited memory space. All the sensor node are also capable of constantly monitoring their available resources including memory denoted as RM and number of connected neighboring nodes denoted as NoN. All the sensor nodes periodically exchange their resources' information with their neighbors including RM and NoN through broadcast messages. Each node maintains a list of its neighboring nodes along with their updated attributes called the neighbor's attribute table denoted as N_{Att}. This information is utilized in the data replication

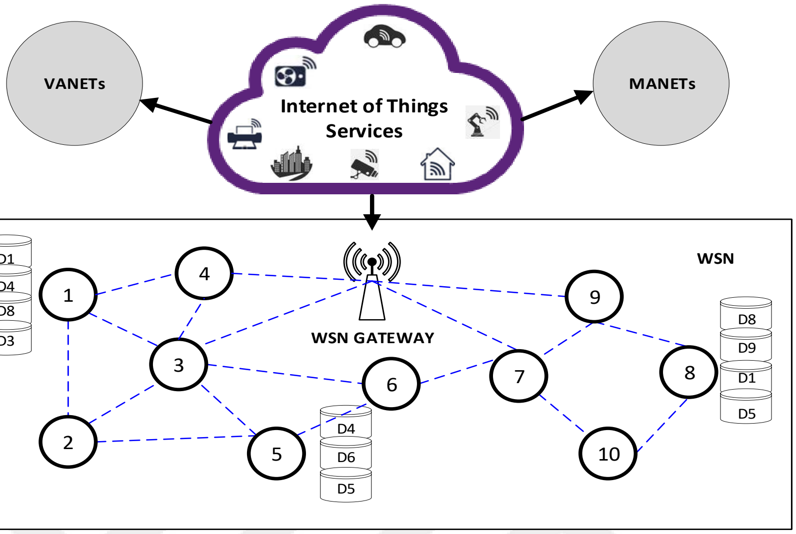


Figure 3.1: Various ad hoc networks supporting IoT services and system model for WSN.

phase where each node selects the best node among its neighbors to create a replica. Number of replicas to create per data item is controlled by the replication degree R. For instance, a replication degree of 2 means to create 2 copies of each data item generated in the network. The sensor node that generates a data item is called the *data owner* and the node that creates a copy of the data item generated by some other node in the network is called the *replica node*. To ensure maximum spread of replicas in the network, each replica node appends the list of its 1-hop neighbors to the data message, denoted as CN. The number of hops in terms of nodes, traveled by a data item is called the hop count denoted as H. Set of nodes holding replicas of a particular data item are referred to as the previous replicas denoted as PR. Set of nodes visited by a data item are called previously visited nodes, denoted as PV. Each node uses this information (R, RM, PR, PV, and CN) in order to select the best candidate among its neighbors to be a replica node for a particular data item.

The system also consists of a mobile sink node also referred to as the WSN gateway. In an IoT-based WSN, this sink node connects the WSN to rest of the world through the internet cloud as can be seen in figure 3.1. The mobile sink node visits the sensor field to collect sensed data from the sensor nodes. This mobile sink node is considered to be more powerful as compared to other ordinary sensor nodes in the system in terms of memory and processing power with no significant resource limitations. The mobile sink node is assumed to have no knowledge about the sensor nodes in the system and their locations. It utilizes the sensor field diagonal coordinates denoted as $C1, C2$ to know the boundaries of the sensor field. The mobile sink node generates random coordinates within the boundary of the sensor field to visit nodes for data collection where it discovers the sensor nodes in its communication radius denoted as CR through advertising its presence and then selects the best node to collect data items (details in the data collection section). The sink also keeps track of the previously visited sites denoted as PVS and nodes to avoid repetition. The set of collected data items by the sink node is denoted as D_c.

3.2 Node Failure Model

To analyze the performance of the proposed data replication and collection mechanisms for system robustness under node failure scenarios, we implement a random node failure model to simulate node failures. In this model, sensor nodes can randomly fail any time during network operation. A failed sensor node no longer remains part of the system. Thus, a failed node does not participate in data generation or replication. Moreover, a failed node is assumed to leave the network forever and does not return to the system, thus resulting in loss of data items it holds not replicated on any other node in the system.

3.3 Simulation Setup

The proposed algorithms have been implemented in Network Simulator-3 (NS-3) [53]. The simulation setup consists of 100 sensor nodes (unless otherwise stated) randomly deployed in a 100 m x 100 m square region. Sensor nodes are assumed to be homogeneous in characteristics and resources. Each sensor node has a fixed buffer size and periodically generates data items as per the randomly assigned sensing interval from the sensing interval window of [1-4] seconds. The transmit power of all the nodes is set to -20dBm. Initially, each node only knows about its identity and its resources. However, nodes learn about their immediate neighbors through regular broadcast messages. Each node maintains a neighbors' attribute table N_{Att} for all 1-hop neighbors in its vicinity. N_{Att} holds node id, MAC address, number of neighbors, and the remaining memory of each neighbor learned through periodic resource advertisement messages broadcasted by each node every 10s. The simulation time is set to 400s (corresponding to the highest sensing interval). Each node communicates with other nodes in the network either through a broadcast message for advertising its resources or through unicast messages using MAC addresses for data dissemination. Thus, totally eliminating the routing overhead. Each simulation comprises a data dissemination phase and a data collection phase. In the data dissemination phase, nodes periodically generate and disseminate data among their neighbors. At the end of the data dissemination phase, a mobile sink node collects data from the sensor nodes in the data collection phase. The mobile sink node is assumed to perform as many visits inside the sensor field as required to collect the required amount of network data (which can be tuned as per the requirements of the application). Table 3.1 lists our main system parameters.

Number of nodes	100
Sensing Interval	[1-4] s
Broadcast Interval	10 s
Simulation Time	400 s
Sensing Field	100 x 100 meter square
Propagation Loss Model	Log distance
Node Failure Model	Random
Network Topology	Random
MAC Protocol	802.15.4

Table 3.1: Main System Parameters

3.4 Performance Metrics

The portfolio of metrics used to analyze the comparative performance of the proposed framework includes the following.

3.4.1 Data Availability

This metric is defined as the average percentage of available data items in the system (excluding copies) in the presence of node failures. The availability of a particular data item equals 1 if, at least a single copy of that data item exists in the network. Otherwise, it is considered as 0.

3.4.2 Average Replicas Created

This parameter gives the average number of copies created for each data item in the network. Although the replication degree is statically set in the range of [2-5], however, achieving the same replication degree for each data item is not possible due to unavailability of suitable nodes to create all the copies. Thus, it is important to measure the average replicas created in the system.

3.4.3 Replica Spread

Replica spread is determined by the average Euclidean distance traveled by each replica from the data owner node. It represents how well replicas spread across the network.

3.4.4 Data Collection Efficiency

Data Collection efficiency is defined as the percentage of network data collected by the mobile sink node after visiting a given number of sensor nodes. If a mobile sink node is able to collect higher percentages of network data by visiting fewer nodes, such a data collection approach is considered to be more efficient. Efficiency in data collection is linked with both the data dissemination mechanism of the sensor nodes as well as the data collection strategy of the sink node. If each visited sensor node provides more new information that a sink has not collected previously, it will result in higher data collection efficiency.

Chapter 4

DATA REPLICATION

In this chapter, we provide details of the proposed data replication algorithm as part of DRACO framework to efficiently disseminate sensed data among peers in the network [54]. The idea is to distribute and store redundant copies of the generated data at multiple locations in the network to enhance data availability and system robustness.

4.1 Algorithm Overview

We propose a fully distributed hop-by-hop data replication algorithm to reduce data loss in IoT-based WSNs due to local memory shortage at sensor node level and to enhance data availability in the presences of node failures in the system. The unique feature of the proposed technique is that it makes use of limited neighbor's information to perform in-network data replication to preserve data.

Algorithm 1 is invoked at each sensor node whenever either it generates a data item or it receives a data item generated by some other node in the network.

4.2 Algorithm Inputs and Output

As inputs, the algorithm requires the data item D_i, replication degree R, remaining memory RM, node IDs of previous replicas PR for data item D_i, node IDs of previously visited nodes PV by data item D_i, node IDs of 1-hop neighbors of previous replicas CN, hop count H, and neighbors' attribute table N_{Att}. As output, the algorithm generates node ID of the best candidate among the neighbors of a node to create a replica of data item D_i.

Algorithm 1: Data Replication

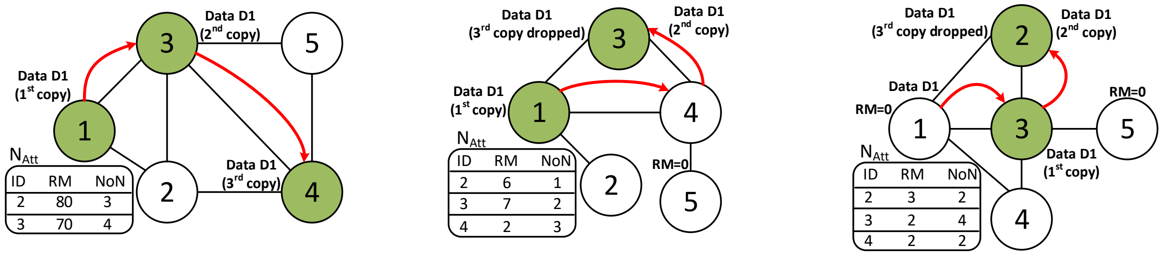
```
Input: Data item $D_i$, Replication Degree $R$, Remaining Memory $RM$,
       Previous Replicas $PR$, Previously visited nodes $PV$, Common
       Neighbors $CN$, Hop Count $H$, and Neighbors attribute table $N_{Att}$
Output: Best Replica Candidate Node $RC$

if $RM > 0$ then
    create replica;
    decrement $R$;
    append node id to $PV$ and $PR$;
    append node ids of neighbors to $CN$;
else
    append node id to $PV$;
if data owner then
    if $R > 0$ then
        search for a node with max. $NoN$ and $RM > 0$ in $N_{Att}$;
        if found then
            select as $RC$;
            increment $H$;
else
    if $R > 0$ then
        search for a node with max. $NoN$, $RM > 0$, $!PV$, $!PR$, $!CN$ in $N_{Att}$;
        if found then
            select as $RC$;
            increment $H$;
        else
            search for a node with max. $NoN$, $RM > 0$, $!PV$, $!PR$ in $N_{Att}$;
            if found then
                select as $RC$;
                increment $H$;
return $RC$;
```

4.3 Algorithm Description

The algorithm starts with a check of the remaining memory RM of the node. If it has enough memory space available (line 1), it performs a number of tasks, i.e., it creates a replica of the data item, decrements the replication degree, appends its node ID to the lists of previously visited nodes PV and previous replica nodes PR (lines 2-4). It also adds the node IDs of its 1-hop neighbors to the list of neighbors CN (line 5). However, if the node does not have enough memory space to create a replica, it simply appends its node ID to the list of previously visited nodes PV (lines 6-7) to avoid loops.

Selection heuristic for the best candidate node to create the next replica slightly differs for the data owner than any other node in the network. If the node is a data owner (line 8), it checks the replication degree. If the replication degree is greater than zero (line 9), i.e., more copies need to be created in the network, it selects a node with the maximum number of neighbors NoN with enough memory space to create a replica from the neighbors attribute table N_{Att} and the hop count is incremented (lines 10-13).

If a node receives a data item from some other node in the network, it performs the same steps to create a replica as defined above (lines 1-7). For the selection of best replica candidate node, it checks if the replication degree is greater than zero (line 15). If the condition holds true, it searches for a node that fulfills certain criteria, i.e., it has the maximum number of neighbors NoN, it has enough memory space available to create a replica, it's not a previous replica PR of the received data item, not previously visited PV and not a common neighbor CN (line 16). If such a node is found it is selected as the next replica node and the hop count is incremented (line 17-19). Otherwise, if no such node could be found among the uncommon neighbors, the replica node looks for a suitable node in the common neighbors. If a suitable node is found in the common neighbors it is selected to be the next replica node and the hop count is incremented (lines 20-24). For each data item generated in the network, this process continues until either the replication degree is met or no suitable candidate



(a) Total Replicas created = R (b) Total Replicas created < R (c) Total Replicas created < R

Figure 4.1: Illustration for R=3 under several scenarios: (a) All the desired replicas are created including data owner and avoiding common neighbors (b) Replication degree not met because of unavailability of enough memory at intermediate node (c) Replication degree not met due to unavailability of enough memory at data owner node.

can be found.

Since each node in the system, whether a data owner node or a replica node, has to search its neighbors to determine the best replica candidate node, we estimated the number of neighbors per node in our system. Assuming an omnidirectional antenna, each sensor node's communication field is a circle around it with fixed radius α (determined by its transmit power, refer to section 3.3). Therefore, the estimated number of neighbors per node in the system ($NoN/Node$) can be formalized as follows:

$$NoN/Node = \frac{\pi \times \alpha^2 \times n}{A} - 1 \tag{4.1}$$

Where $\pi \times \alpha^2$ gives the area of the circular communication field of a node, n is the total number of nodes in the system, and A is the area of the sensor field. For a given area of the sensor field and for a fixed transmit power of each node, we see that the number of neighbors per node in equation 4.1 depends on the system size, i.e., the total number of nodes n. Therefore, we can say that the worst-complexity of the proposed data replication algorithm is O(n) which continues for at most R times per data item.

4.4 Illustrative Scenarios

Figure 4.1 shows three illustrative scenarios to represent the operation of the proposed data replication algorithm for replication degree R=3. In this illustration, node 1 triggers replication process upon generation of data item D1. However, any node in the network may trigger replication upon data generation. In figure 4.1(a), all the desired replicas are created at nodes 1, 3 and 4. Each node selects the best replica candidate node among its neighbors based on the replica selection criteria explained above. While selecting the next replica candidate, node 3 has two possible options as nodes 2 and 4. However, it avoids node 2, since node 2 is a common neighbor with node 1 to ensure maximum replica spread.

In figure 4.1(b), node 1 selects node 4 to create a replica. However, by the time this message reaches node 4, it has already consumed its memory. Hence, instead of creating a local replica, it simply forwards the data to node 3. It can be observed that avoiding common neighbors is not possible at node 4; since the only available option is node 3 which is a common neighbor with node 1. Thus, to create maximum replicas it considers node 3 to create a replica. In this case, the replica spread is low.

In figure 4.1(c), node 1 has no memory to create a local replica. Thus, it selects node 3 from its neighbors to create a replica instead of discarding it. Two replicas are created at nodes 3 and 2; however, the third copy is dropped at node 2, since it can not find any other suitable node to create a replica. In both cases depicted in figures 4.1(b) and 4.1(c), the replication degree is not met.

4.5 Comparison Algorithms

For performance comparison, we implemented two closely related data replication mechanisms as detailed below.

Greedy:

Greedy is a memory-based fully distributed data replication algorithm available in the literature for homogeneous WSNs that utilizes limited neighbors' information to create replicas of data items [49]. In the implementation of Greedy, each node creates a local copy of the data item it generates if it has enough memory space available. Furthermore, it selects the best candidate among its neighbors, that has the highest remaining memory to create a replica. Replication process continues until the replication degree is met or no candidate node is available.

Random replication:

We implemented another replication algorithm called the random approach. In the implementation of the random replication mechanism, each node creates a local copy of the data item it generates if it has enough memory space available. However, to create a replica of the data item, it randomly selects a node among its neighbors, thus the name random replication. Similarly, as in the case of Greedy, the replication process continues until the replication degree is met or no candidate node is available.

4.6 Performance Analysis

4.6.1 Effect of increasing Replication Degree on Data Availability

Figure 4.2 shows the effect of increasing replication degree on data availability in the system, where it can be observed that data availability increases with increasing replication degree. This is due to the fact that, as more copies are created, it is more likely that at least a single copy survives in the presence of node failures. The trend is shown for different values of replication degree against multiple node failure rates. We can see that our proposed approach clearly outperforms the other two due to its efficient replica selection mechanism. Although, greedy performs better than the random approach because of its memory-based replica selection heuristic. However, random replication performs worst since random replica selection mechanism can not

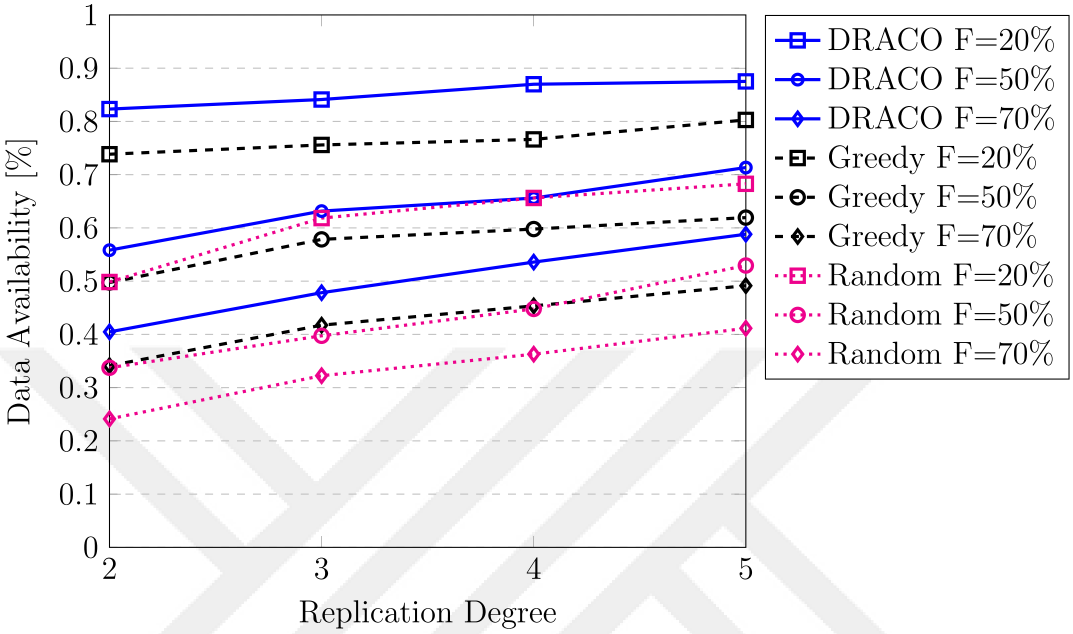


Figure 4.2: Effect of increasing replication degree on data availability in the system against node failure rates of 20%, 50% and 70% with 100 sensor nodes randomly deployed in 100 x 100 meter square region.

ensure maximum replication for each data item. It can be observed that, even in the worst case of 70% node failure rate, the proposed data replication algorithm can achieve approximately 60% data availability with a replication degree of 5. It improves data availability in the network with a maximum gain of about 15% and 34% (calculated by taking the average of corresponding data points in results) in comparison to greedy and random replication respectively.

4.6.2 Effect of increasing Node Failure Rate on Data Availability

Figure 4.3 shows the effect of node failures on data availability in the system. It can be observed that, as the node failures increase, data availability is reduced. However, with the help of data replication, a certain threshold of data availability can be maintained. From the figure, it can be observed that even if half of the nodes fail in the system, our proposed data replication mechanism can achieve a data availability of more than 70%. Whereas, Greedy and Random approaches can provide data

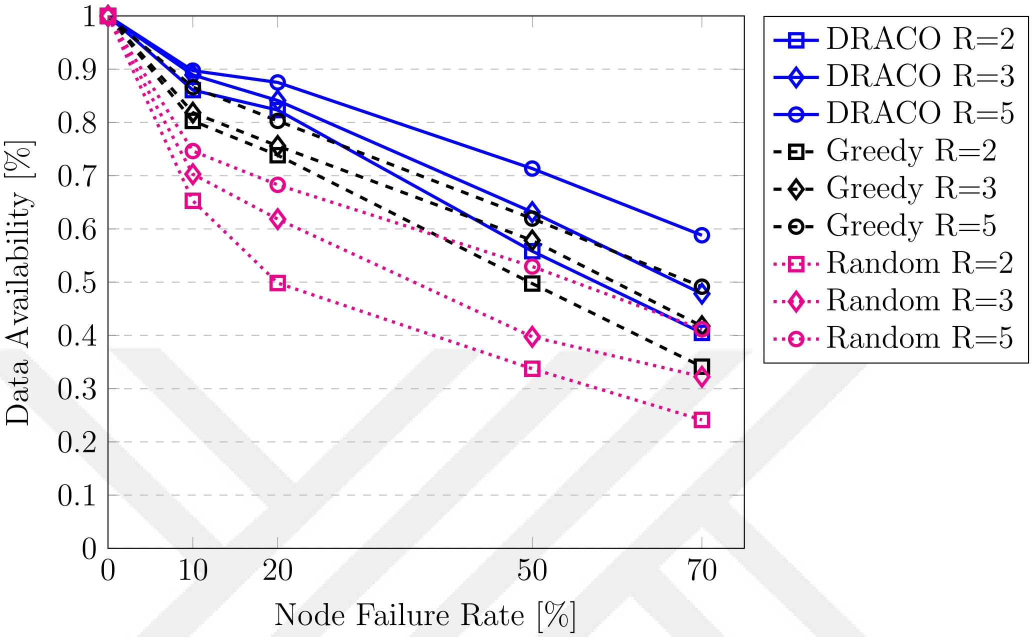


Figure 4.3: Effect of increasing node failure rate on data availability in the system under replication degrees of 2, 3 and 5 with 100 sensor nodes randomly deployed in 100 x 100 meter square region.

availability of slightly above 60% and 50%, respectively.

4.6.3 Effect of Node Density on Average Replicas Created in the Network

Figure 4.4 presents the average number of replicas created per data item. It is given as a function of the total number of nodes in the system. To find an optimal node density level, experiments were performed with varying number of nodes in the range of [20-200]. From the figure, it can be observed that, generally, for all three algorithms, the performance initially increases as the number of nodes increase. It is due to the fact that with a higher number of nodes in the system, it is more likely that a node finds a suitable candidate among its neighbors to create a replica which increases the total number of replicas. However, increasing number of nodes beyond a certain point degrades system performance. This behavior is due to a trade-off, i.e. increasing number of nodes may increase replica candidates, but at the same time, it also increases total data items generated, which adversely affects the system performance.

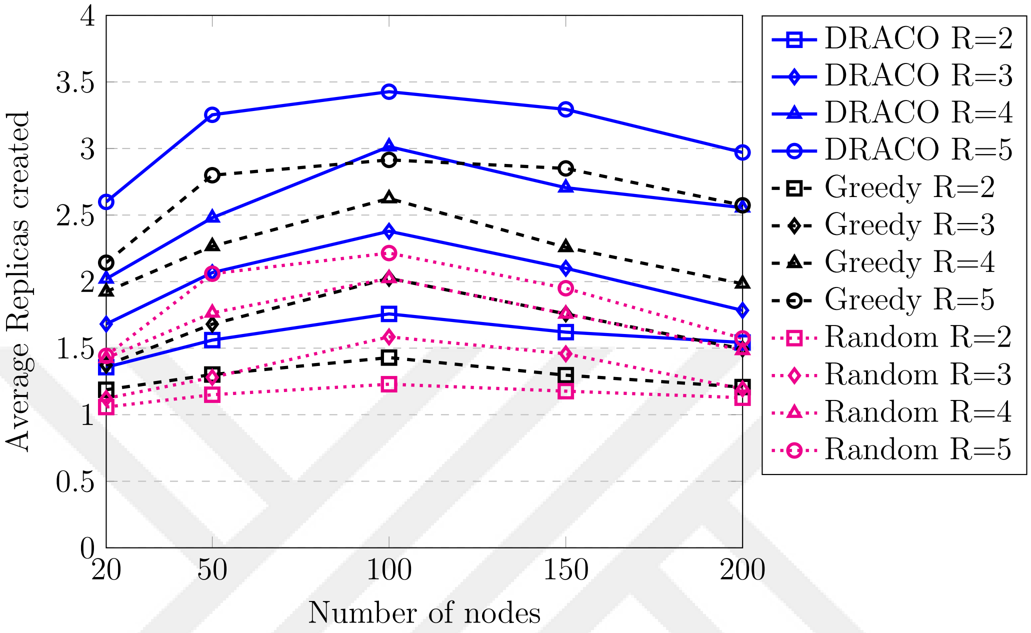


Figure 4.4: Actual Replication Degree achieved in the network for varying node densities. Number of nodes are varied in the range of 20-200 randomly deployed in 100 x 100 meter square region.

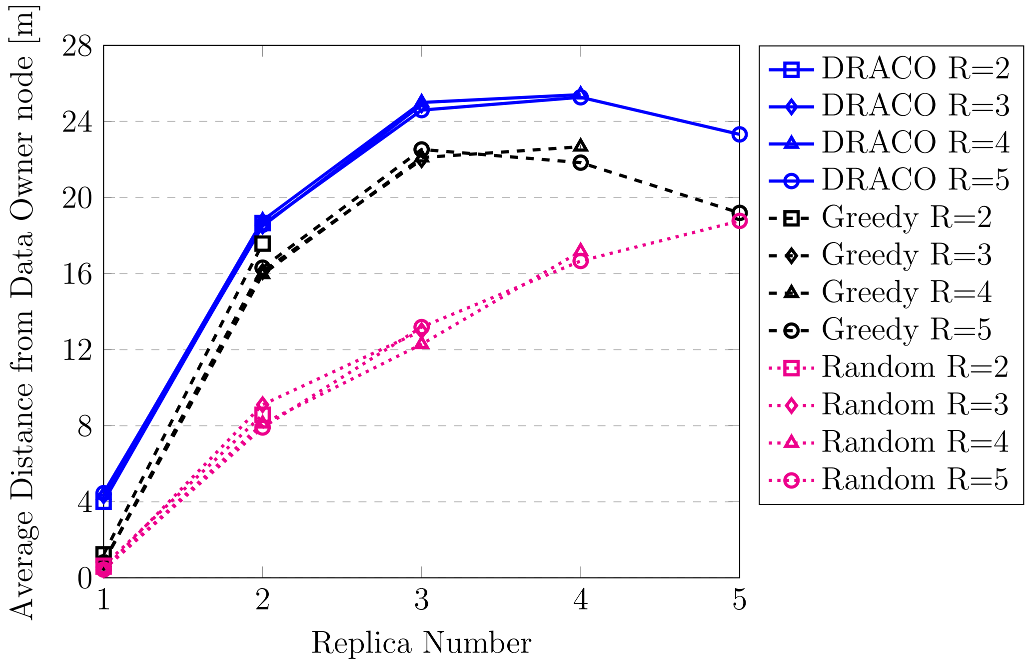


Figure 4.5: Replica distribution across the network in terms of average distance traveled by each successive replica from the data owner node as a function of the replica number.

From the figure, it can be observed that our proposed algorithm clearly outperforms the other two due to its intelligent decisions while allocating replicas. Greedy shows better performance than Random approach due to its memory-based replica selection criteria which ensures that the selected node has enough memory space to create a replica. Whereas, Random approach shows the worst performance due to its random replica selection strategy where some randomly selected neighboring node might not be able to ensure maximum replication due to unavailability of local memory space and suitable nodes. In comparison to Greedy and Random approaches, the proposed algorithm improves the average replicas created in the network with a maximum gain of about 18% and 40% respectively. Hence, it can be concluded that increasing node density may give a performance boost. However, beyond a certain level, it may turn out to be a curse.

4.6.4 Replica Distribution across the Network

Figure 4.5 shows the average Euclidean distance traveled by successive replicas in the network from the data owner node. Distance curves are drawn for different replication degrees as a function of the replica number. It can be observed that the trend is almost overlapping and increasing as the replica numbers increase for all the three algorithms. However, as the network saturates, higher values of replicas tend to diverge from this trend as it is not always possible to find a suitable candidate node at a higher distance. Comparative analysis shows that our proposed approach clearly outperforms Greedy and Random techniques as the replicas tend to spread better which determines the quality of data dissemination in the network.

Chapter 5

DATA COLLECTION

In this chapter, we provide details of the proposed data collection algorithm of the DRACO framework where a mobile sink node is used to efficiently collect the network data by randomly visiting sensor nodes in the field. The goal is to enhance data collection efficiency, i.e., collecting a higher percentage of network data by visiting a considerably lower number of network nodes.

5.1 Algorithm Overview

Algorithm 2 is invoked at the mobile sink node by the end of each data dissemination phase to collect the generated network data. The mobile sink node is deemed to be a more powerful node as compared to the rest of the network nodes with no significant resource constraints. The mobile sink node has no information about the location of the nodes and the amount and availability of network data. Therefore, by knowing only the diagonal coordinates of the sensor field, the sink node is allowed to follow any randomly chosen trajectory and visit sensor nodes to collect network data.

5.2 Algorithm Inputs and Output

As inputs, the algorithm requires the list of previously visited sites PVS, Diagonal coordinates of sensor field $C1, C2$, Maximum number of sites to visit M, number of sites visited S and the communication radius of sink node CR. As output, the algorithm returns the set of collected data items DC in each round.

Algorithm 2: Data Collection

Input: Previously visited sites PVS, Diagonal coordinates of sensor field $C1, C2$, Maximum number of sites to visit M, Number of sites visited S, Communication Radius of Sink node CR

Output: Set of collected data items DC

```
1  while ($S < M$) do
2      $X$ = generate a random site coordinates ($C1, C2$);
3      if ($X$ is not in $PVS$ and $X$ is not inside $CR$ at current site) then
4          move to $X$;
5          advertise presence with memory snapshot of sink node;
6          collect responses from the nodes inside $CR$;
7          select node with highest number of new data items NOT already
             available at sink;
8          send a data transfer request to the selected node for the new data items;
9          collect all the new data items from the selected node;
10         append newly collected data items to $DC$;
11         append $X$ to $PVS$;
12         increment $S$;
13 return $DC$;
```

5.3 Algorithm Description

The algorithm executes repeatedly until the predefined total number of sites M are visited (line 1). The sink node generates the coordinates of a random site inside the sensor field denoted as X by utilizing the diagonal coordinates of sensor field $C1$ and $C2$ (line 2). If the randomly generated site is not previously visited as well as it is outside the communication radius of the sink at its present location(line 3), it proceeds to the site (line 4) and performs a number of tasks. The sink node advertises its presence through a broadcast message along with its memory snapshot for the data items already available at the sink node (line 5). All the nodes in the vicinity receive this broadcast message and respond with the number of new data items they have not available at the sink (line 6). Upon reception of the messages from all the neighboring nodes inside the communication radius of the sink, the sink node selects the node with the highest number of new data items and sends it a data transfer request (lines 7-8) through a unicast message. Upon reception of the data transfer request message, the sensor node sends the requested data items to the sink node (line 9). The new data items are appended to the set of collected data items DC (line 10). Subsequently, the sink node appends the coordinates of the visited site to the list of previously visited sites to avoid loops and increments the number of visited sites S (line 11-12). This process continues until the sink has visited the required number of sites in the sensor field to collect network data. Finally, the algorithm returns the set of collected data items DC (line 13).

Since at each randomly visited site, the sink node has to determine the best node for data collection based on the availability of maximum new data items, therefore, the search is limited to the number of sensor nodes inside the communication radius of the sink node. A sink node is also assumed to have a circular communication field just like any other sensor node in the system. Therefore, similar to equation 4.1, the estimated number of sensor nodes available around the sink per site ($NoN_{sink}/Site$) can be formalized as follows.

$$NoN_{sink}/Site = \frac{\pi \times \alpha^2 \times n}{A} \tag{5.1}$$

Where $\pi \times \alpha^2$ gives the area of the circular communication field of the sink node, n is the total number of nodes in the system, and A is the area of the sensor field. Hence, for a given area of the sensor field and for a fixed transmit power of the sink node, we see that the number of neighboring sensor nodes around the sink per site in equation 5.1 depends on the system size, i.e., the total number of nodes n. Therefore, we can say that the worst-case complexity of the proposed data collection algorithm is O(n) which continues for the total number of sites visited by the sink node to collect the required amount of network data.

5.4 Illustration of Data Collection

Figure 5.1 shows the operation of the data collection mechanism through a mobile sink node. The mobile sink node visits randomly generated sites inside the sensor field. The figure shows three random sites visited by the mobile sink node. As soon as the mobile node reaches each site, it advertises about its presence as well as share its memory snapshot as a broadcast message in the area. Sensor nodes in the vicinity of the mobile sink node that receive this broadcast message, know about the presence of the mobile sink node and respond through an acknowledgment message also sharing the number of new data items available at each node which are not already collected by the sink node. The sink node receives these acknowledgment messages from all the nodes in its vicinity and creates a table for the available nodes around it with corresponding new data items each node holds as can be seen in the figure. This process is repeated at each site. At site 1, node 3 holds the highest number of new data items not already available at the sink node. Therefore, the sink node sends a data transfer request to node 3 for the new data items it holds through a unicast message. Similarly, at site 2, node 14 is found to be the best node and at site 3, node 7 has the highest number of new data items as can be seen by the corresponding data tables maintained by the sink node for each site. While visiting a new site, the mobile sink node checks that the new site should not be inside the communication radius of the existing site. However, a new site may have overlapping communication radius with any of the previously visited sites as can be seen for sites 3 and 1. In such a case, some of the nodes might be overlapping. Overlapping nodes which have not been previously visited can be counted as potential sensor nodes for data collection. However, a previously visited node is not added to the data table maintained by the sink node as can be seen in the table for site 3, node 3 is not added since it has been already visited at site 1. Finally, if at a given site, a sink is unable to find any sensor node in its communication radius or all the available nodes have been previously

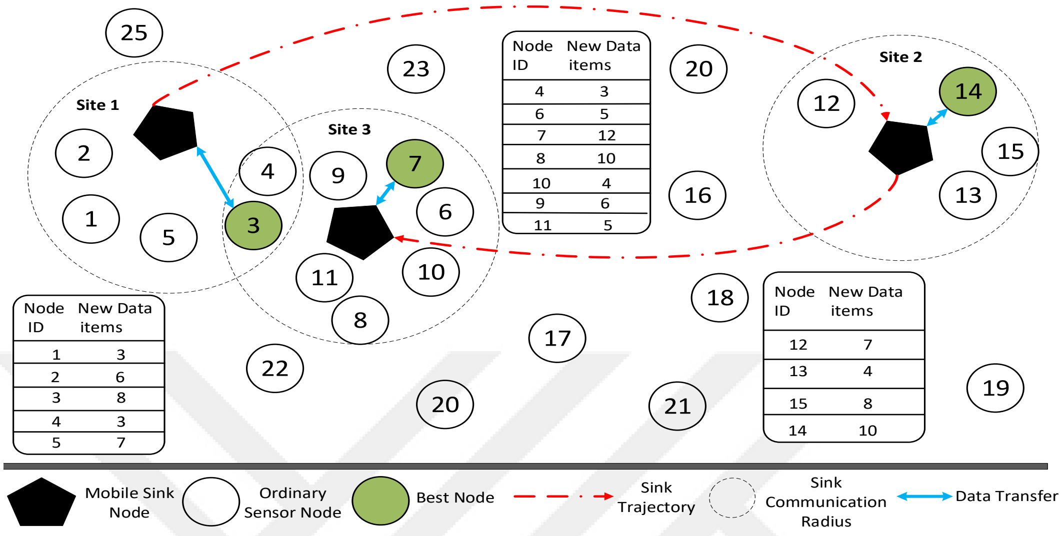


Figure 5.1: Illustration for data collection using a mobile sink node visiting randomly generated sites inside the sensor field and searching the best node based on the availability of new data items for data collection.

visited, it simply moves on to the next site.

5.5 Comparison Algorithms

For performance comparison of our proposed data collection algorithm, we implemented two most frequently used and state-of-the-art data collection algorithms for WSNs with uncontrolled sink mobility as defined below.

Self Avoiding Random Walk (SA-RW)

In the implementation of SA-RW, the mobile sink node visits a randomly generated site inside the sensor field and collects data items from the nearest sensor node. It keeps on visiting sites to collect network data until all or the desired number of nodes have been visited. SA-RW keeps track of the previously visited nodes to avoid revisiting the same nodes, thus the name self-avoiding. If at a given site, no node exists in the communication radius of the mobile sink or the available nodes have been already visited, it simply moves to the next randomly generated site [39].

Random Walk (RW)

In the implementation of RW, the mobile sink node visits a randomly generated site inside the sensor field and collects data items from the nearest sensor node. In contrast to SA-RW, it does not keep track of the previously visited nodes. Therefore, a node may be visited more than once, thus resulting in no data collection from that node since the data items from that have been already collected. It keeps on generating sites and visiting sensor nodes to collect network data until all or the desired number of nodes have been visited. If at a given site, no node exists in the communication radius of the mobile sink, it simply moves to the next randomly generated site [37].

5.6 Performance Analysis

We performed several experiments to validate and compare the performance of the proposed data collection algorithm coupled with the data replication mechanism (proposed in section 4) as detailed below.

5.6.1 Effect of Data Replication on Data Collection Efficiency

Figure 5.2 shows the effect of data replication on data collection efficiency. We performed a comparative analysis of our proposed data collection algorithm with SA-RW and RW (as previously described) algorithms to analyze the effect of increasing data replication on data collection efficiency. Experiments were performed with different replication degrees in the range of R=1-5. Where R=1 means no replication. As a result, only a single copy of each data item is stored on the data generating node only. From the figure, it can be observed that data collection efficiency increases as the replication degree is increased. However, at higher replication degrees the comparative increase in data collection efficiency becomes less. The maximum data collection efficiency is achieved with R=5 where the data replication algorithms attempt to create 5 copies of each data item in the system; thus increasing the data collection efficiency. It can be also observed that our proposed data collection algorithm clearly

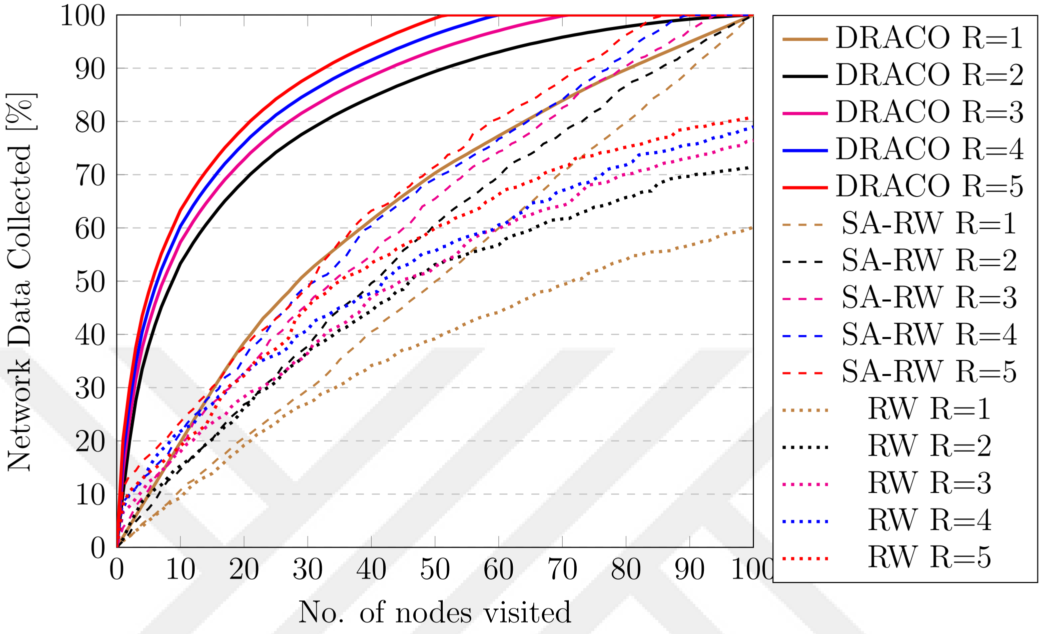


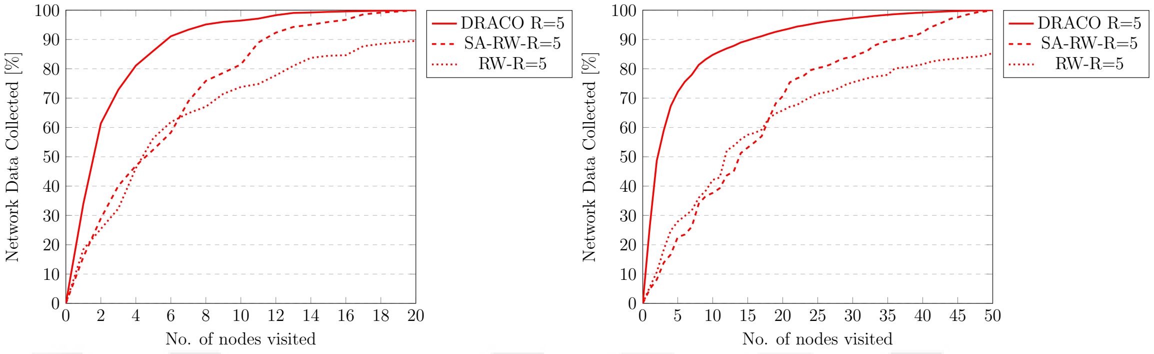
Figure 5.2: Effect of increasing replication degree (R=1-5) on data collection efficiency in the system with 100 sensor nodes randomly deployed in 100 x 100 meter square region.

outperforms the other two approaches which is due to the intelligent node selection heuristic of our proposed approach. It can be seen that SA-RW also gathers complete network data after visiting all the nodes. However, with a considerably lower data collection efficiency. Whereas, RW has the worst performance which is due to the fact that the mobile sink node randomly visits nodes in the sensor field to gather network data where some nodes might be visited more than once. Hence, it can not ensure 100% data collection after visiting the desired number of nodes. Interestingly, for the case with R=1, it can be observed that there is a slight curve in the trend. Some might think that with one copy per data item getting stored only at the data generating node, the curve should be a straight linear line. However, the reason for such a curved trend of data collection efficiency is due to the randomness in sensing intervals of the sensor nodes. Since sensor nodes generate data with a randomly assigned sensing interval in the range of [1-4] seconds, thus some nodes might end up generating more data than others. Thus, the mobile sink node visiting such a node with a lower sensing interval, that generates data more frequently, will be able to collect a higher percentage of overall network data. For performance comparison of the three approaches, if we have a look at the number of nodes visited by each approach to collect 80% of network data (R=5), it can be observed that our proposed data collection mechanism is able to gather 80% network data by visiting only 21 nodes. Whereas, SA-RW and RW approaches need to visit 60 and 96 number of nodes respectively in a system size of 100 nodes.

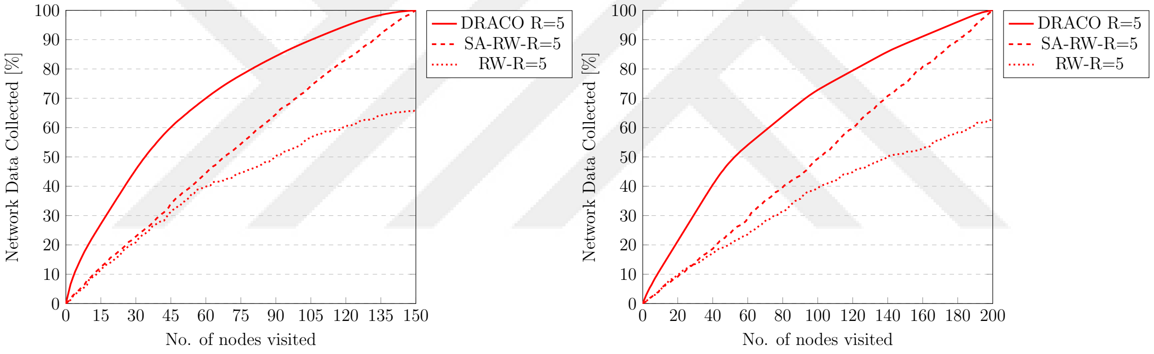
Since all the three approaches present the best performance for R=5, thus for the rest of the experiments, we present results for the case of R=5 only, for the sake of clarity and brevity.

5.6.2 Effect of Node Density on Data Collection Efficiency

We also analyzed the effect of node density on the data collection efficiency of all three approaches. Experiments were performed to analyze the comparative performance of data collection algorithms in sparse and dense networks by varying the number of nodes in a fixed area. Figure 5.3 shows the trends of data collection efficiency with varying number of nodes in the system ranging from 20 to 200 nodes randomly deployed in a 100 x 100-meter square region. Generally, it can be observed that the proposed data collection algorithm clearly outperforms SA-RW and RW for all system sizes due to its intelligent data collection mechanism. Although, SA-RW manages to collect whole network data by visiting all the nodes in the system. However, there is a clear difference in the data collection efficiency in comparison to our proposed algorithm. Whereas, RW cannot guarantee 100% network data collection. For a given threshold of network data collection, it can be observed that our proposed algorithm collects the required amount of network data by visiting fewer nodes as compared to the other two approaches. Also, it provides significantly impressive performance for all network sizes. However, in dense networks, it requires visiting higher percentage of network nodes as compared to sparse networks which is due to the fact that when the number of nodes increases the total amount of generated network data also increases



(a) Data collection efficiency (20 nodes) (b) Data collection efficiency (50 nodes)



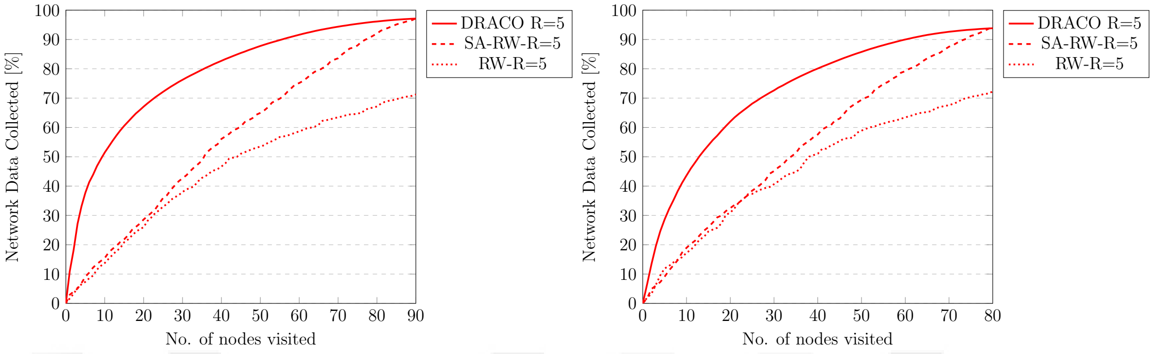
(c) Data collection efficiency (150 nodes) (d) Data collection efficiency (200 nodes)

Figure 5.3: Effect of increasing node density on data collection efficiency in the system. Number of nodes are varied in the range of 20-200 randomly deployed in 100 x 100 meter square region.

which require visiting a higher number of nodes in the system. Best data collection efficiency is achieved with a system size of 100 nodes (refer to figure 5.2) in comparison to other system sizes. The reason for such a behavior is the high replication degree achieved in a system size of 100 nodes as also discussed in section 3.

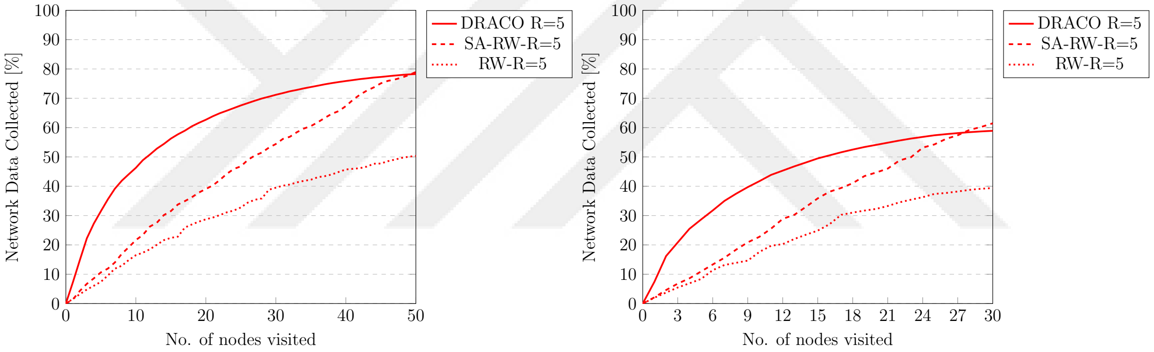
5.6.3 Effect of Node Failures on Data Collection Efficiency

As a prime objective, we leverage data replication for system robustness. Therefore, experiments were performed to analyze the effect of node failures on data collection



(a) 10% nodes failing

(b) 20% nodes failing



(c) 50% nodes failing

(d) 70% nodes failing

Figure 5.4: Effect of increasing node failure rate (10%, 20%, 50% and 70%) on data collection efficiency in the system with 100 sensor nodes randomly deployed in 100 x 100 meter square region.

efficiency. Efficiency was measured for different percentage of node failures where nodes follow a random node failure model as described in section 2. We measured the comparative performance of all three data collection algorithms for node failure rates of 10%, 20%, 50% and 70% for a 100 node network size with a replication degree of 5 as depicted in figure 5.4. It can be observed that, as the node failure percentage increases, the total number of nodes a mobile sink node can visit decreases because the sink can only visit the alive nodes and attempt to recover the data items of the whole network. It can be seen that, under all node failure cases, a mobile sink is unable to collect 100% network data which is due to the fact that although data replication preserves data items of failing nodes on their neighboring nodes. However, there are still few data items which are totally lost because those data items could not be replicated due to unavailability of suitable replica candidate nodes. It can be observed that our proposed data collection algorithm can collect a significant amount of network data as the node failure rate increases with comparatively much higher efficiency. For instance, in the case of 50% node failures, our proposed algorithm and SA-RW manage to gather 80% network data. However, our proposed algorithm has a much higher efficiency as compared to SA-RW. Whereas, RW performs worst in the presence of node failures as can be observed in all the scenarios. In the case of 50% node failures, it can hardly gather 50% network data.

Chapter 6

CONCLUSION AND FUTURE DIRECTIONS

After the advent of IoT technology, numerous smart solutions have been developed that are transforming the modern way of life. In the current era of IoT-based smart systems, data generated by the IoT-enabled devices is of utmost importance in order to benefit from the observations recorded through sensor nodes. Considering the importance of efficient data management, in this thesis, we first provided a comprehensive survey and taxonomy of state-of-the-art data replication protocols in IoT-based sensor systems. We presented a detailed discussion of the existing solutions by classifying them into subcategories of data retrieval, query balancing, system robustness, and data availability methods. Benefits and drawbacks of each method have been highlighted under all subcategories along with a comparison of solutions based on generic system properties.

We showed that these IoT-enabled devices are prone to failures due to their intrinsic properties as well as due to the harsh operating conditions. Thus, resulting in loss of valuable sensed data and making the system unreliable to accomplish the required task. Therefore, in this thesis, we propose a data replication and collection (DRACO) framework, composed of a fully distributed hop-by-hop data replication algorithm for a homogeneous WSN complemented with an efficient data collection mechanism using a mobile sink node for IoT-based sensor systems. In the proposed framework, sensor nodes coordinate to redundantly store sensed data inside the network to prevent data loss in the presence of node failures which results in enhanced data availability and increased system robustness. The proposed framework consists of fully distributed homogeneous sensor nodes with limited network information where nodes communicate with their immediate neighbors only through broadcast and point-to-point connections. Thus, no routing structure is needed to enable communication among network nodes which completely eliminates the routing overhead which is a key feature of the proposed approach. In the proposed data replication algorithm, each node takes local decisions regarding the selection of the best replica candidate node among its neighbors based on the available memory space and neighbors information learned through periodic broadcast messages. Thus, each data item generated is replicated at several nodes in the system. Hence, for any particular data item, if at least a single copy survives in the network, the data is preserved.

Furthermore, an intelligent data collection mechanism is proposed to collect the sensed data from the network nodes using a mobile sink node. A mobile sink node with uncontrolled mobility is proposed that follows a random trajectory to collect network data. This allows free sink mobility with zero mobility control overhead. However, while wandering across the sensor field for data collection, the sink node takes intelligent decisions to visit sensor nodes based on the availability of maximum new data items on surrounding nodes in its communication radius to enhance data collection efficiency.

We performed extensive simulations in NS3 to validate and compare the performance of the proposed framework. Comparative simulation results reveal that in comparison to two other state-of-the-art approaches, namely greedy and random replication techniques, DRACO improves data availability with maximum gains of about 15% and 34%, respectively. Similarly, it improves average replicas created in the network with maximum gains of about 18% and 40%, respectively. Furthermore, DRACO provides a better replica spread in the network in terms of average distance traveled by replicas from data owner which results in efficient data collection. The data collection mechanism has been tested for various experiments to analyze the effect of increasing data replication on data collection efficiency. We also analyzed the effect of increasing node density as well as node failures on data collection efficiency. It has been observed that DRACO clearly outperforms the commonly used uncontrolled sink mobility approaches namely random walk and self-avoiding random walk and provides higher data collection efficiency in different scenarios.

Hence, it can be concluded that data replication can significantly improve the performance of the network and make it more robust against node failures as well as facilitate data collection through a mobile sink. The downside, however, is the number of distinct data items that can be stored in the network decreases as more replicas are created. Moreover, the number of extra messages exchanged to create replicas are an added overhead.

Recently, energy efficiency has gained much attention in the research community. Therefore, as a future work, we plan to investigate the issue of energy consumption in the system since sensor nodes are battery-powered devices with limited energy resources. Considering the importance of energy efficiency, we provided a comprehensive survey and taxonomy of state-of-the-art energy aware data replication protocols for ad hoc nodes in IoT-based systems in [55]. We identified several design issues inherent to ad hoc networks to be considered for developing efficient data replication algorithms in the context of IoT. We also presented general analytical models for performance parameters along with the trade-offs involved. Furthermore, the available techniques have been classified into subcategories of network lifetime prolonging, network partition aware, scalable, and replica relocation methods. Although data replication significantly enhances system performance, we showed that designing high performance and energy efficient solutions for IoT-based ad hoc networks is a challenging task. Due to the trade-offs involved, it is not trivial to achieve both performance and energy efficiency for all situations. Therefore, we aim to balance the tradeoff between network lifetime and system robustness achieved through data replication in IoT-based sensor systems.

BIBLIOGRAPHY

[1] D. Miorandi, S. Sicari, F. De Pellegrini, and I. Chlamtac, "Internet of things: Vision, applications and research challenges," *Ad Hoc Networks*, vol. 10, no. 7, pp. 1497–1516, 2012.

[2] R. Amin, N. Kumar, G. Biswas, R. Iqbal, and V. Chang, "A light weight authentication protocol for iot-enabled devices in distributed cloud computing environment," *Future Generation Computer Systems*, vol. 78, pp. 1005–1019, 2018.

[3] C. G. García, D. Meana-Llorián, J. M. C. Lovelle, *et al.*, "A review about smart objects, sensors, and actuators.," *International Journal of Interactive Multimedia & Artificial Intelligence*, vol. 4, no. 3, 2017.

[4] P. Bellavista, G. Cardone, A. Corradi, and L. Foschini, "Convergence of manet and wsn in iot urban scenarios," *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3558–3567, 2013.

[5] D. G. Reina, S. L. Toral, F. Barrero, N. Bessis, and E. Asimakopoulou, "The role of ad hoc networks in the internet of things: A case scenario for smart environments," in *Internet of Things and Inter-Cooperative Computational Technologies for Collective Intelligence*, pp. 89–113, Springer, 2013.

[6] S. Al-Sultan, M. M. Al-Doori, A. H. Al-Bayatti, and H. Zedan, "A comprehensive survey on vehicular ad hoc network," *Journal of network and computer applications*, vol. 37, pp. 380–392, 2014.

[7] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *IEEE SoftCOM*, 2011.

[8] D. Hortelano, T. Olivares, M. Ruiz, C. Garrido-Hidalgo, and V. López, "From sensor networks to internet of things. bluetooth low energy, a standard for this evolution," *Sensors*, vol. 17, no. 2, p. 372, 2017.

[9] L. Feng, T. Qiu, Z. Sun, F. Xia, and Y. Zhou, "A coverage strategy for wireless sensor networks in a three–dimensional environment," *International Journal of Ad Hoc and Ubiquitous Computing*, vol. 15, no. 1-3, pp. 83–94, 2014.

[10] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer networks*, 2002.

[11] I. Yaqoob, E. Ahmed, I. A. T. Hashem, A. I. A. Ahmed, A. Gani, M. Imran, and M. Guizani, "Internet of things architecture: Recent advances, taxonomy, requirements, and open challenges," *IEEE wireless communications*, vol. 24, no. 3, pp. 10–16, 2017.

[12] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, "Internet of things (iot): A vision, architectural elements, and future directions," *Future generation computer systems*, 2013.

[13] P. Neamatollahi, M. Naghibzadeh, S. Abrishami, and M.-H. Yaghmaee, "Distributed clustering-task scheduling for wireless sensor networks using dynamic hyper round policy," *IEEE Transactions on Mobile Computing*, vol. 17, no. 2, pp. 334–347, 2018.

[14] Y. Yun and Y. Xia, "Maximizing the lifetime of wireless sensor networks with mobile sink in delay-tolerant applications," *IEEE Transactions on mobile computing*, vol. 9, no. 9, pp. 1308–1318, 2010.

[15] X. Wang and J. Li, "Improving the network lifetime of manets through cooperative mac protocol design," *IEEE Transactions On Parallel And Distributed Systems*, vol. 26, no. 4, pp. 1010–1020, 2015.

[16] S. H. Lee, S. Lee, H. Song, and H. S. Lee, "Wireless sensor network design for tactical military applications: Remote large-scale environments," in *IEEE MILCOM*, 2009.

[17] G. V. Merrett and Y. K. Tan, *Wireless sensor networks: application-centric design.* InTech, 2010.

[18] M. Kocakulak and I. Butun, "An overview of wireless sensor networks towards internet of things," in *Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual*, pp. 1–6, IEEE, 2017.

[19] T. K. Hui, R. S. Sherratt, and D. D. Sánchez, "Major requirements for building smart homes in smart cities based on internet of things technologies," *Future Generation Computer Systems*, 2017.

[20] E. Ahmed, I. Yaqoob, A. Gani, M. Imran, and M. Guizani, "Internet-of-things-based smart environments: state of the art, taxonomy, and open research challenges," *IEEE Wireless Communications*, vol. 23, no. 5, pp. 10–16, 2016.

[21] L. Filipponi, A. Vitaletti, G. Landi, V. Memeo, G. Laura, and P. Pucci, "Smart city: An event driven architecture for monitoring public spaces with heterogeneous sensors," in *Proceedings of the 2010 Fourth International Conference on Sensor Technologies and Applications*, pp. 281–286, IEEE Computer Society, 2010.

[22] M. Bonola, L. Bracciale, P. Loreti, R. Amici, A. Rabuffi, and G. Bianchi, "Opportunistic communication in smart city: Experimental insight with small-scale taxi fleets as data carriers," *Ad Hoc Networks*, vol. 43, pp. 43–55, 2016.

[23] T.-h. Kim, C. Ramos, and S. Mohammed, "Smart city and iot," 2017.

[24] B. Farahani, F. Firouzi, V. Chang, M. Badaroglu, N. Constant, and K. Mankodiya, "Towards fog-driven iot ehealth: Promises and challenges of

iot in medicine and healthcare," *Future Generation Computer Systems*, vol. 78, pp. 659–676, 2018.

[25] A. M. Rahmani, T. N. Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, and P. Liljeberg, "Exploiting smart e-health gateways at the edge of healthcare internet-of-things: a fog computing approach," *Future Generation Computer Systems*, vol. 78, pp. 641–658, 2018.

[26] C. Zhu, L. Shu, T. Hara, L. Wang, S. Nishio, and L. T. Yang, "A survey on communication and data management issues in mobile sensor networks," *Wireless Communications and Mobile Computing*, 2014.

[27] C. C. Aggarwal, N. Ashish, and A. Sheth, "The internet of things: A survey from the data-centric perspective," *Managing and Mining Sensor Data*, pp. 383–428, 2013.

[28] Y. Qin, Q. Z. Sheng, N. J. Falkner, S. Dustdar, H. Wang, and A. V. Vasilakos, "When things matter: A survey on data-centric internet of things," *Journal of Network and Computer Applications*, vol. 64, pp. 137–153, 2016.

[29] M. Younis, I. F. Senturk, K. Akkaya, S. Lee, and F. Senel, "Topology management techniques for tolerating node failures in wireless sensor networks: A survey," *Computer Networks*, 2014.

[30] M. Ma, P. Wang, and C.-H. Chu, "Data management for internet of things: Challenges, approaches and opportunities," in *IEEE GreenCom*, 2013.

[31] M. Abu-Elkheir, M. Hayajneh, and N. A. Ali, "Data management for the internet of things: Design primitives and solution," *Sensors*, 2013.

[32] K. Ahmed and M. A. Gregory, "Techniques and challenges of data centric storage scheme in wireless sensor network," *JSAN*, 2012.

[33] B. A. Milani and N. J. Navimipour, "A systematic literature review of the data replication techniques in the cloud environments," *Big Data Research*, 2017.

[34] T. Hara and S. K. Madria, "Data replication for improving data accessibility in ad hoc networks," *IEEE transactions on mobile computing*, vol. 5, no. 11, pp. 1515–1532, 2006.

[35] V. Martins, E. Pacitti, and P. Valduriez, *Survey of data replication in P2P systems.* PhD thesis, INRIA, 2006.

[36] W. B. Qaim and O. Ozkasap, "State-of-the-art data replication techniques in iot-based sensor systems," *IEEE GLOBECOM Workshop: Communication, Computing, and Networking in Cyber Physical Systems*, 2018.

[37] G. Maia, D. L. Guidoni, A. C. Viana, A. L. Aquino, R. A. Mini, and A. A. Loureiro, "A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks," *Ad Hoc Networks*, 2013.

[38] J. Zhang, J. Tang, T. Wang, and F. Chen, "Energy-efficient data-gathering rendezvous algorithms with mobile sinks for wireless sensor networks," *International Journal of Sensor Networks*, vol. 23, no. 4, pp. 248–257, 2017.

[39] M. Vecchio, A. C. Viana, A. Ziviani, and R. Friedman, "Deep: Density-based proactive data dissemination protocol for wireless sensor networks with uncontrolled sink mobility," *Computer Communications*, 2010.

[40] A. Carneiro Viana, T. Herault, T. Largillier, S. Peyronnet, and F. Zaïdi, "Supple: a flexible probabilistic data dissemination protocol for wireless sensor networks," in *ACM MASWMS*, 2010.

[41] A. C. Viana, A. Ziviani, and R. Friedman, "Decoupling data dissemination from mobile sink's trajectory in wireless sensor networks," *IEEE Communications Letters*, vol. 13, no. 3, pp. 178–180, 2009.

[42] T.-S. Chen, N.-C. Wang, and J.-S. Wu, "An efficient adjustable grid-based data replication scheme for wireless sensor networks," *Ad Hoc Networks*, 2016.

[43] J. Neumann, N. Hoeller, C. Reinke, and V. Linnemann, "Redundancy infrastructure for service-oriented wireless sensor networks," in *IEEE NCA*, 2010.

[44] S. Ishihara and T. Suda, "Replica arrangement scheme for location dependent information on sensor networks with unpredictable query frequency," in *IEEE Communications*, 2009.

[45] S. Ratnasamy, B. Karp, L. Yin, F. Yu, D. Estrin, R. Govindan, and S. Shenker, "Ght: a geographic hash table for data-centric storage," in *ACM WSNA*.

[46] B. Karp and H.-T. Kung, "Gpsr: Greedy perimeter stateless routing for wireless networks," in *ACM MCN*.

[47] R. Sarkar, X. Zhu, and J. Gao, "Double rulings for information brokerage in sensor networks," *IEEE TON*, 2009.

[48] T. Hara, R. Hagihara, and S. Nishio, "Data replication for top-k query processing in mobile wireless sensor networks," in *IEEE SUTC*, 2010.

[49] P. Gonizzi, G. Ferrari, V. Gay, and J. Leguay, "Data dissemination scheme for distributed storage for iot observation systems at large scale," *Information Fusion*, 2015.

[50] K. Piotrowski, P. Langendoerfer, and S. Peter, "tinydsm: A highly reliable cooperative data storage for wireless sensor networks," in *IEEE CTS'09*, 2009.

[51] K. Mekki, A. Zouinkhi, W. Derigent, E. Rondeau, A. Thomas, and M. N. Abdelkrim, "Usee: A uniform data dissemination and energy efficient protocol for communicating materials," *FGCS*, 2016.

[52] T. Hara, N. Murakami, and S. Nishio, "Replica allocation for correlated data items in ad hoc sensor networks," *ACM Sigmod Record*, 2004.

[53] "The ns-3 network simulator. http://www.nsnam.org/.,"

[54] W. B. Qaim and O. Ozkasap, "Draw: Data replication for enhanced data availability in iot-based sensor systems," *IEEE PICom, 16th IEEE International Conference on Pervasive Intelligence and Computing, (IEEE Best Paper Award)*, 2018.

[55] W. B. Qaim and O. Ozkasap, "A survey of energy aware data replication for ad hoc nodes in iot-based systems," *Elsevier, Ad Hoc Networks, (under revision)*, 2018.

VITA

Waleed Bin Qaim was born in Quetta, Pakistan on April 13, 1990. He received his early education and completed his high school from Garrison Academy, Quetta. He completed his B.Sc degree in Telecommunication Engineering with distinction from National University of Computer and Emerging Sciences (NUCES), Peshawar, Pakistan in 2011. He was awarded 4-years fully funded scholarship for his B.Sc degree at NUCES by the Ministry of IT (MoIT), Pakistan under its National Outreach Scholarship Program (OSP) 2007. After graduating in 2011, he joined Pakistan Telecommunication Company Limited (PTCL) as Assistant Manager in the Optical Fiber Systems Department, Quetta. He worked in the industry for 3 years and was also awarded honorarium by the Regional General Manager PTCL, for best performance in the region. Later on, he joined Balochistan University of IT, Engineering and Management Sciences (BUITEMS) as Lecturer in the Telecommunication Engineering Department at the Faculty of Information and Communication Technology (ICT) in 2015. To pursue higher studies, he joined MS program in Computer Science and Engineering at Koç University, Turkey in 2016 as a research and teaching assistant in the Distributed Systems and Reliable Networks (DISNET) research lab. He was awarded an overseas scholarship for MS at Koç from Higher Education Commission (HEC) Pakistan in 2016. During his stay at Koç, he worked on Data Replication techniques in Ad Hoc Networks, Wireless Sensor Networks, Internet of Things, and Distributed Computing Systems. He has co-authored 2 conference papers in the IEEE PICom 2018 and IEEE GLOBECOM 2018. His conference paper in the IEEE PICom 2018 received the IEEE Best Paper Award. He has also co-authored 2 journal papers under submission.