Internet of Things and Statistical Analysis

Chapter · January 2019

DOI: 10.1007/978-3-319-93557-7_8

1 author:

Ali Tasiran
Birkbeck, University of London
42 PUBLICATIONS **434** CITATIONS

SEE PROFILE

EAI/Springer Innovations in Communication and Computing

Fadi Al-Turjman *Editor*

Performability in Internet of Things

EAI
RESEARCH MEETS INNOVATION

Springer

EAI/Springer Innovations in Communication and Computing

Series editor
Imrich Chlamtac, CreateNet, Trento, Italy

Editor's Note

The impact of information technologies is creating a new world yet not fully understood. The extent and speed of economic, life style and social changes already perceived in everyday life is hard to estimate without understanding the technological driving forces behind it. This series presents contributed volumes featuring the latest research and development in the various information engineering technologies that play a key role in this process.

The range of topics, focusing primarily on communications and computing engineering include, but are not limited to, wireless networks; mobile communication; design and learning; gaming; interaction; e-health and pervasive healthcare; energy management; smart grids; internet of things; cognitive radio networks; computation; cloud computing; ubiquitous connectivity, and in mode general smart living, smart cities, Internet of Things and more. The series publishes a combination of expanded papers selected from hosted and sponsored European Alliance for Innovation (EAI) conferences that present cutting edge, global research as well as provide new perspectives on traditional related engineering fields. This content, complemented with open calls for contribution of book titles and individual chapters, together maintain Springer's and EAI's high standards of academic excellence. The audience for the books consists of researchers, industry professionals, advanced level students as well as practitioners in related fields of activity include information and communication specialists, security experts, economists, urban planners, doctors, and in general representatives in all those walks of life affected ad contributing to the information revolution.

About EAI

EAI is a grassroots member organization initiated through cooperation between businesses, public, private and government organizations to address the global challenges of Europe's future competitiveness and link the European Research community with its counterparts around the globe. EAI reaches out to hundreds of thousands of individual subscribers on all continents and collaborates with an institutional member base including Fortune 500 companies, government organizations, and educational institutions, provide a free research and innovation platform.

Through its open free membership model EAI promotes a new research and innovation culture based on collaboration, connectivity and recognition of excellence by community.

More information about this series at http://www.springer.com/series/15427

Fadi Al-Turjman
Editor

Performability in Internet of Things

Springer

Editor
Fadi Al-Turjman
Computer Engineering Department
Antalya Bilim University
Antalya, Turkey

ISSN 2522-8595 ISSN 2522-8609 (electronic)
EAI/Springer Innovations in Communication and Computing
ISBN 978-3-319-93556-0 ISBN 978-3-319-93557-7 (eBook)
https://doi.org/10.1007/978-3-319-93557-7

Library of Congress Control Number: 2018951410

This Springer imprint is published by the registered company Springer Nature Switzerland AG
The registered company address is: Gewerbestrasse 11, 6330 Cham, Switzerland

“To our families and our beloved friends.”
“To my wife and my little prince.”

Preface

We are living in an era where the Internet is becoming a global platform for the communication of machines and smart objects, on top of more familiar applications such as browsing the Web, emails, accessing multimedia sources and services, real-time distributed applications and many similar tasks.

With the application areas such as smart homes, smart cities, smart grids, connected autonomous cars, connected health, smart supply chain and precision agriculture, we can already consider Internet as combination of communication networks and networked objects. From this perspective, it is essential to understand the role of Internet infrastructure which will provide interconnection of physical objects with computing/communication capabilities on top of its well-known role as global backbone for worldwide information sharing.

Embedding of electronics into everyday physical objects, making them a significant part of the global network, has introduced the cyberphysical infrastructure and undoubtedly given rise to new opportunities for information and communication technologies (ICT). As a result, in the near future, significant investments are expected particularly in IoT technologies, products and services with the involvement of billions of IoT devices.

Of course, introduction and merging of this new phenomenon, which comes with relatively large scales and potential to cause additional traffic in significant levels, open up new challenges for researchers especially in terms of quality of service (QoS). Therefore, it is essential to consider these new-generation infrastructures, namely enabling technologies such as wireless sensor networks, various radio technologies and cellular infrastructures, radio-frequency identification (RFID), cloud services and fog computing facilities for evaluation and optimization.

It is possible to consider these systems from various perspectives for the evaluation process. Existing studies consider the pure performance characteristics such as response time, delay, capacity as well as energy efficiency or the overall system reliability as criteria for evaluation and optimization. It is even possible to consider the system security levels as a measure of QoS for specific applications.

The objective of this book is to present a survey of existing techniques for QoS evaluation, optimization and improvements of IoT systems. The main focus is on performability evaluation methods which combine the performance and availability/reliability-related issues unlike the existing pure performance and pure availability-based modelling approaches. The applications of performability in IoT, evaluation metrics, constraints and open issues about the addressed topic are included for discussion as well. Analytical modelling attempts for performability evaluation of IoT enabling technologies, open issues about the existing simulation tools as well as employed statistical models, evaluation of IoT applications in various fields and the future of performance as well as availability/reliability modelling of IoT infrastructures are considered critically. This conceptual book, which is unique in the field, will assist researchers and professionals working in the area of performability to better assess the proposed IoT paradigms which are already beginning to be a significant part of the global infrastructure.

Antalya, Turkey

Fadi Al-Turjman
Enver Ever

Contents

Chapter 1
Performability Analysis Methods for Clustered WSNs as Enabling Technology for IoT

Enver Ever

1.1 Introduction

Growing numbers of systems such as vehicles, environment monitoring sensors, wearable health monitoring devices, augmented reality using devices as well as computers and smartphones are being connected to the Internet at rapid rate realizing the idea of the Internet of Things (IoT) [2, 6]. The investment through the use of IoT for various business areas has also been quite impressive. Companies are interested to employ IoT for increasing customer satisfaction, improving the overall quality and of course to reduce the costs involved. When the most valuable cases to involve IoT solutions are investigated [13], the analysis shows that the systems with ability to use sensors for the prediction of reliability-related features (time of failure and need for maintenance) of machinery, self-optimizing production, automated inventory and supply chain management, remote health monitoring, smart grids and meters, track and trace, connected cards, distributed generation and storage, fleet management and demand response are the most popular ones. The investment in these areas is predicted to reach 267 billion dollars particularly on IoT technologies, products and services (application development, device hardware, system integration, data storage, security and connectivity) [13]. That is expected to happen with the involvement of more than 24 billion IoT devices which would mean that there will be approximately four devices for every person. In return, the investment on all these IoT infrastructures is expected to generate 13 trillion dollars as early as by 2025 [13, 22].

The IoT systems are expected to improve the existing autonomousness even further by the capacity they offer to gather and convey data in the absence of user-

E. Ever (✉)
Computer Engineering Middle East Technical University, Northern Cyprus Campus, Guzelyurt, Mersin, Turkey
e-mail: eever@metu.edu.tr

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_1

to-user or user-to-computer interactions. With these features, it is possible for the people to have close interaction with the physical world based on the activity of the sensor nodes [5]. In other words, wireless sensor networks (WSNs) are an integral part of IoT [1, 19]. As the key technology for IoT, WSNs form the digital skin, providing a virtual layer where the information about the physical world can be accessed by interconnected computational systems. The benefits and necessity of connecting WSNs and other IoT elements go beyond the concept of remote access by facilitating heterogeneous information systems that can collaborate and provide common services. This integration makes WSNs invaluable resources for realizing the vision of the IoT.

The sensor nodes are equipped with limited resources in terms of queue capacity and power supply. Therefore, their capabilities in terms of processing and radio communication are also restricted. Once deployed, sensors can work unattended and transmit the gathered information to the sink directly or via multiple hops through other relay sensor nodes in the network. WSNs QoS demands are highly dependent on the applications they are being used for. The demands of the application are influenced by the complexities of the deployment environments, the nature of the information being collected and the characteristics of the phenomena being observed. Typical applications vary from target tracking to industrial automation, smart city transport systems, smart agriculture, disaster monitoring, assisted living and many others [6, 11, 19, 39]. The emergence of IoT has been a significant enhancement for the use of WSNs deployment in nearly all areas requiring any form of monitoring.

The mass numbers of interconnected devices and all the investment made which targets effective use of IoT cause a great demand on the underlying communication networks and affect variety of factors related with the expected quality of service significantly. Keeping this in mind, it is essential to design and develop service models that ensure appropriate level of QoS in terms of delay sensitivity, capacity, reliability and so on for IoT applications. On the other hand, the diversity of IoT application environments, that demand different QoS, make performance and availability evaluation of WSNs a very challenging task considering that majority of such network scenarios depend on battery-powered nodes with limited energy, computation and communication facilities [12, 40]. Furthermore, sensor node and communication link failures are probable in WSNs similar to other communication networks. The failures may result in network degradation in terms of availability and performance. Moreover, for data intensive applications, limited queue capacity may result in high packet loss, further degrading performance below the expected QoS levels. In view of these factors, WSN designers ought to consider fault tolerance, operating environment, scalability, network topology and energy efficiency.

Since pure performance models tend to overestimate the ability of the system to perform, and pure availability models tend to be too conservative [36], a composite study for performance and availability (performability) is necessary for realistic evaluation and optimisation of WSNs. Furthermore, prolonging the lifespan of WSNs is a well-investigated research topic [11–40] and a widely used energy saving technique is to place idle nodes into sleep mode [12, 39]. The sleep state

guarantees low power consumption by switching off most device components like microprocessor, memory and radio frequency (RF). However, even though significant energy saving may be achieved, there exists a trade-off between the node energy savings versus the network performance in terms of throughput and data delivery delay [12]. These kinds of operative concerns should also be considered together with performance and reliability characteristics to present realistic performability models.

This chapter considers existing methods for performability evaluation and potential application of these methods in clustered WSNs as one of the core elements of IoT systems. Proposed methodologies for modelling and evaluation of sensor networks which are widely used for IoT applications are investigated by integrating performance and availability/reliability studies in the presence of node and link failures. The effects of replacement and restoration of the failing nodes are also considered. Stochastic performability models which consider challenges resulting from resource constraints and operation dynamics are presented and discussed in detail. Performability models considered are capable of incorporating sleep schedule schemes implemented in the MAC layer, effects of limited queue capacities and challenges resulting from channel and node failures and restoration. Numerical results are presented comparatively with results obtained from pure performance analysis in order to show the importance of performability analysis.

The remainder of the chapter is organised as follows. Section 1.2 provides a detailed overview of performance and availability analysis approaches while Sect. 1.3 focuses on existing analytical modelling attempts for WSNs. In Sect. 1.4, the models employed for pure performance, and performability evaluation are discussed for clustered WSN systems. In Sect. 1.5, pure performance and performability results are presented and discussed. Finally, in Sect. 1.6, the conclusion remarks are presented.

1.2 Existing Methods for Performance Evaluation

In this section, commonly used analytical modelling techniques for performance and availability modelling are considered. Pure performance and availability models are classified and explained.

1.2.1 Pure Performance Evaluation Models for Multiserver Systems

One popular way for the evaluation of any multiserver communication system is to use pure performance evaluation techniques. These approaches assume that the system considered is available at any time. In other words, potential breakdowns or failures of systems considered are not taken into account.

Queuing theory and Markov birth and death processes are commonly used in performance evaluation. In case of open queuing networks, there are one or more sources of job arrivals and correspondingly, one or more sinks that await the jobs departing. On the other hand, in a closed queuing network, jobs neither enter into nor depart from the system. In closed queuing networks, arrivals and departures can be accepted as feedbacks [34]. Kendall's notation is commonly used for representation of a typical queuing system. It uses six parameters to specify a queuing system. The notation takes the form $A/S/m/B/K/SD$, where A is the type of arrival process distribution, S is the service process or the service time distribution, m is the number of servers, B is system capacity, K is the population size and SD is the service discipline of the queue. Markov processes are commonly used for inter arrival and service time distributions.

Since while Markov processes evolve, future states (future state evolution or probability of all possible future states) are not dependent on the past states (even if there is dependence it is only on the present state); to predict the future of a continuous-time Markov chain (CTMC), it is sufficient to know the current state. It is not necessary to know the past states, how it has come to the present state or how long it has been in the present state. In other words, the time spent in a state, which is another random variable, has a memoryless distribution [34]. The discrete state Markov processes are usually represented by integers, and from state n it can only change to state $n + 1$ or state $n - 1$. A typical birth and death process is given in Fig. 1.1, where b_i and d_i are the birth and death transitions originally for state i, respectively. For such a simple CTMC, it is possible to determine the state probabilities (P_i) by using the relations given as:

$$P_{i+1} = \frac{b_i}{d_{i+1}} P_i, \sum_i P_i = 1 \tag{1.1}$$

Once the steady-state probabilities are found, well-known queuing theory formulae can be used in order to obtain various pure performance evaluation measures.

It is possible to expand existing theories and terminology given in Queuing theory and Markov process analysis for performance and availability modelling of communication systems. For wireless communication systems, under certain assumptions it may be possible to obtain a simple exact solution for the joint queue

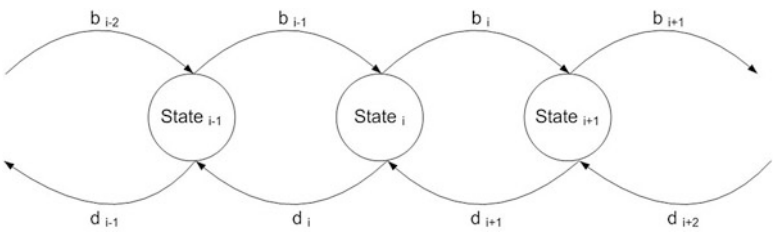

Fig. 1.1 Markov birth and death processes

length distribution in a separable form. This form is called as the product form [34]. In [18], a class of parallel processing systems where jobs are subdivided to several asynchronous tasks are presented. Then, the jobs of this non-product-form network are iteratively approximated by a sequence of product-form networks. Another development in product-form theory has been introduced in [17]. In this study, Gelenbe has considered performance analysis of resource request and allocation models with positive and negative signals (arrivals). These models have given rise to product-form networks with positive and negative customers. The main drawback of this approach is the fact that modelling of parallel and distributed systems usually does not lead to product forms.

The performance modelling of multiserver systems with multiple queues usually leads to multidimensional models, and various analytic-algorithmic methods have been developed to solve multidimensional queuing systems. The two most important of these methods are well-known Matrix Geometric method [28] and Spectral Expansion [8] method.

The analysis of a system from the pure performance point of view tends to be optimistic. This mainly happens since it is assumed that the system under study never fails and continues handling incoming request in best possible configuration. However, in many multiserver systems, especially when wireless communication systems are considered, failures are expected and they have significant effects on the system's performance since in case of failures some delay stages, and/or stages where the system is in a degraded mode are possible. Also, when the servers in a system are prone to breakdowns and repairs, the system does not have a simple product- form solution. This is because the irregularities caused by server breakdowns and repairs affect performance and dependability of the system significantly.

1.2.2 Availability Models for Multiserver Systems

The reliability of a system can be defined as the ability to perform a required function or operation under certain conditions within given time intervals. On the other hand, availability is the ability of a system to be in a state to perform a function or an operation at a given instant of time, or at any instant of time within a given time interval. An important difference between reliability and availability is that reliability refers to failure-free operation during an interval, whereas availability refers to failure-free operation at a given instant of time [34]. Mathematical definition of instantaneous availability or point availability $A(t)$ of a component which is equal to the probability that the component is properly functioning at time t can be given by:

$$A(t) = R(t) + \int_0^t R(t-x)m(x)dx \tag{1.2}$$

where $R(t)$ is the probability of having no failure in interval $(0, t]$ and $m(x)$ is the repair density. The system is available either if no failure occurs in interval $(0, t]$, or failure occurs but repair of the system is completed before time t. The mathematical definition of limiting availability can be expressed through the following well-known equation:

$$A = \lim_{t \longrightarrow \infty} A(t) = \frac{\text{MTTF}}{\text{MTTF} + \text{MTTR}} \tag{1.3}$$

where MTTF is the mean time to failure and MTTR is the mean time to repair. Fault-tolerant systems provide high availability through their ability of operating continuously, where short down times during their operation can be tolerated. In these systems, both preventive and corrective maintenance can be performed to obtain the desired level of service. Since for many of the sensor-based IoT systems, the operation should be continuous with replacement of failed nodes, or reconfiguration of wireless characteristics, we can easily classify them as fault-tolerant multiserver systems. Trivedi groups model types used to study availability under three main titles [34] as:

1. Combinatorial model types
2. State-space models
3. Hierarchical models

Combinatorial type of models can capture conditions that cause system failures, in terms of the structural relationships between the system components. The state-space representation of a system is not employed at all. They are known to be quite effective in determination of safety critical components and probabilities of system failures. Typical examples for these types of models are reliability block diagrams (RBD), reliability graphs and fault trees. The RBDs graphically represent how the components of a system are reliability-wise connected. Reliability graphs instead use nodes and a number of directed edges between them, where each edge represents a component that can fail. Fault trees are very popularly used in risk and reliability evaluation of complex systems from space shuttles to infrastructure of communication networks where graphic models of the pathways within a system that can lead to a foreseeable failure are illustrated. Among the combinatorial type models, the fault tree analysis is most widely used due to its expression power, and applicability to complex systems. However, drawing a fault tree can be a cumbersome task, and requires a great amount of attention and caution to represent a system correctly. All types of combinatorial models can be solved using fast algorithms; however, they all involve the strong assumption of independence between system components. In other words, they assume that the failure or repair of a component does not affect other components. Also, in combinatorial type models, it is assumed that there are as many repair facilities as necessary.

In many practical systems, dependencies have been observed among system components, and repair facilities can be restricted. Therefore, for more realistic

evaluation of complicated interactions among components, state-space models are preferred where a description of a configuration of states is employed. State-space models consist of states and transitions between these states. Markov chains, Markov reward models, stochastic reward nets and petri nets are well-known state-space-based models.

Pure availability concepts for analysis of communication networks are known to be very conservative since they are not able to consider different levels of performance especially for fault-tolerant systems. A more realistic analysis method has been introduced in [7] and a conceptual framework of performability has been considered by Meyer [23]. Beaudry has proposed (1978) [7] a simple method for computing the distribution of performability in a Markov reward process. The combined evaluation of performance and availability is called performability. This modelling technique is useful especially for the systems which can operate in a degraded mode in case of breakdowns and failures.

1.2.3 Performability Models for Multiserver Systems

Performability models are effectively used to combine performance and availability/reliability concerns as they can specify the amount of work that will be performed in a given interval while the failures and repairs affect the system. For modelling WSNs, it is important to consider and analyse concerns such as limited queuing capacities, the mean response time, mean number of requests in queue and the probability of an incoming request to be blocked. Quasi-birth and death (QBDs) processes are a special class of finite or infinite state CTMC and they can be used to model multiserver systems including WSNs with bounded queuing capacities.

It is possible to develop an effective and accurate analytical model for performability measures of multiserver systems by using a two dimensional representation of the states of the system, in a Markovian framework [8, 12, 31]. One approach to employ these representations is to use a pair of integer-valued random variables, $I(t)$ and $J(t)$, specifying the server configuration (can also be termed, operative state of a multiserver system) and the number of requests in the system, respectively.

In general, if there are $N + 1$ server configurations, represented by the values $I(t) = 0, 1, \ldots, N$, these $N + 1$ configurations can be used to represent the possible operative states of the model where $I(t), t \geq 0$, is an irreducible Markov process. $J(t)(\leq L)$ is the total number of requests in the system at time t, including the one(s) in service. Then, $Z = \{[I(t), J(t)]; t \geq 0\}$ is an irreducible Markov process on a lattice strip (a QBD process), that models the system. Its state space is, $\{0, 1, \ldots, N\} \times \{0, 1, \ldots, L\}$ (where L is the queue capacity). Once the steady-state probabilities of such a system are computed, it is possible to analyse and evaluate the system under study from a performability measure's point of view.

1.3 Performability Modelling of Wireless Sensor Networks

The restricted resources of WSNs such as queue capacity, and energy, cause limitations to their capabilities in terms of processing and radio communication. Various IoT applications such as smart city transport systems, smart agriculture, disaster monitoring, assisted living, target tracking, industrial automation and many others [6, 11, 19, 39] introduce various QoS demands which are also dependent on the complexity of the deployed environments.

Performance and availability evaluation of WSNs is a challenging task mainly because of the diversity of IoT application environments and different demands in terms of QoS. Because of the energy-related restrictions, the evaluation of network lifetime should be considered in the modelling attempts as well [12, 40]. Furthermore, sensor node and communication link failures are probable in WSNs similar to other wireless communication networks. As discussed in the previous sections, these failures can cause serious degradations in terms of availability and performance. Moreover, especially when data intensive applications are considered, limited queue capacity may result into high packet loss further degrading performance below the expected QoS levels. In addition, for WSNs a widely used energy saving technique is to place idle nodes into sleep mode [12, 39] where low power consumption is provided by switching off most device components like microprocessor, memory and radio frequency (RF) components. However, reducing the basic functionalities introduces a trade-off between the node energy savings and the network performance in terms of throughput and data delivery delay [12]. Therefore, when WSNs which are essential parts of IoT applications are considered for modelling, fault tolerance, operating environment, scalability, network topology and energy efficiency should be taken into account.

The variety of application areas, operational challenges and varying demands makes WSNs interesting for the researchers and scientists in areas of distributed computing, communication and software engineering. However, most of the existing studies consider WSNs for evaluation from a pure performance or pure availability point of view. For example, in [21] a relay network used for IoT connectivity in LTE/LTE-A systems is modelled as a single server queuing system with Poisson packet arrival process for pure performance analysis. Similarly in [14], M/M/1/C, M/M/1 and M/M/n/K queuing models are employed to model Edges Computing, Cloud Gateway and Physical Servers, respectively, as parts of fog computing infrastructure for IoT. On the other hand, while performance evaluation is considered, the sleeping states of WSN nodes are also taken into account for presenting analytical models in [12, 39]. In [11], CTMC representation of the system is employed in order to evaluate the framework with adaptive duty-cycling scheme for sensor networks. The availability- and reliability-related concerns of WSNs have also attracted some attention. A Markov process-based model for a fault-tolerant network was proposed in [26] and an automation of fault tree generation for evaluating reliability and availability of WSNs for industrial applications is considered in [33]. Sleep/active scheduling mechanisms are considered for studying reliability as well in studies such as [37].

Noting that pure performance models tend to overestimate the ability of the system to perform, and pure availability models tend to be too conservative [36]; a composite study for performance and availability (performability) is necessary for realistic evaluation and optimisation of WSNs. Conceptual performability frameworks are used in modelling and analysis of a number of computer and communication networks [15, 16, 35]. Employing similar methodologies for WSNs would allow the researchers to obtain more realistic models and measures for evaluation.

1.4 System Description and Modelling for Performability Evaluation

The clustered WSN infrastructures are modelled for pure performance and performability evaluation in this section.

1.4.1 System Description

It is possible to consider a typical scenario of clustered WSNs in order to demonstrate the difference between pure performance evaluation and more realistic performability modelling. Let us assume that we have M number of homogeneous, stationary sensor nodes with omnidirectional antennas which are uniformly distributed. Each cluster would have a single cluster head (CH) which is responsible for coordination and in total there are K clusters. The member nodes of a cluster communicate directly with their CH. The CH in turn acts as a link between these nodes and the sink either using single hop (directly) or multi-hop (through other intermediate CHs) routing protocols.

The CH operation can be transferred to other nodes following various measures such as energy levels and other metrics deemed appropriate [27]. It is also possible to introduce redundancy by deploying sensor nodes that remain inactive until there is a need to replace a failing node [26]. In studies such as [30] and [31], performability modelling of cluster-tree network topology such that for every cluster, there is at least one connecting route to the sink is considered. The network topology of the system considered in this study is given in Fig. 1.2.

Similar to any other communication network, this system is also subject to channel and node failures that can be resulting from errors in software configurations, system vulnerability attacks, hardware and link degradations. Renewable power sources, and energy harvesting methods (solar cells, vibration approaches and thermoelectric generators) which are commonly used to recharge the batteries can also be employed to minimise failures caused by power depletion [25].

Due to power constraints, sleep/active operations are widely used to conserve the limited energy resources [4, 12, 27, 39], in WSNs. The sleep scheduling implementations used in WSNs further complicate the operation dynamics of the

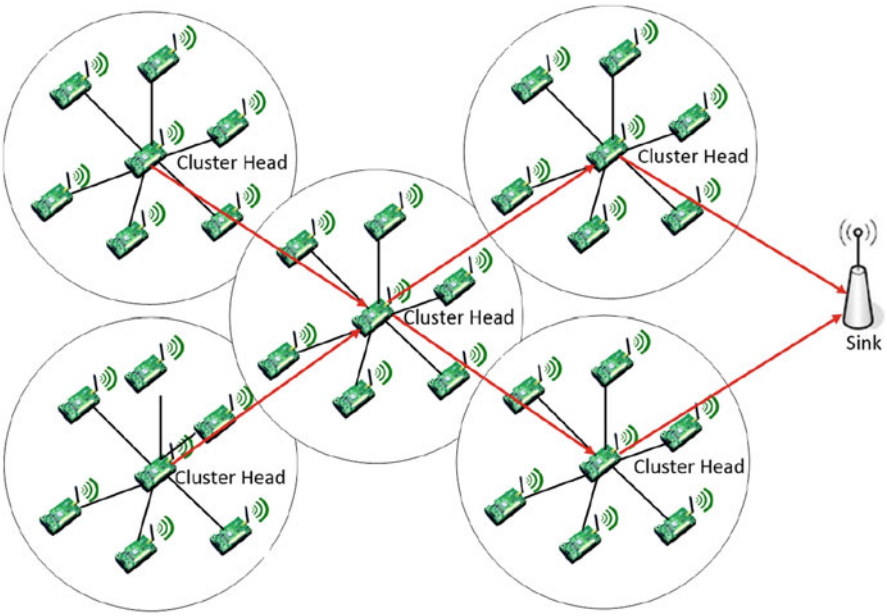

Fig. 1.2 Network topology of the reference system scenario

sensor nodes and the overall network. The two main sleep scheduling approaches used include adaptive duty-cycling schemes implemented at the MAC layer to adjust sleep–wakeup periods based upon the observed operating conditions [3, 4, 38] and the ON-demand wakeup scheduling implemented using a second low-power radio transceiver [20, 29]. The second low-power transceiver is used to monitor the channel for packet arrivals while the main radio transceiver is switched off. In low traffic application areas, on-demand sleep scheduling scheme is preferred because of its ability to switch the node ON when a new packet arrives and allow the node to enter sleep mode automatically after servicing the last packet.

1.4.2 System Modelling

In this section, the ON-demand scheme and corresponding operative states are employed to model the system presented in Fig. 1.2. The model developed considers four main operative states such as active, sleep, node failure and channel failure. Packet arrival time at the CH is assumed to follow Poisson distribution with rate λ and service time is assumed to follow exponential distribution with a mean of $1/\mu$ where the service discipline is First Come First Served ($FCFS$). Since sensor nodes have a finite queue capacity, any data packets arriving when the buffer is full are dropped and considered lost.

In WSN clusters, the CH expects incoming traffic from the N sensing nodes within the cluster communication radius (d). On top of that, once processed at the CH, the packets are forwarded to the sink directly or through an intermediary CH (Node r) which causes another incoming arrival stream for CHs. The CH has limited queue capacity (L). Assuming that kth node is the CH, jobs leaving node k are rerouted to node r with probability $q_{k,r}$ for service at node r, otherwise $q_{k,r} = 0$ if jobs are not routed. It is assumed without loss of generality that as far as the queue length distributions are concerned $q_{k,k} = 0, (k = 1, 2, \ldots, K)$. Also, $q_{k,K+1} = 1 - \sum_{r=1}^{K} q_{k,r}$ is the exit probability from the system after a job is serviced at node k. The exit probability $q_{k,K+1}$ is assumed to be non-zero for at least one value of k. Q is the routing probability matrix of size $K \times K$, such that $Q_{k,r} = q_{k,r}; (1 \leq k, r \leq K)$. Considering this notation, the total arrival rate (λ_k) at CH (node k) can be presented as the sum of external and internal traffic rates and can be expressed as:

$$\lambda_k = \sigma_k + \sum_{r=1}^{K} \lambda_r q_{r,k}; \qquad k = 1, 2, \ldots, K \tag{1.4}$$

where σ_k is the sum of all internal arrivals and may be expressed as: $\sigma_k = \sum_{n=0}^{N} \lambda_n q_{n,k}; \quad n = 0, 1, 2, \ldots, N$. In order to define the total arrival rates for each node, the row vectors $\lambda = (\lambda_0, \lambda_1, \lambda_2, \ldots, \lambda_N)$ and $\sigma = (\sigma_0, \sigma_1, \sigma_2, \ldots, \sigma_N)$ can be employed. Let E_k be the unit matrix of size $K \times K$, then $\lambda(E_k - Q) = \sigma$. Letting $\hat{\mu_k}$ and $\lambda_{k,e}$ be effective service and arrival rates, respectively. For stable systems, the condition $\hat{\mu_k} > \lambda_{k,e}; k = 0, 1, 2, \ldots K$, is necessary.

The system considered is modelled from a pure performance point of view in Fig. 1.3. Since failures are not considered, a typical birth and death process representing an M/M/1 queuing system for cluster operations is used to model the discrete states of the processes. From a given state "i", the transitions into either states "$i + 1$" or "$i - 1$" are only possible by arrival or departure of requests from the system, respectively.

L in this case represents the total queue capacity beyond which no more arriving jobs are accepted into the CH. λ and μ illustrate birth and death transitions originating from state "i", respectively. Employing well-known queuing theory formulae, the steady-state probabilities (P_i) are computed using Eqs. (1.5) and (1.6). Once the state probabilities are obtained, desired performance measures can be computed.

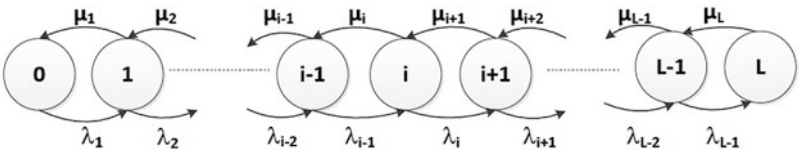

Fig. 1.3 Performance model for the proposed system

$$P_{i+1} = \frac{\lambda_i}{\mu_{i+1}} P_i \tag{1.5}$$

$$\sum_i P_i = 1 \tag{1.6}$$

The model capable of capturing various operative states of the CH for more realistic performability analysis is considered in studies such as [30, 31] and presented in Fig. 1.4. The state variable presented on vertical axis (j) in this figure is used to represent the number of requests in the system where the horizontal axis (i) represents system operative states. Apart from sleep state, three other operative states considered are active (R), node failure (F_M) and channel failure (F_C) states. Sleep state (S_{LP}) can only be entered at the end of service for the last job in the system with rate μ or when system is restored from either node or channel failures with rates η and θ, respectively.

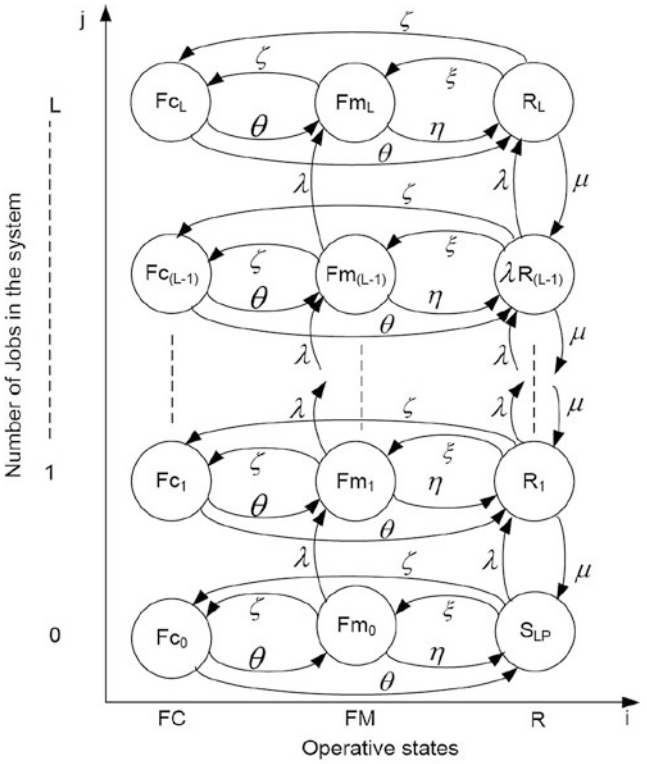

Fig. 1.4 Performability model

In active state (R), the duration a sensor node can stay in sleep state is assumed to be exponentially distributed with a mean of $1/\lambda$. Data packets can be received and serviced as illustrated in Fig. 1.4. The sensors may experience failures in either active or sleep states. The time between sensor node failures is assumed to be distributed exponentially with rate $1/\xi$. Channel failures on the other hand may occur either when operating in phase R or when the node is down. The time between channel failures is assumed to be exponentially distributed with rate $1/\zeta$. The duration taken to restore a channel after failure is assumed exponentially distributed with a mean of $1/\theta$. The system may reach either state F_{C_L}, F_{m_L} or R_L when the CH buffer is full. Packets can continue to arrive in phase F_M as long as there is some space in the buffer.

The random variables, $I(t)$ and $J(t)$, are employed to describe the system state at time t as explained in Sect. 1.2. The irreducible QBD process $X = [I(t), J(t)]; t \geq 0]$ on the lattice strip has the state space of $(0, 1, \ldots, \mathrm{N}) \times (0, 1, \ldots, \mathrm{L})$. In turn, the state probabilities of the resulting Markov process can be defined as $P_{i,j}$ with i and j representing the system operative state and the numbers of packets in the system including the one in service, respectively. The possible transitions in the process X are defined with transition rates λ, μ, η, ξ, θ and ζ.

In order to obtain the state probabilities represented with two-dimensional Markov processes, spectral expansion and matrix geometric methods can be employed. These approaches have been compared critically in [24]. I this section, for the steady-state analysis of the two-dimensional Markov process on the finite lattice strip, Spectral Expansion method is employed. This method has successfully been used to solve performance and dependability problems arising in computing and communication systems [10, 16, 36]. Details of spectral expansion method can be found in studies such as [9, 16, 24]. Once the steady-state probabilities are computed, the mean queue length (MQL) can be expressed as:

$$\mathrm{MQL} = \sum_{j=0}^{L} j \sum_{i=0}^{N} P_{i,j}; \quad i \leq 0; \quad j = 0, 1, 2, \ldots, L. \tag{1.7}$$

Considering that service is only possible when the system is in the active state, throughput (γ) can be expressed using Eq. (1.8).

$$\gamma = \mu \sum_{j=1}^{L} P_{i,j}; \quad i = 2; \quad j = 0, 1, 2, \ldots, L. \tag{1.8}$$

MQL and throughput can be employed to compute the system response time R_T using Eq. (1.9).

$$R_T = \mathrm{MQL}/\gamma \tag{1.9}$$

1.5 Performability Results and Discussions

For optimal CH operations, 25–35 nodes were used for the experiments. Please note that the maximum number of nodes in each cluster recommended by IEEE 802.15.4/ZigBee standard is 36 . The arrival rates ranging from 1 packet/s to 20 packets/h are possible for these systems [11, 32, 40]. The total arrival rate at the CH per hour can be expressed as the superposition of all the packet streams from all the members $\lambda_k = \sum_{n=1}^{N} \lambda$. The arrival rate at the CH is varied between 1 and 20 packets/h in this study. In order to achieve steady-state operation at full load, the CH requires at least a slightly higher service rate. Since the CH may also have its internal operations in addition to own observations, a service rate of $\mu_k = 300$ packets per hour is chosen in order to ensure that steady-state conditions are met during operations at full capacity. In order to identify optimum operation, queue capacities can be taken as $L = 10, 30, 50$ and 100.

It is possible to assume that all sensors use the CC2420 radio transceiver and are attached with a $2 \times AA$ batteries of 2.7–3.3 V capable of continuous operations for 3.25 days. In addition, good availability mechanisms like use of backup CHs and solar charging systems [26] are assumed, hence failures resulting from battery depletion are not considered. In the literature, failure rates of 0.001 per day, 0.0001 per year, 0.00002 every 5 years, and 0.00001 have been employed [27, 33]; however, for the numerical results presented, a failure rate of $\xi = 0.001$ per hour is assumed as well as a repair rate of $\eta = 0.5$ per hour.

Channel interferences in WSNs may result from many sources ranging from physical obstruction to radio frequency interference that may be caused by devices such as the ones using Bluetooth and sharing the same communication channel. Electrical interference is another well-known factor; however, the main contributor is environmental factors which depend heavily on the deployment location of the sensor networks. In order to incorporate the random nature of these interferences, channel failure and restoration rates are taken as $\zeta = 0.001$ and $\theta = 0.6$, respectively.

Results given in Tables 1.1 and 1.2 for MQL and response time are used together with Fig. 1.5a, b, respectively, to illustrate the difference between the results obtained for systems with failures and pure performance systems. It is clear from both of the tables that pure performance model results are overestimated. Greater differences are observed between the two models during relatively lower arrival rates. Noting that in most WSN environments event occurrences are characterised by low arrival rates, pure performance models cause serious overestimations. The term “Disc” is used to denote the discrepancies in Tables 1.1 and 1.2.

Table 1.1 Comparison of MQL results obtained from pure and integrated performance and availability system models

Arrival rate (λ)	Pure 25 sources	Integrated 25 sources	Disc (%)	Pure 30 sources	Integrated 30 sources	Disc (%)	Pure 35 sources	Integrated 35 sources	Disc (%)
1	0.09091	0.2273	60.0048	0.1111	0.2653	58.1187	0.132071	0.3013	56.1648
2	0.2	0.4027	50.3352	0.25	0.4686	46.6496	0.3043	0.5357	43.1869
3	0.3333	0.5702	41.541	0.42857	0.6793	36.91	0.53846	0.8005	32.734
4	0.5	0.7585	34.0804	0.6667	0.9384	28.9571	0.875	1.1581	24.445
5	0.71428	0.989	27.777	1	1.2885	22.3904	1.4	1.7022	17.753
6	1	1.2885	22.39	1.5	1.805	16.8975	2.333	2.6585	12.231
7	1.4	1.7022	17.7535	2.333	2.6585	12.231	4.453	4.8191	7.599
8	1.999	2.3177	13.71	3.9994	4.3569	8.205	12.4422	14.3647	13.38
9	3.0	3.34	10.148	8.7623	9.4234	7.015			
10	4.9953	5.3734	7.0359						

Table 1.2 Comparison of response time results obtained from pure and integrated performance and availability system models

Arrival rate (λ)	Pure 25 sources	Integrated 25 sources	Disc (%)	Pure 30 sources	Integrated 30 sources	Disc (%)	Pure 35 sources	Integrated 35 sources	Disc (%)
1	0.00363	0.0091	60.04	0.003704	0.0089	58.3853	0.00377	0.0086	56.121
2	0.004	0.0081	50.617	0.00416	0.0078	46.5812	0.004348	0.0077	43.5347
3	0.00444	0.0076	41.5204	0.00476	0.0076	37.343	0.0051	0.0076	32.523
4	0.005	0.0076	34.2105	0.00555	0.0078	28.775	0.00625	0.0083	24.6988
5	0.005714286	0.0079	27.66726944	0.006666667	0.0086	22.48062016	0.008	0.0098	18.36734694
6	0.006667	0.0086	22.4806	0.00833	0.0101	17.4917	0.0111	0.0127	12.511
7	0.008	0.0098	18.3673	0.0111	0.0127	12.511	0.01818	0.0197	7.74
8	0.01	0.0116	13.7931	0.01666	0.0182	8.438	0.04453	0.0515	13.527
9	0.0133	0.0149	10.515	0.03247	0.035	7.2288			
10	0.019982	0.0216	7.492						

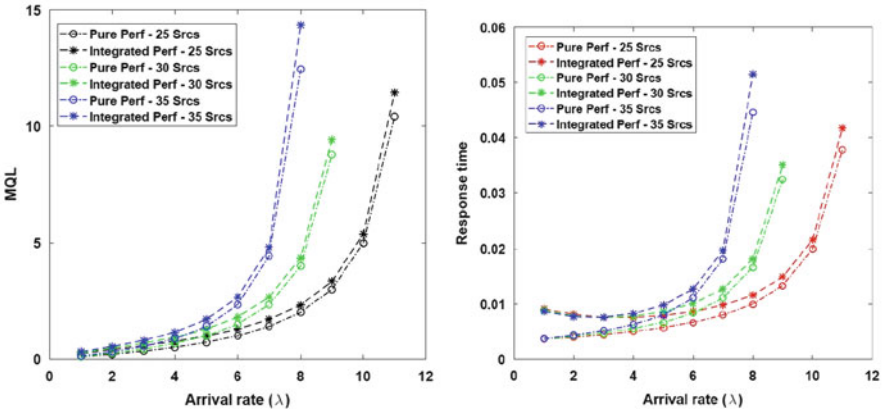

Fig. 1.5 Effects of variable buffer capacity on system performance. (**a**) Comparison of mean queue length (MQL). (**b**) Comparison of response time

1.6 Conclusion and Future Directions

In this chapter, performability modelling and solution approaches are discussed comparatively with pure performance and availability evaluation techniques. Analytical models are presented for IoT applications which employ clustering mechanisms for WSNs. The modelling approach considered for performability evaluation integrates WSN performance and availability/reliability in the presence of both node and channel failures. The models developed also consider limitations of sensor queue capacity and sleep/active operation dynamics. The results obtained using spectral expansion solution method clearly show the overestimations that may be caused by ignoring the fault-tolerant nature of enabling technologies which can be essential for IoT infrastructures.

References

1. A.P. Abidoye, I.C. Obagbuwa, Models for integrating wireless sensor networks into the internet of things. IET Wireless Sens. Syst. **7**(3), 65–72 (2017)
2. A. Al-Fuqaha, M. Guizani, M. Mohammadi, M. Aledhari, M. Ayyash, Internet of things: a survey on enabling technologies, protocols, and applications. IEEE Commun. Surv. Tutorials **17**(4), 2347–2376 (2015)
3. M.A. Ameen, S.M.R. Islam, K.S. Kwak, Energy saving mechanisms for MAC protocols in wireless sensor networks. Int. J. Distrib. Sens. Netw. **2010**, Article ID 163413 (2010)
4. G. Anastasi, M. Conti, M.D. Francesco, Extending the lifetime of wireless sensor networks through adaptive sleep. IEEE Trans. Ind. Inf. **5**(3), 351–365 (2009)
5. Q.M. Ashraf, M.H. Habaebi, Autonomic schemes for threat mitigation in internet of things. J. Netw. Comput. Appl. **49**, 112–127 (2015)

6. L. Atzori, A. Iera, G. Morabito, Understanding the internet of things: definition, potentials, and societal role of a fast evolving paradigm. Ad Hoc Netw. **56**, 122–140 (2017)
7. M.D. Beaudry, Performance-related reliability measures for computing systems. IEEE Trans. Comput. **6**, 540–547 (1978)
8. R. Chakka, Performance and reliability modelling of computing systems using spectral expansion. PhD, Newcastle University (1995)
9. R. Chakka, Spectral expansion solution for some finite capacity queues. Ann. Oper. Res. **79**, 27–44 (1998)
10. R. Chakka, E. Ever, O. Gemikonakli, Joint-state modeling for open queuing networks with breakdowns, repairs and finite buffers, in *15th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems, 2007. MASCOTS'07* (2007), pp. 260–266
11. W.H.R. Chan, P. Zhang, W. Zhang, I. Nevat, A.C. Valera, H. Tan, N. Gautam, Adaptive duty cycling in sensor networks via continuous time Markov chain modelling, in *2015 IEEE International Conference on Communications, ICC 2015*, London, 8–12 June 2015, pp. 6669–6674
12. C. Chiasserini, M. Garetto, An analytical model for wireless sensor networks with sleeping nodes. IEEE Trans. Mob. Comput. **5**(12), 1706–1718 (2006)
13. L. Columbus, Internet of things market to reach $267 B by 2020 (2017), https://www.forbes.com
14. S. El Kafhali, K. Salah, Efficient and dynamic scaling of fog nodes for IoT devices. J. Supercomput. **73**, 5261–5284 (2017)
15. E. Ever, Fault-tolerant two-stage open queuing systems with server failures at both stages. IEEE Commun. Lett. **18**(9), 1523–1526 (2014)
16. E. Ever, O. Gemikonakli, A. Koçyigit, E. Gemikonakli, A hybrid approach to minimize state space explosion problem for the solution of two stage tandem queues. J. Netw. Comput. Appl. **36**(2), 908–926 (2013)
17. E. Gelenbe, Product-form queueing networks with negative and positive customers. J. Appl. Probab. **28**(3), 656–663 (1991)
18. P. Heidelberger, K.S. Trivedi, Queueing network models for parallel processing with asynchronous tasks. IEEE Trans. Comput. **11**, 1099–1109 (1982)
19. A. Javed, H. Larijani, A. Ahmadinia, R. Emmanuel, M. Mannion, D. Gibson, Design and implementation of a cloud enabled random neural network-based decentralized smart controller with intelligent sensor nodes for HVAC. IEEE Internet Things J. **4**(2), 393–403 (2017)
20. R. Jurdak, A.G. Ruzzelli, G.M. O'Hare, Radio sleep mode optimization in wireless sensor networks. IEEE Trans. Mob. Comput. **9**(7), 955–968 (2010)
21. J. Li, Y.Q. Zhao, F.R. Yu, X. Huang, Queuing analysis of two-hop relay technology in LTE/LTE-A networks with unsaturated and asymmetric traffic. IEEE Internet Things J. **3**(3), 378–385 (2016)
22. A. Meola, What is the Internet of Things (IoT)? Business Insider (2016)
23. J.F. Meyer, On evaluating the performability of degradable computing systems. IEEE Trans. Comput. **8**, 720–731 (1980)
24. I. Mitrani, R. Chakka, Spectral expansion solution for a class of Markov models: application and comparison with the matrix-geometric method. Perform. Eval. **23**(3), 241–260 (1995)
25. Z. Mukter, H.W. Yand, Md.I. Shabiul, N. Amin, An integrated hybrid energy harvester for autonomous wireless sensor network nodes. Int. J. Photoenergy **2014**, 760534 (2014)
26. A. Munir, A. Gordon-Ross, Markov modeling of fault-tolerant wireless sensor networks, in *Proceedings of 20th International Conference on Computer Communications and Networks, ICCCN 2011*, Maui, HI, 31 July–4 August, pp. 1–6 2011
27. A. Munir, J. Antoon, A. Gordon-Ross, Modeling and analysis of fault detection and fault tolerance in wireless sensor networks. ACM Trans. Embed. Comput. Syst. **14**(1), 3:1–3:43 (2015)

28. M.F. Neuts, *Matrix-Geometric Solutions in Stochastic Models: An Algorithmic Approach* (Courier Corporation, New York, 1981)
29. A.J. Odey, D. Li, Low power transceiver design parameters for wireless sensor networks. Wirel. Sens. Netw. **4**, 243–249 (2012)
30. F.A. Omondi, E. Ever, P. Shah, O. Gemikonakli, Modelling wireless sensor networks for performability evaluation, in *International Conference on Ad-Hoc Networks and Wireless* (Springer, New York, 2013), pp. 172–184
31. F.A. Omondi, P. Shah, O. Gemikonakli, E. Ever, An analytical model for bounded WSNs with unreliable cluster heads and links, in *2015 IEEE 40th Conference on Local Computer Networks (LCN)* (IEEE, Clearwater Beach, 2015), pp. 201–204
32. T. Qiu, L. Feng, F. Xia, G. Wu, Y. Zhou, A packet buffer evaluation method exploiting queueing theory for wireless sensor networks. Comput. Sci. Inf. Syst. **8**, 1028–1049 (2011)
33. I. Silva, L.A. Guedes, P. Portugal, F. Vasques, Reliability and availability evaluation of wireless sensor networks for industrial applications. Sensors **12**(1), 806–838 (2012)
34. K.S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications* (Wiley, Chichester, 2008)
35. K.S. Trivedi, M. Malhotra, Reliability and performability techniques and tools: a survey, in *MMB* (1993), pp. 27–48
36. K.S. Trivedi, X. Ma, S. Dharmaraja, Performability modelling of wireless communication systems. Int. J. Commun. Syst. **16**(6), 561–577 (2003)
37. C. Wang, L. Xing, V. Vokkarane, Y.L. Sun, Reliability and lifetime modeling of wireless sensor nodes. Microelectron. Reliab. **54**(1), 160–166 (2014)
38. O. Yang, W. Heinzelman, An adaptive sensor sleeping solution based on sleeping multipath routing and duty-cycled MAC protocols. ACM Trans. Sensor Netw. **10**(1), 10 (2013)
39. Y. Zhang, W.W. Li, Modeling and energy consumption evaluation of a stochastic wireless sensor network. EURASIP J. Wirel. Commun. Netw. **2012**, 282 (2012)
40. H. Zhou, D. Luo, Y. Gao, D.C. Zuo, Modeling of node energy consumption for wireless sensor networks. Wirel. Sensor Netw. **3**(1), 18–23 (2011)

Chapter 2
Practical Performability Assessment for ZigBee-Based Sensors in the IoT Era

Umit D. Ulusar, Gurkan Celik, Erdinc Turk, Fadi Al-Turjman, and Halil Guvenc

2.1 Introduction

The IoT is a recent technology paradigm envisioned as a global network of sensors and devices capable of interacting with each other. With the rapid advances in technology, IoT is providing great opportunities for novel applications that promise to improve the quality of our lives. Sensor networks are widely used in daily life in different areas such as smart cities, home and building automation, e-Health, security, localization, and many other areas [1].

With the development of IoT, LBS has gained significant attention and inventing efficient positioning mechanisms for sensor nodes has become an active research area. LBS can be used in a variety of contexts and may contain services to locate a person or object, such as the nearest supermarket or a child. Use of LBS is not only limited to consumer use but also provides new services for companies such as location-based advertising, employee management, and location-based games [2].

On the data transmission and networking side of IoT, for smart data transmission, it is necessary to choose the most appropriate communication technique by considering the position, battery status, and the signal strength of the nodes. By reducing energy consumption, the nodes and components of the network can operate for a longer period.

U. D. Ulusar (✉) · E. Turk
Computer Engineering Department, Akdeniz University, Antalya, Turkey
e-mail: umitulusar@akdeniz.edu.tr

G. Celik · H. Guvenc
Electrical-Electronic Engineering Department, Institute of Science and Technology, Akdeniz University, Antalya, Turkey

F. Al-Turjman
Computer Engineering Department, Antalya Bilim University, Antalya, Turkey

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_2

Table 2.1 Abbreviations

Abbreviation	Description
IoT	Internet of things
LBS	Location-based service
RSSI	Received signal strength indicator
LQI	Link quality indicator
ToF	Time of flight
FFD	Full function device
RFD	Reduced function device
PL	Path loss
WPAN	Wireless personal area network
PAN	Personal area network
T_{rt}	Round trip time
T_{total}	Total time
T_{ta}	Turnaround time
ACK	Acknowledgement message
P_{rx}	Power of received signal
P_{tx}	Power of transmitted signal
P_{ref}	Reference power
G_{tx}	Gain of transmitter antenna
G_{rx}	Gain of receiver antenna

Received signal strength and quality of link can also affect the performance of various systems such as bio-inspired networking in the Internet of nano-things applications [3], integrated vehicular-IoT [3], context-awareness in smart systems [4], context-sensitive access in industrial IoTs [5], and information-centric sensor networks for cognitive IoT [6] and so on.

In this study, we have conducted empirical studies on positioning accuracy and have compared RSSI and ToF values obtained at varying distances with their theoretical values. Findings of this study can be used to develop more realistic data routing algorithms and improve node positioning. Table 2.1 shows the list of abbreviations used in this chapter.

2.2 Background

ZigBee is an umbrella term that is used to define specifications for a bunch of communication protocols based on IEEE 802.15.4. Typical applications of ZigBee are PANs with tiny low-power digital radios. ZigBee-based devices are also widely used in home and building automation, consumer electronics, and many other areas. With ZigBee, it is possible to create wireless mesh networks that can transfer data between long distances. ZigBee's data rate per channel is 250 kbps at 2.4 GHz, 40 kbps at 915 MHz, 20 kbps at 868 MHz. A ZigBee-based mesh network offers scalability, stability, and tolerance for connection failures. There are typically two

types of devices: FFDs and RFDs. FFDs can be used in three modes such as PAN coordinator, sub-coordinator, or end-device. RFDs can run in only end-device mode. In addition, an FFD can communicate with RFDs or other FFDs, while an RFD can communicate only with an FFD [7].

There are some studies in the literature on positioning using ZigBee-based technologies. Bedford and Kennedy studied underground navigation in mines for emergency situations. In order to increase accuracy of positioning, they used ToF measurements obtained for multiple channels and combined using filtered averaging algorithm [7]. Chen et al. have compared indoor and outdoor position estimation techniques and proposed a two-step scheme: partition, in which the target region is split into small grids, and localization refinement [2].

Some of the techniques that are used for sensor node localization are ToF and RSSI. ToF specifies the amount of time a signal takes to propagate between transmitter and receiver nodes. Because the signal propagation speed is constant and known as speed of light (approx. 300,000 km/s), the travel time of a signal can be used to directly calculate distance as shown in Eq. (2.1).

$$d = \text{ToF} * c \,[\text{m}] \tag{2.1}$$

ToF calculation needs time synchronization between two nodes to measure the delay between the transmission of a signal and its receipt. Measurements are typically made with fast sequential communication and averages of the values for the same distance are calculated to reduce the instantaneous variation of the results. Since the distance is measured using the time difference between send and receive, the clock frequencies of the processing units at both ends and synchronization may create measurement errors. In order to eliminate this, the measurements are performed in forward and backward directions.

Another way to estimate the distance between nodes is to measure the amount of signal attenuation during transmission. PL specifies the amount of attenuation and besides the physical phenomena, PL can also happen due to numerous factors such as obstacles between antennas, communication frequency, and antenna gains [8]. There are different path loss models. Propagation models such as IMT 2000, Herring, and Hata Okumura provide theoretical values for indoor to indoor, outdoor to indoor, and vehicular environments [9].

In this study, we used a free space propagation path loss model. PL at distance d is given by Eq. (2.2). By using this equation, Fig. 2.1 shows PL versus distance.

$$PL(d) = 20 * \log\left(4 * \pi * d/\lambda\right) \quad [\text{dB}] \tag{2.2}$$

When the signal power at the transmitter is known, received signal power at any distance d can be calculate by using Friis' transmission Eq. (2.3) [10].

$$Prx = Ptx * Gtx * Grx * (\lambda/4\pi d)^2 \quad [\text{mW}] \tag{2.3}$$

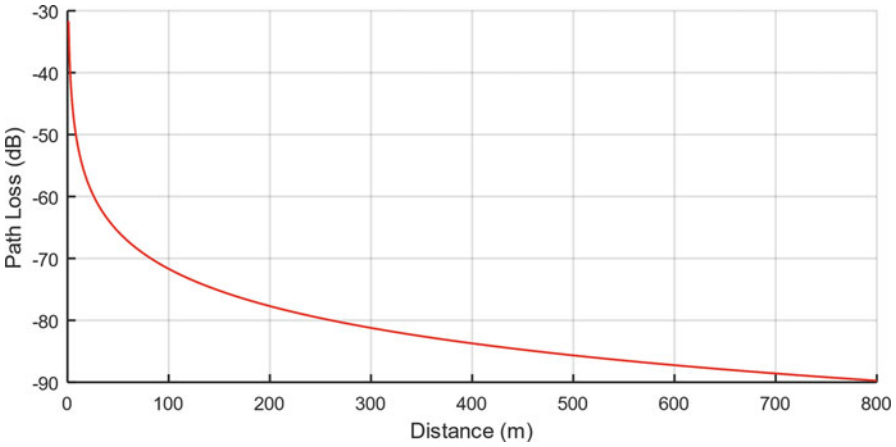

Fig. 2.1 PL versus distance between the transmitter and the receiver

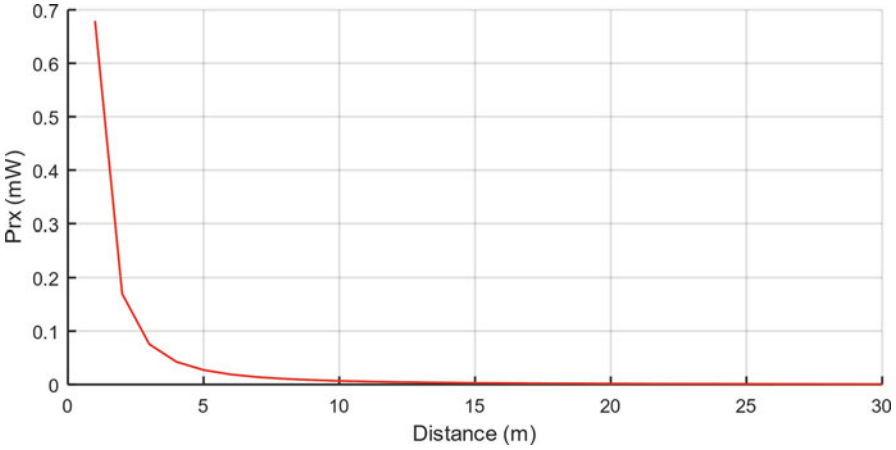

Fig. 2.2 Received power (P_{rx}) versus distance between the transmitter and the receiver

where n is signal propagation constant, also named propagation exponent, P_{tx} is transmission power of sender, P_{rx} is remaining power of wave at receiver, G_{tx} is gain of transmitter, G_{rx} is gain of receiver, and λ is wavelength.

For embedded devices, the transmitted and received signal strengths are close to 0 (mW) (Fig. 2.2), and typically the logarithm of the ratio of power of the signal to a specific reference power (P_{ref}) measure called RSSI is used (Eq. (2.4)). Typically, the reference power is defined as 1 mW.

$$RSSI = 10 * \log\left(Prx/Pref\right) \text{ [dBm]} \tag{2.4}$$

Fig. 2.3 illustrates the relation between RSSI and the received signal power [7].

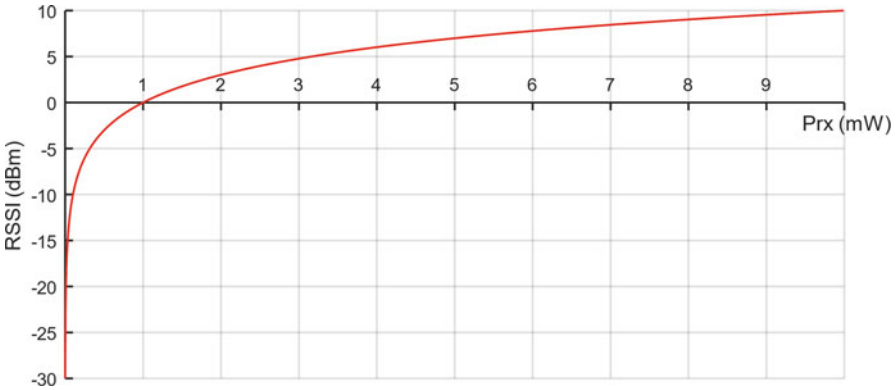

Fig. 2.3 RSSI versus received signal power (P_{rx})

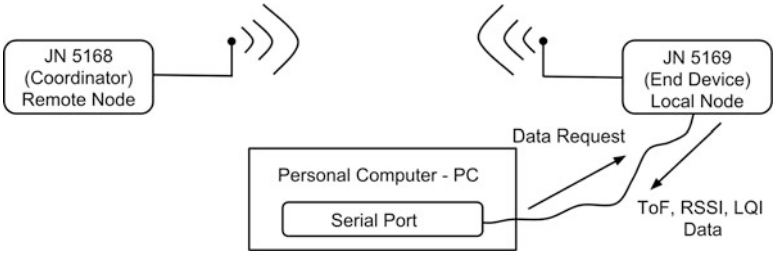

Fig. 2.4 Communication block diagram between transmitter and receiver nodes

LQI is another parameter that is used for positioning. It characterizes the quality of received packages, typically a measure between 0 and 255 where the quality of the received signal increases as the number increases. Different from RSSI, LQI measures transmission quality/error rather than signal power. It is possible to have an error-free and lossless transmission but typically, if the LQI is low, RSSI is also low. Similar to RSSI, increase in distance adversely affects LQI.

2.3 System Architecture

A point-to-point network is built with ZigBee devices, JN5168 and JN5169 (NXP Semiconductors, the Netherlands). Jennic ZigBee modules consist of powerful 32-bit RISC CPUs, 4/6 inputs of 10-bit ADC, battery and temperature sensors, and serial interfaces (I2C, SPI). Receiver sensitivity is −96 dBm and transmit power is 2.5 dBm [11].

Communication is initiated from the local node by sending a command to the remote node. When the required read values for the ToF data are complete, the local node receives the timing data from the remote node. Overall design of the system is shown in Fig. 2.4.

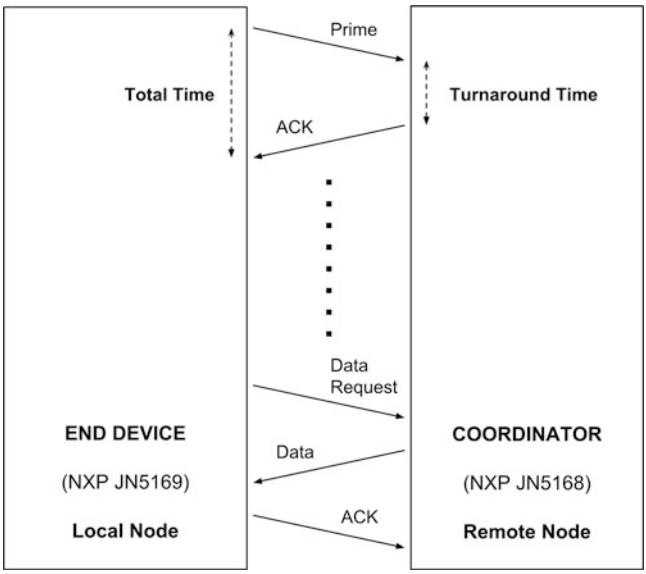

Fig. 2.5 Packet transfers between the local and remote nodes during a burst of ToF measurements

To perform the ToF measurement, the remote node sends a prime message to the end device as shown in Fig. 2.5. The end device then terminates the communication by sending an ACK message indicating that the end device has received the message. The local node measures the time difference from sent of the prime message until the receipt of the ACK message and records this value as the total time T_{total}. The remote node also measures the time difference between the message time for the data request and the transmission time of the data and stores this value as the turnaround time T_{ta}. The total duration round trip time T_{rt} is the time that the message travels twice the distance between the receiver and the transmitter. Assuming that the bilateral transmission takes approximately equal time, the transmission period of the data (ToF) is equal to half of the round-trip time as shown in Eq. (2.5) [7].

$$ToF = \frac{T_{\text{rt}}}{2} = \frac{T_{\text{total}} - T_{\text{ta}}}{2} \tag{2.5}$$

In this study, we used LQI percentage in order to calculate probability of successful communication using Eq. (2.6).

$$\text{LQI}_{\text{percent}} = 100 * \frac{(\text{LQI}_{\text{value}} - \text{LQI}_{\text{min}})}{(\text{LQI}_{\text{max}} - \text{LQI}_{\text{min}})} \tag{2.6}$$

Experiments were performed in the Akdeniz University campus. JN5169 was programmed as end-device, and JN5168 was in coordinator mode. The transmitter

Fig. 2.6 ZigBee devices that are used in experiments. The left shows receiver node and the right shows transmitter node

Fig. 2.7 Experiments were performed in Akdeniz university campus area. Blue dot shows the position of transmitter node and black dots show positions of receiver node

and receiver devices are shown in Fig. 2.6 and the test area is shown in Fig. 2.7. Position of the receiving node was kept constant and that of the transmitting node was mobile.

2.4 Results

In this study, in order to compare the ToF and RSSI, we performed 500 measurements. The measured and calculated parameters are ToF, standard deviation of ToF, TOF-based distance calculation, Local RSSI, Remote RSSI, Local LQI, and Remote LQI (Tables 2.2 and 2.3). Each packet had 10 messages (0–9).

Table 2.2 Data packet at 5-m distance

No	ToF (ps)	Local RSSI	Local LQI	Remote RSSI	Remote LQI
0	13,456	−60	242	−58	246
1	18,300	−60	242	−58	240
2	23,456	−61	249	−58	249
3	15,400	−60	247	−59	239
4	17,262	−60	254	−56	245
5	23,168	−60	245	−56	245
6	16,481	−59	247	−57	246
7	20,762	−60	240	−56	244
8	11,512	−60	252	−56	242
9	16,862	−61	252	−58	254

Table 2.3 Data packet at 100-m distance

No	ToF (ps)	Local RSSI	Local LQI	Remote RSSI	Remote LQI
0	318,669	−88	221	−86	193
1	317,619	−89	217	−87	208
2	317,712	−89	214	−87	223
3	329,462	−88	215	−88	214
4	320,681	−89	222	−87	234
5	321,281	−89	223	−88	223
6	318,812	−89	236	−88	222
7	333,712	−88	210	−87	231
8	321,524	−89	235	−87	226
9	326,337	−87	226	−86	235

For 5 m distance, mean ToF was 17,665 ps, standard deviation of ToF values was 3698 ps, calculated distance using ToF was 5.2 m, and calculated distance using RSSI was 17 m.

For 100 m, mean ToF was 322,580 ps, standard deviation of ToF values was 5192 ps, calculated distance using ToF was 96 m, and calculated distance using RSSI was 191 m.

Figure 2.8 shows the ToF and RSSI-based distance estimations with reference distance. There is a significant correlation between ToF-based location estimation.

Figure 2.9 shows the $LQI_{percent}$ with respect to distance. The $LQI_{percent}$ value decays almost linearly (0.3% per m) up to 120 m (80%) and shows a rapid decay after that (1.1% per m). These findings are consistent with our previous results [10, 12].

Figure 2.10 shows the empirical and theoretical RSSI values. It should be noted that received signals can be affected by environmental factors such as trees, buildings, other radio waves, and weather conditions.

In this study, we have used one-cell lithium polymer battery with 350 mAh capacity as energy source for nodes. Figure 2.11 illustrates the battery consumption

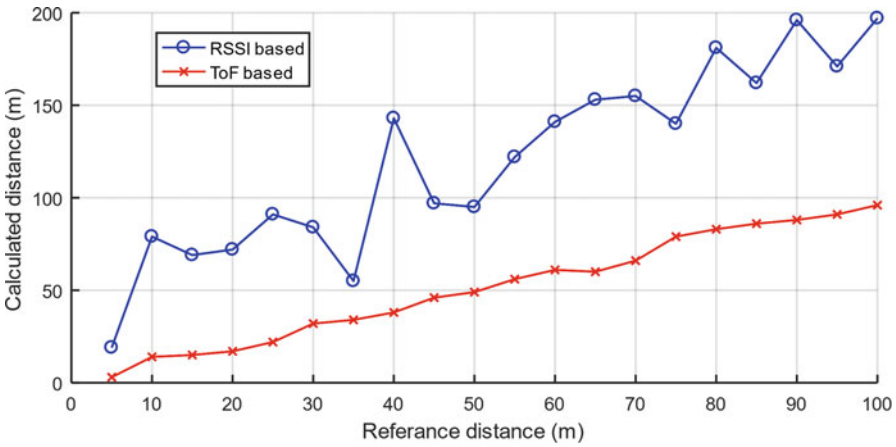

Fig. 2.8 Reference distance versus calculated distances

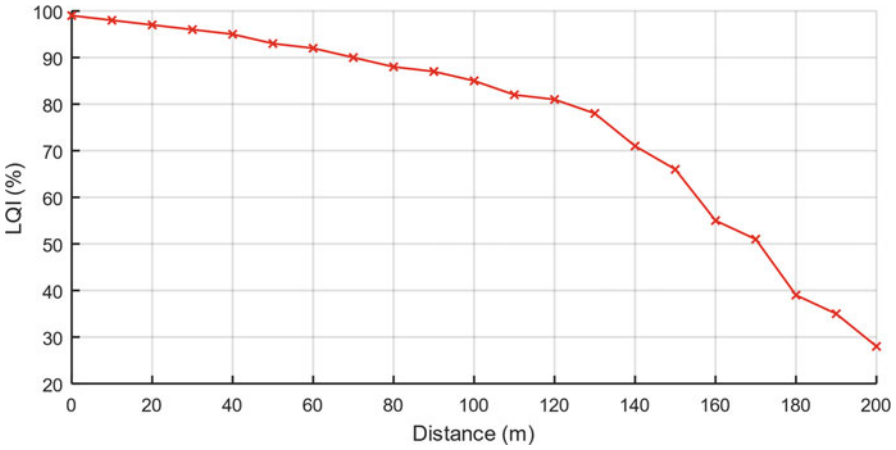

Fig. 2.9 LQI percentage versus distance between transmitter and receiver nodes

over time. One hundred percent represents 4.2 V and 0% represents 3.7 V. The battery voltage was sampled in every 30 min by using a voltmeter.

2.5 Conclusion

Positioning sensor nodes in an efficient way is desirable for energy-efficient communication and providing location-based services. In this study, information of two different signals for accurate positioning of sensor nodes was compared. Results indicate that ToF-based measurements are more accurate than RSSI-based techniques.

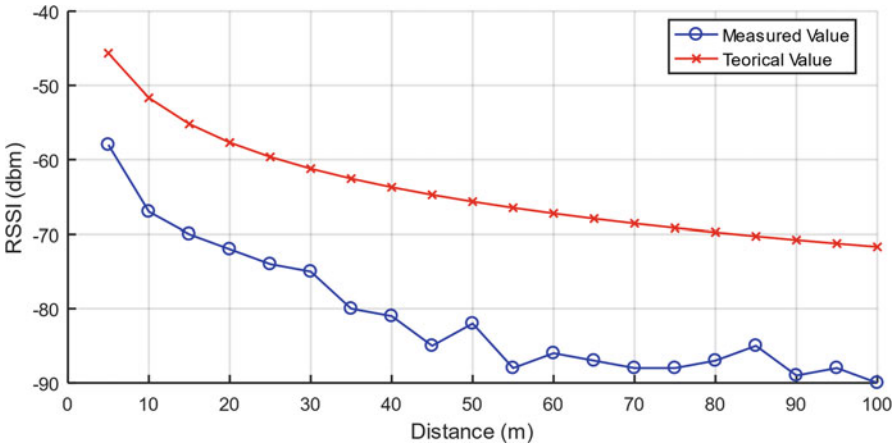

Fig. 2.10 Received Signal Strength Indicator (RSSI) versus distance between the transmitter and the receiver

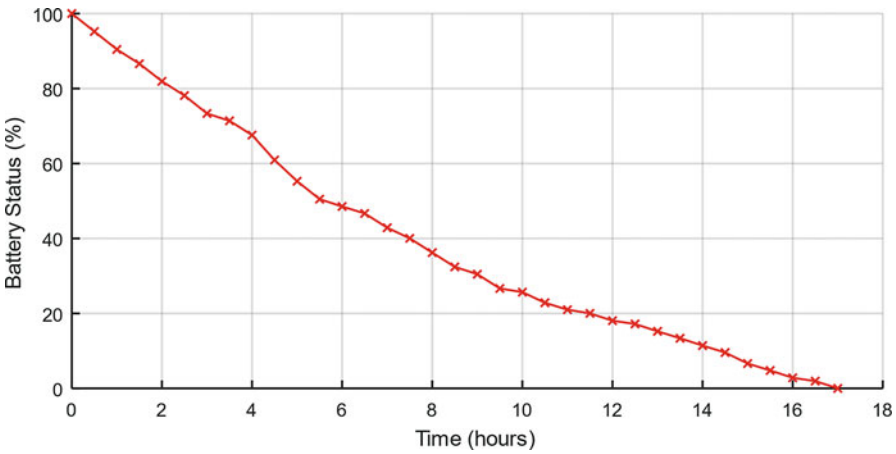

Fig. 2.11 Battery status during the operation of modules

Findings of this study can be useful for a wide range of applications that make sensor networks an integral part of our daily life. For real-world applications, efficient realization of sensor networks requires a careful balance between different factors such as fault tolerance, scalability, hardware limitations, network topology changes, and environment and power consumption conditions.

Acknowledgments This work is supported by Tubitak project 115E198.

References

1. U. D. Ulusar, F. Al-Turjman, G. Celik, "An overview of Internet of things and wireless communications," in *2017 International Conference on Computer Science and Engineering (UBMK)*, 2017, pp. 506–509
2. Z. Chen, F. Xia, T. Huang, F. Bu, H. Wang, A localization method for the internet of things. J. Supercomput. **63**(3), 657–674 (2013)
3. F. Al-Turjman, QoS—aware data delivery framework for safety-inspired multimedia in integrated vehicular-IoT. Comput. Commun. **121**, 33–43 (2018)
4. F. Al-Turjman, 5G-enabled devices and smart-spaces in social-IoT: an overview. Futur. Gener. Comput. Syst. (2017). https://doi.org/10.1016/j.future.2017.11.035
5. F. Al-Turjman, S. Alturjman, Context-sensitive access in industrial internet of things (IIoT) healthcare applications. IEEE Trans. Ind. Inform. **PP**(99), 1 (2018)
6. F.M. Al-Turjman, Information-centric sensor networks for cognitive IoT: an overview. Ann. Telecommun. **72**(1–2), 3–18 (2017)
7. M.D. Bedford, G.A. Kennedy, Evaluation of ZigBee (IEEE 802.15.4) time-of-flight-based distance measurement for application in emergency underground navigation. IEEE Trans. Antennas Propag. **60**(5), 2502–2510 (2012)
8. U. D. Ulusar, G. Celik, F. Al-Turjman, Wireless Communication Aspects in the Internet of Things: An Overview, in *2017 IEEE 42nd Conference on Local Computer Networks Workshops (LCN Workshops)*, 2017, pp. 165–169
9. C. Phillips, D. Sicker, D. Grunwald, A survey of wireless path loss prediction and coverage mapping methods. IEEE Commun. Surv. Tutor. **15**(1), 255–270 (2013)
10. F. Al-Turjman, Optimized hexagon-based deployment for large-scale ubiquitous sensor networks. J. Netw. Syst. Manag. **26**(2), 255–283 (2018)
11. Data Sheet: JN516x NXP-IEEE802.15.4 Wireless Microcontroller. NXP Laboratories UK, 2013
12. F. Al-Turjman, Cognitive-Node Architecture and a Deployment Strategy for the Future WSNs, *Mob. Netw. Appl.*, May 2017, pp. 1–19

Chapter 3
Evaluation of Simulation Approaches and Need for MDE in Energy Efficiency, Performance and Availability Assessment of IoT

Krishna Doddapaneni and Yoney Kirsal Ever

3.1 Introduction

Considering rapidly evolving technology enhancements for Internet, data networks and communications, within the last decade, significant advancements have converged with various applications and services. The Internet of Things (IoT) became a novel paradigm for this purpose. Among the emerging enabling technologies, wireless sensor networks (WSNs), intelligent sensing, remote sensing and low-energy wireless communications, cloud computing (CC) has attracted the interest of computer scientists, engineers and researchers [28]. These technologies are involved in other very important domains such as health monitoring, smart environments, smart cities and various pervasive systems as well [28].

WSNs, with a wide range of applications are rapidly becoming an integral part of our lives. Over the last 15 years, WSNs have appeared as one of the most prominent enabling technologies, which combines automated sensing, embedded computing and wireless capabilities into tiny devices, bringing promises of understanding and instrumenting nature at scales that were unimaginable before [2, 12]. Recently, considerable amount of research efforts have enabled the actual implementation and deployment of sensor networks tailored to the unique requirements of certain sensing and monitoring applications. The applications of sensor networks are diverse, ranging from habitat monitoring to surveillance and physical intrusion detection and can be categorised into environment, health, military, home, disaster

K. Doddapaneni
Altiux Innovations, Mountain View, CA, USA
e-mail: krishna.c@altiux.com

Y. K. Ever (✉)
Software Engineering Department, Faculty of Engineering, Near East University, Nicosia, Mersin, Turkey
e-mail: yoneykirsal.ever@neu.edu.tr

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_3

relief, space exploration and other commercial areas. The flexibility, fault tolerance, low cost, rapid deployment characteristics and high sensing fidelity of sensor networks create many new and exciting applications in the field of remote sensing. WSN applications and communication protocols are tailored mainly to provide higher energy efficiency, as sensor nodes carry limited power sources. Energy efficiency is crucial because of the scale and application environments in which sensors are deployed [1, 24].

IoT architecture can be implemented as either Internet centric or object centric. The former aims at provisioning services within the Internet, where data are contributed by objects and vendors who deterministically deploy these objects, whereas the latter aims at provisioning services via network of smart objects. Scalability and cost efficiency of IoT services can be achieved by the integration of cloud computing into the IoT architecture, i.e. cloud-centric IoT [20]. In a cloud-centric IoT framework, sensors provide their sensed data to a storage cloud as a service, which then undergoes data analytics and data mining tools for information retrieval and knowledge discovery [20].

WSNs are composed of base stations and numerous low-cost mobility nodes which have restricted resources, in terms of communication, storage and computation facilities. For more than 15 years, especially in last decade, different approaches have been proposed and designed for considering the collaborative nature of WSNs. Since each mobility node has its own sensing unit, data-processing unit, a module for short-range communication and a power-supply unit, the energy limitations, and potential solutions for these restrictions are investigated heavily in the existing literature [7]. Various improvements are introduced from physical layer to energy-efficient medium access control procedures, and a wealth of studies are performed for effective duty cycles as well as routing and clustering algorithms [9].

The design of WSNs requires ample knowledge of a wide variety of research fields including wireless communication, networking, embedded systems, digital signal processing and software engineering [9]. The design choices at various layers significantly impact the operation and resource efficiency of sensor nodes and networks.

For all these new approaches and novel techniques, the performance evaluation in terms of energy efficiency, coverage, performance and availability is essential [9, 10].

Performance modelling and evaluation should consider metrics for WSNs, such as system lifetime and energy efficiency, and the introduction of new traffic attributes, which enables to evaluate WSNs in a better way.

A sensor network is a special type of network which generally consists of a data acquisition system and a data distribution system. The unique characteristics of WSNs in terms of data collection and energy constraints separate them from other communication networks. In Fig. 3.1, we show the most common techniques for performance evaluation that are analytical modelling, simulation and benchmarking. The existing studies consider benchmarking in the form of test beds and measurements for real deployment. The energy constraints of WSNs limit their processing capabilities and communication. Therefore, using one of these

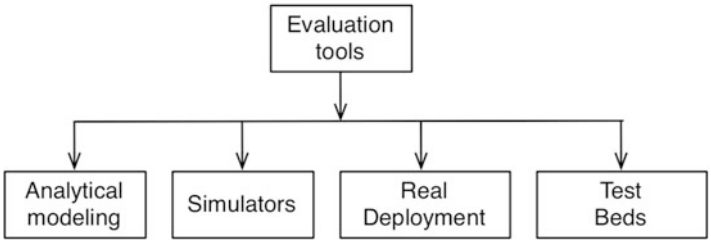

Fig. 3.1 Performance evaluation methods

performance evaluation methods, analysis of deployment and management of such complex systems is a challenging task [2].

Due to inherent complexity and diverse nature of WSNs (dynamic topology, wireless channel characteristics, mobility, density of the nodes, etc.), analytical methods may become inappropriate as they require certain simplifications to model and predict the performance of the system. The simplifications may lead to inaccurate results in case of unrealistic assumptions [7, 22]. Experimental studies such as [17, 35, 36, 47] are not always practical for evaluation of systems with different architectures and under various conditions, mainly because of the difficulties in deployment of real systems. Potential difficulties associated may be deploying tens or hundreds of sensor nodes in the physical environment, program the nodes and monitor their behaviour, the high costs involved in obtaining the instrumentation and other aspects such as fault tolerance, and scalability. It is well known that when it comes to benchmarking, the results in many cases cannot be extrapolated to suit the changes in the system or environment. Hence, testing and performance evaluation of WSNs through analytical modelling, real deployment and test beds can become complex, inaccurate, time consuming and/or costly [10].

Simulation is currently the most widely adopted method for analysing WSNs. Simulation studies provide quicker evaluation, optimisation and modification of the proposed algorithms and protocols at design, development and implementation stages. A number of simulation tools are available with different features, models, architectures and characteristics for performance evaluation in WSNs [10]. Simulation studies are very flexible providing fairly accurate and acceptable results; however, for sufficient accuracy, simulations require relatively high computational times.

In this paper, simulation tools are investigated and analysed in detail to specify an approach which considers collaborative nature of WSNs, since scalability and cost efficiency of IoT services are directly related with optimum use of information provider sensor nodes.

The rest of the paper is organised as follows: Sect. 3.2 considers related research works done for various types of simulators. In Sect. 3.3, methodology is presented. Section 3.4 provides results and discussions. In Sect. 3.5, conclusion and future studies are presented.

3.2 Related Works

The emergence of WSN paradigm has triggered extensive research, with emphasis on potential applications that can be realised using WSNs. Most of the time, the behaviour of the sensor network highly depends on the applications, within a specific environment. Sensor networks are a powerful combination of distributed sensing, computing and communications. Though they pose formidable challenges due to their peculiarities, they lend themselves to various countless applications, virtually in all fields of science and technology and hence making their way to the forefront of the scientific community [3, 39].

Although sensor network research is initially driven by military applications such as battle field surveillance and tracking, and high-end applications such as radiation and nuclear threat detection, sensor networks now are widely deployed in diverse applications including home automation, environmental monitoring, microsurgery, robotics, support for logistics, agriculture, etc. [3, 39].

Energy consumption of nodes is a crucial factor that constrains the networks lifetime for WSNs. The main concern in the existing architectural and optimisation studies is to prolong the network lifetime. The lifetime of the sensor nodes is affected by various components such as the microprocessor, the sensing module and the wireless transmitter/receiver. The existing works mainly consider these components to decide on best deployment, topology, protocols and so on. Recent studies have also considered the monitoring and evaluation of the path loss caused by environmental factors. Path loss is always considered in isolation from the higher layers such as application and network. It is necessary to combine path loss computations used in physical layer, with information from upper layers such as application layer for a more realistic evaluation. Simulation is an inevitable methodology for specification, design and analysis of computer and communication networks. It is extensively used at all levels ranging from hardware to network. However, to obtain valid results that properly predict the behaviour of a real system, all relevant effects must be captured in the simulation model. This task has become very challenging today, since one can rarely model and simulate the different levels in isolation [9]. Further, systems have become extremely complex and so have the simulation models. To get relevant statistical results and to cover critical corner cases, the simulated time (i.e. the time elapsed in the simulated system) has to be sufficiently long. To avoid excessive simulation time even on high-performance computers, modelling not only has to be proper but also efficient. In the study of [9], a simulation-based case study is presented that uses path loss model and application layer information in order to predict the network lifetime, and also physical environment is considered as well [9]. This work showed that when path loss is introduced, increasing the transmission power is needed to reduce the amount of packets lost. The authors presented a trade-off between the residual energy and the successful transmission rate when more realistic settings are employed for simulation [9]. It is proved that this is a challenging task to optimise the transmission power of WSNs, in the presence of path loss. Because although increasing the

transmission power reduces the residual energy, it also reduces the number of retransmissions required [9].

Considering the existing studies, simulation tools for WSNs can be classified based on the level of complexity into three main categories: instruction-, algorithm- and packet-level simulators. A detailed taxonomy on WSN simulation tools is also presented in [11, 23]. The unique features of simulation tools in various categories are also presented in detail, along with the features of classical simulators for WSNs.

3.2.1 Instruction-Level Simulators

Instruction-level simulators are often regarded as emulators. They model the CPU execution at the level of instructions or even cycles. TOSSIM [25], Atemu [38] and Avrora [50] are well-known emulators. TOSSIM is the most commonly used emulator. However, compared to other emulators, it is not the most precise one. TOSSIM is a platform specific simulator (a TinyOS mote simulator) which can compile any code written for TinyOS to an executable file. TinyViz is the basic GUI for TOSSIM which can visualise and interact with the running simulations. TOSSIM is specific for TinyOS applications on Mica mote sensors and does not include power models. Avrora is a java-based emulator used for programs specifically written for AVR microcontrollers produced by Amtel and the Mica2 sensor modes. Atemu provides low-level emulation of the operation of individual sensor nodes. A unique feature of Atemu is its ability to simulate a heterogeneous sensor network. It is scalable and its high fidelity platform is used as a pre-deployment tool for sensor networks.

3.2.2 Algorithm-Level Simulators

Shawn [46], AlgoSensim [14] and Sinalgo [48] are well-known algorithm-level simulators with emphasis on the logic, data structure and presentation of the algorithms. They rely on some form of graphical data structure to demonstrate the communication between the nodes. Shawn is a very powerful tool in simulating large-scale networks with an abstract point of view. It supports distributed protocols and generic high-level algorithms. AlgoSensim focuses on network-specific analysis of algorithms like localisation, distributed routing and flooding. AlgoSensim mainly facilitates the implementation and quality analysis of new algorithms. Sinalgo focuses on the verification of network algorithms and abstracts from the underlying layers. It also offers a message passing view of the network. Sinalgo can be employed for quick prototyping and verification in freely customisable network settings.

3.2.3 Packet-Level Simulators

OPNET, Qualnet, NS-2 and GloMoSim are some of the most commonly used packet-level simulators. They implement the data link and physical layers in the OSI network layers. Hence, radio models, 802.11b or newer MAC protocols, fading, collisions, noise and wave diffractions are commonly implemented [9]. Network simulator (NS) is a discrete event simulator written in combination of C++ and OTcl. OTcl is an object-oriented scripting language, developed mainly for networking research. It provides extensive support for simulation of TCP, multicast protocols and routing for wired and wireless networks. With protocol implementations being widely produced and developed, the extensibility of NS-2 has been a major contributor to its success. It has an object-oriented design which allows for easy creation of new protocols. The key features for WSNs include battery models, hybrid simulation support, sensor channels, scenario generation tools and a visualisation tool [19]. Scalability, lack of application model and the lack of customisation are few limitations of NS-2 along with lacking an application model [7]. OPNET [2011, 201] [34] and Qualnet [2011, 201] [41] are commercialised network simulator software with powerful standard modules and they provide good simulation environment. OPNET is an excellent choice to simulate Zigbee-based networks with the implementation of Zigbee protocol and IEEE 802.15.4 MAC protocol. However, performance measures related with energy are not available in OPNET, which is a major setback, as energy is a very significant parameter for performance evaluation. Qualnet performs well in simulating large-scale sensor networks due to its scalability in wireless simulation, but OPNET simulation requires a long time when the number of sensors considered is large [9].

The above-mentioned simulators use rather simple radio/channel models [21]. Also, the simulators are still platform specific and moderately scalable, making them unsuitable for protocol/algorithm design and testing. The major power consumption of the node is based on the time the radio is on, either transmitting, receiving or listening, and how long the radio stays in each of the states. Hence, it is also of significant importance to consider the energy consumed for listening as well, for performance evaluation. Furthermore, the environmental details, especially the effects of path loss, and the effects of collisions have not been considered in any of the given simulation packages.

In order to resolve the above-discussed issues, Doddapaneni et al. [10] offered an approach, that is implemented in a tool called PlaceLife. PlaceLife takes advantage of the three modelling views in order to provide an estimate of the WSN lifetime. All modelling views are analysed, combined and translated into low-level simulation scripts that can be executed to estimate the WSN lifetime. In the same study, Doddapaneni et al. [10], Castalia [6] is used as a simulation tool. It is stated that Castalia is a WSN simulator used for initial testing of protocols and/or algorithms with a realistic node behaviour, wireless channel and radio models. Since Castalia is highly tunable and can simulate a wide range of platforms, it is used to evaluate different platform characteristics. Because of the unpredictability of the

wireless channel, energy spent in transmission/receiving packets and performance degradation experienced by duty cycles, collisions are usually overlooked by simple simulators. However, these details are well established in Castalia. This work emphasises that while Castalia provides a good low-level simulation platform; it does not provide any means to specify the application behaviour, the environment and the path loss models. The application behaviour is needed to derive application-level simulation parameters. The environment and the path loss models allow the calculation of the path loss. In fact, while Castalia assumes that the user provides path loss-related parameters, this approach automatically derives those values from high-level models such as the environment and path loss [9, 10].

3.3 Methodology: Performance Modelling of Wireless Sensor Networks

Performance modelling and analysis continues to be of great importance in supporting research as well as in the design, development and optimisation of WSN and their applications. The current trend towards the use of WSNs for sensing and control now has the potential for significant advances, not only in science and engineering but also on a broad range of applications. This brings the need for performance modelling for the optimisation of deployment of WSNs. However, the special design, characteristics of sensors and their applications separate them from the traditional networks. These characteristics pose great challenges for the architecture, protocol design, performance modelling and their implementation. It is essential to consider energy efficiency of WSNs because of their limited energy sources (most of the times batteries). In order to minimise the energy consumption, one of the effective techniques is to place sensors in sleep mode during the idle period [49]. In [44, 53, 54], a wake-up scheduling scheme at the MAC layer is proposed, which wakes up the sleeping nodes when there is a need to transmit or receive, thus avoiding a degradation in network connectivity or quality of service (QoS) provisioning.

Characterising delay in distributed systems has been considered in various contexts. However, it can be observed that accurately characterising end-to-end delay at the CH is still an open problem. Considerable amount of research on sensor networks reported recently has been ranging from network capacity and signal processing techniques to topology management algorithms for traffic routing and channel access control. The model presented in [8] is used to investigate system performance in terms of energy consumption, network capacity, delay in data delivery along with the trade-offs that exist between performance metrics and sensor dynamics in active/sleep modes. A Markov model is presented for WSNs, where the nodes may enter into sleep mode. Through standard Markovian techniques, a system model representing the behaviour of a single sensor has been constructed along with the dynamics of the entire network, and the channel contention among interfering sensors. The proposed solution of the system model is then obtained by

means of a fixed point approximation (FPA) procedure, and the model has been validated via simulation.

Due to hardware constraints for energy efficiency, optimising node packet buffer and maximising the performance is necessary to improve the QoS for transmission in WSNs. In [40], a packet buffer evaluation method using queuing network models is proposed, where the blocking probabilities and system performance indicators of each node are calculated using an approximate iterative algorithm. The model considered focuses on a single server model in WSNs and the method used to calculate packet buffer capacity for nodes also indicates that the sink node requires higher performance, when compared to the other nodes in the network. The Markov model of the sensor sleep/active dynamics is presented in [27], which predicts the sensor energy consumption by acquiring this information for each sensor, while a central controller constructs the network energy map representing the energy reserves available in various parts of the system. Only a single node is represented by a Markov chain, while the network energy status is derived with the help of simulation studies.

With regard to analytical studies, results on the capacity of large stationary ad hoc networks are presented in [16]. Two network scenarios were considered; one including arbitrarily located nodes and traffic patterns, while the other one with randomly located nodes and traffic patterns. An analytical approach on network coverage and connectivity of sensor grids is presented in [45]. The sensors are considered unreliable and fail with a certain probability leading to random grid networks. Results on coverage and connectivity are derived as functions of key parameters such as the number of nodes and their transmission radius.

Several approaches based on simulations and experiments have been proposed for performance evaluation of IEEE 802.15.4 networks [13]. In [5], an analytical framework based on a Markov chain characterisation of the MAC protocol is proposed for IEEE 802.11 networks in saturation conditions. Based on this pioneering work, several approaches have been proposed for the characterisation of the MAC performance in IEEE 802.15.4 networks with a star topology. In this work, a scenario with acknowledgement (ACK) messages is considered and an evaluation of the network performance in both saturation and non-saturation regimes is presented, while trying to characterise the conditions under which the network enters the saturation region [31]. A simple Markov chain theoretical model to characterise the sensors as well as the channel status is proposed in [42]. The models show a good agreement with NS-2-based simulations. This model allows to investigate throughput and energy consumption metrics within WSNs. In [26], an extended framework of the one proposed by Ramachandran et al. [42] is presented for a 2-hop network scenario, i.e. networks where sensors communicate with the coordinator through an intermediate relay node, which forwards data packets from the sources (the sensors) towards the destination (the coordinator). Similar works have been presented in [29, 30], emphasising the use of a relay for interconnecting two different clusters in IEEE 802.15.4 networks and analysing the performance through a queueing theoretical analysis. However, the proposed scenario models the (simpler) cases where the relay does not content the medium access to the sensors.

Hence, it is observed that accurately characterising arrivals at the cluster head in WSNs is still an open problem. Although it is quite difficult to analyse each possible application in WSNs, it is sufficient to analyse each class of application classified by data delivery models, as most of these applications in each class have common requirements on the network [12].

3.4 Results and Discussions

Despite the ever-increasing usage of WSNs in modern applications, their development is still plagued by the following issues: (1) development is still performed directly on the top of the operating systems and relies greatly on individual's hard-earned programming skills across all levels of the communication stack (e.g. application, routing, data link levels, etc.) [32]; (2) challenging extra-functional requirements such as performance, security, energy consumption, with poor support for early testing, debugging and simulation of the WSN in an integrated fashion must be addressed by WSN engineers [18]; (3) in order to achieve the desired level of efficiency, the software of a WSN application is tied to specific hardware platforms, thus hampering the reuse of source code and software components across different projects or organisations [32]; (4) due to the intrinsic multidisciplinary nature of the WSN problem space, WSN engineers must continuously collaborate with a high number of system stakeholders (e.g. WSN users, application domain experts, hardware designers and software developers) with different background and training [43].

In current practice, programmers do not only face functional requirements but also challenging non-functional requirements such as lifetime, performance and security. Besides the need of programming abstraction, it is well-accepted the need of abstracting an implementation view into an architectural design. As remarked in [37], end users require high-level abstractions that simplify the configuration of the WSN at large, possibly allowing one to define its software architecture based on pre-canned components. Abstraction is fundamental for future WSN development, as sensors and WSNs in general are becoming important components in pervasive, mobile systems, with new types of stakeholders (e.g. mobile system engineers and developers) with reduced domain-specific technical skills.

As possible solution to the above-mentioned issues, the WSN community is becoming aware of the need of using software engineering approaches in order to support the design, analysis, simulation and implementation of WSNs [37, 52].

In order to simplify the design and configuration of the WSN at large, and abstract from technical low-level details, a number of model-driven engineering (MDE) approaches for WSN engineering have been proposed. Currently, these approaches are used to specify a WSN at different levels of abstraction (hardware, application, communication protocols, etc.) with the recurrent goals of code generation, communication overhead analysis and energy consumption. Many approaches intend to use domain-specific modelling languages (DSML) for

representing WSNs from different viewpoints. For example, the authors in [51] proposed modelling languages with concepts such as node group, region, resource and wireless link; whereas, the authors in [18] proposed a set of languages spanning from application-level actions (e.g. sense, send message and store data) to hardware specifications (e.g. processor, sensing devices and radio transceivers), and so on. Other approaches, such as those proposed by the authors in [15, 33], are based on generic modelling languages; mainly, they use extensions of UML and Simulink for representing a WSN. For what concerns the physical environment of the WSN is that the majority of approaches in the literature does not allow designers to specify the physical deployment of the WSN nodes. Among those that support this feature in some way, there is great variability. There are some that support an explicit definition of the physical environment, others that allow the designers to define physical quantities (e.g. in Ben Maissa et al. [4] engineers can define models of the evolution of each physical quantity in a given scenario), and so on. However, all these approaches do not provide any intuitive and abstract means to easily define the deployment environment of the WSN.

Hence, there is a need for an approach which considers the collaborative nature of WSNs along with its correlation characteristics and various issues from physical layer to application layer together as entities to enable a framework. Also, there is a need to compare work on the existing work on collaborative architectures/methods of WSNs. A clear separation of concerns is also needed as the hardware and software aspects are locked and tied down only to a specific type of nodes, hampering the possibility of reuse across projects and organisations. Along with this, a realistic WSN lifetime estimation and performance evaluation should be possible, in an attempt to improve performance maximising the lifetime of the network.

3.5 Conclusion and Future Directions

The vast number of solutions that have driven the research community over the years have made WSN phenomenon a reality, and hence are recognised as one of the most important technologies for the twenty-first century. It is envisioned that in the near future WSNs will be widely used in various civilian and military fields, and revolutionise the way we live, work and interact with the physical world.

In this research, the simulation approaches considered consider careful assumptions for various layers, also providing a clear separation of concerns among software architecture of the applications, the hardware configuration and the WSN deployment unlike the existing tools for evaluation. Simulation-based studies are presented for all the issues considered. The performance of the layered protocol stack in realistic settings reveals several important interactions between different layers. These interactions are especially important for the design of WSNs in terms of providing realistic estimations and performance evaluation and for maximising the lifetime of the network.

In order to encompass the applicability of WSN architecture and to provide useful information anytime and anywhere, it is of crucial importance to integrate with the Internet. Realisation of these networks will require tight integration and interoperability; however, so far, research has progressed in each of these areas separately. Therefore, it is of crucial significance to develop energy-efficient, location- and spectrum-aware cross-layer communication protocols as well as heterogeneous network management tools for the integration of WSNs, cognitive radio networks, mesh networks and the Internet.

Rapid technological advancement in WSNs has contributed for the great increase in the number of IP-connected smaller smart sensors that, in turn, become part of the IoT. Thus, the interconnection of wireless sensor devices to the Internet facilitates the M2M communication with many application areas such as smart grid, metering, health, environment, vehicle and home appliances. The innovations of integration of WSNs into iThings offer many interesting avenues of research for scientific communities. The research into WSNs for IoT is extremely important which could possibly change our day-to-day lives. The key for IoT applications is the ability to interact with physical world through computation, communication and machine control. However, each sensor device in IoT cannot conveniently communicate with other terminal devices through internet protocol. So, it is ideal to establish protocol translation stack or equipment between two WSN groups.

Sensor network integrates various interdisciplinary technologies, such as sensor technology and embedded computing technology. A multi-hop self-organising system is established in the sensor network through wireless communication, which is responsible for perception collection, processing information perceived within the coverage area of the network and allowing flexible monitoring of the environment. To achieve this vision, there is a need for scalable and interoperable networking systems to support the challenging requirements for future internet and web. The challenges include security, reliability, energy-efficient and cost-effective large-scale sensor networks, machine-to-machine communications and information networking architectures that are suitable for low-end devices through to high-end consumers.

References

1. I.F. Akyildiz, M.C. Vuran, *Wireless Sensor Networks. Advanced Texts in Communications and Networking* (Wiley, New York, 2010)
2. I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, Wireless sensor networks: a survey. Comput. Netw. **38**, 393–422 (2002)
3. T. Arampatzis, J. Lygeros, Senior Member, S. Manesis, A survey of applications of wireless sensors and wireless sensor networks, in *Proceedings of 13th Mediterranean Conference on Control and Automation*, Limassol (2005), pp. 719–724
4. Y. Ben Maissa, F. Kordon, S. Mouline, Y. Thierry-Mieg, Modeling and analyzing wireless sensor networks with VeriSensor (2012), pp. 60–76

5. G. Bianchi, Performance analysis of the IEEE 802.11 distributed coordination function. IEEE J. Sel. Areas Commun. **18**(3), 535–547 (2006)
6. Castalia, Dec (2011). http://castalia.npc.nicta.com.au
7. G. Chen, J. Branch, M.J. Pflug, L. Zhu, B.K. Szymanski, Sense: a wireless sensor network simulator, in *Advances in Pervasive Computing and Networking*, ed. by B.K. Szymanski, B. Yener (Springer, Boston, 2005), pp. 246–267. https://doi.org/10.1007/0-387-23466-7_13
8. C.F. Chiasserini, M. Garetto, Modeling the performance of wireless sensor networks, in *INFOCOM, Twenty-Third Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 1, March 2004
9. K.C. Doddapaneni, Energy aware performance evaluation of WSNs. PhD Thesis, School of Science and Technology, Middlesex University (2015)
10. K. Doddapaneni, E. Ever, O. Gemikonakli, I. Malavolta, L. Mostarda, H. Muccini, Path loss effect on energy consumption in a WSN, in *UKSim* (2012), pp. 569–574
11. W. Du, F. Mieyeville, D. Navarro, I. Connor, L. Carrel, Modelling and simulation of networked low-power embedded systems: a taxonomy. EURASIP J. Wirel. Commun. Netw. **2014**(1), 106 (2014)
12. I.M.M.E. Emary, S. Ramakrishnan, *Wireless Sensor Networks: From Theory to Applications. Telecommunications Books* (Taylor & Francis, Boca Raton, 2013)
13. G. Ferrari, P. Medagliani, S. Di Piazza, M. Martalò, Wireless sensor networks: performance analysis in indoor scenarios. EURASIP J. Wirel. Commun. Netw. **2007**(1), 41 (2007)
14. J. Fontignie, A. Marculescu, Algosensim, Dec (2011). http://tcs.unige.ch/doku.php/code/algosensim/overview
15. G. Fuchs, R. German, Uml2 activity diagram based programming of wireless sensor networks, in *Proceedings of the 2010 ICSE Workshop on Software Engineering for Sensor Network Applications, SESENA '10*, New York (ACM, New York, 2010), pp. 8–13
16. P. Gupta, P.R. Kumar, The capacity of wireless networks. IEEE Trans. Inf. Theory **46**(2), 388–404 (2000)
17. M. Halgamuge, T.-K. Chan, P. Mendis, Experiences of deploying an indoor building sensor network, in *Third International Conference on Sensor Technologies and Applications, SENSORCOMM'09*, pp. 378–381, June 2009
18. M. Imran, A.M. Said, H. Hasbullah, A survey of simulators, emulators and testbeds for wireless sensor networks, in *International Symposium in Information Technology (ITSim)*, vol. 2, pp. 897–902 (2010)
19. T. Issariyakul, E. Hossain, *Introduction to Network Simulator NS2*, 1st edn. (Springer Publishing Company, Incorporated, Berlin, 2008)
20. B. Kantarci, H.T, Mouftah, Trustworthy sensing for public safety in cloud-centric internet of things. IEEE Internet Things J. **1**(4), 360–368 (2014)
21. D. Kotz, C. Newport, R.S. Gray, J. Liu, Y. Yuan, C. Elliott, Experimental evaluation of wireless simulation assumptions, in *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems, MSWiM '04*, New York (ACM, New York, 2004), pp. 78–82
22. T. Krop, M. Bredel, M. Hollick, R. Steinmetz, JiST/MobNet: combined simulation, emulation, and real-world testbed for ad hoc networks, in *Proceedings of the Second ACM International Workshop on Wireless Network Testbeds, Experimental Evaluation and Characterisation, Series WinTECH'07*, New York (ACM, New York, 2007), pp. 27–34. [Online]. Available: http://doi.acm.org/10.1145/1287767.1287774
23. K. Lahmar, R. Cheour, M. Abid, Wireless sensor networks: trends, power consumption and simulators, in *Modelling Symposium (AMS)*, 2012 Sixth Asia, pp. 200–204, May 2012
24. K. Langendoen, W. Hu, F. Ferrari, M. Zimmerling, L. Mottola, *Real- World Wireless Sensor Networks: Proceedings of the 5th International Workshop, REALWSN 2013*, Como (Italy), Sept 19–20. Lecture Notes in Electrical Engineering (Springer, Berlin, 2014)
25. P. Levis, N. Lee, M. Welsh, D. Culler, TOSSIM: accurate and scalable simulation of entire TinyOS applications, in *Proceedings of the 1st International Conference on Embedded Networked Sensor Systems, SenSys'03*, New York (ACM, New York, 2003), pp. 126–137

26. M. Martalò, S. Busanelli, G. Ferrari, Markov chain-based performance analysis of multihop IEEE 802.15.4 wireless networks. Perform. Eval. **66**(12), 722–741 (2009)
27. A.F. Mini, B. Nath, A.A.F. Loureiro, A probabilistic approach to predict the energy consumption in wireless sensor networks, in *IV Workshop de Comunicao sem Fio e Computao Mvel*. So Paulo (2002), pp. 23–25
28. D. Minoli, S. Kazem, B. Occhiogrosso, IoT considerations, requirements, and architectures for smart buildings-energy optimization and next-generation building management systems. IEEE Internet Things J. **4**(1), 269–283 (2017)
29. J. Misic, R. Udayshankar, Slave-slave bridging in 802.15.4 beacon enabled networks in *Wireless Communications and Networking Conference, WCNC 2007* (IEEE, New York, 2007), pp. 3890–3895
30. J. Misic, J. Fung, V.B. Misic, Interconnecting 802.15.4 clusters in master-slave mode: queueing theoretic analysis, in *Proceedings. 8th International Symposium on Parallel Architectures, Algorithms and Networks, ISPAN 2005* (2005)
31. J. Misic, S. Sha, V.B. Misic, Performance of a beacon enabled IEEE 802.15.4 cluster with downlink and uplink track. IEEE Trans. Parallel Distrib. Syst. **17**(4), 361–376 (2006)
32. L. Mottola, G.P. Picco, Programming wireless sensor networks: fundamental concepts and state of the art. ACM Comput. Surv. **43**(3), 191–195 (2011)
33. M.M.R. Mozumdar, F. Gregoretti, L. Lavagno, L. Vanzago, S. Olivieri, A framework for modeling, simulation and automatic code generation of sensor network application, in *5th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad Hoc Communications and Networks, SECON '08* (2008), pp. 515–522
34. Opnet, Dec (2011). http://www.opnet.com
35. M.-S. Pan, L.-W. Yeh, Y.-A. Chen, Y.-H. Lin, Y.-C. Tseng, A WSN-based intelligent light control system considering user activities and profiles. IEEE Sensors J. **8**(10), 1710–1721 (2008)
36. K. Phaebua, T. Lertwiriyaprapa, C. Phongcharoenpanich, M. Krairiksh, Path loss prediction in durian orchard using uniform geometrical theory of diffraction, in *Antennas and Propagation Society International Symposium, APSURSI'09* (IEEE, New York, 2009), pp. 1–4
37. G.P. Picco, Software engineering and wireless sensor networks: happy marriage or consensual divorce? in *Proceedings of the FSE/SDP Workshop on Future of Software Engineering Research, FoSER '10*, New York (2010), pp. 283–286
38. J. Polley, D. Blazakis, J. McGee, D. Rusk, J.S. Baras. Atemu: a fine-grained sensor network simulator, in *First Annual IEEE Communications Society Conference on Sensor and Ad Hoc Communications and Networks, IEEE SECON*, pp. 145–152, Oct 2004
39. D. Puccinelli, M. Haenggi, Wireless sensor networks: applications and challenges of ubiquitous sensing. IEEE Circuits Syst. Mag. **5**, 19–31 (2005)
40. T. Qiu, L. Feng, F. Xia, G. Wu, Y. Zhou, A packet buffer evaluation method exploiting queueing theory for wireless sensor networks. Comput. Sci. Inf. Syst. **8**(4), 1028–1049 (2011)
41. Qualnet, Dec (2011). http://www.scalable-networks.com
42. I. Ramachandran, A.K. Das, S. Roy, Analysis of the contention access period of IEEE 802.15.4 MAC. ACM Trans. Sens. Netw. **3**(1), 4 (2007)
43. K. Romer, F. Mattern, The design space of wireless sensor networks. IEEE Wirel. Commun. **11**(6), 54–61 (2004)
44. C. Schurgers, V. Tsiatsis, S. Ganeriwal, M. Srivastava, Topology management for sensor networks: exploiting latency and density, in *Proceedings of the 3rd ACM International Symposium on Mobile ad hoc Networking & Computing* (2002), pp. 135–145
45. S. Shakkottai, R. Srikant, N. Shro, Unreliable sensor grids: coverage, connectivity and diameter, in *Twenty-Second Annual Joint Conference of the IEEE Computer and Communications, INFOCOM 2003*, vol. 2 (IEEE Societies, Los Alamitos, 2003), pp. 1073–1083
46. Shawnwiki, Jan (2012). http://shawnwiki.coalesenses.com/index.php/Shawn_Introduction
47. S. Shuo, S. Hao, S. Yang, Design of an experimental indoor position system based on RSSI, in *2010 2nd International Conference on Information Science and Engineering (ICISE)*, pp. 1989–1992, Dec 2010

48. Sinalgo, Dec (2011). http://dcg.ethz.ch/projects/sinalgo
49. S. Singh, C.S. Raghavendra, PAMAS - power aware multi- access protocol with signalling for ad hoc networks. SIGCOMM Comput. Commun. Rev. **28**(3), 5–26 (1998)
50. B.L. Titzer et al., Avrora: scalable sensor network simulation with precise timing, in *Proceedings of the 4th International Conference on Information Processing in Sensor Networks (IPSN)* (2005), pp. 477–482
51. C. Vicente-Chicote, F. Losilla, B. AAlvarez, A. Iborra, Pedro Sanchez. Applying MDE to the development of flexible and reusable wireless sensor networks. Int. J. Cooperative Inf. Syst. **16**(3/4), 393–412 (2007)
52. A. Willig, Wireless sensor networks: concept, challenges and approaches. Elektrotechnik und Informationstechnik **123**, 21–31 (2006)
53. W. Ye, J. Heidemann, D. Estrin, An energy-efficient MAC protocol for wireless sensor networks, in *Proceedings of the IEEE Infocom*, New York, June 2002 (IEEE, New York, 2002), pp. 1567–1576
54. R. Zheng, J.C. Hou, L. Sha, Asynchronous wakeup for ad hoc networks, in *Proceedings of the 4th ACM International Symposium on Mobile Ad Hoc Networking and Computing, MobiHoc '03*, New York (2003), pp. 35–45

Chapter 4
False Data Injection Attacks in Internet of Things

Biozid Bostami, Mohiuddin Ahmed, and Salimur Choudhury

4.1 False Data Injection Attacks

For any cyber system, data security and data aggregation is a vital issue. A system's performance, effectiveness, and efficiency all are related to data aggregation and acceptance is related to data privacy. These are one of the vital issues for a cyber system and also become the main target of cyber attacks. With the evolution of technology, the cyber systems are upgrading and new types of cyber attacks are discovered too. With the help of the state-of-the-art technology, the attackers attack different cyber system. There are many forms of cyber attacks and in a summary all of the attacks are done to exploit the target system. In this chapter, we will only focus on a particular type of cyber attack well known as false data injection attack (FDIA). The chapter is organized in the following order: Sect. 4.2 focuses on the FDIA basics and their impact. Section 4.3 outlines FDIA from IoT perspective and also contains the different countermeasures proposed by researchers. Section 4.4 discusses the open challenges that need to be addressed to in designing FDIA prevention application and in Sect. 4.5 the promising impact that can be achieved by deep learning and machine learning to solve FDIA problems is discussed. In Sect. 4.5, we conclude the chapter followed by the references.

B. Bostami
Department of Computer Science and Engineering, Islamic University of Technology, Gazipur, Bangladesh

M. Ahmed (✉)
Centre for Cyber Security and Games, Canberra Institute of Technology, REID, Australian Capital Territory, Australia
e-mail: m.ahmed.au@ieee.org

S. Choudhury
Department of Computer Science, Lakehead University, Thunder Bay, ON, Canada
e-mail: Salimur.choudhury@lakeheadu.ca

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_4

4.2 Impact of False Data Injection Attacks

By definition, false data injection is a cyber attack where the compromised host constructs events which do not take place in that instance of time. In this type of cyber attacks, the attacker can take advantage of the small error rate tolerated by the system algorithms and gradually increase the impact of the injection so that increase of false data is undetected. FDIA requires strong analysis of the target system by the attacker that is the attacker must know the topology of the system. Another requirement for false data injection is that the attacker should have physical access to tamper with the system. FDIA has major impacts on the system. In the following section, we will focus on the impacts of FDIA.

FDIA is a one of the major cyber attacks which can cause high-level damage by altering the data aggregation and creating false results and these attacks can escape bad detection layers too. A lot of research has been done on FDIA. According to the survey of authors of [1, 2], it has severe security and economic impacts on the power system. The impact of FDIA on electricity market was studied and was proposed that huge economic loss can be a result of false data attacks[3–5]. FDIA in cyber controlled systems affects the controllers randomized packet drop and also the performance of the estimator which was studied by Sinopoli et al. [6]. FDIA may lead to system breakdown too. Application like healthcare monitoring and home automation depends on the sensor data. If the sensors are compromised, then wrong report is generated due to false data. Moreover, military surveillance, habitat monitoring, and health care require high level of security because these applications contain sensitive data so privacy and data integrity is an unquestionable requirement for such systems. With the false data injection not only the system generate wrong reports, it also get vulnerable to other cyber attacks. Security and false data in sensor-based network is studied in the following literature [7–10]. In sensor network, false data injection can generate false negative estimations and alter the decision [11]. Moreover, due to the injection false data many legitimate reports are dropped by the compromised nodes. Even the power consumption is affected by FDIA because it generates fabricated events.

In our chapter, our main objective is to investigate the impact of FDIA in IoT. In the following section, we will be focusing on IoT and FDIA.

4.3 False Data Injection Attack in Internet of Things

The Internet of Things (IoT) is becoming the revolution in cyber world. According to the reports found in Internet, profit gaining from IoT will cross 300 billion by 2020. The reason behind such a revolution is that IoT devices are used in different sections of cyber world including habitat monitoring, military surveillance, medical and e-health monitoring, smart home, smart grid, nation security, control systems, inventory management, education systems, wireless sensor networks, RFID, etc.

Due to value and ownership of different applications of IoT, it is targeted by the cyber attackers. One of main challenges in IoT is data privacy, because IoT application contains very sensitive data. Among the different types of attacks, false data injection is one of the major attacks. Since most of the IoT devices remain distributed over a large area in a distributed nature with minimal security, that is why they are more vulnerable. The attacker can tamper with the equipments and compromised nodes can be used to inject bad data. False data injection inserts erroneous data into the system leading the system to generate false reports. Impact of FDIA in smart grid was first studied by Deng et al. [12]. This attack breaks the data integrity and privacy. FDIA cannot be detected by the detection mechanism of the system. Since, in all of the IoT applications data integrity is a questionable requirement that is why FDIA poses a much greater threat. In the following section, we will be focusing on the impacts and countermeasures of FDIA.

4.3.1 Importance of False Data Injection Attack in Internet of Things Applications

IoT devices collect data from various sources and based on data collected by the distributed nodes central system generates the reports. In most cases, these nodes become easy target of false data injection. The impact of false data injection in smart grid has been studied in many literatures [13–16]. Authors in [17–19] focus on the economic effects and loss associated to FDIA in smart grid. False data can lead to fake reports which may cause unnecessary load shedding which was presented by Yuan et al. in [20] and [21]. Moreover, FDIA can cause unnecessary energy consumption to sensor nodes. In applications where nodes have limited power supply, false data will trigger false event which will waste the powers of the node. Again, false data can generate false reports. In this chapter, we highlighted the dangerous impact of false data in IoT application. In the following sections, we will present the countermeasures taken against false data injection presented by different researchers.

4.3.2 Analysis of False Data Injection Attack Countermeasures

In this section, we will review the countermeasures proposed by the researchers in different time and look into their contribution against FDIA in IoT.

In order to prevent false report generation and false data injection, an en route filtering framework was presented in [22]. In their approach, each node contains a symmetric key which is used for report generation. For events, all the nodes collectively generate reports using the symmetric keys containing a set of authentication codes (MACs) following a security threshold. When the reports are

sent to the sink node, each forwarding nodes verify the MAC code for validation in a probabilistic manner. If the MAC is invalid or there is less keys generated threshold, then the respective report is considered as false report. Followed by the en routing filtering, many other research has been done but they all focus on improving the filtering and power consumption of nodes [23–28].

Another type of false data injection addressed as collaborative FDIA was presented in [29]. They attached the keys to nodes based on the geographical position. They proposed two schemes for detecting false reports generation addressed as GFFS and NFFS. According to their proposal, they filter out false data based on the assumption that near sensors should behave similarly to an event depending on the MAC keys and geographical location of the nodes.

Lu et al. [30] presented an authentication scheme for false data detection addressed as BECAN. As proposed, majority of false data is filtered out based on the bit-compressed authentication and random graph property before it reaches the sink node. Such filtering puts minor overhead at the en route nodes but saves the power consumption and lowers the traffic at sink node. Also, sink node encounters very low amount of false data to verify and filter out.

Yang et al. [31] presented PCREF scheme which is also an en routing filter but it is not based on node location and statistical routing methods, instead on polynomial-based scheme. In PCREF, two type of polynomials are used, namely: authentication polynomial and check polynomial to verify the filter out false data unlike checking against the MAC codes. Theoretical and simulation results of this approach show high accuracy even with large set of compromised node set. Yang et al. [32] presented scheme called MDSEF (multidimensional resilient statistical en route filtering) in which key pool contains sets of different groups each containing set of keys. Each node can be associated to a particular group from each set of keys in the key pool. Associating a node to different group improves the filtering accuracy. For nodes to join a group, distributed group joining policy was also proposed. Simulation results showed high accuracy for MDSEF.

Yu et al. [33] presented a scheme for en routing filtering based on the Hill climbing algorithm. This scheme is named as dynamic en routing filtering, DEFS. According to the scheme, the nodes are clustered into groups and cluster head assigns the keys. Based on the assumption that closer nodes in the cluster contain more authentic keys than the far nodes, data is filtered. This approach lowers the overhead close to the sink node. Unfortunately, this approach consumes more power and not suitable for sensor networks with limited power.

In case of smart grid false data injection, geometrical residual filters and algorithms like generalized likelihood were proposed in [34]. A Bayesian-based correlative monitoring approach was presented in [35] which detects false data by training hypothesis testing.

Chen and Abur [36] presented a scheme for false data detection in smart grid with the help of PMU (phasor measurement units) placement. In their work, they formulate integer programming problem and proved that extra PMU can increase false data detection level. Kim and Poor [37] also presented their work based on

PMU for encountering FDIA. The algorithm has low complexity and was able to handle various types of measurements. It also showed that the attacker needs to change the measurement residuals when system is protected by PMUs. However, Gong et al. [38] presented an attack based on timestamp by spoofing GPS and sending attack signals which may fail the PMU systems. Liu et al. [39] presented that FDIA is still possible if the attacker mask the outage of a single line when attacking multiple measurements.

Many other methods have been presented with an assumption that network topology is the key knowledge for the cyber attacker. Talebi et al. [40] presented a scheme for defending false data injection by dynamically changing the information structure of the grid. Huang et al. [41] presented an algorithm of adaptive CUSUM which defends FDIA in smart grid. Hop-by-hop authentication scheme for lowering the cost at base stations was presented by Zhu et al. [42]. Liu et al. [43] in his work converted the FDIA as matrix separation problem and he presented nuclear norm minimization and low rank matrix factorization to solve the problem. Chaojun et al. [44] presented a scheme for FDIA detection based on Kullback–Leibler distance (KLD) which tracks dynamics of measurement variations.

Bi et al. [45] in their work take into account many critical state variables and convert the problem into Steiner tree used in graph theory. Then, graphical methods were introduced to select the minimum number of meter measurements. In addition, they presented mixed protection strategy [46].

Yu et al. [47] presented anomaly detection mixed with watermarking-based detection scheme which prevents stealthy attacks like FDIA in smart grid. Similar work has been done in wireless sensor networks with low power by Kamel and Juma [48].

In this section, we have tried to review some of the literatures that have been focusing on the FDIA in smart grid and wireless sensor networking. Although these are a part of IoT, but still many other parts are still undiscovered and no significant research has not been found.

4.4 False Data Injection Attack Detection and Prevention Challenges in Internet of Things

FDIA is one of the major types of attacks in IoT. It not only injects false data in the system but also degrades the service of the system too. FDIA destroys the data integrity and also the data privacy. Since during FDIA is a type of stealth attack, detection and prevention from this kind of attack need extra care. Many research have been carried out for detection of false data injection and proposed some counter measures against such attack but most of them are domain specific like: smart grid, smart control system wireless sensor networks, etc. Again, each proposal has some drawbacks as well. In our chapter, we tried to outline some of open challenges associated with designing false data detection system in IoT.

4.4.1 System Security

In FDIA, the attacker tampers with the system and uses the compromised system to carry out the attack. That is why, we need to make sure that the system should not be easy enough to reconfigure by an attacker without being detected. For example, the attacker can mislead the meter readings. System security does not mean only the physical security. System security includes which protocol to use in case of data transmission and how the devices should communicate, deciding on the routing protocols for secure data transmission. In [49], two routing protocols were presented for vehicle-to-vehicle (V2V) communication. In one protocol, the final node address was known which is source routing and in the second case next node address was known which is known as hop-to-hop routing. Their work addresses the environment with a large number of nodes. Some routing protocols are also based on the geographical location of the nodes, but each has some limitations on data transmission and security vulnerability.

The challenge here is to design a secure system which ensures secure data transmission over the network. Secure system is hard to tamper with and it is also very challenging to make a system highly secure. We need to make sure that the network protocol by which the data is exchanged should be secure too. Even the whole application should be well designed so that no false value can be injected.

4.4.2 Data Storage

Another challenge in preventing FDIA is data storage. IoT devices' data collection and transmission produces a high volume of data which need to be stored for analysis. In order to overcome the data loss, data needs to be kept on the sender side until a successful transmission is done, which multiples the volume of data in the network. But, IoT devices have limited data storage. FDIA can cause high data traffic, leading to system failure. Moreover, if the current storage limit of a node is reached due to false data that node will discard incoming true data which also affects the whole system. That is why, we need to make sure that false data are discarded at early stage. The longer false value stays in the system, the higher chances are that true value may get neglected as there is limited storage. Network congestion and traffic handling due to storage limitation is also an open challenge of IoT. Protocols for handling high traffic congestion with low storage need to be addressed as well.

4.4.3 Data Sanitization

Another issue with IoT device is the data sanitization. IoT comprises of wide range of devices, each device is different from other in nature. Each device interpreted

data in various formats, leading data interoperability very challenging. To ensure security, we need to sanitize the data so that they can be interpreted by other devices. False data can be detected in early stage of transmission if all the collected data are sanitized and transformed to a certain format. Moreover, data sanitization discards the unused data which also reduce the size and network traffic. Since FDIA injects false value to system by using a compromised node which behaves like a normal node, that is why data coming from each node needs to be verified before sending it to the base stations or report generator. Algorithm must be introduced to make sure that the collected data is sanitized and endorsed before they are aggregated.

4.4.4 Power Consumption

One major constraint of IoT devices is the power consumption. Since most of the IoT device location changes dynamically, that is why they use battery to sustain and sometimes regenerate by consuming power from surrounding environment using some device. That is why, all the schemes need to design for longer life of the device, for that low power consumption needs to be addressed. [50] addressed the low power consumption issue and presented a modular circuit design and it was proved to be highly effective in theory. But, this area needs a lot of more research to be done.

In order to address the false data detection, we also need to address the power consumption constraint. Since devices have limited power supply and life power, that is why a scheme is needed which uses less power consumption and carries out the filtering of bad data. The main challenge here is to minimize the trade-off.

4.4.5 Tolerance Level

Every detection algorithm has a minimum error rate for which a system cannot detect false data with full accuracy. In FDIA, the attacker takes advantage of this error rate to exploit the system and inject wrong data. The attacker analyzes the system state over a time frame and learns the error rate. Based on the error rate, false data is injected into the system so that the detection algorithms do not get triggered. The attacker slowly increases amount of false data, tricking the algorithm to consider the false data to be true.

Tolerance level of the system indicates how robust the system is to fake data. In order to prevent false data injection, system tolerance level should be kept as low as possible. Not only FDIA, other cyber attacks also take advantage. It is also a challenge in IoT domain, as IoT has heterogeneous system design.

4.4.6 Access Control

In order to protect the data from attacker, the access control should be limited. Data encryption techniques can protect the data integrity and prevent injecting wrong data. [51] presented an algorithm for user authorization based on dynamic key generation. Even though encryption techniques prevent outsider attacks but they cannot prevent insider attacker. The insider attacker can be an authorized user. So, access control should be carefully distributed among the users. In order to prevent FDIA, access to data manipulation by any user should be restricted. In FDIA, the attacker can be an authorized user so any manipulation on the data should not be allowed and if any such event occurred then system should be able to trigger the intrusion.

Access control is critical for IoT devices. Since most of the devices are located in an unsecured location, by limiting the user access control we can easily detect if any attacker tries to tamper with any device after the deployment. New techniques are needed to be addressed for monitoring the sensors after they are deployed. This is also an open challenge of IoT.

4.4.7 Resilience to Attacks

IoT device redeployment can be sometimes very costly in most cases. And, as there are many kinds of cyber attacks, resilience to attacks is also another challenge in IoT domain. New technology should be introduced in order to recover from the FDIA attacks. Attacker captures valid nodes to inject bad data in the systems, it is necessary for the system to be intelligent enough to overcome this type of attacks. By introducing new resilience techniques, IoT devices can overcome the FDIA. The system should be designed in an intelligent way so that if the system is tampered with attacker then it should recover on by its own.

No significant approach was found which addressed the FDIA and proposed recovery methods. So, a lot of research can be carried out in this area.

4.5 Future Directions

Even though FDIA is one of the major attacks in IoT domain, still very few research has been done on it. Most of the research are done only on targeted domains like sensor networks and smart grids. But, other application domains like e-health, smart city, environment monitoring, education, nation security, and home networks were neglected in which FDIA has equal or more impact. Moreover, these domains are different from architectural point. They all have their own protocols and limitations. In this section, we will be focusing on the emerging technique, i.e., deep learning for solving the FDIA problems much faster than traditional approaches.

4.5.1 *False Data Injection Attack Modelling Using Deep Learning*

Very few research has been done on FDIA using machine learning, deep learning, statistical methods, and artificial intelligence. FDIA's online sequential detection presented in [52] used cumulative sum algorithm based upon GLR (generalized likelihood ratio). Their methods outperformed the first-ordered cumulative sum detectors in speed and accuracy. Valenzuela et al. [53] in their work presented a PCA-based algorithm to detect false data in real-time flow. PCA was used to differentiate between the false power flow from regular flow in smart grid. Kosut et al. presented FDIA detection algorithm based on Bayesian framework [54]. They used the heuristics to detect the false data with very low overhead. Landford et al. presented a machine learning method with two-class SVM. This method analyzes the changes of PMU parameters for FDIA detection. They used Pearson correlation coefficient [55]. Supervised learning can be introduced to label the data and detect the false data. Support vector machine (SVM) and artificial neural network (ANN) can also be introduced in FDIA detection for better performance. These machine learning algorithm and deep learning need to be studied more in IoT and FDIA. With careful design, deep learning algorithm can outperform the former methods in speed and accuracy, even with low overhead to nodes' computational power and battery life. Deep learning in FDIA is still an undiscovered world to the research community.

4.6 Conclusion

In this chapter, we tried to explore the existing research that has been done on FDIA and also outline the domains which were neglected so far, e.g., control system, smart education, smart home, e-health, environment surveillance, military surveillance, etc. All the research done so far focusing on the FDIA is limited to smart grid section. Very few research has been done focusing on the impact of FDIA on wireless sensor networks too. But, other domains of IoT were heavily ignored so far, which also poses threat of false data injection. Due to much difference in IoT devices, proposing an algorithm targeting only one type is not sufficient. Here, we reviewed the works of the researchers done on the smart grid and sensor network domain of IoT, targeting the FDIA and analyzed the impacts. Moreover, we have highlighted some of open challenges in IoT which needs to be addressed in preventing FDIA and also highlighted that how deep learning can be used to IoT domain to gain better performance.

Our main purpose of this chapter was to focus on the fact that even though the implications of FDIA on IoT are addressed, but it was limited to smart grid only. Some of the domain of IoT is still undiscovered. A lot of research are yet to be carried out to meet the challenges in preventing FDIA. Moreover, we tried to highlight another fact that deep learning can have significant effect in designing

IoT application as IoT is merging with Big data. Hope, this chapter will help the community of researchers to start discovering the undiscovered domain of IoT and help in their future works.

References

1. Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids, in *Proceedings of the 16th ACM Conference on Computer and Communications Security (CCS'09)*, Chicago, IL, 9–13 November 2009, 12 pp
2. Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids. ACM Trans. Inf. Syst. Secur. **14**(1), Article 13/1–33 (2011)
3. L. Xie, Y.L. Mo, B. Sinopoli, Integrity data attacks in power market operations. IEEE Trans. Smart Grid **2**(4), 659–666 (2011)
4. L.Y. Jia, J. Kim, R.J. Thomas et al., Impact of data quality on real-time locational marginal price. IEEE Trans. Power Syst. **29**(2), 627–636 (2014)
5. D.H. Choi, L. Xie, Impact analysis of locational marginal price subject to power system topology errors, in *Proceedings of the 2013 IEEE International Conference on Smart Grid Communications*, Vancouver, 21–24 October 2013, pp. 55–60
6. L. Schenato, B. Sinopoli, M. Franceschetti, K. Poolla, S. Sastry, Foundations of control and estimation over lossy networks. Proc. IEEE **95**(1), 163–187 (2007)
7. S. Jaggi, M. Langberg, S. Katti, T. Ho, D. Katabi, Resilient network coding in the presence of byzantine adversaries. Proc. IEEE Trans. Inf. Theory **54**(6), 2596–2603 (2008)
8. M. Ma, Resilience of sink filtering scheme in wireless sensor networks. Comput. Commun. **30**(1), 55–65 (2006)
9. A. Parakh, S. Kak, Space efficient secret sharing for implicit data security. Inf. Sci. **181**(2), 335–341 (2011)
10. F. Ye, H. Luo, L. Zhang, Statistical en-route filtering of injected false data in sensor networks, in *Proceedings of 23th Annual Joint Conference of the IEEE Computer and Communications Societies* (2004), pp. 2446–2457
11. Z. Su, C. Lin, F.J. Feng, F.Y. Ren, Key management schemes and protocols for wireless sensor networks. J. Softw. **18**(5), 1218–1231 (2007)
12. R. Deng, G. Xiao, R. Lu, H. Liang, A.V. Vasilakos, False data injection on state estimation in power systems-Attacks, impacts, and defense: a survey. IEEE Trans. Ind. Inf. **13**(2), 411–423 (2017)
13. Y. Liu, P. Ning, M.K. Reiter, False data injection attacks against state estimation in electric power grids. ACM Trans. Inf. Syst. Secur. **14**(1), Article 13/1–33 (2011)
14. G.Q. Liang, J.H. Zhao, F.J. Luo et al., A review of false data injection attacks against modern power systems. IEEE Trans Smart Grid **8**, 1630–1638 (2016)
15. J.W. Liang, L. Sankar, O. Kosut, Vulnerability analysis and consequences of false data injection attack on power system state estimation. IEEE Trans. Power Syst. **31**, 3864–3872 (2016)
16. G. Hug, J.A. Giampapa, Vulnerability assessment of AC state estimation with respect to false data injection cyber-attacks. IEEE Trans. Smart Grid **3**(3), 1362–1370 (2012)
17. L. Xie, Y.L. Mo, B. Sinopoli, Integrity data attacks in power market operations. IEEE Trans. Smart Grid **2**(4), 659–666 (2011)
18. L.Y. Jia, J. Kim, R.J. Thomas et al., Impact of data quality on real-time locational marginal price. IEEE Trans. Power Syst. **29**(2), 627–636 (2014)
19. D.H. Choi, L. Xie, Impact analysis of locational marginal price subject to power system topology errors. In: Proceedings of the 2013 IEEE International Conference on Smart Grid Communications, Vancouver, 21–24 October 2013, pp. 55–60

20. Y.L. Yuan, Z.Y. Li, K. Ren, Modeling load redistribution attacks in power systems. IEEE Trans. Smart Grid **2**(2), 382–390 (2011)
21. Y.L. Yuan, Z.Y. Li, K. Ren, Quantitative analysis of load redistribution attacks in power systems. IEEE Trans. Parallel Distrib. Syst. **23**(9), 1731–1738 (2012)
22. F. Ye, H. Luo, L. Zhang, Statistical en-route filtering of injected false data in sensor networks, in *Proceedings of 23th Annual Joint Conference of the IEEE Computer and Communications Societies* (2004), pp. 2446–2457
23. E. Ayday, F. Delgosha, F. Fekri, Location-aware security services for wireless sensor networks using network coding, in *IEEE Conference on Computer Communications* (2007), pp. 1226–1234
24. K. Ren, W. Lou, Y. Zhang, Providing location-aware end-to-end data security in wireless sensor networks, in *Proceedings of the IEEE Conference on Computing and Communicating* (2006), pp. 585–598
25. H. Wang, Q. Li, PDF: a public-key based false data filtering scheme in sensor networks, in *Proceedings of the International Conference on Wireless Algorithms, Systems and Applications* (2007), pp. 129–138
26. Y. Zhang, J. Yang, H. Vu, The interleaved authentication for filtering false reports in multipath routing based sensor networks, in *Proceedings of 20th International Parallel and Distributed Processing Symposium* (2006), pp. 1–10
27. S. Zhu, S. Setia, S. Jajodia, An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks, in *Proceeding IEEE Symposium on Security and Privacy* (2004), pp. 259–271
28. L. Zhou, C. Ravishankar, A fault localized scheme for false report filtering in sensor networks, in *Proceedings of the IEEE International Conference on Pervasive Services* (2005), pp. 59–68
29. J.X. Wang, Z.X. Liu, S.G. Zhang, X. Zhang, Defending collaborative false data injection attacks in wireless sensor networks. Inf. Sci. **254**, 39–53 (2014)
30. R.X. Lu, X.D. Lin, H.J. Zhu, X.H. Liang, X.M. Shen, BECAN: a bandwidth-efficient cooperative authentication scheme for filtering injected false data in wireless sensor networks. IEEE Trans. Parallel Distrib. Syst. **23**(1), 32–43 (2012)
31. X.Y. Yang, J. Lin, P. Moulema, W. Yu, X.W. Fu, W. Zhao, A novel en-route filtering scheme against false data injection attacks in cyber-physical networked systems, in *Proceedings of the IEEE International Conference on Distributed Computing Systems (ICDCS)* (2012), pp. 92–101
32. F. Yang, X.H. Zhou, Q.Y. Zhang, Multi-dimensional resilient statistical en-route filtering in wireless sensor networks, in *Advances in Grid and Pervasive Computing*. Lecture Notes in Computer Science (Springer, Berlin, 2010), pp. 130–139
33. Z. Yu, Y. Guan, A dynamic en-route scheme for filtering false data injection in wireless sensor networks, in *Proceedings of 25th Annual Joint Conference of the IEEE Computer and Communications Societies* (2006), pp. 1–12
34. S. Cui, Z. Han, S. Kar, T.T. Kim, H. Poor, A. Tajer, Coordinated data-injection attack and detection in the smart grid: a detailed look at enriching detection solutions. IEEE Signal Process. Mag. **29**(5), 106–115 (2012)
35. M.G. Kallitsis, S. Bhattacharya, S. Stoev, G. Michailidis, Adaptive statistical detection of false data injection attacks in smart grids, in *2016 IEEE Global Conference on Signal and Information Processing (GlobalSIP)* (2016), pp. 826–830
36. J. Chen, A. Abur, Placement of PMUs to enable bad data detection in state estimation. IEEE Trans. Power Syst. **21**(4), 1608–1615 (2006)
37. T.T. Kim, H.V. Poor, Strategic protection against data injection attacks on power grids. IEEE Trans. Smart Grid **2**(2), 326–333 (2011)
38. S. Gong, Z. Zhang, H. Li, A.D. Dimitrovski, Time stamp attack in smart grid: physical mechanism and damage analysis, preprint (2012), http://arxiv.org/abs/1201.2578
39. X. Liu, Z. Li, Z. Li, Impacts of bad data on the PMU based line outage detection, Preprint (2015), http://arxiv.org/abs/1502.04236

40. M. Talebi, C. Li, Z. Qu, Enhanced protection against false data injection by dynamically changing information structure of microgrids, in *IEEE 7th Sensor Array and Multichannel Signal Processing Workshop (SAM)* (2012), pp. 393–396
41. Y. Huang, H. Li, K.A. Campbell, Z. Han, Defending false data injection attack on smart grid network using adaptive CUSUM test, in *IEEE 45th Annual Conference on Information Sciences and Systems (CISS)* (2011), pp. 1–6
42. S. Zhu, S. Setia, S. Jajodia, P. Ning, An interleaved hop-by-hop authentication scheme for filtering of injected false data in sensor networks, in *Proceedings of IEEE Symposium on Security and Privacy*, Oakland, CA (2004), pp. 259–271
43. L. Liu, M. Esmalifalak, Q. Ding, V.A. Emesih, Z. Han Detecting false data injection attacks on power grid by sparse optimization. IEEE Trans. Smart Grid **5**(2), 612–621 (2014)
44. G. Chaojun, P. Jirutitijaroen, M. Motani, Detecting false data injection attacks in AC state estimation. IEEE Trans. Smart Grid **6**, 2476–2483 (2015)
45. S. Bi, Y.J. Zhang, Defending mechanisms against false-data injection attacks in the power system state estimation, in *Proceedings of IEEE GLOBECOM Workshops (GC Wkshps)* (2011), pp. 1162–1167
46. S. Bi, Y.J. Zhang, Graphical methods for defense against false-data injection attacks on power system state estimation. IEEE Trans. Smart Grid **5**(3), 1216–1227 (2014)
47. W. Yu, D. Griffith, L. Ge, S. Bhattarai, N. Golmie, An integrated detection system against false data injection attacks in the smart grid. Secur. Commun. Netw. **8**, 91–109 (2015). https://doi.org/10.1002/sec.957
48. I. Kamel, H. Juma, Simplified watermarking scheme for sensor networks. Int. J. Internet Protoc. Technol. **5**(1), 101–111 (2010)
49. S. Agrawal, D. Vieira, A survey on internet of things: security and privacy issues. Abakós **1**, 78–95 (2013)
50. D. Blaauw, D. Sylvester, P. Dutta, Y. Lee, I. Lee, S. Bang, Y. Kim, G. Kim, P. Pannuto, Y.-S. Kuo, D. Yoon, W. Jung, Z. Foo, Y.-P. Chen, S. Oh, S. Jeong, M. Choi, IoT design space challenges: circuits and systems, in *2014 Symposium on VLSI Technology (VLSITechnology): Digest of Technical Papers* (2014), pp. 1–2
51. L. Ting, L. Yang, S. Yao, M. Yashan, G. Xiaohong, A dynamic secret-based encryption method in smart grids wireless communication. IEEE Trans. Smart Grid **5**, 1175–1182 (2013)
52. S. Li, Y. Yilmaz, X. Wang, Quickest detection of false data injection attack in wide-area smart grids. IEEE Trans. Smart Grid **6**(6), 2725–2735 (2015)
53. J. Valenzuela, J. Wang, N. Bissinger, Real-time intrusion detection in power system operations. IEEE Trans. Power Syst. **28**(2), 1052–1062 (2013)
54. O. Kosut, L. Jia, R.J. Thomas, L. Tong, Malicious data attacks on the smart grid. IEEE Trans. Smart Grid **2**(4), 645–658 (2011)
55. J. Landford et al., Fast sequence component analysis for attack detection in synchrophasor networks, in *Proceedings of 5th International Conference on Smart Cities Green ICT Systems (SmartGreens)*, Rome (2016), p. 268

Chapter 5
Energy-Efficient Clustering for Wireless Sensor Devices in Internet of Things

Diletta Cacciagrano, Rosario Culmone, Matteo Micheletti, and Leonardo Mostarda

5.1 Introduction

The Internet of Things (IoT) is composed of interrelated smart objects, wireless devices, and people that can autonomously communicate data over the network. Different studies predict that the global IoT market will grow from $157B in 2016 to $457B by 2020. Transportation and logistics, smart homes, smart supply chain, smart cities, connected cars, smart industry, and smart retails are examples of applications that will benefit from the Internet of things technology.

Wireless sensor networks (WSNs) play a very important role for implementing the vision of the IoT; they behave as a digital skin and implement a virtual layer where the information of the physical world can be read by the computational system. Wireless sensor networks (WSNs) are composed of spatially distributed sensors that can autonomously collect environmental data.

Sensors can produce a large volume of data and can have heterogeneous features such as computational power, memory, and communication capabilities. WSNs are referred to as homogeneous when all the nodes are equal, for instance, they have the same hardware and the same transmission rate. A WSN which is not homogeneous is referred to as heterogeneous. Devices are usually battery-powered; thus, gathering data from a WSN in an energy-efficient way is quite important.

Clustering is one of the energy-efficient solutions that has been proposed by the research community in order to gather data from a WSN. This produces a set of clusters. Each cluster has a set of member nodes and a cluster head (CH). This gathers data from its members (intra-cluster communication). CHs cooperate in order to report data to a centralized base station (BS) (intercluster communication).

D. Cacciagrano · R. Culmone · M. Micheletti · L. Mostarda (✉)
Department of Computer Science, University of Camerino, Camerino, Italye-mail: diletta.cacciagrano@unicam.it; rosario.culmone@unicam.it; matteo.micheletti@unicam.it; leonardo.mostarda@unicam.it

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_5

In this chapter, we review and compare different energy-efficient clustering protocols for heterogeneous WSNs. We also consider various protocols for homogeneous WSNs which have been adapted in the heterogeneous context. We describe our novel Rotating Energy-Efficient Clustering for Heterogeneous Devices (REECHD) [17]. REECHD is a clustering protocol for heterogeneous WSNs that introduces a novel leader election protocol which considers the node residual energy and the node induced work. This is estimated by using the node transmission rate. REECHD also introduces the concept of intra-traffic rate limit (ITRL). This defines a limit on the intra-traffic communication that all WSN clusters must comply with. ITRL can be used to improve energy efficiency. We compare REECHD with various clustering protocols. Comparison is performed by simulating all protocols with the same case study and the same assumptions. This ensures a fair comparison.

The rest of the article is organized as follows: Sect. 5.2 reviews the state of the art of clustering for homogeneous and heterogeneous WSNs; Sect. 5.3 details the REECHD election and its novel contribution as well as the algorithm for cluster formation; Sect. 5.4 describes the network model and the simulation results; finally, Sect. 5.5 concludes this article.

5.2 State of the Art of Clustering for WSNs

A great deal of literature and research articles are available on clustering protocols. In this section, we focus on existing prominent clustering protocols for homogeneous and heterogeneous WSNs. We consider clustering approaches having equal- and unequal-size clustering, rotation and non-rotation, single-hop and multi-hop. We conclude the section with clustering protocols that consider harvesting and IoT devices, and protocols which are based on machine learning.

5.2.1 Clustering Protocols for Homogeneous WSNs

Low-Energy Adaptive Clustering Hierarchy (LEACH) [9] is one of the pioneering routing protocols that introduced the idea of clustering into the field of WSNs. Unlike most of the clustering protocols, which use the node residual energy for cluster election, LEACH uses a probabilistic function. All cluster heads can directly communicate with BS, i.e., multi-hop communication never takes place. Once a node has been elected as a CH, it cannot take the same role in the next cluster election. LEACH proposes a randomized rotation of CHs and data aggregation at each CH.

HEED [25] clustering protocol produces clusters of equal size, i.e., each cluster has the same radius. The HEED algorithm is composed of the following two phases: (1) clustering and (2) network operation. During the clustering phase, CHs get

elected based on the residual energy, and member nodes join the closest CH.[1] During the network operation phase, data messages get delivered from the members to the BS. Clustering and network operation phases are repeated over time. HEED generally prevents two nodes within the same transmission range from becoming CHs. As reported in [25], sensor nodes close to the BS deplete their energy faster with respect to nodes that are farther away. This problem is referred to as hot spot problem. In fact, while all CHs will have the same amount of average intra-traffic communication (i.e., the traffic inside a cluster) CHs close to the BS have a higher intercluster communication (i.e., relay traffic amongst CHs).

Distributed Weight-based Energy-efficient Hierarchical Clustering protocol (DWEHC) [4] is an equal-size clustering-based protocol for WSNs. It optimizes intra-cluster communication by introducing multi-hop transmission within the clusters. All sensor nodes execute DWEHC individually to decide whether to be a cluster head or a member node. DWEHC clustering formation phase is based on HEED topology. Resultant clusters arrangement is well-balanced and leads to enhance network lifetime.

Voluminous literature has been developed on devising energy-efficient unequal-size clustering protocols for WSNs.

Energy-Efficient Unequal Clustering (EEUC) algorithm [15] for WSNs is one of the first approaches that had been conceived. EEUC is based on the idea that a larger cluster size should be used when the CH resides in zones farthest from BS, whereas zones nearest to BS should be populated with a considerable amount of smaller clusters. This approach would minimize excessive overhead burden on cluster heads nearest to BS and should alleviate the energy hole or hot spot problem.

Unequal clustering algorithm based on HEED (UHEED) [7] is an unequal-size clustering-based protocol for WSNs. UHEED incorporates the idea of EEUC protocol into HEED in order to build unequal-size clusters. The size of a cluster CH depends on its distance from the BS. The farther away CH is from the BS, the larger its competition radius is. In other words, clusters that are farther away from the BS have a larger radius with respect to clusters nearer to the BS. UHEED reduces the hot spot problem and increases network lifetime when compared to HEED and LEACH.

Rotated Unequal HEED (RUHEED) [1] uses an unequal-size clustering-based approach that not only improves the hot spot problem but also enhances the network lifetime. RUHEED is composed of three stages that are CH election, clusters formation, and CH rotation. HEED is used to elect CHs based on its residual energy and communication cost. EEUC concept, which is based on the sensor node distance from the BS, is used in order to establish unequal-sized clusters. During CH rotation phase, current CH selects the member nodes with the highest energy and directly designates it as the next cluster head. Rotation strategy avoids re-clustering of the network; thus, network lifetime is improved. Re-clustering of the network takes place when any of the sensor nodes drain its entire energy. RUHEED preserves energy and minimizes the number of cluster election and cluster formation phases.

[1] Communication costs can be considered to elect or join a CH.

ER-HEED [23] is a clustering protocol that enhances performance of HEED by introducing CH role rotation inside clusters. ER-HEED is composed of three stages that are cluster head election, cluster formation using HEED, and cluster head rotation. Like RUHEED, CHs nominate the next CHs that have the highest residual energies. This concept of CH selection within the cluster member nodes reduces the number of cluster elections. HEED-based cluster head election is performed only when any of the sensor nodes depletes its energy completely. ER-HEED performance in terms of first node dies measure criteria is far superior to RUHEED, HEED, and UHEED.

5.2.2 Clustering Protocols for Heterogeneous WSNs

While WSNs have homogeneous nodes, heterogeneous WSNs introduce nodes that can have differences in the following features: (1) energy level, (2) data rate, (3) transmission range, (4) aggregation performance, and (5) processing capabilities. Heterogeneity affects significantly the network lifetime and lessens network response time [24]. In this section, we describe various clustering algorithms that have been devised for heterogeneous wireless sensor networks. Different protocols can make different assumptions about the heterogeneity of the WSNs.

DEEC (distributed energy-efficient clustering algorithm for heterogeneous WSNs) [19] is an equal-size clustering protocol. DEEC cluster head election is based on a probability that is calculated by considering the ratio of the residual sensor node energy and the network average energy. The CH role is rotated among sensor nodes on the basis of their residual energies. This ensures a uniform energy consumption over the entire network. Sensor nodes that have the highest residual and highest initial energies will be more likely selected as cluster heads. BS broadcasts the network average energy information to all wireless sensor network nodes.

Distributed energy balance clustering Protocol for heterogeneous WSNs (DEBC) [5] is a clustering protocol for heterogeneous WSNs. DEBC assumes that sensor nodes have heterogeneous energy levels. The cluster head election is based on the sensor node residual energy. Sensor nodes that have the highest initial energy and the highest residual energies are highly probable to be selected as cluster heads. The simulation results show that the performance of DEBC is superior to that of LEACH and SEP [22].

The authors in [10] describe a distributed clustering with load balancing (DBLC) for forming clusters efficiently and balancing load in intercluster communication. Size (range) is important in terms of energy efficiency and balancing load in multihop communication of CHs. This avoids energy inefficiency and produces balanced load of cluster. Balanced intertraffic communication is achieved by using clusters with different sizes at each step.

The authors in [12] proposed a distributed CH election approach for heterogeneous WSNs. The election of cluster heads is based on a weighted probability.

Member nodes communicate with their CH and then CHs communicate the aggregated information to the base station. Three different types of nodes are considered and all have different thresholds. The weight assigned to each node will decide the selection of cluster head for each type.

Energy-efficient heterogeneous clustered scheme for wireless sensor networks (EEHC) [11] is a clustering protocol for heterogeneous WSNs. In EEHC, a percentage of sensor nodes are equipped with various levels of battery capacity. EEHC aims at enhancing network efficiency and reliability. Like DEEC and DEBC, the cluster head election probability of EEHC depends on sensor node residual energies.

A stable election protocol for clustered heterogeneous wireless sensor networks (SEP) [22] is a heterogeneous protocol and intends to enhance network lifetime according to the first node dies network lifetime measure. SEP assumes two different types of nodes that are normal and advanced sensor nodes. CH election is based on sensor node initial and residual energies. Simulation results show that SEP prolongs network lifetime and average throughput.

FMUC (feedback mechanism-based unequal clustering) [16] is a feedback mechanism-based unequal heterogeneous protocol. FMUC is specifically designed to avoid the energy hole problem when balancing the energy load in application-based WSNs. Initially, FMUC divides the network into layers which are computed analytically. A mathematical model is used in order to uniform the ratio of the energy consumption and the total initial energy of each layer. Each cluster will belong to the one of the layers. The size of each cluster is calculated by considering the ratio of the energy consumption of each layer. Clusters send their sizes as feedback to the sink which broadcasts the collected values into the network. All nodes of the WSN receive the feedback values but only the cluster heads change their competition radius according to received values.

5.2.3 Clustering Protocols with Harvesting

Harvesting is the capability of sensor nodes to be able to harvest energy either from a dedicated or an opportunistic ambient source such as solar, thermal, wind, and vibration.

The authors in [20] give a detailed list of various sources that can be used to harvest energy in WNSs. They can be categorized into ambient sources and external sources.

Ambient sources are

- Radio Frequency-based energy harvesting where RF-based energy harvesting received radio waves are converted to DC power after conditioning.
- Solar-based energy harvesting where solar energy is an affordable and clean energy source useful to eliminate the energy problem in WSNs. The photovoltaic effect converts solar rays into DC power when certain semiconductor materials

are exposed to sunlight. We remark that it is not possible to perform solar energy harvesting during the night. Thus, developers have to ensure the highest possible efficiency during daylight hours to guarantee the viability of solar-power.
- Thermal-based energy harvesting where it is possible to convert heat energy into electrical energy exploiting the Seebeck effect. This requires a load to be attached across the heated and cold faces of a Thermoelectric Generator (TEG) for thermal energy harvesting. This can be done at different scales, from large to small. In WSNs, we often need to keep the scale as small as possible. In this scenario, for instance, it can be interesting to generate power from human body temperatures.
- Flow-based energy harvesting generally uses turbines and rotors to convert rotational movement into electrical energy using electromagnetic induction principal.

External sources are

- Mechanical-based energy harvesting that is performed with sources such as vibrations, pressure, and stress–strain. To do this, a suitable Mechanical-to-Electrical Energy Generator (MEEG) is needed. A MEEG uses electromagnetic, electrostatic, or piezoelectric mechanisms to harvest energy.
- Human-based energy harvesting that is used in Wireless Body Area Network (WBAN). In these networks, sensor nodes are deployed on or inside of the human body to monitor physiological parameters continuously. These nodes need to be operational for long periods of time or even for the lifetime of the humans being monitored. Human-based energy harvesting can be categorized as activity based harvesters and inherent physiological parameters based harvesters. More precisely, energy can be harvested from humans in several ways, such as through locomotion, changes in finger position, body heat, and blood flow. Nevertheless, the main challenge still is to miniaturize sensors to make them easier for human adoption.

In [3], the following different energy-harvesting approaches are investigated: (i) energy harvesting combined with simultaneous data decoding (a trade-off between the amount of energy that can be stored for future use and the amount of energy that should be spent for signal decoding); (ii) energy-efficient operation of wireless sensor networks, making use of appropriate routing schemes or scheduled operation of sensor nodes; (iii) mobile chargers which stop at optimal locations to perform charging; and (iv) energy sharing. The approaches (iii) and (iv) are chosen in a two-step protocol combining a mobile charger which moves inside the network toward the next discharged CH to overcharge it and an energy trading between overcharged CHs and other nodes. The energy trading takes place inside each cluster: A CH is chosen getting the node which has the highest number of neighbors inside its inclusion circle. In the first stage, the mobile charger follows the optimal path to reach the next discharged CH, then it stops (the mobile charger can "decide" to stop or move anytime a time interval is passed) and overcharges the CH. Then overcharged CHs sell their energy to their cluster members with no competition (the number of seller nodes is significantly larger than the number of buyer nodes thanks

to the first stage). It is not CHs which offer the energy, but the nodes that broadcast a message (composed of an ID and the amount of energy needed) and wait for one or more CHs to give them energy. CHs will serve near nodes first. Simulations produced with Omnet++ and Castalia show that EH-WSN protocol works best with greater nodes inclusion circle radius than that of smaller ones.

The authors in [18] use a cross-layer cooperative TDMA scheme instead of a classical one to optimise the CHs relaying performance. The CH role is alternated between the nodes using duty cycling as a function of their individual energy-harvesting capabilities. This protocol defines the optimal number of clusters according to the intensity of the energy source (which is solar energy in the paper). The protocol is based on LEACH. The CH choice is based on a probability function that uses duty cycle mechanism where a node cannot become CH before n duty cycles are passed. This number is computed for each CH as the ratio of the CH required energy to its allocated energy is rounded off to the next integer value. The protocol can include cooperation (cooperative transmission protocol) ECO-LEACH or not (ENCO-LEACH). The cooperative transmission protocol makes use of the energy unconsumed in data transmission to relay undelivered packets from cluster members to CHs and also from CHs to the sink node.

5.2.4 Clustering Protocols with Machine Learning

Machine learning (ML) is a technique of the late 1950s for artificial intelligence (AI) and for the definition of computationally viable and robust algorithms. During the years, ML has been applied to different fields such as bioinformatics, speech recognition, spam detection, computer vision, fraud detection, and advertising networks. ML learning techniques have been used for many tasks like classification, regression, and density estimation.

From the point of view of [6] and [13], machine learning can be defined as

- The development of computer models for learning processes that provide solutions to the problem of knowledge acquisition and enhance the performance of developed systems
- The adoption of computational methods for improving machine performance by detecting and describing consistencies and patterns in training data

Applying ML to the field on WSN routing protocols is a process which has both pros and cons. Some of the ML algorithms best properties are their ability to automatically calibrate according to newly acquired knowledge, their generally low complexity, and their capability to uncover correlation between sensor data and improve sensor deployment for maximum data coverage. On the other hand, ML algorithms drawbacks lie in the high amount of computational power they need, which escalates when requiring more accuracy, and the large set of existing data and samples they require to achieve high generalization capabilities.

There are several ML techniques that can be applied to WSNs to perform clustering. These techniques try to improve node clustering and data aggregation mainly in two ways:

- Compress data locally at CHs by efficiently extracting similarity and dissimilarity (e.g., from faulty nodes) in different sensors' readings
- CHs election, where appropriate cluster head selection will significantly reduce energy consumption and enhance the network's lifetime.

Some classic ML approaches have been investigated in the past to check their suitability for WSNs clustering and data aggregation [2]. Clustering can be performed based on (1) neural networks, (2) decision trees, and (3) role-free CHs selection, while data can be aggregated using (1) self-organizing map (SOM), (2) learning vector quantization, (3) principal component analysis, (4) k-means algorithm, and (5) decentralized learning. However, very few clustering protocols have been implemented through the approaches listed above. Moreover, most of them do not compare their results with well-known adaptive clustering protocols such as HEED, ER-HEED, and LEACH. One ML-based protocol which makes a comparison with other existing protocols is LEACH GA[14]. LEACH GA is a genetic algorithm based on LEACH [8]. LEACH GA modifies the LEACH algorithm, adding a preparation phase only once before the set-up phase of the first round. Initially, nodes perform cluster head selection, then they send their messages stating if they candidate to become cluster head, their node IDs, and their geographical positions to the base station. At that point, the base station uses data received from nodes to determine the optimal probability p_{opt} by performing GA operations, then it broadcasts this value to all nodes. The following set-up and steady-state phases are performed in every round and are the same as LEACH. Recently, [21] has proposed a comparison of LEACH GA performance over LEACH and LEACH-C using MATLAB simulation tool. In the simulations, nodes are randomly distributed in an area of 100 m $\times$ 100 m with the base station located at the centre point (50, 50). According to the simulation results, LEACH-GA performs better when compared to LEACH and LEACH-C under different initial energy and probability thresholds. In particular, LEACH-GA increases the network lifetime on the average of 54% and 47% over LEACH and LEACH-C. However, simulation results do not take into account novel clustering protocols proposed after LEACH and its variations [17, 23, 25], which are proved to perform better under various situations.

Table 5.1 shows a categorization of the clustering protocols that are described in this section. Clutering protocols are categorized by considering several attributes.

5.3 REECHD Clustering Protocol

In this section, we describe the leader election novelty introduced by REECHD as well as the REECHD cluster formation and rotation algorithms.

Table 5.1 Comparison of well-known clustering protocols for WSNs

Protocols	Node Deployment	Heterogeneous	Homogeneous	Clustering Method	Distributed(D)	Centralized(C)	Equal Sized Clusters	Unequal Sized Clusters	Rotation	Location Awareness	Harvesting	Machine Learning	Probability	No Probability
LEACH[8]	Random		✓		D		✓			N			✓	
SEP[22]	Random	✓			D		✓			N			✓	
HEED[25]	Random		✓		D		✓			N			✓	
ERHEED[23]	Random		✓		D		✓		✓	N			✓	
UHEED[7]	Random		✓		D			✓		N			✓	
LEACH-GA[14]	Random		✓		D					N		✓	✓	
RUHEED[1]	Random		✓		D			✓	✓	N			✓	
DEEC[19]	Random	✓			D		✓			N			✓	
DWEHC[4]	Random		✓		D		✓			N			✓	
IEEUC[15]	Random		✓		D			✓		N			✓	
DEBC[5]	Random	✓			D		✓			N			✓	
DCLB[10]	Random		✓		C		✓			Y				✓
EEHC[11]	Random		✓		D		✓			N			✓	
DCHE[12]	Uniform	✓			D					N			✓	
FMUC[16]	Random	✓			D			✓		N			✓	
REECHD[17]	Random	✓			D		✓		✓	N			✓	
EHWSN[3]	Random				D		✓			N	✓		✓	

5.3.1 REECHD Leader Election Probability

REECHD is a clustering algorithm for heterogeneous WSNs that produces clusters of equal size. REECHD reduces the amount of leader election phases by using rotation. This decreases the amount of overhead messages, thus prolonging the WSN lifetime. The novelty of REECHD is in its probabilistic election process and the use of the intra-traffic limit.

$$CH_{prob} = max\left(\frac{C_{prob}}{K}\left(\frac{E_{residual}}{E_{max}} + IW^{-1}\right), P_{min}\right) \tag{5.1}$$

Eq. (5.1) defines the leader election probability CH_{prob}. This is the probability a node has of becoming CH when a new leader election phase takes place. In the following, we summarise the components of the probability CH_{prob}:

- C_{prob} is a predefined initial probability (e.g., 5%) that sets the initial percentage of cluster heads among all WSN nodes. This is used to limit the initial CH announcements, and does not impact on the final clustering.
- P_{min} defines a minimum probability value that CH_{prob} must have. This is selected to be inversely proportional to E_{max} (e.g., 10^{-4}) so that the algorithm terminates in $N_{iter} = O(1)$ iterations [25].
- $E_{residual}$ defines the residual energy of the node while E_{max} defines the maximum energy of the node (it defines a fully charged battery).
- The constant k is chosen in order to ensure the probability CH_{prob} is always between 0 and 1. In our case, k is equal to two.
- The positive quantity IW is the node induced work rate. This estimates the energy the node spends and induces on other nodes when it plays the CH role. Thus, a node with higher induced work should have less probability to be elected.

$$IW = \frac{D_{Rmax}}{D_R} \tag{5.2}$$

We estimate the node induced work IW by using Eq. (5.2) where D_R is the average transmission rate of the node and D_{Rmax} is rate of the node with the highest transmission rate of the WSN. This equation assigns a lower induced work to nodes with higher transmission rate, that is, nodes with higher rate should have a higher probability of becoming cluster head. In fact, when a node with a high rate is not selected as CH (it is a member node), more intra-traffic communication is generated. On the other hand, when a node n with high rate is selected as CH, the cluster will be not overloaded with messages from n. Nodes with lower transmission rate should have less probability of becoming cluster head since they generate little intra-traffic communication inside the cluster. It is worth mentioning that the node induced work could be further refined by considering further sources of energy consumption such as the energy the node spends to run the sensor hardware or the intertraffic the node can potentially generate.

We emphasize that election probability of Eq. (5.1) combines energy and induced work together. More precisely, nodes with higher energy and higher transmission rate should have more probability of becoming cluster head.

5.3.2 REECHD Intra-Traffic Rate Limit

The intra-traffic rate limit (ITRL) defines a rate that each CH must use during cluster formation. More precisely, each CH must ensure that the sum of transmission rates of its member nodes never exceeds ITRL. This is defined by the following equation:

$$\sum_{i=1}^{|member\_set|} sending\_rate(n_i) < ITRL$$

where $member\_set$ contains all member nodes that compose the cluster, $|member\_set|$ is the cardinality of $member\_set$, n_i is a node that belongs to $member\_set$, and $sending\_rate(n_i)$ is the transmission rate of the node n_i. We can define a lower and upper bound for the ITRL:

$$\left[0, \sum_{i=1}^{|WSN\_nodes|} sending\_rate(n_i)\right]$$

where $|WSN\_nodes|$ is the number of WSN nodes. We have a flat routing (i.e., each node of the WSN is cluster head and has no member nodes) when the ITRL is equal to zero. We can have a single cluster that contains all nodes when the ITRL is the sum of all node sending rates.

The ITRL is a quite useful means to control the number of clusters inside the WSN. Low ITRL values can generate more clusters than high ITRL values. More clusters can lead to lower intra-traffic communication at the cost of higher intertraffic communication. As we see in Sect. 5.4.2, the choice of the ITRL depends on the aggregation rate. We emphasize that the use of the ITRL is also useful when nodes are not uniformly deployed since a denser area can get a higher number of clusters. This allows the balance of the intra-cluster communication, thus balancing the energy consumption and prolonging the WSN lifetime.

5.3.3 REECHD Algorithm

REECHD is a clustering algorithm for heterogeneous WSNs that produces clusters of equal size and uses rotation in order to prolong the WSN lifetime. Member nodes of a cluster can directly communicate with their CH. This is referred to as 1-hop communication [25]. REECHD includes the following four main phases: (1) cluster head election, (2) cluster formation and iteration, (3) rotation, and (4) network operation. Cluster head election, formation and iteration are performed at the beginning and anytime a node dies. When no node dies, the rotation and network operation phases are performed in alternation. All REECHD phases are described in detail in the following.

5.3.4 REECHD Cluster Head Election

This phase takes place at the beginning and anytime a node dies. In this phase, each node can become cluster head according to the probability that is defined by Eq. (5.1) of Sect. 5.3.1.

Figure 5.1 outlines the cluster head election algorithm that a node B executes. An $initialization$ procedure is used to set some variables and is executed only

```
1  Initialisation ()
2    iterations = 0
3    max_iterations = n
4    set_parameter(ITRL)
5
6  Cluster_head_election()
7    cluster_head_set = tentative_CH_set = ∅
8    neighbours = all neighbour nodes which are alive
9    CH_prob = max(0.5 * C_prob * (E_residual/E_max + D_R/DR_max), P_min)
10   iterations = iterations + 1
11
12   Repeat
13     if (tentative_CH_set ≠ ∅)
14       CH = least_cost(neighbours)
15       if (CH = myself)
16         if (CH_prob = 1)
17           broadcast_election_msg(neighbours)
18           add_to(final_CH_set)
19         else
20           broadcast_tentative_msg(neighbours)
21           add_to(tentative_CH_set)
22     else if (CH_prob = 1)
23       broadcast_election_msg(neighbours)
24       add_to(final_CH_set)
25     else if (CH_prob >= random(0,1)) f
26       broadcast_tentative_msg(neighbours)
27       add_to(tentative_CH_set)
28     previous_prob = CH_prob
29     CH_prob = min(CH_prob * 2, 1)
30   Until previous_prob = 1
```

Fig. 5.1 REECHD CH election at node B

once before the $cluster\_head\_election$ procedure. The $initialization$ procedure sets the following variables: (1) $iterations$, (2) $cluster\_head\_election$, and (3) $max\_iterations$. The variable $iterations$ stores the number of times the member node B tried to perform the $cluster\_head\_election$ procedure. B needs to redo the cluster election when it is unable to join any CHs. This happens when CHs in the neighborhood of B reached the intra-traffic limit. In this case, B repeats the $cluster\_head\_election$ procedure. The variable $max\_iterations = n$ defines the maximum amount of times B repeats the $cluster\_head\_election$ procedure before it elects itself as cluster head. The procedure $set\_parameter(ITRL)$ sets the intra-traffic limit ITRL to a predefined constant value.

The $cluster\_head\_election$ procedure initializes the set $cluster\_head\_set$ and $tentative\_CH\_set$ to empty. The set $cluster\_head\_set$ contains all nodes in the neighborhood of B that proposed as CH. The set $tentative\_CH\_set$ contains all nodes that attempt to become CH but their election is not finalized yet. The variable $neighbors$ contains all nodes that are within the radius range of B and are alive. The $cluster\_head\_election$ procedure sets its probability of becoming CH (line 9 of the algorithm in Fig. 5.1), increases the $iterations$ counter, and starts the $repeat$ loop (line 12–30).

```
1  Cluster_formation()
2
3    if(myself ∈ final_CH_set)
4      broadcast_election_msg(neighbours)
5      member_set = member_selection(join_set, ITRL)
6      send(member_set, join)
7      send(join_set − member_set, unjoin)
8    else
9      while(final_CH_set ≠ ∅)
10       CH = least_cost(final_CH_set)
11       final_CH_set = final_CH_set − CH
12       join(CH)
13       if(join_msg_received)
14         return
15     end while
16
17     if(iterations ≤ max_iterations)
18       Cluster_head_election()
19     else
20       broadcast_election_msg(neighbours)
```

Fig. 5.2 REECHD cluster formation at node B

B selects the least-cost CH[2] from $tentative\_CH\_set$ when this set is not empty (line 13). When the selected CH is the node itself, it can broadcast either an election message or a tentative message. The election message is broadcast when CH_{prob} has reached 1 while the tentative message when CH_{prob} is less than 1. When no nodes proposed as CH and CH_{prob} are equal to one (lines 28–30), B proposes itself as cluster head. When no nodes proposed as CH and CH_{prob} are less than one (lines 25–27), B decides whether or not to become tentative CH by considering its probability CH_{prob}; at the end of each repeat cycle, the probability is doubled (line 29). This ensures that REECHD terminates in $\mathcal{O}(1)$ number of steps.

We emphasize that the $least\_cost$ function is used to break the tie and select a cluster head when two tentative nodes lie within the same communication range. This behavior prevents two nodes within the same transmission range from becoming CHs, that is, REECHD creates a set of disjoint clusters.

5.3.4.1 Cluster Formation and Iteration

In Fig. 5.2, we detail the cluster formation algorithm that is executed by the node B. A node playing the CH role executes the $then$ branch of the if control structure (lines 3–8) while a member node executes the $else$ branch (lines 9–30):

- B **playing the CH role.** B sends a broadcast election message. A member node can reply with a join message when B is the least-cost CH it can reach. The node B keeps all join requests in the set $join\_set$. This is used together with the $ITRL$ in order to call the $member\_selection$ procedure. This returns a node set

[2]We have experimented various cost functions such as selecting the closest CH or selecting the CH which has the largest member-set. A node selects the closest CH.

$member\_set$ that contains all nodes B selected as cluster members. We recall that the intra-traffic communication generated by the $member\_set$ nodes must be less than the $ITRL$ (see Sect. 5.3.2 for details). Various member selection strategies can be adopted. For instance, a random pick can be performed until $ITRL$ is reached (this is the strategy we used in the presented simulation results). The member nodes can be selected starting from the highest rate node until the $ITRL$ is reached. The member nodes can be selected so that the total rate is less than or equal to $ITRL$ and is as large as possible. B sends an $unjoin$ message to all nodes that are not included in the $member\_set$ (line 8).

- B **playing the member role.** B tries to join one after another all reachable CHs (from the $least\_cost$ to the $worst\_cost$). B will join the first CH that replies with a join message (lines 10–16). After joining, the nodes terminate the cluster formation procedure.

B iterates the cluster head election when it is not CH and was not able to join any cluster. The cluster head election can be repeated a maximum number of times (i.e., $max\_iterations$).

5.3.4.2 Cluster Rotation

The current CH designates the next CH directly by using Eq. (5.1).[3] More precisely, the current CH calculates the probability CH_{prob} of each member node and chooses the one with the highest CH_{prob} as the next CH. The new CH is elected without the need for performing any election protocol. We refer to as *operation phase* the one where member nodes send data to their CHs. These cooperate in order to report data to the base station.

5.4 Comparing the State-of-the-Art Clustering Protocols

In this section, we compare REECHD with various clustering protocols. Comparison is performed by simulating all protocols by using the same WSN features, and the same network and communication models.

5.4.1 Network Model

In our network model, nodes are not mobile and are uniformly distributed in a two-dimensional area. We have energy heterogeneity since nodes can have

[3]It is assumed that each data packet received by the CH contains energy information of its member nodes. This is needed in order to calculate CH_{prob}.

different initial energy. Nodes have different data transmission rates within a defined maximum and minimum rate. Nodes have the same processing and aggregation capabilities. Nodes have unique IDs and nodes can transmit at various power levels depending on the distance of the receiver.

The BS is not mobile, has no energy constraints, and is located outside the WSN area. The BS has more communication and processing capabilities with respect to normal sensor nodes. Each CH can aggregate the intra-traffic data in order to reduce the amount of bits that are forwarded to the BS. Intertraffic is not aggregated, that is, a CH forwards (toward the BS) messages received from other CHs with no aggregation.

We use a network operation model that has been adopted in quite a few papers such as LEACH, HEED, RUHEED, FMUC, and so on. We recall that a clustering protocol usually includes the following phases: (1) cluster election and formation; (2) network operation phase; (3) rotation (if any); (4) re-election and formation. During the data network operation phase, a TDMA is composed of the following two activities: (1) each member node sends one variable size message to its cluster head, and (2) all CHs' data reaches the BS. In other words, a TDMA starts from the collection of data from the member nodes and ends when all the data reaches the base station. A round is composed of multiple TDMAs.

We define two types of WSN nodes that are homogeneous and heterogeneous. Homogeneous nodes have an initial energy of 0.5 J and send messages of 1000 bits. Heterogeneous nodes have an initial energy that falls within the interval [0.2, 0.8] J and send messages of a size that falls within the interval [100, 1900] bits. We define the heterogeneity level as the ratio between the number of the heterogeneous nodes and all WSN nodes. For instance, a heterogeneous level of 20% means that 20% of the WSN nodes are heterogeneous. Table 5.2 summarizes all network parameters.

For simulation purposes, we define the aggregation rate (AR) which is a number between 0 and 1. This is used to calculate the intertraffic message size that is generated by the CH as follows:

$$MIN\left(\sum_{i=1}^{|cluster\_set|} sending\_rate(n_i) * (1 - AR), min\_msg\_size\right) \tag{5.3}$$

where $cluster\_set$ is the set of nodes that compose a cluster (including the CH), $|cluster\_set|$ is its cardinality, n_i is a node that belongs to $cluster\_set$, $sending\_rate(n_i)$ is the transmission rate of the node n_i, AR is the aggregation rate, and min_msg_size is a constant that denotes the minimum size of message that is forwarded by a CH. When the aggregation rate (AR) is zero, a CH packs all messages received by the members (during a TDMA) and forwards them to the next hope. In this case, no aggregation takes place. When the aggregation rate is one, the CH aggregates all messages received by the members in a TDMA by producing a message with a minimum size. We set this minimum size to 100, that is, the minimum rate of a node. A more refined min_msg_size value could consider

Table 5.2 Simulation parameters

Simulation parameters	
Parameters	Values
Network grid	From (0, 0) to (100, 100)
BS	(175, 50)
E_{elec}	50 nJ/bit
E_{fs}	10 pJ/bit/m^2
E_{mp}	0.0013 pJ/bit/m^4
R_0	30, 35, 40, 45, and 50 m
Control parameter UHEED	$c = 0.5$
Number of nodes	200

the node with the smaller (greater) rate inside the cluster or the average rate of the cluster.

The adopted radio model utilizes free space and multi-path channel model. The assumed network grid size is 100 by 100 m and BS is placed at position (175, 50). The simulation parameters are outlined in Table 5.2. Transceiver circuitry of a sensor node consumes $E_{elec} = 50$ nJ/bit. Sensor node amplification energy E_a depends on the distance between the sender and the receiver. When $d < d_0 = 75$ m, E_a becomes $E_{fs} = 10$ pJ/bit/m^2 and when $d \geq d_0 = 75$ m, E_a reduces to $E_{mf} = 0.0013$ pJ/bit/m^4. The transmission and reception energy consumed in sending and receiving a data packet k (bits) over distance d can be be computed [9] as

$$E_{Tx} = k(E_{elec} + E_a d^n) \tag{5.4}$$

$$E_{Rx} = k(E_{elec}) \tag{5.5}$$

Table 5.2 summarizes all network parameters.

5.4.2 Simulation Results and Analysis

We simulated REECHD, UHEED, HEED, ERHEED, and FMUC on a WSN composed of 200 nodes and with a grid size of 100 by 100 m. The heterogeneity level varied from 20% to 80% with a step of 20, the node competition radius R_0 from 30 to 50 m, and we set the aggregation to 50%. For REECHD, we also set ITRL percentage by multiplying the maximum ITRL value by a number between zero and one. We have used an ITRL percentage of 0.1, 0.2, 0.5, and 0.8. Each simulation is an average of hundred runs.

Figures 5.3, 5.4, 5.5 and 5.6 shows the lifetime of the network for different heterogeneity levels until first node dies (FND) for REECHD, ERHEED, FMUC,

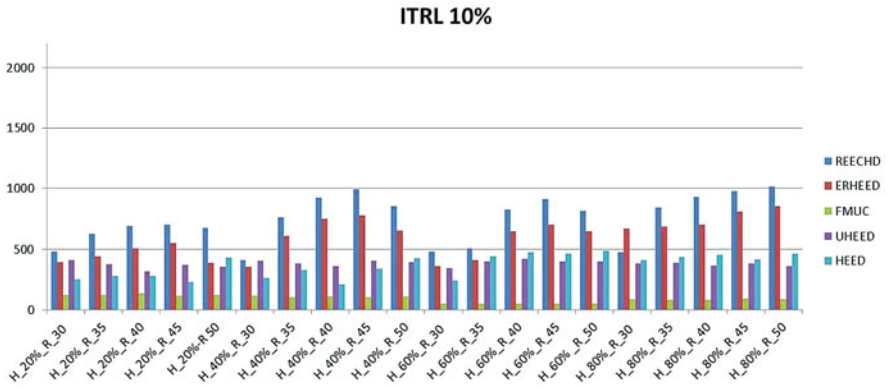

Fig. 5.3 Lifetime measure = FND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 10%

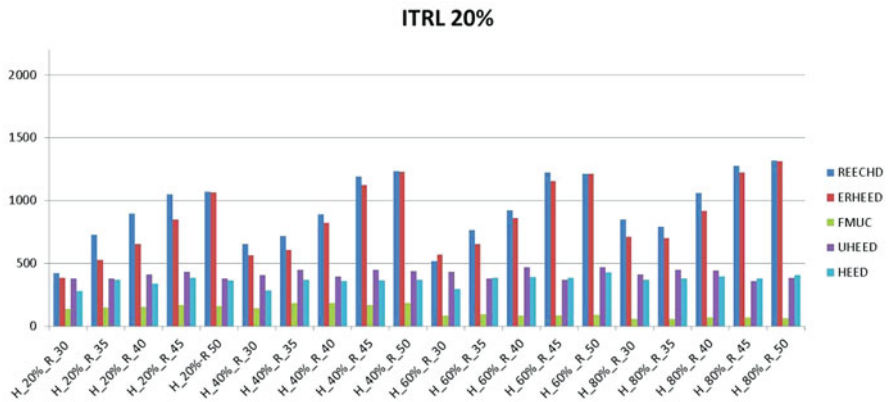

Fig. 5.4 Lifetime measure = FND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 20%

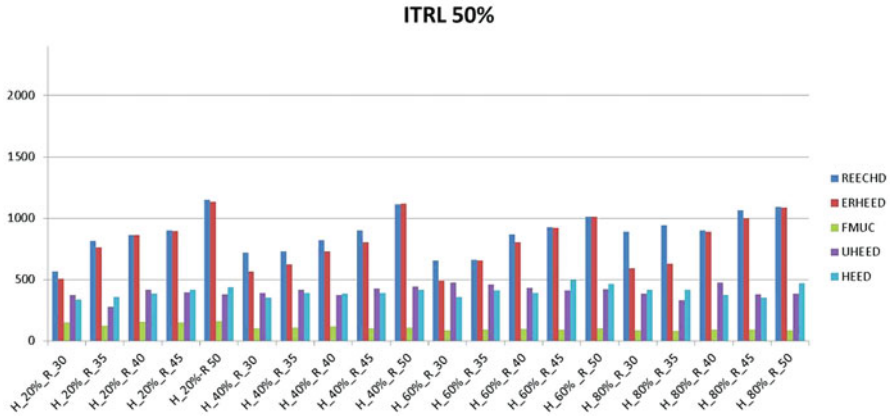

Fig. 5.5 Lifetime measure = FND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 50%

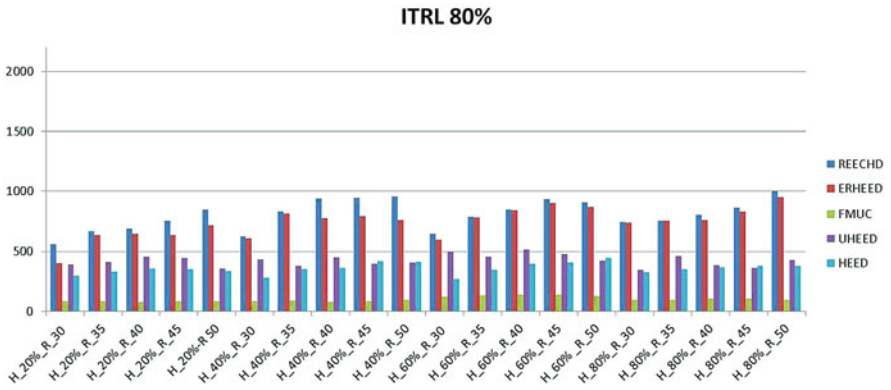

Fig. 5.6 Lifetime measure = FND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 80%

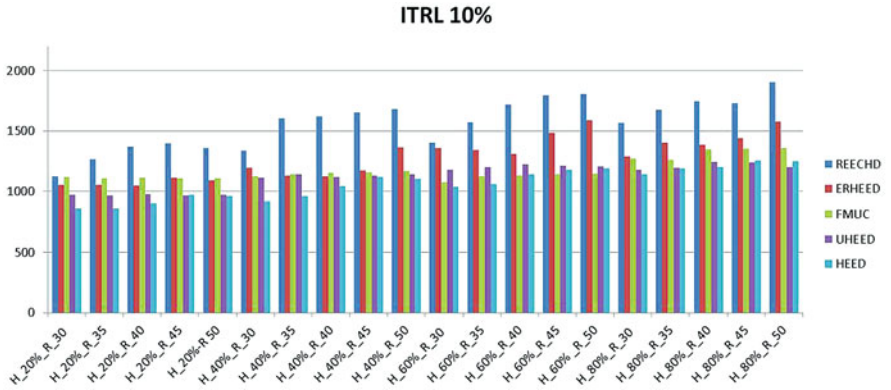

Fig. 5.7 Lifetime measure = HND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 10%

UHEED, and HEED protocols. These are run for an increasing heterogeneity level, and radius from 30 m to 50 m.

Figures 5.7, 5.8, 5.9 and 5.10 shows the lifetime of the network for different heterogeneity levels until half of the nodes die (HND) for REECHD, ERHEED, FMUC, UHEED and HEED protocols. These are run for an increasing heterogeneity level, and radius from 30 m to 50 m.

The network lifetime for all the protocols increases as the heterogeneity level approaches 80%. For each heterogeneity level, we show the network lifetime for different ITRL percentages. The most energy-efficient results are achieved when the ITRL percentage is equal to 0.5 for HND and 0.2 for FND.

By looking at the figures and we can observe that REECHD outperforms all other clustering protocols for HND lifetime measure.

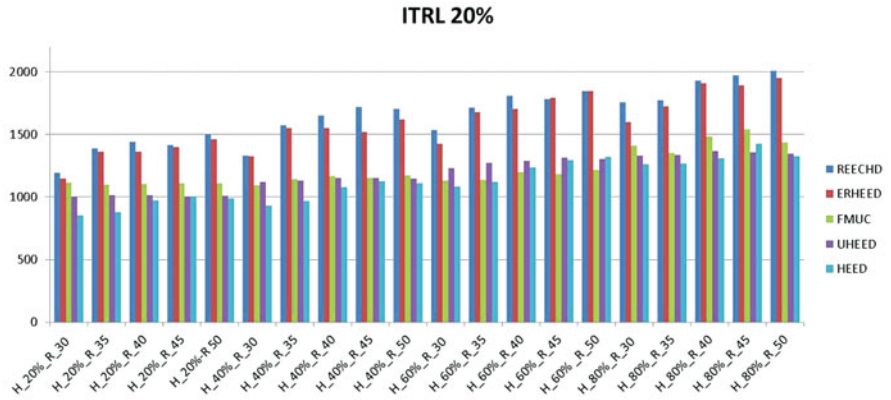

Fig. 5.8 Lifetime measure = HND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii= 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 20%

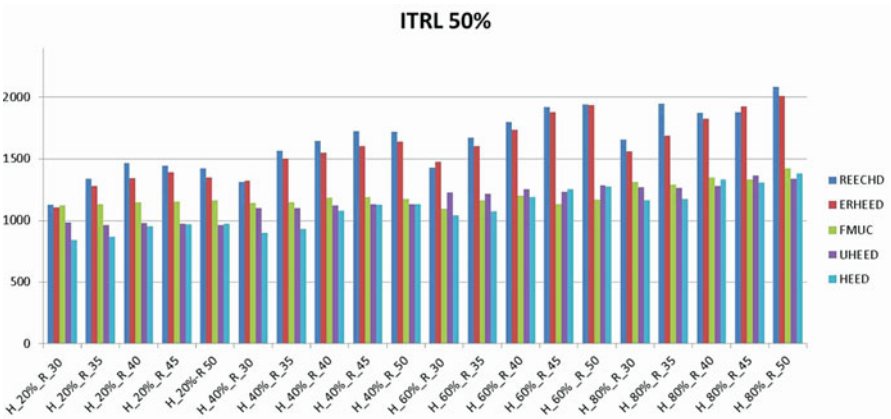

Fig. 5.9 Lifetime measure = HND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 50%

Is it worth mentioning that REECHD outperforms FMUC [16], a protocol that has been conceived in the heterogeneous WSN context. We used the same simulation settings of FMUC, which outperforms the EEUC and DEBUC protocols. Thus, REECHD outperforms both EEUC and DEBUC.

5.5 Conclusions

In this chapter, we reviewed the state of the art of the most prominent energy-efficient clustering for WSNs. We reviewed algorithms for heterogeneous and

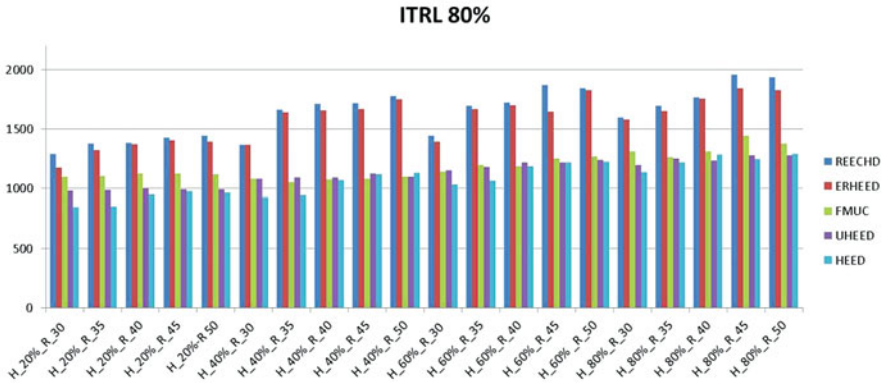

Fig. 5.10 Lifetime measure = HND; aggregation = 50%; heterogeneity level = 20, 40, 60, and 80%; radii = 30, 35, 40, 45, and 50 m; intra-traffic rate limit = 80%

homogeneous WSNs, clustering protocols that consider the introduction of harvesting into WSNs, and clustering protocols that are based on machine learning techniques.

We proposed the REECHD protocol for heterogenous WSNs. When selecting new CHs, REECHD considers not only the residual energy of the devices but also their induced work. This is estimated by using the node transmission rate. REECHD also introduces the concept of intra-traffic rate limit (ITRL). This defines a limit on the intra-traffic communication that all WSN clusters must comply with. REECHD is more suitable for cluster heterogeneous networks in which the communication rates of the devices are heterogeneous.

REECHD is more energy efficient when compared with well-known clustering protocols for homogeneous WSNs that are HEED, UHEED, and ERHEED. REECHD also outperforms various clustering protocols that have been conceived in the heterogeneous WSN context that are FMUC, EEUC, and DEBUC. In future work, we plan to implement a variation of REECHD which uses unequal-size clustering. We plan to experiment various member selection strategies for cluster formation such as Knapsack. We plan to study heuristics to find the best ITRL under various WSN settings.

References

1. N. Aierken, R. Gagliardi, L. Mostarda, Z. Ullah, Ruheed-rotated unequal clustering algorithm for wireless sensor networks, in *29th IEEE International Conference on Advanced Information Networking and Applications Workshops, AINA 2015 Workshops*, Gwangju, 24–27 March 2015, pp. 170–174
2. M.A. Alsheikh, S. Lin, D. Niyato, H.P. Tan, Machine learning in wireless sensor networks: algorithms, strategies, and applications. IEEE Commun. Surv. Tutorials **16**(4), 1996–2018 (2014)

3. M.S. Bahbahani, E. Alsusa, A cooperative clustering protocol with duty cycling for energy harvesting enabled wireless sensor networks. IEEE Trans. Wirel. Commun. **17**(1), 101–111 (2018)
4. P. Ding, J. Holliday, A. Celik, Distributed energy-efficient hierarchical clustering for wireless sensor networks, in *Proceedings of the First IEEE International Conference on Distributed Computing in Sensor Systems, DCOSS'05* (Springer, Berlin/Heidelberg, 2005), pp. 322–339
5. C. Duan, H. Fan, A distributed energy balance clustering protocol for heterogeneous wireless sensor networks, in *2007 International Conference on Wireless Communications, Networking and Mobile Computing* (2007), pp. 2469–2473
6. A.H. Duffy, The "what" and "how" of learning in design. IEEE Intell. Syst. **12**, 71–76 (1997)
7. E. Ever, R. Luchmun, L. Mostarda, A. Navarra, P. Shah, Uheed - an unequal clustering algorithm for wireless sensor networks, in *SENSORNETS* (2012)
8. W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd Annual Hawaii International Conference on System Sciences*, vol. 2 (2000), 10 pp
9. W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in *Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8 - Volume 8, HICSS '00* (IEEE Computer Society, Washington, DC, 2000), p. 8020
10. F. Ishmanov, S.W. Kim, Distributed clustering algorithm with load balancing in wireless sensor network, in *2009 WRI World Congress on Computer Science and Information Engineering*, vol. 1 (2009), pp. 19–23
11. D. Kumar, T.C. Aseri, R.B. Patel, Eehc: energy efficient heterogeneous clustered scheme for wireless sensor networks. Comput. Commun. **32**(4), 662–667 (2009)
12. D. Kumar, T.C Aseri, R. Patel, Distributed cluster head election (DCHE) scheme for improving lifetime of heterogeneous sensor networks. Tamkang J. Sci. Eng. **13**, 337–348 (2010)
13. P. Langley, H.A. Simon, Applications of machine learning and rule induction. Commun. ACM **38**(11), 54–64 (1995)
14. J.-L. Liu, C. Ravishankar, LEACH-GA: genetic algorithm-based energy-efficient adaptive clustering protocol for wireless sensor networks. Int. J. Mach. Learn. Comput. **1**, 79–85 (2011)
15. P. Liu, T.l. Huang, X.Y. Zhou, G.X. Wu, An improved energy efficient unequal clustering algorithm of wireless sensor network, in *2010 International Conference on Intelligent Computing and Integrated Systems* (2010), pp. 930–933
16. T. Liu, J. Peng, J. Yang, G. Chen, W. Xu, Avoidance of energy hole problem based on feedback mechanism for heterogeneous sensor networks. Int. J. Distrib. Sensor Netw. **13**(6), 1550147717713625 (2017)
17. M. Micheletti, L. Mostarda, A. Piermarteri, Rotating energy efficient clustering for heterogeneous devices (REECHD), in *32nd IEEE International Conference on. Advanced Information Networking and Applications (IEEE AINA 2018)*, Pedagogical University of Cracow, 16–18 May 2018
18. C. Moraes, D. Har, Charging distributed sensor nodes exploiting clustering and energy trading. IEEE Sensors J. **17**(2), 546–555 (2017)
19. L. Qing, Q. Zhu, M. Wang, Design of a distributed energy-efficient clustering algorithm for heterogeneous wireless sensor networks. Comput. Commun. **29**(12), 2230–2237 (2006)
20. F.K. Shaikh, S. Zeadally, Energy harvesting in wireless sensor networks: a comprehensive review. Renew. Sustain. Energy Rev. **55**, 1041–1054 (2016)
21. P. Sivakumar, M. Radhika, Performance analysis of LEACH-GA over leach and LEACH-C in WSN. Proc. Comput. Sci. **125**, 248–256 (2018); *The 6th International Conference on Smart Computing and Communications*
22. G. Smaragdakis, I. Matta, A. Bestavros, SEP: a stable election protocol for clustered heterogeneous wireless sensor networks, in *Second International Workshop on Sensor and Actor Network Protocols and Applications (SANPA 2004)*, Boston, MA (2004)

23. Z. Ullah, L. Mostarda, R. Gagliardi, D. Cacciagrano, F. Corradini, A comparison of heed based clustering algorithms – introducing er-heed, in *2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)* (2016), pp. 339–345
24. M. Yarvis, N. Kushalnagar, H. Singh, A. Rangarajan, Y. Liu, S. Singh, Exploiting heterogeneity in sensor networks, in *Proceedings IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies*, vol. 2 (2005), pp. 878–890
25. O. Younis, S. Fahmy, Heed: A hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks. IEEE Trans. Mob. Comput. **3**(4), 366–379 (2004)

Chapter 6
Toward Optimum Topology Protocol in Health Monitoring

Mohammad E. Haque and Mohammad A. Hannan

6.1 Background

Structural health monitoring (SHM) is an active area of research devoted to systems that can autonomously and proactively assess the structural damage of bridges, buildings, and aerospace vehicles. Here, damage is defined as the changes of the material and/or geometric properties of these systems [1]. The process of implementing a damage identification strategy for civil, aerospace, and mechanical infrastructure is referred to as SHM. Nowadays, wireless sensor networks (WSNs) are a more popular technique to detect the damage such as temperature measurement, vibration, and several other parameters [2]. Large civil structures such as buildings are the backbone of our society. However, it is critical in daily operation. The damage of civil infrastructural health occurs enormously due to catastrophic events like earthquakes, flooding, and terrorist attacks. The typical assessment of such type of structure is costly due to inaccessible frequent monitoring, inaccurate position of instrumentation, and specific sensor data collection. However, an automated network computer system could automatically assess structural damage before the entry system damage occurs [3, 4]. The general purpose of SHM includes hazard mitigation, improvement of safety and reliability of the structural system, sustainability, and life cycle cost reduction. Traditionally, the data collection system

M. E. Haque (✉)
Department of Electrical and Electronic Engineering, Z.H.Sikder University of Science and Technology, Kartikpur, Bangladesh

M. A. Hannan
Department of Electrical Power Engineering, College of Engineering, Universiti Tenaga Nasional, Kajang, Selangor Darul Ehsan, Malaysia
e-mail: hannan@uniten.edu.my

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_6

of the SHM system is wire based, which can acquire sensor data periodically. These systems measure structural behavior and assess structural safety circumstances using various types of sensing devices for a certain damage diagnosis and prognosis method [5, 6].

Due to technological advancement in WSNs and micro-electromechanical-systems (MEMS), SHM system is eventually able to cover a large civil structural health with low-cost WSN [7–10]. A WSN consists of a large number of sensor nodes, which are densely deployed in building SHM application [11, 12]. In WSN, topology control protocols are an important requirement that are distributed with the number of deployed sensor nodes in the area of monitoring of interest. This is often achieved using the distributed sensor node and the topology construction algorithm [8]. Coverage area is an important factor in designing an efficient routing protocol for WSNs in building damage monitoring [13, 14]. In a large number of sensor nodes, in building structural health, coverage area becomes an important issue to collect sensor node information.

Figure 6.1 shows the overview of building, bridge, and tunnel monitoring using WSNs. The system consists of sensor nodes, wireless sensors, and central computers. In the building monitoring system, the sensors provide the response of building movement due to a strong earthquake or wind. During the event, shock absorbers and mass dampers can be used to reduce oscillations or damage. In the event of an earthquake, buildings could generate an alarm based on shockwave approach, so the other nearby buildings prepare themselves accordingly. In case of bridge monitoring, the mounted sensor nodes detect displacement, vibration, and temperature information and transmit it to the central computer across "hops" for further analysis. For any problems or defects such as loose cable or initial crack formation, a warning message can be sent to be aware about system damage. When WSN is mounted on the tunnel wall, it identifies the tunnel health information like crack, humidity, and temperature and sends the information to the analysis section to define the exact damage [15].

The topology control protocol performs well in collecting node information of the larger network [16, 17]. The way the sensor nodes are connected plays an important role in providing signals with required strengths. Out of many topologies, dense and sparse topologies were extensively used in monitoring system for collecting damage information. By increasing the size of the network, it is important to maintain the requirement of the optimal sensor placement in topology construction protocol. Based on this issue, dense topology sensor networks have the ability to self-organize, higher possible employed nodes, and redundant communication path. For this reason, dense networks have the ability to configure and reconfigure the routes itself when fail the nodes or add the nodes to the network [18]. In a sparse topology sensor network, a set of wireless network nodes are constructed to route the node information in a unicast manner. For unicast routing information, minimum energy consumption is needed than that of constant factor to make the path between the source and the destination. If there is a power-efficient path between two nodes

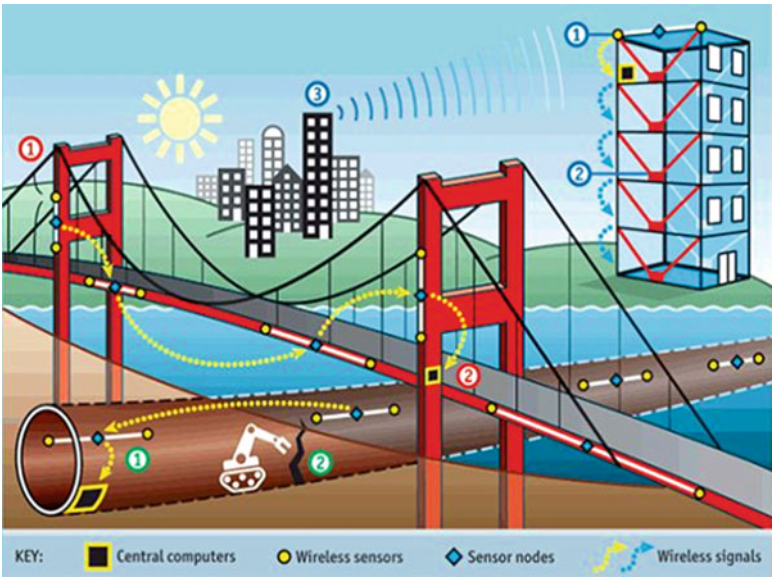

Fig. 6.1 Building, bridge, and tunnel monitoring using WSN. Source: [15]

then the network is said to be powerfully efficient [19]. However, it is still unclear which topology is better in obtaining health information in terms of network lifetime [20]. This work is an attempt to fill this gap.

In this study, a literature survey was conducted to update the knowledge on the works in this field and identify the problems related to the WSN monitoring system in SHM. Numerous SHM systems have been developed to measure the structural damage using WSN. This study has focused on the simulation-based extensive experiment of the dense and sparse topology sensor networks using Theory of Random Graph Approach (TRGA) for SHM application. The main aim of this study is to apply the idea of TRGA to the WSN monitoring system that produces the random graph using some random process. A random graph is a graph produced by some random procedure. Generally, the network nodes are fixed before employed in the network and the edges are generated using some random technique [21]. To save the sensor node energy and prolong the monitoring system lifetime, several experiments are conducted using topology construction protocol. The performance of the dense and sparse topology sensor networks has been measured to define a better topology sensor network in terms of network lifetime. The lifetime of the monitoring system has been studied using the topology construction protocol in terms of the total number of active nodes, the number of active nodes reachable from sink, covered area for communication, and covered area for sensing.

6.2 Motivation

With the need for building SHM, use of WSNs has increased in this application area. The health of the building structure needs to be monitored continuously using sensors placed at various locations on the structure [8]. In recent years, SHM is an important area of the continuously monitoring application that has received increasing research interest [22]. Various studies have shown that the costs for the monitoring system for structural health associated with disasters are much lower than economic losses due to structure defects. Deployment of the SHM system on an expected basis to monitor an early stage of structural damage is too complex. In the USA, the annual federal expenses for replacement of structural obstacles of bridges based on Nondestructive Evaluation Methods (NDEs) are approximately $10 billion [23]. The Federal Highway Administration (FHWA) reported that about 25% bridges of the country are functionally outdated. Nondestructive Evaluation Methods (NDEs) of the monitoring system include visual inspections that are inaccurate and mainly exaggerated by structural difficulties [24]. The present method of the structural monitoring system is difficult to be installed in the monitoring area where maintenance cost is very high. For example, sensor installation cost may vary for small-scale structure as $1000–5000 per sensor and for large-scale structure as $27,000 per sensor [25]. Therefore, there is a need for a monitoring system that could automatically monitor the building structural health. WSN is such a type of potential candidate that can automatically monitor the events in the building structural health.

A WSN is a possible approach for monitoring structural health due to its low cost compared to the other monitoring system like wired [16]. For SHM, the monitoring network should be efficient, lossless, and scalable to cover the large monitoring area of interest of structure. In this way, WSN performed well in SHM [26]. However, the existing WSNs do not completely satisfy the requirements of the SHM system due to several issues such as coverage area, lifetime, reliability, and interference. The goal of this study is the design and development of the large-area WSN with high-density deployed nodes to increase lifetime of the SHM system. To do so, dense and sparse topology sensor networks are simulated to cover the monitoring area of interest. Topology construction protocol is also used to reduce the topology of the sensor network nodes and save the node energy to prolong the network lifetime.

6.3 Problem Statement

SHM has been done using visual inspection procedure as well as wire-based data connection sensor system from many years ago. Most of the monitoring methods such as visual check [24] can only identify damages visible on the structure surface. The visual check method could not give any reliable data of the structure during the whole service life of structure; it only gives some information of structure during

a short period of examination. Besides this short period, the monitoring data is not sufficient for evaluating the structural condition with regard to whether its material and geometric properties change due to earthquakes, flooding, terrorist attacks, and others. Additionally, wire-based sensing system for SHM lost its popularity due to its several drawbacks. The wire-based sensors have high installation cost in large coverage area and are difficult to be installed in solitary area [14].

In recent years, WSN technology has been used to overcome the above problems. The most important limitations of the WSN for SHM are scalability, coverage area for large-scale coverage, reliability, and accuracy of the transmitted data. Various issues related to a networking layer are still open now. The most technical challenges of the SHM system are the lifetime of the WSN due to the topology creation and energy consumption [3, 27–30]. At the initial phase of the WSN, employed nodes discover each other using their maximum transmission power to make the first topology. After the initial phase, the second phase builds the reduced topology sensor network using topology construction protocols that save the employed nodes energy. Although the benefits of the topology construction algorithm are accepted widely, the problem of this complex system is it produces inaccurate results. Another problem of topology construction protocol is that if the design of the protocol is not energy efficient, then it may consume energy greater than save energy. Therefore, an efficient design of topology construction protocol is an important factor to save the node energy and extend the monitoring system lifetime [31]. Although many researchers have substituted wire-based sensor networks with WSNs for monitoring structural health, but until now very little attention has been paid on the monitoring coverage area or lifetime of the monitoring network. Thus, finding which topological sensor network can provide better coverage area with optimal results that minimizes energy but maximizes lifetime of the sensor nodes is a challenging issue. Therefore, it is quite necessary to conduct simulation-based experiments on different topologies to compare their performances of minimizing energy with area of coverage of interest and maximizing the monitoring system lifetime in SHM.

6.4 Objective of the Research

The aim of this study is to develop a monitoring system for structural heath using the WSN. The objectives of this study are as follows:

1. To develop the simulation model of SHM system using the dense and sparse topology wireless sensor networks.
2. To analyze the lifetime performance metrics of SHM system using the dense and sparse topology sensor networks under topology construction protocol.

6.5 Scope of the Research

This research focuses on the lifetime-related problem of the SHM. The dense and sparse topology sensor networks are proposed to address the lifetime-related issue in building SHM system. In the dense and sparse topology sensor networks, the topology construction protocol is defined with a number of high-density employed nodes that cover the area of monitoring of interest. Topology construction protocols are to be used to save the sensor node energy that extends the monitoring system lifetime. The simulation model of the sparse and dense topology sensor network for monitoring structural health (i.e., of buildings) is conducted using the Atarraya simulator. The simulation model is developed based on theoretical principal of the dense and sparse topology sensor networks using a graph theory approach. Also, the proposed model would be helpful to define optimum placement of sensors and provide the optimum lifetime topology construction protocol in SHM system.

6.6 Approach Description

This section presents the energy model that used to simulate the topology sensor network as the energy model. The analytical and simulation model of the dense and sparse topology sensor networks is performed using a theory of geometric random graphs approach.

6.6.1 Energy Model

It is important to include a model to drain the sensor node energy every time they perform in any action in order to perform the lifetime of the monitoring network. In this thesis, the energy model used to model the node energy consumption is based on Eq. 6.1, introduced in [32]. Mainly, the above model is based on the receiving and transmitting node data.

$$E_{\mathrm{Tx}} = E_{\mathrm{elec}} + E_{\mathrm{amp}}{}^{*} R_{\mathrm{comm}}^{2}{}^{*} \pi \tag{6.1}$$

$$E_{\mathrm{Rx}} = E_{\mathrm{elec}} \tag{6.2}$$

where E_{Tx} is the required transmit signal energy to transmit 1 bit and E_{Rx} is the receiver energy to receive the same number of bits like E_{Tx}. The energy of the electronics component of the radio signal is denoted by E_{elec} and amplifier radio energy is represented by E_{amp}. The second terms present the square area of the transmission range that is achieved by the radio signal. Due to the simplicity of the

Table 6.1 Energy model parameters mapping

Initial source energy	1 J
E_{elec}	50 nJ/bit
E_{amp}	10 pJ/bits/m^2

energy model, it is frequently used in the WSN. It is supposed that, under ideal conditions, the energy consumption is negligible. The energy model parameters values are summarized in Table 6.1.

6.6.2 Analytical Model of Dense and Sparse Topology Sensor Networks

This section presents the development model of the dense and sparse topology sensor networks that is presented in this chapter. A theory of Geometric Random Graphs approach (TGRA) is used to develop both dense and sparse topology sensor networks. The evaluations model and parameter definition of the dense and sparse topology scenarios for each set of experiments are presented in this section of this chapter.

6.6.2.1 Dense Topology Sensor Network Model

In dense topology sensor network, the maximum number of each sensor node connected to all other sensor nodes is near the total number of nodes used in the network. When each sensor node is directly connected to all other nodes, the network is called fully connected network. Figure 6.2 shows the example of dense topology sensor network for $N = 9$ number of sensor nodes. Figure 6.2 shows the fully connected dense topology sensor network scenario, since all the nodes in the network connect with each other directly. Figure 6.3 represents the functional block diagram of the dense topology sensor network. The deployment block contains the predefined set of wireless sensor nodes with a fixed area of monitoring of interest. In the deployment block, communication radius, sensing radius, number of sink, node energy distribution, the energy model, and communication model are defined. The Atarraya block contains the following options: (a) TC (transmission control) protocol; (b) TM (transmission maintenance) protocol; (c) sensor and data protocol; (d) routing (forwarding) protocol; (e) node mobility model. The visualization block visualizes the deployment area and nodes stats describe the state of the node. TC theoretical block defines topology of the monitoring network. To deploy the dense topology network, *CTR* of the dense network set by the *CTR* block. The report block converts the machine-readable data to the human-readable format.

Figure 6.4 shows the flow diagram of the dense topology sensor network. As illustrated in Fig. 6.4, the deployment area parameters define all parameters related

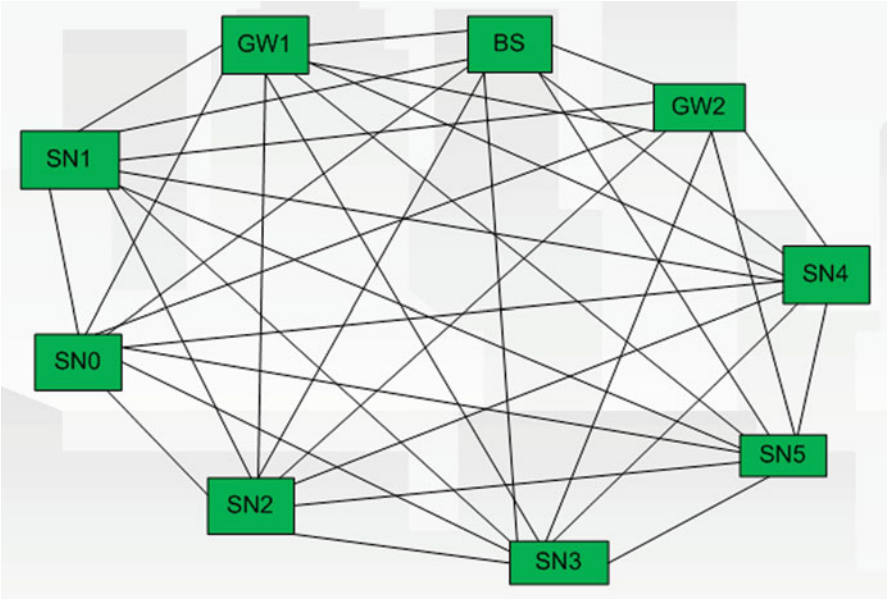

Fig. 6.2 Dense topology sensor network scenario

to deployment area. The *CTR* of the dense network is added to the deployment area network. The number of nodes in the deployment area determines the density of the dense network. After created deployment area, it changes to the Atarraya block. After that the selected protocol with other required parameters is added. All the protocols with required parameters are checked in a proper way. After running the live network using Java, the reports are transferred in a human-readable format.

TGRG approach [33] is used to provide an analytical solution to the communication range problem with high probability (w.h.p.) and produces a connected topology under some consideration. Consider, n is the number of sensor nodes that are uniformly distributed in a square area L. The nodes organization is uniformly distributed which means all the sensor nodes are equally distant in the monitoring area. The Penrose formula is used to determine the *critical transmission range* (*CTR*) value for the dense topology sensor network. The Penrose formula is defined as follows:

$$\mathrm{CTR}_{\mathrm{dense}} = \sqrt{\frac{\ln\ n + f(n)}{n\pi}} \tag{6.3}$$

where n is the total number of nodes and $f\ (n)$ is the function of n. The increasing value of n leads the incremental value of the function $f\ (n)$. The Penrose formula only applies to the dense topology sensor network. Table 6.2 shows the simulation setup for dense topology sensor network. The whole simulation result is obtained using an updated version of the Atarraya simulator [31]. The number of nodes

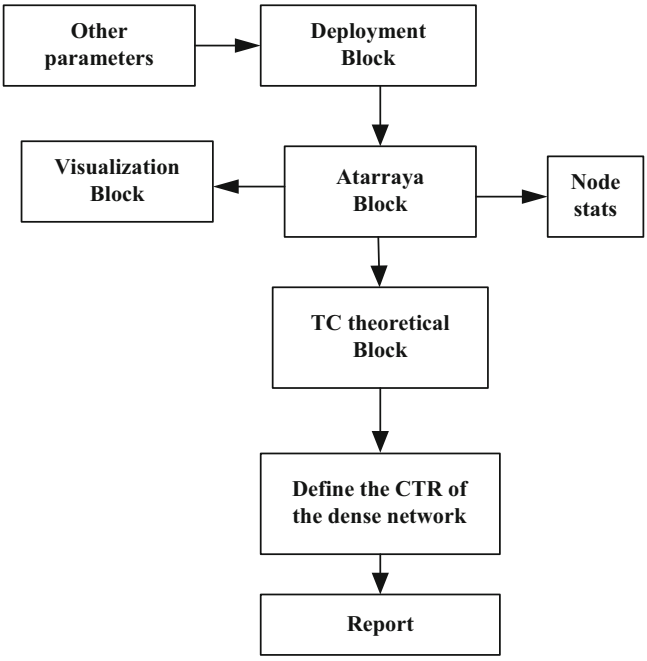

Fig. 6.3 Functional block diagram of the dense topology sensor network

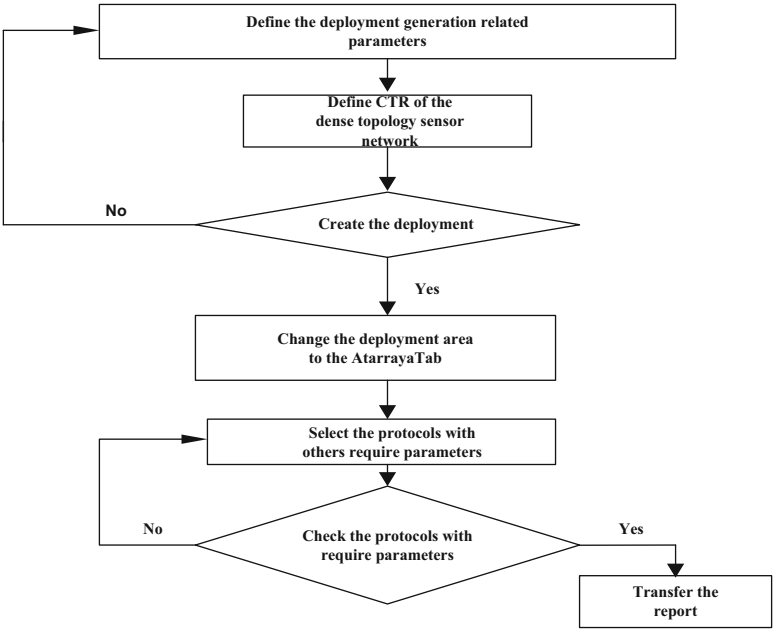

Fig. 6.4 Flow diagram of the dense topology sensor network

Table 6.2 Simulation setup for lifetime analysis of dense topology network

Parameters	Design values
Number of nodes	100
Communication radius	100 m
Sensing radius	20 m
Deployment area	600 × 600 m with central 300 × 300 m
Node energy distribution	1000 mJ (constant)
TC protocol	*A3, CDS-rule-K, EECDS, K-neigh, A3 coverage*
TM protocol	Non isolated sink (no topology maintenance)
Sensor and data protocol	Simple S and D protocol
Routing(forwarding protocol)	Simple forwarding
Performance metrics	Number of nodes alive, number of active nodes reachable from sink, covered area for communication, covered area for sensing

defines the density of the monitoring network. Initial *CTR* defines the initial value of the monitoring network. The *CTR* step defines the increasing value from the initial *CTR*. The number of topologies of the monitoring network is predefined using topology parameters. The area side of the monitoring network defines the deployment area of the dense network.

Table 6.2 shows the parameters setup for lifetime analysis purpose of the dense topology sensor network. The table contains the design parameter value of the dense topology sensor network and uses major parameters in the simulation case.

6.6.2.2 Sparse Topology Sensor Network Model

In sparse topology sensor network, the minimum number of links is connected compared with that of dense topology sensor network. This type of sensor network topology can be found to be more difficult to create network links between nodes. For example, Fig. 6.5 shows the sparse topology sensor network for $N = 9$ number of sensors nodes, in which a minimum number of links are seen to connect the sensor nodes with each other and with also the base station.

Figure 6.6 represents the functional block diagram of the sparse topology sensor network. The functional block diagram consists of 8 blocks, among those five are major and the remaining three are minor blocks. The visualization block, node stats, and other block are the minor blocks, and the remaining are considered as major blocks. Initially, the deployment block defines the parameters those related to deployment-related parameters models of the monitoring area by considering supportive parameters. The second block provides the same function as dense but works on sparse-based discipline. The visualization block visualizes the deployment area, and nodes stats describe the state of the node which is the minor block. Without these two minor blocks, the other block can generate the deployment area for monitoring purposes. *Transmission Control* (*TC*) theoretical block is the third block

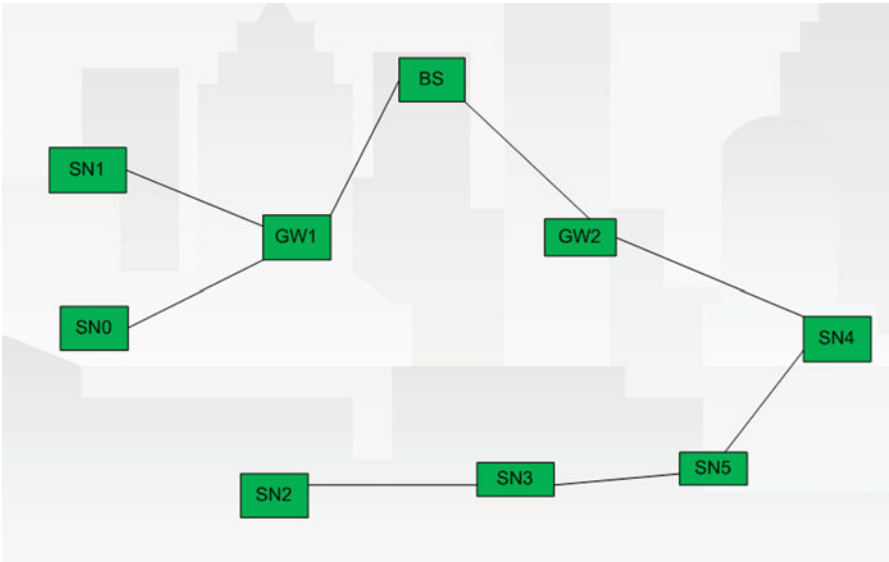

Fig. 6.5 Sparse topology sensor network scenario

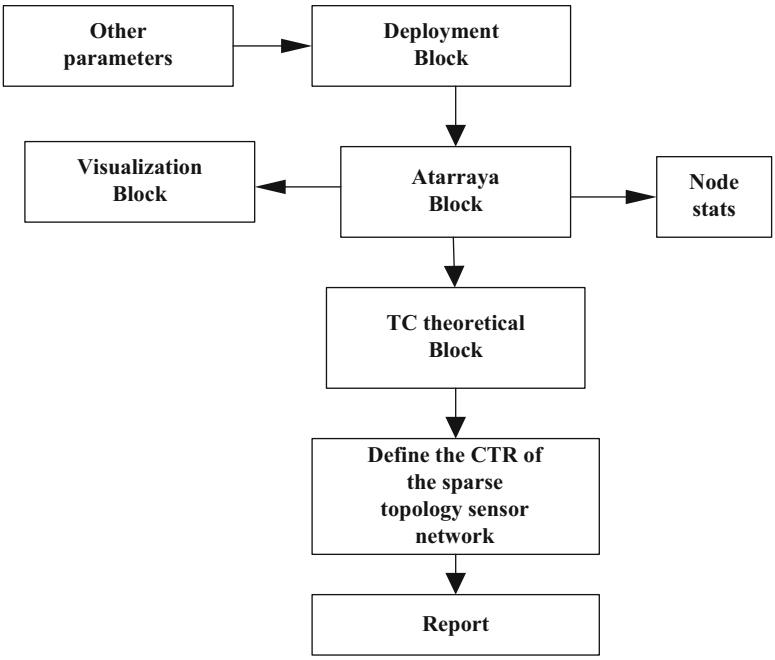

Fig. 6.6 Functional block diagram of the sparse topology sensor network

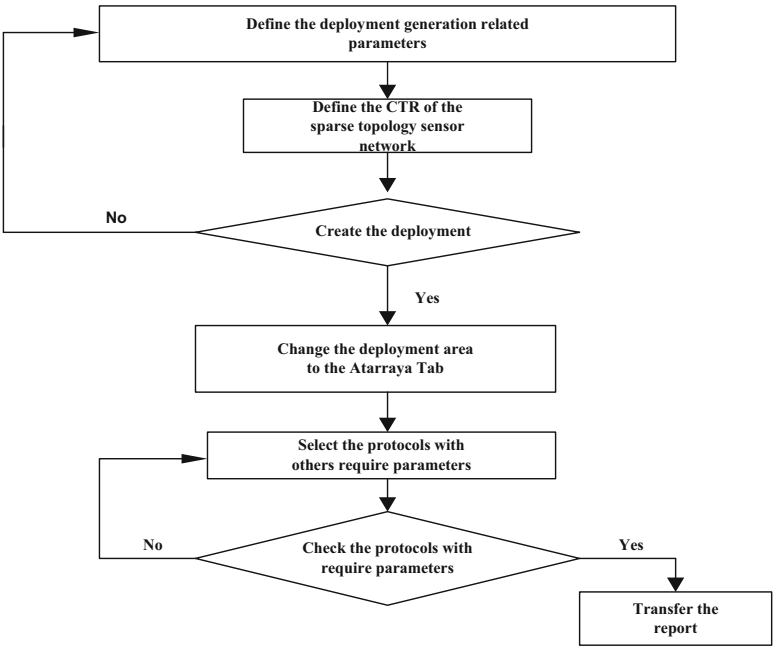

Fig. 6.7 Flow diagram of the sparse topology sensor network

that defines the topology of the deployment area. To develop the sparse topology sensor network, the *CTR* of the sparse network is defined by sparse-based *CTR* formula. The report block converts the machine-readable data to the human-readable format which consist of the last mandatory block among major blocks.

Figure 6.7 shows the flow diagram of the sparse topology sensor network. As illustrated in Fig. 6.7, all the parameters related to deployment area are defined first. The *CTR* of the sparse topology sensor network is added to the monitoring network. If the *CTR* of the sparse topology sensor network is defined in a proper way, then create the sparse topology deployment area. Otherwise, go back to the first step that defines the deployment generation-related parameters. The deployment area is added to the Atarraya tab after developing the monitoring area. Then select the deployment area protocol with other required parameters for the monitoring network. After checking all the protocols with required parameters in a proper way, then the simulation results are transferred to the readable format.

Using theorem from [34], the *CTR* can be calculated for one-dimensional sparse topology network as follows:

$$\mathrm{CTR} = k\frac{l\ \log\ l}{l} \tag{6.4}$$

Table 6.3 Simulation setup for lifetime analysis of sparse topology network

Parameters	Design values
Number of nodes	100
Communication radius	100 m
Sensing radius	20 m
Deployment area	600 × 600 m with central 300 × 300 m
Node energy distribution	1000 mJ (constant)
TC protocol	*A3*, *CDS-Rule-K*, *EECDS*, *K-neigh*, *A3 Cov*
TM protocol	Non isolated sink
Sensor and data protocol	Simple S and D protocol
Routing(forwarding protocol)	Simple forwarding
Performance metrics	Number of nodes alive, number of active nodes reachable from sink, covered area for communication, covered area for sensing

where the value of k is constant with $1 \leq k \leq 2$ and l is the length of uniformly distributed deployment area.

Santi [34] proposed a partial solution to find the *CTR* connectivity for d-dimensional sensor network as

$$\mathrm{CTR} = k\frac{l^d\ (\log\ l)}{n} \tag{6.5}$$

where $d = 2, 3, \ldots$ and $0 \leq k \leq 2^d\ d\frac{d}{2+l}$.

Equation (6.5) is used for testing the greatest component in sparse topology sensor network [34]. Considering a communication range, $r_{\mathrm{com}} = kl^{3/4}\sqrt{\log_2\ l}$ with $n = \sqrt{\mathrm{l}}$, $0.5 \leq k \leq l$ and $l = 2^{2i}$, where $4 \leq i \leq 10$, large amounts of random topologies with different values of k and l were assembled to determine the network lifetime.

Table 6.3 shows the parameters setup for lifetime analysis purpose of the sparse topology sensor network. The parameter design value of the sparse topology sensor network is described in Table 6.3 that consists of major parameters of interest in terms of the number of active nodes, number of active nodes reachable from sink, covered area for communication, and covered area for sensing.

6.7 Lifetime Evaluation of the Dense Topology Sensor Network

In this section, the experiment results related to the dense topology monitoring system to determine its lifetime are presented using topology construction protocols. The comparison results of the *EECDS*, *CDS-Rule-K*, *K-neigh*, *A3* and *A3-Cov* topology construction protocols are presented with considered performance metrics.

In those experiments, the dense topology sensor networks are defined in which the communication radius is calculated based on the *CTR* formula of Penrose-Santi [34]. The implementations of those protocols were coded and tested using the Atarraya simulation tool, which is designed for the purpose of testing the topology construction algorithm. Four main performance metrics are utilized to assess the lifetime of the dense topology monitoring system: (1) Number of active nodes; (2) number of active nodes reachable from sink; (3) coverage area for communication; (4) covered area for sensing. Table 6.2 mentioned in Sect. 6.6.2.1 is used as a summary of the simulation parameters for monitoring system lifetime. The first and second metrics show with preserving network connectivity and coverage how the topology construction protocol effectively reduces the amount of active nodes and reachable nodes from sink in the monitoring network. The other two metrics show efficiency of the topology construction protocols in terms of communication and sensing coverage area of the monitoring system.

This section presents the evaluation result of the experiments considered in this chapter for the lifetime analysis of the dense topology network. In this section, four sets of experiments are evaluated to identify the dense topology monitoring lifetime. Section 6.7.1 presents the first experiment of the dense topology network by considering the number of active nodes in terms of network transmission time. Section 6.7.2 describes experiment 2 to define the number of active nodes reachable from sink. This experiment compares the topology construction protocols results to provide better topology sensor network and observe the network behavior with high-density nodes. Section 6.7.3 describes experiment 3 to identify which topology construction protocols offer the better lifetime of the monitoring system in terms of communication coverage area. Sections 6.7.4 describes experiment 4 that uses the sensing coverage area of the network with considered topology construction protocols. This experiment observes how the sensing coverage area of the network behaves with topology construction protocols with high-density nodes in terms of network transmission time.

6.7.1 Experiment 1: Number of Active Nodes

The main goal of this experiment is to compare the topology construction algorithm in terms of the number of active nodes by increasing the transmission time of the network. Those topology construction protocols work based on node information of neighbors. Therefore, it is important to measure the performance of topology construction protocol with the number of active nodes in the network. Figure 6.8 shows the number of active nodes versus the lifetime of the network using *EECDS*, *K*-neigh, *CDS-Rule-K*, *A3*, *A3-Coverage* topology construction protocols with energy- and time-based criteria and no topology maintenance at all. The trends are cleared regardless of the topology construction algorithm used; *A3* and *EECDS* improve the lifetime of the monitoring network compared with the *EECDS* topology construction protocol in terms of the number of active nodes performance metrics.

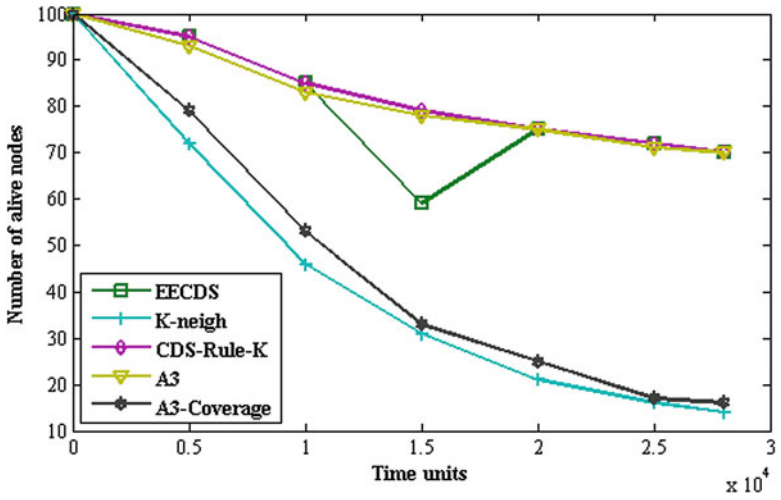

Fig. 6.8 Network lifetime for a number of active (alive) nodes

The *CDS-Rule-K* approach produced the best performance, while the performance of the *A3* technique continues to be very close to that of *CDS-Rule-K*. This result is expected due to the ability to create a preliminary version of the *CDS*, and adds or removes nodes to obtain a better approximation to the optimal *CDS* in the network.

The conclusion of this experiment is that the *K-neigh* topology construction protocol approach is the best number of active nodes in the network for monitoring structural health. In the case of the number of active nodes, all protocols provide higher value at initial operation of the monitoring network. The *A3-Coverage* protocol improves the number of active nodes that means lifetime of the network over the *K-neigh* topology construction protocol that degrades the network lifetime until the network dies at 2.8 time units. After that, the number of active nodes in case of the *EECDS* protocol degrades the system performance compared with that of the *A3* protocol. But the *A3* protocol result is always dominant over *A3-Coverage* and *K-neigh* protocols until the network dies at 2.8 time units. The *K-neigh* topology construction protocol produced best performance for extending the monitoring system lifetime, while the *A3* protocol extends the monitoring system lifetime.

6.7.2 Experiment 2: Number of Reachable Nodes from Sink

The main goal of this experiment is to compare the results produced by the topology recreation protocols in terms of the number of reachable nodes from sink while a fixed communication range of nodes 100 m and 100 numbers of nodes uniformly distributed in the area of 600 $\times$ 600 m deployment area. This experiment is important to show how much amount of active nodes is reachable from sink in dense topologies and how the resource usage depends on the number of active nodes

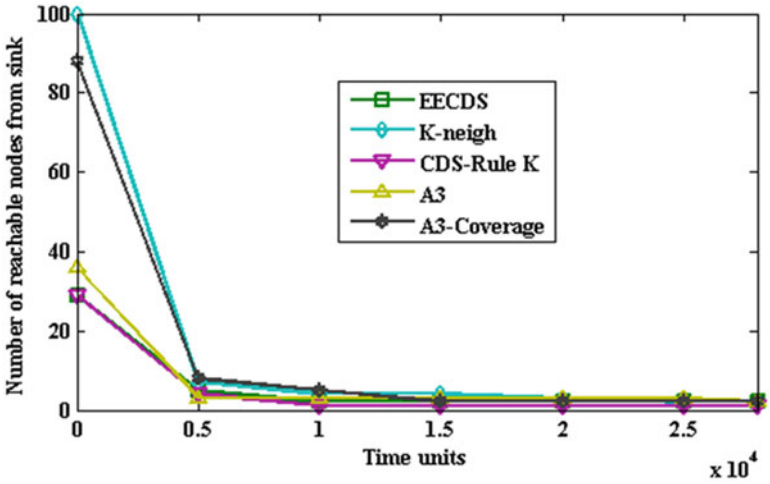

Fig. 6.9 Network lifetime for a number of reachable nodes from sink

reachable from sink. The results shown in Fig. 6.8 are not similar to the ones shown in Experiment 1.

Figure 6.9 shows the performance of the topology construction protocol technique in dense network in terms of a number of reachable nodes from sink. The *A3* protocol improves the lifetime of the network compared with *EECDS* and *CDS-Rule-K*. It is seen that the performance of the *EECDS* protocol continues to be very close to that of the *CDS-Rule-K* topology construction protocol. The *A3-Coverage* protocol shows the improvement when *EECDS*, *CDS-Rule-K* degrade the system performance compared to *A3-Coverage*. The *K-neigh* tree mechanism extends the network lifetime, while *A3-Coverage* provides very close results continuously compared with *K-neigh*. This result is expected because *K-neigh* has the ability to connect with a minimum number of neighbor set and transmission power.

The conclusion of this experiment is that *the A3-Coverage* and *K-neigh* protocols are the best policy for a number of active nodes reachable from sink in monitoring high-rise building structural health. Figure 6.9 shows that all topology construction protocols need a similar amount of active nodes reachable from sink from 0.5 time units to until the network dies. Before time units 0.5 of the network, the number of active nodes reachable from sink of *K-neigh* protocol is 100% but *A3-Cov*erage provides the number of active nodes 12%, after that *A3-Cov*erage provides a better number of reachable nodes from sink compared with K-neigh until the network dies. The behavior of *A3-Coverage* protocol can be explained by the fact that having a higher number of active nodes reachable from sink not only consumes more energy, but also message complexity because a higher number of active nodes generate a higher number of messages and travel to the sink. It is also important mentioning that this experiment is performed to show that various topology construction protocols do have an impact on the number of active nodes

reachable from sink and lifetime of the network. The results show how the *CDS-Rule-K* topology construction mechanism provides a better number of active nodes and network lifetime compared with *EECDS*, *K-neigh*, *A3*, *CDS-Rule-K*, mainly because the *A3-Coverage* protocol uses all available resources in the network.

6.7.3 Experiment 3: Communication Covered Area

The main goal of this experiment is to compare the results produced by the topology construction approach in terms of communication coverage area performance metrics while a communication range of nodes 100 m and 100 numbers of nodes uniformly distributed in the area of 600 × 600 m deployment area. This experiment is important to show how much coverage area is gained in dense topology networks and how the resource usage depends on the communication coverage area of the network. After the execution of the topology construction algorithm, the active nodes in the network determine the communication coverage area. To cover the deployment area for monitoring structural health, the communication coverage area is expected as much as greater. Figure 6.10 shows the network lifetime simulation results using *EECDS*, *K-neigh*, *CDS-Rule-K*, *A3*, *A3-Coverage* topology construction protocols in the dense network in terms of communication coverage area. The covered area for communication of *EECDS* protocol improves the coverage area and lifetime of the network, while *CDS-Rule-K* provides too close result to and is slightly better than *EECDS*. The *A3-coverage* protocol extends the network lifetime, when the *K-neigh* topology construction protocol degrades the system performance which is smaller amount compared to *A3-Coverage*. The *A3* protocol approach produced better performance; while the performance of *A3-Coverage* technique is similar and initially better than that of *A3*. In terms of communication coverage area and network lifetime: *A3* is still better compared with *A3-Coverage*. This result is expected because *A3* protocol is an energy-efficient topology construction protocol that has the ability to find a suboptimal connected dominating set to turn off unnecessary nodes.

The conclusion of this experiment is that the *A3* topology construction protocol approach is the best communication coverage policy for monitoring structural health. In the case of communication coverage area, the *A3-Coverage* protocol provides a communication coverage ratio of 100% at initial operation of the monitoring network. The *K-neigh* approach provides 98% ratio of the communication coverage area for monitoring that is 2% smaller than that of *A3-Coverage* protocol initially. In the third and fourth places are the *A3* and *CDS-Rule-K* according to the highest number of sensing gain, which is 96 and 94%, respectively at the initial operation of the network. The *EECDS* protocol attained the fifth place with about 93% of the communication coverage area initially and after that the result continues to be very close to that of *CDS-Rule-K*. From 0.5 time units until the network dies, the *A3* protocol provides better results than other topology mechanisms which have similar results.

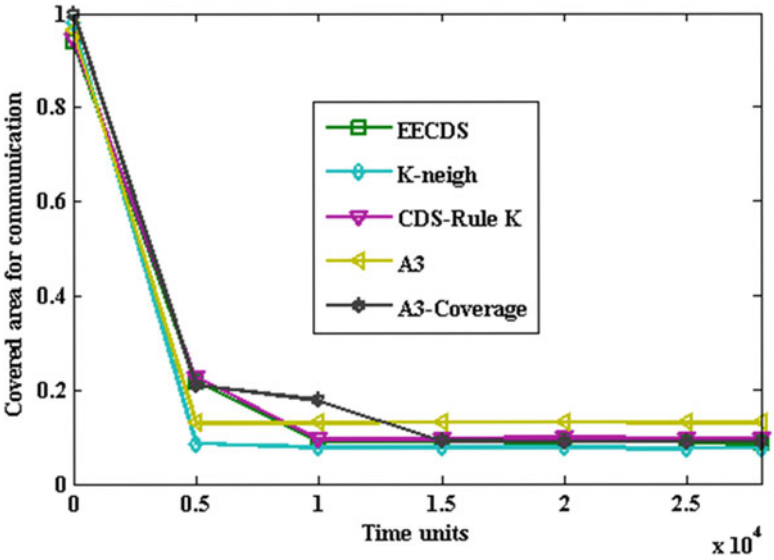

Fig. 6.10 Network lifetime for communication coverage area

6.7.4 Experiment 4: Sensing Coverage Area

The main goal of this experiment is to compare the experimental result of the topology construction algorithm in dense topology sensor networks in terms of sensing coverage area. After the execution of the topology construction algorithm, the sensing coverage area in the network determines the monitoring area of interest. To cover the deployment area for monitoring structural health, the sensing coverage area is expected to be greater near to area of interest. Therefore, it is important to measure the performance of topology construction protocol sensing area. Figure 6.11 shows the ratio of sensing covered area versus transmission time of the network using *EECDS*, *K-neigh*, *CDS-Rule-K*, *A3*, *A3-Coverage* protocols, using no topology maintenance over time- and energy-based triggering criteria. The result shows that all protocols provide nonlinear decrease. The results in Fig. 6.11 show that *EECDS, CDS-Rule-K, A3* protocols have a similar effect. While *EECDS, CDS-Rule-K, A3* protocols degrade the system performance compared with *K-neigh* and *A3-Coverage* protocols, and *K-neigh* protocols improve the coverage area and network lifetime when other considered protocols provide a small coverage area in the monitoring system. The result shows that the *A3-coverage* protocol produced a greater ratio of sensing area than *K-neigh*. This result is desired because it has the ability to add extra nodes to provide extra coverage area for sensing with minimum complexity and node energy.

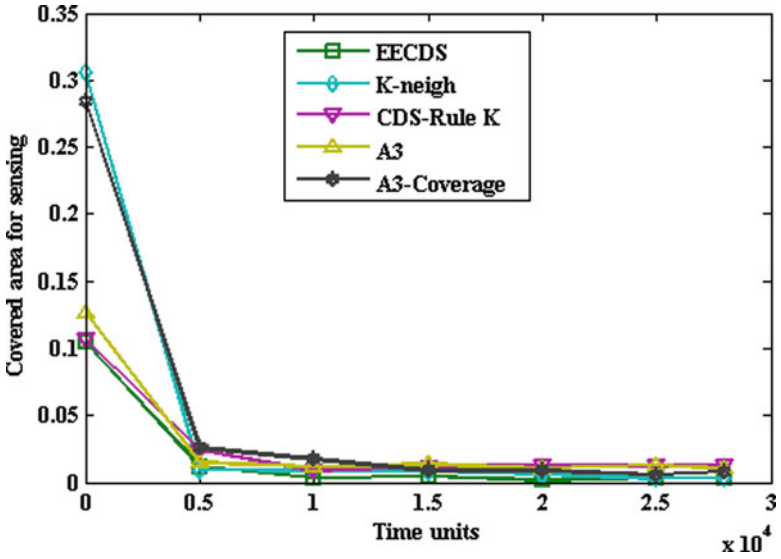

Fig. 6.11 Network lifetime for sensing coverage area

The conclusion of this experiment is that the *A3-Coverage* topology construction protocol approach is the best coverage policy for monitoring high-rise building structural health. In the case of sensing coverage area, the *A3-Coverage* protocol provides a coverage ratio of 28% at initial operation of the monitoring network. The *K-neigh* approach provides 30% ratio of the sensing coverage area for monitoring that is 2% greater than that of *A3-Coverage* protocol initially. From 0 to 0.5 time units, the sensing coverage ratio decays and *A3-Coverage* leads the *K-neigh* protocol. After that, the sensing coverage area decays and *A3-Coverage* area always is dominant until the network dies at 2.8 time units. On the other hand, *A3* protocol provides a sensing coverage ratio of 12% initially, and then decays until the network dies at 2.8 time units. Initially, *EECDS* and *CDS-Rule-K* gain the same sensing coverage ratio of 10% and after that the results continue to be very close with each other until 0.5 time units. From 0.5 to 2.8 time units, the sensing coverage of *CDS-Rule-K* protocol is dominant over that of the *EECDS* approach. After the time units 0.5 until 2.8, all protocols provide similar sensing coverage area and it is hard to define a better topology construction protocol based on this range. The trade-off between *A3-Coverage* and *K-neigh* approaches is very clear: Although *A3-Coverage* covers 2% less sensing area than *K-neigh* initially, after that its better coverage area is compared with that of *K-neigh*. This behavior can be explained by the fact that having more sensing area not only consumes more energy, limiting their use for future, but also more energy because of the number of messages that they generate and travel to the sink, using resources from all nodes in the path. It is also worth mentioning that these experiments are performed to show that different topology construction protocols do have an impact on the sensing coverage area and lifetime

of the network. The results show how the *A3-Coverage* topology construction mechanism provides a better sensing coverage area and network lifetime than the *EECDS, K-neigh, A3, CDS-Rule-K*, mainly because the *A3-Coverage* protocol uses all available resources in the network.

6.8 Lifetime Evaluation of the Sparse Topology Sensor Network

In this section, the experiment results related to the lifetime of the sparse topology monitoring network are presented using topology construction protocols. The comparison results of the *EECDS, CDS-Rule-K, K-neigh, A3 and A3-Coverage t*opology construction protocols are presented. In all considered experiments, the sparse topology sensor network is defined in which the communication radius is calculated based on the *CTR* formula of Penrose-Santi [34]. Similar numbers of performance metrics compared to the dense network are considered to assess the lifetime of the sparse topology sensor network monitoring system. Table 6.3 in Sect. 6.6.2.2 describes the summary of the simulation parameters to determine sparse topology lifetime.

This section presents the evaluation result of the experiments considered for the lifetime analysis of the sparse topology sensor network. The four sets of experiments are evaluated that define the sparse topology sensor network monitoring system lifetime. Section 6.8.1 presents the first set of experiments of the sparse topology sensor network, which consider the number of active nodes of the monitoring system to determine the monitoring system lifetime in terms of transmission time. Section 6.8.2 describes experiment 2 to determine the amount of active nodes reachable from sink of the monitoring network by considering high-density network nodes and topology construction protocols. Section 6.8.3 describes experiment 3 that compares the topology construction protocols results in terms of monitoring network communication coverage area. This experiment observes which topology offers better monitoring system lifetime in terms of communication coverage area. Section 6.8.4 describes experiment 4 that determines the sensing coverage area of the monitoring network with high-density network nodes. This experiment observes how the sensing coverage of the monitoring network behaves with considered topology construction protocols.

6.8.1 Experiment 1: Number of Active Nodes

Figure 6.12 shows the number of alive nodes versus the lifetime of the network using *EECDS, K-neigh, CDS-Rule-K, A3, A3-Coverage* topology construction protocols with energy- and time-based criteria. The trends are cleared regardless of the

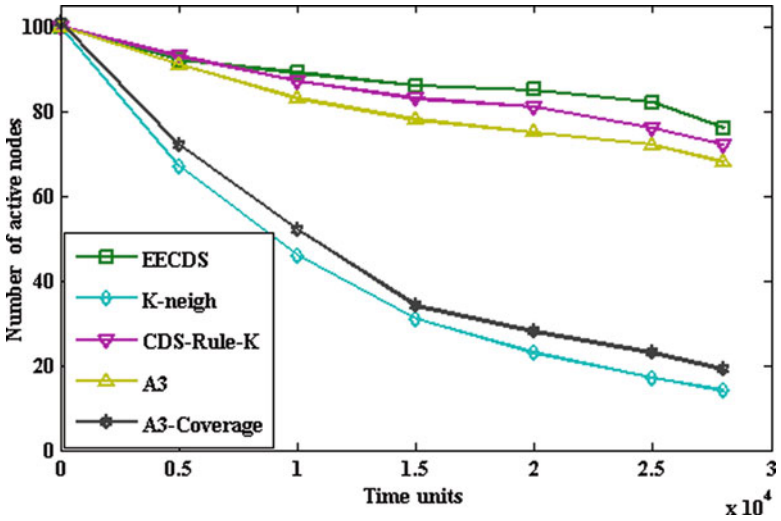

Fig. 6.12 Network lifetime for a number of active (alive) nodes

topology construction algorithm used; *A3-Coverage* and *A3* improve the monitoring system lifetime compared with the *K-neigh* topology construction protocol in terms of the number of active (alive) nodes performance metrics. The *EECDS* approach produced the best performance; while the performance of the *CDS-Rule-K* technique continues to be very close with that of *EECDS*. This result is expected due to the ability to create maximum independent sets in the first phase and during the second phase to select gateway nodes to connect the independent set.

The conclusion of this experiment is that the *K-neigh* topology construction protocol approach is the best number of active nodes in the network for monitoring structural health. Although the EECDS topology construction protocol provides higher value than other topology construction protocols, due to the goal of the topology construction protocol it is not suitable.

6.8.2 Experiment 2: Number of Reachable Nodes from Sink

Figure 6.13 shows the performance of the topology construction protocol using no topology maintenance technique in the sparse topology sensor network in terms of the number of reachable nodes from sink. The results shown in Fig. 6.13 are not similar to the ones shown in experiment 1. Before 0.5 time units, *A3, CDS-Rule-K, EECDS* provide almost similar results but after 0.5 time unit the result becomes closest with each other. The *A3* protocol improves the lifetime of the network compared with *A3-Coverage, EECDS, and CDS-Rule-K*. It is seen that the performance of the *K-neigh* protocol continues to be very close to

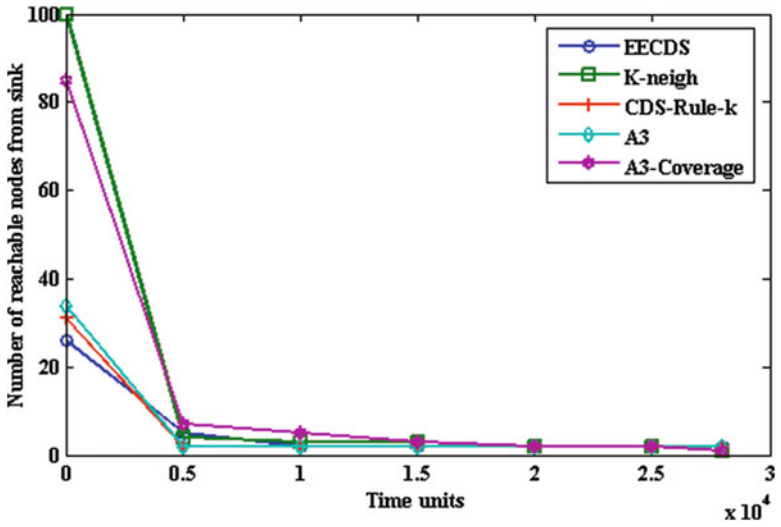

Fig. 6.13 Network lifetime for a number of reachable nodes from sink

that of the *A3-Coverage* topology construction protocol. The *A3* protocol shows the improvement when *EECDS*, *CDS-Rule-K*, *A3-Coverage* degrade the system performance compared to *A3*. The *K-neigh* tree mechanisms extend the network lifetime, while *A3-Coverage* provides better result continuously than *K-neigh*. This result is expected because *A3-Coverage* protocol uses all available resources in the network.

The conclusion of this experiment is that the *A3* and *K-neigh* protocols are the best policy for the number of active nodes reachable from sink in monitoring structural health. Figure 6.13 shows that all topology construction protocols need a similar amount of active nodes reachable from sink from 0.5 time units until the network dies. Before time units 0.5, the number of active nodes reachable from sink of *K-neigh* protocol is 100% but that of *A3-Cov* is 85%. Since the goal of the topology construction protocol is to connect a minimum number of nodes while keeping an important property of the network such as connectivity. Therefore, the result shows that the *A3* provides a better number of reachable nodes from sink than *K-neigh* until the network dies.

6.8.3 Experiment 3: Coverage Area for Communication

After the execution of the topology construction protocol, the active nodes in the network determine the communication coverage area. The communication coverage area is expected to be greater to cover the deployment area for monitoring building structure. Figure 6.14 shows the network lifetime simulation results using *EECDS*,

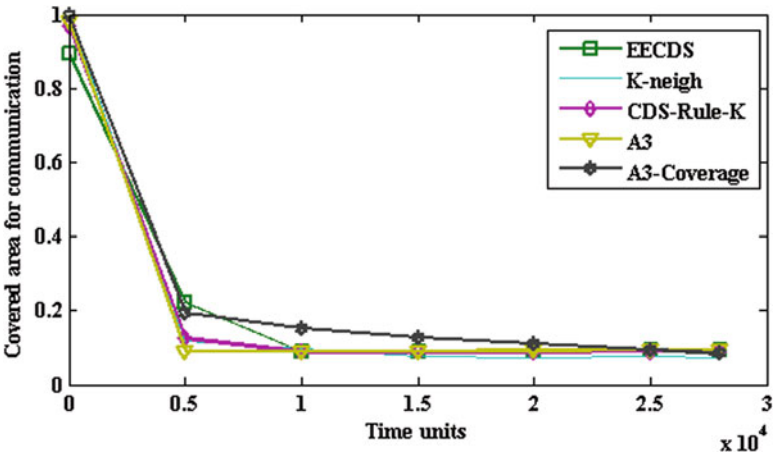

Fig. 6.14 Network lifetime for communication coverage area

K-neigh, *CDS-Rule-K*, *A3*, *A3-Coverage* topology construction (*TC*) protocols in sparse network in terms of ratio of communication coverage area performance metrics. The covered area for communication of *EECDS* protocol improves the coverage area and lifetime of the network, while *CDS-Rule-K* provides a very similar result to and is slightly better than *EECDS*. The *A3* and *K-neigh* protocols provide similar results for network lifetime. When *CDS-Rule-K* and *A3* topology construction protocols degrade the system performance by a smaller amount compared to *A3-Coverage*, the *A3-Coverage* protocol approach produced better performance than all others. In terms of communication coverage area, *A3-Coverage* is still better. This result is expected because *A3-Coverage* protocol has the ability to use all available resources in the network.

The conclusion of this experiment is that the *A3-Coverage* topology construction protocol approach is the best communication coverage policy for the sparse topology sensor network in monitoring high-rise building structural health. In the case of communication coverage area, the *A3-Coverage* protocol provides a communication coverage ratio of 100% at initial operation of the monitoring network. The *A3* approach provides 98% ratio of the communication coverage area for monitoring that is 3% smaller than that of *A3-Coverage* protocol initially. In the third and fourth places are the *CDS-Rule-K* and *EECDS* according to the highest number of sensing gain, which is 97 and 89%, respectively at the initial operation of the network. The *K-neigh* protocol attained about 100% of the communication coverage area initially and after that degrades the system performance compared with *A3-Coverage* protocol. From 0.5 time units until the network dies, the *A3-Coverage TC* protocol provides better results than other topology mechanisms.

6.8.4 Experiment 4: Coverage Area for Sensing

Figure 6.15 show the ratio of sensing covered area versus the lifetime of the network using the *EECDS*, *K-neigh*, *CDS-Rule-K, A3, A3-Coverage* protocols and no topology maintenance over time- and energy-based triggering criteria in the sparse topology sensor network. The results show that all protocols provide the nonlinear decrease. The results show that *EECDS, CDS-Rule-K, A3* protocols are similar to each other. *EECDS, CDS-Rule-K, A3* protocols degrade the system performance compared with *K-neigh* and *A3-Coverage* protocols. *A3-Coverage and K-neigh* protocols improve the sensing coverage area and network lifetime when other considered protocols provide small coverage area in the monitoring system. The result also shows that *A3-coverage* protocol produces a slight better ratio of sensing coverage area than *K-neigh*. This result is desired because it has the ability to add extra nodes to provide extra coverage area for sensing with minimum complexity and node energy.

The conclusion of this experiment is that the *A3-Coverage* topology construction protocol approach is the best coverage policy for monitoring high-rise building structural health. In the case of sensing coverage area, the *A3-Coverage* protocol provides a coverage ratio of 27% at initial operation of the monitoring network. The *K-neigh* approach provides 29% ratio of the sensing coverage area for monitoring that is 2% greater than that of *A3-Coverage* protocol initially. From 0 to 0.5 time units, the sensing coverage ratio decays and *A3-Coverage* leads the *K-neigh* protocol. After that, the *K-neigh* sensing coverage area decays and *A3-Coverage* area is always dominant until the network dies at 2.8 time units. On the other hand,

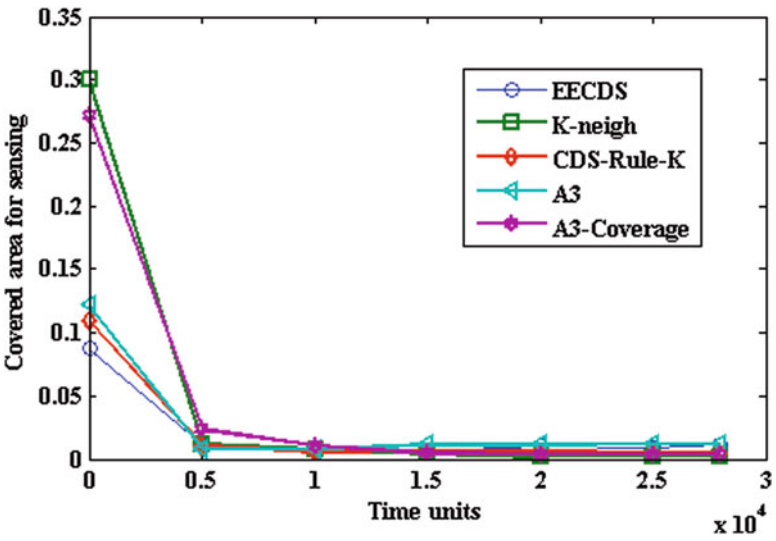

Fig. 6.15 Network lifetime for sensing coverage area

A3 protocol provides a sensing coverage ratio of 12% initially, and then decays until the network dies at 2.8 time units. Initially, *EECDS* and *CDS-Rule-K* gain the sensing coverage ratio of 10 and 8% after that the results continue to decrease until 0.5 time units. From 0.5 to 2.8 time units, the sensing coverage of *EECDS* protocol is dominant over that of the *CDS-Rule-K* approach. After the time units 0.5 until 2.8, all protocols provide similar sensing coverage area and it is hard to define the better topology construction protocol based on this range. The results show that the *A3-Coverage* topology construction mechanism provides a better sensing coverage area and network lifetime than other topology construction protocols, mainly because the *A3-Coverage* protocol uses all available resources in the network.

6.9 Comparison Between Dense and Sparse Topology Sensor Networks

In this section, the results of experiment related to the lifetime of the WSN monitoring system are compared using considered performance metrics, in both dense and sparse topology sensor networks. The lifetime-related experimental results of the dense and sparse topology sensors are compared using topology construction protocols in terms of the number of active nodes, the number of active nodes reachable from sink, covered area for communication, and covered area for sensing.

Table 6.4 shows the comparison results of the dense and sparse topology sensor networks in terms of the number of active nodes with lifetime performance metrics. Comparison results show that the dense topology K-neigh protocol exhibits better results than the dense topology sensor network.

For better understanding of the comparison results of the dense and sparse topology sensor networks, Table 6.5 shows the detailed description of the result

Table 6.4 Experiment 1: number of active nodes

		Transmission time			
Topology	Protocols	0	10,000	20,000	28,000
Dense topology network	EECDS	100	85	75	70
	K-neigh	100	46	21	14
	CDS-rule-K	100	85	75	70
	A3	100	83	75	70
	A3-Cov	100	53	25	16
Sparse topology network	EECDS	100	87	81	72
	K-neigh	100	46	23	14
	CDS-rule-K	100	89	85	76
	A3	100	83	75	68
	A3-Cov	100	52	28	19

Table 6.5 Experiment 2: number of active nodes reachable from sink

		Transmission time			
Topology	Protocols	0	10,000	20,000	28,000
Dense topology network	EECDS	29	2	2	2
	K-neigh	100	4	3	2
	CDS-rule-K	29	1	1	1
	A3	100	83	75	70
	A3-Cov	88	5	2	2
Sparse topology network	EECDS	25	2	2	2
	K-neigh	100	3	2	1
	CDS-rule-K	31	2	2	2
	A3	34	2	2	2
	A3-Cov	85	5	2	1

Table 6.6 Experiment 3: covered area for communication

		Transmission time			
Topology	Protocols	0	10,000	20,000	28,000
Dense topology sensor network	EECDS	0.937	0.090	0.087	0.084
	K-neigh	0.980	0.077	0.076	0.073
	CDS-rule-K	0.945	0.094	0.098	0.096
	A3	0.964	0.128	0.130	0.129
	A3-Cov	1	0.176	0.089	0.089
Sparse topology sensor network	EECDS	0.895	0.089	0.090	0.091
	K-neigh	1	0.099	0.073	0.074
	CDS-rule-K	0.970	0.087	0.088	0.089
	A3	0.983	0.090	0.091	0.094
	A3-Cov	1	0.152	0.111	0.083

in terms of the number of active nodes reachable from sink. Comparison results show that the dense topology sensor network exhibit better results than the sparse topology sensor network. The dense CDS-Rule-K provides better results in terms of the number of active nodes than other considered dense and sparse topology construction protocols.

Table 6.6 shows the comparison results of the dense and sparse topology sensor networks which contain the detailed description of the experimental result in terms of communication coverage with lifetime performance metrics. Results show that, in both dense and sparse topology sensor networks, A3-Coverage topology construction protocols exhibit better results than the considered topology construction protocols.

The experimental results of both dense and sparse topology sensor networks in case of sensing coverage area are presented in Table 6.7. Results show that, in both dense and sparse topology sensor networks, *A3-Coverage* topology construction protocols demonstrate better results than the other considered topology construction protocols in terms of sensing coverage area of the monitoring network.

Table 6.7 Experiment 4: covered area for sensing

		Transmission time			
Topology	Protocols	0	10,000	20,000	28,000
Dense topology network	EECDS	0.104	0.003	0.001	0.003
	K-neigh	0.305	0.008	0.005	0.002
	CDS-rule-K	0.106	0.008	0.012	0.013
	A3	0.126	0.011	0.010	0.010
	A3-coverage	0.284	0.017	0.008	0.008
Sparse topology network	EECDS	0.087	0.005	0.009	0.010
	K-neigh	0.299	0.007	0.001	0.001
	CDS-rule-K	0.109	0.005	0.006	0.004
	A3	0.121	0.008	0.011	0.011
	A3-coverage	0.271	0.010	0.003	0.003

6.10 Discussion and Conclusion

In this study, the Theory of Geometric Random Graphs approach has been proposed for monitoring structural health. In this article, the lifetime of the monitoring system has been investigated in dense and sparse topology sensor networks for monitoring structural health. The practice of SHM suffers from large coverage area information with lifetime of the monitoring system. Research review has been shown that the problem of the monitoring system can be addressed by modifying the monitoring system using the WSN technique. However, a challenge arises to select an optimum topology construction protocol to fulfill the current needs in the WSN monitoring network because every system has its own requirements. The goal of this study is to develop an improved WSN monitoring system using Theory of Random Graph Approach to overcome the limitation of coverage area and lifetime in the existing WSN monitoring system.

To achieve the first objective, the simulation model of building SHM system has been developed using dense and sparse topology WSNs.

To achieve the second objective, the lifetime of the dense and sparse topology sensor networks has been analyzed in terms of lifetime performance metrics. A distributed model has been derived for dense and sparse topology sensor networks using critical transmission range formula. The topology construction protocol provides reliable information to identify the optimum topology construction protocol for the monitoring network. Various topology construction protocols were used to identify the better monitoring network. The developed monitoring system was tested using the Atarraya Java-based simulator.

Results also show that the dense topology *K-neigh* provides better results in terms of the number of active nodes than other considered dense topology construction protocols. In case of the number of active nodes reachable from sink, the dense *CDS-Rule-K* topology construction protocol provides better results than other considered dense and sparse topology construction protocols. For communication coverage

area, dense and sparse A3-Coverage topology construction protocols exhibit the same result and any one would be a good choice for the monitoring network. In case of sensing coverage area, the *K-neigh* dense topology construction protocol proved better performance than others. Finally, it is seen that the dense topology sensor network is selected as an optimum lifetime topology construction for monitoring structural health compared with the sparse topology sensor network. The author believes that the results presented in this article provide a better understanding of lifetime comparison between dense and sparse topology sensor networks in structural health monitoring application.

Acknowledgments The research described in this paper was partially financially supported by the Science Fund Malaysia.

References

1. C.R. Farrar, K. Worden, An introduction to structural health monitoring. Philos. Trans. R. Soc. A Math. Phys. Eng. Sci. **365**(1851), 303–315 (2007)
2. J.M. Renno, B.R. Mace, Vibration modelling of structural networks using a hybrid finite element/wave and finite element approach. Wave Motion **51**(4), 566–580 (2014)
3. M. Choi, B. Sweetman, Efficient calculation sensor technology for structural health monitoring. Struct. Health Monit. **9**(1), 13–24 (2010)
4. M.E. Haque, M.A. Hannan, M.F. Hossain, M.M. Islam, M.J. Abedin, Lifetime measure of dense and sparse topology sensor network in structural health monitoring. EAI Endorsed Trans Scalable Info Syst **4**, e6 (2017)
5. S.D. Glaser, H. Li, M.L. Wang, J. Ou, J. Lynch, Sensor technology innovation for the advancement of structural health monitoring: a strategic program of US-China research for the next decade. Smart Struct Syst **3**(2), 221–244 (2007)
6. G. Park, T. Rosing, M.D. Todd, C.R. Farrar, W. Hodgkiss, Energy harvesting for structural health monitoring sensor networks. J. Infrastruct. Syst. **14**(1), 64–79 (2008)
7. S. Alahakoon, D.M. Preethichandra, E.M.I. Ekanayake, Sensor network applications in structures: a survey. EJSE Int., 1–10 (2009). www.ejse.org/Archives/Fulltext/2009/Special/2009SP1.pdf
8. F. Casciati, F. Lucia, Sensor placement driven by a model order reduction (MOR) reasoning. Smart Struct Syst **13**(3), 343–352 (2014)
9. S. Cho, C.B. Yun, J.P. Lynch, A.T. Zimmerman, B.F. Spencer Jr., T. Nagayama, Smart wireless sensor technology for structural health monitoring of civil structures. Int J Steel Struct **8**(4), 267–275 (2008)
10. P.C. Chang, A. Flatau, S.C. Liu, Review paper: Health monitoring of civil infrastructure. Struct. Health Monit. **2**(3), 257–267 (2003)
11. K. Mechitov, W. Kim, G. Agha, T. Nagayama, High-frequency distributed sensing for structure monitoring, in *Proceeding of First International Workshop on Networked Sensing Systems (INSS 04)* 6 (2004), pp. 101–105
12. Y.Z. Song, C.R. Bowen, A.H. Kim, A. Nassehi, J. Padget, N. Gathercole, Virtual visual sensors and their application in structural health monitoring. Struct Health Monit. **3**, 1475921714522841 (2014)
13. V.A. Attarian, F.B. Cegla, P. Cawley, Long-term stability of guided wave structural health monitoring using distributed adhesively bonded piezoelectric transducers. Struct. Health Monit. **13**, 265 (2014)

14. J. Paek, N. Kothari, K. Chintalapudi, S. Rangwala, R. Govindan, *The Performance of a Wireless Sensor Network for Structural Health Monitoring* (Center for Embedded Network Sensing, California, 2004)
15. M.E. Haque, *Development of dense and sparse topology sensor networks for structural health monitoring using atarraya simulator*. Master of Science dissertation (National University of Malaysia, Selangor, 2015)
16. R.P. Bandara, T.H. Chan, D.P. Thambiratnam, Structural damage detection method using frequency response functions. Struct. Health Monit. **13**, 418 (2014)
17. L.L. Halpern, J.Y.P. Bahl, Y.M. Wang, R. Wattenhofer, A cone-based distributed topology-control algorithm for wireless multi-hop networks. IEEE/ACM Trans. Networking **13**(1), 147–159 (2005)
18. W.S. Conner, L. Krishnamurthy, R. Want, Making everyday life easier using dense sensor networks, in *Ubicomp 2001: ubiquitous computing*, (Springer, Berlin, 2001), pp. 49–55
19. M. Li, P.J. Wan, Y. Wang, O. Frieder, Sparse power efficient topology for wireless networks, in *Proceedings of the 35th Annual Hawaii International Conference on IEEE* (2002), pp. 3839–3848
20. G. Fortino, A. Guerrieri, G.M. O'Hare, A. Ruzzelli, A flexible building management framework based on wireless sensor and actuator networks. J. Netw. Comput. Appl. **35**(6), 1934–1952 (2012)
21. H. Kawahigashi, Y. Terashima, N. Miyauchi, T. Nakakawaji, Modeling ad hoc sensor networks using random graph theory. in *Consumer Communications and Networking Conference on IEEE* (2005), pp. 104–109
22. B. Nikola, A. Dimitris, K. Berberidis, F. Casciati, J. Plata-Chaves, Spatio-temporal protocol for power-efficient acquisition wireless sensors based SHM. Smart Struct Syst **14**(1), 1–16 (2014)
23. S.B. Chase, The role of smart structures in managing an aging highway infrastructure. in *Keynote Presentation, SPIE Conference on Health Monitoring of Highway Transportation Infrastructure* (2001)
24. M. Moore, B. Phares, B. Graybeal, D. Rolander, G. Washer, Reliability of visual inspection for highway bridges. Volume I: Final report (No. FHWA-RD-01-020) (2001)
25. K. Vidya, *Wireless sensor network for structural health monitoring*. Doctor of Philosophy dissertation (Clarkson University, Potsdam, 2008)
26. U. Yildirim, O. Onur, B. Nikola, A prediction-error-based method for data transmission and damage detection in wireless sensor networks for structural health monitoring. J. Vib. Control. **19**(15), 2244–2254 (2013)
27. A. Araujo, J. García-Palacios, J. Blesa, F. Tirado, E. Romero, A. Samartín, O. Nieto-Taladriz, Wireless measurement system for structural health monitoring with high time-synchronization accuracy. IEEE Trans. Instrum. Meas. **61**(3), 801–810 (2012)
28. W. Chen, X. Jiang, X. Li, J. Gao, X. Xu, S. Ding, Wireless sensor network nodes correlation method in coal mine tunnel based on Bayesian decision. Measurement **46**(8), 2335–2340 (2013)
29. J. Khash-Erdene, N.S. Rohan, F. Kallirroi, A.K. Marios, F. Glauco, O. Toula, Layout optimization of wireless sensor networks for structural health monitoring. Smart Struct Syst **14**(1), 39–54 (2014)
30. G. Sun, J. Chen, W. Guo, K.J.R. Liu, Signal processing techniques in network-aided positioning. IEEE Signal Process. Mag. **7**, 12–23 (2005)
31. M.A. Labrador, P.M. Wightman, *Topology Control in Wireless Sensor Networks 412* (Springer, Heidelberg, 2009)
32. W.R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient communication protocol for wireless microsensor networks, in *System Sciences, 2000. Proceedings of the 33rd Annual Hawaii International Conference on*: 10. IEEE (2000)
33. S. Nath, V.N. Ekambaram, A. Kumar, P.V. Kumar, Theory and algorithms for hop-count-based localization with random geometric graph models of dense sensor networks. ACM Trans Sens Netw (TOSN) **8**(4), 35 (2012)
34. P. Santi, Topology control in wireless ad hoc and sensor networks. ACM Comput Surv (CSUR) **37**(2), 164–194 (2005)

Chapter 7
Internet of Things (IoT) Considerations, Requirements, and Architectures for Disaster Management System

Kamran Ali, Huan X. Nguyen, Purav Shah, Quoc-Tuan Vien, and Enver Ever

7.1 Introduction

Information and communication technologies provide vital services and systems for our daily lives with great potential to aid us in emergency and disaster situations as well. Many people around the world are adversely affected by numerous unexpected disasters such as earthquakes, tsunami, volcanic eruptions, and floods. Disasters and emergency crisis are usually unpredicted events that cause panic situations for the civilians and affect the existing resources. The need for communication and other types of information exchange services is in very high demand after such events. The communication infrastructure is often damaged to large extents, making services unavailable or at least heavily congested. Some of the unforgettable disasters in the history of mankind are Quetta, Kashmir (Pakistan) 1935 and 2008 earthquakes, Chernobyl (Russia) nuclear accident in 1986, Japan tsunami in 2011, Nepal earthquake 2015, Fort McMurry (Canada) forest fire in 2016, and recently the US forest fire in 2017. Around more than 12 million people have directly got affected during the last decade [1, 2].

In recent years, the focus of research in disaster management systems (DMS) has been on designing architecture that allows affected people in disaster situations to communicate with outside area and rescue teams. However, most of the present disaster communication systems rely on an existing network infrastructure and fail to provide services because of physical destruction of network equipment.

K. Ali (✉) · H. X. Nguyen · P. Shah · Q.-T. Vien
Faculty of Science and Technology, Middlesex University London, London, UK
e-mail: k.ali@mdx.ac.uk; H.Nguyen@mdx.ac.uk; P.Shah@mdx.ac.uk; Q.Vien@mdx.ac.uk

E. Ever
Computer Engineering Middle East Technical University, Northern Cyprus Campus, Guzelyurt, Mersin, Turkey
e-mail: eever@metu.edu.tr

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_7

According to Cisco forecasts, the number of mobile-connected devices exceeded world's population in 2014 and there will be 11.5 billion mobile-connected device by 2019 [3]. The ever-growing number of mobile devices introduces the potential for them to become an essential component of disaster management.

The world has just observed the origination of Internet of Things (IoT) that has previously created a huge buzz in social, economic, and technological domains. Although the combination of telecommunication systems with the IoT introduces unprecedentedly high requirements for, this combination is the most promising infrastructure for emerging applications that cover numerous domains, such as smart homes, entertainment, intelligent transportation systems, industrial automation, e-Health, public safety, smart grid, and Tactile Internet. The emergence of enhanced mobile devices with better computation and storage abilities also supports the boost for the study of innovative models and solutions in order to exploit support among users/machines for integrated IoT services [4]. The devices share their capabilities including sensing, storage, and computation whenever they reside in geographical proximity. However, a lot of work is required to adopt the future 5G-oriented IoT system. When considering the heterogeneity of devices, the constraints of the wireless communication and the specific features of the IoT, an important question is how to handle the stringent requirements of mission-critical communications [5], which demands low-latency and high-reliability connections during disaster situations.

In this context, for long-term evolution (LTE) technology, which is the basis for next-generation public safety broadband networks, it is essential to consider the addition of certain key features and functionality, such as direct device-to-device (D2D) broadcast. D2D communications and IoT-based networks are considered as key enabling technologies in future public safety networks and, thus, they have become an intriguing topic for research [6].

In this chapter, we focus on how disaster communication management can benefit from recent advancements in wireless communication, IoT, and mobile technologies and devices. The chapter surveys various potential emerging communication technologies such as IoT, 5G, D2D, and small cell for disaster situations. In particular, we focus on the use of ubiquitous mobile devices and applications in disaster situations.

7.2 Considerations and Requirements for Disaster Management System

According to research and observations, emergency communication systems should be deployed rapidly and deliver communication services in disaster areas. Due to the stringent time constraint and potential panic cases, there are many challenges which we face during emergency communication. Moreover, the survival rate is

highly dependent on the rescue speed and rescue operations which also rely on the communication system. The earlier the victims are saved, the higher the survival rate is. The survival rate is 90% within 24 h; 50% between 25 and 48 h; 20% between 49 and 72 h, and less than 5% after exceeding 72 h. To save more lives, disaster communication system should be deployed as quickly as possible [7].

We can categorize DMS into two phases: pre-disaster and post-disaster as shown in Fig. 7.1. The pre-disaster phase involves disaster mitigation whereas in the post-disaster phase the focus is on disaster communication connectivity and recovery operation in the affected areas. Some basic strategies are adopted before any critical event occurs to make communication system more reliable and scalable using IoT applications by deploying sensors. The sensors are mounted on building (e.g. exchanges), mobile towers, and poles which include temperature, motion, and speed sensors. All these sensors communicate with the remote station using steady ad hoc network. When there occurs any abnormal activity, these sensors generate an alarm to the remote station that can then distinguish the area of alarm with MAC or IP address. Figure 7.2 presents the concept of pre-disaster scenario. Pre-disaster measures are the basic strategy for the quick services for victims and rescue teams. However, it is essential to minimize the rebuilding/functional times required to fully activate the ICT services which are expected to explosively increase after the occurrence of an event. Pre-disaster phase is as important as the post-disaster phase, because an effective approach in pre-disaster phase leads to an efficient DMS in the post-disaster phase. The post-disaster phase covers setting up of communication infrastructure, locating and rescuing the victims by providing basic needs. An efficient disaster communication should strongly be based on post-disaster phase because it depends on the technologies and equipment available that can be used by the affected people on the ground as well as the first responders. The victims should be able to use the available technologies as fast as possible to communicate their location and whereabouts to the first responders. Mobile phones might be the first thing carried by most victims or volunteers in disaster situations. It is easy and ready to use without deployment of any extra network infrastructure, which means cost and time. Hence, using mobile phone as the terminals is a good option. Consider the popularity of notebook and tablet PC as well. We are interested in investigating current technologies that empower a reliable and rapid rescue operation in a disaster situation. Figure 7.1 highlights the key challenges and some features with requirements that should be met by a DMS. The authors in [8] proposed IoT-based solutions for efficient disaster management using task-technology fit (TTF) approach. Particularly, they analysed the strategic solution that can be implemented for realizing the benefits of IoT in disaster management.

In this paper, we assume DMS based on wireless networks and, therefore, the requirements are specific to wireless communication networks. Furthermore, a disaster communication system should provide some basic communication services as listed in Fig. 7.1 with the related main features. Note that this set of services and features allow developing more efficient disaster applications. In the case of data connectivity and messaging services-based applications, an important requirement

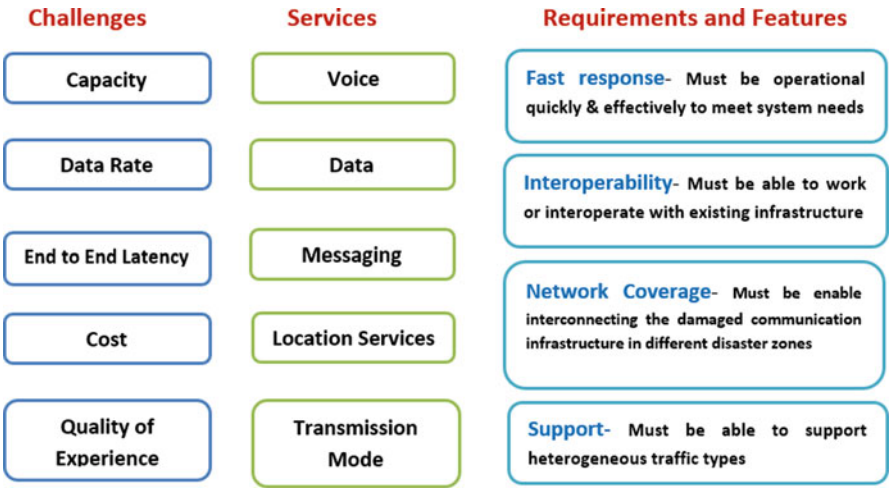

Fig. 7.1 Disaster communication management

is the bandwidth available to support the application. Good numbers of studies on investigation and exploration of disaster communication-based architectures and applied technologies exist in the literature. However, due to the climate change and, among other reasons, natural disasters have increased significantly over the years does not only cost in terms of assets and communication infrastructure destruction but also in casualties. Numerous solutions have been presented effectively to sustain communication after a disaster. Moreover, capacity, coverage enhancements, and performance gains during disaster and critical communication were also addressed in many proposals and research works. The 4G networks are now moving toward maturity and making the researchers to explicit new generation of wireless network 5G. The 5G will be a generation with enormously large bandwidths with very high carrier frequencies and with vast devices connectivity. In this study, we are investigating technologies that enable a rapid recovery, sustainability, and with efficient adaptability of current wireless devices during disaster-related communication.

7.3 Emerging Technologies for Disaster Management System

Today, to fulfil expectations and challenges of IoT-based networks we intend to advance in various ways, some of the key intentions or demands that need to be addressed are connectivity, better capacity, improvement on transfer data rate, and enhanced quality of services. A sustainable DMS network should not only be robust, resilient but also energy efficient. To provide better quality connectivity and energy consumption are imminent issues in critical and disaster communication scenario. In Fig. 7.2, we explain the role of interconnectivity among the different emerging

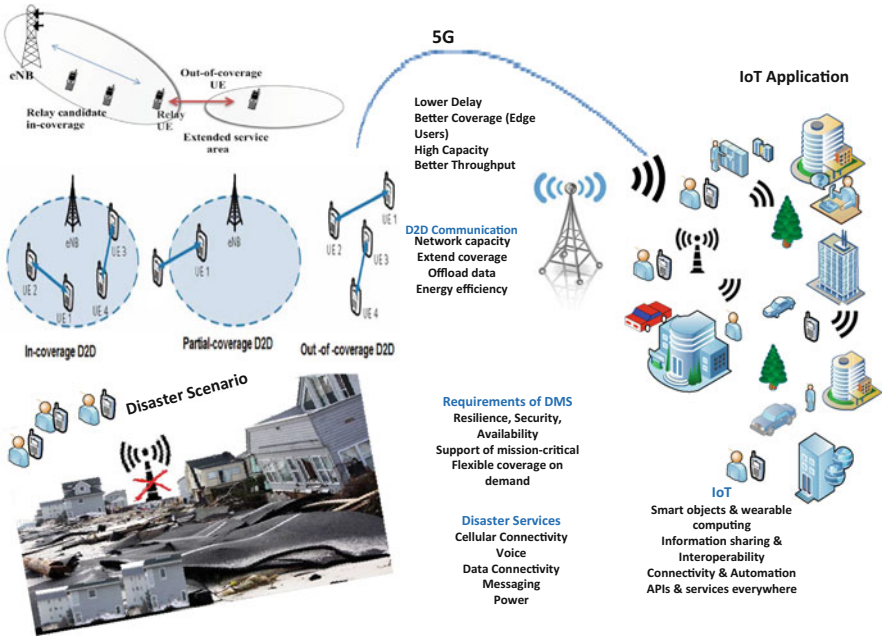

Fig. 7.2 Schematic of Internet of Things (IoT)-based disaster management system (DMS) and 5G wireless networks

technologies. Our basic aim is here to deliver the concept of IoT architecture based on D2D communication and to incorporate in the proposed architecture of 5G for improved DMS.

7.3.1 Device to Device

In LTE technology, which is considered to be the basis for next-generation public safety broadband networks, it is essential to consider the addition of certain key features and functionalities such as direct D2D broadcast. This is mainly because of the need to better support the public safety use cases and requirements. D2D communications are considered as a key enabling technology in future cellular networks and, thus, it has become an intriguing topic for research [9]. It refers to a state-of-the-art technology that enables user equipments (UEs) to communicate directly with each other without using the access network (i.e. eNodeB). One of the most vital functions of D2D communication is the proximity service (ProSe). This is indeed an inspiring technique for critical communications, e.g. in public safety or emergency situations. Feasibility of D2D proximity services is studied with the purpose of recognizing use cases and potential requirement for discovery

and communications between UEs that are in proximity, including network operator control and direct communication [10].

- In-coverage: This scenario indicates that all the considered UEs are within eNB coverage (in-coverage UEs) to receive services/signals from an eNB.
- Partial-coverage: This scenario indicates that some UEs are within eNB coverage, while other UEs are outside eNB coverage.
- Out-of-coverage: This scenario indicates that all the considered UEs are outside eNB coverage (out-of-coverage UEs) and cannot receive services/signals from an eNB.

In public safety scenarios, large areas may lose cellular coverage when rescue teams and public need it most. D2D communications can be a good solution for extending the coverage of those sites that remain active in a partial coverage scenario. In such a critical condition, energy efficiency, timely response, and network connectivity are important factors for long and reliable communication. D2D communication can enable direct communication between first responders and rescue teams even if they are out of the coverage areas of the LTE system. Such a situation can happen when base stations (BSs) experience a power fault or damage due to a disaster. D2D communication is useful not only for local communication but also for communication between a BS and out-of-coverage user equipment (UE). This facility can be enabled by D2D-based relay [9], an out-of-coverage UE can communicate with a neighbour UE via D2D communication; later, UE becomes a relay to the BS as shown in Fig. 7.2. These features were standardized in 3GPP Release 13 [11]. We can also describe the device-level communication into four main types which can be more beneficiary in disaster situations [12].

- *Device relaying with BS controlled link formation*—Applicable for a device which is at the edge of a cell within coverage area.
- *Direct D2D communication with BS controlled link formation*—Source and destination devices exchange data with each other without the contribution of a BS, but they are reinforced by the BS for link formation.
- *Device relaying with device controlled link formation*—Here, BS is intricate neither for link formation nor for communication; devices are responsible for synchronizing connectivity and communication with relays.
- *Direct D2D communication with device controlled link formation*—The devices create direct link among source and destination and the link formation is itself controlled by the devices deprived of any support from the BS.

The upcoming 5G systems are envisioned to have the crucial proficiencies like network flexibility, (re)configuration, and resilience and, therefore, expected to play a key role in refining disaster situation communications. In the context of 5G wireless networks as well, the D2D technology can be employed for emergency situations. The key requirements for a public safety system are to provide dynamic radio resource management in order to retain the lifetime of the end devices [13].

7.3.2 *Internet of Things*

The IoT is a capable technology that can be used to solve some of the issues we face in disaster situations like electricity outages, last mile connectivity loss, network traffic congestion, etc. However, we come more optimistic when the ITU, the UN's specialized agency for information and Communication technologies, defined IoT as, "A global infrastructure for the information society, enabling advanced services by interconnecting (physical and virtual) things based on existing and evolving interoperable information and communication technologies" [14]. Similarly, IoT-based D2D communications not only have the potential to help us in disaster management, but it can also help to offload network traffic even in normal situations.

The key features of D2D drive the proposal of this chapter, which aims at investigating the adoption of D2D technology network to support mobile IoT devices. Certainly, in these cases, proximity-based transmissions can improve support as well as sharing of devices and their capabilities. Furthermore, whenever the network experiences a lack of radio resources because of traffic overload, UEs in proximity may exploit their direct connection for connectivity and sharing data.

Shown in Fig. 7.2 is our analysis on 5G-oriented IoT scenario services based on D2D transmission for network edge UEs supporting low latency and delay content. Additionally, Fig. 7.2 shows that heterogeneous network and various types of devices will be connected together to provide connectivity for better communication and services, for example: (1) working BS of a cellular network (nearest to disaster area), (2) any working wired connection (fibre optic), and (3) devices having a capability to communicate with each other or satellite.

7.4 Architecture for Disaster Management System

We consider disaster scenario as shown in Figs. 7.3 and 7.4, where one BS is transmitting at full power while in lower zone BS are nonfunctional. The dashed black line separates covered and uncovered areas. As mentioned before, in emergency situations users located in nonfunctional area can benefit from D2D UE proximity services. D2D relay allows multi-hop links to be formed between two D2D devices or between the cellular infrastructure and an endpoint UE. Moreover, we know that D2D with UE relay enhances data throughput of edge user and can be used to connect far-away UEs with cellular coverage to the BS which supports the extension of cellular coverage.

In this paper, we consider scenario of multi-hop D2D communication with two modes.

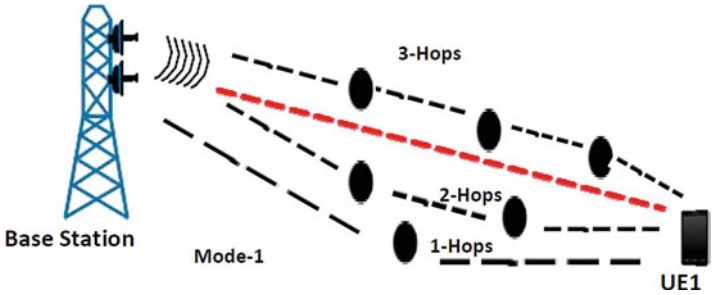

Fig. 7.3 System model for device to device (D2D) extended out-of-coverage scenarios

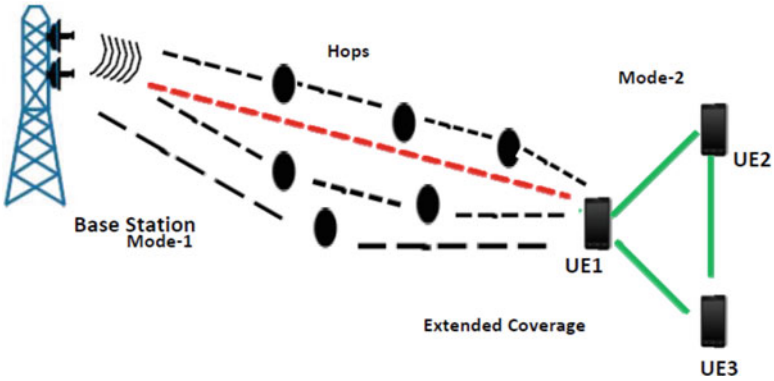

Fig. 7.4 System Model for Multi-hop D2D and extended out-of-coverage scenarios

7.4.1 Mode-1

Four predefined paths in which UEs intend to exchange information with each other using multi-hop D2D and extend communication from coverage area to nonfunctional area.

7.4.2 Mode-2

Using multi-hop D2D communication to cover nonfunctional area and exchange information.

Routing in wireless multi-hop communication is a well-addressed research area and it has been studied extensively in the past and several algorithms have been designed and proposed for supporting D2D communication. However, these algorithms are subject to their specific characteristics and may apply on specific scenarios. Let us consider our scenario of Fig. 7.3 and we analyse some metrics

like number of hops and distance between them, energy and capacity efficiency, extendable coverage area, and support communication among UEs by finding quick and shortest path for communication. We studied two algorithms (shortest path routing (SPR) and interference-aware routing) and here in this paper on the bases of our matrices we used the SPR in our scenario.

7.4.2.1 Shortest Path Routing Algorithm

In SPR, each D2D UE knows its location through GPS and other wireless localization means (i.e. wireless fingerprinting and triangulation) [15]. Now, we outline the step-by-step D2D algorithm needed to achieve SPR from a generic UE pair (source to destination). Assuming that SPR is chosen as the routing strategy, the multi-hop algorithm works in the following manner:

- Source UE is able to detect the neighbouring UEs that can make successful transmissions to destination UE with some arbitrary threshold of signal to noise ratio (SNR) required for successful connection.
- Then, source UE sends relay request to all the neighbouring UEs in its communication range.
- After receiving information of potential relay UEs, source UE sends data packet to the relay that is the closest to the destination UE.

This process will continue until the destination UE is reached.

7.4.2.2 Network Configuration

For communication purpose, we consider a scenario of D2D underlying a cellular network where UE1 intends to communicate with active BS which is nearer to its position. According to Fig. 7.3 in mode-1, UE_1 is located in nonfunctional area and tries to connect with the BS_1 directly or by using different UEs as relay. Furthermore, for simplicity four different pathways have been predefined:

- Path-1. Direct communication with BS (0-hop) not possible because of the distance
- Path-2. Communication with BS using one UE as relay (1 hop)
- Path-3. Communication with BS using two UEs as relay (2 hops)
- Path-4. Communication with BS using three UEs as relay (3 hops)

Each communication hop is occurring in different time frames $T_{f\,p_t,l_n,h_o}$, where p_t, l_n, and h_o define path number, link number for a specific route, and hop number within specific link, respectively. Let us assume that every communication link L_{p_t,l_n} is composed by H hops h_{p_t,l_n,h_o}. So, every hop which is h_{p_t,l_n,h_o} from the same route resides a different time frame resource. Subsequently, interferences are only possible with the same hops h_{p_t,l_n,h_o} duration from the different routes. Here, SPR algorithm is adopted for simulation purpose. The path selection combines SPR with channel quality information so that the route selected for communication will contain nodes with the best channel that follows SPR rules.

7.4.3 Energy Efficiency and Spectral Efficiency

Energy efficiency (EE) and spectral efficiency (SE) have become two of the dynamic requirements in the design of future public safety communications architecture system. Due to the expected limitations in energy and spectral resources during emergency situations, we need to use D2D communication in such a way to increase the cellular spectrum efficiency. This can be performed and controlled with proper interference management and resource allocation. As shown in Fig. 7.3, every communication path within a route is independent and follows in different time slots, we can present capacity of mode-1 in link l ($C_{1,l}$) same as [16] in Eq. (7.1):

$$C_{1,l_n} = \sum_{i=1}^{N} B_w \log_2 \left(1 + \text{SINR}_{l_n,i}\right) \tag{7.1}$$

where B_w is communication bandwidth and $\text{SINR}_{l_n,i}$ the signal-to-interference-plus-noise ratio for every hop i within the link l_n. As noise can be considered negligible, $\text{SINR}_{l_n,i}$ becomes signal-to-interference ratio $\text{SIR}_{l_n,i}$ and is calculated by Eq. (7.2):

$$\text{SIR}_{l_n,i} = \frac{P_d r_{l_n,i}}{I_t r_{l_n,i}} \tag{7.2}$$

$P_d r_{l_n,i}$ and $I_t r_{l_n,i}$ are D2D communication received power and received power with interference in hop i and link l_n, respectively. At the end, total capacity for all multi-hop communications in mode-1 will define EE and SE performance for multi-hop D2D communications in our model. The overall instantaneous transmission vector for energy efficiency $\text{EE}(1, l_n)$ comprised by the EE elements from every link L_{1,l_n} is defined by Eq. (7.3):

$$\text{EE}_{1,l_n} = \frac{C_{1,l_n}}{H p_{tx}^{\text{ue}}} \tag{7.3}$$

where H represents the number of hops on every link L_{1,l_n} and p_{tx}^{ue} the maximum transmission power of the UE. In the same way, we present SE vector for mode-1 in Eq. (7.4):

$$\text{SE}_{1,l} = \frac{C_{1,l_n}}{B_w}. \tag{7.4}$$

According to scenario mode-2, distance between UE_1 and UE_2 is d_u and we know that UE_3 is in the middle of both $d_u/2$. The transmission power of UE_t, $t = 1, 2, 3$, is the set of P_{ue} and channel gain between UE_t and Bs is C_{gt}.

In this mode, we are extending the coverage in nonfunctional area and UE_1 and UE_2 will exchange information via UE_3 within two time slots. In the first time slot, both UE_1 and UE_2 transmit to UE_3 at the same time. The received signal at UE_3 is

Table 7.1 Simulation parameters

Parameters	Values
System bandwidth	[2 3 4 5 10 12] MHz
Carrier frequency	700 MHz
Subfram duration	1 ms
UE max. trans. power	23 dBm
Min distance UE2UE	3 m
Resource groups (RGs)	6 14 25 50 75 100

$$y_{M_2} = \sqrt{P_{\text{ue}}/2}\left(C_{gt_1}x_1 + C_{gt_2}x_2\right) + n \tag{7.5}$$

where x is the transmitted data from UE_t; $n_t, t = 1, 2$, is the additive Gaussian noise. The average energy efficiency and SNR for mode-2 is calculated same as [17],

$$\begin{aligned}\text{EE} &= \mathbb{E}_{C_{gt1,0}, C_{gt2,0}\{ee\}} \\ &\geq \frac{2B_w}{3P_{\text{ue}}}\left[1 + \frac{1}{\text{SNR}}\log_2\left(1 + \text{SNR}\right) - \log_2 e\right].\end{aligned} \tag{7.6}$$

7.4.4 Simulation Results

In this section, we present the simulation results to show the performance of an efficient resource management using multi-hop D2D communication. As BS knows every channel in the field, it can easily determine which node can perform best in the area. In order to select relay shortest path toward BS, we used SPR algorithm. Table 7.1 represents the parameters used to evaluate EE and SE.

In Fig. 7.5, we plot the energy efficiency as a function of spectral efficiency for variations in resource groups (RG). Simulation results show that increasing the size of RGs will benefit both energy efficiency and spectral efficiency of the multi-hop D2D transmissions. Energy efficiency rises for the same UE transmission power. This is due to the fact that UE acts as a relay which influences the system capacity to improve due to the shorter communication path, which results better channel conditions. Also, these results show that the use of shorter D2D links while increasing the number of hops guarantee or even improve the QoS. The spectral efficiency increases when using higher number of RGs for the same number of hops. This is due to the fact that the increase in size of RGs is associated to bandwidth increase (Table 7.1). However, when using higher number of hops, we also notice that energy efficiency is bounded by a certain limit. This limitation is due to higher aggregate interference effects in multi-hop communications.

Figures 7.6 and 7.7 show the energy efficiency and spectral efficiency vs number of hops for different size of RGs. It is observed that energy efficiency has dramatical improvements with higher number of hops for every mode of communication whereas spectral efficiency has nearly double changed for every mode of communication. This confirms the first observations. Figure 7.8 shows

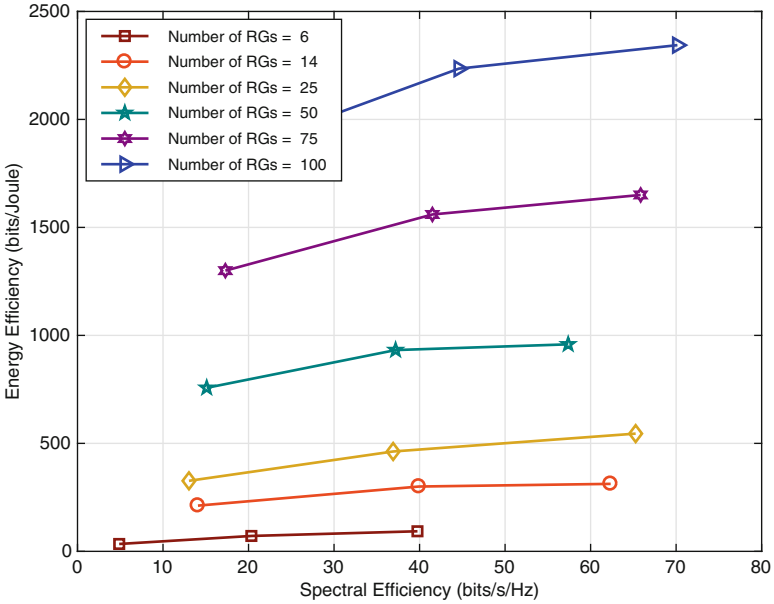

Fig. 7.5 Energy efficiency performance vs spectral efficiency while increasing resource group number

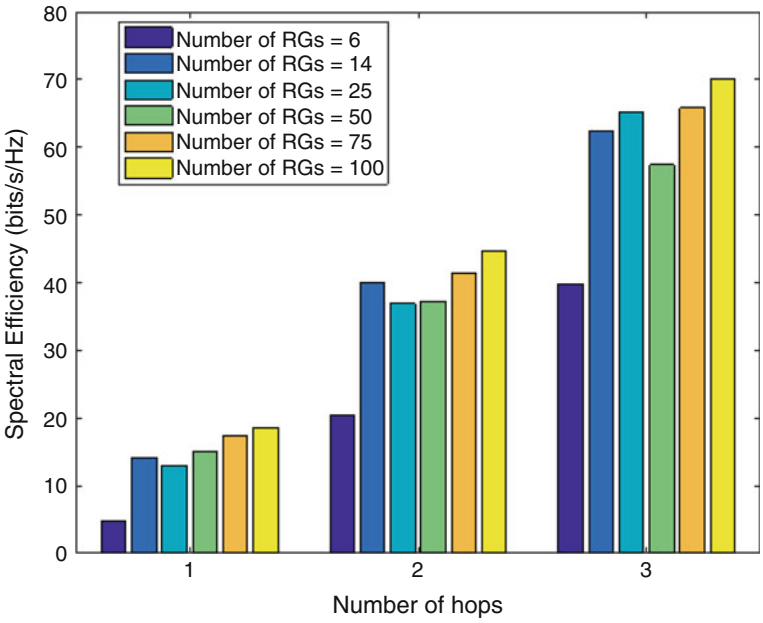

Fig. 7.6 Spectral efficiency as a function of number of hops for different resource group values

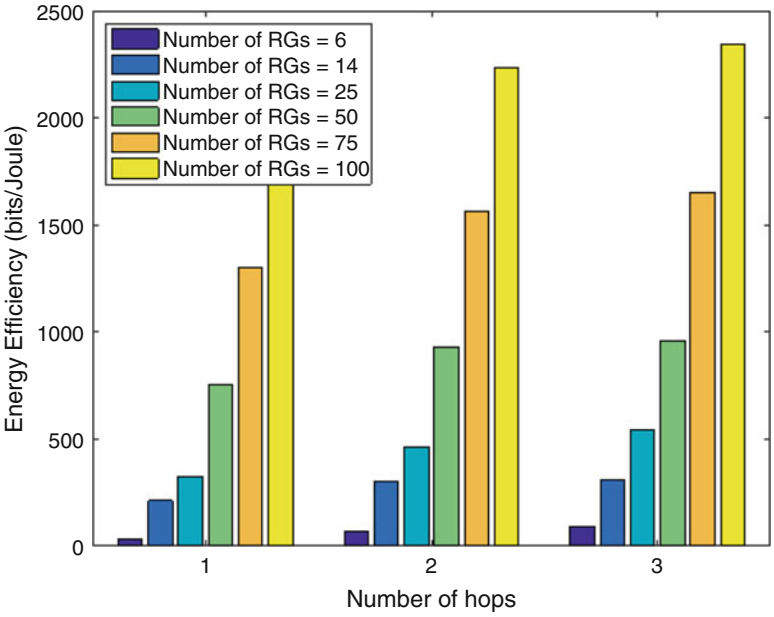

Fig. 7.7 Energy efficiency as a function of number of hops for different resource group values

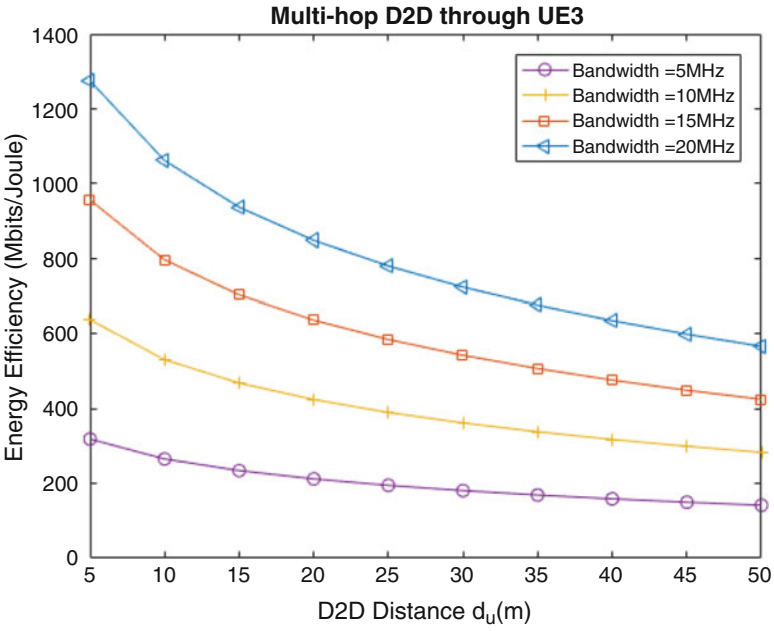

Fig. 7.8 Multi-hop D2D communication in noncoverage area

the simulation result in which we analysed the energy efficiency of the mode-2 and quantify the performance of the D2D multi-hop communication with various D2D distance settings. We observed that when the distance between the UEs is less, the energy efficiency is high and as the distance between the UEs increases the EE decreases. The rate at which the EE decreases depends upon the available bandwidth. The energy efficiency at 10 m for 20 MHz bandwidth is (approximately) two times greater than 10 MHz bandwidth. Even through the EE drops exponentially as the bandwidth is reduced, still the performance of our proposed network is better and considerable for communication in critical situations.

7.5 Conclusion

In this paper, we highlighted the key enabling technologies suitable for DMS-based IoT and D2D communications. We presented how D2D communication may enhance the IoT cooperation at the edge of the network for supporting public safety applications. D2D communications is a technology that permits mobile UEs to relay information to each other, without accessing the cellular network. We consider the cases where there has been a disaster and the cellular network is damaged and/or congested. Emergency technologies desire to coexist with the conventional cellular communications. SPR algorithm is used to select the best and quickest path for better performance. We consider the multi-hop D2D communications underlying cellular networks. Results show the average energy efficiency of these transmission types and compare the performance. Performance studies show that when the two UEs forming the D2D pair are far away from the BS, multi-hop D2D gives a higher energy efficiency and a better throughput than the other direct mode. In addition, when the two UEs are close to each other, multi-hop D2D still gives a higher energy efficiency than the direct D2D mode, with a comparable throughput performance.

References

1. Nepal Earthquake: Eight Million People Affected, UN Says. (Apr 2015). Available: www.bbc.com/news/worldasia-32492232. Accessed 13 Dec 2017
2. P.P. Ray, M. Mukherjee, L. Shu, Internet of Things for disaster management: state-of-the-art and prospects. IEEE Access Intell. Syst. Internet Things **5**, 18818–18835 (2017)
3. Index, Cisco Visual Networking, Global mobile data traffic forecast update, 2014–2019 white paper, 2015
4. E. Olshannikova, A. Ometov, Y. Koucheryavy, T. Olsson, Visualizing Big Data with augmented and virtual reality: challenges and research agenda. J. Big Data **2**(1), 22 (2015). Available: http://dx.doi.org/10.1186/s40537-015-0031-2
5. Ericsson Research Blog, 5G radio access for ultrareliable and low-latency communications, Technical Report, May 2015. Available: https://www.ericsson.com/research-blog/5g/5g-radio-access-for-ultra-reliable-and-low-latency-communications/

6. G. Apostolos, K. Konstantinos, N. Aikaterini, F. Foukalas, T. Khattab, Energy efficient spectrum allocation and mode selection for mission-critical D2D communications, in *IEEE Conference on Computer Communication Workshops (INFOCOM WKSHPS)*, April 2016
7. J.S. Huang, Y.N. Lien, Challenges of emergency communication network for disaster response, in *2012 IEEE International Conference on Communication Systems (ICCS)*, Nov 2012, pp. 528–532
8. A. Sinha, P. Kumar, N.P. Rana, R. Islam, Y.K. Dwivedi, Impact of internet of things (IoT) in disaster management: a task-technology fit perspective. Ann. Oper. Res. 1–36 (2017)
9. K. Ali, H.X. Nguyen, P. Shah, Q.T. Vien, N. Bhuvanasundaram, Architecture for public safety network using D2D communication, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC)*, Doha, Apr 2016
10. K. Ali, H. X. Nguyen, P. Shah, Q.T. Vien, E. Ever, D2D multi-hop relaying services towards disaster communication system, in *Proceedings of the 24th International Conference on Telecommunications (ICT)*, Limassol, Cyprus, Aug 2017
11. 3GPP, TS 36.331 V13.1.0, Technical Specification Group Radio Access Network; Radio Resource Control (RRC); Protocol specification (Release 13), Mar 2016
12. T. Sakano, S. Kotabe, T. Komukai, T. Kumagai, Y. Shimizu, A. Takahara, T. Ngo, Z.M. Fadlullah, H. Nishiyama, N. Kato, Bringing movable and deployable networks to disaster areas: development and field test of MDRU. IEEE Netw. **30**(1), 86–91, (2016)
13. G. Fodor, S. Parkvall, S. Sorrentino, P. Wallentin, Q. Lu, N. Brahmi, Device-to-device communications for national security and public safety, in *Perspectives of the Next Generation Mobile Communications and Networking, IEEE Access: 5G Wireless Technologies*, vol. 2, 18th Dec 2014, pp.1510–1520
14. ITU-T, Internet of Things Global Standards Initiative, http://www.itu.int/en/ITU-T/gsi/iot/Pages/default.aspx
15. H. Yuan, W. Guo, S. Wang, Emergency route selection for D2D cellular communications during an urban terrorist attack, in *Proceedings of the IEEE International Conference on Communications*, Sydney, Mar 2014
16. L. Babun, A.I. Yurekli, I. Guvenc, Multi-hop and D2D communications for extending coverage in public safety scenarios, in *40th Annual IEEE Conference on Local Computer Network*, Florida, Oct 2015
17. W. Lili, R. Hu, Q. Li, W. Geng, Energy-efficiency of multihop device-to-device communications underlaying cellular networks, in *Proceedings of the IEEE International Conference on Communications*, Sydney, Jun 2014 (IEEE, Los Alamitos, 2014)

Chapter 8
Internet of Things and Statistical Analysis

Ali Cevat Taşıran

8.1 Introduction

IoT is a combination of embedded technologies regarding wired and wireless communications, sensor and actuator devices, and the physical objects connected to the Internet. It is expected that 25–50 billion devices are connected to the Internet by 2020 [1].

The explosive increase in the number of devices connected to the IoT is the source of the greatest sources of new data. Analysis of these data sets, in practice, is done using various data science techniques. Data science is the compounding of different disciplines of the sciences that includes data mining, machine learning, and other techniques to get forms and fresh insights from data. The different algorithms are applied to the information in order to draw out higher-level information from IoT data sets. These techniques include a broad range of algorithms applied in different domains and comprise both unsupervised and supervised methods. The procedure of applying data analytic methods to particular areas involves data models such as neural networks, classification, clustering methods, and applying efficient algorithms that fit with the data characteristics.

In this article, we focus on statistical analysis of the collected data sets and look at the similarities and differences of data mining and statistical methods for IoT systems. By explaining the need for both descriptive and inferential statistical methods, we show the inadequacy of using only unsupervised data mining techniques for IoT data sets. We then focus on statistical inference approaches, such as the design-based and model-based inferences, and discuss the role of explanatory versus forecasting statistical models in an IoT environment. Lastly, we demonstrate

A. C. Taşıran (✉)
Middle East Technical University, Northern Cyprus Campus, Güzelyurt, Turkey
e-mail: atasiran@metu.edu.tr

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_8

the danger of making forecasts using only unsupervised techniques by revealing open challenges as future research directions.

The structure of this paper is as follows. Section 8.2 displays the historical growth of IoT. Section 8.3 reports the increasing volumes of large data sets and the need for online analysis. For this need, the recognized responses in data science are reported in Sect. 8.4 and in Statistics are explained in Sect. 8.5. The final section concludes this chapter.

8.2 Growth of IoT

IoT creates a world where physical objects are integrated into information networks. IoT, the network of connected "smart" devices that communicate over the Internet, is transforming how we live and act. On farms, wireless IoT sensors can transmit information about soil moisture and nutrients to agricultural experts across the country. IoT alarm systems, equipped with batteries that last for years, provide homeowners with long-term protection. Wearable fitness devices can monitor activity levels and provide feedback on heart rate and respiration.

A network comprised of physical objects that are capable of collecting and sharing electronic data. IoT includes a wide variety of "smart" devices, from industrial machines that transmit data about the production process to sensors that track information about the human body. Much, these devices use Internet Protocol (IP), the same protocol that identifies computers over the World Wide Web and permits them to intercommunicate with one another.

The term "Internet of Things" is attributed to Kevin Ashton of Procter & Gamble, who in a 1999 article used the phrase to describe the role of RFID tags in creating supply chains more efficient. At the time, the idea of electronically gathering data in a production facility or warehouse and linking it to computers for analysis was still very new. In recent years, the number of smart sensors has exploded. By one estimate, there will be 50 billion devices connected to the Internet by the year 2020.

IoT paradigm has converged several technologies in terms of sensing, computing, information processing, networking, and controlling intelligent technologies. As the computation and the communication process of heterogeneity hidden networks involve intelligent decision making, the human-to-machine perception of the recognition of the large-scale IoT environment is bridged as a central challenge for the IoT in literature.

Growth of IOT is not only visible in technical areas, but also can be noticeable in the economic scene. The global IoT market is projected to grow from $2.99T in 2014 to $8.9T in 2020, attaining a 19.92% Compound Annual Growth Rate (CAGR). Industrial manufacturing is forecast to increase from $472B in 2014 to $890B in global IoT spending. Health care and life sciences are projected to increase from $520B in 2014 to $1.335T in 2020, attaining a 17% CAGR. Source: Statista, Size of the Internet of Things market worldwide in 2014 and 2020, by industry (in billion US dollars).

8.3 Increasing Volumes of Large Data Sets, the Need for Online Analysis, and the Privacy Challenges

The explosive growth in the number of devices connected to the IoT and the exponential increase in data consumption only reflect how the growth of large data sets is perfectly overlapping with that of IoT. In the digital and computing world, information is generated and collected at a rate that rapidly exceeds the boundary range. Currently, over two billion people worldwide are connected to the Internet, and over five billion individuals own mobile phones. By 2020, 50 billion devices are expected to be connected to the Internet. At this point, predicted data production will be 44 times greater than that in 2009. As information is transferred and shared at light speed on optic fiber and wireless networks, the volume of data and the speed of market growth increase.

Since IoT will be among the greatest sources of new data, data science is used to make IoT applications more intelligent. In addition to increased volume, the IoT-generated large data sets exhibit a variety of multiple modalities and varying data quality. Intelligent processing and analysis of these data sets is the key to developing smart IoT applications. However, the fast growth rate of such large data generates numerous challenges, such as transfer speed, diverse data, security, and the need for online analysis. Analysis of these data sets, in practice, is done using various data science techniques.

Analysts of the IoT-generated large data sets are indeed able to apply smart algorithms, and artificial intelligence to data can discover hidden insights relevant in various scenarios. Some examples are from data-driven decision optimization (e.g., optimization of police proactive tactical decision making to reduce crime), and health care (e.g., patients' risk for certain rare diseases and tracking the spread of influenza viruses), to improving our understanding of human behavior in certain sociotechnical environments. However, as data observations are increasingly viewed as a commodity and new form of currency, the emergence of such huge amounts of aggregated data and their linkability to other datasets clearly introduce a whole new set of privacy challenges, such as threats to people's right to informational self-determination, unfair discrimination, and other prejudicial outcomes [2].

8.4 Data Science

Since IoT will be among the greatest sources of new data, data science will make a great contribution to make IoT applications more intelligent. Data science is the compounding of different disciplines of sciences that uses data mining, machine learning, and other techniques to get forms and fresh insights from data. The different algorithms are applied to the information in order to draw out higher-level information from IoT data sets. These techniques include a wide scope of algorithms applied in different domains and comprise both unsupervised and supervised

methods. The procedure of applying data analytic methods to particular areas involves data models such as neural networks, classification, clustering methods, and applying efficient algorithms that fit with the data characteristics.

Data mining can be defined as the process of selection, exploration, and modeling of large databases in order to discover patterns and models that are unknown a priori. In this sense, data mining is the art and science of intelligence data analysis. The aim in the data mining is to discover meaningful insights and knowledge from data. Discoveries are often expressed as models which can be used to understand the world and to make predictions. Many of the methodologies used in data mining come from two branches of research, one developed in the machine learning community and the other developed in the statistical community. The computational process of discovering patterns in large data sets involves methods at the intersection of artificial intelligence, machine learning, statistics, and database systems. There are data mining teams working in business, government, financial services, biology, medicine, risk and intelligence, science and engineering.

In the 1960s, data mining was a pejorative term among statisticians, used to describe the improper analysis of data that leads to simply coincidental findings. This is now called "Data Dredging," "Data Snooping" or "Data Fishing." "Data mining as a named discipline emerged at the end of the 1980s. The first data mining workshops in the early 1990s attracted the database community researchers. Data mining focuses on the discovery of previously unknown properties of data while machine learning focuses on prediction, based on known properties learned from the training data.

8.4.1 Plenty of Techniques and Algorithms

A wide range of techniques and algorithms are used in data mining. These techniques and algorithms are classified as unsupervised and supervised techniques below and are used to make the following classes of tasks:

- Anomaly detection—the identification of unusual data records.
- Association rule learning—searches for relationships between variables (market basket analysis).
- Clustering—the task of discovering groups and structures in the data.
- Classification—the task of generalizing known structure to apply new data.
- Regression—modeling data for explanatory and predictive purposes.
- Finding patterns in the training data set and evaluating the model in a test data set on which the data mining algorithm was not trained.
- If the learned patterns do not meet the desired standards, then it is necessary to reevaluate and change the preprocessing and data mining steps. If the learned patterns do meet the desired standards, then the final step is to interpret the learned patterns and turn them into knowledge.

The main techniques for data mining are listed below. In the unsupervised learning problem, we observe only the features and have no measurements of the outcome. Our task is rather to describe how the data are organized or clustered.

Descriptive Data Mining (Unsupervised Methods).

- Clusters—Kohonen Maps
- Association Rules
- Principal Components
- Independent Component Analysis

In the supervised learning problem, we predict future developments. We have a training set of data, in which we observe the outcome and features. It is called "supervised" because of the presence of the outcome variable to guide the learning process. In a typical scenario, we have an outcome measure, usually quantitative (such as a stock price) or categorical (such as heart attack/no heart attack), that we wish to predict future developments based on a set of current features.

Data Mining (Supervised Methods).

- Decision Trees
- Random Forests
- Boosting
- Support Vector Machines
- Neural Networks
- Linear Regressions and Logistic Models
- Time Series Analysis and Mining
- Bayesian Methods
- Text Mining
- Social Network Analysis

8.4.2 Predictive Modeling

Once data scientists gather the sample data, they must select the right model. Linear regressions are among the simplest types of predictive models. Linear models essentially take two variables that are correlated—one independent and the other dependent—and plot one on the x-axis and one on the y-axis. The model applies a best-fit line to the resulting data points. Data scientists can use this to predict future occurrences of the dependent variable.

Very well known, another type of predictive modeling is logistic regression where the dependent variable can take only two values, either 1 for success or 0 for failure. The distribution function of the dependent variable is binomial; thus, one cannot use linear regression in such situation.

Other more complex predictive models include decision trees, k-means clustering, and Bayesian inference, to name just a few potential methods. The most complex area of predictive modeling is the neural network. This type of the machine

learning model independently reviews large volumes of labeled data in search of correlations between variables in the data. It can detect even subtle correlations that only emerge after reviewing millions of data points. The algorithm can then make inferences about unlabeled data files that are similar in type to the data set it was trained on. Neural networks form the basis of many of today's examples of artificial intelligence (AI), including image recognition, smart assistants, and natural language generation (NLG).

8.4.3 The Danger of Lack of Theoretical Knowledge

First, no matter how good data are, collected data are noisy and biased in most cases. The multi-faceted nature of data collection makes a big impact on the way data are collected and how they will be mined. Inventing algorithms that measure and quantify the quality of data from different resources, and filtering noise and bias of the data, will be an inevitable part of working with large data sets. It is inadequate using only unsupervised data mining techniques for making forecasts of IoT since these methods are only descriptive in character.

Besides the quality of data, there is something more fundamentally wrong about using only causal inference algorithms which are based only on gathering information from the observed data sets. These observations are the only information source of predictive modeling. A good scientific model as an explanatory and a predictive model needs not only information from observed data sets but also a guidance from theoretical models. Parc [3] gives the following example: "Medicine has perhaps one of the longest histories among the different branches of science. The accumulated knowledge in literature and medical and pharma trials is enormous today. Medical knowledge will continue to expand. Big data in medicine can give us interesting insights only if it goes hand-in-hand with the medical knowledge. Looking for causal relationships between different diseases in big EMR data will lead to robust results only if the existing medical knowledge, e.g. causal relationships between diabetes and kidney diseases, is incorporated into our machine learning algorithms."

This job of combining deterministic theoretical models with stochastic models of data observations is done using statistical models as we will see below in Sect. 8.5.

8.5 Statistical Modeling

Statistics is the science of observing data and making inferences about the characteristics of a random mechanism that has generated data. It is also called as science of uncertainty. Statistical Analysis can be seen as Descriptive Statistical and Explanatory Statistical Analysis. In *Descriptive Statistical Analysis*, data observations are collected, organized, summarized and the descriptive results are displayed.

In *Explanatory Statistical Analysis*, the task of examining all observations in populations is very difficult or impossible since the populations are very big. Then samples are drawn from populations, and inferences are made from sample observations into population characteristics.

In different scientific disciplines, theoretical models are used to analyze substantive problems. *Theoretical models* are deterministic functions, but in the real world, the relationships are not exact and deterministic rather than uncertain and stochastic. We thus employ either statistical models to make inferences from sample data sets into real populations or distribution functions to make approximations to the actual processes that generate the observed data (conceptual populations). The process that generates data is known as the data generating process (*DGP* or *Super Population*). For example, in the Sciences, to study the relationships between different variables, one can estimate *statistical models*, which are built under guidance of the theoretical deterministic models and by taking into account the properties in the data generating process.

(a) Inference approaches: Classical versus Bayesian

These are probability distributions which in turn can be characterized by some unknown parameters. The statistical theory that is used for such analyses is called as *Classical Inference*, one which will be followed in this course. It is based on two premises:

1. The sample data constitute the only relevant information.
2. The construction and assessment of the different procedures for inference are based on long-run behavior under similar circumstances.

The other type of statistical inference is called *Bayesian inference* where sample information is combined with prior information. This is expressed of a probability distribution known as the prior distribution. When it is combined with the sample information, then a posterior distribution of parameters is obtained. The resulting posterior probability is proportional to the likelihood (sample information) times prior probability. The inverse of an estimator variance is called as the *precision*. In Classical Inference, one can use only parameter's variances, but in Bayesian Inference, one has access to both sample precision and prior precision.

(b) Classical Inference as Design-based and Model-based approaches

Using parameters of estimated statistical models, one makes generalizations about the characteristics of a random mechanism that has generated data. In both Engineering and Social Sciences, we use observed data in the samples to draw conclusions about populations. Populations are either real from which the data came or conceptual as processes by which the data were generated. The inference in the first case is called *design-based* (for experimental data) and used mainly to study samples from populations with known frames. The inference in the second case is called *model-based* (for observational data) and used mainly to study stochastic relationships.

When one speaks statistical models to be estimated or tested, it refers to sets of DGPs in Classical Inference context. In design-based inference, the attention is restricted to a particular sample size and it characterizes a DGP by the law of probability that governs the random variables in a sample of that size. In model-based inference, it refers to a limiting process in which the sample size goes to infinity, it is clear that such a restricted characterization will no longer suffice. In asymptotic theory, the DGPs in question must be stochastic processes.

A *stochastic process* is a collection of random variables indexed by some suitable index set. This index set may be finite, in which case we have no more than a vector of random variables, or it may be infinite, with either a discrete or a continuous infinity of elements. In order to define a DGP, one must be able to specify the joint distribution of the set of random variables corresponding to the observations contained in a sample of arbitrarily large size. This is a very strong requirement. In any other empirical discipline for that matter, researchers deal with finite samples. In the process of estimating a statistical model, what we are doing is to try to obtain some estimated characterization of the DGP that actually did generate the data. We may therefore say that the DGP is either *completely characterized* or *partially characterized* by the model parameters.

We call the model with its associated parameter-defining mapping as a *parametrized model.* The main task in our practical work is to build the association between the DGPs of a statistical model and the model parameters.

(c) Statistical models as bridges between theoretical and practical worlds

A *statistical model* therefore takes center stage in the mathematical framework for modeling a stochastic phenomenon. The simple statistical model, first referred to by Fisher [4], has two interrelated components: probability model and sampling model. The probability model specifies the distribution function of the dependent variable, which will be used as an explained variable in a causal relationship. The sampling model characterizes observations of the random sample.

The probability model specifies a family of densities defined over the range of values of the random variable; one density function for each value of the parameter, as the latter varies over its range of values of parameter space.

A random sample as a set of random variables satisfies two probabilistic assumptions:

(a) Independence of probability density functions for each variable
(b) The identical distribution of the random variables

A simple statistical model builds a bridge between the theoretical concepts and the corresponding chance regularity patterns exhibited by observed data. Theoretical models guide researchers to choose important variables in a simple deterministic framework. A theoretical model helps to obtain a parsimonious statistical model which does not cause to omitted variable bias or to inflate the variances of estimated parameters. Thus, the choice of a statistical model, given a particular data set, constitutes the most crucial and the most difficult decision facing the researcher. In general, an inappropriate choice of a statistical model will invalidate any statistical inference results built upon the premises of the postulated model.

In design-based inference, the specification of the statistical model is decided upon at the design stage and there is therefore no need to utilize the resulting experimental data to choose the model.

This opportunity does not arise in the case of nonexperimental (observational) data and thus we need to utilize the available observed data in our attempt to specify an appropriate statistical model which constitutes an adequate summary of the systematic relationships.

The sample survey inference is historically concerned with finite-population parameters, that is, functions (like means and totals) of the observations for the individuals in the population. In scientific applications, however, interest usually focuses on the "super population" parameters associated with a stochastic mechanism hypothesized to generate the observations in the population rather than the finite-population parameters.

In general, a stochastic process is used to observe the values in the samples. The distributions of the observations generated by the process will be studied using histograms, probability mass and density functions, and cumulative distribution functions. For these purposes, chi-square tests will be used to compare histograms and density functions of the observed time use. Following that, Kolmogorov–Smirnov and Anderson–Darling tests can be employed to compare the cumulative distribution function of observed and theoretical distributions. Doddapaneni et al. [5], for instance, examines whether or not the assumption of exponential distribution in wireless sensor networks holds.

With the completion of the data analysis part, depending on information, probability models will be presented for the dependent variables. Bayesian analysis will in turn be employed in order to interact with the system in a dynamic fashion where prior probabilities are fed into the system and the resulting posterior probabilities are provided by the system.

8.6 Conclusions

During the last decade, we witnessed an explosive growth in the number of devices connected to the IoT and the growth of large data sets is perfectly overlapping with that of IoT. Analysis of these data sets, in practice, is done using various data science techniques. Data science is the compounding of different disciplines of sciences that uses data mining, machine learning, and other techniques to get forms and fresh insights from data.

The different algorithms are applied to the information in order to draw out higher-level information from IoT data sets. These techniques include a wide scope of algorithms applied in different domains and comprise both unsupervised and supervised methods. Unsupervised methods are not suitable for making forecasts. The procedure of applying data analytic methods to particular areas involves data models such as neural networks, classification, clustering methods, and applying efficient algorithms that fit with the data characteristics. These models are used

mainly for predictive purposes. Predictive models without theoretical model backup are not appropriate. Such predictive models of supervised learning give also space for competition of the development of the fastest algorithms for data analysis. New techniques and algorithms can be developed but without theoretical deterministic explanations these new methods will not be satisfactory for developing good scientific explanatory and predictive models.

In this article, thus we focused on statistical analysis of the collected data sets and look at the similarities and differences of data mining and statistical methods for IoT systems. By explaining the need for both descriptive and inferential statistical methods, we showed the inadequacy of using only unsupervised data mining techniques for IoT data sets. We then concentrated on statistical inference approaches, such as the design-based and model-based inferences, and discussed the role of explanatory versus forecasting statistical models in an IoT environment. Lastly, we demonstrated the inadequacy of making forecasts using only unsupervised techniques in data mining and underlined the need for utilizing theoretical deterministic models together with the stochastic character of observed data sets in statistical models.

References

1. M.S. Mahdavinejad, M. Rezvan, M. Barekatain, P. Adibi, P. Barnaghi, A.P. Sheth, Machine learning for internet of things data analysis: a survey, Digital Commun Netw. (2017). https://doi.org/10.1016/j.dcan.2017.10.002
2. H. Simo, *Big data: opportunities and privacy challenges* (Fraunhofer-Institut für Sichere Informationstechnologie, Darmstadt, 2015)
3. M.N. Parc, Why big data needs a unified theory of everything (2016). https://venturebeat.com/2016/04/09/why-big-data-needs-a-unified-theory-of-everything/
4. R.A. Fisher, On the mathematical foundations of theoretical statistics. Phil. Trans. R. Soc. Lond. A **222**(1922), 309–368 (1922)
5. K. Doddapaneni, A. Tasiran, E. Ever, P. Shah, F.A. Omondi, L. Mostarda, O. Gemikonakli, Does the assumption of exponential arrival distribution in wireless sensor networks hold? Int J Sens Netw **26**(2), 81–100 (2018)

Chapter 9
Internet of Vehicle (IoV) Applications in Expediting the Implementation of Smart Highway of Autonomous Vehicle: A Survey

Umar Zakir Abdul Hamid, Hairi Zamzuri, and Dilip Kumar Limbu

9.1 Introduction

Chronologically, the usage of the term Industry 4.0 is reported to be first mentioned in 2011, as a new economic philosophy based on high-tech innovation foundations [1]. Drath and Horch in their works have defined the Industry 4.0 as a common impression of cyber-physical systems (CPSs) [2]. The concept subsequently allows the virtual representation of the actual physical world, where everything will be connected through the means of the Internet. This in exchange provides the door for the breakthrough of new ideas and innovations. Several major recent inventions are introduced in the last few years relative to these developments. These new emerging technologies have been identified as having the disruptive effects towards the current technology and market. Among the inventions which are identified are Bitcoin, Ride Sharing and Driverless Vehicle. Each of the aforementioned innovations is expected to, respectively, change the future of financial sectors, car ownership as well as revolutionizing the transportation industry [3]. The amalgamation of these disruptive innovations subsequently leads to the foundation of Smart City.

U. Z. A. Hamid (✉) · D. K. Limbu
Moovita Pte Ltd, Singapore, Singapore
e-mail: umar@moovita.com; diliplimbu@moovita.com

H. Zamzuri
Vehicle System Engineering iKohza, Malaysia-Japan International Institute of Technology, Universiti Teknologi Malaysia, Kuala Lumpur, Malaysia
e-mail: hairi.kl@utm.my

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_9

9.1.1 Smart City

Smart City is generally defined as a city, where the application of artificial and human intelligence is broadly integrated and assimilated into the municipality. The development of a Smart City involves the wide connectivity between its infrastructure and facilities, where everything is connected to each of its inhabitants through the participation of any devices. This is possible by having a grid of wireless connectivity to provide the massive scale information [4]. The announcement of over 80$ million investments for developing smart cities in the year of 2015 by the USA's government shows that the implementation of smart cities globally is inevitable and will happen in the near future [5]. There are a lot of debates on the terminology definition of the '*Smart City*'. In a lengthy study by Cocchia [6], several definitions of the 'Smart City' are mentioned, where each of the definitions bears different denotation. For example, 'Wired City' refers to a city connected by the means of wire, while the 'Virtual City' refers to the digital conceptualization of cities. Cocchia concluded that most of the studies regarding the concept of 'Smart City' continuously bear the similarity with what is defined with the '*digital city*'. Subsequently, Cocchia is suggesting that Smart City is defined as a comprehensive, web-based depiction of major traits of a physical city, and it revolves around the various fields of the human lives. This is further supported by Jin et al. [7]. In their work, Smart City is classified as a city which utilizes the data and knowledge from the information and communication technology (ICT), thus yielding more coherent, self-aware and well-connected city services and infrastructure [7]. From this brief introduction, it can be safely defined that ICT and IoT play a major role towards the development of Smart City. Smart City is expected to revolutionize various aspects of the way people live globally. Figure 9.1 depicts the sectors which are expected to be affected by the Smart City idea. Among the sectors are communication, energy, municipality as well as transportation [8].

9.1.2 Driverless Vehicle

One of the main features of Smart City is the autonomous vehicles (AV). The driverless means of the transportation system is also one of the main characteristics of the Fourth Industrial Revolution (4IR) and it is expected to be on the road by 2025 [9]. Its ability to reduce road fatalities and improving the driving experiences encouraged most of the major carmakers and startups globally to create their own AV technologies [9]. There are several standards given as the indicator towards the automation of the vehicle. The prominent one of these standards is the Society of Automotive Engineer (SAE), which lists Level 5 as the fully automated vehicle [9]. According to the Fortune News Portal, the newly created industry will be of around 7 Trillion $ market value by the year 2050 [10]. In addition, the rapid progress of research activity in this sector stimulates the birth of several major online

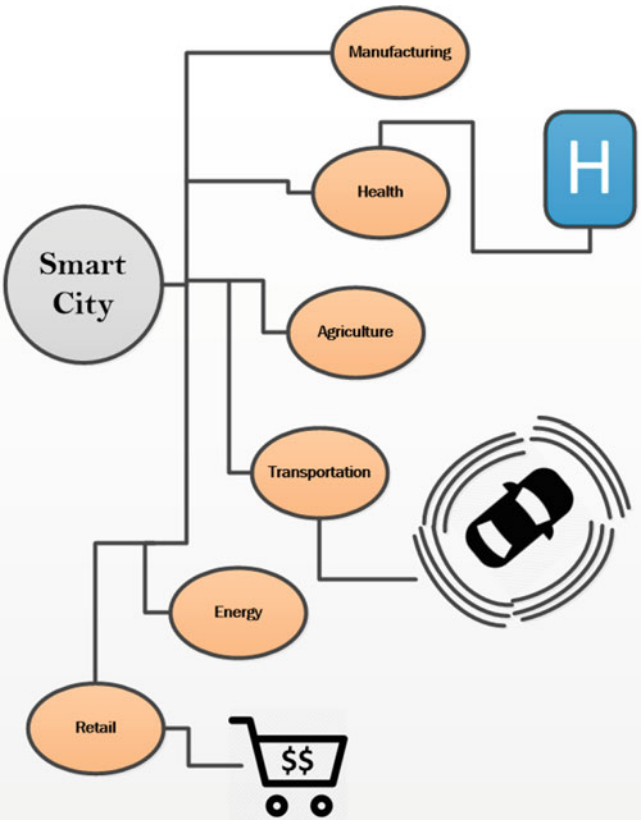

Fig. 9.1 Smart City implementation involves the varied aspects of human lives [8]

courses specifically created for the automated driving field. For example, Udacity, which is known to be a niche specified nanodegree provider has launched a Self-Driving Car nanodegree program [11]. The program has since attracted a lot of major companies' involvement such as Mercedes-Benz, NVIDIA, DiDi and UBER ATG, among many others [11]. This shows the vast importance of this emerging innovation. Despite the wide coverage of the topic in the media, until recently, most of the works done on the AV is still in the preliminary stages. Furthermore, most of the work done on the AV merely involves its development as a standalone platform. This means that the existing work on the connectivity between vehicles is still not in desirable levels. Figure 9.2 depicts the difference between standalone AV and connected AV. While standalone modular AV, as shown in Fig. 9.2a, might be working in a closed environment like tourism centre as the autonomous shuttle, it is not reliable in the more complex environment such as Smart Highway. Connected AV (Fig. 9.2b) on the other hand shares the information with each other about their current environment [12]. This in return helps to enhance the perception information

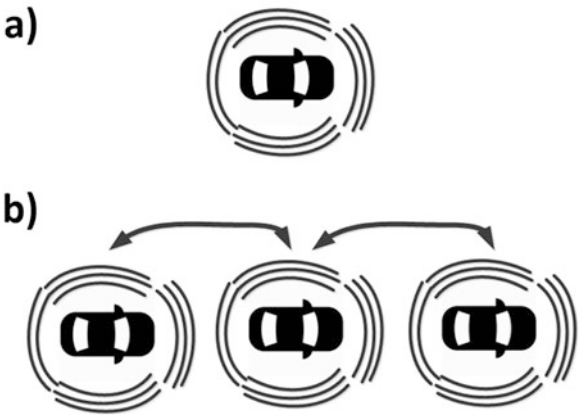

Fig. 9.2 The depiction of standalone AV which relies solely on the perception information from the installed sensors (**a**), while (**b**) depicts the advantage of incorporation of connectivity between the host vehicle and the environments for their perception and navigations [12]

of the vehicle, where the received data act like a web-based sensor. This provides the connected AV some advantages in the context of Smart Highway.

Despite the limited works done in the connected AV due to the large budget required, there are major companies which already commenced the works in the connected AV field. For examples, Jaguar Land Rover and Volvo are currently extensively scouting new talents in the connected vehicle field [13, 14]. This progress shows that it is crucial to have the necessary technological advancements for the implementation of AV available in near future. AV has the potential to change many aspects of the transportation sector. For instance, the ride and car sharing will be seen as a normal thing. With the reduced number of car ownership, car park area will also be reduced. Thus, this will subsequently provide much land space for other industry area development [9]. Ultimately, this will stimulate the birth of Smart Highway, comprises of the fully automated vehicle from the different type of transportation, such as the family car, freight trucks as well as other means of vehicles [15]. Since the first time the AV's concept has been widely discussed in the New York World's Fair in 1939, experts have been identifying the connectivity between vehicle as an important feature which can realize the dream of a reliable 'Smart Highway' [16]. However, as previously mentioned, since most of AV works are still focusing on the AV platform itself as a decentralized unit, the discussions on the vehicle connectivity is still limited. Thus, it is important to address the issue and provide an overview to the general audience to expedite the development of the Smart Highway.

Besides the connectivity issues, to implement a fully reliable Smart Highway of the autonomous vehicles, the current issues with AV should be identified too. In a normal highway navigation, a fully autonomous vehicle should be able to detect its environment, monitor its whereabouts as well as replan its path in the occurrence of hazards. In addition, the actuators of the vehicle should be able to accommodate the current vehicle dynamics to allow for the vehicle to yield the reliable navigation. This demands a good combination of several modules, including localization, motion planning, motion guidance as well as perception. However, for

the Smart Highway, the standalone development of AV is not sufficient. Instead, the connectivity between the vehicles and the environment should be incorporated into the design. To ensure that the AV system can be fully implemented and adopted by the general masses, the connectivity between the vehicle is important to ensure prevention of unwanted incidents during the driverless navigation. This demands a wide study on the vehicular connectivity.

Several topics have been explored as a solution to this issue. One of them is the Internet of Vehicles (IoV) [17]. As one of the branches of the Internet of Things (IoT), it allows the vehicles to be connected to each other by the means of the Internet. By integrating IoV into the development of AV, most of the issues that are faced in the context of vehicle connectivity are expected to be resolved. IoV allows the vehicles to share data with each other in the form of sensory data, risk data, environmental perception data as well as localization data, among many others. In addition, KPMG in their extensive report entitled 'Autonomous Vehicles Readiness Index' has listed the Top 20 countries which have shown the potential of realizing the Smart Highway in the near future [18]. One of the main traits between the listed countries is their Internet connectivity. In relation to the user acceptance, in [19], one of the main reasons which leads to the neglection of consumer towards the fully autonomous vehicles is identified as the fear of malfunction. In [20], it is mentioned that the combination of connected and autonomous vehicles concept will yield safer and more reliable vehicles. Thus, it can be concluded that the issue of connectivity with IoV is important to be addressed in the AV sectors.

9.1.3 Outline and Contributions of the Paper

Internet and Connectivity is profoundly important in the AV sectors, where IoV is identified as the vital solution to this issue. For a clearer understanding towards the topic, this subsection discusses the origin of IoV, which is the IoT. One of the main catalysers of the Smart City is identified as the IoT [21]. IoT allows the creation of a network which connects each object with Internet connections and acts as a web-based sensor, which provides the information from the environment to the objects. With this obtained data, IoT acts like a web-based 'actuators' which aids to yield the desired command for the objects, besides allowing other features such as data sharing, analysis as well as other related applications [22]. IoT as one of the major branches of Fourth Industrial Revolution has a vast coverage of function. The rapid progress in 4IR demands everything to be done quickly. Thus, the fundamental of IoT is to let everything to be connected to anything quickly. Figure 9.3 illustrates the idea of IoT. The significant effect of IoT is illustrated in the figure where important relations between the infrastructure and economical transactions as well as daily human activities are considered. As the autonomous vehicle is a disruptive innovation itself, one of the IoT branch, Internet-of-Vehicle was born. It is expected to lift the current hindrances towards the real implementation and connectivity issues of Smart Highways.

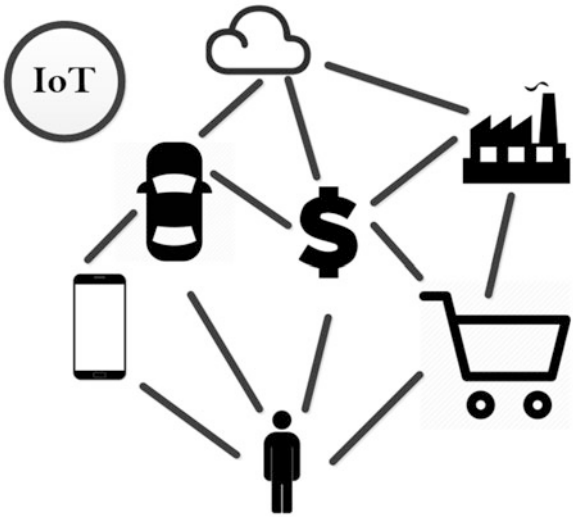

Fig. 9.3 Depiction of Internet of Things (IoT) working concept, where it allows for the connection of everything to anything using the means of Internet [8]

Based on the previous discussions, it is identified that IoV has the potential to resolve the issues with AV development in relation to the Smart Highway of a Smart City. However, as IoV is a relatively new field, not many discussions have been written to introduce this particular idea towards the general audience. Thus, the main contribution of this paper is to review the IoV and its relation towards aiding the AV implementation. The discussions will involve the brief background of IoV, current implementation of IoV as well as the issues with IoV. In addition, the major contribution will be the review of the relation between IoV and AV, and how it can expedite the Smart Highway implementation. The chapter is organized as follows. Section 9.2 describes the background of IoV. The application of IoV in expediting the implementation of Smart Highway is discussed in Sect. 9.3, where the relation towards IoV to the AV function modules is also discussed. However, as this work is intended for general audiences, the in-depth technical details are omitted. Finally, the deduced conclusions and future work suggestions are concisely written in the final sections. This work is hoped to give a broad idea about several topics, particularly IoV, AV as well as their relations towards the whole idea of Smart City.

9.2 Internet of Vehicles

It is obvious that IoV is an important element which can enhance the feature of AV, thus improving its efficiency in developing Smart Highway. In this section, for better understanding of the general audience, we will initially discuss the background of IoV. This encompasses the current implementation of IoV and their relation to the AV.

9.2.1 *Internet of Vehicles Background*

Most of the existing driverless vehicle implementations utilize a decentralized philosophy where the host vehicle obtains the information regarding its environment from the perception module from an individual set of sensors, installed on their physical body [23]. This is because most of the current implementation of the AV are still done in a controlled environment, instead of the public road. Figure 9.4 shows a prototype of a driverless vehicle by Moovita Pte Ltd during their first publicly held fully automated vehicle demo in Malaysia [24]. As can be seen, the vehicle is installed with several Radio Detection and Ranging (Radar) and Light Detection and Ranging (Lidar) as its perception sensors. For the frontal obstacle detection, the integrated sensor architecture allows for a reduced blind spot area relative to the environment.

As the growth of the developments in the AV sector foresees the technology to be implemented on full highway soon [25], it is important to incorporate the IoV into the architecture of the driverless vehicles. The integration of IoV with the platform will allow the data of different vehicles to be distributed and be shared into a network of AV during the fully autonomous highway navigation. The web-based sensor nature of IoV integration will improve the AV perception module performance [26]. Apart from the aforementioned advantages, vehicle-to-x (V2X), which is one of the main features of IoV, will enable the interaction between the

Fig. 9.4 Typical architecture of an autonomous vehicle, where the perception modules consist of physical sensors integration [24]

host vehicle and the environment, which includes the traffic light, road border and pedestrian, among many others. This will aid the autonomous vehicle in preventing crashes, improving its localization aspects as well as its other modules, such as path planning and motion guidance. The use of IoV is also able to yield a better in-vehicle entertainment experience during the long journey. In addition, the incorporation of the IoV will help in reducing the traffic jam by allowing the traffic information to be obtained by the vehicle instantaneously, prompting a new path replanning action by the AV.

IoV was initially created when many vehicles are started to be connected to each other using the IoT. Subsequently, the traditional method of communication of the vehicle, i.e. Vehicle Adhoc Networks (VANETs), is replaced as IoV. Yang et al. [27] explained that IoV consists of two main parts, (i) vehicles' networking and (ii) vehicles' intelligentialize. The summation of the two modules allows for the vehicles interconnection, vehicle telematics (connected vehicles), mobile Internet as well as the vehicle intelligence feature of the AV, such as deep learning and swarm intelligence among many others [27]. Currently, limited implementation of IoV has been done, with major aspects of the implementations focusing on the enhancement of the user experience, i.e. in-vehicle entertainment, as well as improvement of the active safety feature of the vehicle, where most of the focus is given in the collision avoidance system [27]. However, for the autonomous vehicles, no major implementation and production of IoV have been reported to the best of author's knowledge. Thus, in the next sections, the author describes the currently reported issues of AV implementation and how IoV can improve the issue with its incorporation into the vehicle network architecture.

9.3 Internet of Vehicles in Expediting the Autonomous Vehicle Smart Highway

In this section, based on the currently available literature, in total ten features are listed and suggested which are identified as the potential aspects which IoV can contribute in aiding and expediting in the realization of Smart Highway of AV. In each subsection, the current issue with the AV modules is listed followed by the potential solutions provided by IoV. In addition, several future works are hinted.

9.3.1 Improving Vehicle Connectivity

Yang et al. in their work [27] have stated that one of the potential benefits of IoV is that it improves the intra-vehicle network, which is a communication infrastructure between the host autonomous vehicle, human (either inside or outside of the vehicle) as well as object (surrounding or inside the vehicle). This improved connectivity

between the vehicle and the environment brings a lot of advantage. For example, since the autonomous vehicle can utilize the concept of ride sharing, where no car ownership is required in the future [3], IoV can provide better connectivity between the potential passenger of a driverless car with the host vehicle. This in return will aid in guiding the host vehicle towards certain specified pickup locations or point of destination for certain passengers. Though current application of ride sharing (e.g. Uber and Grab) provides the interaction between the human driver and the passenger, there is still no marketed application which allows the direct connectivity between the vehicle itself with the human outside the vehicle. This is one of the examples of how IoV will improve the connectivity of the driverless vehicle.

9.3.2 Enhanced Perception Modules

In a conventional design of a perception module for the self-driving vehicle, the typical flowchart is as depicted in Fig. 9.5. In a standard navigation of an AV, the integrated sensor architecture, typically consists of radar, camera and lidar monitors, which are used to detect and track the environment [28]. The information obtained by the sensors are then fused and utilized by the decision-making for subsequent motion planning actions. In the occurrence of hazardous scenarios, the motion control tracks the newly replanned emergency path output by the motion planning strategy, and this leads to a new feasible AV navigation. The schematic is illustrated and summarized by the author of this work based on the work of Hamid et al. [29] and Levinson et al. [30]. However, as this type of perception module is decentralized in nature, it possesses drawbacks. Particularly, in the event of unwanted natural catastrophic incidents, such as the earthquake, tsunami and whirlwind, as to the best of author's knowledge, no current vehicle sensors have the ability to measure the mentioned events. This can be solved by incorporating IoV into the perception module. The suggested new IoV-fused perception design is shown in Fig. 9.6. During a catastrophic event, for example, in the occurrence of an earthquake 10 km ahead of the current AV position, the information obtained using IoV, which is then sent to the AV acts as a web-based sensor. This information is then fused with other sensors to prevent the vehicle from continuing the navigation on the same path, and subsequently allows the new path replanning to a different route. The advantage of the assimilation of IoV into the system, as can be seen, is that it ensures reduced risk of AV navigation, thus increasing the comfort of the user experience during the journey. In addition, the web-based IoV sensor can compensate the disability of the current sensors to measure catastrophic natural events, for instance.

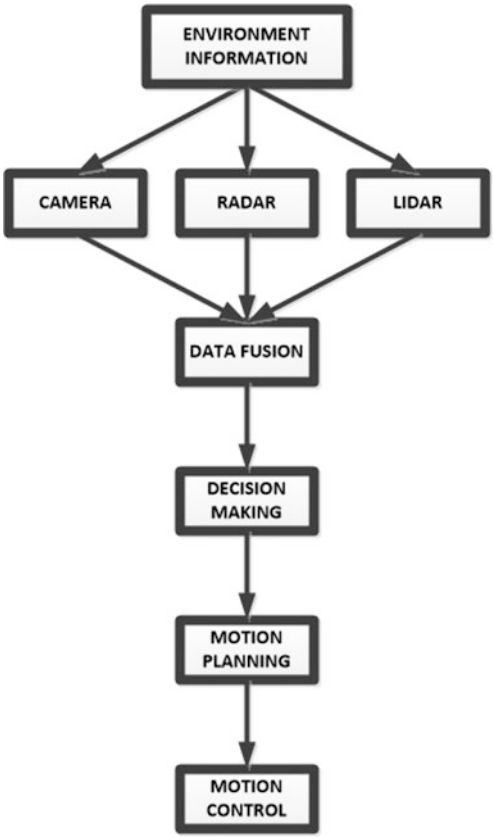

Fig. 9.5 Typical perception modules of an autonomous vehicle

9.3.3 Data Transferring Between Platforms

Autonomous Vehicle functions are based on the combination of several main modules, i.e. perception, localization, feature function as well as the mapping. These modules are connected to each other by the means of communication protocols, such as Transmission Control Protocol/Internet Protocol (TCP/IP), User Datagram Protocol (UDP) as well as Ethernet, among many others [31, 32]. For example, the motion guidance module receives the information of the vehicle's environment from the perception using the UDP module, while for the velocity tracking controller, the communication between the controller with the wheel speed sensors (velocity feedback) can be made using the serial bus. However, unstable communication is still an issue for works related to Self-Driving Vehicle. Unstable communication is crucial to be solved, as a failure in the communication between these modules in a hazardous scenario can cause an unfortunate event, including death. There are many companies currently working to provide the stable and improved connectivity between and inside the vehicles. One of these

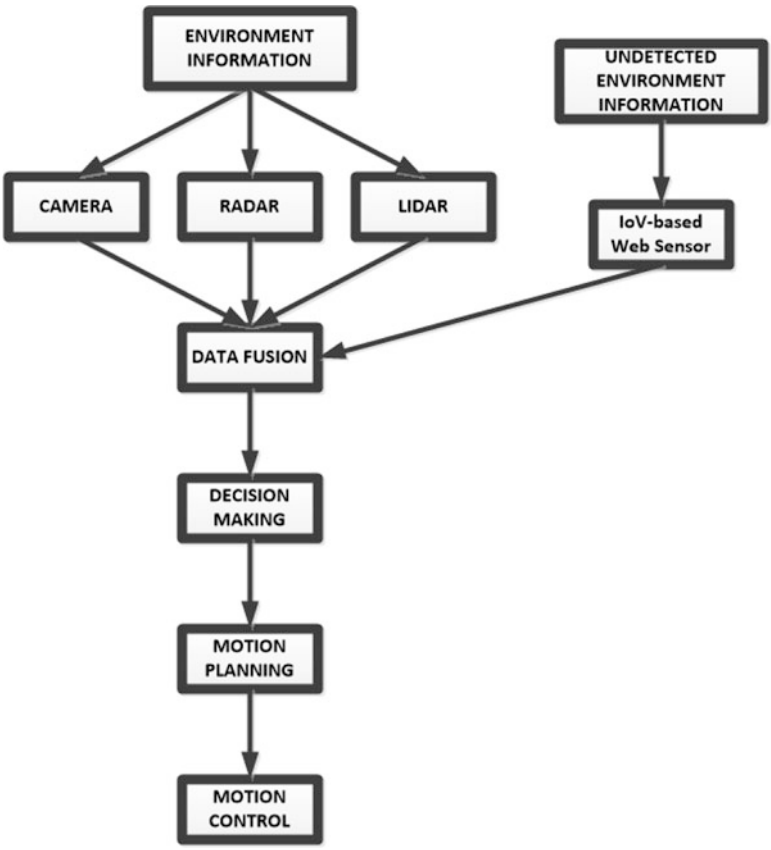

Fig. 9.6 Enhanced AV perception modules with integration of Internet of Vehicles (IoV)

companies is TTTech Computertechnik in Vienna [33]. This shows the importance of having a reliable network communication in an AV. By assimilating IoV into the network modules of the automated vehicle, this can be improved. For example, in the occurrence of a traffic congestion due to road accidents in an automated Smart Highway, the information obtained from the first vehicle in the row can be sent to the last vehicle in the row. This information, possibly consists of safe speed for navigation to prevent additional collisions, can be shared between the vehicle platoons, and directly sent via the means of IoV to the platoon members' actuators to guide the vehicle speed, thus allowing the vehicle to navigate safely.

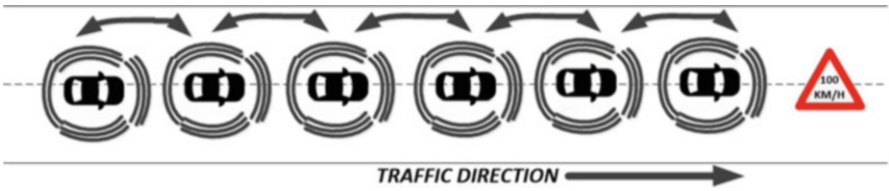

Fig. 9.7 IoV will allow for a dependable vehicle-to-x (V2X) implementation of autonomous vehicles

9.3.4 Vehicle-to-X

In Malaysia, the high number of casualties in road accidents is due to many factors. Among them are the appearance of unwanted objects in the middle of the road such as wild animal and unknown objects [34]. One of the current solutions which allows for the vehicle to know the existence of unknown object beforehand is a strategy called V2X [35]. X consists either vehicle, infrastructure or other environment entity. For a Smart Highway, by having this type of feature incorporated into the AV, particularly for a long distance journey, unwanted incidents can be prevented. As V2X demands a reliable Internet connection, IoV feature can help in improving the V2X strategy of Autonomous Vehicles in Smart Highway. Figure 9.7 reflects one of the examples, wherein the occurrence of unwanted objects, the first vehicle will automatically slow down to the allowed vehicle maximum speed. The sent information will then be received by the last vehicle. This will allow for a safer AV platooning in a highway. Without the incorporation of IoV, it is more difficult to realize the V2X idea.

9.3.5 Aiding Collision Avoidance Systems

According to [29], collisions with sudden appearing obstacle remains one of the main factors of casualties. With IoV, this number can be reduced. IoV will allow the vehicle to interact with the environment (i.e. traffic light and road border), pedestrian as well as other vehicle's sensory information through the means of V2X. This will aid the autonomous vehicle in preventing unwanted collisions. The fusion of V2X, as well as IoV, particularly will be helpful in preventing unwanted incidents such as the one illustrated in Fig. 9.8. From the figure, with IoV, the rear host vehicle, which navigates, receives the information obtained from the vehicle-to-vehicle strategy from the frontal vehicle about the sudden emerging obstacle during a curvilinear corner trajectory. This is very helpful particularly to aid the host vehicle in avoiding the collisions by reducing the time for the path replanning. With IoV, the newly replanned path can be fed directly to the host vehicle from the frontal vehicle. This in return will help to prevent the sudden emerging obstacle avoidance. IoV will help

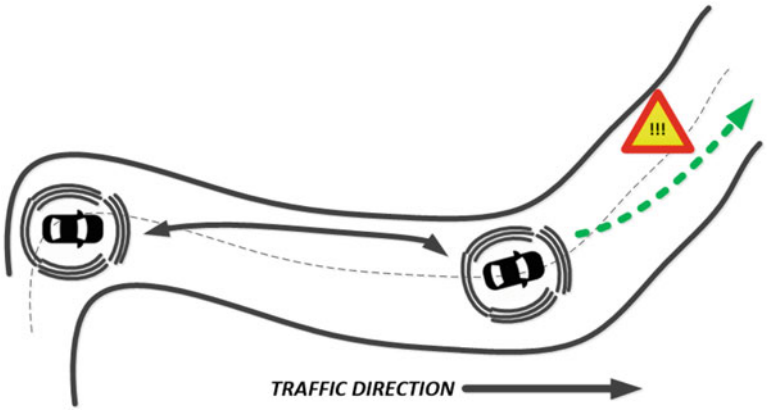

Fig. 9.8 Information about future road risks can be shared to other autonomous vehicles via the means of IoV, thus preventing potential collisions

to prevent collisions by providing information about the replanned path a priori to the rear vehicle in the platoons.

9.3.6 Improved Localization

Localization of AV refers to a module which provides its whereabouts knowledge to the vehicle, before passing the information for subsequent actions, such as path replanning. In a simulation study of AV, localization features are usually replaced by the kinematics mathematical model. In the real-time implementation of AV, even in the controlled environment testing, changes and drifts of the mapping strategy could happen due to the unwanted appearance of other vehicles/objects. With IoV, vehicle-to-infrastructure (V2I) interaction could provide information of the road environment to the vehicle quickly, thus allowing for better localization in the sudden appearance of object, e.g. a garbage truck. Figure 9.9 depicts a real-time human machine interface display of an AV [24]. With IoV, any changes in the environment of the future trajectory can be sent to the vehicle beforehand, thus allowing better localization for the vehicle, and subsequently reducing the potential of sudden intervention of human driver and emergency avoidance, due to unwanted risks.

Fig. 9.9 IoV can aid in improving the localization feature of a driverless vehicle [36]

9.3.7 Blind Spot

Blind spot region is defined as the area where the driver is not able to monitor the environment due to its location is out of reach from the driver's sights. Even with the help of the side and rear mirror, blind spot region limited the driver's view to the environment [37]. In the near future, the implementation of AV will not only involve the four-wheel vehicle, but also smaller vehicle, such as autonomous bike. In fact, several companies have launched the study about the two wheels autonomous vehicle [38]. As motorcycle has high potential to be in the area of the blind spot region, IoV will allow both the autonomous motorbike and car to connect to each other, thus preventing the potential crash. Figure 9.10a illustrates the idea. With IoV, collisions due to the Blind spot with regard to a passing by motorbike can be prevented. According to [39], near-miss incidents at intersections still occurred despite much prevention actions being taken. In addition, for the vehicle navigation in crowded urban area or highway exit intersection, the case of collisions with sudden appearing moving vehicle has been reported. With IoV, for the implementation of a fully automated Smart Highway, the case of intersection collisions with sudden appearing moving vehicle can be prevented (Fig. 9.10b).

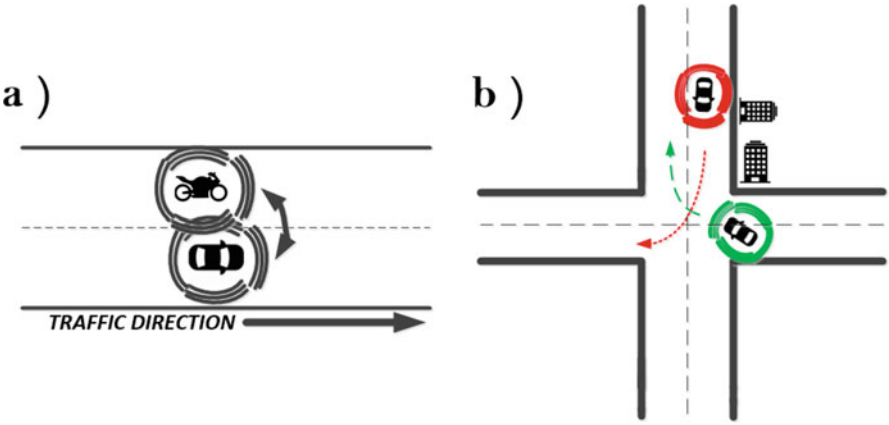

Fig. 9.10 Hazards relative to the blind spot area of an autonomous vehicle can be hindered with the integration of IoV, where the examples are shown in (**a**) where IoV can aid the autonomous vehicle to monitor the other vehicles in its blind spot area, and in (**b**), IoV can provide warning to the autonomous vehicle about the appearance of oncoming moving vehicle at intersections

9.3.8 Improving Path Planning and Motion Guidance During Nonlinear Vehicle Dynamics Scenario

According to [29], highly nonlinear dynamics during the emergency situations of a vehicle navigation demands a nonlinear controller to control the manoeuvre. However, for long automated driving trip, for example in North America, where several types of road surfaces can be passed by a vehicle during the navigation of multi-road surface, the information shared by the V2I regarding the road surface to the vehicle will help in improving the path planning and motion control actions, particularly during the slippery road surface. The incorporation of IoV into the AV design will enable this, thus improving the vehicle comfort and increasing users' acceptance towards the new innovation.

9.3.9 Better Infotainment

In addition to the race of development in order to build the first working driverless vehicle on the roads, major car makers are also competing for patenting AV technologies. Companies like Adient have also proposed a futuristic design of AV without the presence of steering and foot pedals, manipulating the x-by-wire technology. This subsequently yields for a futuristic AV design [40]. The design predicts that the human passenger will enjoy and have a rest during the autonomous vehicle riding, while simultaneously doing other activities on the commute. This in return demands the involvement of IoV. With IoV in the AV,

better in-house entertainment can be provided to the passenger, particularly for the longer journey. Better Internet connectivity will help to expedite not only AV highway but also allow for on-the-go financial transactions during the commuting session. This in return will also realize the smart city philosophy. Thus, IoV incorporation into the AV will help to yield an enjoyable Smart Highway AV navigation.

9.3.10 Reducing Traffic Jam

Automated Driving is expected to witness the implementation of ride sharing which will disrupt the idea of vehicle ownership. However, despite this, there are reports which say that the ride sharing has increased the traffic congestion in certain places [41]. With IoV incorporated into the AV design, in the occurrence of traffic jam, the last car in the vehicle platoon can utilize the information provided by the leader vehicle of the platoon to change their route with the replanned path. This in return will lead to a reduced traffic jam rate, by having a dynamic AV navigation throughout the smart city.

9.4 Assimilation of Internet of Vehicles in a Collision Avoidance System for Autonomous Vehicle: A Case Study

To support the claim of the IoV importance, a series of computational simulation is done, which involved the integration of IoV into a collision avoidance (CA) system. The proposed scenario depicts that the host vehicle navigates on a straight initial trajectory when it approaches a pothole. In the absence of IoV, the usual AV sensors are not able to sense the hole. The proposed scenario schematic is shown in Fig. 9.11.

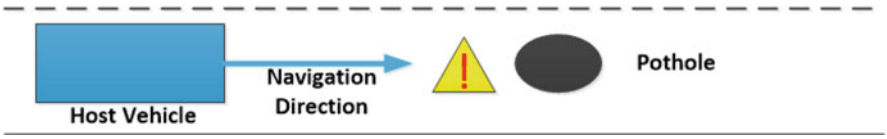

Fig. 9.11 Proposed simulation scenario, where a driverless host vehicle is approaching a pothole. The integration of IoV is expected to aid the path replanning to prevent the risk of pothole

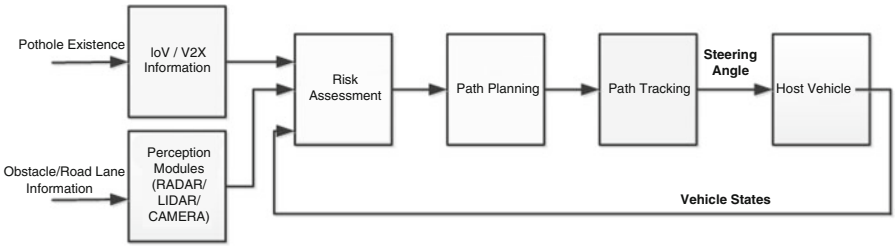

Fig. 9.12 Information from the IoV-based module aid the host vehicle to know about the pothole existence, thus allowing for a newly CA replanned path. The CA system is based on the work of Hamid et al. [42]

In this work, the function of IoV, i.e. vehicle to environment, is assimilated into the CA system (Fig. 9.12). With the assimilation, the vehicle will receive the information from the environment via V2X regarding the pothole existence. With the information, path planning submodule of the collision avoidance system will replan the path, thus allowing the host autonomous vehicle to avoid the hole. The aim is to show the importance of IoV in aiding more reliable and safer AV navigation. The proposed system works based on the algorithms and formulations based on the work of Hamid et al. [42]. For brevity, technical discussions of the algorithms are omitted.

9.4.1 Results and Discussions

The performance of the proposed improved CA system is simulated and compared with the nominal CA system with no IoV integration. The results are depicted in Figs. 9.13 and 9.14. As can be seen, without IoV (Fig. 9.13), the vehicle will keep navigating, and subsequently hitting the pothole due to no replanning action performed by the CA system. While with the IoV (Fig. 9.14), the vehicle successfully replans the path using the warning information, thus preventing uncomfortable AV navigation due to hitting the pothole. In both figures, the inclusion of IoV feedback into the system is symbolized with '1' or '0', where '0' indicates no feedback, and '1' indicates the integration of IoV feedback into the CA system.

With the results, it is proven that the assimilation will help in aiding the better AV navigation. As this chapter's objective is not on the algorithm, thus more details of the algorithms can be obtained in the original work [42]. This same concept can be used in more risky scenarios such as whirlwind, earthquake and heavy snow, among many others.

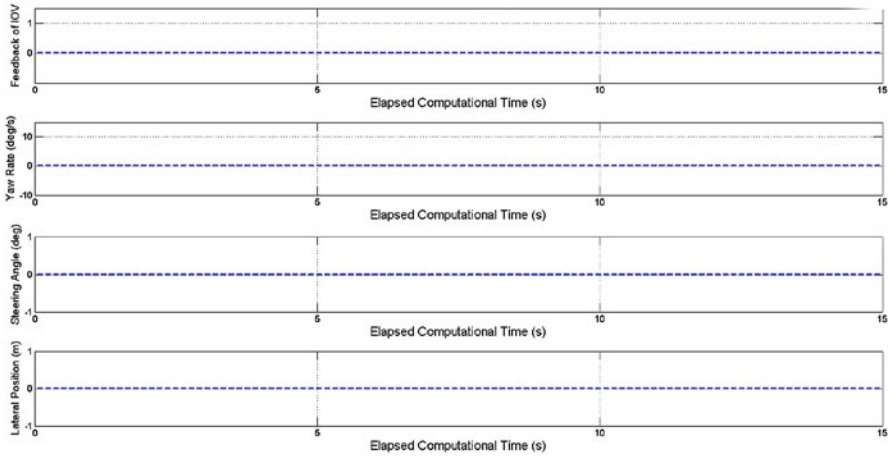

Fig. 9.13 Computational results of the proposed risk scenario. In this figure, the CA system is not assimilated with the IoV, thus no replanning actions are taken due to the failure of detecting the pothole

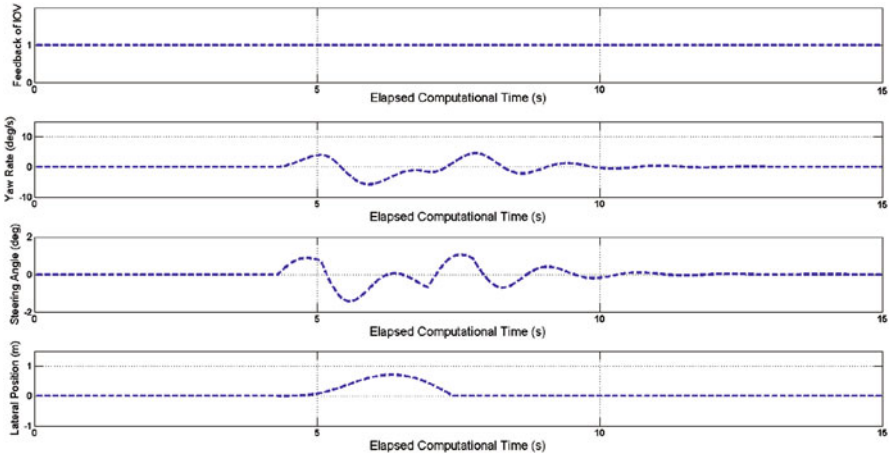

Fig. 9.14 Computational results of the proposed risk scenario. In this figure, the CA system is assimilated with the IoV, thus the actions are taken in relation to the IoV feedback to the CA system

9.5 Future Works

From the literature survey in this work, it is obvious that the assimilation of IoV will expedite the construction of real Smart Highway. However, there are a lot of works remain to be done. For example, the reliable and fast 5G Internet connection is required throughout the Smart City for the IoV of the Smart Highway feature

to be working properly. This demands the cooperation between the municipality office, government sector as well as industry representatives. In addition, as IoV is an Internet-based medium which can be easily manipulated by irresponsible sides, security concerns regarding the network and connectivity should be put into focus. In addition, privacy of the AV user during the Smart Highway navigation should be addressed too. The recent incident of the first fatalities involving a driverless car demands more detailed studies of the connectivity of autonomous vehicles with its environment [43].

9.6 Conclusions

This work is intended to provide introductory ideas on IoV applications in expediting the Smart Highway development, to the researchers who are new to this field. Based on the reviews in the previous sections, it can be deduced that a successful Smart Highway implementation demands the integration of IoV to improve the AV modules such as localization, motion guidance, perception as well as mapping. The authors are suggesting ten ways of which IoV can expedite the Smart Highway AV development. These include the improvement of the vehicle connectivity, enhancing the perception modules, aiding the V2X application, reducing the blind spot area as well as providing better entertainment systems.

Acknowledgements The authors would like to express their appreciation for the Vehicle System Engineering iKohza (VSE) research group in Universiti Teknologi Malaysia, Kuala Lumpur and Murtadha Bazli Tukimat (Wekanta Search Engine) for their interesting discussions on the topics considered.

References

1. V. Roblek, M. Mesko, A. Krapez, A complex view of industry 4.0. SAGE Open **6**(2), 1–11 (2016). https://doi.org/10.1177/2158244016653987
2. R. Drath, A. Horch, Industrie 4.0: Hit or hype? [Industry Forum]. IEEE Ind. Electron. Mag. **8**(2), 56–58 (2014)
3. A.A. Rahman, U.Z.A. Hamid, T.A. Chin, Emerging technologies with disruptive effects: a review. PERINTIS eJournal **7**(2), 111–128 (2017)
4. F. Al-Turjman, Mobile couriers' selection for the smart-grid in smart-cities' pervasive sensing. Futur. Gener. Comput. Syst. **82**, 327–341 (2017)
5. Fact Sheet: Announcing Over $80 million in New Federal Investment and a Doubling of Participating Communities in the White House Smart Cities Initiative. Available via The White House (President Barack Obama). https://obamawhitehouse.archives.gov/the-press-office/2016/09/26/fact-sheet-announcing-over-80-million-new-federal-investment-and. Cited 9 Apr 2018
6. A. Cocchia, Smart and digital city: a systematic literature review, in *Smart City* (Springer, Cham, 2014), pp. 13–43

7. J. Jin, J. Gubbi, S. Marusic, M. Palaniswami, An information framework for creating a smart city through internet of things. IEEE Internet Things J. **1**(2), 112–121 (2014)
8. L. Da Xu, W. He, S. Li, Internet of things in industries: a survey. IEEE Trans. Ind. Inf. **10**(4), 2233–2243 (2014)
9. U.Z.A. Hamid, K. Pushkin, H. Zamzuri, D. Gueraiche, M.A.A. Rahman, Current collision mitigation technologies for advanced driver assistance systems – a survey. PERINTIS eJournal **6**(2), 78–90 (2016)
10. Driverless Cars Will Be Part of a $7 Trillion Market by 2050. Available via Fortune.com. http://fortune.com/2017/06/03/autonomous-vehicles-market/. Cited 1 Feb 2018
11. Self-Driving Car Engineer Nanodegree by Udacity. Available via Udacity. https://www.udacity.com/course/self-driving-car-engineer-nanodegree--nd013. Cited 1 Feb 2018
12. T. Litman, *Autonomous Vehicle Implementation Predictions* (Victoria Transport Policy Institute, Victoria, 2017)
13. Land Rover to Start Real-World Tests of Innovative Connected and Autonomous Vehicle Technology. Available via Jaguar Land Rover. https://www.landrover.com/experiences/news/land-rover-to-start-real-world-tests.html. Cited 1 Feb 2018
14. Your life, your car, connected. Available via Volvo. https://www.volvocars.com/intl/about/our-stories/connected-car. Cited 1 Feb 2018
15. W. Brenner, A. Herrmann, An overview of technology, benefits and impact of automated and autonomous driving on the automotive industry, in *Digital Marketplaces Unleashed* (Springer, Berlin, 2018), pp. 427–442
16. A. Stocker, S. Shaheen, Shared automated vehicles: review of business models, in *International Transport Forum*, July 2017
17. O. Kaiwartya, A.H. Abdullah, Y. Cao, A. Altameem, M. Prasad, C.T. Lin, X. Liu, Internet of vehicles: motivation, layered architecture, network model, challenges, and future aspects. IEEE Access **4**, 5356–5373 (2016)
18. Autonomous Vehicles Readiness Index: Assessing countries openness and preparedness for autonomous vehicles. Available via KPMG. https://assets.kpmg.com/content/dam/kpmg/nl/pdf/2018/sector/automotive/autonomous-vehicles-readiness-index.pdf. Cited 1 Feb 2018
19. H. Abraham, B. Reimer, B. Seppelt, C. Fitzgerald, B. Mehler, J.F. Coughlin, Consumer interest in automation: change over one year. Transp. Res. Rec. J. Transp. Res. Board, No. 18–02666 (2018). https://trid.trb.org/view/1495407
20. C. Johnson, Readiness of the road network for connected and autonomous vehicles (2017). https://www.racfoundation.org/research/mobility/readiness-of-the-road-network-for-connected-and-autonomous-vehicles
21. L. Atzori, A. Iera, G. Morabito, The internet of things: a survey. Comput. Netw. **54**(15), 2787–2805 (2010)
22. Association Institutes Carnot White Paper, Smart Sensored Objects and Internet of Things, Greece, 2011
23. M. Gerla, E.K. Lee, G. Pau, U. Lee, Internet of vehicles: from intelligent grid to autonomous cars and vehicular clouds, in *2014 IEEE World Forum on Internet of Things (WF-IoT)*, March 2014 (IEEE, Piscataway, 2014), pp. 241–246
24. Moovita - Driving Autonomous Movers Solutions. http://moovita.com/. Cited 1 Feb 2018
25. C.W. Axelrod, Integrating in-vehicle, vehicle-to-vehicle, and intelligent roadway systems. Int. J. Des. Nat. Ecodyn. **13**(1), 23–38 (2018)
26. E.K. Lee, M. Gerla, G. Pau, U. Lee, J.H. Lim, Internet of vehicles: from intelligent grid to autonomous cars and vehicular fogs. Int. J. Distrib. Sens. Netw. **12**(9) (2016). https://doi.org/10.1177/1550147716665500
27. Y. Fangchun, W. Shangguang, L. Jinglin, L. Zhihan, S. Qibo, An overview of internet of vehicles. China Commun. **11**(10), 1–15 (2014)
28. J. Leonard, J. How, S. Teller, M. Berger, S. Campbell, G. Fiore, L. Fletcher et al., A perception-driven autonomous urban vehicle. J. Field Robot. **25**(10), 727–774 (2008)
29. U.Z.A. Hamid, H. Zamzuri, T. Yamada, M.A.A. Rahman, Y. Saito, P. Raksincharoensak, Modular design of artificial potential field and nonlinear model predictive control for a vehicle

collision avoidance system with move blocking strategy. Proc. Inst. Mech. Eng. Part D J. Automob. Eng. (2017). https://doi.org/10.1177/0954407017729057
30. J. Levinson, J. Askeland, J. Becker, J. Dolson, D. Held, S. Kammel, M. Sokolsky, Towards fully autonomous driving: systems and algorithms, in *Intelligent Vehicles Symposium (IV)*, June 2011 (IEEE, Piscataway, 2011), pp. 163–168
31. D. Shim, H. Chung, H.J. Kim, S. Sastry, Autonomous exploration in unknown urban environments for unmanned aerial vehicles, in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, Chicago, Aug 2005, p. 6478
32. M.L. Sichitiu, M. Kihl, Inter-vehicle communication systems: a survey. IEEE Commun. Surv. Tutorials **10**(2), 88–105 (2008). https://doi.org/10.1109/COMST.2008.4564481
33. TTTech: Robust Networked Safety Controls. https://www.tttech.com/. Cited 1 Feb 2018
34. B. Azhar, D. Lindenmayer, J. Wood, J. Fischer, A. Manning, C. McElhinny, M. Zakaria, Contribution of illegal hunting, culling of pest species, road accidents and feral dogs to biodiversity loss in established oil-palm landscapes. Wildl. Res. **40**(1), 1–9 (2012)
35. K. Abboud, H.A. Omar, W. Zhuang, Interworking of DSRC and cellular network technologies for V2X communications: a survey. IEEE Trans. Veh. Technol. **65**(12), 9457–9470 (2016)
36. MooVita - Fully-autonomous vehicle testing in Singapore. https://youtu.be/yDGHJhrAh1U. Cited 1 Feb 2018
37. F.R.A. Zakuan, H. Zamzuri, M.A.A. Rahman, W.J. Yahya, N. Hazima, F. Ismail, M.S. Zakariya, K.A. Zulkepli, M.Z. Azmi, Threat assessment algorithm for active blind spot assist system using short range radar sensor. ARPN J. Eng. Appl. Sci. **12**, 4270–4275 (2017)
38. Beating Valentino Rossi: The Race for Autonomous Motorcycles. https://www.2025ad.com/latest/2017-08/autonomous-motorcycles/. Cited 1 Feb 2018
39. U.Z.A. Hamid, H. Zamzuri, M.A.A. Rahman, Y. Saito, P. Raksincharoensak, Collision avoidance system using artificial potential field and nonlinear model predictive control: A case study of intersection collisions with sudden appearing moving vehicles, in *Dynamics of Vehicles on Roads and Tracks Volume 1: Proceedings of the 25th International Symposium on Dynamics of Vehicles on Roads and Tracks (IAVSD 2017)*, 14–18 Aug 2017, Rockhampton (CRC Press, Boca Raton, 2017), p. 367
40. Adient re-thinks interior technology for autonomous vehicles. http://articles.sae.org/15863/. Cited 1 Feb 2018
41. Uber and Lyft are creating a traffic problem for big cities. http://money.cnn.com/2017/10/11/technology/future/ride-hailing-cities-public-transit/index.html. Cited 1 Feb 2018
42. U.Z.A. Hamid, M.H.M. Ariff, H. Zamzuri, Y. Saito, M.A. Zakaria, M.A.A. Rahman, P. Raksincharoensak, Piecewise trajectory replanner for highway collision avoidance systems with safe-distance based threat assessment strategy and nonlinear model predictive control. J. Intell. Robot. Syst. 1–23 (2017)
43. How a Self-Driving Uber Killed a Pedestrian in Arizona. https://www.nytimes.com/interactive/2018/03/20/us/self-driving-uber-pedestrian-killed.html. Cited 9 Apr 2018

Chapter 10
Virtual Coordinate Systems and Coordinate-Based Operations for IoT

Gayatri A. Pendharkar and Anura P. Jayasumana

10.1 Introduction

Internet of Things (IoT) is expanding into diverse environments including manufacturing, environmental sensing (atmospheric, underground, and underwater) [1], smart cities, smart grids, and precision agriculture. Wireless sensor networks (WSNs) are the key building block in many of such IoT systems, where devices capable of sensing, actuating, communicating, and computing provide the interface to physical plants, environments, terrains, and phenomena. Such IoT networks have greatly benefitted and created novel approaches in different fields. Sensor nodes are deployed, for example, on farms to measure microclimates and soil conditions to improve yield, and for monitoring human presence in houses and offices to reduce the wastage of energy for heating and lighting. Decreasing costs, increasing capabilities, and advances in sensor networking technologies now make it possible to deploy large-scale networks of wireless sensor and actuator nodes that self-organize and adapt to carry out needed functionality robustly and adaptively.

Large-scale WSNs embedded in complex physical spaces depend on scalable and robust algorithms and protocols. Node localization and routing are among essential functions for many such network operations. Node localization refers to identifying the positions of different nodes in the network. In complex 2D and 3D networks, node location information by itself cannot facilitate routing. Furthermore, obtaining location information in the form of physical coordinates is costly and unreliable at best, or is simply infeasible. Coordinate systems also play a vital role in other IoT applications where a network is formed from heterogeneous devices, with each device acting as the fundamental node or unit. This chapter focuses on

G. A. Pendharkar (✉) · A. P. Jayasumana
Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO, USA
e-mail: Gayatri.Pendharkar@Colostate.edu; Anura.Jayasumana@Colostate.edu

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_10

virtual coordinates (VCs) that are more economical to obtain, less susceptible to parametric variations and interferences, and in many cases, provide equal or better routing performance compared to physical coordinates or geographical coordinates.

The outline of this chapter is as follows. In Sect. 10.2, we discuss the physical coordinates, complexities associated with obtaining them, and their limitations. Next, we introduce the concept of VCs, and how they serve as an alternative to physical coordinates. Section 10.3 describes a classification for VC schemes. In a network, several methodologies could be used as a foundation for coordinate assignment. Election of arbitrary or selective anchor nodes and structure embedding are examples of techniques that assist in VC selection. In Sect. 10.4, we present the attributes that distinguish the different virtual coordinate systems (VCSs). These attributes also provide a set of metrics to compare the different VCSs and analyze their strengths and deficiencies. Section 10.5 describes in detail the different VC techniques. It defines the techniques for coordinate assignment followed by a brief description of representative routing algorithms associated with it. We also provide comparison tables using the parameters from Sect. 10.4 to facilitate portrayal of weaknesses and assets of each technique. Section 10.6 summarizes and concludes the chapter.

10.2 Physical Coordinates vs. Virtual Coordinates

The common and familiar methods for dealing with points in a physical space are based on physical coordinates, e.g., (X, Y) in case of 2D and (X, Y, Z) in case of 3D. Thus, the use of physical or geographical coordinates (GC) for IoT networks deployed in 2D and 3D has been the obvious and default choice, and many of the WSN protocols rely on the availability of accurate GCs. The process of obtaining the GCs of nodes is termed physical localization [2], and the routing protocols based on these geographic coordinates are known as geographic routing (GR).

Two categories of routing protocols have emerged for large-scale networks of sensors, namely, address-based protocols and content-based protocols. The former relies on explicit node addresses, i.e., a set of coordinates defined using a specific algorithm. The latter defines the set of destinations with the use of certain attributes [3]. The content-based protocols rely on the use of approaches such as flooding and random walks to reach the destinations and hence have issues such as large overhead, limited scalability, and excessive uses of resources, e.g., excessive bandwidth and power consumption. The traditional Internet in comparison works on the principle of maintaining per node/subnet routing state, which grows as a function of the network size and number of destinations [3, 4, 5]. With constraints related to memory space per node with IoT subnets such as WSNs, approaches requiring such large amounts of data per node are infeasible.

Two fundamental limitations are faced by GCs in large-scale IoT subnets or WSNs. First is the difficulty and infeasibility of obtaining the physical coordinates. Due to confines in cost per unit and energy budget, it is unfeasible for individual

sensors to be global positioning system (GPS) enabled. GPS is not available indoors, and even outdoors its resolution may not be sufficient for dense networks. The alternative is to use analog measurements, such as RSSI or time-of-arrival (TOA) to estimate distances to other nodes, and thereby obtain node positions [6–8]. Many such techniques have been proposed in the literature [8, 9], but they are not accurate as they are susceptible to phenomena such as noise, fading, multipath, and interference, and errors in localization tend to accumulate [8]. Thus, such localization techniques have not been demonstrated in large-scale networks outside laboratory settings, and of course not in harsh and complex environments. The second limitation of geographic coordinates occurs even if one assumes the ability to obtain coordinates with sufficient accuracy, e.g., with manual calibration or using GPS. Specifically, GCs do not help achieve high routability in networks occupying complex physical shapes [10–16]. It is quite possible for two nodes to be physically very close to each other but separated by a long distance in the communication topology. Two nodes within proximity could be separated by obstacles or voids, e.g., a metal partition or concrete floor, creating a hole in the communication topology; in such cases, the packets will have to follow a long multi-hop path around the obstacle. Such scenarios are extremely common in many 3D IoT application environments, including those inside buildings, factories, and warehouses. Thus, the routing schemes must be able to overcome local minima created by concave geographical voids.

The existing geographic routing algorithms mostly focus on 2D networks [10, 17, 18]. However, these 2D algorithms are not effective nor scalable to 3D environments due to many causes. The geometric differences between 2D and 3D networks result in significantly increased computational complexity. Harsh and complex environments with 3D obstacles reflect or absorb radio signals rendering the GF and RSSI methods ineffective. Complex geographical topologies deployed on 3D surfaces or 3D volumes contain voids, causing the decoupling of GCs from the communication topology making them ineffective for many network operations [19]. A common element of most GR schemes is greedy forwarding (GF), in which a packet is forwarded to a node that is physically closer to the destination [11–15]. Some of the GR protocols are nearest forward progress and greedy forwarding [15, 16]. In the presence of voids or obstacles in the network, these protocols fail due to the inability to bypass complex-shaped voids in the network. Overcoming such local minima requires backtracking or some other scheme to navigate around the voids. For example, greedy perimeter stateless routing (GPSR) algorithm [10] attempts to navigate around the voids by following a certain direction. Such schemes become extremely costly or inefficient when the voids have complex shapes even in 2D deployments. With 3D deployments, such methods fail except in cases of very simple shapes of voids. A few proposed approaches for 3D geographic routing work albeit with some flaws. greedy random greedy (GRG) routing, a randomized geographic routing algorithm routes the packets based on a random walk algorithm, but only for network with unit ball graph (UBG) topology [20]. Greedy Hull Greedy (GHG) routing [2] constructs network hulls using planarization for routing; again, it applies to specific network types such as UBG, and GHG, and must deal

with complexities due to planarization computation. Furthermore, errors in node positions may lead to unrecoverable routing failures, which significantly degrades the performance of GR protocols [3]. A method to obtain the geographical addresses of an area without using geological information like GPS is addressed in [21]. This technique uses text processing, address preprocessing, and clustering to achieve accurate positions. This approach mostly provides an efficient but complex location discovery method for major e-commerce organizations.

To overcome the challenges associated with measurement and use of physical coordinates in IoT, coordinate frameworks have been developed that do not depend on the measurement of geographic location or distance information, yet can be used for functions such as routing and localization [22]. We call such coordinate systems VCSs (virtual coordinate systems). A VCS defines each node in the sensor network with a coordinate vector of some dimension that may be different from the dimension of the space the network is deployed in. Over the years, different types of VCSs have emerged that use different parameters of interest for VC election. A VCS depends on measures such as connectivity, packet loss, and topology. Some of the systems are significantly different from the Euclidean coordinate framework while others are Euclidean frameworks where node relationships and connections are preserved but not the actual physical distances. We use the generic term virtual coordinate routing (VCR) to denote routing schemes specifically developed for or based on a VCS.

One of the fundamental techniques for VCS is the graph embedding (GEM). The nodes are spread across the network with node connectivity information embedded inside. In this technique, a map or a sub-map of the topology with connectivity information is embedded which is later used by the routing algorithm to route the data efficiently around the network while capturing the voids and obstacles.

Many of the VC assignment techniques use a set of anchor nodes to build the coordinate framework. Anchors are nodes in the network selected randomly or through specific techniques, and the coordinate vector of each node, for example, may be the shortest hop distances to these anchors. The number of anchors becomes the VCS dimensionality of the network hence making it a parameter of interest. Routing is achieved using these VCs by greedy forwarding (GF) or some other technique. VCs of nodes are used for distance evaluation between nodes as well as for node identification (ID) [23]. If the VCS is based on anchors as the reference points, then the anchor selection could significantly affect the routing performance. Selection of an adequate number of anchors with apt placement helps acquire nonidentical VCs for each node.

VCs have also been used in the context of Internet and overlay networks—to obtain maps or coordinates that reflect or capture properties such as the delay or other network measurement parameters. With the advent of several VCSs, to preserve the network topology, most of the overlay networks need optimum selection of the neighboring nodes and communication paths depending on proximity, network delays, and round-trip time (RTT) [24]. Gathering this sort of information in real time could lead to a large amount of measurement traffic in the network. To achieve this, network coordinate systems (NCS) have been proposed. This

essentially couples network measurement parameters with the VCS. maximum likelihood topology maps [25] rely on the packet reception probability function to capture the graph topology. Topology preserving maps [26] too retain the graph topology yet are also homeomorphic to physical layout.

With increased interest on VCS for large-scale sensor and IoT networks, there have been several related developments related to VCS. Several concepts are in place to develop security means to address attacks on VCS. Several attacks that could potentially disrupt the VC formation are identified and techniques for alleviating them are proposed in [27]. A decentralized VCS capable of withstanding any sort of insider attacks is proposed in [28]. Technique to construct a robust coordinate assignment technique with less cost of communication that can sustain malevolent attacks is presented in [29]. These methods use spatial and temporal correlations for statistical analysis of real-time data sets. A game theory-based model to detect the best attacks and defense strategies are presented in [30]. A self-structuring algorithm that allows each node in a network to identify its position and all the nodes to collaboratively impose a geometric structure to the network is presented in [31]. The distributed algorithm runs on every entity or node without providing any prior knowledge of geographical location. With IoT expected to connect a massive number of nodes in the near future, there is a significant need to have sophisticated searching criterions; such approaches may also be VCS based as demonstrated in [32]. The method suggests the use of VCS in finding network statistics like delays, latencies, etc., using a decentralized approach. The real-time traveling path tracking algorithm for smart vehicles with encoders installed on the left and right side of the wheels to capture the rolling distance of the vehicles [33] relies on a VCS framework. The VCS in this case is fixed on the ground with factors such as the vehicle position and heading angle accounting for the experimental results of the path. The techniques give very accurate results despite obstacles, fog, rain, etc.

Several interesting schemes for routing and related operations using VCS have also appeared in recent literature. A method to use greedy routing on a virtual raw anchor coordinate (VRAC) system is considered in [34]. The VRAC coordinate computation involves measurement of roughly three raw node distances to be used as coordinates. Given that a saturated graph or network exists, greedy routing provides guaranteed packet delivery using VRAC system. Physical coordinate computation is a costly and complex technique for large-scale networks. Technique presented in [35] for deriving topology maps of the networks from the anchor-based VCs does not require a complete VC set. Using the theory of low-rank matrix completion, the topology maps are extracted for 2-D and 3-D networks using small subsets of VCs. A distributed protocol, viz., hexagonal virtual coordinate (HVC) to construct a VCS is presented in [36]. Using this HVC information, a source node can find an auxiliary routing path to the destination. This algorithm provides suitably placed landmarks and unique VCs throughout the network irrespective of any voids.

The above techniques are examples of VCS-based or related methods that make use of diverse parameters to devise coordinate schemes and thus network algorithms such as routing. VCS-based routing possesses certain advantages over traditional

routing techniques such as substantially high routing capability without relying on location information, consistent performance regardless of the presence of voids, and no localization errors. They may face problems such as identical coordinates and local minima due to lost directionality [23] which can be resolved by modifying the VC assignment algorithms. Identical coordinates arise if a sufficient number of anchors are not deployed, and local minima in these coordinate spaces appear as virtual voids in the network. In this chapter, we will review different VCSs, together with their properties and the corresponding routing techniques.

10.3 Classification of Virtual Coordinate Systems

As mentioned in the previous section, geographic routing (GR) uses geographical coordinates while VCR relies on some VCSs. Former depends on the physical distance while latter depends on some distance measure defined in the corresponding coordinate space. VCS approach relies exclusively on the relative distances (e.g., hop count) among nodes in the network. The general idea is to define a VCS and use it to induce a routing protocol based on the proposed VCs.

An anchor-based VCS overlays VCs on the nodes of a network based on their network distance from some fixed reference points (anchors or landmarks). The coordinates are computed during an initialization phase. From then on, the VCs serve in place of the geographic location for the purposes of network operations such as packet forwarding. As a VCS does not require precise location information or distance measurements, it is not sensitive to localization errors.

We classify the VCS into four categories as follows.

10.3.1 Virtual Coordinate Systems Embedding a Graph/Tree Topology

This technique as the name suggests embeds a graph in the network; the graph may be a tree (say) or a topology that is more complex. Based on that topology, the nodes in the network are spread relative to each other with connectivity information as a part of the embedding. Once the structure is established and the connectivity information is known, a routing algorithm may be developed to route across the network acquiring maximum efficiency and avoiding failures.

Examples: *Gradient landmark-based VCS (GL-VCS) [37], Medial Axis Protocol for VCS (MAP-VCS) [38], Graph Embedding for VCS (GEM-VCS) [39],* and *Hyperbolic Embedding in Dynamic Graphs for VCS (HEDG-VCS) [40].*

10.3.2 Virtual Coordinate Systems Based on Hop Distances to Anchors

The second approach is the most frequently used approach to establish a VCS. Each node here is first characterized using its hop distance to a specific set of nodes called anchors. This information may be used directly as a set of coordinates, or processed to extract a coordinate system with more desirable properties. Due to its sheer simplicity and effectiveness of the results, this category provides effective and flexible algorithms for VC assignment and routing.

Examples: *Anchor-Based Virtual Coordinate System [3], Axis-Based Virtual Coordinate Assignment Protocol (ABVCap-VCS) [41],* and *Directional Virtual Coordinate System (DVCS) [23].*

10.3.3 Topological Coordinate Systems

This approach talks about the techniques that help extract the topology of the network using node connectivity information and a few other parameters. It helps retrieve the informational graph that helps understand the network map without using any physical distance information.

Example: *Topology Preserving Maps (TPMs) [35].*

10.3.4 Virtual Coordinate Systems using Network Measurement Parameters

This technique listed avails of the network measurement parameters such as delays, RTT, or hop distances to configure a VCS that captures specific underlying network properties. Examples include coordinate systems motivated by the need to estimate delays without performing direct delay measurements [42], hence reducing the consumption of network resources considerably. It models the Internet as a geometric space, depicting the position of all the present nodes in the Internet by a coordinate in the space [42]. These techniques attempt to retain the physical topology of the network, the connectivity, shape, delay, or some other property.

Examples: *Vivaldi—A decentralized Network Coordinate System [42],* and *Maximum Likelihood Topology Maps for Wireless Sensor Networks Using an Automated Robot [25].*

In the following sections, we will discuss the examples from these three classes of VCs. Prior to that, we present the parameters that are useful for comparison of the different VCSs.

10.4 Attributes of Virtual Coordinate System

The purpose of a VCS, as stated earlier, is to serve one or more purposes related to networking a large set of nodes. A VCS, for example, often acts as a proxy for node locations for efficient routing or topology control. A good VCS based on a parameter such as delay or packet loss may allow estimation of such parameters with reduced network measurements, i.e., with minimal cost and effort. Different systems have their own algorithms for assignment of coordinates to the nodes in the network. Such attributes can also be used to compare different VCSs, and to select the proper VCS for a given criterion.

10.4.1 Use of Anchors

Anchors correspond to a set of nodes in the network that act as intermediaries to the other nodes in the network for calculating the VCs. Basically, the number of anchors determines the coordinate dimension for the nodes in the network. Intuitively, the higher the number of anchors, the higher the cost of generating coordinates and the more accurate the node position in the corresponding space. In fact, if all the nodes are anchors, the coordinate system corresponds to the (hop) distance matrix of the graph.

10.4.2 Efficiency of Routing/Measurements

The primary purpose of a VCS is to serve some network-related function(s) such as routing or delay estimation. Thus, the efficacy of the coordinate set to meet the stated purpose is of importance. The efficiency may be quantified by performing routing or the appropriate network measurement scheme. This parameter indicates the precision of the coordinates that are assigned to the nodes in the network.

10.4.3 Susceptibility to Local Minima Issue

Even with an ideal implementation of the algorithm of interest (e.g., routing) for which the VCS is targeted, there could be cases that cause the algorithm to fail. These anomalies impair the desired functionalities, i.e., sensing and communication. Examples of issues with VCS include the following: identical coordinate assignment, local minima, formation of logical voids or holes, etc., which would lead to geographically correlated problem areas such as coverage holes and routing voids.

10.4.4 Ability to Deal with Node Failures and Changing Topologies

In an IoT or a WSN, a node may fail at random points in space and time, or new nodes may be added to the network. As such events change the network topology and connectivity, they may cause disruptive changes to the coordinate system or the resulting algorithms. This attribute aims at capturing whether the VCS is susceptible to such changes, and if so the ease or difficulty with which coordinate system may be restored following such an event. A robust coordinate system will maintain performance within a narrow margin even when such events occur in the network.

10.4.5 Ability to Capture the Network Shape and Voids

The node deployment could be of any shape and contain voids of different shapes disrupting communication among the nodes. Ability to capture such topology properties is an important aspect of a coordinate system. Hence, the efficacy of the coordinate system to retain these topological assets is vital.

10.4.6 Applicability to 3-D Networks

Node placement of a WSN in the physical space determines the kind of network it is. A planar network with nodes along two axes is a 2-D network. If the surface on which the nodes are placed is not planar, we may call it a 3-D surface network. A 3-D volume network refers to a network deployed in a three-dimensional volume. We use the term 3-D networks to refer to 3-D surface networks, 3-D volume networks, as well as networks containing both 3-D surfaces and 3-D volumes. It is also important to note that some of the VC techniques have been extended to networks with no associated physical dimensionality such as social networks [35]. Ability of the protocol to implement the VCS in these sort of networks increases the number of application areas as well.

10.4.7 Distributed Computability of VCs

In a WSN, the computation can occur at every fundamental unit, i.e., at the individual nodes or it could occur at a centralized unit, i.e., a centralized server. These are two different types of methods for the WSNs. Consider, for example, a WSN consisting of many nodes sensing the environmental conditions. In the centralized approach, these nodes send the sensed data to a centralized server known

as the base station (BS). Due to energy constraints per node, centralized approach proves efficient in those terms. Here, all the nodes are grouped into clusters and, then, one representative node is assigned as the cluster head (CH). This node collects all the data within the cluster and sends the data to the BS. Now, only the CH nodes are required to perform long distance transmission hence saving the energy consumption for the other nodes [43]. On the other hand, in the distributed approach, the computation is autonomous, down to the single fundamental unit of the network. The distributed approach takes into consideration the battery restraints per node and the density of the WSN. Here, the computations occur based on communication among the neighboring nodes. Distributed approach is more desirable than the centralized one due to several reasons. In centralized computing, if the BS fails, the entire network may fail. Additionally, in case of individual node failures in the network, the recovery or repair must be done at the respective central node unlike with distributed algorithms, where the node recovery can be done at its own level. All the VCSs discussed in this paper are based on a distributed computing approach. The selection of these cluster heads or landmarks for centralized network is based on a landmark selection algorithm. The set of anchors can be predetermined [44–46] or randomly selected [48, 49]. Distributed VCSs are independent of explicitly designated infrastructure components, requiring any node in the system to act as a reference node. Examples of such systems include PIC [49], Vivaldi [42], and PCoord [43, 44].

10.4.8 Directionality

VCSs are an efficient way of characterizing the node locations thus replacing the geographical coordinate assignment approach. VCS offers a lot of attractive properties such as high routability, consistent performance in the presence of physical voids in the network, and efficient connection information embedded in the VCs. In some VCSs, when there is a mapping of the physical domain to a virtual domain, coordinates become insensitive to directional information. Many deficiencies associated with VCS are due to the missing directionality information and the lack of knowledge of the physical layout. Some of the VCS techniques discussed in this paper can capture or extract directionality information while others do not.

10.4.9 Applicability to Wireless Sensor Networks

Certain VCS are not useful for resource limited networks such as WSNs or IoT subnets. There are different performance metrics of interest such as routing performance, efficacy of latency estimation, and boundary detectability, which are important in different contexts. Although our focus is on IoT subnets and WSNs,

coordinate systems are also of interest for other applications, e.g., coordinate spaces that capture latency information among devices.

10.5 Virtual Coordinate Systems

10.5.1 Virtual Coordinate Systems Based on an Embedded Graph/Tree Topology

In this section, we will consider several VCSs which are based on discovery of the global topological structure of the network, e.g., by embedding of a graph structure such as a tree in the topology. Such a structure may then be used to characterize positions of the nodes and for routing.

10.5.1.1 Gradient Landmark-Based Virtual Coordinate System [37]

GLIDER is a novel technique which only uses the node connectivity information and a few selected landmark (anchor) nodes to achieve distinctive node coordinates. This approach is divided into two phases—the global planning phase and a local greedy routing phase. The global preprocessing step helps in mapping the topology of the network using the node connectivity to account for obstacles or holes in the sensor field. Following this phase, the network is partitioned into tiles with each tile containing a subset of network nodes. These tiles are expected to have a trivial topology and simple greedy forwarding methods using local VCs that help achieve good routability.

Consider a communication graph $G = (N, E)$, where N is the set of sensor nodes and E is the set of unweighted edges. The graph (hop) distance between two nodes is the shortest hop count (number of edges) between them. For a packet traversing from a source node to the destination node, the successor to the source node is always the node which reduces the hop count to the destination node. An auxiliary atlas $M(G)$ is constructed which is shared with every node. It helps achieve awareness about the global topology of the network and connectivity information by partitioning the nodes into tiles and mining the adjacency relations between these tiles. Each of the partitions or tiles is uniquely identified by a landmark or anchor node. It is critical to make an equitable selection of landmark nodes to achieve best results for this algorithm.

Landmark Voronoi Complex [37]

For a set of nodes N in a communication graph G, the landmark Voronoi complex (LVC) helps retain the global topology of the network using local connectivity

information. Each Voronoi cell is connected to its neighboring cells. Combinatorial Delaunay triangulation (CDT) is the method to retrieve the connectivity data between the cells to form a condensed atlas of the sensor network, which subsequently is used for global route planning. The graph is assumed to be connected and $t\,(u, v)$ denotes the shortest path in terms of hop count. For a graph $G = (N, E)$ and a subgroup of anchors $A \in N$, the Voronoi cell $T(n)$ of a node $n \in A$ is given by [37]:

$$T(n) = \{u \in N \mid \forall w \in A, \tau\,(u, n) \leq \tau\,(u, w)\} \tag{10.1}$$

Several properties are associated with an LVC:

- *Property* 1 [37]: This property of a Voronoi cell states that for any given node $u \in T(n)$, the shortest path from u to n be completely contained in $T(n)$.
- *Property* 2 [37]: This property states that if the graph G is connected, the combinatorial Delaunay graph $D(L)$ is also connected.

Selection of Landmark Coordinates [34]

Any set of landmarks work with the properties stated for CDT and LVC. However, the selection of landmarks still plays a crucial role in attaining higher routability. CDT provides every node an auxiliary atlas with details on global topology. This should be as compact as possible for easy replication across the network with minimal cost. There should be an adequate number of landmarks to have better coverage of the network. Once the landmarks are picked, all the other nodes are supplied with a coordinate system with unique VCs. Then, a greedy routing algorithm is used on these coordinates for routing. These coordinates eliminate the local minima issue as well. There are two approaches to derive these VCs—Continuous and Discrete.

Computation of Virtual Coordinate [37]

- Continuous version (VCs)—The objective is to form a coordinate system using the shortest Euclidean distances to the landmarks. This distance should be such that the gradient descent with each hop assures reaching the target successfully. This method guarantees finding the target without the occurrence of local minima.

  Property [37]: In a continuous Euclidean plane of nodes, the gradient descent of the function $s \rightarrow d(s, t)$ is defined where s is source node and t is the target. This gradient allows the packets to be routed to the target, only position where the gradient is zero. Thus, a packet will always converge to the target if there are at least three noncollinear landmark nodes.
- Discrete version (VCs)—The main difference between this technique and continuous version is that the distance is now measured in terms of hop counts. The

hop distance is a fair approximation of the Euclidean distance. The occurrence of local minima is a possibility.

The VCs derived using these techniques are called *centered virtual coordinates.* These coordinates are also called local landmark coordinates and are mainly used for inter-tile routing.

Naming and Routing in GLIDER [37]

Naming – Each node in the network is assigned an ID and name. The ID is unique to a node, e.g., may be hardcoded. The name is assigned after preprocessing and it need not be unique. After the calculation of the Voronoi complex, every node in network belongs to a Voronoi cell/tile. This is the *resident tile* of the node and the landmark of that tile is the *home landmark*. The node name includes ID of its *home landmark*. Additionally, every node has a list of the distances to its neighboring landmarks.

Routing—Consider a case with source node s and target t. To route a packet from source to the target, GLIDER goes through two stages of routing—global and local.

- Global Routing—It accounts to finding the shortest path between source and target. This is also called as intra-tile routing. A primary step would be a look-up at the pre-computed shortest-path tree at source $h(s)$.This is a path that progresses through c tiles; say, $T_1, T_2, T_3, \ldots, T_c$ where $T_i = T(s_i)$ for all the landmarks s_i, with $s_1 = h(s)$, $s_c = h(t)$.
- Local Routing—This method exploits the landmark coordinates to achieve inter-tile routing. It is responsible to discover path from tile T_i to T_{i+1}. This is achieved by using the gradient descent on the VCs. Once the packet reaches a node $n \in T(h(d))$, it is transmitted to the neighbor closest to the landmark node d in terms of Euclidean distances (local landmark coordinates). In case of local minima, flooding is initiated within the tile.

10.5.1.2 Medial Axis Protocol for Virtual Coordinate System [38]

Medial axis protocol (MAP) is a coordinate assignment and routing protocol that runs without any geographic information and performs routing and load balancing efficiently. MAP constructs the medial axis of the network field by selecting the set of nodes, each of which has at least two closest nodes on the boundary. The routing algorithm uses these coordinates to locally route the packets in the network. Many of the techniques discussed rely on the optimum selection of landmark nodes that are used to compute the local landmark coordinates of the nodes in the system. However, landmark selection is a complex problem which does not have a conventional method. MAP ensures the retention of the geometrical and topological features of the network working as the backbone scheme. It is a protocol serving many applications like robot path planning [50] and surface reconstruction

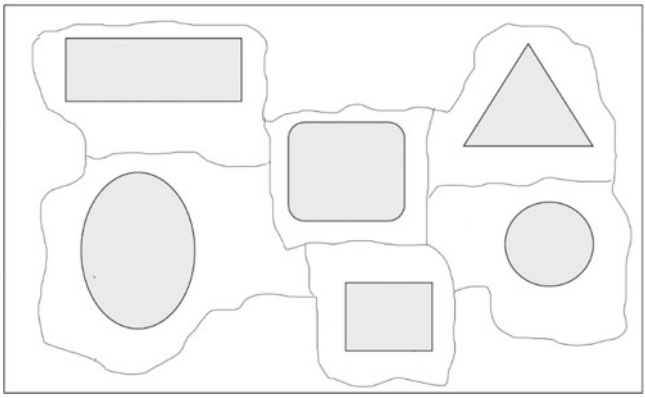

Fig. 10.1 Examples of medial axis formation in medial axis protocol (MAP) in a random network with obstacles

[51, 52]. Medial axis for MAP uses only the connectivity information. This can be represented by a graph whose size is directly proportional to the size of the sensor network.

MAP works without using any location information and uses only the graph connectivity information. The protocol is extremely light weight and compact. The medial axis is represented by a graph whose size is directly proportional to the size, geometry, and complexity of the sensor network. It requires no maintenance whatsoever once constructed. Since the medial axis serves as a skeleton, MAP has a good coverage throughout the network. It is also robust to variations in the network model following the steps of naming and routing in both discrete and continuous domains.

Medial Axis Protocol Naming and Routing Schemes [38]

- Euclidean (Continuous) Domain Naming Scheme—This step involves construction of a medial axis [50] in the Euclidean plane (see Fig. 10.1). A medial axis of the curve C is a set of points in the plane having two or more closest points in C. Consider S as a bounded open set. The boundary of the curve S is denoted by ∂S. The medial axis of the boundary curve is stated as M. This medial axis is constructed as a combination of many continuous curves [54]. M has two parts divided into the exterior and interior of S. The interior part is of interest for further computations. From every point on medial axis, a maximal disk can be drawn inside S with two or more tangent points on the boundary curve ∂S. The line connecting these tangent points and the point on medial axis is called as the chord of that point on M. Medial radius of the point is denoted by $r(p)$ where p is the point of interest. Additionally, a medial vertex is defined by a point of the

medial axis with at least three closest points in ∂S and a segment on the medial axis joining two medial vertices is a medial edge.

Every point in the set is assigned a name. Given two points a and b, the Euclidean distance between them is denoted by $|ab|$. Every point has a 3-dimensional name given by $N(a) = x(a), y(a), h(a)$. Here, a lies on the chord given by $x(a)y(a)$. $x(a) \in M$, $y(a) \in \partial S$ and $d(a)$ denotes the normalized distance between a and $x(a)$. $d(a) = \frac{|ax(a)|}{r(x(a))}$. This naming scheme assigns unique coordinates to all the nodes. There are a few lemmas and theorem stated below:

Lemma 1 [38] *For any point a in the given set which does not lie on the medial axis M, if the point lies on the chord given by pq, where $p \in M$, $q \in \partial S$, then q is the only point on ∂S closest to a.*

Lemma 2 [38] *If point a does not lie on the medial axis M, there is a unique chord that passes through a.*

Theorem 1 [38] *Every point in the set S has a unique name.*

The medial axis and the chords divide the region of S into many cells called canonical cells. Every cell is bounded by two chords, segment of boundary curve ∂S and a medial edge. A medial edge is shared by two canonical cells. Also, a point on M with j chords is adjacent to j canonical cells. An $h - latitude$ curve is defined as a set of points in all the cells with height h, $0 \leq h \leq 1$. Also, an $x - longiude$ curve is a continuous line segment; a chord in the set of cells with a medial point x.

Lemma 3 [38] *No two chords inside a canonical cell have a common intersection.*

Theorem 2 [38] *For any canonical cell given by C, collection of the points with height h, $0 \leq h \leq$, is a continuous curve. Every cell is separated by the medial axis and all the chords of medial vertices.*

Euclidean (Continuous) Domain Routing Scheme—This algorithm exploits the naming scheme for nodes. Given a source s and a destination d and the median axis M of the set S, the names of the nodes will be given by, $N(s) = x(s), y(s), h(s)$ and $N(d) = x(d), y(d), h(d)$. Two points on the medial axis are $x(s)$ and $x(d)$. Shortest path between $x(s)$ and $x(d)$ is called as the reference path and the length of that path is known as the reference distance. A straightforward method to route would be going from s to $x(s)$ on the medial axis, following the shortest path between $x(s)$ and $x(d)$, and then routing from $x(d)$ to d again. However, this excessively uses medial axis as the backbone and causes it to be heavily loaded. Hence, routing is divided into two paths—Initially, routing is done in parallel to the shortest path between s and d until a point p is reached which has a medial point $x(d)$. Once $x(d)$ is reached, the chords are followed to reach the destination.

- *Discrete Domain Naming Scheme*—This scheme differs from the continuous scheme only in a way that the distance between the sensors is now measured in terms of hops. As hop counts approximate the actual distance, marginal errors

in routing could occur. The medial axis protocol follows three main steps—detecting boundaries of the sensor field, construction of the medial axis graph, and naming the sensors.

Every node in the network identifies itself as a boundary node. Depending on whether it is an outer boundary or boundary of an inner hole, the node distinguishes itself between the two sets. Applying the crust algorithm to the dense sensor networks helps construct these boundaries better. Simplest way is to request the closest samples of nodes to determine themselves by local flooding and include the nodes with the shortest paths between them as boundary nodes.

A medial node is identified as the node which has equal hop counts to two or more boundary nodes that are the closest. The medial nodes defined in discrete domain are noisy. A small bump on medial axis could result in a long branch in the graph. These medial nodes are identified when the boundary nodes initiate flooding. The medial axis is constructed using these medial nodes. Once the axis is constructed, entire network is flooded with the medial axis information and the medial axis graph is constructed. Every node receives and stores this graph locally.

Nodes are assigned names using the medial axis as reference. An x-range $[m(p), n(p)]$ and height $h(p)$, where p is the node assigned per node. The x-range defines the part of the medial axis in which p lies and the height tells the distance of p from the medial axis. Thus, every node stores—the medial axis graph, name (coordinates) for itself and 1-hop neighbors, medial axial neighbors, and flag to detect if node lies on the medial axis.

Discrete Domain Routing Scheme—This routing algorithm has the same implementation as the continuous case except for a few alterations. The protocol starts with the global planning step which finds the shortest path between the source and destination node in the medial axis graph. A path parallel to this shortest path is devised until a node whose medial point is same as the destination is reached. With the help of shortest-path trees from the medial node, packet is routed to the destination.

10.5.1.3 Graph Embedding for Virtual Coordinate System [39]

With increasing size of WSNs, scalability is one of the main factors to be taken care of. With a larger network, the number of measurements increases, causing a humungous amount of data across the network. Also, each sensor node has limited memory, storage, communication range, battery power, and computational ability. Hence, it is important to utilize resources per node efficiently. There are several techniques to retrieve the data sensed by nodes—local storage, external storage, and the data-centric approach. The data-centric approach is the most energy efficient approach which relies on proficient routing mechanism. The GEM is a routing technique for a sensor network with data-centric storage and information processing. GEM is devoid of any geographical information and gives decent results even in the presence of physical voids.

GEM is basically a setup that is a graph with labeled nodes, embedded in the original sensor network topology in a distributed and efficient manner. Many of the existing overlay protocols for routing are not suitable for sensor networks as each hop in the overlay could be several hops in actual network. Hence, it is crucial to have an underlying system for routing. GEM works in such a way that every node is aware of its neighboring nodes through the labels assigned to them. These nodes can perform routing by exploiting these labels. Additionally, the data names can be mapped to the labels to use data-centric storage. This approach is based on constructing a virtual polar coordinate space (VPCS) without any physical placement information of the topology. There are two techniques developed to embed this virtual apace with the network topology—the first one requires for the nodes to find out distances between them and their neighbors unlike the second one. Virtual polar coordinate routing (VPCR) is a routing algorithm that uses the VPCS.

There are three key steps in implementing the GEM, which are as follows.

Introducing the Graph Embedding Framework [39]

This step involves mapping of one interconnected network onto another. A simply serviceable network topology is chosen and is mapped onto the real network technology through the process of GEM. Here, we have a guest graph G which is chosen by the user mapped onto the host graph H using a function, say α. Here, each node from G is mapped one-on-one to nodes in H.

There is an additional edge-routing function ρ which assigns every edge $e = \{u, v\}$, where $e \in E(G)$ a path in the graph H connecting the nodes $\alpha(u)$ and $\alpha(v)$.

Construction of the Virtual Polar Coordinate Space [39]

This step works with the formation of a ring-tree graph which must be aligned with the real network topology. Every node in the tree is assigned two things:

- A level which is counted as the number of hops to the root node of the tree.
- Virtual angle range which identifies the node uniquely from all other nodes in that level.

The ring-tree formation is highly based on the building process of a spanning tree. The root node broadcasts a message to all the nodes within its range with its own level stated as level 0. All these nodes now become level 1 and they broadcast a message with that information to their children nodes. This goes on until all nodes have a level assigned to them. In case, a node hears from more than one node, it can arbitrarily choose any one of the messages. Once the tree is built, the data about each subtree size is broadcasted towards the root node.

Once this is done, the assignment of virtual angles for all level nodes begins. The root is assigned with the set of the angle range to be utilized. Each subtree is then

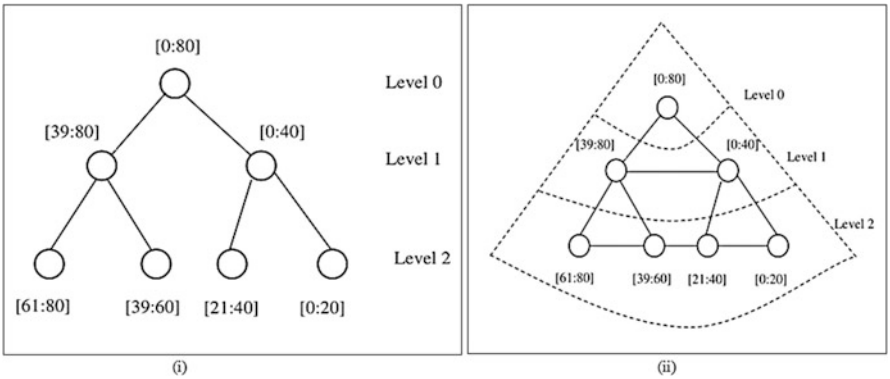

Fig. 10.2 The layout of graph embedding (GEM) coordinate naming techniques: (**a**) virtual angle assignment for tree nodes and (**b**) ordered virtual angle assignment for tree nodes

assigned a part of this set. This subset assigned per subtree is proportional to the size of the subtree (see Fig. 10.2a, for example). This method utilizes minimum energy and cost. Three messages are propagated by every node—while building the tree, propagating size of subtrees to the root node, assignment of virtual angles. After all the nodes have levels and virtual angles assigned to them, the ring tree must be aligned with the real network topology. This can be achieved by aligning the virtual angles for the nodes in an ascending or descending order until they wrap around as shown in Fig. 10.2b.

There are two schemes which work to achieve this—naïve scheme which uses the distance information and improved scheme which uses a global coordinate system that uses hops as distance parameter. The second technique, problems of having identical coordinates occur sometimes due to dense placement of nodes. This issue is resolved by sorting the virtual angles of children by their centers of mass.

Design of Virtual Polar Coordinate Routing Algorithm [39]

There are two types of routing stated for the VPCS—Naïve Tree Routing and Smart-Tree Routing. The former approach does not make use of the cross-links of the nodes while the latter uses them. Smart-tree routing is achieved by making minor changes to the naïve tree routing. Further, the VPCR technique uses greedy forwarding algorithm for routing in the ring-tree structure.

Naïve Tree Routing—Each message here is routed up until the root node before it reaches the ancestor of the destination node. Every hop is used to determine if the destination node is contained in the subtree of the current node. This is validated by checking if the level of destination node is greater than current level and the destination virtual angle is in the range of the set of angles of current node. If anything fails here, the packet is traversed upwards. Figure 10.2 shows the method described.

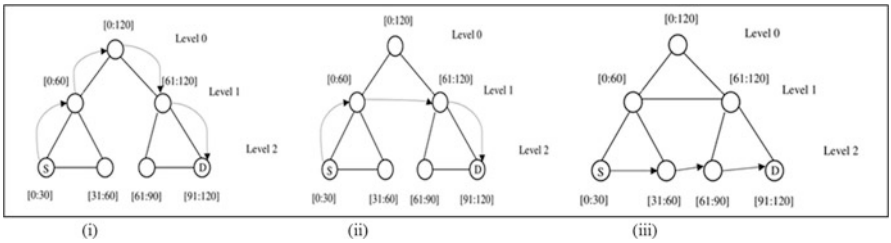

Fig. 10.3 Routing mechanisms in GEM: (**a**) naïve tree routing, (**b**) suffix-tree routing, and (**c**) virtual polar coordinate routing (VPCR)

Once the packet arrives at the ancestral node, path derived from the ancestor to destination is the shortest path available. The main issue with this algorithm is the inefficient method to route to the root node every time while searching the part. This can be eliminated with the use of cross-links as stated in the smart-tree routing. Figure 10.3a shows an example of this routing technique.

Smart-Tree Routing—As stated, this method looks for the ancestor of destination node or the destination node itself in its neighbors connected by the cross-links. This at least saves a hop to reach the destination. The method can be improved by making each node save information about its two hop neighbors. Depending on the storage efficiency per node, each node can save up to *n*-hop information about its neighbors. Figure 10.3b shows smart-tree routing.

Virtual polar coordinate routing (VPCR) – Even with cross-link information stored in smart-tree routing in case of nodes that are located far away, the packet would still be forwarded a couple of redundant hops up the tree and routed accordingly. VPCR eliminates the problem by using the cross-links to achieve lateral routing without knowing the ancestor of destination node. This is done under the assumption that the virtual angles are strictly aligned in order, either increasing or decreasing. VPCR essentially exploits the VPCSs as local coordinates to check the virtual angles. Figure 10.3c shows VPCR routing.

10.5.1.4 Hyperbolic Embedding in Dynamic Graphs for Virtual Coordinate System [40]

This is a routing and embedding technique which allows for external addition of nodes after the network has been formed without disturbing the existing network. To attain this, a simple routing algorithm called as gravity–pressure (GP) routing is introduced. Given that a path exists in the network between any two nodes, this method guarantees successful routing even when a few nodes or links between the nodes are removed after the final graph is formed. This method focuses on—constructing a graph with greedy embedding which allows the addition of a random number of nodes to the network without making any changes to the already formed

graph. The second step follows with the greedy routing algorithm which works even in the unexpected failures or downtimes of nodes or links in the graph.

Greedy Embedding Procedure—Consider a Euclidean or a hyperbolic space of n dimension. Few definitions and Lemmas are stated below prior to the online embedding procedure:

Definition 1 [40] For a connected graph G with set of vertices given by V, every vertex is assigned a VC given by $C(v)$.

Definition 2 [40] For two given nodes, the hyperbolic bisector of the Euclidean with respect to the hyperbolic line segment is the locus of points in the mapping.

Lemma 1 [40] *Consider X being the mapping of G in Euclidean or hyperbolic space. Let p and q be the nodes in X and ρ is the distance function. Parameter b is the bisector of the segment joining the mentioned nodes. In this scenario,* $\forall n \in X$, $\rho(p,n) < \rho(q,n)$ *only if the nodes p and q are in the same half space with respect to the bisector.*

Lemma 2 [40] *Given a graph G, let ST be the spanning tree for it and e be any of the edges for the graph. For every edge* $e \in ST$, $b(e)$ *is the bisector of the embedded edge given by* $C(e)$. *If all the stated is sufficed, the condition for C to become a greedy embedding of the graph G is,* $\forall e \in ST$, $b(e)$ *intersects no other edge than* $C(e)$. *Figure 10.4 shows the example of GEM in the network. The end points of curved lines originating from "r" are the sensor nodes.*

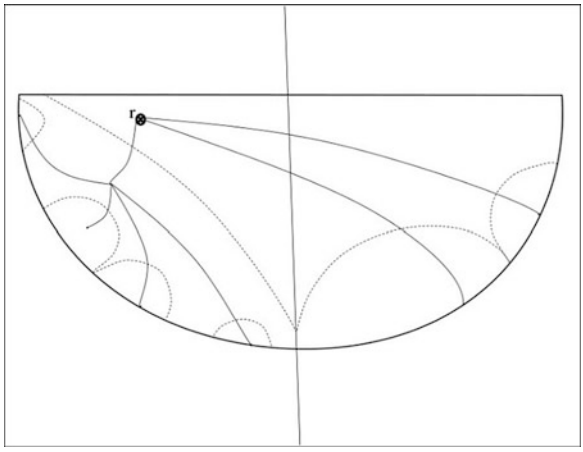

Fig. 10.4 Online embedding of nodes in a hyperbolic space with root node identified as "*r*"

The Online Greedy Embedding Algorithm

Step 1: Initialize the root node of the tree as r as follows:

1. Assign a virtual coordinate $C(r)$ in the hyperbolic plane
2. Determine the angles $\alpha = \pi$ and $\beta = 2\pi$

Step 2: For $\forall$ node $n \in G$,

1. The parent p_n – sends the virtual coordinate $C(p_n)$, $\alpha_n = \alpha_{p_n}$ and $\beta_n = \left(\alpha_{p_n} + \beta_{p_n}\right)/2$ to the node n and updates $\alpha_n := \beta_n$.
2. The node calculates c and R values and then its own virtual coordinate given by (2) [40],

$$C(n) = \frac{R^2}{(C\,(p_n))^* - c^*} + c \tag{10.2}$$

It also updates $\alpha_n := (\alpha_n + \beta_n)/2$.

Gravity–Pressure Routing—This routing protocol forwards the packet to the neighboring node which is closest to the destination than the other nodes. The analogy is to a liquid flowing through the pipes in the presence of a spherical symmetry at the center of the destination node. Due to this, the routing is referred to as gravity routing. In case of local minima, the packet is forwarded to the next node that provides the least negative progress towards the destination node. The routing takes place in two modes—Gravity and Pressure. The algorithm is as follows:

Packet Forwarded at Node N_i

Packet arrives at node N_i

If $N_i \neq N_d$ {

Gravity_Mode:
If mode = Gravity_Mode {
N_{Next}:= argmin dist (M, N_d); $M \in N_{neighbors}(N_i)$

If dist $(N_{next}, N_d) < (N_i, N_d)$ {
Forward_packet_to(N_{next});
}

Else {
mode := Pressure_Mode;
d:= dist(N_i, N_d);
$num_{visits}\,(N_i)$ += 1;
}
}

Pressure_Mode:
If mode = Pressure_Mode {
If dist $(N_i, N_d) \geq d$ {
$min_{visits} := \min num_{visits}(M); M \in N_{neighbors}(N_i)$
Candidates $(N_i) := \{M \in Neighbors(N_i) \mid num_{visits} = min_{visits}\}$;
N_{next}:= argmin dist (M, N_d); $M \in$ Candidates (N_i)
num_{visits} (N_i) += 1;
Forward_packet_to (N_{next});
}
Else {
mode := Gravity_Mode;
Go to Gravity_Mode;
}
}
}

Table 10.1 shows the comparison of VCSs embedding a graph/tree topology using parameters of Sect. 10.4.

10.5.2 Virtual Coordinate Systems Based on Hop Distances to Anchors

This section describes VCSs that rely on the hop distances to a subset of nodes. They may be generated in a distributed manner and therefore are especially useful for large-scale networks where obtaining all the information centrally and distributing the coordinates back to the networks are not trivial tasks.

Table 10.1 Comparison table for virtual coordinate systems embedding a graph/tree topology (category A)

	VCS technique			
Parameters	GL-VCS	MAP-VCS	GEM-VCS	HEDG-VCS
Use of anchors	✓	–	–	–
Efficient routing	✓	✓	✓	✓
Asserting the local minima	✓	–	✓	✓
Ability to deal with node failures and changing topologies	–	–	✓	✓
Ability to capture the network shape and voids	–	–	✓	–
Use in 3-D networks	–	–	–	–
Distributed computation of VCs	✓	✓	✓	✓
Directionality	–	–	–	✓
WSN applicability	✓	✓	✓	✓

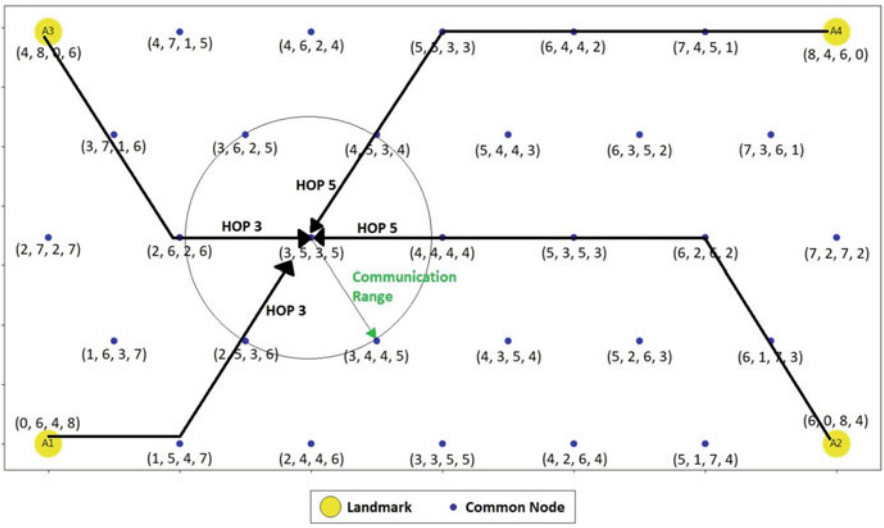

Fig. 10.5 The logical coordinate framework for logical coordinate-based routing (LCR-VCS) for a rectangular network with triangular grid placement

10.5.2.1 Anchor-Based Virtual Coordinates

Anchor-based coordinate framework provides an efficient addressing mechanism for self-organization and routing without the need for physical location information. In anchor-based VC schemes, few of the nodes in the network are marked as landmarks or anchors. Once the coordinates are evaluated, each node in the network is identified by a vector, called its virtual coordinates or logical coordinates, consisting of the hop counts to each of the anchors. Figure 10.5 illustrates the anchor-based virtual coordinates for a network where the four corner nodes are selected as anchors. This coordinate system is the basis for routing schemes such as logical coordinate-based routing (LCR-VCS) [3] and convex subspace routing (CSR) [55], as well as several derivative coordinate schemes and routing methods [59].

Anchor-Based Coordinate Space

- The logical coordinate space is constructed using the landmarks (anchors). These are network nodes selected randomly, using specific guidelines [3], or an anchor selection algorithm [22].
- Each anchor node broadcasts a beacon, initialized with a hop count 0, to its neighbors. Each node that receives a beacon and increments the hop count and broadcasts it to its neighbors. Nodes record the hop distance from each of the landmarks once the corresponding beacon is received. In case, the same beacon is received from different routes, the smallest hop count is selected. Beacons from

each anchor are thus forwarded to all the nodes in the network. Now, each node in the network has a logical coordinate vector that contains the minimum hop distance of that node from all the landmarks.

- In case of the example illustrated in Fig. 10.5, with four anchors the 2D physical plane is transformed to a 4D logical space. In general, anchor-based VCs transform the network from its original space, e.g., 2D or 3D, to a higher-dimensional space the dimensionality of which is determined by the number of anchors.
- Virtual coordinate space may suffer certain inconsistencies due to the factors such as packet losses, which, for example, may lead to assignment of null coordinates to nodes and gaps in the coordinate values among neighboring nodes.
- A significant limitation of anchor-based VCS is that node failures and node additions to the network result in changes to the topology, which necessarily invalidate the coordinates of individual nodes. However, as the routing algorithms have built-in features to overcome local minima and to find alternate paths, they can tolerate a certain amount of coordinate inconsistence. Maintaining the fidelity of the coordinates when topology changes requires regeneration of coordinates. Therefore, frequent changes to the network topology will sustain additional overhead, making these coordinates unattractive for mobile networks. A method for approximating the coordinates following node failure without regeneration is presented in [56].

Properties of Coordinate Space

- The neighborhood property—In a consistent logical coordinate space, the corresponding coordinates for the same landmark between two neighbors differ at most by one.
- Bounds on path length—Consider the two nodes with logical coordinate vectors $V(V_1, \ldots, V_A)$ and $W(W_1, \ldots, W_A)$, respectively, where A is the number of anchors. The hop count of the shortest path connecting them has a lower bound of MAX ($| V_1 - W_1 |, \ldots, | V_A - W_A |$), and an upper bound of MIN ($| V_1 + W_1 |, \ldots, | V_A + W_A |$).
- Path length—The exact hop distance between a pair of nodes cannot be calculated from their coordinates except in special and/or trivial cases. Thus, different methods are used to obtain approximate distance values for routing. The distance between two vectors, assuming orthogonal dimensions, is defined as the L^N norm. LCR-VCS [3] uses L^2 norm, while CSR [55] uses L^2 norm with respect to a subset of anchors only.

$$D = N\sqrt{\sum_{i}^{n} \left(|Vi - Wi| \right)^N} \tag{10.3}$$

Logical Coordinate-Based Routing [2]

This method is among the earliest routing schemes to use anchor-based VCs in the context of sensor networks.

- Landmark selection uses a set of guidelines to minimize potential delivery failures during greedy forwarding. This algorithm runs through three phases, clustering, voting, and landmark admission.
- The logical distance between two nodes is defined as norm-2 distance between the logical coordinate vectors of the nodes.
- Greedy forwarding is used to identify the node to which a packet is forwarded in the next hop, i.e., the next node is selected such that the distance to the destination is minimized. In case of local minima, a backtracking algorithm is used. These local minima are referred to as logical voids, in comparison to physical voids in geographic coordinate spaces.
- Backtracking algorithm is based on three main rules:

  Rule 1—Each new packet or returned packet is forwarded to the neighbor with minimum distance to the destination excluding the predecessor and any node to which the packet was forwarded earlier.

  Rule 2—If no neighboring node satisfies the above condition, the packet is returned to the predecessor.

  Rule 3—If the packet is forwarded to the current node again by a neighbor node, it is returned to the neighbor.

- CSR [55] reduces the likelihood of encountering logical voids during routing by dynamically selecting three anchors at a time that enclose the current node and the destination node.

Anchor Selection Algorithms

Most of the VCSs assign coordinates to individual nodes based on a set of anchor or landmark nodes in the network. The dimension of VC space depends on the number such anchor nodes, i.e., with M anchors, a node would have an M-tuple as the coordinate vector. The performance of the VCS algorithm is highly affected by the choice and the number of anchors. Many of the protocols have been evaluated with random anchor placement while others have specific declarations about selection of anchors. The technique in [15] suggests placing the anchors on the boundary of the sensor network given that we are aware of the boundary nodes. This would require identification or knowledge of the boundary nodes. There are few techniques which too mostly depend on electing anchors furthest apart from each other, and frequently such nodes are likely to fall on the boundaries. Depending on the network and the number and placement of anchors, it is possible for different sensor nodes to end up with identical VCs. Detection of identical coordinates in a distributed manner is not trivial. The single anchor-based VCS [57] uses depth first search algorithm for computing the coordinates for sensor nodes in a network. For optimum results, the anchor is placed at the center in this scheme.

Here, we summarize several anchor selection and placement techniques, namely Random Anchor Placement, Single Mobile-Based Anchor, and Extreme Node Search (ENS) Algorithm. Anchor selection problem includes both the selection of

an adequate amount of anchor nodes and their placement in the network to achieve satisfactory routing results. Often, these two steps are interdependent on each other.

Random Anchor Placement—This procedure selects random nodes from the network to serve as anchors. This is easy to implement in a distributed manner as each node can decide with a certain probability to become an anchor. In some cases, the anchors nodes are chosen in such a way that they are located far apart from each other, e.g., selected from among the boundary nodes. In this case, a node needs to know or can find out whether it is on a boundary. Another related aspect is selecting the number of anchors. The smallest number of anchors needed is highly dependent on parameters such as the number of nodes in the network, network topology, density of nodes, and communication range. To avoid identical coordinate problem and also achieve good performance, it is customary to select a reasonably high number of anchors rather than trying to minimize the number of anchors.

Single Mobile-Based Anchor [57]—In this method, the local coordinate of each node is derived from local distance measurements from an anchor node. The anchor node is a mobile robot that traverses around to help the coordinate assignment for the nodes. The robot device is GPS-enabled which enables it to identify its location. This information can be exploited to find out VCs for network nodes. While moving around, the mobile robot transmits its position coordinate to the sensor nodes within its communication range. The distance between a sensor node n and the mobile robot is determined using the RSSI. It calculates Barycentric coordinates for the nodes and then uses a distributed routing algorithm. This method assigns unique VCs to all the nodes in the network successfully. Distance measurement errors could sometimes cause inaccurate results.

Extreme Node Search (ENS) [22]—This technique is a simple but effective anchor placement scheme to find both the number and placement of the anchors. Anchors selected by this algorithm correspond to extreme nodes of the network such as corner nodes and boundary nodes. It does not rely on a priori knowledge about the position of nodes, such as whether it is on a boundary. ENS starts by selecting a random pair of anchors to generate a directional virtual coordinate (DVC) [7] for every node in the network. This DVC is then used to identify extreme nodes, which become the anchors. ENS follows three steps for anchor selection

Step 1: Two nodes are randomly selected from the network to serve as initial anchors for extreme node identification. The network is flooded with beacons originating at these two nodes providing every sensor node a two-tuple VC. Using these two values, each node computes its DVC using the DVCS algorithm [23].

Step 2: Each node identifies whether it is a local minimum/maximum in terms of the DVC within its h-hop neighborhood [23]. This can be carried out in parallel and involves communication in h-hop neighborhood. The value of h is typically very small, e.g., in the range of 2–5, and the value typically determines the total number of self-identified extreme nodes.

Step 3: Any node that identifies itself as a local minima or maxima in its h-hop neighborhood, it becomes a new anchor node of the network. Thus, VC generation begins with each such anchor. As the VCs are already available for the two random

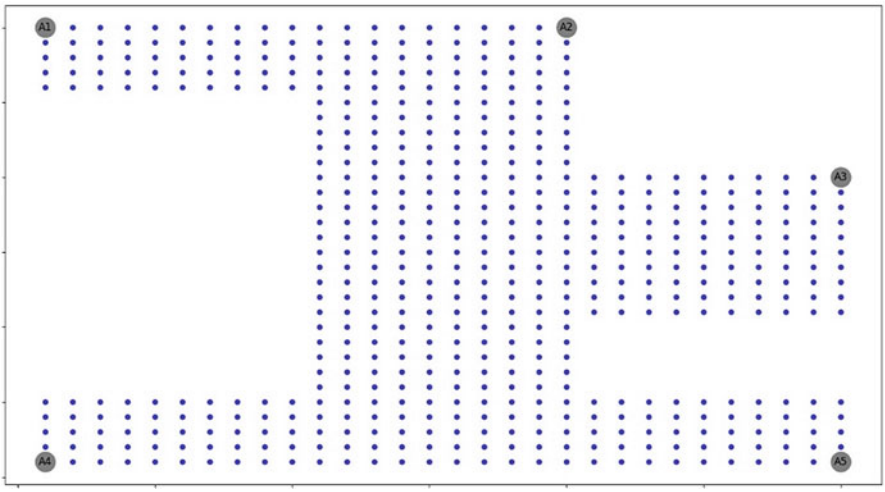

Fig. 10.6 Anchor placement using the extreme node search (ENS) algorithm

anchors selected originally, they could be considered as a part of the anchor set without any additional cost though they may not be extreme nodes. Figure 10.6 shows extreme nodes identified to be anchors for an odd-shaped 2-D network using the ENS technique.

10.5.2.2 Axis-Based Virtual Coordinate Assignment Protocol [41]

ABVCap is another approach based on assignment of VCs in WSNs without the need for geographical coordinates, generation of which requires GPS or distance measurement. It is a VC assignment technique that provides packet delivery across a network. ABVCap is based on the VC assignment protocol with quite a few improvisations. Each node in the WSN is static or quasi-static and has a unique ID and same transmission range (default = 1.5) [41]. It assigns at least one 5-tuple VC (*u.lo, u.la, u.rp, u.up, u.dn*) per node in the network by following assigned steps. A node in the network may have more than one 5-tuple VC. Such nodes are perceived as multiple virtual nodes at one location. Once the VC assignment is complete, the routing protocol follows. It consists of two phases—choosing a VC among multiple VCs for the node, and routing the received packet based on that VC.

Assignment of VCs

The coordinate assignment follows a four-phase procedure. All the nodes have the same transmission range of $R = 1.5$. Each node has a unique identifier (ID) that is used to break ties. Four stages with election of anchors followed by implementation of axes (parallel of Latitude and Meridians) assign the VCs.

Election of Anchors X, Y, Z, Z'

- W is chosen as the sink node randomly.
- X is chosen as the node that has the maximum hop distance from W. Next, Y is chosen as the node that has the maximum hop distance from X.
- After this, Z is chosen as the node that has the maximum hop distance from W among all nodes whose x and y coordinates each satisfy the relationship, $x = y \pm 1$. In case of parity, the node that has the maximum ID value is selected.
- After this, the node that has the maximum hop distance from Z among all nodes whose x and y coordinates each satisfy the relationship $x = y \pm 1$ is selected to be anchor Z'. In case of parity, the node having the maximum ID value is selected.
- By the conclusion of this phase, each node has x, y, z, and z' coordinates, where the z' coordinate denotes the hop distance of the node from anchor Z'.

Establishment of Axes: Parallel of Latitude and Meridians

- Once the anchors have been elected, axes are established which are used to assign VCs to the nodes. The parallel of latitude is constructed such that it consists the nodes in one of the shortest paths from anchor Y to anchor X.
- The meridians are formed such as it consists of the node u in the parallel of latitude whose x coordinate is equal to the meridian number, the nodes in one of the shortest paths from node u to anchor Z, and the nodes in one of the shortest paths from node u to anchor Z'. Figure 10.7 shows the establishment of the axes.

Parallel of Latitude: Anchor Y generates a *Parallel SET* message, which is forwarded by a node to another node whose x coordinate is smaller by 1 until anchor X is reached. Each node that receives the Parallel SET message is in the parallel of latitude.

Meridians: Every node in the parallel of latitude generates a *Meridian* $i + SET$ message containing its z coordinate, and a *Meridian* $i - SET$ message containing its z' coordinate. The *Meridian* $i + SET$ (or correspondingly the *Meridian* $i - SET$) message is forwarded by a node to another node whose z (or z') coordinate is smaller by 1 until anchor Z (or Z') is reached. If the case where more than one node is eligible to be forwarded, the node having the minimum distance to the node that generates the message is selected.

- The distance between two nodes u and v is defined as [41]:

$$|u \cdot x - v \cdot x| + |u \cdot y - v \cdot y| \tag{10.4}$$

$u \cdot x$, $u \cdot y$, $v \cdot x$, and $v \cdot y$ denote the x coordinate of u, the y coordinate of u, the x coordinate of v, and the y coordinate of v, respectively.

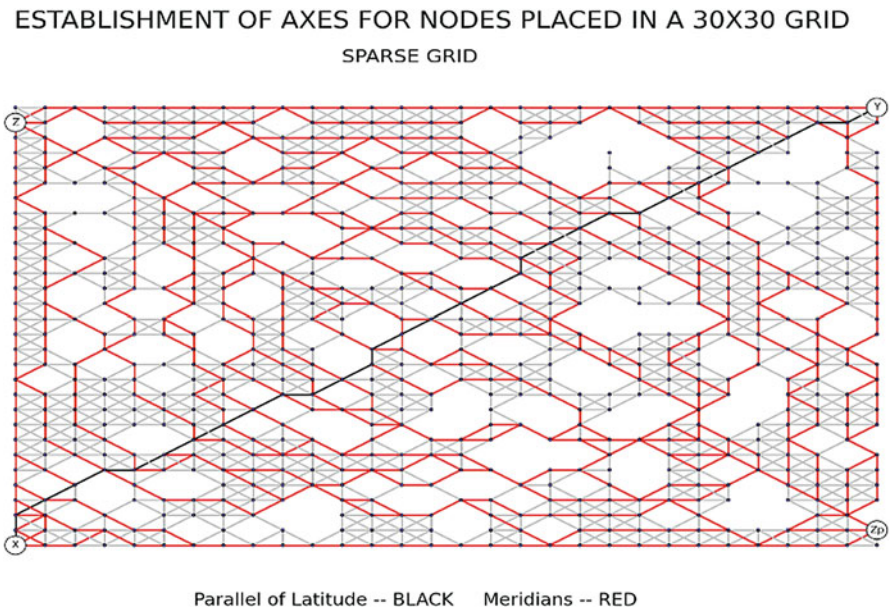

Fig. 10.7 Establishment of axes—parallel of latitude and meridians

Assignment of Longitude, Latitude, and Ripple Coordinates

Once the axes have been established, VCs are assigned to all the nodes based on their placement on/from axis.

Parallel of Latitude Nodes: When a node receives the parallel SET message (for generation of parallel of latitude), the longitude, latitude, and ripple coordinates are assigned to $(x, 0, 0)$. Here, x is the *X-coordinate* of the node.

Meridian Nodes: Once a node receives the *Meridian* i + *SET* message, the longitude coordinate is assigned to i (the meridian number), the latitude coordinate to the z coordinate of the node that generates the *Meridian* i + *SET* message minus its own z coordinate, and the ripple coordinate to 0 (for nodes located near to Z), i.e., $(i, Z_s - Z_n, 0)$.

If a node receives the *Meridian* i − *SET* message, the longitude coordinate is assigned to i (the meridian number), the latitude coordinate to its z' coordinate minus the z' coordinate of the node that generates the *Meridian* i − *SET* message, and the ripple coordinate to 0 (for nodes located near to Z'), i.e., $(i, Z_s' - Z_n', 0)$.

Non-Axial Nodes: Once the coordinates have been assigned to axis nodes, a new phase begins where the longitude, latitude, and ripple coordinates are assigned to the rest of the non-axial nodes.

Every axis node generates a *3COOR_SET* message which contains the corresponding longitude and latitude coordinates and a hop counter initially set to 1. This counter increases with every hop taken. When a non-axis node receives a

3COOR_SET message, it locally broadcasts the message, simultaneously assigning the longitude and latitude coordinates to the longitude and latitude coordinates of the node that generates the message and the ripple coordinate to the hop counter. In a condition where more than one message is received from different nodes, the non-axis node broadcasts the message with the lowest hop count and assigns its first three VCs accordingly [41].

Assignment of Up and Down Coordinates

- After the assignment of the longitude, latitude, and ripple coordinates, this stage follows.
- For all the virtual nodes, the up and down coordinate is set accordingly depending on the longitude coordinate of the neighbor. Accordingly, an *UP_SET* or *DOWN_SET* message is generated that determines the up/down coordinate of the node.
- Furthermore, depending on several scenarios, these coordinates are determined for all the nodes in the network.
- By the end of this phase, all nodes in the network have one or more 5-tuple VC assigned to them which is used further for routing in the network.

Routing Protocol

- It is assumed that all the nodes receive all multiple VCs of all the neighbors. It is also assumed that the VC of the destination is unique.
- The packet contains the longitude and latitude coordinates of the destination node and a routing direction bit (set to 1 if the longitude coordinate of the source *(s.lo)* is less than the longitude coordinate of the destination *(d.lo)*).

The routing takes place in two phases

- Choosing VCs among multiple VCs.
- Routing of the packet based on the chosen VCs.

ABVCap scheme sets a path for every source–destination pair successfully. With virtual coordinate assignment protocol (VCap) or Euclidean routing, the network density for ABVCap is directly proportional to the delivery rate. This is intuitive as a lower density network has more dead-end nodes due to the occurrence of more holes. In a high-density network, the delivery rate approaches 100% [41].

10.5.2.3 Directional Virtual Coordinate Systems [23]

With the new emerging techniques for assignment of VCs in a system, there have been several advances to overcome the limitations in the existing techniques.

Directional virtual coordinate system (DVCS) is a systematic approach designed to eliminate the problem of lost directionality in the conventional anchor-based VCS. DVCS starts with the anchor-based VCs for all the nodes across the network. It uses a transformation that combines two anchor-based coordinates at a time to regain the lost directional information; DVCS help mitigate the logical voids associated with anchor-based VCS by providing more geographic-like set of coordinates. This coordinate assignment technique also allows for novel routing strictly with the help of deterministic algorithms for constrained tree network. DVCR significantly outperforms existing VCS routing schemes CSR [55] and LCR [3], while achieving a performance like or better than geographical routing scheme GPSR, but without the need for node location information [23].

DVCS proposes novel technique for transformation of traditional VCs to directional VCS. Properties of DVCS are discussed with assignment of coordinates in a constrained tree network. Also, an efficient DVCS routing protocol which uses DVCS coordinate assignment is elucidated. The routing protocol is compared with other protocols like with CSR [55], LCR [3] and geographical routing scheme called GPSR [10]. DVCR outperforms CSR and LCR with a noticeable value achieving similar performance as GPSR.

Directional Virtual Space Transformation [23]

As described earlier, the anchor-based VCs of a node correspond to the hop distances to the respective anchors. These coordinates spread concentrically around the anchor thus losing the directionality information. To illustrate the transformation, first consider the 1-D network shown in Fig. 10.8 with 8 sensor nodes. Let A_1 and A_2 be two anchors. The hop distances from the two anchors to node N_i, $(h_{N_iA_1}, h_{N_iA_2})$ are shown in Table 10.2.

The anchors A_1 and A_2 are $h_{A_1A_2}$ hops apart. From the values for $h_{N_iA_1}$ and $h_{N_iA_2}$, we see that h_{NiAi} spreads concentrically about the anchor, hence losing directional

Fig. 10.8 One-dimensional network with two anchors

Table 10.2 Virtual coordinates for 1-D network from Fig. 10.8

NODE ID	A1	N1	N2	N3	N4	N5	N6	A2	N7	N8
h_{NiA1}	0	1	2	3	4	5	6	7	8	9
h_{NiA2}	7	6	5	4	3	2	1	0	1	2
$h_{NiA1} - h_{NiA2}$	−7	−5	−3	−1	1	3	5	7	7	7
$h_{NiA1} + h_{NiA2}$	7	7	7	7	7	7	7	7	9	11
$f(h_{NiA1}, h_{NiA2})$	−1	−2.5	−1.5	−0.5	0.5	1.5	2.5	3.5	4.5	5.5

information. $\left(h_{N_i A_1} - h_{N_i A_2}\right)$ provides a sense of directionality only for the region between the anchors, while $\left(h_{N_i A_1} + h_{N_i A_2}\right)$ provides the same for region outside the anchors. By taking the product of the two terms, the function $f\left(h_{N_i A_1}, h_{N_i A_2}\right)$ achieves a sense of directionality in both the regions. Thus, a coordinate in DVCS is formed by combining the ordinates $\left(h_{N_i A_1}, h_{N_i A_2}\right)$ to form a single ordinate,

$$f\left(h_{N_i A_1}, h_{N_i A_2}\right) = \frac{1}{2h_{A_1 A_2}} \left(h_{N_i A_1} - h_{N_i A_2}\right)\left(h_{N_i A_1} + h_{N_i A_2}\right) \tag{10.5}$$

The table lists hop distances from the anchors to any of the node N_i with the function $f(h_{NiA1}, h_{NiA2})$ giving the VCs mapped from the physical domain. With this transformation, the local minima problem is eliminated in this linear topology. Each node in this 1-D network can be viewed as a point in the vector space with its unit vector pointing the direction of A_1A_2 given by [23],

$$\overrightarrow{f}\left(h_{N_i A_1}, h_{N_i A_2}\right) = f\left(h_{N_i A_1}, h_{N_i A_2}\right) \overrightarrow{u}_{A_1 A_2} \tag{10.6}$$

Similarly, extending the concept to a 2-D grid network, a mapping from anchor-based VCs to a geographic coordinate like directional VC is obtained. The transformed coordinates are given by $[f(A_1, A_2), f(A_3, A_4)]$ which delivers directionality. Figure 10.9 illustrates the extension of DVCS concept to a 2-D network.

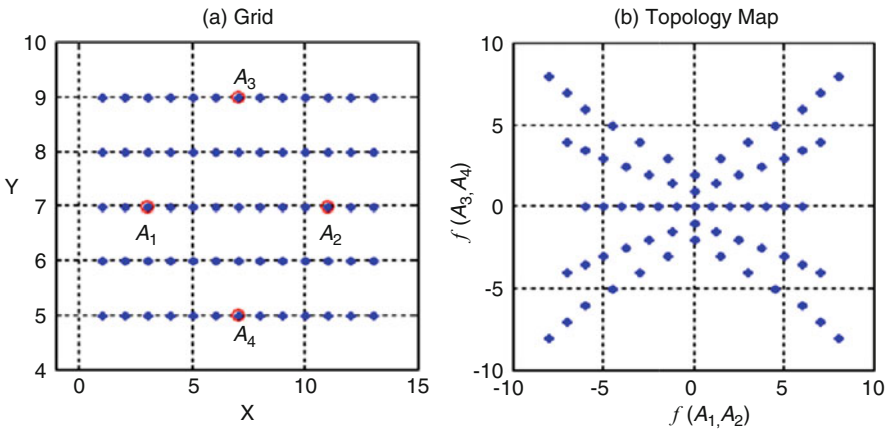

Fig. 10.9 Directional virtual coordinates for a 2-D network with four anchors [23]

Routing in Directional Virtual Coordinate System

DVCS offers a novel routing scheme based on the Directional VCs. In a 2-D WSN, let the mapped directional VCs of a node be $N_i \equiv [N_{i1}, N_{i2}]$ and of the destination be $N_d \equiv [N_{d1}, N_{d2}]$. Here, the L^2 distance between N_i and N_d is given by [7],

$$D_{N_i N_d} = \sqrt{\sum_{\forall j} \left(n_{ij} - n_{id}\right)^2}; \; j = 1 : J \leq C_2^M \tag{10.7}$$

It uses greedy forwarding to pass the packets from a node to its neighbors. The local minima problem is eliminated by the minima node performing an approximate estimation of hop distance between itself and from neighbors based on the stated algorithm. Here, we assume that L^1 in the transformed domain is a good representation of the hop distance [23]. The estimation stated needs minimum two directional ordinates.

Using greedy forwarding, the packet is forwarded towards destination. Average routability, average path length, and average energy consumption per successful packet delivery results for five different types of networks are given in Fig. 10.10.

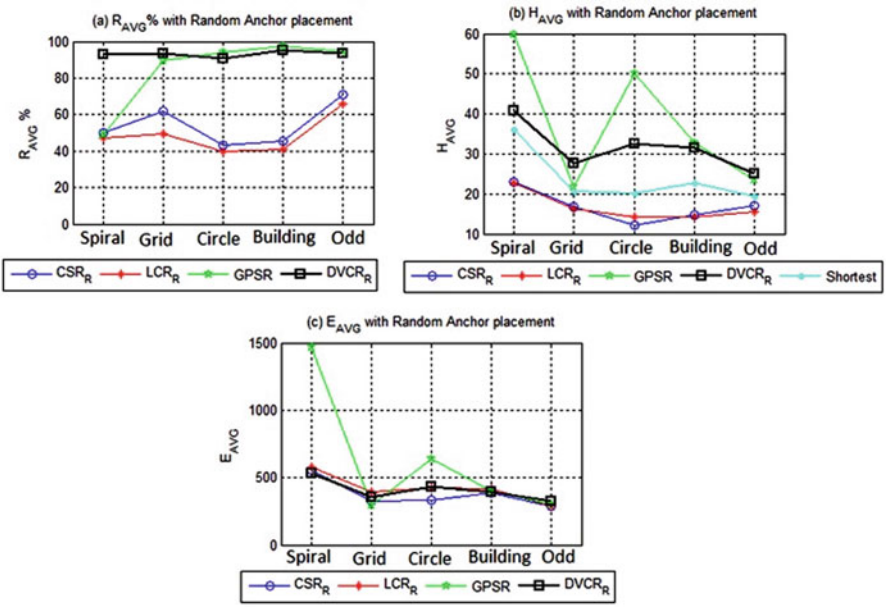

Fig. 10.10 Average routability, path length, and energy consumption results for different networks for directional virtual coordinate system (DVCS) compared with convex subspace routing (CSR) and LCR [23]

Table 10.3 Comparison table for decentralized virtual coordinate systems (category B)

Parameters	VCS technique		
	LCR-VCS	ABVCap	DVCS
Use of anchors	✓	✓	✓
Efficient routing	✓	✓	✓
Asserting the local minima	✓	✓	✓
Ability to deal with node failures and changing topologies	–	–	–
Ability to capture the network shape and voids	–	✓	✓
Distributed computation of VCs	✓	✓	✓
Directionality	–	–	✓
WSN applicability	✓	✓	✓

Table 10.3 shows the comparison for the decentralized VCSs using the parameters in Sect. 10.4.

10.5.3 Topology Preserving Maps

10.5.3.1 Topology Preserving Maps: Extracting Layout Maps of Wireless Sensor Networks from Virtual Coordinates [35]

The elementary anchor-based VCS characterizes each node with a coordinate vector consisting of distances to each of the anchor nodes. In the process, the layout information of the WSN such as physical voids, obstacles, shape, and even relative physical positions of sensor nodes with respect to (x, y) directions are lost. TPM technique uses the singular value decomposition (SVD) scheme to recover the network layout in 2D and 3D network surfaces or volumes by isolating and removing the radial component that dominates the VCs. The topological coordinates (TCs) are computed using the coordinates of a subset of nodes. Topology preservation error (E_{TP}), defined to capture the among and degree of node flips, is used to evaluate 2-D TPMs. The defined method extracts TPMs with less than 2% error. Topology coordinates provide an economical and efficient alternative to geographical coordinates [35].

This scheme achieves a map that is homomorphic to the physical layout of the network absorbent of the information about node connectivity, physical layout, and physical voids. The topology map itself is not a physical map, but a distorted version of it taking into consideration the connectivity parametric.

2-D Topology Preserving Maps from VCs

Consider a 2-D WSN with N nodes and A anchors. The VCs for each node are a vector of length M (i.e., hop distance to each of the anchors). Let P be the $N \times M$

matrix with the VCs for all the nodes in the network with respect to each anchor. The *i*th node corresponds to *M*-long VC vector for the node.

$$P = \left[h_{n_i} A_j\right] \tag{10.8}$$

Here, $h_{n_i} A_j$ is the hop distance from the node n_i to anchor A_j. As per the anchor method for determining the VCs, the number of anchors is much smaller subset than the number of sensor nodes in the network, i.e., $M \ll N$. The anchor coordinates transform the 2-D network to an X-dimensional coordinate space. Thus, the goal is to extract a 2-D depiction of the network from this X-dimensional space. The SVD [53, 58] of the matrix P is denoted by,

$$P = U \cdot S \cdot V^T \tag{10.9}$$

where, U, S, *and* V are $N \times N$, $N \times M$, and $M \times M$ matrices, respectively [58]. The SVD extracts the characteristics of the dataset P with two principle components in such a way that it gives an optimal bias for P. U and V are the principle components (PCs) of P, the dot product of that gives the $N \times A$ P_{SVD} matrix describing every node with a new set of length coordinate vectors. The columns of P_{SVD}, i.e., the PC values of the VC set are arranged in the descending order of information about the original coordinate set with the first PC capturing the highest variance of the dataset, and each succeeding component with the highest variance possible under the limitation that it be orthogonal to the preceding components [35].

Figure 10.11 shows the plot of the P_{SVD} components against physical coordinates. This shows the variation of each component with the physical map layout.

The second and third columns of P_{SVD} provide a set of 2-D Cartesian coordinates to determine node positions, less burdened by the dominant radial information in VCs, which was captured by the first column. Hence, instead of using the M coordinates of a row of to characterize a node, the second and third columns of P_{SVD} are used as Cartesian coordinates to plot an approximate map of the network.

$$[X_{TC}, Y_{TC}] = \left[P_{SVD}^{(2)}, P_{SVD}^{(3)}\right] \tag{10.10}$$

The TCs are computed without any physical, directional, or position information beyond the radial information (hop distance) with respect to the anchors.

3-D Topology Preserving Maps from VCs

Sophisticated WSN deployment nowadays is no more restricted to 2-D dimensions. The WSNs extend to 3-D surfaces and volumes. These surfaces may wrap around, hence affecting VC propagation in complex ways.

As per the SVD principles, the second, third, and fourth PCs are orthogonal to the first ordinate while being perpendicular to each other [58]. Extending the

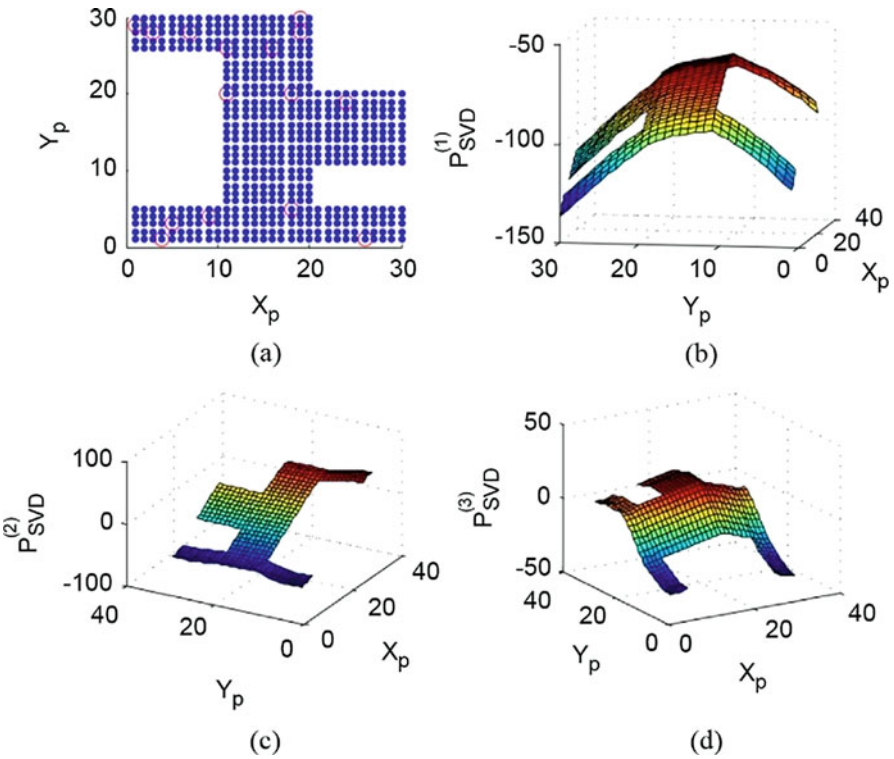

Fig. 10.11 (**a**) Odd-shaped 2-D network with 550 nodes. (**b–d**) Three PCs plotted against the physical positions with random anchors [35]

2-D concept, the radial propagation of coordinates is captured by the first PC. The lowest value is recorded at the center of the surface while it increases towards the edges resulting in a convex variation along the height. Thus, removing the first PC from consideration allows us to uncover linear patterns embedded in the VC set. The second PC varies monotonically along the height of the cylinder and can be used to obtain the coordinates for the topology map. Additionally, third and fourth PCs are taken as x and y coordinates. They are directionally distributed in such a way that they are orthogonal to each other while being normal to the second PC [58]. Figure 10.12 shows a network with 900 nodes on a cylindrical surface and the TPM for the same. Figure 10.13 shows the first four PCs plotted for all the nodes in the network.

Extending the concept for 2-D coordinates, the TCs for 3-D can be given as,

$$[X_{TC}, Y_{TC}, Z_{TC}]_{(i)} = \left[P^{(2)}_{SVD}, P^{(3)}_{SVD}, P^{(4)}_{SVD}\right]_{(i)} \tag{10.11}$$

These results hold for 3-D volumes as well where first PC in that case will radially proliferate outward from the center of the volume.

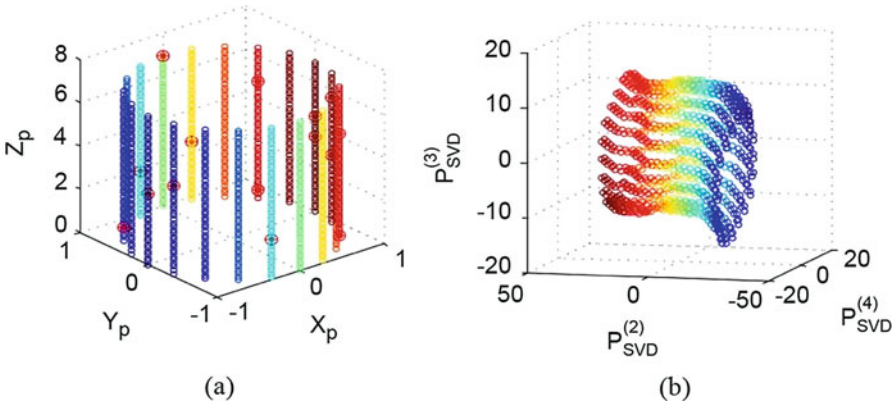

Fig. 10.12 (**a**) A 900-node network on cylindrical surface with 20 random anchors. (**b**) Topology preserving map for (**a**) [35]

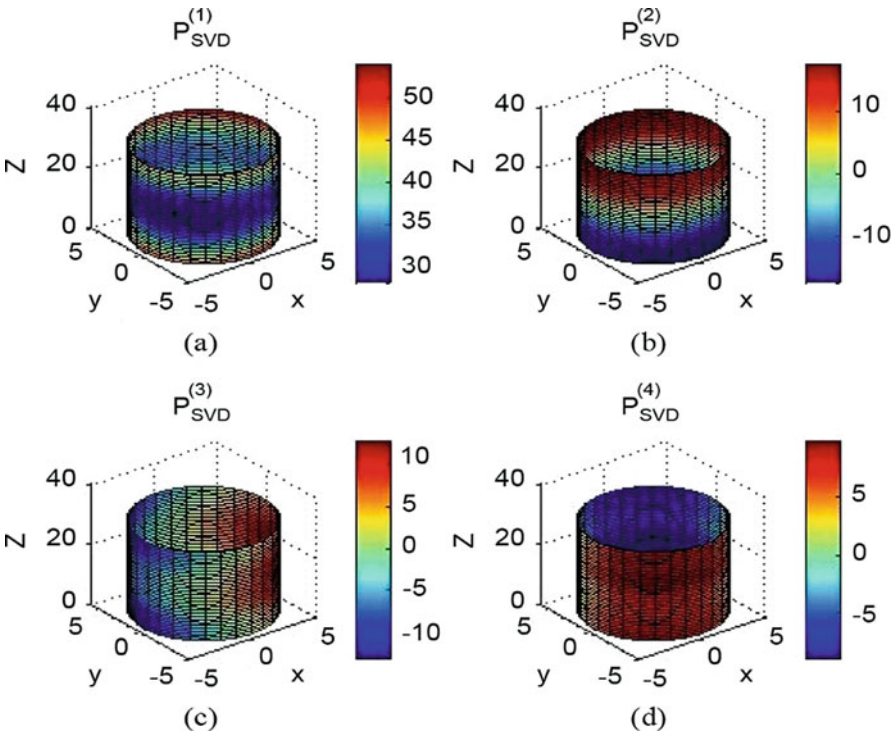

Fig. 10.13 First four PCs of the cylindrical network plotted as a color map on the network surface [35]

In addition to computing the TCs for the network and plotting the physical map, this method works on developing a metric to evaluate the 2-D topology map preservation and TPM-based routing. This is vital to investigate the effectiveness of the proposed scheme. The visual inspection provides preliminary evidence of its effectiveness. In addition to that, a parameter topology preservation error (E_{TP}) is introduced for testing.

Routing Using the Topology Preserving Maps

TPM proves to be a better scheme for GR than the original physical map. The former is based on actual GPS connectivity information rather than the node position. The TCs can be efficiently used for selecting the accurate neighbors to forward the information and achieve efficient routing. It is evaluated as follows,

$$P\left[\text{Selecting Correct Neighbor}\right] = \sum_{N_j \in N} \sum_{N_i \in N} \frac{\text{No.of times a } N_j \text{ selected a correct neighbor to FWD the packet when destinaton is } N_i}{\text{Total No.of nodes}} \tag{10.12}$$

The VC generation mostly needs to be done only once during initialization for static WSNs. Hence, TCs need not be updated frequently. Thus, the cost incurred in calculating Cartesian coordinates may be more than compensated by efficiency gains in terms of performance during long-term operation [61]. For example, as demonstrated in [58], the GLR scheme that uses VCS and TPM both to overcome shortcomings in each other's domains successfully outperforms the physical information-based routing scheme—GPSR [10].

10.5.4 Coordinate Systems Using Network Properties

10.5.4.1 Vivaldi (Network Coordinate System) [42]

Emerging network applications and services are highly intelligible and flexible due to the ability of choosing their own communication paths among the available ones. These communication paths are chosen based on certain network measurement parameters such as latency. However, explicit measurements by injecting probes in network would generate a huge amount of measurement traffic in the network making it infeasible to obtain measurements. This is the conventional way of attaining network information for efficiently choosing communication paths in a network. Network measurements benefit several areas such as peer-to-peer file sharing applications, content distribution systems, and decentralized web caches. To make these measurements viable with minimum effort and cost, NCSs have been

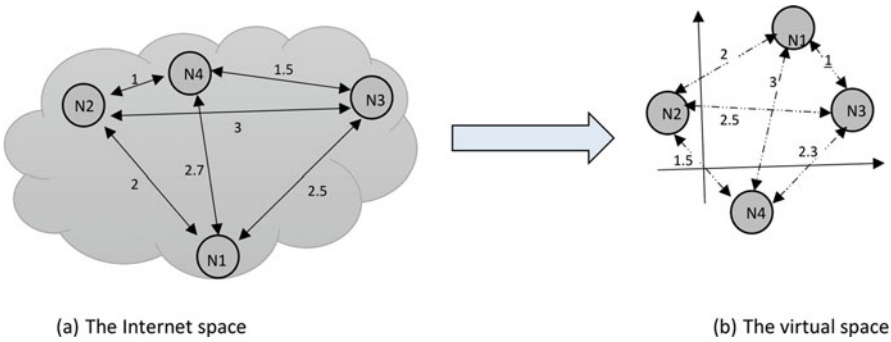

Fig. 10.14 Mapping of Internet space to the virtual space

proposed. With constraints, such as limited resources per node, NCS allows the host to perform measurements with minimal resource consumption [47].

The foundation of an NCS is to model the Internet as any geometric space and characterize the position of all the nodes in the network by coordinate in this space [24, 60]. The distance between any of the nodes could be given as the geometric distance between them. Network systems usually involve construction of overlays and look-ups. With such structures, it proves to be difficult to take measurement as compared to measurement of node distances in a standard planar network. Moreover, injecting probes in the network for these proximity measurements would further lead to complications and unnecessary overheads. Additionally, in the case of ever-changing topologies, it is impractical to take measurements each time the node changes positions or fails in the network. Figure 10.14 shows modeling the internet as a geometric space and mapping it to a virtual space.

In the real world, the hosts can be present anywhere. These hosts are connected to each other and the distances show one of the network measurement parameters (say RTT). These distances are mapped into the virtual space using an estimate from a conversion distance function in the geometric space.

To achieve a consent between requirements for optimum performance for the overlay networks and scalability constraints imposed by underlying IP networks, NCSs for estimating network distances and latencies have been proposed [24]. This technique offers several advantages like:

- *Easy support to peer-to-peer applications*—Node positioning plays a crucial role in P2P applications to maintain proximity information with neighboring nodes.
- *Accuracy*—The mapping from the physical domain to the virtual domain is reasonably accurate. The errors in today's NCS are still acceptable for most applications instead of acquiring exact proximity information using physical measurements.
- *Scalability*—This is the most imperative con of using NCS. Estimation of the network distances using the locally computed NCS coordinates for the nodes generates a very low overhead irrespective of the size of the network.

Vivaldi is a fully distributed and decentralized, requiring no fixed infrastructure system [42]. It has minimum communication requirements and hence it is possible to piggyback on the communication patterns of the application using it. It is highly efficient in terms of scalability and resource consumption. The host node can compute coordinates for its own by collecting the latency information from few other nodes. Vivaldi calculates synthetic (virtual) coordinates for the nodes to predict the RTT to other nodes given a few measurements. Consider two nodes N_1 and N_2. Once node N_1 learns the synthetic coordinates of node N_2, no direct measurements for calculation of RTT from N_1 to N_2 are required. It would be predicted by distance between these nodes accurately enough. The working of these synthetic coordinates highly depends on the properties of the Internet. These properties are not accurate which results in the modeling of the Internet space not being very accurate. Thus, a 2-D coordinate system does not suffice for latency calculations. Vivaldi eliminates the prediction errors by augmenting these 2-D coordinates with height.

Certain challenges are faced while mapping the Internet into a suitable metric space. The space should deal with several networking phenomena such as queuing, routing, etc. Most of the applications evolving today are exhaustive which essentially needs the coordinate systems to be scalable. For small-scale networks, direct measurements are practical. Also, peer-to-peer applications follow a distributed approach which needs the coordinate systems to be decentralized. Additionally, with changing network conditions such as node failures, topology changes, etc., we need our coordinate system to be adaptive to these changes. Vivaldi helps achieve all the above conditions and predict the latency with low error.

Vivaldi Algorithm

Vivaldi assigns VCs, also, called as synthetic coordinates to all the nodes in a WSN such that the RTT between the nodes could be accurately predicted by the distance between them in that coordinate space. It uses an n-dimensional coordinate system with standard Euclidean distance function. Figure 10.15 shows the algorithm for simple Vivaldi.

The Vivaldi algorithm works on the foundation of centralized algorithm where each node computes and adjusts its own coordinates on a timely basis based on measured RTT between itself and few other nodes using the synthetic coordinates. In a spring system, node maintains its own coordinates starting with origin; measuring RTT to other node and learning its coordinates. Each node in the network moves a short step by corresponding spring *dir.* Each consecutive step of a node reduces its error with respect to another node in system. Eventually, as the nodes in network communicate continuously with each other, they converge to form coordinates that predict RTT accurately.

The Simple Vivaldi Algorithm

Step 1: Distance between node n and node m is computed as rtt ms with coordinates of node m as x_m.

Step 2: Simple Vivaldi Algorithm (rtt, x_m).

Step 3: Compute the error for the above sample Er as,

$$Er = rtt - \|x_n - x_m\|$$

Step 4: Compute the direction Dir that the error Er is causing,

$$Dir = u(x_n - x_m)$$

Step 5: Compute the force vector f that is proportional to the error Er,

$$f = Dir \times Er$$

Step 6: Move a small step in the direction of the force f,

$$x_n = x_n + \delta \times Dir$$

Fig. 10.15 The simple Vivaldi algorithm [39]

If node A with coordinates X_A learns about node B with coordinates X_B, and the RTT between is measured as *rtt*, the node A updates its coordinates using the update rule given by [42],

$$X_A = X_A + \partial \times (rtt - ||X_A - X_B||) \times u\,(X_A - X_B) \tag{10.13}$$

Here, ∂ is time step which determines the rate of convergence. Large values of ∂ cause adjusting the coordinates in large steps which might cause oscillations and, hence, failure to converge. Consecutively, smaller values of ∂ would take long time for convergence. To obtain both fast convergence and avoidance of oscillation, it is helpful to use a constant fraction ($c_c < 1$) of node's estimated error. The local error decides how early convergence can be achieved with less oscillations. The Vivaldi time step is given by [42],

$$\partial = c_c \times \frac{\text{local error}}{\text{local error} + \text{remote error}} \tag{10.14}$$

∂ can be constant or kept adaptive. With adaptive ∂, the nodes are required to be aware of the accuracy of their coordinates. The adaptive ∂ Vivaldi algorithm is shown in Fig. 10.16.

In Vivaldi, the prediction error is the performance metric; squared error function is given by [42],

The Adaptive Vivaldi Algorithm

Step 1: Distance between node n and node m is computed as rtt ms with coordinates of node m as x_m.

Step 2: Error estimate is depicted as Er_m and the coordinate of source node is given by x_n.

Step 3: Tuning parameters are given as constants c_e and c_c.

Step 4: Adaptive Vivaldi Algorithm (rtt, x_m, Er_m),

Step 5: Sample weight balances local and remote error.

$$w = Er_m/(Er_n + Er_m)$$

Step 6: Compute the relative error for the sample,

$$Er_s = \frac{|\| x_n - x_m\| - rtt|}{rtt}$$

Step 7: Update weighted moving average of the local error,

$$Er_n = Er_s \times c_e \times w + Er_n \times (1 - c_e \times w)$$

Step 8: Update local coordinates,

$$\delta = c_c \times w$$

$$x_n = x_n + \delta \times (rtt - \| x_n - x_m\|) \times u(x_n - x_m)$$

Fig. 10.16 Adaptive ∂ Vivaldi algorithm [39]

$$E = \sum_i \sum_j \left(L_{ij} - |\,|x_i - x_j|\,|\right)^2 \tag{10.15}$$

where L_{ij} is the actual RTT between nodes i and j. x_i and x_j are the coordinates of the nodes i and j.

10.5.4.2 Maximum Likelihood Topology Maps for Wireless Sensor Networks Using an Automated Robot [25]

Topology maps play a vital role in characterization of a physical network of sensor nodes while maintaining the node connectivity information. It is a nonlinear mapping of a physical network to a topology map. Maximum likelihood-topology maps (ML-TM) is a novel concept that creates a topology map using a packet reception probability function, which is sensitive to the distance between nodes [25]. This method retains the physical shape of the network more accurately than other topology maps in comparison. It supersedes many existing range-based and range-free localization scheme to determine node addresses in terms of cost.

This technique proposes to generate a map of the network with the use of a mobile robot. The robot traverses through network using a defined geometric path

and hence, finding the maximum likelihood-topology coordinates using a binary matrix. While moving through the deployed network, the robot gathers a binary matrix based on the packets received from the nodes from different locations. Using that information, the topology coordinates are calculated by the binary matrix and a packet receiving probability function which is sensitive to the distance. It overcomes the flaws of RSSI algorithms [6] which extract the distances from received power, hence comes across significant errors due to RF communication effects. The proposed topology map preserves the dimensions and shapes of features such as physical voids and network boundaries. It outperforms the RSSI geographical localization and hop-based topology maps.

Maximum-Likelihood Topology Map Algorithm

Unlike other techniques that use hop distances between the nodes for position estimation, ML-TM relies on the set of locations from which packets can be received from a given node. Consider a mobile robot traversing in a network deployed with sensor nodes. The robot can receive packets from the nodes in its vicinity at a given time. The probability of receiving a packet is sensitive to the distance. Thus, the robot keeps track of which nodes it received packets from at different times, and then finds the maximum likelihood position of each node using a binary matrix. There are three crucial considerations of the ML-TM algorithm, which are as follows.

The Packet Receiving Probability Function

As the name suggests, this function describes the probability of receiving packets from a sensor node given that the robot is at a given distance. If $S(d)$ be the probability value when robot is at distance d from the sensor. Then, $S(d)$ satisfies the following constraints: [25]

$$\begin{aligned} 0 \leq S(d) \leq 1 \qquad & \forall d \\ S(d_1) \leq S(d_2) \quad & \forall d_1 \geq d_2 \\ S(d) = 0 \qquad & \forall d > R \end{aligned} \tag{10.16}$$

where R is some distance. The results in the paper are presented using the results,

$$\begin{aligned} S(d) := p_0 \qquad & \forall d \leq r \\ S(d) := 0 \qquad & \forall d \geq R \end{aligned}$$

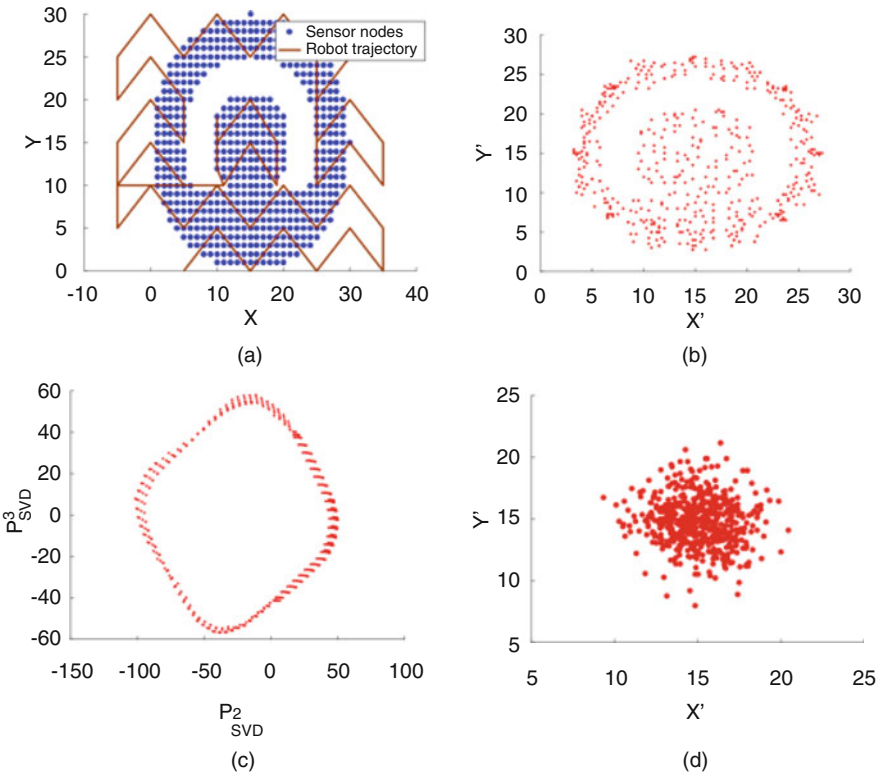

Fig. 10.17 (**a**) Concave void network with 554 nodes. Performance of (**b**) ML-TM, (**c**) singular value decomposition (SVD)-based topology preserving maps (TPM), and (**d**) received signal strength indicator (RSSI)-based map [25]

$$S(d) := \frac{p_0 (R - d)}{(R - r)} \quad \forall r < d < R \tag{10.17}$$

where $0 < p_0 \leq 1$, $0 < r < R \leq R_c$ are some given constants. R_c is the communication range of a sensor node.

This receiving probability function for ML-TM is an intermediate model between RSSI [7] and VC [26]. VC maps a range of values from 0 to R to a unit one hop, which is the major cause for the map being distorted from a physical map. Whereas, RSSI is based on an exact assumed relationship between the signal strength and the distance. This model lacks accuracy due to fading, interference, and noise. It is also quite hard to estimate the parameters for different environmental situations (Fig. 10.17).

Calculation of Topological Coordinates

This calculation involves the information gathered by mobile robot and prediction of the ML topology coordinates. Consider $(x_R(t), y(t))$ to be the coordinates of the mobile robot at a time instant t. The network consists of n stationary sensor nodes labeled $i = 1, 2..., n$, whose locations are not known. We assume that the robot is assumed to able to receive packets at locations $L_1, L_2, \ldots L_N$ on the robot trajectory at discrete time instances, $t_1, t_2..., t_N$. Robot stops at each location L at time t and listens to its local neighborhood nodes and then moves on to the next position in the trajectory. Each node can be heard by the robot from multiple locations. A binary matrix M called as packet receiving matrix of order $n \times N$ is constructed by the following rule:

$M(i, j) = 1$,
if a packet is received from the sensor i at the time tj;
$M(i, j) = 0$,
if a packet is not received from the sensor i at the time tj;

Trajectory of the Mobile Robot

The main objective of the robot is to traverse through the entire network covering maximum number of nodes in least amount of time. The robot could follow a patter for the walk or could be a random path. However, random walk could take longer time to cover the entire network. Hence, a trajectory following a pattern such as square curve, triangle curve, or a sine curve has been tested. With further examination, it has been determined that triangular path provides the best combination of accuracy and time.

Using this robot's trajectory $(x_R(t), y(t))$, the receiving probability function $S(d)$ and the packet receiving matrix, the function $P(x_1, y_1, x_2, y_2, \ldots . ., x_n, y_n)$. The function is the probability to obtain the packet receiving matrix M for the robots trajectory when the sensors 1, 2, $\ldots$, n are located at the points (x_1, y_1), $(x_2, y_2), \ldots, (x_n, y_n)$, respectively. The probability values are obtained using the function $S(d)$ where d is the distance between the robot and the sensor at the time of sending the packet.

The performance of ML-TM is compared in Fig. 10.13 with the SVD-based TPM [26] and RSSI location method based which is a range-based technique on the triangle centroid localization [8]. The comparison clearly shows how ML-TM captures the physical voids and boundaries of the actual physical network competitively. It can also be seen that the RSSI method mapping is less accurate when the environment is noisy and full of obstacles. Also, in SVD-based TPM, the orientation of the network is completely distorted.

Table 10.4 shows the comparison of VCSs using network measurement parameters and topology preserving maps using parameters of Sect. 10.4.

Table 10.4 Comparison table for virtual coordinate systems using network measurement parameters (category C + category D)

Parameters	VCS technique		
	VIVALDI	ML-TM	TPMs
Use of anchors	–	–	✓
Efficient routing	✓	✓	✓
Asserting the local minima	–	–	✓
Ability to deal with node failures and changing topologies	–	✓	✓
Ability to capture the network shape and voids	✓	✓	✓
Use in 3-D networks	✓	–	✓
Distributed computation of VCs	✓	✓	✓
Directionality	–	✓	-
WSN applicability	✓	✓	✓

10.6 Conclusion

A VCS provides an attractive and an economical method to characterize the location of nodes in a network for networking functions such as routing, placement, and topology control. A VCS does not rely on geographical information such as GPS coordinates or distance measurements, and thus can be useful in many harsh and complex environments. This chapter surveyed three categories of VC assignment techniques. They were compared with respect to parameters such as the level of computation involved, the presence of directional information in the resulting coordinates, and the applicability to sensor and IoT networks. There has been significant research to localize nodes in the geographic domain as it is the familiar and obvious choice. However, it is important to note that the performance of geographic coordinate systems for operations such as routing deteriorates in the presence of concave voids. In fact, overcoming local minima in geographic domain purely based on node locations is highly ineffective with 3D-volume and 3D-surface networks of complex shapes. Such networks can be expected to be very common with emerging IoT applications. Connectivity-based VCSs have shown to be much more effective in such cases compared to geographical coordinates. While many of the VCSs are based on connectivity information, others rely on parameters such as packet loss and path delay to determine coordinates.

References

1. T.H. Illangasekare, Q. Han, A.P. Jayasumana, in *Environmental Underground Sensing and Monitoring*, eds. By S. Pamukcu and L. Cheng. Underground Sensing: Monitoring and Hazard Detection for Environment and Infrastructure (Academic, London, 2018), pp. 203–246.
2. P. Bose, P. Morin, I. Stojmenović, J. Urrutia, Routing with guaranteed delivery in ad-hoc wireless networks. Wirel. Netw **7**(6), 609–616 (2001)

3. Q. Cao, T. Abdelzaher, Scalable logical coordinates framework for routing in wireless sensor networks. ACM Trans. Sens. Netw. **2**, 557–593 (2006)
4. D. Johnson, D. Maltz, J. Broch, *The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks* (Ad-Hoc Networking/Addison-Wesley Longman Publishing Co., Inc., Boston, 2001)
5. C. Perkins, E. Royer, Ad-hoc On-Demand Distance Vector Routing, in *Proceedings of the Second IEEE Workshop on Mobile Computer Systems and Applications*, 25–26 February 1999, p. 90
6. P. Pathirana, N. Bulusu, A. Savkin, S. Jha, Node Localization using mobile robots in delay-tolerant sensor networks. IEEE Trans. Mob. Comput. **4**, 285–296 (2005)
7. D. Turner, S. Savage, A.C. Snoeren, On the Empirical Performance of Self-Calibrating Wi-Fi Location Systems, in *Proceedings of the 2011 IEEE 36th Conference on Local Computer Networks, LCN '11, (Washington, DC, USA)*, IEEE Computer Society, 2011, pp. 76–84
8. B. Mukhopadhyay, S. Sarangi, S. Kar, Novel Rssi Evaluation Models for Accurate Indoor Localization with Sensor Networks, in *2014 Twentieth National Conference* on *Communications (NCC)*, February 2014, pp. 1–6
9. R. Jin, H. Wang, B. Peng, N. Ge, Research on RSSI-Based Localization in Wireless Sensor Networks, in *2008 4th International Conference on Wireless Communications, Networking and Mobile Computing*, Dalian, 2008, pp. 1–4
10. B. Karp, H.T. Kung, Greedy Perimeter Stateless Routing (GPSR) for Wireless Networks, in *Proc. 6th Annual ACM/IEEE Int. Conf. on Mobile Computing and Networking (Mobicom)*, 2000, pp. 243–254
11. C. Finn, D. Williams, An aeromagnetic study of Mount St. Helens. J. Geophys. Res. **92**, 10194–110206 (1987)
12. I. Bose et al., Assembly of Scaffold-mediated complexes containing Cdc42p, the exchange factor Cdc24p, and the effector Cla4p required for cell cycle-regulated phosphorylation of Cdc24p. J. Biol. Chem. **276**(10), 7176–7186 (2001)
13. D. Niculescu, B. Nath, DV based positioning in ad-hoc networks. J. Telecommun. Syst. **22**, 267–280 (2003)
14. E.J. Kuhn, M.M. Viering, K.M. Rhodes, P.K. Geyer, A test of insulator interactions in Drosophila. EMBO J. **22**(10), 2463–2471 (2003)
15. A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, I. Stoica, Geographic Routing without Location Information, in *Proc. 9th Int. Conf. on Mobile Computing and Networking*, 2003, pp. 96–108
16. H. Frey, S. Rührup, I. Stojmenović, Routing in Wireless Sensor Networks, in *Guide to Wireless Sensor Networks*, (Springer, Berlin, 2009)
17. J.N. Al-Karaki, A.E. Kamal, Routing techniques in wireless sensor networks: a survey. Wireless Commun. IEEE **11**(6), 6–28 (2004)
18. I. Aumndson, X.D. Koutsoukos, A Survey on Localization for Mobile Wireless Sensor Networks, in *Workshops on Mobile Entity Localization and Tracking (MELT) Springer Lecture Notes in Computer Science (LNCS5801),* Orlando, Florida, 2009, pp. 235–254
19. A.P. Jayasumana, Q. Han, T. Illangasekare, Virtual sensor networks-A Resource Efficient Approach for Concurrent Applications, in *Proceedings of the 4th International Conference on Information Technology: New Generations (ITNG 2007)*, April 2007, pp. 111–115
20. R. Flury, R.R. Wattenhofer, Randomized 3D Geographic Routing, in *Infocom 2008. Proc. 27th Conference on Computer Communications*. IEEE, April 2008, pp. 13–18
21. T.R. Babu, A. Chatterjee, S. Khandeparker, A.V. Subhash, S. Gupta, Geographical Address Classification without using Geolocation Coordinates, in *Proceedings of the 9th Workshop on Geographic Information Retrieval*, ACM, 2015, p. 8
22. D. Dhanapala, A.P. Jayasumana, Anchor Selection and Topology Preserving Maps in WSNs – A Directional Virtual Coordinate Based Approach, in *2011 IEEE 36th Conference on Local Computer Networks (LCN)*, October 2011, pp. 571–579

23. D.C. Dhanapala, A.P. Jayasumana, Directional Virtual Coordinate Systems for Wireless Sensor Networks, in *Proceedings of the IEEE International Conference on Communications (ICC-11)*, 2011, p. 16
24. B. Donnet, B. Gueye, M.A. Kaafar, A survey on network coordinates systems, design, and security. IEEE Commun. Surv. Tutorials **12**(4), 488–503 (2010)
25. A. Gunathillake, A.V. Savkin, A.P. Jayasumana, Maximum Likelihood Topology Maps for Wireless Sensor Networks Using an Automated Robot, *2016 IEEE 41st Conference on Local Computer Networks (LCN)*, Dubai, 2016, pp. 339–347
26. D.C. Dhanapala, A.P. Jayasumana, Topology preserving maps: extracting layout maps of wireless sensor networks from virtual coordinates. IEEE/ACM Trans. Networking **22**(3), 784–797 (2014)
27. J. Dong, K. Ackermann, B. Bavar, C. Nita-Rotaru, Secure and robust virtual coordinate system in wireless sensor networks. J. ACM Trans. Sens. Netw. **6**(4), 29 (2010)
28. J. Seibert, S. Becker, C. Nita-Rotaru, R. State, Newton: securing virtual coordinates by enforcing physical laws. IEEE/ACM Trans. Networking **22**(3), 798–811 (2014)
29. D. Zage, C. Nita-Rotaru, Robust decentralized virtual coordinate systems in adversarial environments. ACM Trans. Inf. Syst. Secur. **13**(4), 38 (2010)
30. S. Beckery, J. Seibert, D. Zage, C. Nita-Rotaru, R. Statey, Applying Game Theory to Analyze Attacks and Defenses in Virtual Coordinate Systems, in *2011 IEEE/IFIP 41st International Conference on Dependable Systems & Networks (DSN)*, Hong Kong, 2011, pp. 133–144
31. A. Kermarrec, A. Mostefaoui, M. Raynal, G. Tredan, A. Carneiro Viana, Large-Scale Networked Systems: From Anarchy to Geometric Self-Structuring, in *10th Intnl. Conf. Distrib. Comput. Netw. (ICDCN), volume 5408 of Lect. Notes Computing. Sc.*, Hyderabad, India, January 2009. Springer, p. 25–36
32. M. Shah, A. Sardana, Searching in Internet of Things using VCS, in *Proceedings of the First International Conference on Security of Internet of Things*, Kollam, India, 17–19 August 2012
33. M. Li, P. Jia, Y. Xu, Y. Yuan, Traveling Path Tracking Algorithm in Virtual Coordinate System for Intelligent Vehicle, in *2012 IEEE 2nd International Conference on Cloud Computing and Intelligence Systems*, Hangzhou, 2012, pp. 1183–1187
34. P. Leone, K. Samarasinghe, Greedy Routing on Virtual Raw Anchor Coordinate (VRAC) System, in *2016 International Conference on Distributed Computing in Sensor Systems (DCOSS)*, Washington, DC, 2016, pp. 52–58
35. A.P. Jayasumana, R. Paffenroth, S. Ramasamy, Topology Maps and Distance-Free Localization from Partial Virtual Coordinates for IoT Networks, in *Proceedings of the IEEE ICC,* May 2016, pp. 1–6
36. J.-P. Sheu, M.-L. Ding, K.-Y. Hsieh. 2007, Routing with Hexagonal Virtual Coordinates in Wireless Sensor Networks, in *Proceedings of the IEEE Wireless Communications and Networking Conference (WCNC'07)*. pp. 2929–2934
37. Q. Fang, J. Gao, L.J. Guibas, V.D. Silva, L. Zhang, GLIDER: gradient landmark-based distributed routing for sensor networks. IEEE Infocom **1**, 339–350 (2005)
38. J. Bruck, J. Gao, A. Jiang, MAP: Medial Axis Based Geometric Routing in Sensor Networks, in *IEEE/ACM Mobicom*, 2005, pp. 88–102
39. J. Newsome, D. Song, GEM: Graph Embedding for Routing and Data-Centric Storage in Sensor Networks Without Geographic Information, in *ACM Conference on Embedded Networked Sensor Systems*, 2003
40. A. Cvetkovski, M. Crovella, Hyperbolic Embedding and Routing for Dynamic Graphs, in *Infocom*, 2009, pp. 1647–1655.
41. M.J.Tsai, H.Y.Yang, W. Huang, Axis-Based Virtual Coordinate Assignment Protocol and Delivery-Guaranteed Routing Protocol in Wireless Sensor Networks, in *IEEE Infocom'07*
42. F. Dabek, R. Cox, F. Kaashoek, R. Morris, Vivaldi: A Decentralized Network Coordinate System, in *Sigcom*, Portland, OR, August 2004
43. L. Wei Lehman, S. Lerman, A Decentralized Network Coordinate System for Robust Internet Distance, in *Proceedings of the ITNG*, 2006

44. T. Ng, H. Zhang, A Network Positioning System for the Internet, in *Proceedings of the USENIX*, 2004
45. E. Ng, H. Zhang, Predicting Internet Network Distance with Coordinates-Based Approaches, in *Proceedings of the Infocom*, 2002
46. P. Francis, S. Jamin, C. Jin, Y. Jin, D. Raz, Y. Shavitt, L. Zhang, Idmaps: A Global Internet Host Distance Estimation Service, 2000
47. L. Tang, M. Crovella, Virtual Landmarks for the Internet, in *Proceedings of the Sigcomm*, 2003
48. M. Pias, J. Crowcroft, S. Wilbur, S. Bhatti, T. Harris, Lighthouses for Scalable Distributed Location, in *Proceedings of the IPTPS*, 2003
49. M. Costa, M. Castro, R. Rowstron, P. Key, PIC: Practical Internet Coordinates for Distance Estimation, in *Proceedings of the ICDCS*, 2004
50. L. Guibas, C. Holleman, L.E. Kavraki, A Probabilistic Roadmap Planner for Flexible Objects with a Work Space Medial-Axis Based Sampling Approach, in *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, Kyongju, Korea, 1999, IEEE Press, pp. 254–260
51. N. Amenta, M. Bern, D. Eppstein, The crust and the β-skeleton: combinatorial curve reconstruction. Graphical Models Image Process. **60**, 125–135 (1998)
52. N. Amenta, S. Choi, R.K. Kolluri, The power crust, unions of balls, and the medial axis transform. Comput. Geom. Theory Appl. **19**, 127–153 (2001)
53. H. Blum, in *A Transformation for Extracting New Descriptors of Shape*, ed. By W. Wathen-Dunn. Models for the Perception of Speech and Visual Form (MIT Press, Cambridge, 1967), pp. 362–380.
54. H.I. Choi, S.W. Choi, H.P. Moon, Mathematical theory of medial axis transform. Pac. J. Math. **181**(1), 57–88 (1997)
55. D.C. Dhanapala, A.P. Jayasumana, CSR: Convex Subspace Routing Protocol for WSNs, in *Proc. 34th IEEE Conf. on Local Computer Networks*, October 2009
56. G. Mahindre, A.P. Jayasumana, Post Failure Recovery of Virtual Coordinates in Wireless Sensor Networks, in *Proc. 7th International Conference on Information and Automation for Sustainability (CIAfS'14)*, Colombo, Sri Lanka, December 2014
57. P. Cheng, T. Han, X. Zhang, R. Zheng, Z. Lin, A Single Mobile Anchor Based Distributed Localization Scheme for Sensor Networks, in *2016 35th Chinese Control Conference (CCC)*, Chengdu, 2016, pp. 8026–8031
58. M. Kirby, *Geometric Data Analysis—An Empirical Approach to Dimensionality Reduction and the Study of Patterns* (Wiley, New York, 2001)
59. J. Li, J. Jannotti, D. DeCouto, D. Karger, R. Morris, A Scalable Location Service for Geographic Ad-Hoc Routing, in *IEEE/ACM Mobicomm*, 2000, pp. 120–130
60. K.L. Calvert, M.B. Doar, E.W. Zegura, Modeling internet topology. IEEE Commun. **35**(6), 160–163 (1997)
61. K. Liu, N. Abu-Ghazaleh, Stateless and Guaranteed Geometric Routing on Virtual Coordinate Systems, in *Proceedings of the 5th IEEE Int. Conf. on Mobile Ad Hoc and Sensor Systems (MASS)2008*, September 2008, pp. 340–346

Chapter 11
Small Data in IoT: An MCS Perspective

Sherif B. Azmy, Ruslan Abu Sneineh, Nizar Zorba, and Hossam S. Hassanein

11.1 Introduction

With the advent of the new millennium, a new revolution loomed over the horizon. Two new technologies have contributed to the Information Age as we now know it. Third-generation (3G) wireless technology came hand in hand with Internet-enabled "smart" devices which had communication and processing capabilities that allowed mobile Internet on demand, en masse. With data rates going up to 200 kbps, significant sizes of data were transferable over the network for the first time. Having a mobile phone with Internet connectivity allowed the intercommunication of numerous devices. A while later, with the advent of *smartphones*, as we know them today, researchers and tech companies realized the opportunities provided by connecting devices to smartphones via the Internet, and thus the concept of Internet of Things (IoT) was born.

Nowadays, IoT is growing more integrated in our lives at an unprecedented rate. Setting aside obvious examples such as smartphones, smart devices such as smart TVs, smart cars, smart sensors, smart air conditioners, everything can be connected to the Internet. In many cases, the smartphone is often the enabling medium that connects users, i.e., humans, with "things," i.e., interconnected devices via the Internet. This marks the beginning of a technological revolution that is capable of transforming people's lives, becoming a pillar in the smart city of tomorrow. As of now, devices are smart enough to allow users to schedule their air conditioners' operation, maintaining a comfortable atmosphere that is achieved efficiently without

S. B. Azmy (✉) · R. Abu Sneineh · N. Zorba
College of Engineering, Qatar University, Doha, Qatar
e-mail: sherif.azmy@qu.edu.qa; ruslan.abusneineh@qu.edu.qa; nizarz@qu.edu.qa

H. S. Hassanein
School of Computing, Queen's University, Kingston, ON, Canada
e-mail: hossam@cs.queensu.ca

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7_11

wasting energy. Modern cars come with embedded operating systems which can connect to the Internet to keep their firmwares up-to-date, providing the user with real-time data—on their phone—about their vehicle's condition. It is even possible for a user to turn any ordinary TV into a smart TV via devices as small as Google's Chromecast.

Nowadays, the proliferation of IoT is most noticeable in urban environments, particularly cities. Some cities have taken the initiative to increase the prevalence of IoT, while others are gradually progressing towards the same end via the citizens' natural adoption of technology through their uses. In both cases, these "smart cities" are characterized by the extensive presence of IoT devices and IoT-enabled technologies.

Smart cities are characterized by the fast flow of information, continuous generation of data, and real-time efficient management. To deal with the huge number of sensed values in smart cities, the paradigm of mobile crowdsensing (MCS) has emerged, with device-to-device (D2D) or evolved communication involving humans. These networking combinations are based on the IoT and its enabling infrastructure. For example, detecting the number of vehicles on a road segment allows the smart city to send recommendations to drivers to reroute for faster navigation and less congestion. Furthermore, having a system that monitors the overall traffic distribution in real-time, traffic signals can operate in a dynamic manner to allow optimal car passage from all directions, rendering the infrastructure itself as *smart* as it tries to achieve efficient smooth transportation. With the advent of self-driving cars, such information can be utilized without any need to involve the car driver. This technology can also be utilized in order to help drivers find parking spots in public areas.

Various applications are enabled by IoT in one way or another, making IoT an integral component of smart cities. Nevertheless, in certain cases there could be certain challenges. With the extreme prevalence of mobile phones and other IoT devices, the pressure on the network could be huge, which necessitates the development of a proper infrastructure to handle such *huge* amounts of data. On the other hand, data needs to be mined for different purposes, but it might not be always available, or the amount of data present could be very small. This leads us to another aspect of IoT, relating to the scale of data availability.

11.1.1 Small Data

MCS is an interesting IoT framework. With the unprecedented increase in the availability of smartphones and their computational power, quality, cost, and quantity of sensors implemented, MCS has emerged. MCS systems exploit the ubiquity of smartphones in the crowd of users to collect data and compile inferences about the environment of the users via the readings of their smartphones' sensors. An example of an MCS application is the Waze service which provides real-time information about traffic congestion and road conditions by *opportunistically* collecting the data

needed when the user is at a specific location, or *participatorily* by having users submit reports about the road conditions. MCS, as a paradigm, is relatively new, as it has appeared during the early 2010s [1]. However, there were some precedents to its birth such as Google Traffic which utilized GPS data from GPS-equipped phones from users as early as 2007 [2]. In an MCS system, smartphone users would give permission for MCS client apps on their phones to collect sensor data, agreeing to participate in sensing tasks, which is performed in exchange for an incentive payment or a service. Nevertheless, the amount of users available in a location may not be sufficient, which results in situations where the data is scarce, limited, or sparsely distributed. In such a case, the MCS system should be capable of making the most given the least by using techniques that work with *small data sets*. Also, from an economical point of view, the collection of data usually involves the use of resources (or payments), especially in *participatory MCS*. It is desired for the MCS system to decrease the required sensing, thus optimizing the system's costs, with the constraint of maintaining sensing quality. To characterize the least number of measurements (or sensors) would be a great benefit for operators and decision makers.

The usage of the term "small data" in recent literature has two interpretations: the first relates to small data packets in data transmission [3, 4], while the second deals with small data sets and small-sample sizes. Throughout this book chapter, we always refer to the latter definition of "small data." Small data sets refer to situations in which data is of a niche small scale causing difficulty in discerning the truth of a reading. Small data may not be a problem when dealing with an established ad-hoc network of IoT devices, or wireless sensor networks (WSNs) [5], meant for a specific purpose such as measuring the temperature, noise levels, or air pollution around the city. It is often the case that WSN devices are reliable enough to provide an estimation of the true value of a physical quantity. However, the situation is different when the sensors are the users' smartphones, as is the case in MCS, where the human users—MCS participants—can be mobile, heterogeneous, scarce, or unreliable. They could even be malicious with the purpose of throwing off the system, hindering its operation, or exploiting it for personal gains. In such a case, the amount of *small data* might not safely satisfy the criteria for proper high-level data collection.

This book chapter aims to discuss scenarios of MCS which have a significant scarcity of data, and introduce the problem of *Small Data* in IoT. To that end, the chapter is designed as follows: Sect. 11.2 provides an overview of MCS, its architectures, frameworks developed as well as the data scale of MCS; Sect. 11.3 tackles the necessary mathematical foundations intended for Sect. 11.4 which develops a method to quantify the quality of samples in MCS; and finally Sect. 11.6 concludes with a discussion of the overlap between IoT, MCS, and small data and sheds light on the possible future research directions.

11.2 Literature Review and Related Work

Smartphones have dominated the market over the past few years, where these omnipresent smartphones have not only Internet access but are also equipped with sophisticated sensors. Other than their microphones, smartphones come also with high-sensitivity gyroscopes, GPS receivers, light sensors, etc. Researchers have realized the potential lying within *smartphone ubiquity*, and developed what is now called in the academic literature the MCS. The core of MCS systems revolves around their architecture, which is a general design of how individual blocks interact. Generally, the architecture of MCS systems has the following main blocks:

- **MCS Administrator:** or the Task Publisher, refers to an entity that decides and designs the MCS task and commissions it. Such an entity could be a city municipality or a smart city administrator aiming to collect data about a specific quantity for a specific purpose, e.g., noise levels on streets.
- **MCS Participants:** refer to members of the crowd, from the general public, who own smartphones and mobile devices with sensing capabilities. Crowd members who choose to participate in an MCS system, often called *workers* or *participants*, receive a task and decide whether to execute it or not, whether in a *participatory* manner, i.e., actively involving the user in its execution, or in an *opportunistic* manner, passively being executed when the user satisfies spatiotemporal conditions. Participants can be offered incentives, or rewarded, by the MCS system for successful task execution.
- **MCS System:** refers to the server that MCS participants connect to, and to whom the tasks designed by the MCS Administrator are assigned in an automatic manner. It is also where the participants' performance is evaluated and rewarded, and where data is collected.

Figure 11.1 shows a simple diagram of how MCS systems generally are.

Various research efforts have led to the development of many topics relating to MCS. Some of which aim to tackle technical problems relating to data transmission [6], data processing [7], optimizing computational costs [8], and power usage [8], while others focus on a more managerial aspect of MCS, such as optimal task allocation [9], user quality assessment and trustworthiness [10], quality of information [11], and even privacy concerns [12] and workers' incentivization methods [13].

Data quality is one of the main research aspects in MCS applications. Ultimately, the quality of an MCS system depends on the reliability of the workers in providing data that serves the purpose designated by the MCS Administrator. Ideally, the MCS participants[1] are expected to submit reliable and truthful data. However, some users may unintentionally submit corrupt or incorrect data due to an error in the sensing process. For example, if the device is supposed to sense noise levels on streets,

[1]From here on, we shall refer to MCS participants and their devices interchangeably as "participants," or "users."

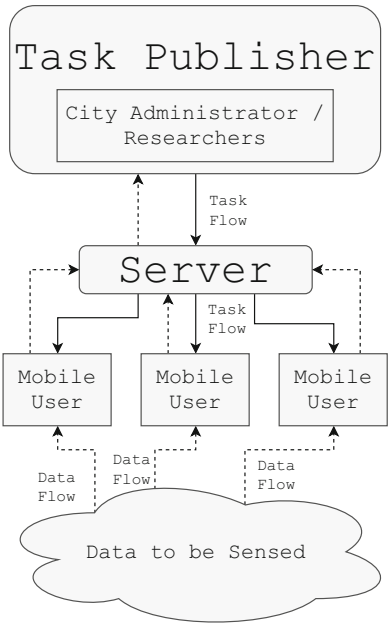

Fig. 11.1 Block diagram of general mobile crowdsensing (MCS) systems

but the user carelessly has the device in their pocket, the noise levels reported will be significantly attenuated in comparison to what they are in reality. On the other hand, some users may *maliciously* submit incorrect data in order to hinder the MCS system, or exploit it for financial gains by submitting fake data. Authors of [14] deal with the problem of erroneous data by utilizing a *dynamic Bayesian network* to correlate data spatiotemporally in order to remove noise and also filter untrustworthy data. Another approach to this problem is also employed by Luo and Zeynalvand [15] who introduced a cross-validation technique to utilize *the wisdom of the crowds* by means of a lightweight plugin that aims to have the data validated by referring to other measurements, public data, and social networks.

Another research problem in MCS relates to the privacy, security, and anonymity. MCS systems require access to mobile devices' sensors, some of which may reveal sensitive information about the users (especially information about user's location). Many users are concerned that their data maybe leaked, or exploited for malicious purposes. To solve this problem, an algorithm based on public key infrastructure as well as a combined public key was employed to guarantee anonymous authentication of users [16]. Another approach, discussed in [17], contributes a mechanism for protecting users from maliciously acting servers, such as a server being hacked. It also explores incentive schemes based on the SPEAR peer-to-peer security architecture [18].

As users are generally reluctant to sacrifice their resources (e.g., cellular data, processing power, battery, etc.), and risk their private information in exchange of sharing sensed data, incentive mechanisms are developed to motivate and incentivize user participation in MCS systems. Incentive schemes are one of the

most widely researched topics in MCS. However, most incentive schemes funnel down into auction bids in one way or another, with some introducing game theory mechanics into incentivizing methods. One such scheme is developed in [19], in which not only the contributor of the data is rewarded, but also those who refer (or delegate) tasks to other workers—who in turn successfully execute it—are also rewarded. Thus, the *parent* reaps the rewards for any task performed by the *child (or children)*. The further down the line the referral within the tree is, the less reward the parent gets. The mechanism also deals with the idea of "Sybil-Attack," a scenario in which a malicious worker fakes an identity and refers to themselves in order to accumulate these extra rewards. Meanwhile, an approximation algorithm is employed in [20] based on a greedy approach which iteratively selects auction bid winners. The proposed scheme introduces the concept of a "social cost" that is employed as a constraint for the greedy approach, operating in an online (i.e., real-time), location-aware manner, to optimize the assignment of MCS tasks to workers.

A major subfield of MCS systems is that of task allocation. Tasks can be allocated to participants in more than a way, some of which pay attention to situational details such as the user's location, battery level, and available sensors, while others employ historical details, such as trust and reputation levels in the selection process to guarantee the quality of data. In [11], a two-level iterative algorithm is proposed for estimating the quality of data contributed by a user. The estimated data quality is taken into consideration when deciding whether to assign the tasks to the user in question or not. Meanwhile, in [21] a novel approach that treats the mobile device network as a social network is proposed, in which mobile devices are modeled as virtual social objects. The framework designed in [21] addresses scalability issues to find devices that can partake in an MCS task.

MCS has found for itself endless uses, some of which are more refined than others, limited only by our current mathematical understanding and technological advancement. In [22], for example, authors suggest a unique solution for the simple problem of assigning users, or friends in this case, to pass by a store and then by the task publisher. The algorithm developed by them allows prediction of the users' path and recommends which users are most fit for the task. In [23], an MCS framework is proposed to help users navigate during extreme conditions, such as emergencies, crises, or extreme congestion. The authors have developed an application that guides the users through the least crowded areas, avoiding the panic-driven frantic flow of masses. The authors of [24] propose a framework to measure air pollution using smartphone cameras, by means of collecting large amounts of image samples and analyzing the scatter of light. The algorithm developed allows reliable estimation of the Particle Mass 2.5 (PM2.5) concentration in the air. In [25], an MCS framework is designed to predict communities and their activities by modeling them as tensors. Its framework analyzes individuals and categorizes them as groups based on physical and virtual information, allowing the prediction of possible group activities. Furthermore, they also reduce computational load by treating these groups rather than individuals. The framework facilitates other applications, such as the design of smart grids or customized restaurant recommendation services. Authors of [26] suggest a way of detecting road anomalies. The method developed utilizes accelerometer and

gyroscope data collected from sensors present in mobile devices, such as tablets and vehicle-embedded sensors. Machine learning is then used to recognize and classify the different types of road anomalies based on the data collected.

11.3 Mobile Crowdsensing Model and Preliminary Mathematics

In MCS, an area needs to be divided by the MCS administrator into areas of interest, or *geofences*. This is a division in space, but it also needs to be divided in time to acquire the variations in the signal over time by assigning a sampling frequency (and thus a sampling time period). These divisions in space and time can be represented by means of a *spatiotemporal diagram* [27]. Figure 11.2 illustrates a spatiotemporal diagram, in which the area of interest is divided into spatiotemporal cells. The MCS administrator can divide the space in a way that satisfies the objective of the system, or in a manner that is consistent with the temporal and spatial versions of the Sampling Theorem [28, 29]. MCS participants are distributed over the spatiotemporal by the MCS system according to their availability. For example, if the MCS administrator requires temperature values to be sensed by the MCS participants, a set of sensed values, X_m, is to be obtained:

$$X_m = \{x_{m,1}, x_{m,2}, \ldots, x_{m,N}\} \tag{11.1}$$

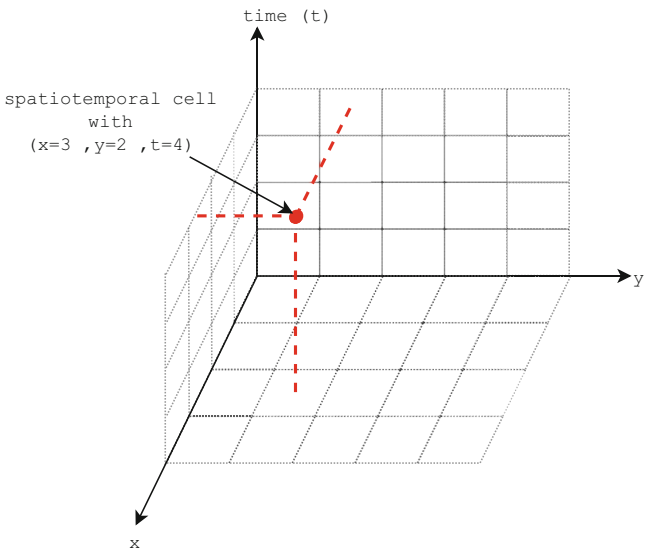

Fig. 11.2 Spatiotemporal cells. Partition is illustrated as a square grid for simplicity. Any spatial partition, such as the Voronoi partition, can be employed

where each MCS participant's reading would be modeled as a random variable $x_{m,i}$, with N being the number of participants present in the mth spatiotemporal cell that has the 3-tuple (α, β, γ), where $\alpha, \beta, \gamma \in \mathbb{N}$.

The true value of the sensed quantity, μ, can be estimated from the *sample* described in Eq. (11.1) by finding the mean:

$$\hat{\mu}_m = \bar{X}_m = \text{mean}(X) = \frac{1}{N}\sum_{i=1}^{N} x_{m,i} \tag{11.2}$$

and its standard deviation, $\hat{\sigma}_m$, can also be estimated:

$$\hat{\sigma}_m = \sqrt{\frac{\sum_{i=1}^{N}(x_{m,i} - \hat{\mu}_m)^2}{N-1}} \tag{11.3}$$

This is the case when the sample is large enough that the population can be represented by $N \geq 30$. At such a size, the *central limit theorem* becomes viable to use as each of the $x_{m,i}$ random variables would ultimately contribute to a normal distribution of the values [30]. This assumption is very useful.

However, what about the case when $N \ll 30$? That is, when there are barely enough measurements within a spatiotemporal cell. The problem then becomes of the *small data* scale, where one would have to infer as much as possible under the stringent sample size. When the data scale is of the *small data* scale, Eqs. (11.2) and (11.3) are less accurate to estimate the true value, μ, as any outlier, $x_{m,o}$, would throw off the estimation completely. For example, if there were N_o outliers in X_m, we would have:

$$\begin{aligned}
\hat{\mu}_m &= \frac{1}{N}\sum_{i=1}^{N} x_i \\
&= \frac{1}{N}\Bigg[\underbrace{\sum_{i=1}^{N-N_o} x_{m,i}}_{\text{non-outliers}} + \underbrace{\sum_{j=N-N_o}^{N} xj}_{\text{outliers}}\Bigg] \\
&= \underbrace{\frac{N-N_o}{N}\hat{\mu}_{\bar{X}_m}}_{\text{non-outlier effect}} + \underbrace{\frac{N_o}{N}\hat{\mu}_{\bar{X}_{m,o}}}_{\text{outlier effect}} \\
&= \underbrace{\frac{N-N_o}{N}(\mu_m + \xi_{\bar{X}_m})}_{\text{correct estimation in } \hat{\mu}_m} + \underbrace{\frac{N_o}{N}(\mu_m + \xi_{\bar{X}_{m,o}})}_{\text{incorrect estimation in } \hat{\mu}_m}
\end{aligned} \tag{11.4}$$

By considering the outlier value, $x_{m,o}$, as an offset from the true value, μ, we can express $x_{m,o} = \mu_m + \Theta_m\sigma_m$ where Θ_m is the number of standard deviations the

outlier value is far from the truth. This value is called the *Outlier Deviation Factor*. Using this knowledge, the outlier error, $\xi_{\bar{X}_{m,o}}$, can be written as $\xi_{\bar{X}_{m,o}} = \bar{\Theta}_m \sigma_m$, where $\bar{\Theta}_m$ is the *Average Outlier Deviation Factor* (AODF) for N_o outliers present in X_m. Equation (11.4) can be rewritten to show how the sample size N impacts the outlier error:

$$\hat{\mu}_m - \frac{N - N_o}{N}(\mu_m + \xi_{\bar{X}_m}) = \frac{N_o}{N}(\mu_m + \xi_{\bar{X}_{m,o}}) \tag{11.5}$$

It is evident from Eq. (11.5) that the left side should be zero. However, for that to happen, the right side must also be zero. This is only possible in one of two cases:

a. $N_o/N \to 0$, that is, the proportion of the outliers in the sample is small or,
b. $N \to \infty$, that is, the sample size becomes very large.

Case "b" is when the data is of the large scale, i.e., $N \geq 30$. However, case "a" is trickier, as the proportion N_o/N cannot easily become zero, as N_o is always an integer that is greater than 1, i.e., $N_o \in \mathbb{N}$.

In this section, we describe methods to deal with the problems encountered in case "a," which we refer to as *Small Data Problems*.

11.3.1 Robust Statistics

Robustness is an important notion when dealing with small data. As discussed earlier in this section, the presence of an outlier becomes significant. Robust statistics is the branch of statistics that has to deal with the presence of outlier. There are robust measures of centrality and spread, perhaps the most famous of them is the sample median, which is the *most robust* measure of centrality that has a *breakdown point* of 0.5. The breakdown point is a property of a statistic that indicates at what proportion of outliers, N_o/N, will the statistic fail to differentiate between outlier and non-outlier samples. For a statistic θ, the breakdown point b_θ is generally between 0 and 0.5. That is, because a $b_\theta = 0$ implies that the statistic is vulnerable to outliers, that their presence will certainly affect θ, while $b_\theta = 0.5$ implies that θ will not be thrown off unless 50% of the sample is outliers, a proportion at which the outliers represent a part of the sample itself. We shall describe to the reader two robust measures of centrality, and a robust measure of dispersion. Both of which are very relevant to the small data problem within the scope of MCS problems.

11.3.1.1 MAD-Mean: Median Absolute Deviation Filtered Mean

An alternative measure of dispersion is the median absolute deviation (MAD) that has the robust breakdown point of the median, 0.5 [31]. The MAD is defined as the median of the absolute deviations from the sample median, stated as:

$$\mathrm{MAD}_m = \mathrm{median}\Big\{\big|x_{m,i} - \overbrace{\mathrm{median}(X_m)}^{\tilde{x}_{m,M}}\big|\Big\} \tag{11.6}$$

where $x_{m,i}$ is the ith sample in X_m, and the median is denoted $\tilde{x}_{m,M}$. It is a robust measure of dispersion that is based on *deviations from the median*, rather than deviations from the mean as is the case in the standard deviation. The MAD, however, is a *consistent estimator* of the standard deviation with:

$$\hat{\sigma}_{\mathrm{MAD}_m} = \frac{1}{\Phi^{-1}(3/4)}\mathrm{MAD}_m = \frac{1}{1.4826}\mathrm{MAD}_m\Big|_{f(x)=N(\mu,\sigma)} \tag{11.7}$$

where $\hat{\sigma}_{\mathrm{MAD}_m}$ is the MAD-based standard deviation estimator, $f(x)$ is the probability distribution followed by the population, $N(\mu, \sigma)$ represents the special case of the normal distribution centered around μ with a standard deviation of σ, Φ^{-1} is the quantile function (which is the inverse of the cumulative distribution function $F(x)$). The quantile function is evaluated at the probability of 75%, as the MAD is defined to cover the *median distance from the sample median* ($\tilde{x}_{m,M}$, which—for a symmetric distribution—covers 50% from the left to the right of the median $\tilde{x}_{m,M}$), and $\Phi^{-1} = 1.4826$ for any normal distribution $N(\mu, \sigma)$. We note that through the normal distribution, the general relation between the MAD and the standard deviation for any analytically defined distribution, $f(x)$, was derived [31].

The MAD inherits the robustness of the median, which makes it useful for the detection and removal of outliers. However, the definition of outliers is vague and depends on the application in hand. For MCS systems, sensor measurements tend to follow the normal distribution, which involves *normal outliers* that rise from extremes of the distribution. However, there are *abnormal outliers* which lie far outside the three-sigma range defined by the three-sigma rule [32]. These abnormal outliers rather come from an *unexpected phenomenon* that is not modeled by the normal distribution.

In an MCS system, these abnormal outliers need to be isolated as they represent a hindrance for the purpose of the MCS system, to estimate the truth at a specific time in a specific location. The MAD allows the removal of outlier samples present in X_m, producing an MAD-trimmed sample, denoted X_{MAD_m} [33], where *only values from within the range of the normal distribution are considered*, expressed as:

$$X_{\mathrm{MAD}_m} = \{X_m : x_{m,j} \in [\tilde{x}_{m,M} \pm \underbrace{\lambda \mathrm{MAD}_m}_{\delta\hat{\sigma}_{\mathrm{MAD}_m}}]\} \tag{11.8}$$

where λ is how many MADs away from the median is the threshold beyond which samples are outliers. The product $\lambda \mathrm{MAD}_m$ can be related to the consistent estimation of the standard deviation $\delta\hat{\sigma}_{\mathrm{MAD}_m}$ to define it in terms of multiples (δ) of deviations from the mean ($\hat{\sigma}_{\mathrm{MAD}_m}$).

Nevertheless, normal outliers are *expected*, unlike *abnormal outliers*, which lie farther beyond the three-sigma range. Thus, to ensure proper estimation of the true value, μ_m, samples beyond the 3-σ range could be considered as non-outliers, around $\lambda = 4$, and therefore only the values belonging to the interval $\tilde{x}_{m,M} \pm 4\text{MAD}_m$ will be accepted. Therefore, the MAD-trimmed robust estimate of μ can be defined as:

$$\bar{X}_{\text{MAD}_m} = \frac{1}{N - N_o} \sum_{i=1}^{N-N_o} x_{\text{MAD}_{m,i}} \tag{11.9}$$

where N is the sample size, N_o is the number of outliers present (i.e., samples outside the $\tilde{x}_{m,M} \pm \lambda\text{MAD}_m$ range), and x_{MAD_i} is the ith element in the MAD-trimmed sample, X_{MAD_m}.

Notice that the value of λ provides a degree of freedom for the system administrator to decide the range of measurements and the consideration of outliers. For the quality metric, λ takes values less than 2 for the purpose of maximizing robustness due to the small-sample sizes. The MAD-based outlier detection is a technique that is especially useful for small-sample sizes [34], which makes it useful for the small-sample scenarios present in MCS systems' spatiotemporal cells. Algorithm 1 summarizes the procedure to obtain the MAD-mean.

Algorithm 1 Computing the MAD-mean

Input: A sample: $X = \{x_1, x_2, ..., x_n\}$
Output: MAD-mean: $\bar{x}_{\text{MAD}}$
Initialize : λ
MAD(X) = median(median$(X) - X_i$)
for all x_i **do**
  if $x_i \notin$ [median$(X) \pm \lambda$MAD] **then**
    $X_o = \texttt{append}(x_i, X_o)$
  else
    $X_{\text{MAD}} = \texttt{append}(x_i, X_{\text{MAD}})$
  end if
end for
return $\bar{x}_{\text{MAD}} = \texttt{mean}(X_{\text{MAD}})$

Another measure of centrality that gives the user some control over the breakdown point. That is, the k-trimmed mean. The k-trimmed mean trims a total of $\lceil k\% \rceil$ samples from X_m, after sorting ascendingly. The freedom to select the percentage, k, allows control over the breakdown point as only the lowest and the highest values are trimmed, where $b_k = k\%$.

With this, the small data problem can be approached using the robust statistics discussed in this section.

11.3.2 Nonparametric Bootstrap

Another approach to small data would involve utilizing the *nonparametric bootstrap*. The nonparametric bootstrap [35] is a population-agnostic method that allows the construction of sample distributions without prior assumptions about the population's distribution. Its numerical nature allows it to achieve its target by means of sampling with replacement a large number of times, B. The nonparametric bootstrap resamples the original sample $X = \{x_1, x_2, \ldots, x_N\}$, generating B resamples, $X_b = \{x_{b1}, x_{b2}, \ldots, x_{bN}\}$ where x_{bi} can appear more than once in X_b, where x_{bi} is a random variable that samples uniformly from X. To construct the sample distribution of a statistic θ, each X_b is employed to compute B θ_b statistics, gathered in a vector $\boldsymbol{\theta}^*$, whose histogram represents the sample distribution.

Despite its numerical complexity for a large B, the bootstrap is useful for crowdsensing applications where the scenario is sparse [36]. It is of particular usefulness for small-sample cases due to the fact that its fair resampling has a low probability, P, of selecting a homogeneous sample (i.e., all samples in X_b being exactly the same), which is obtained as:

$$P(X_b = \{x_i, x_i, \ldots, x_i\}) = [1 - (1 - 1/N)^N]^N \tag{11.10}$$

Inspecting the Eq. (11.10), out of B resamples, only 3.45% will be extremely biased for a sample of size $N = 8$.

The usefulness of the bootstrap in obtaining the quality of small-sample scenarios comes from the presence of outliers. Since resampling is uniform, the outliers presence is promoted to have a probability of $1/N$, like any other sample x_i in X. Further analysis shows that the probability of an element's inclusion is:

$$P(x_i \in X_b) = 1 - (1 - 1/N)^N \tag{11.11}$$

which converges for a large N to 67%, i.e., each sample is present in 67% of the B X_b samples.

Furthermore, since B is a sufficiently large number, the properties of the central limit theorem are also applicable to the B θ_b statistics obtained. This property is of particular usefulness in the discussion of central measures of tendency (mean, median, mode, etc.), which causes the tendency of the bootstrap for the mean to follow a Gaussian distribution. However, due to the robustness of the MAD-mean, the resulting distribution is multimodal because of the employment of the median in the MAD-mean's trimming process. It limits the set of *medians* to be selected to a discrete number of medians (for an odd N there are N possible medians, for an even N there are $N^2 - \sum_{i=0}^{N-1} i$ possible medians).

11.3.3 The Bootlier

The bootlier, a graphical tool developed in [37], exploits the outlier promotion flaw in the bootstrap to detect outliers. In the bootlier, the difference between the mean and the trimmed mean, $\bar{X}_m - \bar{X}_{m,k}$, is bootstrapped to construct a sample distribution. The histogram of the resulting sample distribution is called a bootlier plot, which shows the impact of outliers present in a sample. The authors in [37] investigated the bootlier's multimodality and smoothness, and developed the bootlier index as a measure to quantify its smoothness. However, their work heavily depends on human intuition for assessing the quality of a sample, and for detecting outliers, as well as does not provide a straight metric for the evaluation of a sample's *quality*.

We have analyzed the bootlier plot and found that its ideal reference is in fact the distribution $N(0, 0)$, which is best described as the Dirac delta impulse [38], centered at 0. We have also found that the bootlier is a *superposition of leaking impulses*, which are very distinct when an outlier is present as it introduces another impulse at $xo - \bar{X}_{m,k}$. The source of leakage is the variation within the original sample, which causes the resulting sample distribution, i.e., the bootlier, to be smooth. The bootlier statistic hides the resulting multimodalities which combine to form a seemingly normal distribution around 0 that should ideally—for an absolutely perfect sample—be an impulse.

11.4 MMTM: A Quality Metric for Small Data

We have seen in the previous section how MAD-mean is a robust centrality estimate for the true value within a spatiotemporal cell, μ_m. We employ it in [39] along the trimmed mean to develop a quality metric that accounts for both range and accuracy. For range, one of the best solutions is the mean-trimmed mean difference, $\bar{X}_m - \bar{X}_{m,k}$, exhibits acceptable sensitivity to outliers below the breakdown point of $k\%$. For accuracy, a suitable solution would be the mean-MAD-mean difference, $\bar{X}_m - \bar{X}_{m,\text{MAD}}$, that is always robust about the breakdown point of 50%, which makes it less sensitive to the presence of outliers than the trimmed mean $\bar{X}_{m,k}$. So, we introduce in this section a hybrid quality metric that utilizes this trade-off between range and accuracy.

The selection of the suitable point in the trade-off-range accuracy depends on the scenario, the application, as well as the administrator's economical concerns. In this section, we will provide a dynamic metric that can stand at any point in the trade-off. Notice that high-outlier sensitivity affects the perception of the sample's quality, which is an important criterion in selecting the number of participants, that impacts the system's cost. We define a quality of source metric $Q_{m,s}$ that allows the MCS administrator to include or exclude normal/abnormal outliers by means of a sensitivity parameter β. In order to obtain the quality of a sample, we introduce the $\theta_{m,\text{MMTM}}$ statistic, defined as:

$$\begin{aligned}\theta_{m,\text{MMTM}} &= \beta(\bar{X}_m - \bar{X}_{m,k}) + (1-\beta)(\bar{X}_m - \bar{X}_{m,\text{MAD}}) \\ &= \bar{X}_m - \beta\bar{X}_{m,k} - (1-\beta)\bar{X}_{m,\text{MAD}}\end{aligned} \tag{11.12}$$

where β is a sensitivity parameter between 0 and 1.

The $\theta_{m,\text{MMTM}}$ is the difference between the mean and a weighted average of the more robust centrality estimates, the trimmed mean $\bar{X}_{m,k}$, and the MAD-mean $\bar{X}_{m,\text{MAD}}$. $\theta_{m,\text{MMTM}}$ can be rewritten in terms of the errors, where $\bar{X}_m$, $\bar{X}_{m,k}$, and $\bar{X}_{m,\text{MAD}}$ are reformulated as $\mu + \xi_{\bar{X}_m}$, $\mu + \xi_{\bar{X}_{m,k}}$, and $\mu + \xi_{\bar{X}_{m,\text{MAD}}}$, respectively. By substituting in Eq. (11.12), we obtain:

$$\begin{aligned}\theta_{m,\text{MMTM}} &= \mu + \xi_{\bar{X}_m} - \beta(\mu + \xi_{\bar{X}_{m,k}}) - (1-\beta)(\mu + \xi_{\bar{X}_{m,\text{MAD}}}) \\ &= \xi_{\bar{X}_m} - \beta\xi_{\bar{X}_{m,k}} - (1-\beta)\xi_{\bar{X}_{m,\text{MAD}}}\end{aligned} \tag{11.13}$$

which shows $\theta_{m,\text{MMTM}}$ defined in terms of the errors $\xi_{\bar{X}_m}$, $\xi_{\bar{X}_{m,k}}$, and $\xi_{\bar{X}_{m,\text{MAD}}}$. Ideally, for a perfect sample, this value should be zero. Nevertheless, this is impossible to happen empirically. We employ this fact to define quality as the closeness of this value to zero, where we formulate the quality Q_s based on a modified-$\theta_{m,\text{MMTM}}$ as:

$$Q_{m,s} = \log_\gamma \left(\frac{1}{\xi_{\bar{X}_m} - \beta\xi_{\bar{X}_{m,k}} - (1-\beta)\xi_{\bar{X}_{m,\text{MAD}}}} \right) \tag{11.14}$$

with γ a mathematical scaling factor selected for proper representation of the error, which we will fix to 10, the common log, in the simulations section. The derived quality metric is the common logarithm of the reciprocal of the difference between the mean error and the robust estimates error, which is ultimately the reciprocal of error due to outliers as:

$$\begin{aligned}&\xi_{\bar{X}_m} - \beta\xi_{\bar{X}_{m,k}} - (1-\beta)\xi_{\bar{X}_{m,\text{MAD}}} \\ &= \overbrace{\hat{\mu}_{\bar{X}_m} - \beta\hat{\mu}_{\bar{X}_{m,k}} - (1-\beta)\hat{\mu}_{\bar{X}_{m,\text{MAD}}}}^{\text{Estimation Errors}} + \overbrace{\beta\frac{N_{ko}}{N_k}\hat{\mu}_{\bar{X}_{m,k}} - \frac{N_o}{N}\hat{\mu}_{\bar{X}_m}}^{\text{Outlier}-\text{free } N \text{ Errors}} \\ &+ \underbrace{\beta\frac{N_{ko}}{N_k}\bar{\Theta}_k\sigma - \frac{N_o}{N}\bar{\Theta}\sigma}_{\text{Outlier Errors}}\end{aligned} \tag{11.15}$$

that comprises the sum of three difference terms. The first being between outlier-free estimation errors of the mean, the trimmed mean, and the MAD-mean; the second being the outlier-free sample size errors of the mean and the k-trimmed mean (whose breakdown point is at $k\%$); and the third being the error due to outliers, untrimmed and k-trimmed. Furthermore, the parameter λ is considered to be less than 2 for the

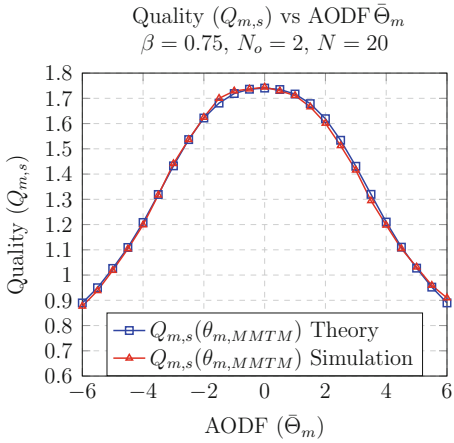

Fig. 11.3 MMTM quality metric for a variable average outlier deviation factor

purpose of increasing robustness due the small size of the samples present. $\lambda < 2$ provides a reasonable quality indication for the extreme when $\beta = 0$ and only the MAD-mean is present allowing β to have an impact on quality and its normal outliers tolerance.

However, this method has shown that it is suitable only down to sample sizes of 11. But, the bootstrap can be employed to go further to 8; however, the quality is redefined in terms of the bootstrap, by comparing it with the Dirac delta:

$$Q_{m,s} = \frac{1}{2}\left[\log_\gamma\left(\mu_{\text{MMTM}}^{-1}\right) + \log_\gamma\left(\sigma_{\text{MMTM}}^{-2}\right)\right] \tag{11.16}$$

where μ_{MMTM} and σ^2_{MMTM}, respectively, are the mean and variance of the $\theta_{m,\text{MMTM}}$ sample distribution, and $\log_\gamma$ is used to scale the values as later shown in the simulation section. The procedure to compute Q_s is summarized in Algorithm 2.

Algorithm 2 Bootstrap algorithm for sample quality assessment

Input: Readings from N sensors: $X = \{x_1, x_2, ..., x_n\}$
Output: Quality of Source: Q_s
Initialize : $B, \beta, k, \lambda, \gamma$
1: $\boldsymbol{\theta}^*_{\textbf{MMTM}} = \texttt{bootstrap}(X, B, \theta_{m,MMTM})$
2: $\mu_{\text{MMTM}} = \texttt{mean}(\boldsymbol{\theta}^*_{\textbf{MMTM}})$
3: $\mu_{\text{MMTM}} = \texttt{variance}(\boldsymbol{\theta}^*_{\textbf{MMTM}})$
4: **return** $Q_s = 0.5\left[\log_\gamma\left(\mu_{\text{MMTM}}^{-1}\right) + \log_\gamma\left(\sigma_{\text{MMTM}}^{-2}\right)\right]$

Figure 11.3 shows the relation between quality, $Q_{m,s}$, and the AODF, $\bar{\Theta}_m$. It follows a soft parabola, where at $\bar{\Theta}_m = \pm 6$ the quality is very low due to the fact that $N_o = 2$ outliers are present with values as far as $\pm 6\sigma_m$ away from μ_m. On the other hand, Fig. 11.4 shows the relation between quality $Q_{m,s}$ and the sample

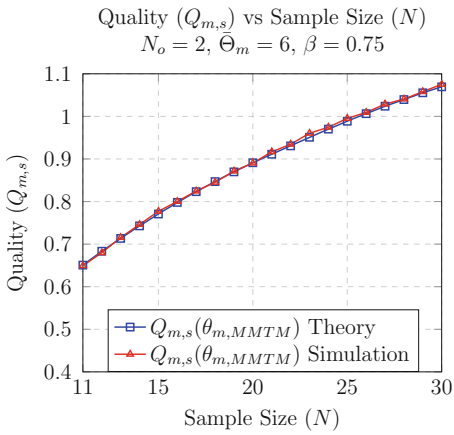

Fig. 11.4 MMTM quality metric for a variable sample size

size N. This relationship can be seen to be logarithmically increasing, which is a natural outcome due to the nature of statistics. A larger sample size implies a better representation of the population.

11.5 Potential Applications and Directions for Mobile Crowdsensing

MCS systems and task execution depend on user participation in one way or another. Due to users' reluctance or mobility, not all users may be available in a particular place at a particular time. Also, some user may act maliciously for whatever reason they deem worthy, intentionally submitting falsified data. All things considered, this ultimately leads to degradation of both quantity and quality of data received. This is when small data statistical techniques come into play. This section discusses some of the techniques developed to deal with small data sets, especially when infested in outliers. Techniques that improve both quality and trust of the received data.

Users are expected to be incentivized to provide their resources for MCS task execution; task requesters naturally prefer to decrease their costs, especially when they are on a limited budget. Reducing the number of participating users results in reduction of payment needed to incentivize these users. Thus, some papers take the budget approach as a cause for their "sparse data sensing." In [40], the authors introduce the concept of sparse MCS, which in essence deals with small data. It works under the assumption that there is a high spatiotemporal correlation in urban environments, i.e., larger context data can be interpreted from lesser inputs, proximate to the location geographically and in time. The framework proposed further reduces the load on users by shifting the statistical computation towards the server side. For that purpose, they propose Compressive Sensing, Dynamic Bayesian Network, and Gaussian Process Regression. While for assuring the data quality, the

paper utilizes the Bootstrap method. However, one of the downsides of the paper is that it does not assume mobility of users, i.e., participating users will remain in the targeted location for the targeted time.

Another way of dealing with small data set is proposed in [41]. The paper treats spatiotemporal space in which the tasks are being executed as an image. Each spatiotemporal cell is treated as a pixel. There is a sound assumption that some spatial cells will not have any willing, or otherwise available participants. However, using data correlation techniques such as *collaborative tensor decomposition* in order to achieve higher accuracy of missing data estimation. *Correlated time slots combination* technique is used in cases when the change in the sensed parameters is relatively infrequent, thus enabling to have better estimates with lesser sampling. It should be noted that the paper does not use interpolation techniques in order to fill cells with the missing data. On the contrary, the authors claim that their method outperforms the traditional interpolation numerical analysis techniques.

Yet another rather interesting approach is presented in [42]. This may not be strictly a small data set application in its narrow sense. The paper tries to extract knowledge about a particular user, and fill in the gaps for times when that user does not have access to a network (and thus cannot upload his/her data). Particularly the paper aims at knowing locations of particular interest to the user in question based on user's visits, such as work, home, friend's house, etc. The paper works under the assumption that all proximate user who do have Internet access should have similar data uploaded. Thus, through mobile tracing technique (a form of spatiotemporal clustering) the platform is capable of inferring the places of interest for the user. Using Enhanced Stay Point Clustering technique, the platform is able to determine how long the user stays in that particular place. In case of network unavailability to the user, the platform borrows information from other proximate users, giving priority to those who are geographically closer to where the user was last detected. Another example of what can be described as small data technique, with a stretch, is the compressed sensing (CS). CS allows for data sampling and acquisition and much smaller rates than the traditional Nyquist rate. Authors of [43] develop a CS technique which integrates it with IoT and WSNs. The technique allows to significantly reduce the amount of sampled data, and then accurately reconstruct. Despite the sampling frequency being lower than that of the Nyquist rate, the technique allows to avoid signal aliasing if the measurement matrices are selected properly. The reconstruction is done through linear and convex optimization methods.

11.6 Overview and a Closing Word

Utilization of small data set techniques opens many doors for both researchers and industry when it comes to MCS-specific applications. These techniques reduce the number of devices required to gather enough data, keeping information revenue unchanged, positively impacting attitude towards MCS systems. The most obvious

enhancement brought by employing small data techniques comes in the reduction of incentivization cost. By decreasing the number of devices employed in the sensing task, the number of users who should be paid for their resources is reduced as well. This should encourage both researchers and industry to adopt MCS systems for data gathering at a faster rate. Another positive consequence of small data set techniques adoption is the general shift towards smarter cities. Lesser data required to be transmitted requires less bandwidth. This decrease in one source of traffic congestion allows more devices to access the same bandwidth, using the same infrastructure. This allows MCS to be applied in situations such as crowd monitoring in stadiums during matches or concerts, a time when the amount of uploaded data may overload the system. This is may be especially useful when gathering data using MCS techniques in places where telecommunication companies do not provide newer generation cellular Internet connection. Another improvement brought on with utilization of small data sets techniques is the decrease of carbon footprint of an MCS task execution. With less data required to acquire the same information revenue, the greener the execution is, as less mobile devices are utilized, saving on energy costs. Finally, small data set techniques can potentially allow compression of data for storage purposes.

In case of humanity's miraculous revival against the odds of all the potential political upheavals facing it, we as humans are bound to live in our absolute majority in urban environments. In the beginning of the twentieth century, only about 10% of world's population lived in cities. Now, a mere 100 years later, this proportion has grown to over 50% [44]. This percentage is projected to grow higher in the future. One of the drivers of growth of cities' allure is the ever-increasing standards of life in cities when compared to rural areas. This includes education, health, technological, and infrastructural developments. With maturity of urban environments, connectivity of their inhabitants becomes more profound. This connectivity translates into larger number of interconnected devices, which ultimately gives rise to the IoT.

IoT is a very general paradigm. It is taking ground in our modern lives already. TVs, cars, and even refrigerators can be connected to and interacted with via Internet nowadays. However, MCS systems, which are included by the IoT paradigm, are still a very new topic. MCS yet requires more research and more development to be viable on a mass scale and be adopted by the industry and public service providers. This adoption, when happens, will be among the defining features of what we call a "smart city."

However, as stated, MCS and IoT in general do require further research, where some examples are presented in this chapter. One of the fields that is underrepresented in the area is the subfield of small data sets. These techniques are required to deal with whenever the data is not enough or the number of participants is not sufficient. From a trial to discern the truth from a small population, to actively trying to reduce the number of participants in order to save on budget, to compression of data, small data set techniques can find uses in many fields of IoT.

Acknowledgements This work was made possible by NPRP grant NPRP 9-185-2-096 from the Qatar National Research Fund (a member of the Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. R.K. Ganti, F. Ye, H. Lei, Mobile crowdsensing: current state and future challenges. IEEE Commun. Mag. **49**(11), (2011)
2. D. Wang, Stuck in traffic? (2007). Available: https://googleblog.blogspot.qa/2007/02/stuck-in-traffic.html
3. D. Sijabat, R. Harwahyu, R.-G. Cheng, Energy-efficiency of RACH-based small data transmission scheme in LTE networks, in *2017 40th International Conference on Telecommunications and Signal Processing (TSP)* (IEEE, Piscataway, 2017), pp. 106–109
4. M. Koseoglu, Performance analysis of small data transmission schemes for cellular m2m communications, in *2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net)* (IEEE, Piscataway, 2017), pp. 1–6
5. F.M. Al-Turjman, H.S. Hassanein, M. Ibnkahla, Quantifying connectivity in wireless sensor networks with grid-based deployments. J. Netw. Comput. Appl. **36**(1), 368–377 (2013)
6. B. Jia, S. Liu, T. Zhou, Z. Xu, Opportunistic transmission mechanism based on SI in mobile crowd sensing networks, in *2017 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)* (IEEE, Piscataway, 2017), pp. 211–215
7. S.K. Datta, R.P.F. Da Costa, C. Bonnet, J. Härri, oneM2M architecture based IoT framework for mobile crowd sensing in smart cities, in *2016 European Conference on Networks and Communications (EuCNC)* (IEEE, Piscataway, 2016), pp. 168–173
8. J. Ni, X. Lin, K. Zhang, Y. Yu, Secure and deduplicated spatial crowdsourcing: a fog-based approach, in *2016 IEEE Global Communications Conference (GLOBECOM)* (IEEE, Piscataway, 2016), pp. 1–6
9. R.F. El-Khatib, N. Zorba, H.S. Hassanein, Performance analysis of small data transmission schemes for cellular M2M communications, in *IEEE ICC*, Kansas City (IEEE, Piscataway, 2018)
10. M. Pouryazdan, B. Kantarci, T. Soyata, L. Foschini, H. Song, Quantifying user reputation scores, data trustworthiness, and user incentives in mobile crowd-sensing. IEEE Access **5**, 1382–1397 (2017)
11. H. Gao, C.H. Liu, Y. Tian, T. Xi, W. Wang, Ensuring high-quality data collection for mobile crowd sensing, in *2017 IEEE Wireless Communications and Networking Conference (WCNC)* (IEEE, Piscataway, 2017), pp. 1–6
12. J. Hamm, A.C. Champion, G. Chen, M. Belkin, D. Xuan, Crowd-ML: a privacy-preserving learning framework for a crowd of smart devices, in *2015 IEEE 35th International Conference on Distributed Computing Systems (ICDCS)* (IEEE, Piscataway, 2015), pp. 11–20
13. Y. Wu, Y. Wu, J. Zeng, H. Chen, C. Li, PIE: a personalized incentive for location-aware mobile crowd sensing, in *2017 IEEE Symposium on Computers and Communications (ISCC)* (IEEE, Piscataway, 2017), pp. 981–986
14. X. Ma, Z. Zheng, F. Wu, G. Chen, Trust-based time series data model for mobile crowdsensing, in *2017 IEEE International Conference on Communications (ICC)* (IEEE, Piscataway, 2017), pp. 1–6
15. T. Luo, L. Zeynalvand, Reshaping mobile crowd sensing using cross validation to improve data credibility (2017, Preprint). arXiv:1709.03495
16. P. Ma, D. Tao, T. Wu, A pseudonym based anonymous identity authentication mechanism for mobile crowd sensing, in *2017 3rd International Conference on Big Data Computing and Communications (BIGCOM)* (IEEE, Piscataway, 2017), pp. 10–14

17. S. Gisdakis, T. Giannetsos, P. Papadimitratos, Security, privacy, and incentive provision for mobile crowd sensing systems. IEEE Internet Things J. **3**(5), 839–853 (2016)
18. J. Misra, P. Pal, A. Bagchi, Spear: design of a secured peer-to-peer architecture, in *International Conference on Information Systems Security* (Springer, Cham, 2005), pp. 322–327
19. X. Zhang, G. Xue, R. Yu, D. Yang, J. Tang, Robust incentive tree design for mobile crowdsensing, in *2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS)* (IEEE, Piscataway, 2017), pp. 458–468
20. R. Zhou, Z. Li, C. Wu, A truthful online mechanism for location-aware tasks in mobile crowd sensing. IEEE Trans. Mobile Comput. (2017). Early Access
21. L. Atzori, R. Girau, S. Martis, V. Pilloni, M. Uras, A SIoT-aware approach to the resource management issue in mobile crowdsensing, in *2017 20th Conference on Innovations in Clouds, Internet and Networks (ICIN)* (IEEE, Piscataway, 2017), pp. 232–237
22. Y. Qu, Z. Yu, X. Zheng, Buy4Me: a delivery system via mobility prediction based on mobile crowd sensing, in *2015 IEEE UIC-ATC-ScalCom* (IEEE, Piscataway, 2015), pp. 1199–1203
23. Y. Krytska, I. Skarga-Bandurova, A. Velykzhanin, IoT-based situation awareness support system for real-time emergency management, in *2017 9th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications (IDAACS)*, vol. 2 (IEEE, Piscataway, 2017), pp. 955–960
24. C. Feng, W. Wang, Y. Tian, X. Que, X. Gong, Estimate air quality based on mobile crowd sensing and big data, in *2017 IEEE 18th International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)* (IEEE, Piscataway, 2017), pp. 1–9.
25. Y. Zhang, M. Chen, S. Mao, L. Hu, V. Leung, CAP: community activity prediction based on big data analysis. IEEE Netw. **28**(4), 52–57 (2014)
26. A.S. El-Wakeel, J. Li, A. Noureldin, H.S. Hassanein, N. Zorba, Towards a practical crowd-sensing system for road surface conditions monitoring. IEEE Internet Things J. (2018). Early Access
27. L. Wang, D. Zhang, C. Chen, X. Han, A. M'hamed, Sparse mobile crowdsensing: challenges and opportunities. IEEE Commun. Mag. **54**, 161–167 (2016)
28. C.E. Shannon, Communication in the presence of noise. Proc. IRE **37**(1), 10–21 (1949)
29. N. Stubbs, S. Park, Optimal sensor placement for mode shapes via Shannon's sampling theorem. Comput. Aided Civ. Inf. Eng. **11**(6), 411–419 (1996). Available: https://onlinelibrary.wiley.com/doi/abs/10.1111/j.1467-8667.1996.tb00353.x
30. R. Hogg, E. Tanis, *Probability and Statistical Inference* (Prentice Hall, Englewood Cliffs, 2006). Available: https://books.google.com.qa/books?id=MKl1AAAACAAJ
31. P. Rousseeuw, C. Croux, Alternatives to the median absolute deviation. J. Am. Stat. Assoc. **88**, 1273–1283 (1993)
32. E. Grafarend, J. Awang, Sampling distributions and their use: confidence intervals and confidence regions, in *Linear and Nonlinear Models: Fixed Effects, Random Effects, and Total Least Squares* (Springer, Dordrecht, 2012), pp. 637–751
33. C. Leys, C. Ley, O. Klein, P. Bernard, L. Licata, Detecting outliers: do not use standard deviation around the mean, use absolute deviation around the median. J. Exp. Soc. Psychol. **49**, 764–766 (2013)
34. G. Buzzi-Ferraris, F. Manenti, Outlier detection in large data sets. Comput. Chem. Eng. **35**(2), 388–390 (2011)
35. B. Efron, Bootstrap methods: another look at the jackknife. Ann. Stat. **7**, 1–26 (1979)
36. V. Freschi, S. Delpiori, E. Lattanzi, A. Bogliolo, Bootstrap based uncertainty propagation for data quality estimation in crowdsensing systems. IEEE Access **5**, 1146–1155 (2017)
37. K. Singh, M. Xie, Bootlier-plot: bootstrap based outlier detection plot. Sankhyä Indian J. Stat. **65**, 532–559 (2003)
38. G. Arfken, H. Weber, *Mathematical Methods for Physicists*, 6th edn. (Elsevier Academic Press, Burlington, 2005)
39. S.B. Azmy, N. Zorba, H.S. Hassanein, Robust quality metric for scarce mobile crowd-sensing scenarios, in *IEEE ICC*, Kansas City (IEEE, Piscataway, 2018)

40. L. Wang, D. Zhang, Y. Wang, C. Chen, X. Han, A. M'hamed, Sparse mobile crowdsensing: challenges and opportunities. IEEE Commun. Mag. **54**(7), 161–167 (2016)
41. X. Kang, L. Liu, H. Ma, Enhance the quality of crowdsensing for fine-grained urban environment monitoring via data correlation. Sensors **17**(1), 88 (2017)
42. S. Vhaduri, C. Poellabauer, Hierarchical cooperative discovery of personal places from location traces. IEEE Trans. Mob. Comput. (2017). Early Access
43. S. Li, L. Da Xu, X. Wang, Compressed sensing signal and data acquisition in wireless sensor networks and internet of things. IEEE Trans. Ind. Inf. **9**(4), 2177–2186 (2013)
44. C. Fiandrino, A. Capponi, G. Cacciatore, D. Kliazovich, U. Sorger, P. Bouvry, B. Kantarci, F. Granelli, S. Giordano, Crowdsensim: a simulation platform for mobile crowdsensing in realistic urban environments. IEEE Access **5**, 3490–3503 (2017)

Index

F. Al-Turjman (ed.), *Performability in Internet of Things*, EAI/Springer Innovations in Communication and Computing, https://doi.org/10.1007/978-3-319-93557-7