Jimma University

Jimma Institute of Technology

School of Computing

Computer Networking MSc Program

Designing a model for improvement of network efficiency in mobile crowd-sensing

By

Adugna Asrat Ketero

A Thesis Submitted to the School of Graduate Studies of Jimma University in Partial fulfilment of the Requirements for the Degree of Masters of Science in Computer Networking

July, 2017 G.C

Jimma, Ethiopia

Jimma University

Jimma Institute of Technology

School of Computing

Computer Networking MSc Program

Designing a model for improvement of network efficiency in mobile crowd-sensing

By

Adugna Asrat Ketero

A Thesis Submitted to the School of Graduate Studies of Jimma University in Partial fulfilment of the Requirements for the Degree of Masters of Science in Computer Networking

July, 2017 G.C

Jimma, Ethiopia

Declaration

I, the undersigned, declare that this thesis entitled Designing a model for improvement of network efficiency in mobile crowd-sensing is my original work and has not been presented for a degree in this or any other universities, and all sources of references used for the thesis work have been appropriately acknowledged.

Name: Mr. Adugna Asrat Ketero

Signature: ______________ Date: _________________

This thesis has been submitted for examination with my approval as a University advisor.

Advisor Name: Dr. Ahmedin M. Ahmed (PhD)

Signature: _________Date: July 21, 2017

This thesis has been submitted for examination with my approval as a University Co-advisor.

Co-Advisor: Mr. Kebebew Abebu (MSc)

Signature: ______________ Date: _________________

This Thesis is approved by Examining Board:

Examiner's Name: ______________________________________

Signature: ______________________ Date:______________

Chairman's Name:______________________________________

Signature: _________________ Date: ____________

DEDICATION

I dedicate this thesis for my beloved wife Mitsiwat Yohannis

ACKNOWLEDGEMENT

There are helps and supports behind someone to achieve to success of any good work. First of all, I thank God who supports me in any challenge that faces me all in my life time. It is my great pleasure to gratitude my advisers Dr. Ahimedin M. Ahimed and Mr. Kebebew Ababu (M.Sc.) for their devotion, continuous support and follow up throughout the thesis work. I wish to extend my acknowledgement to my friends and family for their important help and supports. Finally, I thank school of computing for initiating me to investigate my postgraduate study.

Table of Contents

List of Figures

List of tables

List of Algorithms

List of Equations

List of Acronyms

API	Application Programming Interface
BS	Base Station
CSMA	Crier Senor Multiple Access
D2D	Device to Device
DSR	Dynamic Source Routing
FTP	File Transfer Protocol
GPS	Geographical position system
GSM	Global System for Mobile communication
HTTP	Hyper Text Transfer Protocol
IoT	Internet of Things
LBSN	Location-based Social Networks
MAC	Machine Access Code
MSN	mobile social network
MTE	Minimum Transmission Energy
NS2	Network Simulator version 2
SDK	Software Development Kit
TCP	Transmission Control Protocol
TDMA	Time Division Multiple Access
UDP	Uniform Datagram Protocol
WiFi	Wireless Fidelity
WMSN	Wireless Multimedia Sensor Network
WSN	Wireless Sensor Network

Abstract

Mobile crowd sensing is human centric computing technology with sensing and computing devices in the mobile users. The tendency to the recent technology, the smartphone user rapidly increases from day to day. This provides the good opportunity to sense with smartphone about the environment and urban events without deploying special sensor devices. There are numbers of application and data uploading models that motivates users to contribute data by using their smartphones. Recently, mobile crowd sensing (MCS) architecture have two components in sensing and uploading architecture, those are, sensing devices that receive information form environment and back-end server for analysis of sensor data. However, each data contributing user collect and send the redundant sensed data to the cloud server. There are no common devices or mechanism that faces the common challenges such as resource allocation, data collection and data processing tasks. In this paper, we designed a model for improvement of network efficiency in mobile crowd-sensing by providing device to device communication with their group leader and data fusion before uploading to the backend server. We conducted simulation with NS2 network simulator for experimentation and to evaluate the performance of our model versus raw data uploading model with different evaluation metrics. The result show that our scenario model improves the performance and reduce the data redundancy in mobile crowd sensing network.

Keywords – mobile crowd sensing, mobile Sensor Networks, IOT, smart city.

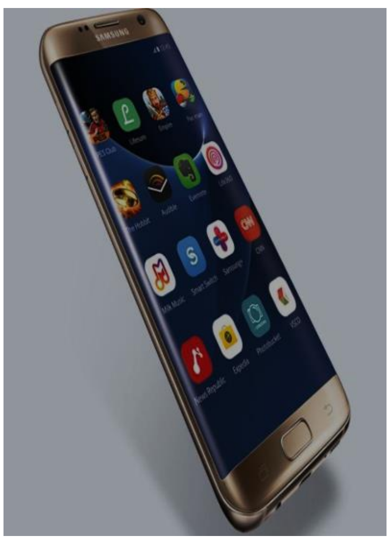
CHAPTER ONE

INTRODUCTION

1.1 Background

Mobile devices such as smart phones and in-vehicle systems represent a new type of geographically distributed perceptual infrastructure that enables the movement of people-centered perceptions [1][3]. According to the forecast of global Smartphone shipments from 2010 to 2020, more than 1.5 billion mobile phones are expected to be shipped worldwide [13].

As shown in Figure 1, smart phones already have several sensors: cameras, microphones, GPS, accelerometers, digital compasses, optical sensors, Bluetooth as proximity sensors and in the near future they are envisioned to include health and contamination monitoring sensors.



Samsung galaxy s7

- ✓ Camera
- ✓ Audio
- ✓ Accelerometer
- ✓ proximity
- ✓ Ambient light
- ✓ GPS
- ✓ Compass

Figure 1: smart phone Samsung galaxy s7[1]

Mobile crowdsensing is one of the new events used to exchange everyday information for successful management of smart city. Smart-phone (iphone6, Samsung galaxy), iPods (music player) and vehicle sensing devices are become well known sources of sensor data and capable to upload data to the Internet. Mobile sensing is the most recent technology and considered as optimal solution to the information of the smart city by reducing the cost of the device that is needed to install in every road. This system replaces the cost full sensor devices by using individual mobile devices instead of new devices. This opens the good opportunities to each individual to share their daily information. A typical mobile investigation system is involving three main actors: (1) end-users (data providers) who contribute sensor data, (2) service providers (SPs, also known as collection points (CPs)) to process the collected data to generate services, and (3) the end-user (data requester) requesting the service. Data collection (including request collection from data requesters and data collection from data providers) is the basic framework for constructing mobile sensing systems. Some current literature assumes that end users, once generated by their device sensors, use cellular network resources to transfer data to a collection point. However, this approach increases the communication cost and creates additional workloads for the cellular network. This problem becomes worse when large amounts of data are generated (for example, when the data type is a quality photo) or when a network busy time occurs. We proposed a model for network efficiency in mobile crowd-sensing to refer the novel model to overcome the sensor data redundancy problem, to minimize communication cost and to reduce workloads of cellular network by providing the D2D communication system that able to consider the common resource such as data collection, energy conservation and resource allocation.

1.2 Motivation

Migration of technology highly accelerated from desktop to mobile devices (such as Smartphone, palmtop, tablet and laptop) initiates to investigate on mobile devices technology. Multi-modality of sensing capabilities in the mobile devices (i.e. Bluetooth, microphone, compass, camera, Gyroscope, Proximity, ambient light and GPS) are motivates to focus on the mobile sensing paradigm. This multimodality allows people to take were ever they go and whatever they do instead of deploying in the field and the dynamic conditions in the collection of data in the mobile devices is good opportunity to gather information in everywhere. Also cost of sensing device that deployed in the field is expensive when we compare to the mobile sensing devices.

1.3 Statement of the problem

The numbers of mobile crowd sensing model such as MSF, an extensible mobile sensing platform [3] and Silent Mobile Sensing Framework for Smart Cities [4] have been proposed in mobile sensing infrastructure. However, the existing architecture scenario for MCS uses application silo approach (i.e. each application is built independent from each other). In recently architecture there are two components, sensing devices that receive information form environment and back-end server for processing and analysis of sensor data [1] [5]. This architecture illustrated in the fig 2.

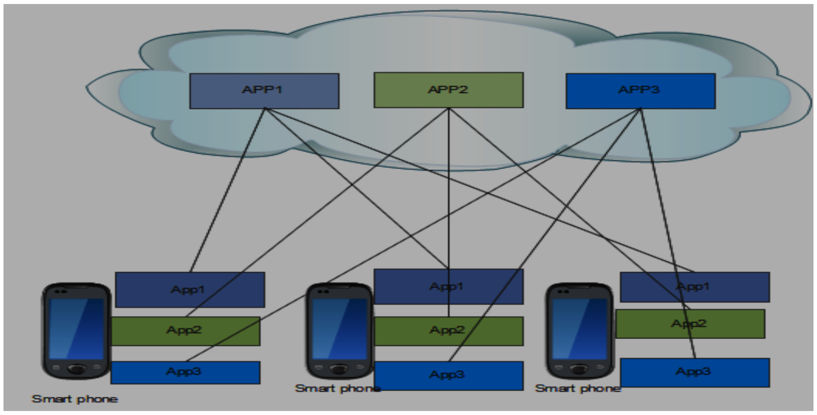


Figure 2:Existing MCS applications take an "application silo" approach[2]

There are no common devices or communication system that faces the common challenges such as resource allocation, data collection and energy conservation [1] [5]. Applications perform sensing and processing activities independently without considering the consequences on each other will result in low efficiency on resource constrained platform. There is a high possibility of replicating sensing and processing among multiple applications.

For example, though measuring ambient noise levels, if two users standing at same location and record noise levels on their cellphones and upload it to the server, this is a case of data redundancy. Air pollution, noise and traffic sensing needs location information, but this all applications use its own data without considering reusing the same data samples. To the best of our knowledge the silo approach does not enable to eliminate duplicate sensing data and processing in multiple applications. This paper answers the questions, how to reduce mobile phone sensing and uploading data redundancy in mobile crowdsensing network and when uploading data to the backend server there are communication channel bandwidth consumption, so how to optimize the MCS communication efficiency that happens when uploading redundant data by optimizing the data transmission

1.4 Objectives

1.4.1. General objectives

- To analyze the performance of a MCS model by designing novel model to reduce task load in mobile crowdsensing networks and data cost to the participants.

1.4.2. Specific objectives

- To investigate the existing data collecting and processing architecture.
- To analyze the effectiveness of existing mobile crowd sensing architecture.
- To propose, design and simulate a solution to reduce workload of server that processing similar data for multiple time.
- To evaluate the performance of the proposed approach using an appropriate metrics.
- To compare the proposed scenario with an existing scenario through simulation.

1.5 Methodology

Designing a model for improvement of network efficiency in mobile crowd-sensing by considering duplication of sensing and processing data when collecting and sharing among multiple nodes is the aim of the study. To achieve this aim effectively we conducted the following methodology;

Literature review:

- Investigate in the existing system of mobile crowd sensing and its client server architecture.
- Reviewing different literatures which related to the topic issue.
- Examine the existing approaches and understand the mechanism how to achieve the proposed objectives.

Analysis and Modeling:

- Designing the model and developing the algorithm which solves the discussed problem above.

Mechanism of Driving Conclusion:

- Testing and evaluating with NS2 simulator to test the performance of the new model.
- The simulation result of the new scenario compared with an existing scenario.

1.7 Scope of the study

Mobile crowd sensing is current and hot research area that include various type of issue such as application development to gather information, energy conservation management ,user privacy, location optimization, routing algorithm that forwarding sensed data to the backend server and others are included in mobile crowd sensing. However, this study focuses on designing a model for improvement of network efficiency in mobile crowd-sensing.

1.8 Limitation of the study

- Does not support data replication and Distributed caching at group head schemes
- Does not consider location optimization
- Does not include application development to gather information

1.9 Contribution

This thesis work demonstrates how to reduce data redundancy in mobile crowd sensing network through analysis and rebuilding existing model of MCS. The intensions of related works focus on, how to use the smart mobile devices as sensor node instead of infrastructure based deployed sensor node but, this introduce the new challenges with data cost. To the best of our Knowledge this paper can contribute performance improvement in the mobile crowd sensing network by optimizing resources such as channel bandwidth, storage devices in the backend server and also reduces the data processing workload in the backend server by providing device to device communication when they sharing their sensed data over MCS network.

1.10 Thesis Organization

The remaining of the paper is organized as follows: In Chapter 2, we review the literature on mobile crowdsensing and surveying various sensing architectures. Chapter 3 describes the related work that has been studied in the mobile crowd-aware data collection model. Chapter 4 describes the proposed architecture platform. Chapter 5 describes the implementation and performance evolution of the proposed model. Finally, Chapter 6 summarizes the paper and discusses future work.

CHAPTER TWO

LITERATURE REVIEW

2.1 INTRODUCTION

A MCS (Mobile crowd sensing) needs a client side Application running on the mobile phone for reading an internal mobile phone's sensor, or external sensors in the wireless sensor network and transmitting sensed data to the Web server. To do this, the mobile operating system must offer an API (Application Programming Interface) to handle the data sensing and transmitting[3]. Mobile crowd sensing is different from traditional wireless sensor networks. Compared to traditional wireless sensor network MCS has a lot of unique characters. Recently mobile devices have high computing storage resource and computing capability than ancient wireless sensor network aggregated with multimodality sensing devices such as camera, microphone, accelerator, light sensor and compass. People take this device were ever they go and they do instead of installing vehicle and road side unit, it is possible to detect looping, traffic data and congestion by using Smartphone handled by the user and used to reduce the cost of devices and deployment prices [3]. A successful community and city management is based on effective monitoring of urban and community dynamics for decision making and policy making. To achieve this, traditional sensing techniques, such as sensor networks, can remove sensors that are usually distributed to obtain real-world conditions [3]. However, although there are a growing number of workgroups on sensor networks, they have never been successfully deployed in the real world due to a variety of reasons, including commercial sensor networking techniques, high installation cost, inadequate spatial coverage, and the like[1]. Mobile Crowd Detection is a large-scale perception paradigm based on the power of attached devices, such as mobile phones, smart gadgets, wearable gadgets and the like[4].MCS allows an increasing number of mobile phone users to share local knowledge (eg, local information, media context, noise level and traffic conditions) acquired by sensors developed by sensors and gather information in the cloud for large-scale detection and community intelligence mining [5]. The mobility of large-scale mobile users makes MCS a versatile platform that can often take the place of static perception infrastructures. Thus, a wide range of applications such as traffic planning, environmental monitoring, mobile social advice, public safety and so on are enabled. An official definition of MCS; A new perception paradigm that connects ordinary citizens in the cloud for contributions to the perceived or produced data from mobile devices, collection and extraction of crowd intelligence, and personalized service delivery. From an AI perspective, MCS is built with a distributed problem solving model[6]. In the history of literature, a crowded problem-solving concept has been explored in various research areas. Ten years ago, Surowiecki wrote a book entitled "The Wisdom of Communities" (or crowd wisdom) in which a general phenomenon - the gathering of information or information from a group of people, usually results in better decisions than ever. It comes from one person - from the group. It distinguishes four important qualities that make a crowd smart: the diversity of views, the independence of the thinker, the management of the place and the gathering of ideas. A concept similar to crowded wisdom is collective intelligence. Unlike the two concepts that focus on the advantages of the group decision maker, MCS is mainly concerned with population-based data collection and processing. According to Merriam-Webster Glossary 1, crowdsensing is defined as the practice of obtaining the necessary services or content by claiming contributions from a large group of people, especially an online community. A typical example is Wikipedia, where tens of thousands of contributors have collaboratively created Wikipedia's largest encyclopedia of the world. Compared to MCS, however, crowdsensing focuses on the involvement of online crowds. The closest concept to MCS, participant perception[5]. The average citizen and cooperating mobile devices are tasked with setting up participant sensor networks to collect and share local knowledge. The definition of participant perception emphasizes open user participation when suggested. With the rapid development of smartphone detection and mobile Internet technologies in recent years, the scope of crowded problem solving systems has expanded using mobile devices. This Finally, the definition of participant perception is based on two aspects and the new concept of mobile crowd perception [1]. First, the user receives data (from the physical community) and mobile social network services (from the online community) that are sensed from the MCS mobile devices. In other words, MCS counts both explicit and implicit user participation to collect data. Second, MCS provides the fusion of machine and human intelligence in the perception process.

The types of mobile data which heterogynous device produces shows the heterogeneity of mobile devices and the quality in the latency and accuracy, changes all the time due to the dynamic nature of the mobile devices. Selecting the correct set of devices that can produce the expected data is the challenges in MCS.[2] In the traditional mote-sensor network the storing, the way in which data they produce is well known. This is easy to control the quality of data. In the mobile crowd sensing the sensor data used for many deferent purposes, for example the accelerator reading used to transportation mode identification is transport ministry, also used for human activity identification. The full support of common data reusability is the critical issue in today mobile crowd sensing technology. In the other hand in traditional sensor network common data reusability is rarely needed [1][7][8]. The followings are the features of mobile crowd sensing that make it unique.

2.2 Features MCS

The unique features of the MCS application we first described are distinguished from the traditional nickname-level sensor networks. This idea addresses the challenges faced by MCS applied research readers. Compared to traditional nickname-level sensor networks, mobile crowdsensing has many unique characteristics that bring both new opportunities and problems. First of all, today's mobile computing devices have significantly more than the dust-level sensors, communication and storage resources, they are multi-mode is usually equipped with sensing function. This allow many applications to take advantage of the resources and sensory enforcement methods beyond existing nicknames to have first-class sensors.

2.2.1. Multimodal sensing

Compared to traditional sensor networks, the main difference between MCS and traditional wireless sensor network is that it bases on large-scale sensing. Particularly, subordinate participation offers some advantages to MCS. MCS strengthens the existing sensing and communication infrastructure, thus the distribution costs are very low. The mobility of mobile users provides unprecedented space-time coverage compared to static sensor network deployments. A summary of the differences between MCS and conventional sensor networks is given in Table I.

	Operators	Deployment Cost	coverage	Data Quality
WSN	• Government agencies • public institutions	High: Expensive sensors and infrastructure to deploy the network	Low: Leveraging existing infrastructure, i.e., broad propagation of cellular network and mobile device usage	High, sound level sensors
MCS	Possibly everyone	Low: Leveraging existing infrastructure, i.e., broad propagation of cellular network and mobile device usage	The essential mobility of the phone carriers provides extraordinary spatiotemporal coverage	Low, suffering from problems such as built-in sensor performance and the dependable of user contributed data

Table 1 the comparison between MCS and wireless sensor network

Community participation in the perception cycle is the most important feature of the participants Perception. Participant is data collection method used by perception. Mobile social network (MSN) is the interesting evolution of the WSN. With the rapid development of the mobile Internet, MSN services[9] and objects are growing rapidly, while large-scale user opens a new window to understand the dynamics of the city and the community that is considered as a source of data. Participant mobile detection and the combination of participating MSN data is a unique feature of MCS. Crowd detection style[10] is spreading to a wide range of levels of human contribution based on traditional crowd perception, participatory perception, and opportunistic perception. Having two different participant data contribution modes, we can change the detection style of the MCS and categorize it from a new dimension: user awareness and perception task. On the one hand, for participatory or opportunistic perception, data collection is the primary purpose of the application. As a result, the detection mission is open to the user as knowledgeable. On the other hand, the primary use of MSN services is for social interaction; Whereas the data produced by online interactions are indirectly used for collective intelligence extraction. In other words, we can explain MCS's perception style at two levels: open and closed. Volunteer organization. Participants may be self-organizing community with varying levels of organizational participation, from loosely organized neighborhood groups, against previously shared problems, to previously established involvement groups[11]. In particular, we classify the volunteer organization into the following three modes of operation at different scales.

1. **Group**: It refers to a group of loosely or opportunistically organized neighbors. In other words, telephone sets on the ground working jointly with a shared problem. For example, [12], transit vehicles can detect and share traffic signal information and adjust driving speeds.[13] allows the collection and replay of a local social event using locally captured information. In this mode, people are often referred to as opportunistic or transient groups and the group member is weak or strong. Key techniques include group formation / identification and management. For example, we should divide the mobile device cluster based on related physical or social contexts[14].
2. **Community**: According to Meriam Webster's Dictionary, a community is defined as "people who live in a particular area, or people who are considered as a unit because of their common interests, social groups, or nationality". Community-based MCS, Duties can come from an existing community or can easily build a new community. The members trust each other and are likely to interact with each other during task execution, thus ensuring high quality data. For example, the MIT project benefits from sensor-enhanced smartphones such as GPSs, directional microphones, etc. to evaluate populations in a large area. Hundreds of unprofessional botanists have reported using cell phones to collect pictures of plants to study the link between climate change and ecosystems [15]
3. **Urban:** It targets a wide range of applications such as urban traffic dynamics[16] and air / sound pollution inspection[11]. Any citizen mostly strangers can participate in

perceptual activity at the urban scale. However, when compared to the above two modes, data contributed by ordinary users are generally of low quality data can contribute to fake information. We summarize the properties of the MCS in three different scales in table II

Scale	Data Quality	Collaboration	Social link Strength	Key Techniques	Major Applications Areas
Group[12]	Medium	Close	Weak or strong	Opportunistic connection and group formation	Local area sensing, local event replay
Community[15]	High	Close	medium	Opportunistic connection and group formation	Scientific investigation, community-specific services
Urban[15]	Low	Loose	weak	Data quality, trust maintenance	Urban dynamics sensing, environment monitoring

Table 2:MCS methods of operation at different scales

2.2.2. Hybrid Human Machine Systems

In order to motivate full scale user engagement and improve user experience, MCS systems should be developed in a human centered manner. In particular, it address the following key issues:

Movement of human participation: Mobile devices usually contain limited resources such as energy, bandwidth, etc., and sensory information from them is often very sensitive. It is important to develop an incentive model to facilitate data sharing among peers[12]. Establishing a large-scale MCS system typically requires a significant number of participants and some may be withdrawn from the collection cycle as long as they are not more than the expected return on investment. Questions about human motivation have been central to philosophy and the economy. For example, promises of financial or monetary gain have been an important incentive method for actors in the majority of markets and traditional organizations. Interest and entertainment are important motivators even in the absence of monetary gain in many cases[17]. In addition, people can be motivated to participate in an activity with social and ethical reasons, such as socializing with others, reputation, or being recognized by others. Some crowd detection services require users to present their data for service use. There are also indirect ways to increase user participation, such as energy saving and privacy protection mechanisms in data participation[4].

The bringing together of human-machine intelligence: The involvement of human participation in the perception process will lead to a mixture of human and machine intelligence at MCS. On the one hand, when participating in a crowded data sensing cycle, human intelligence is incorporated into the obtained data such as the contributors may decide what to capture. On the other hand, both machine intelligence i.e. machine learning and data mining techniques as well as human intelligence like cognition, reasoning etc. can be used for data processing i.e. classification, decision making, based on the prosperity of data collected in MCS systems. By combining the intelligence of both crowds and machines, MCS allows the creation of hybrid human machine systems. However, such human intelligence and machine intelligence can be balancing or contrary and make them strong and weak for different computer problems. For this reason, new approaches and techniques should be explored for the optimal fusion of human and machine intelligence in MCS systems.

User security and privacy: Sharing of personal data such as user location, surround sound in MCS applications can cause significant concerns about security and user privacy. To motivate user participation, we must research new techniques to maintain data contributing information security. In particular, the definition of security and confidentiality can continue to evolve as MCS systems mature; Even though this personal information may not be obtained directly by an unwanted party, many of the information may be removed from the aggregate. For example, uniquely identifying an object with an RFID tag and returning it to the user can lead to many privacy problems[4]. For this reason, new privacy protection approaches should be developed to avoid gathering important information from mined models[13].

2.2.3. Temporary Networking

The success of MCS relies on its ability to provide ad hoc network connectivity and use heterogeneous communication capabilities everywhere to efficiently aggregate mobile crowd detection data. Different MCS applications or systems meet a variety of connection architectures and communication requirements, most of which share the following features.

Heterogeneous network connection: Existing mobile devices are often equipped with multiple wireless communication interfaces and are supported by different wireless technologies. For example, a smartphone may have at least GSM, WiFi, and Bluetooth interfaces. GSM and WiFi interfaces can provide network connectivity in large areas, via a preexisting communication infrastructure i.e. via an urban area cellular base station or WiFi access points in a campus building, while Bluetooth or WiFi can also provide a short range of connectivity. To create self-organizing opportunistic networks for sharing[12]. Heterogeneous networks support new research challenges and enriched opportunities for MCS applications by supporting ad-hoc networking services such as connectivity, collaboration detection and data routing / transmission for participants who cross these multiple wireless networks.

Time-varying network topology and human mobility: Both the hardware sensors and the software sensors are located in the MCS. The first is the embedded sensors such as accelerometer, GPS, camera in portable devices, and the second is the user-generated data numbers from the MSN portals. We can group detection styles according to the degree of user participation in MCS as participant detection, opportunistic detection, and hybrid detection i.e. machine calculation and user control combination. On the other hand, we can classify them as covert detection and explicit detection in terms of user consciousness for the detection task. Detection tasks are either continuous (continuous detection) or event triggered detection. For the former case the corresponding sensors are running continuously under the parameter Specifies the settings of the detection task such as sampling rate, time. However, continuous detection in mobile devices can be very expensive in terms of energy [18]and some detection tasks are meaningful only in certain sampling contexts like a specific time interval or places. In such cases, triggers must be defined to capture data in a sensitive manner[8]. Crowd data collection.

As for the scale of the crowded data contribution, a group can be as small or big as an urban scale. MCS network connectivity can be generally classified into three types: infrastructure based, ad hoc and hybrid. The second strengthens the existing infrastructure i.e. cellular, 3G, access points while the second creates opportunistic networks[19]. In addition, as discussed in Section 2.2.2, incentives for user participation, used for crowd perception, include monetary, ethical, entertainment, service delivery, privacy protection, etc. It may contain. Crowdsourced data manipulation. Human involvement in the community detection brings unnecessary, low quality or even false data to MCS systems, and therefore data selection is often required to improve data quality. Temporarily, MCS systems often use heterogeneous devices, and some of these devices have limited computing resources, while others have more knowledge. Thus, two different methods of data processing occur in MCS; The central method sends the aggregated data to a backend server to process an information; Whereas the successful method equips the device with its ability to process data[20]. Recently, some studies have attempted to establish a balance between them based on hybrid methods [5] [6]. Crowd intelligence extraction and use three basic types of crowd intelligence: user, environment and social awareness. The learned crowd intelligence can be used by authorities, public institutions and ordinary citizens, for example in different areas of practice such as public health, urban planning and environmental protection. We also achieve two main objectives of application use of MCS: decision making, visualization and sharing. To make decisions based on learned knowledge such as object classification / recognition, event prediction / repetition, policy making, or suggestion of a tip. Information collected for visualization and sharing is visualized and shared among citizens.

Millions of mobile devices have been "deployed in the field" to those who carry the devices anywhere and anywhere they go and what they do. By utilizing these devices, it is possible to build large-scale sensing applications efficiently (cost and time). For example, instead of installing a roadside camera and a coil detector, we can collect data and use to detect the level of traffic congestion that is carried by the driver to the smart phone. Such a solution reduces the deployment cost of specialized sensing infrastructure.

The set of mobile devices and the data needed for dynamic conditional reuse across the MCS are different from those of the traditional sensor networks used in the past. In the MCS, the mobile device population, each of the sensor data can be generated in the type, and the quality in terms of accuracy, latency, confidence can be changed all the time due to the mobility of the device in their energy Level and communication channel changes, and the device owner's preferences.

Determining the right group of devices can produce the desired data and instruct them to feel that having the proper parameters to ensure the required quality is a complex problem. In traditional sensor networks, population and data can be generated much of them previously known, thus controlling the quality of the data much easier.

The same sensor used for data has been aimed at many different applications of existing MCS. For example, accelerometer readings have been found in transport mode for identification, pothole detection and extraction of human activity patterns. The key to effectively supporting multiple concurrent applications is the need to identify common data for identity and to support the reuse of sensor data across applications. In contrast, previous sensor networks for single-purpose applications are typically expected and re-use seldom require different purposes.

As the equipment individual industrial and commercial households and users to carry, humans are usually involved in the cycle. On the one hand, human intelligence and mobility can be leveraged to help collect higher-quality applications or complex data semantics that may require complex hardware and software otherwise. For example, humans can identify easy-to-obtain on-street parking spaces and report them in graphic or textual information; not just the ultrasound scan system requires special hardware, but complex processing algorithms also ensure data reliability. On the other hand, people naturally have concerns and employee privacy preferences are not necessarily aligned with the ultimate goal of MCS applications. Users may not want to share sensor data that contains or displays private and sensitive information about their current location:

Another important implication of human participation is that. Participating individuals (devices) can incur energy, monetary costs, or even explicit efforts at the owner of a communication device for sensing, processing, and required data. Unless there is a strong enough incentive mechanism, the owners may not be willing to contribute their own resources. For MCS applications to be successful there must be an appropriate mechanism to motivate the recruitment, attraction and retention of participants. Other incentives and people-oriented tools are developed beyond the scope of this article in ESTA because our focus is on system challenges.

2.3 Types of mobile crowd sensing

Five types of crowdsensing can be distinguished on the basis of two criteria. The criteria are: (1) the crowdsensing process and (2) the measurement of the type of phenomena involved by the user. On the basis of the first criterion, we can distinguish crowdsensing and chances of crowdsensing. In crowdsensing, the user of the sensing device sends the sensor data too actively to the server. In opportunistic crowdsensing, the sending of information is automatic, with minimal user intervention. On the basis of the second criteria, we can distinguish between three types of crowdsensing: i.e., environment, infrastructure, and social crowdsensing. Environmental crowdsensing is used to measure the local environment (e.g., level, air pollution, and wildfire). The infrastructure is used for measuring crowdsensing Table III shows the typology of crowdsensing described in this section.

Criterion	Involvement of the user in the crowdsensing process	Type of measured phenomenon
Types of crowdsensing		➢ Environmental crowdsensing ➢ Infrastructure crowdsensing ➢ Social crowdsensing

Table 3 :Types of crowdsensing

2.4. Mobile Crowdsensing Application

In this section we describe the existing mobile crowd sensing application. Based on the types of event that being measured MCS application can be classified as environmental, infrastructure and social, however there are sub classifications such as public safety, health care transportation and traffic planning, environmental monitoring, location service and mobile social recommendation system[3][1][12]. The environmental MCS applications are used to monitor natural environment event such as water level in the creeks, air quality. This application namely common Sense (for air quality) and creek watch (for water level in creek)[2]. Infrastructure application related to public infrastructure such as measuring traffic congestion in the road, parking availability in the city and malfunctioning public services (e.g. broken traffic light). These applications are namely cartel is an application installed with special devices on the car to measure the speed and locations of cars and transmit the measured values by using public WiFi hotspots to a server. The last and the third categories of mobile sensing application is social which an individual can share sensed information among themselves (i.e. to share and compare individual exercise level information of each day among themselves and community to improve their daily excises). The examples of this applications are BikNet[21] which used to measure individual location ,bike road quality (content of CO_2). In the following section we will introduce and discuss a series of representative applications, with the summarized table in Table IV.

2.4.1. Public safety application

Public safety refers to the discovery or protection of a social or natural event such as a crime or disaster that may risk the safety of a general citizen. Crime prevention and investigation. Recently, user provided data has been used to prevent crime[22]. For example, geo-social networks collect data related to crime and suggest safety information that can assess user safety based on spatial and temporal dimensions[9]. Similarly, by analyzing a large number of geotagged Twitter messages from mobile devices, Lee et al. proposed a method of detecting unusually crowded places such as terrorist activities[23]. The spread of technology from security cameras to smartphones in each pocket has proven to help criminal investigations, photos and videos taken by spectators after the explosion have been used for evidence[24]. Mobile based participatory sensing systems can also be used to assist in disaster relief[25]. By analyzing the large-scale mobile phone user data before and after the shock behavior, a model can be established to predict the community's response to future disasters.

2.4.2. Health care

With the arrival of social aging, health care is becoming an increasingly important challenge. Based on the rich data collected from the MCS system, some health monitoring and management services can be implemented. Public health monitoring. MCS can easily monitor the outbreak of the disease. For example, by investigating a health related search query that can be localized through its IP address, Google researchers can estimate the level of flu like disease in the United States Similar to influenza, malaria is another public health problem, and human movement may also affect its spread[26]. Kenya used large-scale mobile location data and malaria prevalence information to study its implicit linkages[26]. Personal health management. MCS also promotes personal health management by recording the user's daily activity trajectory. Dietary habits are considered to be factors of many chronic diseases. DietSense[27] system allows people to shoot and share their dietary choices and get online experts to lose weight advice.

2.4.3. Traffic monitoring and Transportation planning

Urban population data can be used for traffic forecasting, public transport system design, and tourism planning. Traffic dynamics Some studies use GPS equipment for large scale data on vehicles and mobile phones to investigate traffic dynamics. By using data from buses, taxis and cell phones, the real time project[28] can report real-time city dynamics. Similarly, GPS equipment for taxi data and a combination of smart card records from the bus[9]. VTrack[29] is a system that uses mobile phones to accurately estimate the traffic time between different sites. SmartTrace[30] to help determine the mode of transport or given trajectory of the popular planning. SignalGuru[31] a new software service that relies solely on the collection of mobile phones to detect and predict the traffic signal schedule so that the driver can adjust the speed and avoid a complete stop. Road condition. The state of the road for example, traffic flow, the number and speed of icy or bumpy roads is important for vehicle traffic in our daily life.

2.4.4. Environmental monitoring

MCS's participation and mobility provide a new approach for environmental monitoring, such as nature conservation and pollution measurement. Nature Conservation Many scientific research is based on a large amount of data collected from the real world. In the past few years, people have begun to study people's mobile phones, for a variety of scientific research to provide data. For example, with the help of life, cited blood pressure, hearing loss, irritability, etc. The MIT Owl[32] project uses a sensor equipped smartphone network to study owl populations. Scientists also use citizens to collect data to study the effects of climate change, such as the link between temperature rise and the time of plant specific events like the appearance of the first leaf, the results. We believe that the view of large scale citizenship in nature conservation is becoming a reality.

2.4.5. Crowd induction enabled location service

Location is still the most successful and widely used context in everyday use. User friendly awareness supports many popular and emerging mobile applications, including location search, location based advertising. However, these studies rely mainly on personal data, and population data may lead to a variety of new location-based services[13]. For example, CrowdSense at Place [33]is a framework that utilizes opportunistic capture of images and audio clips from smartphone crowdsourcing, connecting location visits with local categories such as shops, restaurants. Numerous user location history has also been used to discover interesting locations, which can help users understand a strange city in a short time.

2.4.6 Mobile Social Advice

Based on the wealth of data collected from the MCS system, some mobile social referral technologies and services can be enabled, including location or friend recommendations, route planning and service or event recommendations.

Local recommendation: Some research focuses on the provision of personalized local recommendations by exploring mobile population perception data. One way to do this is to use the historical location path recorded by the mobile device. For example, by understanding the position history of different individuals, GeoLife[34]measured the similarity between users, based on this provides a personalized recommendation service.

Travel planning: Unlike the local recommendation, the route plan can recommend travel routes to the visitor under given constraints such as time budget, user preference. For example, a travel referral service suggests travel sequences under given space constraints. The travel package recommendation system to help users develop travel plans by using LBSN data. It can generate personalized travel packages by considering user preferences such as travel time and start location.

Service or event suggestion. Another research trend is to provide adaptive services and activity recommendations for mobile users.

2.4.7. Urban dynamics perception

The understanding of urban dynamics is essential to sustainable urban development and to improve the quality of life of citizens in terms of convenience, comfort, safety and safety. However, understanding the dynamics of the city is an increasingly important challenge.

Human urban mobility behavior patterns: an application called EmotionSense[35] was developed under the project to identify the emotional, behavioral, and social signals of the user based on which real-time mood. The table IV blow shows that the summery of MCS application and its function.

Application Type	**Function**	**Typical Applications Used**
Public safety	Crime prevention	crimeSuggest
Healthcare	Personal wellbeing	DietSense
	Public health	Hmonitor
Transportation and traffic planning	Traffic dynamics	VTrack
	Public transportation	SignalGuru
	Road condition	Nericell
	Individual travel planning	VTrack ,SmartTrace, SingalGuru
Environment monitoring	Air pollution measurement	BikeNet
	Noise pollution measurement	NoiseTube
	Nature preservation	MIT Owl
Location services	Interesting location discovery	GeoLife
	Logical localization	CrowdSenseatPlace

Mobile social recommendation	Service/activity recommendation	T-Finder
	Place recommendation	GeoLife
Urban dynamics sensing	Human urban mobility/ behavior patterns	UBhave
	Urban social events	mHealth

Table 4:summery of MCS application and its function.

2.5 Data management system in MCS

Mobile devices have a variety of sensors, such as GPS, accelerometers, microphones and cameras. The operating system allows the application to access the sensor and extract the raw sensor data from it. However, depending on the nature of the raw data and the needs of the application, the physical readings from the sensor may not be suitable for direct consumption of the application. In many cases, some local analysis of some raw processing needs to be performed on the raw data on the device.

They produce intermediate results that are sent to the back-end for further processing and consumption. For example, in potholes detection [5] applications, local analysis computes spikes from 3-axis acceleration sensor data to determine potential cavities. The motivation for such a local analysis is twofold. First, the type of processing performed results in appropriately aggregated data and therefore consumes less energy and bandwidth than transmitting the original sensor readings. This is a well-known trade-off in conventional particle-level sensor networks: the use of computations to conserve energy/bandwidth. Second, it reduces the amount of processing that the backend must perform.

In addition, if a mobile device in a social-scale deployment transmits raw sensor data, the backend may be vulnerable to flooding. Finally, some applications are delay sensitive, and it may be time consuming to transmit raw sensor data on intermittently connected channels as compared to sensor data that is sent to the process. The main challenge in local analysis is to find heuristics and design algorithms to achieve the desired functionality. One type of function is data mediation, such as outliers filtering, noise elimination, or padding data gaps. For example, GPS the collected samples may be inaccurate or missing (due to a lack of sight), in which case an outlier or missing sample needs to be eliminated.

The desired heuristic algorithms and algorithms can be quite specific for the application. Thus, the exact algorithm used for contextual inference depends on the nature of the application and the nature of the context. The current approach is to develop analytics for only one application. This can lead to an "explosion" of the analysis when many common sense applications coexist. Each analysis works alone, and they may be accessing the same sensor, or they may be involved in similar calculations in their inferences.

2.6. Related Work

This section introduces currently designed models which related to the proposed model in the mobile crowd sensing technology that approaches to overcome the exiting problem.

2.6.1. System architectures of MCS

CAROMM: The authors [36]propose and develop a real time open mobile miner (CAROMM) framework, which provides context awareness to support an active and scalable data collection for mobile crowd perception. CAROMM integrates and correlates sensory data with social relevance data on Twitter and Facebook and provides real time information to mobile users by answering questions about specific places of interest. CAROMM source aware and energy efficient locale

Analyze and process data on the mobile device with content information about them, so reducing the amount of data sent to the cloud, and the amount of energy consumed in mobile devices supports highly accurate data collection compared to continuous sensing and information delivery models. Specifically, mobile data flow mining is used, and resource aware clustering on perceived data is used to identify significant changes in the current work context to reduce the frequency and amount of data transferred to the cloud; Important information will not be lost. The change detection sensitivity can be controlled in the CAROMM frame. Finally, the cost of sending raw sensor data to the cloud at predefined intervals to be processed is then evaluated in terms of the cost of data collection and processing on the mobile devices, followed by cloud delivery, data transfer, energy consumption and accuracy.

MEDUSA: The authors[37] propose an extensible and powerful programming framework for managing high level identification and execution of crowd detection tasks between mobile devices and the cloud. In particular, a high level programming language called MedScript provides abstractions for intermediate steps in the crowd detection task, which will be referred to as stages. The Medusa runtime includes a cloud and mobile device component that supports the following features while minimizing task execution on mobile devices: a)multiple concurrent tasks can be performed by a single worker; a) Each employee can determine the resource usage policy that should be respected by the system in each mobile device, c) The task description may include deadlines because a task instance failed and the collected data are not valid, d) The failure mechanism means that the task fails due to temporary errors (due to battery exhaustion or user related actions) and after certain internal timeouts. If they fail, the steps will be retried, e) the results are returned to the person who has just completed all the defined steps, f) the monetary incentives are provided to motivate the users to participate in the duties, and the reverse incentives are paid to the person claiming the privilege of contributing to certain tasks , G) data privacy and subject anonymity are supported by claimants, although users explicitly allow access to data before uploading to the cloud; Having the ability to choose whether or not to participate in the systems privacy policy and data collection, which provides a detailed description of how the collected data can be used.

Effense: The authors[18] suggest effSense, an energy-efficient and cost-effective data loading framework that targets crowd detection tasks that do not require real-time data loading. EffSense considers both data plan users (mostly concerned with energy consumption) and users outside the data plan (data cost sensitive); Participants use customized loading schemes at predefined and fixed time intervals, called cycles; Participants select the best time and network to load data in each cycle. The EffSense framework enhances the predictability of users' activities and mobility (eg, using a zero-cost network to identify critical events and select the most appropriate installation strategies via locally driven lightweight algorithms for user devices), as a WiFi or opportunistically, a 3G voice call data In addition to adding load power, NDP users to choose a device closer to the server to make data migration to the server with low cost or energy consumption), but a certain time delay between data detection and data.

McSense: Authors offer [1], a mobile visual increment platform to ask about task design and assignment, potential workers, regions and their context to fulfill the task assignment process effectively and efficiently. In particular, McSense estimates the time required for each geographical localization task and the number of workers required to complete it. For this purpose, the information in the user / region profile is used together with the related data for the task. The employees in the past and the past have identified three different task assignment policies i.e. random, participation and innovation. They are part of an incentive mechanism based on monetary rewards, and they do not include a set of potential workers for a specific task for workers who restrict battery resources on their mobile devices. In short, McSense ensures that all relevant socio-technical resources are used more effectively and efficiently by putting task definitions, task assignments and mobile perception in closed captions.

Pick-A-Crowd[38]**:** This framework work for discusses a push-based software architecture for efficient assignment of tasks to potential workers based on the profiles of workers and tasks. Worker profiles contain information about skills and interests and are based on past social performance information about the worker's preferences and on completion of tasks in previously assigned tasks. In particular, the pages that each user on Facebook likes are stored in an external database and are used to create employee profiles, including interests. Each task is associated with a degree of difficulty estimated on the basis of worker profiles and job descriptions such as performing a score and an Outcome, calculated on the basis of how many workers have successfully succeeded, and performing a specific task based on skills required. The platform contains monetary rewards for completing a specific task and is divided into challenging small tasks that receive higher rewards by dividing the current budget into defined small tasks. The authors describe three task assignment models i.e. category based, expert profiling, semantic based to automatically assign tasks to the most appropriate employees to complete their tasks. The performance of task assignment techniques is compared pull-based techniques.

M2M architecture IoT framework [39]**:** The authors proposed based on oneM2M standard architecture based on IoT MCS framework. In addition, they propose a power-aware mobile application framework that allows the development of adaptive mobile applications that dynamically adjust their behavior based on the battery power and operating environment of the current mobile device. The proposed framework consists of 1: battery and context monitoring engine, 2: an analyzer engine that evaluates the battery and context information for certain rules in order to determine which adaptation characteristics to apply i.e. light, medium and strong adaptive features. And 3: adaptive features, including hardware resource adaptation, software resource adaptation, user feature adaptation, and additional optimization features. Sensor metadata from heterogeneous sources and domains is processed through semantic Web technology to effectively address inherent incompatibilities and allow data, reasoning.

VITA: Vita[40] is a mobile network physical system that supports the efficient development, deployment and management of multi crowd perceived applications and tasks[33. It is a flexible architecture that combines service oriented design principles with resource optimization mechanisms Allowing intelligent task allocation, as well as dynamic collaboration between the mobile device and the cloud computing platform at runtime. Specifically, the introduction of application based service relationship model, efficient and efficient among the user intelligent allocation of personnel based tasks, and mobile devices and cloud computing platform for computing tasks. The system takes into account different parameters and criteria such as computing power, communication capability, remaining battery time, similar number of tasks, remaining tasks, user preferences, requirements, and constraints that users have competed in the past. It is also possible to use social related information to quantify the distance relationship of two entities, physical or virtual in order to facilitate the development of human and computational tasks according to different application scenarios. The authors use two intelligent computing techniques to provide an optimized solution for the task allocation process: genetic algorithms and K-means clustering. Combining service state synchronization mechanisms to handle potential service failures, ensuring service consistency and correctness of data collected. In short, Vita uses a service based computing, cloud computing, and social computing to provide a comprehensive and flexible architecture that supports application developers and users in the context of mobile people's perceptions. Upload. Application of collaboration and deduction of interest information on new information. Data processing can be performed on the cloud.

senseKit: K. Katevasetal[41] proposed design senseKit mobile sensing framework that present an efficient client server and open source compatible with both android and IOS mobile devices. This model is capable of continuous sensing the devices location GPS, motion accelerometer, magnetometer and others such as Bluetooth. The data are saved to devices memory temporally and transmitted to web server for further analysis.

mHealth: Novak,Crlson and Jarzabek[42] also developed a model called extensible mobile sensing platform Telemedicine and mHealth which installs context sensing flug-ins on appropriate device that gather a wealth of data about users physical and mental states. This framework gathers information concern to missed, outgoing and incoming calls; sound pressure levels; heart rate and light sensor values; etc.

Hariri et al[43] developed mobile sensing framework that utilizes the embedded sensors in smart phones to capture easily context information in urban area. They also designed architecture that facilitates silent mobile sensing in addition to centralized context data analytics and visualization discussed potential application scenarios that can use the proposed framework to develop mobile solutions for smart cities.

Bajwa[6]proposed to hold the entire existing difficulty in distributed environment. Middleware is about integration and interoperability of applications and services running on heterogeneous computing and communication devices. The services it gives such as identification, authorization, authentication, certification and security are used in high range of all appliances and systems, from smart cards and wireless devices to mobile services and e-Commerce.

Vakintis[44] Presented a web platform which is interfaced with the real world during the sensors of various mobile devices in order to cluster and graphically present the recover data based on statistical processing. The platform consists of two components: the client part, work for users to collect sensor data and transmit; the other one is server, which processing, analyzing and visualizing the data in the cloud side. This platform used to collect the data form Internet of things equipment.

Wang et al[20] proposed worker selection framework, called WSelector, to select appropriate workers by taking various contexts into worker history account. They approached it first by providing programming time support to help task worker context constraints. Then the system adopts a two phase procedure to select workers who does not qualified but also more likely to accept a crowd sensing assignment. First it distinguishes workers who please predefined constraints. In the next phase, by forcing the worker’s past participation record, in advance it selects those who are more possibly to accept a crowd sensing job based on a case based logic algorithm. The table V blow shows the summery of related works and our proposed model for mobile crowd sensing.

Authors	Proposed frameworks	Function	Limitation
Sherchan[37] et al.	**CAROMM**	An active and scalable data collection for mobile crowd perception.	Works only on social network ,such as Facebook and twitter
La Porta et al.[38]	**MEDUSA**	Extensible and powerful programming framework for managing high level identification and execution of crowd detection tasks between mobile devices and cloud.	Client-server collaboration not support D2D cooperation
Wang et al.[19]	**Effense**	An energy-efficient and cost-effective data loading framework that targets crowd detection tasks	Do not require real-time data loading
Talasila et al.[2]	**McSense**	Mobile visual increment platform to workers their context assignment process from client to server.	Do not include a set restrict resources on mobile devices.
Difallah et al.[39]	**Pick-A-Crowd**	Push based software architecture for efficient assignment of tasks to potential workers based on the profiles of workers and tasks.	Works only on social network ,such as Facebook and twitter
Datta et al.[40]	**oneM2M**	power aware mobile application framework that allows the development of adaptive mobile applications that dynamically adjust their behavior based on the battery power and operating environment of the current mobile device.	Does not consider the data cost when uploading to the cloud
Chu et al.[41]	**VITA**	service based computing, cloud computing, and social computing to provide a complete and flexible architecture that supports application developers and users in the context of mobile people perceptions.	Data processing can be performed on the cloud rather than D2D
Katevas et al.[42]	**senseKit**	capable of continuous sensing the devices location GPS, motion accelerometer, magnetometer and others such as Bluetooth	The sensed data processed at cloud

Novak et al.[43]	**mHealth**	installs context sensing flug-ins on appropriate device that gather a wealth of data about users physical and mental states	The sensed data processed at cloud server
Hariri et al[45]	**Mobile sensing framework**	centralized context data analytics and visualization	The sensed data processed at cloud server
Bajwa et al.[5]	**Middleware Framework for MobileComputing**	integration and interoperability of applications and services running on heterogeneous computing	Application interoperability but not upload data optimization
Vakintis et al. [46]	**Middleware Platform MCS with Html5**	work for users to collect sensor data and transmit; server side, processing, analyzing	Processing data after uploaded
Wang et al[21]	**WSelector**	forcing the worker's past participation to record crowd information.	Not designed to minimize the data cost
Adugna et al.	**MINEMCS**	-A model to reduce load in mobile crowdsensing networks and data cost to the clients by grouping the mobile nodes to aggregate data before upload -Support D2D collaboration to upload data and to optimize resource utilization ,such as channel bandwidth, battery consumption, and storage space.	- Does not support data replication and Distributed caching - Does not consider location optimization -Does not include application development

Table 5:The summery of related works and our proposed model

2.7. Summary

Mobile crowd sensing is different from traditional wireless sensor networks. Compared to traditional wireless sensor network MCS has a lot of unique characters. such as Multimodal sensing, Hybrid Human Machine Systems, mobile device used as sensing device and low cost to deployment. today there are many applications used in MCS that solve the real world problems in field of Crime prevention, Public health, traffic dynamics, Public transportation, Road condition, Individual travel planning, Air pollution measurement, nature preservation service and Place recommendation also in any social events. On another hand depending on the nature of the raw data and the needs of the application, data handling and management is the challenging issues to the existing MCS application and architectures. Numbers of authors proposed their frameworks such as CAROMM, MEDUSA, Effense, McSense, Pick-A-Crowd, oneM2M, VITA etc. for mobile crowd sensing system with interesting function and unsolved challenges. Thus works are targeted to optimize and develop compatible platform for sensing devices, provide to the user to gather information with the built-in sensors that found in the mobile devices, finally the data processed and uploaded based on the question like what, who and how to use mobile sensing system. However, there are challenging issues that are still not solved in the existing system such as data uploading problem with narrow channel bandwidth and storage space limitation in the cloud server. We also included our proposed model in the summery part of the related work in table V.

CHAPTER THREE

DESIGN FOR MINEMCS MODEL

3.1 Overview

Numerous mobile Sensing Frameworks are proposed for mobile sensing infrastructures. However, this architectures scenario for MCS follows application silo approach (i.e. applications are built independent from each other). In recently architecture there are two components, sensing devices that receive information form environment and data collector base station and behind base station there are backend servers to processing and analysis of sensor data[2][45]. There are no mechanism or component that faces the common challenges such as resource allocation, data collection and energy conservation. Applications perform sensing and processing activities independently without considering the consequences on each other will result in low efficiency on resource constrained platform. There is a high possibility of replicated data when sensing and processing among multiple applications. For instance, air pollution, noise and traffic sensing needs location information, but this all applications use its own data without considering reusing the same data samples. To the best of our knowledge, the silo approach does not enable to eliminate duplicate sensing data and processing in multiple applications also it is the overhead for the network channel bandwidth when uploading data in to the backend server.

We outline a model to improve the network efficiency framework to implement mobile sensing applications. MINEMCS has several features: (i) capturing different types of streaming data from a mobile device and then, (ii) process, manage and analyses the data and associated contextual information with it; (iii) facilitating the mobile user to analyze the collected data. We modeled the data collection, processing and uploading system for sensed data. The core theoretical contribution of this paper is in the data collection and aggregation in mobile sensing information with the novel client server model.

In the following sections, we describe the details of the techniques and models developed with respect to the proposed solution. In Section 4.2, we describe and present our proposed architecture for MINEMCS data management between mobile nodes and base station. The proposed architecture modeling approach in the section 4.3, in this Section, we describe a cluster head selection algorithm, intra-cluster communication, outside cluster communication, transmission phase, transmission creation schedule and data transmission over wireless channel. Finally, in the section 4.3 we will summarize this chapters Proposed architecture for MINEMCS

3.1. Proposed architecture for MINEMCS

MINEMCS is mobile crowd sensing platform that allows users to collect different types of mobile phone sensing data form user cell phones. The interconnected components in our mobile crowd sensing architecture are as the following in fig 3

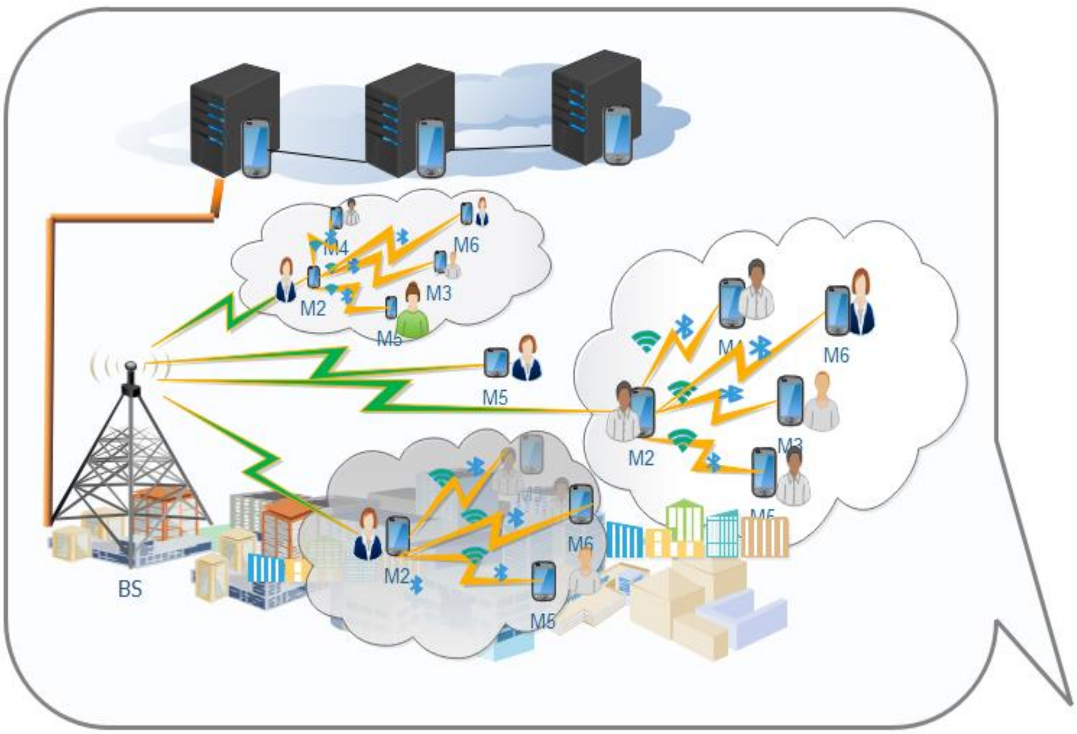


Figure 3: Proposed architecture for MINEMCS

Cloud server: - is distributed backend server that receives and transmits data to the group of sensor nodes. It is a virtual server running in a cloud computing environment. This is why cloud servers are often referred to as virtual private servers. Although each cloud server can be called a virtual private server. This is because a virtual private server can only be placed on a single hardware server, resulting in a single point of failure in any of its hardware failures, and server runs in a software independent method. This means that the cloud server has all the software needed to run without relying on any centralized installation of the software.

Smart phone/client: -the collection or group of nodes that interested to collect sensing data by using the phone sensing components. This phones are not the same by their storage, battery and even sensor components such as quality of camera, our system select the group head nodes by their battery capacity.

Sensing components: -are the collections of sensing applications that found in the smart phone such as camera, Bluetooth, accelerator, microphone and so on.

Base station: - is a fixed communications location and is part of a wireless telephone network system. It transmits information to and from a transmitting/receiving part, such as a mobile phone.

Communication channel: Switching a sensor node to a network requires a device to send and receive information over a wireless channel.

3.3 Proposed architecture modeling approach

In the proposed architecture approach, smart phone sensing is defined sensor data to achieve the following requirements.

- ✓ To minimize the complexity of sensed data and mobile node management, the system able to select nodes by their event condition, event location, and based by data attributes.
- ✓ Sensor nodes are mobiles; they use the random way point mobility model.
- ✓ The base station is static.
- ✓ The nodes are unaware of the location, that is, they may not be equipped with GPS modules, or they may not use any positioning mechanism.
- ✓ To reduce the conception of storage devices, channel bandwidth, traffic and process load the system should intercommunicate with nodes as data requirements.

The mobile devices send query concerned with certain requirements for sensing purpose such as location information to the server then the server sends response to the mobile phone of each nodes to determines either it involves in the sensing or not which is used to minimize the chance to upload information to server.

The cooperation between mobile node and server is achieved by communication infrastructure such as 3G and the collaboration between mobile nodes achieved by WIFI direct or Bluetooth.

The server side program provides function for processing uploaded data, coordinating every mobile node, distributing requests and selects some nodes based on the processed information to involve in more complex task. The application installed on each mobile node responsible to recognize sensing events concerning to query information and upload the sensed data to the backend server.

3.3. MINEMCS Modules architecture

The application silo architecture lacks device to device communication before uploading data to server, there are some components that we shared form the existing MCS data uploading architecture such device manager, sensor manager, event detector, data collection module and data uploading module. However, we provided in our model device communication, data processing and group leader selection. The architecture consists of two main modules - a data collection and processing module in client side and server in the cloud side. The data collection module captures sensory data, performs local continuous fusion on the data, and uploads the analyzed information to the cloud server, where further analysis, management and fusion of the incoming multiple streams is required. To intelligently send only the analytical information from each cluster head, we use resource aware clustering of sensory data to identify significant changes in the situation. This reduces the frequency and number of data that is transferred from each mobile device to the cloud while ensuring that important information is not lost. The cluster provides well processed data when it wants to send updates to the cloud server. For example, GPS data is not considered sufficiently significant to ensure that minor changes to the update will be ignored, but significant changes can be detected and used to perform the update. The following figure 4 illustrates MINEMCS modules architecture.

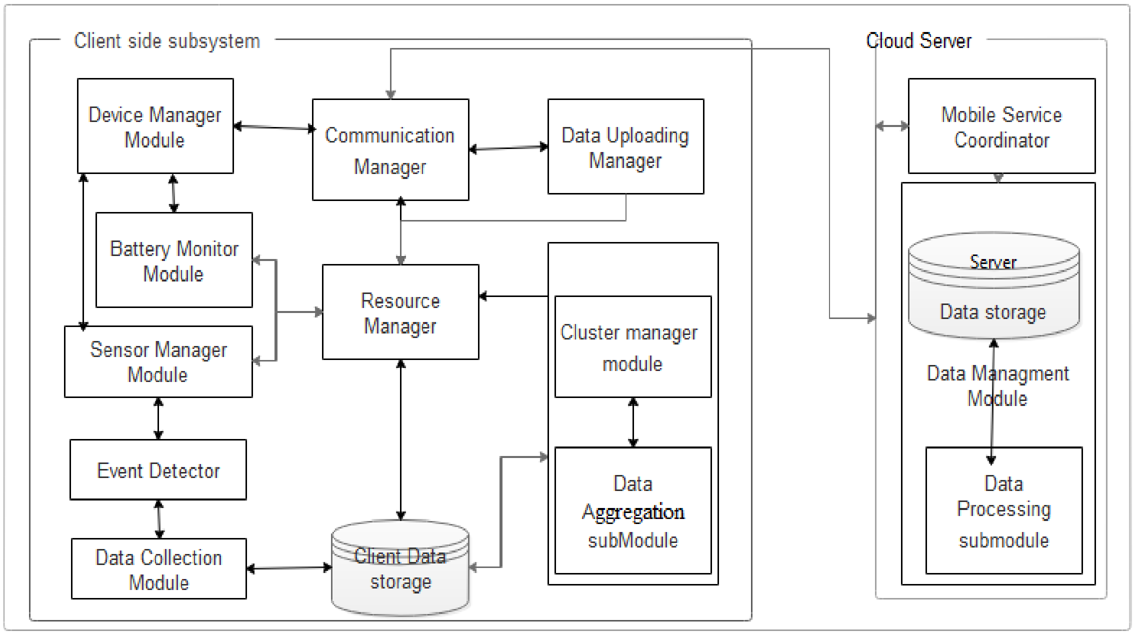


Figure 4:MINEMCS modules architecture

3.3.1. Client side subsystem Modules

3.3.1.1. Device Manager Module

The device manager module is the mobile operating systems that operates on a mobile device and controls over all operation in the mobile devices such as application software program execution, controlling the input data that come from external world and display out from the mobile operation system to the outside the devices and manage graphical user interfaces[46]. The device manager also manages the communication channels and data flow in the mobile phones. In the terms of mobile crowd sensing, this module commands different sensor that applicable with mobile to perform the given tasks.

3.3.1.2. Resource Manager

This is any physical or virtual component of limited availability within mobile phone system[47]. Every device that connected to the mobile system are resources, physical components such as RAM, Storage media, battery etc. and the logical components such as files applications and network connections are the resources of mobile phone. In this module resource manager deny or allow the device sensors weather to sense or not the events in the user environment.

3.3.1.3. Battery Monitor Module

The continuous sensing by using sensor devices can lead to energy consumption problem. if the sensing mechanism is computational intensive the battery will be drain rapidly and the energy spent inside any apps in mobile phone. In this case the objective of saving energy in continuous sensing is to control the events of sensors[48]. On other hand, when the mobile nodes form the cluster the first requirement to select the cluster head is identifying the level of the battery in the group member. Therefore, this module is so important in this module architecture.

3.3.1.4. Sensor manager module

The Sensor Manager is responsible for the sensor connection to the device. It periodically queries the device's sensor data and passes it to the data collection manager. The media manager handles any communication between the data collection module and the camera and microphone of the device[48]. The data collection module has the ability to collect various types of data such as temperature, text and humidity. The data collected from the sensor, i.e., the temperature and location, are passed to the data fusion manager before being uploaded to the cloud.

3.3.1.5. Cluster Manager Module

The cluster manager is a resource that performs dynamic load balancing and improves the availability between connection managers. Each deployment of Connection Manager creates a cluster manager resource and edit the properties of the cluster manager, and define the connection manager as a master node, a slave node, or both. The primary node dispatches traffic to the lower node. A slave node is configured to accept traffic from one or more primary nodes. Define traffic and process data for cluster managers defined as primary and dependent nodes. Each cluster manager has only one master node. To assign traffic to a given network, the messaging service is configured as a master node in the cluster[49]. The following section describes how cluster forms and cluster head selected.

3.3.1.5.1. Clustering Algorithms

Clustering is the process of classifying nodes into different groups by dividing the data sets into a series of subsets called clusters. The clusters may be separated into a high density region or a low density region. The density is considered here as the number of nodes in the neighborhood. Clustering is a technique for achieving high data density. Density based clustering is an efficient clustering method that determines a sufficient number of clusters and provides the appropriate location for any clustering to start.

Proposed algorithm to form cluster for MINEMCS model: **Input**: n Node info, nr - Neighbor radius, mReq - Minimum node required in the cluster. **Output**: Clusters

Algorithm1: Cluster Formation

```
cl=0
for each newNode nP in field {
N = getNeighbors (nP, nr)
If (sizeof (N) < mReq)
Mark nP as single node
Else
++ cl
Mark nP as cMember
Add nP to cluster cl
recurse(N)
}}
```

Algorithm 1:Cluster Formation

In the following section, cluster head selection is based on the algorithm taking into account the residual energy of the node and the node density as a parameter. The entire network is divided into more than one cluster. The main parameters of the residual energy of a node are calculated by means of a first-order radio model. The mobile sensor nodes may evaluate themselves to find the residual energy of their nodes within the cluster. If some nodes have residual energy that is less than the minimum energy, those nodes are removed from the network. The secondary parameters of the node density (the number of cluster members in each cluster) can then be computed within the cluster. The calculated density value and the remaining energy must be broadcast locally to all its neighbors. Thus, the cluster itself chooses the highest residual energy and highest density node as the cluster head of the cluster. By receiving this value, each node can know which node will be the cluster head. After the cluster head has been determined, each cluster head broadcasts the advertisement message to all other nodes using the CSMA MAC protocol and invites other nodes to join the cluster. Other nodes may also participate in the advertising process. Upon hearing these advertisement messages, each sensor node selects the nearest cluster head and registers itself as a cluster member.

- p-actual transmission power
- E-remaining battery power
- D-degree of node

Weight **W**=a*D+b*E+c*P //*where a, b, c are the weighting factors*

Algorithm2: Cluster Head Selection

```
do {
r←random ; //repeat for r rounds
if (mReq (s)>0 & r mod(1/No. of nodes) ≠0){
compute CH(s);
}
for i=0 to k {
NN = Number of nodes;
}
If( W>=expected Level)
CH  =TRUE; node n be a CH
}
else { CH{s}=FALSE; //node n not be a CH
}
if (CH{n}=TRUE)
{      BC (ADV) ← broadcast an advertisement message;
Join(IDi); // non-CH node i join into the closest CH
Cluster(cl); //form a cluster cl
recurse(cl)
}
```

Algorithm 2:Cluster Head Selection

3.3.1.6. Data Collection Manager

The data collection manager performs the task by intermediating between sensor manager module and data storage by accepting *collect data* instruction from resource manager through sensor manager module. After accepting the instruction form sensor manager, it collects the data and forward it to the data storage. The Data Collection Manager also handles timers and asynchronous callbacks from other components.

3.3.1.7. Event Detector

Event detector is part of event listener when data sampled by the sensor that can be used for multiple event feeds. The data is filtered to extract the child feed to be forwarded to the relevant event. The event listener detects that the sensor monitors its received sensor feed event trigger when it sees event occur, it triggers the event to its CH. The event detector does not upload the local record of the trigger event itself; all events are sent to the service cluster head.

3.3.1.8. Client Data Storage

Client data storage is act as temporal place between resource manager, data collection module and data aggregation module by storing the coming data from data collection module and provide the data access to the data aggregation module with the command that come from resource manager that primates either to fetch or not. It also stores the temporal aggregated data that is ready to upload to the cloud server.

3.3.1.9. Data Aggregation Submodule

Data aggregation module located between client data storage and cluster manager module. It performs data processing and analysis task to reduce the cost of uploading data and battery drainage. one of this paper problem solved in this module. it is important part of the module to process and fuse the gathered data with most effective aggregating techniques. Many data aggregation technique is proposed in WSN[50].In generally there are two main types of data aggregation, thus are lossless and lossy. These two methods have their own advantages and disadvantages. Lossless aggregation computes the data without any loss and possible to restore it again but it should forward every node information, for example if there are five nodes in cluster and sense proximity weather light or not, so the answer would be yes or no (1 or 0). If there are light in the cluster, they send 1(there is light) else send 0(no light) in this case lossless aggregation is efficient. But in the case of temperature, humidity, accelerator, magnetic field, level of CO2 gas etc. lossless aggregation is not efficient to reduce upload data cost. On the other hand, lossy aggregation fuses the data by using aggregation functions such as MAX, MIN, COUNT, AVERAGE, MEDIUM and Standard deviation data aggregation techniques. In this method, Hello packets and control packets are not aggregated. Aggregate normal packets. Depending on the application requirements, critical packets can also be aggregated. If the node can take the necessary action in response to the event of interest, the key packet is sent to the cluster head, generating the packet and entering the buffer from the input queue to CH into the next node. The following flow

chart shows the data aggregation in our proposed model.

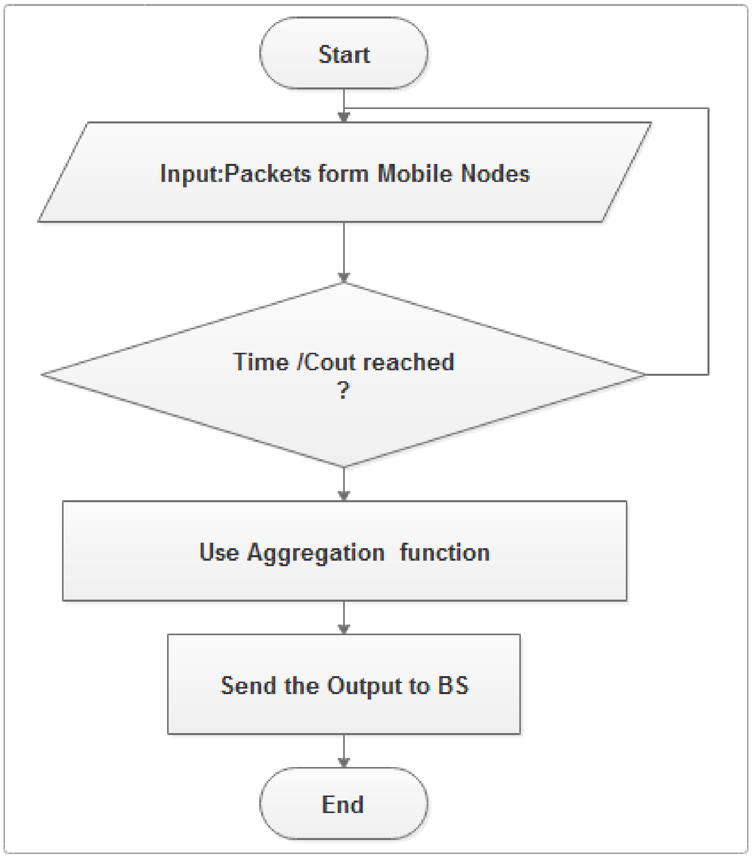


Figure 5:Flow chart for Data aggregation

After the cluster header receives the packet from the input queue, it can perform data packet counting or time-based processing on the data. The system waits until the buffer reaches a specified number of packets based on the count type, where the system waits for a specific amount of time on a time basis that automatically determines the value of the aggregated data based on the type of sensor. In our case, we use the average aggregation because the nodes are in the same environment and send the data to the cluster. So the data is roughly similar. We use our appropriate data aggregation based on lossy data aggregation techniques to present our approach. The calculated average function is used as the sum of the input data I, forming the nodes in the group divided by the total number of nodes in the combined contribution to the cluster head. The following algorithm shows the data aggregation in our model.

***Algorithm3:* Data Aggregation Algorithm**

```
INPUT: packets to CH
{
PUT: packet in Buffer
WAIT: for Count/Time reached
IF (Count/Time Reached)
{
Do: Aggregation
End If
}
}
```

Algorithm 3:Data Aggregation

3.3.1.10. Data Uploading Manager

The Data Upload Manager is responsible for uploading data to the cloud using a mobile device's network connection. The algorithm implemented in the proposed data collection module is presented in pseudo code format in table 7 and 8. The algorithmic data collection includes the functions performed by the data manager, sensor and device manager, and the cloud server manager, initiated by the device manager. The algorithm data analysis and change detection includes the function of performing the aggregated data analysis module. This module uses cloud managers to upload periodic updates of cluster data to the cloud. In addition, the data fusion manager incorporates change detection, i.e. it has the ability to determine significant changes in sensed data. Any significant change in the sensed data causes the data to be uploaded to the cloud. In addition, we also provided single node upload system if there are no neighborhood nodes to make cluster. We have chosen a periodic upload interval to enable data uploads when no changes are detected in the mobile sensing environment.

3.3.1.11. Communication Manager

Connection manager manages mobile phone communication with its cloud server by using WiFi, Bluetooth and 3G cellular network connections. The connection manager module act as communication service provisionary between device manager, resource manager and data uploading manager. After data aggregation finished in the data aggregation module it stores back to client data store and resource manager send request to the connection manager to give permission to the upload manager to upload the data that is ready to be sent to cloud server.

3.3.1.11.1 Intra-device communication

During intra-device communication, each normal node sends information to cluster head. During simulation, it is observed that most packet loss occurs during intra-cluster communication when the normal node attempts to transmit information to its corresponding cluster head, since the cluster head leaves the transmission range of the normal node or the node moves away from the transmission range of the normal node. During this phase, each cluster head collects information from its surrounding nodes associated with the cluster head, and then sends the aggregated data to the final destination discussed in the next section.

3.3.1.11.2 Outside - cluster communication

During inter-cluster communication, the cluster head sends the processed information to their neighboring cluster heads. Outside the cluster, each cluster head sends the processed data to the base station. The base station receives the data and forward it to the backend server for further analysis and for retransmission of the data back to unreached cluster, the data delivered each node either cluster head or directly from base station.

3.3.2. Cloud server Modules

3.3.2.1. Mobile Service Coordinator

To execute a service, the mobile service coordinator sends invitations to mobile phones (contributors) to participate in crowd sensing service. it also sends their parameters advising on how to detect events and construct their messages. Event detection is carried out by mobile phone event detector when listening the sensing devices, which are then sent to the cluster head. An adjusting scheduler preforms sampling of each sensor based on the sensing requirements received from the cluster head for each service being served at the time.

3.3.2.2. Data Management Module

Data management module responsible to coordinate, control and organize the data that uploaded from cluster head and, provide and allocate the storage space for data, where to be saved. The data management module holds server data storage and farther data processing submodules.

3.3.2.2.1 Server Data Storage

Different cluster heads send data form their environments to publish the event in the cloud server and the received that must be stored in the server data storage for further processing. Server data processing submodule responsible to store the different sensors data that to be accessed for redistribution purpose when data sharing is needed.

3.3.2.2.1 Farther Data Processing

The data processed in the client side is belongs to only the cluster members. However, in server side there are data that come from different cluster heads. Therefore, in cloud server it requires farther data processing to republish the fused to the uninformed locations. Farther data processing submodule responsible to aggregate the data come from numbers of cluster heads.

3.4. Transmission phase

After each node determines which cluster it belongs to, it must inform the cluster head node that it will be a member of the cluster. Each node uses the CSMA MAC protocol to send this information back to the cluster head. At this stage, all cluster-head nodes must maintain their receivers.

After receiving all sensed data from a non-CH node, the CH can do data aggregation. In this function, the CH should delete the redundant data and compress the data into a single packet. The packet is transmitted to the base station by multi-hop transmission. Each cluster member will know its respective timeslot. If the time slot is allocated, only the sensed data is transferred to the cluster head, otherwise it goes to sleep. In addition, it sends the value of its remaining power. The cluster head maintains a table of nodes with the highest power recorded in the current round. After it forwards the data to the sink, it selects the node with the largest remaining power as the cluster head of the next round. The algorithm for the transmission phase is explained below.

Algorithm 4: Data Transmission

```
If (CH(s)=TRUE){
Receive(IDi, DataPCK) //receive data from cluster head
TransToBS(IDi, DataPCK); //transmit received data
}
else {
SleepMode(i)=TRUE; //node i at a sleep state
}
```

Algorithm 4:Data Transmission

3.4.1. Schedule creation

The cluster head node receives all messages of the nodes that are to be included in the cluster. Based on the number of nodes in the cluster, the cluster head node creates a TDMA schedule that tells each node when to send. This schedule is broadcast back to the nodes in the cluster.

3.4.2. Routing protocol

Routing protocol allow nodes to select the routes when communicating with each other. It is the formula or algorithm by router to determine the necessary path when data transmitted. The routing protocol enables a network to take dynamic adjustment in challenging conditions. the numbers of routing protocol proposed for wireless sensor network based on the dense sensor node deployment, battery power, resource computation, storage constraints and self-configuration[50]. Based on the objective of our paper, we selected the AD HOC on demand distance vector (AODV) is reactive. The reactive property of the routing protocol suggests that mobile nodes requests a route only when they need to communicate with another mobile nodes and does not require maintaining routes to destinations to when they are not communicating that make it easy and efficient routing protocol which proposed specially for use of multi-hop wireless ad-hoc networks of mobile nodes. It also provides mechanism for self-organizing and self-configuring, needing no existing network arrangement or administration. The network nodes cooperate to forward packets to allow communication over multiple jumps between nodes that do not directly between them wireless transmission range each other. As a node in the network move or join or leave the network, and as a wireless transmission conditions such as interference source changes, all routes are automatically determined and maintained by the AODV routing protocol. Due to the order or sequence of any intermediate hops to be reached the destination may change at any time, resulting from the network topology may be quite rich and rapidly changing. The protocol also supports cellular telephone systems and mobile networks with more than 100 nodes[51].

3.5. Mobility model

The mobility model[52] is an important part of simulation based research on wireless networks. Researchers in the field can choose from the various models developed over the past few decades in the field of mobile computing and wireless communications[53]. A popular and commonly used mobility model is a random road point model[52]. It is implemented in the network simulation tool ns-2 and is used for multiple performance evaluation of ad hoc network protocols. The mobility model is a simple and straightforward stochastic model that describes the mobile behavior of network nodes in a given system area. A node randomly selects the target point called waypoint in the area and moves it to this at a certain speed. Wait for a suspension time, select a new speed and destination move to this destination at a constant speed. The movement of the node from the starting position to the next destination is represented as a movement period[54]. Based on the ingesting futures and compatibility with our system, we selected Random Way Point mobility model to demonstrate the movement of mobile nodes in our paper. The movements of individual nodes can represent the motion of nodes but, there are some circumstances that nodes move together in the group like group of students to learn the same course, group of the same religious and group of football club may move together[53], this needs other mobility model that is called group mobility model. we also used the group mobility model to support the movement by group in our model.

3.6. Security and Privacy Protection

The upcoming technology within mobile devices play an important role in human daily life problem solving. On other hand security and privacy issue is become a critical challenge to contributed mobile crowd sensing data to the publisher. We provided some important requirements to the user when sharing the sensed data over the network. User should continue to sensing process without disclosing their identity such as user name, email address and mobile device identity information such as IMSI (International Mobile Subscriber Identity) and the communication stablished by the trust of each other nodes.

3.7. Summary

The existing mobile crowdsensing data collection platform is follows the application silo approach. This lacks the awareness of the transmission bandwidth, delay and packet loss, we proposed NEIMCS model to overcome this problem. In this model we discussed the issues like proposed model architecture and its module interconnection. Also we presented Crowd detection approach, cluster head selection algorithm, and the mobile nodes communicate with the cluster head, each normal node sends information to its cluster headers and outside of the cluster, the base station receive the data and forward it to the backend server for further analysis and for retransmission of the data back to unreached cluster**.** Transmission phase *of* each node uses the CSMA MAC protocol to send this information back to the cluster head. TDMA schedule is fixed, to transmit data. we provided the mechanism how data collected, processed and uploaded with the cluster head. Finally, AODV routing protocol selected to forward the packet. The demonstration part the system conducted in the next chapter.

CHAPTER FOUR

PROTOTYPE IMPLEMENTATION AND EVALUATION

4.1 Overview

In this chapter, we present a performance evaluation of MCS in terms of data fusion. In the first Simulation scenarios and models are discussed; simulation setup and Next, the movement of the node having the traffic model will be discussed. Then the details Simulation results and decision based on the result will be discussed.

4.2 Development and Simulation Tools

The simulation of the system[55] is the process of the system model. The model can be reconfigured and tested; typically, it is difficult, too expensive, or infeasible in a representative system. The process of the model can be studied, so that the behavior of the actual system or its subsystems can be determined. On the other hand, simulation is a tool for assessing the performance of systems (existing or suggested) under different configurations as well as long real-time problems. Use simulation before the existing system is changed or set up a new system to reduce the likelihood of failure to reach specifications, eliminate unforeseen bottlenecks, prevent resource deficiencies or overuse, and optimize system performance. Computer simulations use the same concepts as any simulation, but it needs to be modeled by computer programming.

Fundamentally, there are three types of simulations[56], which are Monte Carlo, continuous and discrete event simulations. Monte Carlo simulation by John van Neumann and Stanislaw reflects its gambling similarity, using the uncertainty model. The expression of time is not necessary.

Continuous simulation using an equivalent model, usually a physical system, where do not describe the exact time and state relationships that lead to discontinuities. The purpose of the simulation study using this model does not need to be clear indicates the state and time relationship.

Discrete event simulations utilize physical mathematical or logical models describe the state change system at the exact point of the simulation time. The nature of the change of the state and the time task of the change accurate description. Discrete event simulations include some components[57]. This simulation can be listed pending events can be simulated by routines. A global variable that describes the state of the system can be represented simulation time, which allows the scheduler to predict this time in advance. The simulation includes input procedures, outputs Procedures, initial procedures, and trace procedures. In addition, this simulation provides dynamic memory management, also it adds a new entity to the model and delete the old entity. As a result, debugger breakpoints are provided in discrete event simulations the user can check the code step by step without interrupting the program. There are several existing discrete event network simulators like OMNET++, NS2, NS3, QUALNET, OPNET etc.[58].

4.2.1. Network Simulator Version 2

Among them, based on our objective of study and specification requirement, we selected NS2 simulator tool to demonstrate our model. It is a discrete event network simulator for network research simulation. NS2 provides a complete development environment for performance evaluation of communication networks and distributed systems. It provides extensive support for simulating TCP, UDP, FTP, HTTP and DSR protocols[59]. NS2 supports two languages, the system programming language C + + details Implementation and scripting language TCL are configured and experimented with different parameters rapidly. NS-2 has all the necessary functions such as, visualization, abstract, simulation and traffic and scene generation[58].

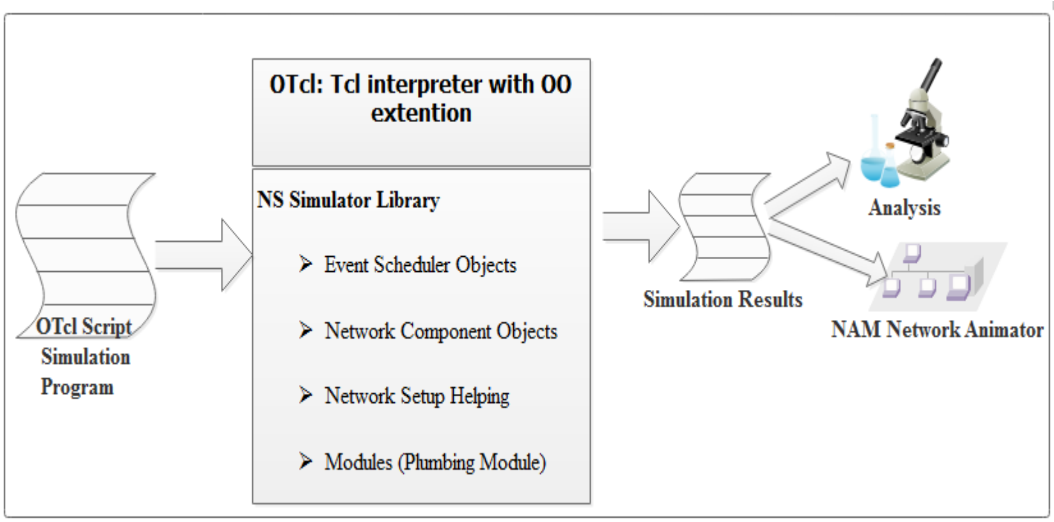


Figure 6:simplified Users View of NS2[60]

4.2.1 Simulation Environment

The simulations have been conducted on Toshiba Notebook R830 Intel(R) Core(TM) i5-2520 CPU @2.50GHz, 4GB of RAM running Linux mint 18.1. NS2 version 2.29 has been used for implementing. The XGraph and AWK used for analyzing their performance with respect graph and average output. Basically, NS2 is chosen because it is the most popular simulator in an area with rich functionality, and most of the related works are done using this tool.

4.2.2. Simulation setup

The Simulation setting sated as, 20 to100 mobile sensor nodes are randomly arranged in the sensor field 200m × 200m and the maximum number of nodes in the cluster is 10, that means, one for cluster head and the remaining nine is for cluster member. The size of each packet is set to 512 bytes. The time for transmitting such a packet is considered to be one unit of delay. The simulation time for each scene was set to 500 seconds, and a repetitive simulation was performed on each of the scenes to verify the reliability of the results. Nodes are inherently mobile, and they can correspond directly to sink node. The initial energy of the sensor node is 2 joules, and the energy of the base station node is infinite. The other parameters of the network environment are shown in Table 5

PARAMETER	**VALUE**
Devices	Smartphone,
Antenna Model	Omni Antenna
Radio Propagation Model	Two Way Ground
Network type	Bluetooth ,cellular communication
Communication model	Bi-directional
Number of Nodes	20 to 100
Simulation time	10 to 300 seconds
Initial Energy	2 Joule

Network size	1000X1000m^2
Data Packet size	512 Bytes
Average user velocity	Distributed uniformly in 1 and 1.5 m/s
MAC Type	Mac/802.11
Mobility Model	Random Way Point and RPGM
Routing protocol	AODV

Table 6 : Simulation setup

4.3. Performance Evaluation Metrics

To compare the scalability and performance optimization of the MINEMCS with AS raw data uploading, quantitative procedures are used to measure and evaluate the performance of the simulation. A set of performance metrics used to compare this proposed model is shown in Figure 7. Each of these metric parameters can be briefly described as follows[61][62][63] with set of performance indicators.

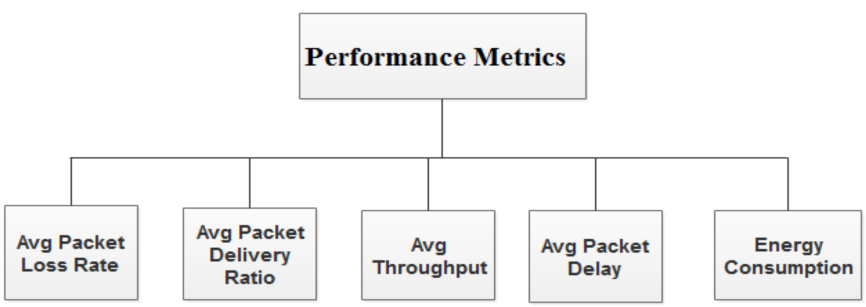


Figure 7: Performance Evaluation Metrics

4.3.1 Average Throughput

Throughput, in general term it is maximum rate of production or process form the given inputs. In terms of computer network, it is the rate of successful delivery of packet over network channel. It is measured by ratio of the total packet received from each node divided by total duration of simulation. It can be expressed as a mathematical equation 1.

$$(1)\textbf{Throughput(bit/sec)} = \frac{\textbf{No. of Deliveried pkt} * \textbf{pkt size}}{\textbf{Total duration fo simulation}}$$

Equation 1:Average Throughput:

4.3.2 Average packet delay

Average packet delay of data packet is the average delay between the sending of data packets by the source and the reception of the data packets at the corresponding receiver, including delays due to route acquisition, buffering and intermediate node processing, If the value of the end-to-end delay is high, this means that the protocol performance is poor due to network congestion. The average delay can be calculated by the following equation 2.

$$(2)\textbf{Avrg pkt delay} = \sum_{k=0}^{n} \frac{\textbf{timePktRecieved} - \textbf{timePktSent}}{\textbf{TotalNo. of Pkts Recieved}}$$

Equation 2: Average packet delay

4.3.3 Average Packet Loss Rate

Packet Loss Transfer Rate indicates the ratio between the number of packets sent by the source and the number of packets received by the receiver. The average packet transfer rate is calculated from the following equation 3.

$$(3)\textbf{Avrg Packet loss} = \sum_{i=0}^{n} \frac{No. of\ pktSent - pktRcv}{time}$$

Equation 3:Average Packet Loss Rate

4.3.4 Average Packet Delivery Ratio

The average packet delivery ratio is the ratio of number of packets that are successfully delivered to a destination calculated to the total number of packets that have been sent is show as represented in the equation 4.

$$(4) \boldsymbol{Avg. Packet\ Delivery\ Ratio} = \sum \frac{\textbf{Number of packet receive}}{\textbf{Number of packet sent}}$$

Equation 4: Average Packet Delivery Ratio

4.3.5 Energy Consumption

The average energy consumption calculated as the initial battery energy before sensing, processing and uploading in the mobile phone minus the remaining amount of battery level after sensing, processing and uploading data to up level data publisher divided to total time of simulation. The mathematical equation 5 shows how energy calculated.

$$(5) \boldsymbol{Energy\ Consumption} = \sum \frac{\textbf{Initial}_{\textbf{enargy}} - \textbf{Remain}_{\textbf{energy}}}{\textbf{Total Simulation Time}}$$

Equation 5:Energy Consumption

The metric indicates the effort required to lose the ratio and receive data from the routing protocol. If the two data are equal, the ratio should equal 1, which is called the ideal ratio. If the ratio is lower than the ideal ratio, it may be an indication of some of the faults in the protocol. Conversely, if the ratio is higher than the ideal ratio, it instructs the sink to receive more than one data packet called a redundant packet. This is detrimental because receiving duplicate packets will consume more energy[64].

4.4. Experimentation

The key challenge of mobile crowd sensing is the ability to efficiently collect environmental data from multiple sources over a long period of time. The term cost efficiency is used to represent the size of channel used to collect, analyze, and upload data and bandwidth usage when uploading data and data accuracy. A good way to save bandwidth is to reduce the number of uploads, as research has shown that communication is a major factor affecting battery use. On the other hand, reducing data uploads can result in loss of data accuracy. As a result, the data collection module must balance these two factors to ensure high data accuracy while reducing bandwidth usage. In this section, we evaluate and validate the ability of the data collection module to perform mobile crowd sensing, thereby reducing bandwidth usage though keeping high levels of accuracy. We conducted a detailed experiment involving data collection of users over several time under different environmental conditions to verify the effectiveness of the proposed data collection module. We have simulated our work for ten minutes' experiment with thirty-one mobile nodes and one Omni directional Antenna. The mobile nodes that are running in our simulator have the same type of operating system (i.e. Android SDK v2.2 and above) .

4.4.1. Approaches to evaluate our work

In this section, we develop two data collection approaches for mobile crowdsensing:

scenario 1: all data aggregations done in the cloud: In this approach, the mobile devices sense context data occasionally and upload to the cloud. There is no data fusion is done on the device level.

scenario 2: in this approach, the collected mobile data analytics committed on the client side, each mobile device performs continuous sensing and one of the mobile device that selected as group leader process data aggregation task and uploaded to the cloud. The goal of this scenario is to reduce data redundancy in backend server and to optimize channel bandwidth in data transmission form client mobile to the server. In this approach we develop two evaluation metrics. They are a data transmission metrics and an energy usage metrics. These metrics proposed to compare the metrics concerned to the above two data collection approaches.

The results of the experimental evaluation are divided into two pats. In the first part, we provide experimental results for observing average packet delay, average packet loss and throughput of bandwidth usage. In the second part, we present the results of using the data accuracy of the proposed data collection module. The results of the proposed data collection module *Approach 2* (we use the term MINEMCS data collection) and the continuous data collection Approach 1 (we use the term *application silo* collection to represent data collection on mobile devices without any device-based processing). Data collected from both methods is uploaded to the cloud at successive intervals.

Table 10 gives the parameters used for the experiment. These experiments were performed on the same device, and the heterogeneous devices however they must support android, i.e. the devices that performed the original collection, and the devices running the MINEMCS data collection were not the same. In most cases, one of the devices is a cell phone and the other can be a tablet. Continuous data collection scenario (we use the term Application silo (AS) raw data collection to represent data collection on mobile devices without any device-based processing). Data collected from both scenarios are uploaded to the cloud at successive intervals.

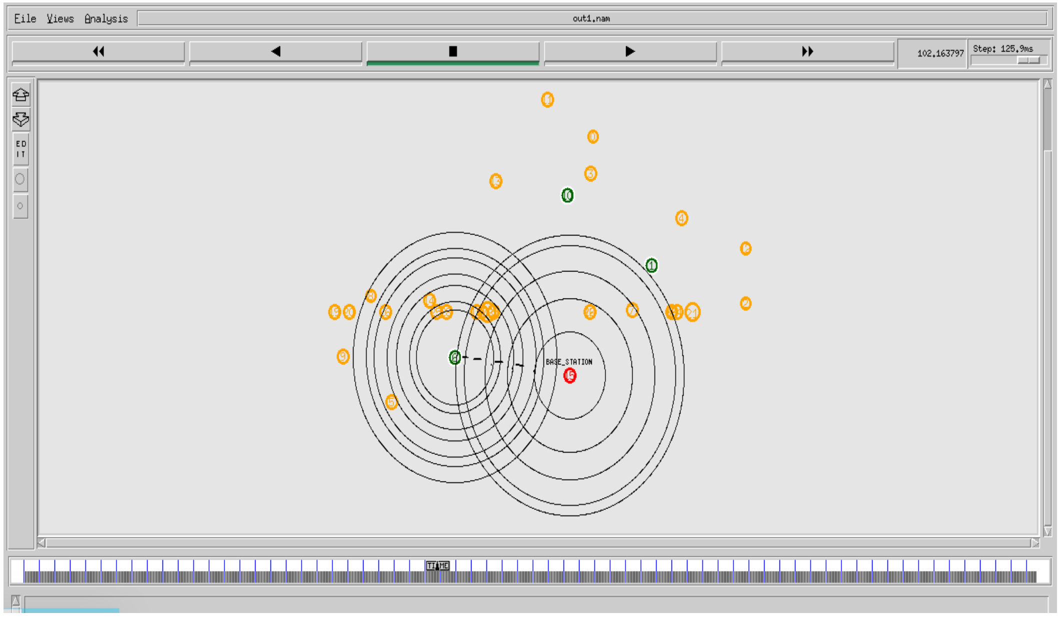


Figure 8:MINEMCS simulation scenario

4.5. Results

The figures 9, 10, 11, 12 13 and 14 show our experimental results, by comparing the MINEMCS method with the raw data collection method. The results given here are the results of the six experiments performed in round eight minutes. For each experiment, we repeated the experiment and presented the results as the average of those experiments. We selected different conditions to observe and verify the performance of the existing data collection model within the MINEMCS framework. Due to the different conditions, we observed different amounts of data transferred in our experiments.

The figure 9 shows average packet delay respect to number of nodes. It is evident from the figure that as the number of node increase, the packet delay also increases on scenarios and MINEMCS tolerates the delay. it can be concluded from Figure 9, MINEMCS has better performance than AS scenario.

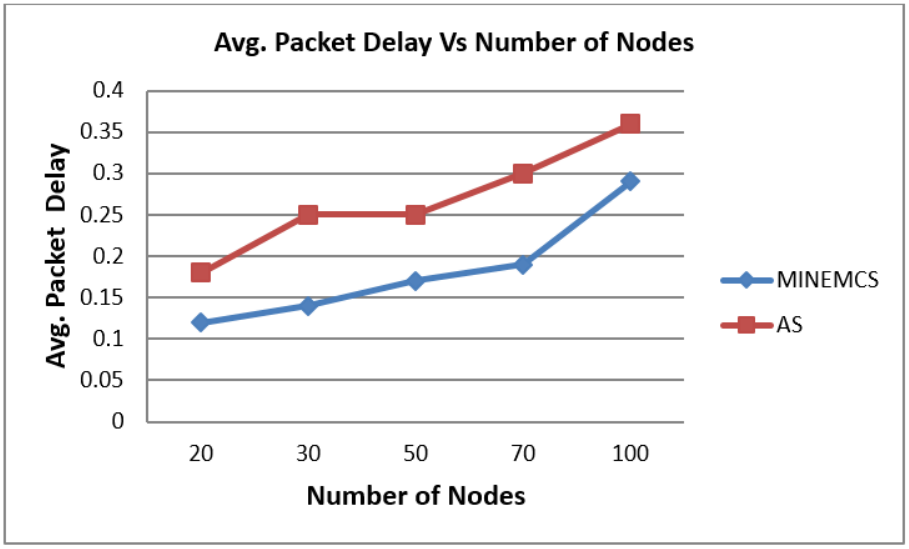


Figure 9: Average packet delay

The figure 10 show the throughput respect to its time line. The throughput is the cumulative percentage of simulation that conducted with different experimentation times. At the beginning AS scenario shows round ten percentage deference and at the meddle it approaches to the new scenario blow ten percentage deference. It can be concluded from figure 10 MINEMCS has improved the throughput in MCS.

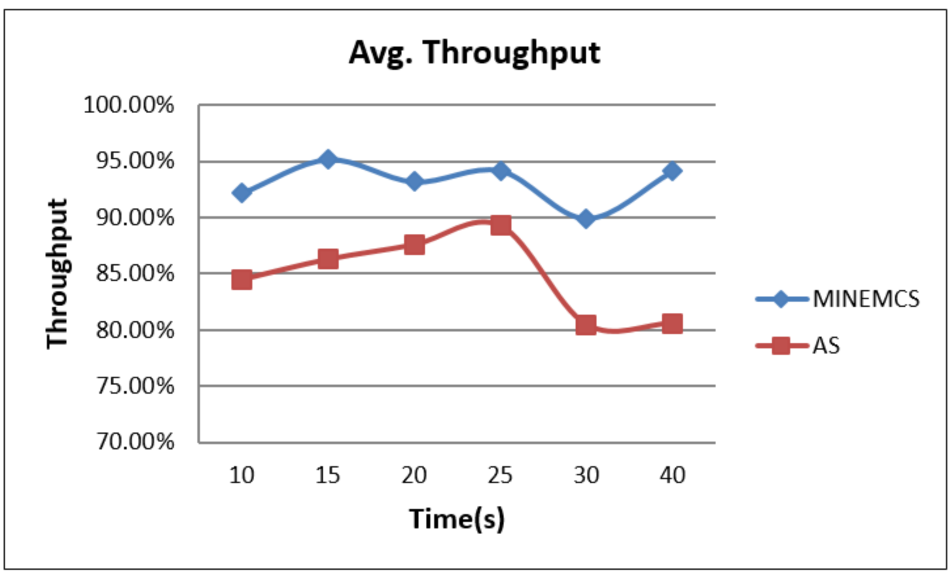


Figure 10: Average throughput

The figure 11 show the packet delay in second respect to the mobility of nodes. The nodes want to move from strong network signal to weak network signal within network coverage range.

Figure 11 concludes that the mobility of nodes does not affect the performance of network with scenario 2.

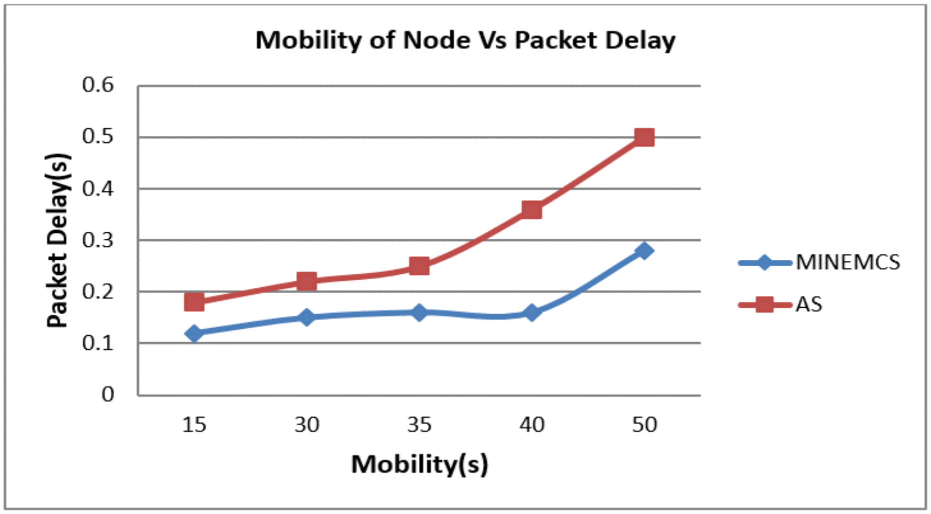


Figure 11:Mobility of Node Vs Packet Delay

The figure 12 show the average packet loss rate respect to the number of nodes. There is great deference between these two scenarios in the begging, at the middle of the graph AS try to improve the packet loss but it continues to proceed lost its packet with high rate. Based on figure 12 we conclude our scenario improved the packet delay rate with big deference.

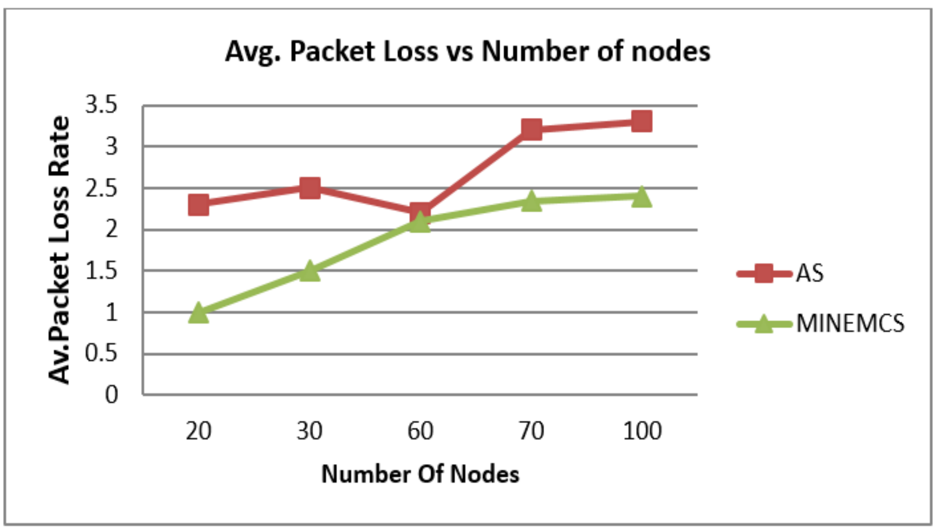


Figure 12: Average packet Loss

The figure 12 show average packet delivery ratio respect to its time line. When the number of nodes increase it proceeds proportional to each other until 15 seconds and after that the delayed to deliver the packet by the expected level. Form the figure 13 we conclude that MINEMCS has better packet delivery ratio than AS scenario.

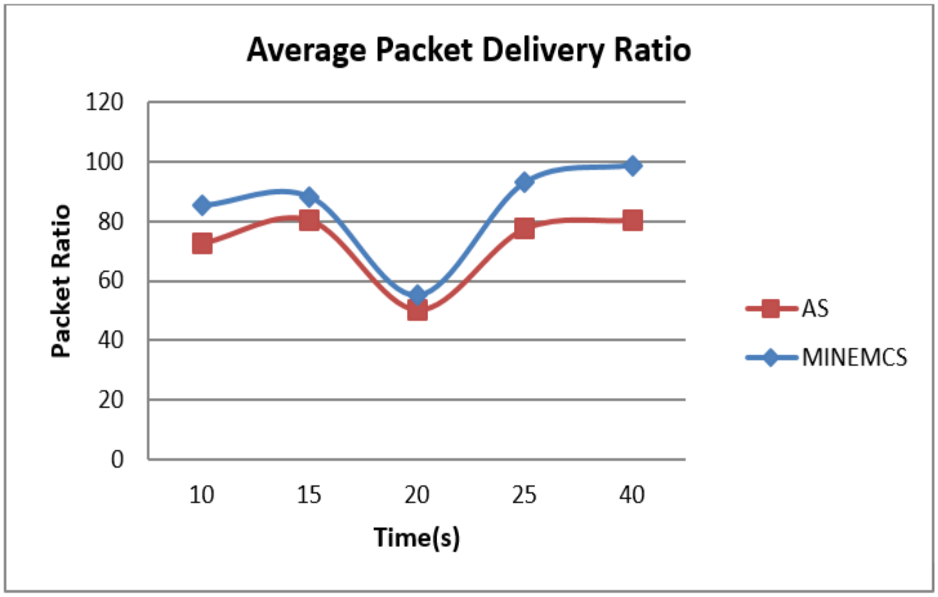


Figure 13:Average Packet Delivery Ratio

The figure 14 show the energy consumption respect to its time line. It manifests when the two scenario starts the result closed to each other but, as the time and number of node increases the AS scenario consumes highest energy to gather, process and upload data to the next publisher node. From figure 14 we summarize that the MINEMCS has low battery consumption than AS scenario.

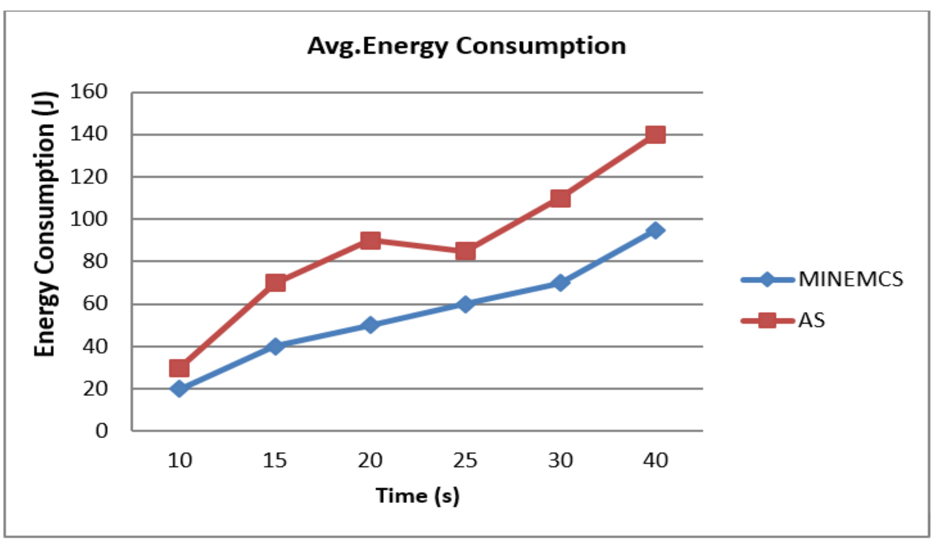


Figure 14: Average Energy Consumption

The results clearly demonstrate the significant improvement of MCS network with the above discussed performance metrics when sensing and transmitting process.

4.6 Summary

Observing the research question and proposed research design through experiment is the scientific way to test either the proposed approach is good or not. We used NS2 tools to evaluate the targeted approach with its simulation setup parameter in the field of 100,000m^2 with one 20,30, 40, 70,100 mobile nodes and grouping these nodes into sub clusters. For the evaluation of performance, we used three metrics; average packet loss, average packet delay, Average Packet Delivery Ratio, energy consumption and average packet throughput. The experiment conducted data collection of the users over several minutes within the environment condition to sense. The result of the experiment compeers two approaches i.e. the new proposed model and the existing application silo raw data uploading. The result Shaw that improvement of MCS network performance by sensing, processing and transmitting process.

CHAPTER FIVE

CONCLUSION AND FUTURE WORK

5.1 Conclusion

We present our MINIMCS system to support mobile sensing and focus on developing efficient and scalable data collection and processing models designed to reduce delay, delivery ration and bandwidth consumption associated with continuous sensing and uploading in such applications. We implement and evaluate our MINIMCS system with ns2 simulator. Based on the data that analyzed that demonstrated with the chart graph, shows that MINEMCS model improved mobile data collection, processing and uploading mechanisms and optimized packet delay and energy consumption, packet throughput and end to end delivery ratio compared to traditional application silo sensor-based upload technologies.

5.2 Future Work

The work presented in this paper is the first step and is an important part of the entire MINIMCS framework. We now intend to strengthen our work by investigating cloud data management methods for mobile sensing system. This includes data analysis and query processing on the cloud. In addition, we plan to expand our work by using sensor data on mobile devices with mobile activity identification to understand the background in which the sensing occurs. One of key challenge of mobile sensing is the ability to use battery efficiently when collecting data from multiple sources over a time. The amount of energy used to collect, analyze, and upload data and bandwidth usage when uploading data user does not satisfy because of small storage of the power in the mobile. Communication is a major factor affecting battery use so investigating on the battery optimization is also another assignment of the future work. The data shared in MCS is based on the voluntary group and who they have awareness of mutual benefits, there should be some incentive mechanism to improve data sharing between nodes. Finally, we aim to investigate and address privacy issues surrounding participatory and opportunistic data sensing applications rather than we discussed in this paper.

References

[1] M. Talasila, R. Curtmola, and C. Borcea, “Mobile Crowd Sensing,” *Handb. Sens. Netw. Adv. Technol. Appl.*, no. JANUARY 2014, 2015.

[2] R. K. Ganti, F. Ye, and H. Lei, “Mobile crowdsensing: current state and future challenges,” *Commun. Mag. IEEE*, vol. 49, no. 11, pp. 32–39, 2011.

[3] E. Macias, A. Suarez, and J. Lloret, “Mobile Sensing Systems,” *Sensors*, vol. 13, no. 12, pp. 17292–17321, 2013.

[4] S. Gisdakis, T. Giannetsos, and P. Papadimitratos, “Security, Privacy, and Incentive Provision for Mobile Crowd Sensing Systems,” *IEEE Internet Things J.*, vol. 3, no. 5, pp. 839–853, 2016.

[5] B. Guo, Z. Yu, and X. Zhou, “From Participatory Sensing to Mobile Crowd Sensing.”

[6] I. S. Bajwa, “Middleware Design Framework for Mobile Computing,” *Int. J. Emerg. Sci.*, vol. 1, no. 1, p. 38, 2011.

[7] T. Leppänen, J. Álvarez Lacasia, Y. Tobe, K. Sezaki, and J. Riekki, *Mobile crowdsensing with mobile agents*. 2015.

[8] M. Ra, B. Liu, T. F. La Porta, and R. Govindan, “Medusa,” *Proc. 10th Int. Conf. Mob. Syst. Appl. Serv. - MobiSys ’12*, no. Section 2, p. 337, 2012.

[9] G. Cardone, L. Foschini, P. Bellavista, and A. Corradi, “Fostering ParticipAction in Smart Cities : A Geo-Social Crowdsensing Platform,” no. June, pp. 112–119, 2013.

[10] Y. Wen *et al.*, “Quality-Driven Auction based Incentive Mechanism for Mobile Crowd Sensing,” vol. 9545, no. c, pp. 1–12, 2014.

[11] N. Maisonneuve, M. Stevens, and B. Ochab, “Participatory noise pollution monitoring using mobile phones,” vol. 15, pp. 51–71, 2010.

[12] P. P. Jayaraman, C. Perera, D. Georgakopoulos, and a. Zaslavsky, “Efficient opportunistic sensing using mobile collaborative platform MOSDEN,” *9th Int. Conf. Collab. Comput. Networking, Appl. Work.*, pp. 77–86, 2013.

[13] M. Shin, C. Cornelius, A. Kapadia, N. Triandopoulos, and D. Kotz, “Location Privacy for Mobile Crowd Sensing through Population Mapping.,” *Sensors (Basel).*, vol. 15, no. 7, pp. 15285–310, 2015.

[14] P. P. Jayaraman, A. Sinha, W. Sherchan, and S. Krishnaswamy, “Here-n-Now : A Framework for Context-Aware Mobile Crowdsensing.”

[15] X. Mo, D. Shi, R. Yang, H. Li, Z. Tong, and F. Wang, “A Framework of Fine-grained Mobile Sensing Data Collection and Behavior Analysis in an Energy-configurable Way,”

2015.

[16] A. Farshad, M. K. Marina, and F. Garcia, “Urban WiFi Characterization via Mobile Crowdsensing,” 2014.

[17] M. Informatics, “Mobile Crowd Sensing : an Approach to Smarter Cities Backgrounds and Motivations,” 2012.

[18] L. Wang, “effSense : Energy-Efficient and Cost-Effective Data Uploading in Mobile Crowdsensing,” pp. 1075–1086, 2013.

[19] S. Bradai, S. Khemakhem, and M. Jmaiel, “Re-OPSEC : Real Time OPportunistic Scheduler framework for Energy aware Mobile Crowdsensing.”

[20] J. Wang, Y. Wang, S. Helal, and D. Zhang, “A Context-Driven Worker Selection Framework for Crowd-Sensing,” vol. 2016, 2016.

[21] S. B. Eisenman, E. Miluzzo, N. D. Lane, R. A. Peterson, G.-S. Ahn, and A. T. Campbell, “BikeNet,” *ACM Trans. Sens. Networks*, vol. 6, no. 1, pp. 1–39, 2009.

[22] Q. Wang, “CrowdWatch : Pedestrian Safety Assistance with Mobile Crowd Sensing.”

[23] W. Guo and S. Wang, “Mobile Crowd-Sensing Wireless Activity with Measured Interference Power,” pp. 1–4, 2013.

[24] H. Chen, B. Guo, S. Member, Z. Yu, S. Member, and L. Chen, “A Generic Framework for Constraint-Driven Data Selection in Mobile Crowd Photographing,” vol. 4662, no. c, pp. 1–12, 2017.

[25] S. Konomi, V. Kostakos, K. Sezaki, and R. Shibasaki, “Crowd Sensing for Disaster Response and Preparedness,” pp. 449–450, 2015.

[26] M. Reichert, J. Herrmann, B. Langguth, and W. Schlee, “Mobile Crowd Sensing in Clinical and Psychological Trials – A Case Study.”

[27] S. Reddy, A. Parker, J. Hyman, J. Burke, D. Estrin, and M. Hansen, “Image Browsing , Processing , and Clustering for Participatory Sensing : Lessons From a DietSense Prototype,” pp. 13–17.

[28] J. Wan, J. Liu, Z. Shao, A. V Vasilakos, and M. Imran, “Mobile Crowd Sensing for Traffic Prediction in Internet of Vehicles,” pp. 1–15, 2016.

[29] H. S. S. Appliance, “PROMISE VTrak A-Class.”

[30] M. C. Sensing, “Mobile Crowd Sensing and Computing : The Review of an Emerging Human-Powered Sensing Paradigm,” vol. V, no. May, 2015.

[31] E. Koukoumidis, L.-S. Peh, and M. R. Martonosi, “SignalGuru,” *Proc. 9th Int. Conf. Mob. Syst. Appl. Serv. - MobiSys ’11*, p. 127, 2011.

[32] P. Tan and E. Yeow, “Thesis on Context Mediation using OWL,” no. June, 2004.

[33] Y. Chon, N. D. Lane, F. Li, H. Cha, and F. Zhao, “Automatically Characterizing Places with Opportunistic CrowdSensing using Smartphones,” 2012.

[34] Y. Zheng, L. Wang, R. Zhang, X. Xie, and W. Ma, “GeoLife : Managing and Understanding Your Past Life over Maps,” pp. 211–212, 2008.

[35] K. K. Rachuri, C. Mascolo, P. J. Rentfrow, and C. Longworth, “EmotionSense : A Mobile Phones based Adaptive Platform for Experimental Social Psychology Research,” *Int. Stud.*, vol. 10, pp. 281--290, 2010.

[36] W. Sherchan, P. P. Jayaraman, S. Krishnaswamy, A. Zaslavsky, S. Loke, and A. Sinha, “Using On-the-move Mining for Mobile Crowdsensing,” 2012.

[37] M. Ra, B. Liu, T. La Porta, and R. Govindan, “Medusa : A Programming Framework for Crowd-Sensing Applications Categories and Subject Descriptors,” no. Section 2.

[38] D. E. Difallah, G. Demartini, and P. Cudré-mauroux, “Pick-A-Crowd : Tell Me What You Like , and I ’ ll Tell You What to Do A Crowdsourcing Platform for Personalized Human Intelligence Task Assignment Based on Social Networks,” pp. 367–377, 2013.

[39] S. K. Datta, R. P. F. Da Costa, C. Bonnet, and J. Harri, “OneM2M architecture based IoT framework for mobile crowd sensing in smart cities,” *EUCNC 2016 - Eur. Conf. Networks Commun.*, no. ii, pp. 168–173, 2016.

[40] X. Hu, T. H. S. Chu, H. C. B. C. Member, and V. C. M. L. Fellow, “Vita : A Crowdsensing-Oriented Mobile Cyber-Physical System,” vol. 1, no. 1, 2013.

[41] K. Katevas, H. Haddadi, and L. Tokarchuk, “Poster: SensingKit: A Multi-platform Mobile Sensing Framework for Large-scale Experiments,” *Proc. 20th Annu. Int. Conf. Mob. Comput. Netw.*, pp. 375–378, 2014.

[42] G. Novak, D. Carlson, and S. Jarzabek, “An Extensible Mobile Sensing Platform for mHealth and Telemedicine Applications,” *Proc. Conf. Mob. Inf. Technol. Med.*, 2013.

[43] F. Hariri, G. Daher, H. Sibai, K. Frenn, S. Doniguian, and Z. Dawy, “Towards a Silent Mobile Sensing Framework for Smart Cities,” *Wirel. World Res. Forum*, 2013.

[44] I. Vakintis, “Middleware Platform for Mobile Crowd Sensing Applications Using Html5 Apis and Web Technologies,” 2015.

[45] P. G. Lopez, A. Montresor, D. Epema, A. Iamnitchi, P. Felber, and E. Riviere, “Edge-centric Computing : Vision and Challenges,” vol. 45, no. 5, pp. 37–42, 2015.

[46] K. Singh, “Mobile Phone Operating Systems :,” vol. 5, no. 3, pp. 610–613, 2014.

[47] S. Ding, X. He, and J. Wang, “Multiobjective Optimization Model for Service Node Selection Based on a Tradeoff Between Quality of Service and Resource Consumption in Mobile Crowd Sensing,” vol. 4, no. 1, pp. 258–268, 2017.

[48] E. Macias, A. Suarez, and J. Lloret, “Mobile Sensing Systems,” pp. 17292–17321, 2013.

[49] S. C. Wang, H. H. Pan, K. Q. Yan, and Y. L. Lo, “A unified framework for cluster manager election and clustering mechanism in mobile ad hoc networks,” *Comput. Stand. Interfaces*, vol. 30, no. 5, pp. 329–338, 2008.

[50] G. Dhand and S. S. Tyagi, “Data aggregation techniques in WSN : Survey,” *Procedia - Procedia Comput. Sci.*, vol. 92, pp. 378–384, 2016.

[51] D. D. M. Johnson, “The Dynamic Source Routing Protocol (DSR) for Mobile Ad Hoc Networks for IPv4 Status,” *Microsoft Res.*, pp. 1–107, 2007.

[52] C. Bettstetter, H. Hartenstein, and P. Xavier, “Stochastic Properties of the Random Waypoint Mobility Model,” pp. 1–34, 2003.

[53] M. Musolesi and C. Mascolo, “Mobility Models for Systems Evaluation A Survey,” *Mirco Musolesi, Dartmouth Coll. USA*, vol. 1, pp. 1–28, 2008.

[54] A. M. Borah and B. Sharma, “A Survey of Random Walk Mobility Model for Congestion Control in MANET ’ s,” vol. 111, no. 7, pp. 10–13, 2015.

[55] S. Andradóttir, K. J. Healy, D. H. Withers, B. L. Nelson, and A. Maria, “Proceedings of the 1997 Winter Simulation Conference,” pp. 7–13, 1997.

[56] B. M. C. M. Albrecht and P. E. Az, “Introduction to Discrete Event Simulation,” no. January, 2010.

[57] R. Jain, “A Survey of Wireless Sensor Network Simulation Tools,” pp. 1–10, 2011.

[58] A. Malik and H. Saini, “Network Simulators : A Comparative Survey,” *IOSR J. Electron. Commun. Eng.*, pp. 52–56, 2015.

[59] K. Liu, “Network Simulator 2 : Introduction NS-2 Overview,” *Wired*, pp. 1–31, 2004.

[60] H. Trivedi, S. U. S. B. College, and M. Aburoad, “A Review on Network Simulator & its Installation,” vol. I, no. I, pp. 2–3, 2012.

[61] J. Birla and B. Sah, “Performance Metrics in Ad-hoc Network,” *Int. J. Latest Trends Eng. Technol. Vol. 1*, vol. 1, no. 1, pp. 46–49, 2012.

[62] E. A. F. B. P. Norrington and N. Bessis, “Multi-objective performance optimization of a probabilistic similarity / dissimilarity-based broadcasting scheme for mobile ad hoc networks in disaster response scenarios,” 2013.

[63] V. Nuno, S. Sim, P. Medi, and S. S. Fios, “Performance Measurement in Wireless Sensor Networks,” 2016.

[64] S. Climent, J. V. Capella, N. Meratnia, and J. J. Serrano, “Underwater sensor networks: A new energy efficient and robust architecture,” *Sensors*, vol. 12, no. 1, pp. 704–731, 2012.

Appendix A: AWK Scripts for Evaluation Metrics Calculations

```
# Store start time
if (level == "AGT" && (event == "+" || event == "s") && pkt_size >= 512) {
if (time < startTime) {startTime = time }
}
# Update total received packets' size and store packets arrival time
if (level == "AGT" && event == "r" && pkt_size >= 512) {
if (time > stopTime) {
    a.  stopTime = time
    b.  }
# Rip off the header
hdr_size = pkt_size % 512
pkt_size -= hdr_size
# Store received packet's size
recvdSize += pkt_size
} }
END {
print "No of pkts send \t\t" send
print "No of pkts recv \t\t" recv
print "Pkt_delivery_ratio: \t\t" recv/send*100
print "Delay: \t\t\t\t" delay/recv
print "Throughput: \t\t\t" bytes*8/(ft-st)
print "No of Pkts Dropped \t\t" send-recv
print "Dropping_Ratio: \t\t" (send-recv)/send*100
print "Total_Energy_Consumption: \t" total_energy
print "Avg_Energy_Consumption: \t"  total_energy/nodes
}
```

Appendix B: TCL Scripts

```
#     Simulation parameters setup
set val(chan)   Channel/WirelessChannel    ;# channel type
set val(prop)   Propagation/TwoRayGround   ;# radio-propagation model
set val(netif)  Phy/WirelessPhy            ;# network interface type
set val(mac)    Mac/802_11                 ;# MAC type
set val(ifq)    Queue/DropTail/PriQueue    ;# interface queue type
set val(ll)     LL                         ;# link layer type
set val(ant)    Antenna/OmniAntenna        ;# antenna model
set val(ifqlen) 50                         ;# max packet in ifq
set val(nn)     31                         ;# number of mobilenodes
set val(rp)     AODV                       ;# routing protocol
set val(x)      881                        ;# X dimension of topography
set val(y)      652                        ;# Y dimension of topography
set val(stop)   250.0                      ;# time of simulation end
#        Initialization
#Create a ns simulator
set ns [new Simulator]
#Setup topography object
set topo       [new Topography]
$topo load_flatgrid $val(x) $val(y)
create-god $val(nn)
#Open the NS trace file
set tracefile [open out1.tr w]
$ns trace-all $tracefile
#creating trace files
set f0 [open MINEMCS.tr w]
set f1 [open AS.tr w]
set f4 [open lossMINEMCS.tr w]
set f5 [open lossAS.tr w]
set f8 [open dMINEMCS.tr w]
set f9 [open dAS.tr w]
#Open the NAM trace file
set namfile [open out1.nam w]
$ns namtrace-all $namfile
$ns namtrace-all-wireless $namfile $val(x) $val(y)
set chan [new $val(chan)];#Create wireless channel
#     Mobile node parameter setup
$ns node-config -adhocRouting  $val(rp) \
-llType        $val(ll) \
-macType       $val(mac) \-ifqType       $val(ifq) \
-ifqLen        $val(ifqlen) \
-antType       $val(ant) \
-propType      $val(prop) \
```

```
-phyType       $val(netif) \
-channel       $chan \
-topoInstance  $topo \
-agentTrace    OFF \
-routerTrace   ON \
-macTrace      ON \
-movementTrace ON
```