# MOBILE CROWD SENSING RPL-BASED ROUTING PROTOCOL FOR SMART CITY

Yahya Al Sawafi, Abderezak Touzene, Khaled Day and Nasser Alzeidi

Department of Computer Science, Sultan Qaboos University, Muscat, Oman

#### ABSTRACT

Recently, Mobile Crowd Sensing (MCS) has been used in many smart city monitoring applications, leveraging the latest smartphone features of sensing and networking. However, most of these applications use a direct internet connection for sending the collected data to the server through a 3G or 4G (LTE) network. This type of communication leads to higher bandwidth, battery consumption, and higher data plan cost. In this paper, we presenta new ad-hoc tree-based routing protocol named MCS-RPL based on theIoT RPL protocol for the smart city context. The proposed protocol aims to utilize smartphone and Mobile CrowdSensing (MCS) opportunistically to support static Wireless Sensor Network (WSN) and to cover more sensing areas with less routing overhead and power consumption. MCS-RPL uses grid-based cluster head to address mobility issues and reduce control packets. The conducted performance evaluation reveals that the proposed protocol outperforms RPL in terms of packet delivery ratio and power consumption due to control packet overhead reduction, which reached more than 75% in the tested scenarios.

### Keywords

MCS; Routing; ad-hoc; Opportunistic Networks; RPL; Smart City;

### **1. INTRODUCTION**

The recent advancements in human-carried everyday devices (e.g., smartphones, wearable devices, tablets) have triggered research in leveraging these devices to be used for sensing and monitoring of large-scale areas. This paradigm is called Mobile CrowdSensing (MCS), which empowers ordinary citizens to contribute data sensed or generated from their mobile devices and send the data for further extraction and service delivery[1].Human mobility and involvement offer participatory and opportunistic dynamic sensing coverage and data transmission. However, most of the existing mobile crowdsensing applications are utilizing infrastructure-based network architecture in which the mobile devices send the sensed data through infrastructure-based (3-4G, cellular). This type of communication introduceshigher implications on cost in terms of battery usage and data tariff[2]. Additionally, in some cases, the infrastructure-based is not available, or the network bandwidth is limited, especially for network overload situations[3].

With the advantages of mobile built-in communications interfaces such as WIFI and Bluetooth, devices can form a network topology using the ad-hoc network or Device 2 Device (D2D). This helps devices tocommunicateandforward datato each other without the need foran infrastructure-based network such as cellar or 4G networks. The motive behind the use of opportunistic ad-hoc routing protocol for MCS in the smart city context is to provide the city administration, users, and ISPsgreat benefits that include: (i) Reduction of the data traffic going to the cell towers, which save data for both users and ISPs. (ii) Reduction of energy consumptions, the study[4] shows that using 3G/4G networks drain the battery more than using WiFi/Bluetooth. (iii) Dense WSN architectures may not be necessary since mobile nodes can be used to cover the regions where node mobility is expected. (ii) Reducing the cost by having a smaller number of static WSN and benefiting opportunistically from the people involvements [5].

DOI: 10.5121/ijcnc.2020.12203

This kind of communication (ad hoc network) usually need a routing protocol to discover and establish a route between source and destination, maintain the availability of such route and facilitate the successful transmission of data along the chosen route. The existing routing protocols for mobile ad-hoc routing protocols (MANET) such as OLSR[6], AODV[7], Epidemic[8], and PRoPHET[9]are mainly designed for different purposes rather than for sensing and collecting data. These protocols are used for multi-hop point-to-point communications or data dissemination. However, the traditional use of sensor networks is to collect data from multiple points to be delivered to one central location. Besides, these protocols flood the network with control messages during route discovery, which leads to a broadcast storm problem. Therefore, they are not efficient to be used insensing networks[10].

We have inspired by the RPL (routing protocol for Low Power and Lossy Network) protocol to adapt it as a base for our proposed MCS routing protocol. RPL is a very promising routing protocol for WSN network and IoT in general. It is an IPv6 based and was standardized by IETF ROLL (Routing Over Low-Power Lossy Links) in 2012 [11]. Adopting IPv6 is very beneficial since it is considered to be the most suitable choice for IoT networking due to its excellent features such as universality, scalability, and stability. The huge addressing space that it offers makes it very suitable to address the large numbers of IoT objects.

The motive behind selecting the RPL as the base for our protocol is that: (i) RPL is mainly designed for data collection purposes supporting primarily multipoint-to-point traffic, similar to the traffic flow in our MCS scenario. (ii) RPL can also run on the top of different MAC layers. It benefits from utilizing built-in Wi-Fi in smartphones in our case. (iii) RPL is a tree-based protocol. The most important reason that makes tree-based topologies a better choice for sensing purposes is the ability to control the sensing and formation of the network. This is done by having one central node (usually the root) disseminates information to other nodes that wish to join the network. This information includes the size of the network, nodes type that can join, sensing types (which sensor to activate), time interval to sense, and other parameters. Although RPL is considered a reliable routing protocol for static WSN, it suffers from mobility support [12-21] that need to be tackled. The proposed work introduces a clustering mechanism and a logical two-dimensional (2D) grid to address the RPL mobility issues.

To the best of our knowledge, no previous work was conducted explore and adopt a protocol based on RPL to be used inMCSnodes (using IEE 802.11 data link interface). In our previous paper[22], we introduced a framework for a hybrid routing protocol based on the RPL protocol that proposes the design of two network trees for WSN and MCS and enables integration between them. In this work, we detailed the design, implement, and evaluate the Ad-hoc MCS-RPL protocol. Our contribution to this paper:

- Propose MCS-RPL, which is an ad-hoc routing protocol for MCS nodes that contributes to sense and/or to forward other nodes data to one central point called the sink. Thisprovidesan alternative way of sending the collected data directly to the server using a 3G/4G or LTE network.
- MCS\_RPL introduces clustering mechanism and a logical two-dimensional (2D) grid of equal-sized cells to address the RPL mobility issues and large control messages needed to maintain the tree.
- Implementand evaluate MCS-RPL using OMNeT++, one of the popular simulation tools for network simulation.

The remainder of this paper is organized as follows: Section 2introduces the backgroundof MCS technology and RPL protocol. Related works to MCS using ad-hoc network and RPL supporting mobility are discussed in section 3. The proposed work of opportunistic Mobile Crowd Sensing Routing Protocol for Smart City(MCS-RPL) is presented and discussed in section 4. Section 5 presents the evaluation methodology and result from the analysis for our approach. Section 6 concludes the paper and outlines future work.

## **2. BACKGROUND**

### 2.1 Mobile Crowd Sensing (MCS)

Mobile Crowd Sensing (MCS) has gained popularity in recent years due to its advantages over the traditional wireless sensing network (WSN). The large spreads of smartphones, which have set of built-in sensors (GPS, gyroscope, accelerometer, microphone, and camera), and communication interfaces (Wifi, Bluetooth) is the key factor of the success of MCS[23]. It offers lower costs and extended sensing covers compared to WSN since it depends on the human that carries those devices.Two classes of user contribution exist in MCS: participatory and opportunistic sensing. In participatory sensing, the participants are required to consciously feed the application, whichusually decides when, where, what, and how to sense. In contrast, there is no user involvement in opportunistic sensing in whichthe data is collected unconsciously in the background. An example of this type of sensing is the google-map traffic condition application that relies on location information signaled by mobile devices on the street. Sensing can be eventtriggered or continuous. In the event triggered case, sensors are triggered according to context (location, time, manual, etc.). In the continuous sensing case, sensors work continuously and usually are not context-aware, which might lead to drain the device battery [24].

Forthe last few years, many application domains have utilized the MCS technology as a better choice for sensing and collecting information. Many studies such as [25], [26, 27]worked on monitoringenvironmental conditions bycollecting data related to the temperature, air, and noise pollutions levels to be used for urban planning and to improve the citizen's quality of life. The latest works [27],[28],[29],[30],[31],[32] utilize MCS for intelligent transportation systems by monitoring the city public transport and services or the citizen mobility. This is togatherinformation about road and traffic conditions, free parking spots, and others. Works in[33]and [34]use MCS to monitor public safety-related events such as crimes, explosion, fires, and others using social media as sensors to report these events. Gathering health information from the peopleusing MCS is discussedin[35-38]. The type of this information includes allergy, blood pressure (hypertension), and cardiac information that is fed to the health system for analyses and to be used to provide better care. In addition, wellbeings like fitness and diet information can be exchanged.

MCS collection networkcan be categorized into three categories: infrastructure, ad-hoc, and hybrid. The infrastructure based is the most commonly used in MCS in which mobile devices use a direct connection with the Internet through 3-4G, cellular, or access points. The majority of the research studies mentioned above utilize infrastructure-based communication (using 3-4G, cellular, and access points). In ad-hoc networks, mobiles leverage built-in radio communications (WiFi, Bluetooth) to form a network topology between each other's and to beused to exchange orto forward the data. A hybridnetworkmixes the features of the prior two types [1].In our proposed work, we entirely use the opportunistic ad-hoc network, in which MCS devices will construct a tree-like network that connects to one central device called the root.

#### 2.2 Overview of the RPL Protocol

RPL is an IPv6 routing protocol that has been standardized for low-power and lossy networks (LLNs) by IETF in 2012[11]. RPL falls under proactive, distance vector, and collection tree types of protocols. To better understand RPL and how it operates, we first describe some of its main components and terminology, and then we illustrate by examplehow the topology is constructed. The RPL protocol arranges the nodes (sensors) into a tree topology called a destination-oriented directed acyclic graph (DODAG). The tree root (sometimes named as a border) is a nonconstrained node that initiates and orchestrates the tree construction. All nodes on each DODAG are rooted at a single root. Multiple DODAGs can be formed and cover the entire network topology without overlapping between them. The union of these DODAGs is called Directed Acyclic Graph (DAG). A node in a DAG can only be part of a single DODAG, but it can also join other DODAGs on different DAGs. An RPL instance may contain a single DODAG with a single root, multiple uncoordinated DODAGs with independent roots, a single DODAG with a virtual root that coordinates LLN sinks over a backbone network. An Objective function is used to determine how a DODAG is structured, which is based on a specific network and/or application need. It uses some metrics and constraints to compute some parameters such as the rank of the node (based on the distance to the root) and the preferred parent of the node. For example, the hop count metric can be used to compute the rank.

In order to exchange messages between the nodes themselves and with the root (in an ad-hoc manner), RPL uses some ICMPv6 based control messages. The most used control message is the DODAG Information Object (DIO). It is used by the root to initiate and maintain a DODAG tree and by other nodes to join this tree and to keep track of its RANK (position with respect to the DODAG root). As shown in Figure 1, the preconfigured root node (Rank 1) multicasts a DIO message that carries essential information for the DODAG construction. This information helps the nodes to discover an RPL DODAG/Instance, join and maintain the DODAG tree. The node that receives the DIO for the first time will extract it and use the information to add the sender to the parent list, compute its RANK based on the objective function, join the DODAG, and multicast the DIO to other nodes within its transmission range. This process is repeated until all nodes receive the DIO and join the DODAG. If the node receives multiple DIOs where the sender RANK is less than the RANK of this node's preferred parent, then it will update its RANK and preferred parent. For DODAG consistency purpose, each node will inspect the next received DIOs for different DODAG version or different RANK number than previous ones.



Figure. 1. RPL Components and Terminology

To search for new DODAG or maintain an existing one, if the node does not receive any DIO message within a specific time, it will multicast a DODAG Information Solicitation (DIS) message. This to solicit a DODAG DIO message from an RPL node. RPL also supports downward traffic routes by using a Destination Advertisement Object (DAO) control message, which is a unicast message that is used to propagate destination information upwards along the DODAG. To govern the transmission of RPL control messages sent by a node, RPL uses a Trickle timer algorithm[39], which is based on dynamic timers. It orchestrates when to send the next DIO using the timer mechanism. The idea is to send more DIO messages when the network is inconsistent or unstable and to send less frequently when it is stable.

# **3. RELATED WORKS**

Few studies considered utilizing ad-hoc networks for MCS compared to infrastructure-base. Additionally, since the proposed approach adapts RPL as the base protocol, this section also presents the works that haveadapted and implemented RPL-based protocols innon-LLNs networks.

### 3.1 MCS using ad-hoc network / D2D mobile crowdsensing

In[40], the author proposed a neighbor collaboration mechanism for MCS nodes using an opportunistic network aiming to enhance energy efficiency. The proposed method detects groups of pedestrians based on the history of radio connectivity between them to form clusters. The sensed data is shared between the cluster nodes using Bluetooth radio communication and then uploaded opportunistically to the server viaWiFi hotspot or 4G networks. Although this solution is suitable for a dense area, where people are forming such groups and short-range radio communication can be used. However, it is not applicable in a fragmented or less dense area.

Similarly, data aggregation among mobile devices for upload traffic reduction in crowdsensing systems is proposed in[41]. Contact history among mobile devices is also used to determine the most frequently contacted by the other devices so that it will collect sensory data intensively from the other devices. This is done by assigning ranks to the mobile devices according to their value of centrality measure, which prioritizes the devices that are frequently contacted by other devices to be the receivers of sensory data. Mobile devices with higher ranks will be favored for collection and receiving sensory data over many other devices that have relatively lower ranks. The aim here is to reduce the traffic volume incurred by uploading large sensory data.

The author in [42]proposed OppNet for mobile sensing and data collection from static sensor infrastructure. It consists of three phases: data collection, data relay, and data uploading. In the data collection phase, sensors that have access to OppNet transfer their data to the system middleware using Bluetooth to be pushed to the phone storage. The data relay phase is a routing mechanism responsible for sending the collected data from a user phone to another within the communication range. The data uploadingphase is responsible for transferring the data to the central server using low-cost communication channels such as WiFi. This solution is considered good for delay-tolerant architecture since the mobile device needs to store the data until it finds another mobile phone in the communication range, which can be used as a relay. However, it is not applicable for real-time applications, where sensed data need to be sent directly to the server.

In[43], the authors proposed a Device-to-Device mobile crowdsensing framework. This framework has a multi-criteria decision algorithm that decides between infrastructure and D2D communication mode, which tries to be less dependent on the infrastructure network (3G and 4G). The framework has an incentive mechanism to encourage people to use their smartphones for sensing purposes. However, this framework does not discuss the technical part (routing) regarding how the mobile nodes interact and how the messages flow between them.

The author in [44]develops a complete execution framework to solve the problem of collaborative mobile crowdsensing in opportunistic D2D. First, they formalize Minimum-Delay-Maximum-Coverage (MDMC)problem and Minimum-Overhead-Maximum-Coverage (MOMC) problem to produce optimized crowdsensing task execution schemes. Then, they proposed a mobility model that predicts user movement during a time window. This mobility model is used to transform MDMC and MOMC problems to routing the weighted directed graph. The aim is to search for the optimal execution schemes for the mobile crowdsensing tasks.

### **3.2. RPL for non-LLN**

In [16], the author adopted and evaluated RPL mobility in a joint roadside infrastructure network (Access Points) and Vehicular ad hoc network (VANET). Because RPL is slow to detect frequent topology changes such as in VANETs, the original DIO and DAOs sending intervals are changed to be fixed, and they are sent immediately upon new parent election. In addition, unlike scheduling ETX probing in RPL to determine the selection of new parent, immediate ETX a new parent is usedin VANET situation.Similarly, probing for the author in[45]proposedadapting RPL in the VANETby modifying Minimum Rank with Hysteresis Objective Function (MRHOF) objective function to include latency as the routing parameter. The work was evaluated and compared with the native RPL objective functions OF0 and MRHOF. The evaluation results concluded that RPL could be suitably modified to operate in the vehicular environment.

However, these proposed solutions disable the trickle algorithm so they can deal with network dynamic change caused by vehicle movement. This increases the network overhead caused by the large numbers of control messagesneeded periodically. Also Besides, they did not measure the energy consumption affected by this change.

Our proposed approach implementsan opportunistic ad-hoc protocol that is infrastructure-less (not using 3G-LTE) except for the root node. The tree-based structure of this protocol makes it a better choice for controlling the sensing mechanism. The root, which is the network orchestrator, control different options of sensing network. It determines who can join the network, network density (number of nodes in each area), type of sensing needed (temperature, GPS, camera, etc.), various times of sensing (night, day or a specific time), and more. This is difficult to implement in non-tree based (most of the previous works use collaboration approach to form an ad-hoc group). Besides, the majority of the previous work usesBluetooth as communication mediums to create the ad-hoc group between the mobile devices. This type of communication can be useful for the dense area but not in a fragmented one since it uses short-range radio.

Our approach usesWiFicommunication (similarly used in MANET), which can cover more areas and work for dense and fragmented areas. Moreover, these previous studies use delay tolerant mode where the data need to be stored temporarily until it finds other devices that can upload this data to them, which might stay for long until it reaches the server. However, many applications cannot tolerate much delay and need this information as fast as possible. In our approach, as soon as the mobile node joinsthe ad-hoc tree, it will know there is a valid path to the server (root) which can send its data through. To control the number of control messages needed to maintain the tree stability because of the mobility of the nodes, we introduce the clustering mechanism, whichwill be more explained in the next section.

# 4. THE PROPOSEDOPPORTUNISTIC MOBILE CROWD SENSING ROUTING PROTOCOL (MCS-RPL) FOR SMART CITY

#### 4.1 Application Scenario and Design Requirements

Our proposed approach is designed for urban areas where many people use their smart mobile devices. The solution may use these mobile devices for collecting and forwarding the sensed data. This data could be anything that can be supported by smart mobile devices sensors (built-in or attached sensors). In our scenario, we targetthe environmental data, such as temperature, humidity, air CO level, vibrations, noise, and others. We assume that not all smartphones have the same sensor types. Each sensor type has a unique identifier (ID) to differentiate it from other sensors. In addition, the smartphonesthatdo not have the desired sensors, they still can forward other MCS data (routing only). All measured data are collectedalong with the location (using a GPS sensor). These data arethen sent to the central server for analysis, which can be used by some applications to improve the city services. Each mobile node participating in the MCS network is willing to sense and forward packets for other nodes in the network. The city administration offers some incentives to the contributors by allowing them to access some information they can benefit from, such as traffic locations, pollution levels in a particular area.

The city is divided into sectors in which each sector is covered by a group of sensors reporting to one sink. The sink (root), is a powerful node that has the high processing power and good storage capacity. It is also equipped with multiple network interfaces (IEEE 802.15.4, WiFi, LAN, and LTE). The root creates thestatic WSNsDODAG tree (using native RPL) on IEEE 802.15.4 interface. We use the second root interface (Wi-Fi) to construct the MCS DODAG tree (MCS-RPL). Smart mobile devices are also utilizing their Wi-Fi physical interface to join the MCS tree and to send their sensed data or forward other devices to the sink. The assumption that all the devices have almost the same radio transmission range. The union of all sinks represents the backbone of the city sensing network that connects with each other's and with the city administration by a cable (fiber) or wireless (LTE).

One of the main challenges that face ad-hoc mobile tree networks (in our case MCS) is the tree stability compared to the static WSN. This issue rises from the mobility of the nodes, which makes the MCS tree suffer from frequently broken paths. Thus, it will need a large number of control messages to maintain the tree backbone. To overcome and address such issues, we consider the solution that tries to reduce the number of control messages and limit the tree backbone maintenance to only specific nodes. We introduce a clustering mechanismand a logical two-dimensional (2D) grid that maps the city area.

The city consists of a virtual grid of equal-sized cells (for example, each cell is 100\*100 meter). The MCS nodes located on each cell will form a cluster (group), and only one node will be elected as the leader (head) or the cluster-head of the group within a cell. The non-cell-head nodes within a cell can only connect to the cell-head inside the same cell. The cell-head will forward the sensed data on behalf of the non-cell-head nodes across other cells to the root. The backbone of the DODAG tree is formed and maintained by the cell-head nodes. This helps in reducing the amount of the control messages and routing information that propagates inside the network. Besides, keeping the cluster heads of the cells as the tree backbone is much easier than maintaining all nodes' connections.

To assist in constructing the tree in such an environment, we use location information provided by positioning devices such as global positioning systems (GPS). Each mobile node (smartphone) has a GPS receiver so that it can be aware of its geographical location and the cell it belongsto.

Localization will also assist the node in knowing when reaching and leaving DODAG borders. Each cell will have a unique ID that identifies it from other cells. The cell ID is inserted in the transmitted control messages (DIOs) so that the nodes will know from which cell this message comes from. This is very important for differentiating the messages transmitted from the local cell from those sent from other cells.

#### **4.2 MCS-RPL Details**

In addition to the default DIO structure, we have added some fields that are essential for our MCS-RPL implementation. These are the (x, y) coordinates of the position of the node (using GPS coordinates), the battery level, mobility status, and the node ID. The location of the sender node is needed to determine whether it belongs to the same cell as the recipient node or not. The mobility status is a one-byte integer field that is set by the node to 1 if it is a mobile node or 0 if it is static. The node ID, along with the battery level, is required for cell-head selection when there are more than one candidate parents within the same cell with the equal rank value. In this case, the node with the lowest ID and highest battery level (as in our implementation) is selected. This will be more explained later on in the following section. Figure2 shows the modified structure of the DIO message we use in our work.



Figure.2. DIO structure in MCS-RPL

Figure3 demonstrates the overall workflow processes of the MCS-RPL protocol. At the startup, the root and the nodes initialize two tables named CellNodeCache and NeighborCache. The CellNodeCachetable storesinformation about the neighbornodes within the same cell. This information includes the MAC address, theIPv6 address, NodeID, and the DODAG ID.The NeighborCachetable is onlyused by cell-head nodes to store information about other cell-head nodes within their radio range. The root propagates a DIO message containing typical information about the tree, such as DODAG ID, version, and Rank. In addition, the DIO includes the objective function (OF) used for constructing the mobile tree (in our case, we use the hop count). Other non-root nodes will add some fields to the original DIO that are necessary for the clustering purpose. These fields are the current cellID and battery level of this node. Pseudocode 1 shows the initialization process for the DODAG tree.



Figure.3. MCS-RPL Processes

Pseudocode 1 MCS-RPL DODAG initialization phase				
1	Initialize()			
2	Begin			
3	For all nodes DO			
4	Initialize CellNodeCacheTable (IPv6_Address, Node_ID, DODAG_ID, Cell_ID), NeighborCache Table (Node_ID, IPv6_Address, Cell_ID)			
5	Initialize x, y Location (get_current_position())			
6	Root initialize DODAG: Create DIO containing DODAG_ID, Version, Rank, OF, cell_ID, etc.			
7	Root multicast DIO to all reachable nodes			
8	End			

Upon receiving the DIO message, the MCS node extracts it for further processing. Joining the DODAG tree and selecting cell-head and leaf nodes are done implicitly using the same DIO message without requiring extra control messages or overhead. This can be achieved by one or more iterations of sending and receiving of these DIO messages. Each node that received its first DIO will compute its RANK, select the sender as prefer parent, and join the tree. A node adds the sender's node information to its CellNodeCache if this sender's DIO is from the same cell as this node cell. The election process of the cell head node is very simple. A node verifies if the DIO sender is from the same cell, and the sender rank is less than this node's rank. If this is the case, it assigns this sender is from the same cell but has the same rank as this node, then the node's highest battery level will be the cell-head. If the sender is from the same cell but has the same rank and same battery level as this node, then the node highest battery level and lowest node ID

will be the cell-head. This is to make sure not to select the same nodes as cell heads each time, in which case their battery will be depleted faster than the other nodes. Note that electing cell-heads does not require extra cost or a specific election algorithm; it uses the same DIO message.

All leaf nodes within the cell set the cell head as their preferred parent. The cell head is the only node that selects its preferred parent outside the cell border. The node, which assigns itself as a leaf node, stops the trickle algorithm from sending more DIO messages. This reduces control messagesas much as possible and prevents looping. Only the cell head node uses the Objective Function (OF) to compute its RANK and to select its preferred parent from other cells according to specific metrics and constraints such as position, the number of hops, ETX, etc. (in our implementation we used hop count as an objective function). After adding its computed rank and other parameters, the cell head node propagates this updated DIO to other nodes within the radio range. This mechanism will be repeated until all the nodes within the DODAG border join the tree. Pseudocode 2 gives the details of handling DIO in MCS-RPL.

Pseudocode 2 DIO process in MCS-RPL			
1	DIO Handling()		
2	Begin		
3	For all nodes receiving DIO do		
4	If DIO sender from the same cell then add this node to the CellNodeCache		
5	If DIO sender from the same cell and the Sender Rank< than this Node Rank then		
6	IsJoined = true (joinDODAG)		
7	Mark this node as LeafNode, Rank becomes infinite, Make the sender as Preferred Parent (default route)		
8	Delete DIO (Don'tbroadcast DIO again)		
9	Return		
10	Else If DIO sender from the same cell and the Sender Rank= this Node Rank and Sender ID< than this		
	Node_IDand Sender_Battery_level> than this Node_Battery_level then		
11	IsJoined = true (joinDODAG)		
12	Mark this node as <i>LeafNode</i> , <i>Rank</i> becomes infinite, Make the sender as <i>Preferred_Parent</i> (default route).		
13	Delete DIO (Don'tbroadcast DIO again)		
14	Return		
15	Else		
16	If (IsJoined&LeafNode)		
17	Delete DIO (Don'tbroadcast DIO again)		
18	Return		
19	If(/IsJoined)		
20	IsJoined = True, Add sender as a parent, Select preferred (default route) with lower Rank, Compute		
	this Node_Rankto (Parent_Rank+1)		
21	Send <i>DIO</i> periodically according to the <i>DIO</i> trickle timer.		
22	Else (If already joined)		
23	If the <i>Sender_Rank</i> ≤ this <i>Node_Rank</i> then		
24	If the sender does not exist in the parent table ( <i>NeighborCache</i> ), then add it		
25	<b>ElseIf</b> the sender exists in the parent table but with different <i>Rank</i> , <b>then</b> update the table		
26	ElseIf the sender exists in the parent table with no change, then do nothing		
27	Update Preferred_Parent (default route)		
28	Else		
29	Suppress DIO (Ignore)		
30	End		

Due to the mobility of the mobile nodes, which leads to frequent tree instability, there is a need for a mechanism to maintain the tree backbone and keep it connected as much as possible. In our approach, we introduce a function called validate position, which is triggered by a node periodically (every 0.5 seconds in our implementation). The primary purpose of this function is to verify whether the node is still in the same cell or has moved to another cell. An action is only needed if the node has moved to anothercell by triggering a DIS message, in addition to other reassignment parameterssuch as clearing the CellNodeCashe table, assign itself as non-leaf node temporary and others. Pseudocode 3 shows the details of this process.

Pseudocode 3 Validate Position Function in MCS-RPL			
1	Node_Position_Validation()		
2	Begin		
3	For all nodes DO		
4	If the node has moved to anothercell (Current_Cell_ID != Previous_Cell_ID)then		
5	If leafNode then		
6	leafNode = false		
7	IsJoined = false		
8	Clear CellNodeCache, Clear its Preferred_Parent (default route)		
9	Create and send (multicast) DIS message		
10	Else		
11	IsJoined = false		
12	Clear CellNodeCache, Clear its Preferred_Parent (default route)		
13	Create and send (multicast) DIS message		
14	End		

Another essential function in this protocol is the handlingof the DIS messages sent by new nodes searching for DODAG network to join or by those nodes who moved from one cell to another cell and triggeredthese DIS messages. The pseudocode 4 outlines this process. The general rule is that all leaf nodes that receive a DIS message discard it and do nothing except if the sender is the preferred parent of this node. In this case, it temporarilyleaves the DODAG tree, does not mark itself as a leaf node, and sends a DIS messagerequesting for DIO message from neighbor nodes. All cell heads receiving DIS message will reset their trickle algorithm to speed up sending the DIO message.

Pseudocode 4 Processing DIS Message in MCS-RPL		
1	DIS_Handling()	
2	Begin	
3	For all nodes receiving DIS DO	
4	If !IsJoined then	
5	Delete DIS (Ignore)	
6	Else If leafNode and the sender is the Preferred Parent (default route) then	
7	leafNode = false	
8	IsJoined = false	
9	Clear CellNodeCache, Clear its Preferred Parent (default route)	
10	Delete DIS, Create and send (multicast) DIS message	
11	Else If (leafNode&&LsJoined) then	
12	Delete DIS	
13	Else	
14	TrickleResetDIO (Not for leafNodes)	
15	Delete DIS	
16	End	

#### **4.3 MCS-RPL Implementation**

We have implemented this protocol using OMNeT++ simulation tool. OMNeT++ is an extensible, modular, component-based C++ simulation library and framework, primarily for building network simulators[46]. It is open-source and can run on top of different operating systems such as Linux, Windows, and MAC. It has the capability of implementing and simulating RPL and other routing protocols at a larger scale. We used the RPL in OMNeT++ source code in [47] as a base for our new protocol. We have done many modifications in the source code to apply the algorithm mentioned above.

As mentioned earlier, the area covered by MCS nodes is divided into a virtual grid with equal cell size, so different cell sizes can be used here. We have decided to make the cell size  $=\frac{r}{2\sqrt{2}}$ , where r is the radius of the circle representing the radio range of a node. The reason behind this choice is that each node within a cell can reach other nodes within neighbor cells at any location (see figure 4).



Figure. 4. Relationship between r and  $d \left( d = \frac{r}{2\sqrt{2}} \right)$ 

### **5. MCS-RPL EVALUATION**

#### 5.1 Simulation Methodology and SetupEnvironment

Since our approach is different from those mentioned in the related works (from design, structure, etc.), or from the one used in MANET protocols (such as AODV and OLSR which we mentioned before that they are not appropriate for sensing purposes), it is difficult and unfair to compare against them. Therefore, the evaluation is conducted against the native RPL protocol to see how our protocol performs and deals with the mobility in MCS. The main evaluation metrics used are: Average control packets overhead, packet delivery ratio (PDR), average End-to-End delay, and average power consumption. By this, we can check whether it can give a better packets delivery ratio and with lower overheads and power consumptions.

Average control packets overhead is the total number of control messages transmitted by nodes to maintain the routes with respect to the data packets. In our scenario, the control messagesrepresent the DIO and DIS messages. The average control traffic (ACF) is calculated as:

$$ACF = \sum \frac{number of DIO and DIS}{number of datapackets}$$
(1)

The control packets overhead havea direct impact on the power consumption of the nodes. More control messages meanmore transmissions are required, which leads to more energy depletion. Also Besides, the increase of the control messages means more collisions with other control messages and data packets, which affects the PDR. The Packet delivery ratio (PDR) is defined as:

$$PDR = \frac{numberof packets recived}{numberof packets sent}$$
(2)

In a multipoint to point scenario like in our case, the sink is the only node that receives but does not send data packets. The other nodes are the only senders. The end-to-end delay is the time between a packet is sent by the source node and received by the sink. The average end-to-end delay is calculated as follows:

$$AED = \sum_{i=1}^{n} \frac{(timepacketreceived_i - timepacketsent_i)}{n},$$
(3)

#### where nist het ot alnumber of packets received

TheAverage power consumption is calculated in a simple battery module as:

$$APC = \frac{initial battery capacity - residual battery capacity}{simulation time}$$
(4)

We have considered different scenarios for this evaluation purpose. In each scenario, we modified the number of nodes and their mobility speed. We set the node number to be 50, 100, 150, and 200 to see how the protocol performs with different densities. We also change the movement speed to be 2, 4, and 6 meters per second, to evaluate how the protocol performs with varying speeds of mobility. We simulate the people walking in an urban area using the random waypoint mobility model. In the random waypoint mobility pattern, a node walks randomly for some time (after selecting a random destination point and a random speed between 0 and some maximum speed) and stops in between for a fixed amount of time before walking randomly again [48].

We also set DIOIntervalMin to 0.5 sec for both protocols so that they can send the DIO faster when the trickle is rested in response to network change.Each node generates and sends one UDP packet (simulate the sensing data) every one second. This data packet should be forwarded upward the tree hop by hop using the cell head nodes until it reaches the sink. Table.1shows the configuration parameters of the simulation. Some of these configuration parameters that are related to the RPL (such as DIOIntMin, DIOInetDouble, etc..) are based on some previous works and also on empirical testing. They are tuned to the values that produced good results. The data generation (represented here by UDPApp) start after 5 seconds of the simulation. This to give appropriate timefor the initial tree formation before start sending data packets.

Table.1.	Simulatio	on configura	tion parameters
		0	1

Parameter	Value(s)
Area	800m * 800m
Cell Size	100m
Number of nodes	50, 100, 150, 200
Physical layer	IEEE 802.11 (WiFi)
Radio range	250m
Simulation time	300s
Repeat	5
DIOIntMin	0.5s
DIOInetDouble	6
DIORedun	5
UDPApp(Packet Generation)	Start at 5s
	End at 300s
UDPApptrafficGeneration rate	Every 1s
Mobility Type	RandomWPMobility
Mobility speed	2mps, 4mps, 6mps

### 5.2 Results and Analysis

The results obtained from the simulation experiment from the simulation experiments measured the effect of varying the density and the mobility speed on the control message overhead, PDR, end-to-end delay, and power consumption. The following subsections discussed the results:

### 5.2.1 Average Control Packets (Overhead)

Figure5 shows the average control messages (DIO) and (DIS) sent by a node in RPL and MCS-RPL in the three mobility scenarios with different node density. It demonstrates that MCS-RPL usesmuch less DIO control messages compared to RPL, with almost 75% less. This is because when an MCS node assigns itself as a leaf node, it stops the trickle algorithm from sending more DIOs.Only cell head nodes send DIOs frequently according to the setting of the trickle algorithm. Since in our design, there is only one cell-head node per cell, and the rest are leaf nodes, this results in a high reduction in the number of DIOs. Moreover, the selection of the cell head node is made implicitly using the same DIOs message without requiring extra DIOs or other control messages. The figure also reveals that the number of control messages slightly increases when increasing the mobility speed and decreaseswhen increasing the density, whereas in RPL, it increases more with respect to both mobility speed and density. The number of DIS messages in both protocols is very few and can be neglected.



Figure. 5. Average Control Messages (DIO) vs. Number of MCS nodes for MCS-RPL and RPL for Different Speeds.

### 5.2.2 Packet Delivery Ratio

Figure6 shows the packet delivery ratio (PDR) of the RPL and MCS-RPL with respect to the network density and mobility speed. It illustrates that MCS-RPL exhibits higher delivery ratio (by almost 10%)than RPL. The overall PDR of MCS-RPL in all density and mobility scenarios is nearly over 80%. This can be linked to the smallnumber of control messages used by MCS-RPL. The increasingnumber of control packets on the air leads to more collisions of these packets with data packets, which degrades the number of delivered data packets to the sink. The figure also shows thatMCS-RPL and RPL perform better at a speed of 2 m/s compared to higher speeds of 4 and 6m/s. With low mobility speed, the default parent (default root) of a node does not change too frequently, which leadstomore extended stability of the node connection. The figure also shows that thebest PDR performance of MCS-RPL in all mobility speed scenarios occurs at high densities(number of nodes between 100 and 150). This can be justified as follows: at lower density, there are more chances that a node does not find a parent or losesits parent due to bigger distancesbetween nodes. However, at high density, there is higher chance of message collisions due to nodes trying to send at the same time.



International Journal of Computer Networks & Communications (IJCNC) Vol.12, No.2, March 2020

Figure.6. Packet Delivery Ratio (PDR) vs. Number of MCS nodes for MCS-RPL and RPL for Different Speeds.

RPL

Number of Nodes (Speed 4m/s

MCS-RPL

#### 5.2.3 Average End-to-End Delay

Number of Nodes (Speed 2m/s

RPL - MCS-RPL

Figure7demonstrates the average end-to-end delays of RPL and MCS-RPL. It shows that RPL has a lowerend-to-end delay than MCS-RPL, which is expected. This is because of the clustering mechanism of MCS nodes in MCS-RPL. The Leaf nodes send their data packets only to the cell head within the same cell and do not send them directly to other cell head nodes outside their cell. They do not work as routerswhen they are in leaf node status and cannot forward packets of other leaf nodes. This implies that the data packets have to move internally first to the cell head and then to the cell heads of other cells until reaching the sink. This explains the slightly higher delay than native RPL. However, this delay is still within a very acceptable rate, especially for sensing applicationswhere, in many cases, the delay is not that critical.



Figure.7. End-to-End Delay vs. Number of MCS nodes for MCS-RPL and RPL for Different Speeds.

#### 5.2.4 Average Power Consumption

Mobile nodes in ad-hoc multi-hop networks are battery-driven. Thus, they suffer from limited energy level problems [49]. The Radio's transmission Tx and receiving Rx states consume a considerably high share of the overall wireless node power [50]. Having more control messages being transmitted in addition to the data packets will affect the power consumption directly. Therefore, the design of the routing protocols should effectively take this consideration and minimize the number of control messages as possible. Figure 8 shows the average power consumption for wireless mobile nodes in MCS-RPL and RPL. MCS-RPL exhibits lower power consumption than RPL. This can be linked to the higher number of control messages produced by RPL comparing to MCS-RPL.

Number of Nodes (Speed 6m/s

- MCS-RPL

RPL



Figure. 8. Average Power Consumption vs. Number of MCS nodes for MCS-RPL and RPL. Mobility Speed 2m/s

### **5.3 Evaluation Limitation**

In any evaluation or experiments, some limitations and deficiencies might affect the results. In our proposed system, the involvements of the mobile nodes are opportunistic and based on the availability of the nodes. In sometime, the contributions of the MCS nodes are high, especially in the daytime, whereas it is very less after midnight. The very low involvement will limit the area coverage and largertree formation. On the other hand, with denser node participations, there are higher chances of duplicated sensed data. The MCS nodes that are very close to each other will probablysense or measure similar data inputs. Another thing that might affect the evaluation of the proposed work is the assumption of wireless node coverage. We consider that all nodes have the same wireless ranges; however, this might not be the case in the reality where nodes have different wireless interfaces ranges. Also, we assume that the nodes work in open areas. However, some physical wireless signal obstacles such as walls, trees, and constructions can affect the communication between the nodes.

### 6. CONCLUSION AND FUTURE WORK

We have introduced opportunistic MCS-RPL routing protocol that collects and routessensing information from MCS nodes to the central devicein an ad-hoc approach rather than using a 3G\LTE network. The protocol shows a higher reduction in the amount of routing overhead (reduction of 75% compared to native RPL). This has a direct impact on lowering the power consumption of the nodes, which is considered very important in battery-operated devices. Our protocol also exhibited a good packet delivery ratio with respect to network density and mobility speed. According to the conducted experiment, the MCS-RPL protocol is a very good ad-hoc candidate for sensing and data collection purposes, especially in urban areas and smart city context. Our future work is to enhance this protocol by controlling the number of nodes on each cell that can participate in the sensing activities. This will improve the end-to-end delay, PDR, andreduce further power consumption.

#### REFERENCES

- [1] B. Guo et al., "Mobile crowd sensing and computing: The review of an emerging human-powered sensing paradigm," ACM Computing Surveys (CSUR), vol. 48, no. 1, p. 7, 2015.
- [2] L. Wang, D. Zhang, H. Xiong, J. P. Gibson, C. Chen, and B. Xie, "ecoSense: Minimize participants' total 3G data cost in mobile crowdsensing using opportunistic relays," IEEE Transactions on Systems, Man, and Cybernetics: Systems, vol. 47, no. 6, pp. 965-978, 2016.
- [3] C. Xu et al., "Efficient 3G/4G budget utilization in mobile sensing applications," IEEE Transactions on Mobile Computing, vol. 16, no. 6, pp. 1601-1614, 2016.
- [4] J. Huang, F. Qian, A. Gerber, Z. M. Mao, S. Sen, and O. Spatscheck, "A close examination of performance and power characteristics of 4G LTE networks," in Proceedings of the 10th international conference on Mobile systems, applications, and services, 2012: ACM, pp. 225-238.
- [5] M. Di Francesco, S. K. Das, and G. Anastasi, "Data collection in wireless sensor networks with mobile elements: A survey," ACM Transactions on Sensor Networks (TOSN), vol. 8, no. 1, p. 7, 2011.
- [6] T. Clausen and P. Jacquet, "Optimized link state routing protocol (OLSR)," 2070-1721, 2003.
- [7] I. D. Chakeres and E. M. Belding-Royer, "AODV routing protocol implementation design," in 24th International Conference on Distributed Computing Systems Workshops, 2004. Proceedings., 2004: IEEE, pp. 698-703.
- [8] R. Ramanathan, R. Hansen, P. Basu, R. Rosales-Hain, and R. Krishnan, "Prioritized epidemic routing for opportunistic networks," in Proceedings of the 1st international MobiSys workshop on Mobile opportunistic networking, 2007: ACM, pp. 62-66.
- [9] A. Lindgren, E. Davies, S. Grasic, and A. Doria, "Probabilistic routing protocol for intermittently connected networks," 2012.
- [10] R. V. Biradar, V. Patil, S. Sawant, and R. Mudholkar, "Classification and comparison of routing protocols in wireless sensor networks," Special Issue on Ubiquitous Computing Security Systems, vol. 4, no. 2, pp. 704-711, 2009.
- [11] T. Winter, "RPL: IPv6 routing protocol for low-power and lossy networks," 2012.
- [12] H. Fotouhi, D. Moreira, and M. Alves, "mRPL: Boosting mobility in the Internet of Things," Ad Hoc Networks, vol. 26, pp. 17-35, 2015.
- [13] O. Gaddour, A. Koubâa, R. Rangarajan, O. Cheikhrouhou, E. Tovar, and M. Abid, "Co-RPL: RPL routing for mobile low power wireless sensor networks using Corona mechanism," in Industrial Embedded Systems (SIES), 2014 9th IEEE International Symposium on, 2014: IEEE, pp. 200-209.
- [14] F. Gara, L. B. Saad, E. B. Hamida, B. Tourancheau, and R. B. Ayed, "An adaptive timer for RPL to handle mobility in wireless sensor networks," in Wireless Communications and Mobile Computing Conference (IWCMC), 2016 International, 2016: IEEE, pp. 678-683.
- [15] J. Ko and M. Chang, "Momoro: Providing mobility support for low-power wireless applications," IEEE Systems Journal, vol. 9, no. 2, pp. 585-594, 2015.
- [16] K. C. Lee, R. Sudhaakar, L. Dai, S. Addepalli, and M. Gerla, "RPL under mobility," in Consumer Communications and Networking Conference (CCNC), 2012 IEEE, 2012: IEEE, pp. 300-304.

- [17] I. El Korbi, M. B. Brahim, C. Adjih, and L. A. Saidane, "Mobility enhanced RPL for wireless sensor networks," in Network of the Future (NOF), 2012 Third International Conference on the, 2012: IEEE, pp. 1-8.
- [18] D. Carels, E. De Poorter, I. Moerman, and P. Demeester, "RPL mobility support for point-to-point traffic flows towards mobile nodes," International Journal of Distributed Sensor Networks, vol. 11, no. 6, p. 470349, 2015.
- [19] M. Bouaziz, A. Rachedi, A. Belghith, M. Berbineau, and S. AlAhmadi, "EMA-RPL: Energy and mobility aware routing for the Internet of Mobile Things," Future Generation Computer Systems, 2019.
- [20] F. Gara, L. B. Saad, R. B. Ayed, and B. Tourancheau, "A new scheme for RPL to handle mobility in wireless sensor networks," International Journal of Ad Hoc and Ubiquitous Computing, vol. 30, no. 3, pp. 173-186, 2019.
- [21] S. Sanshi and C. Jaidhar, "Enhanced mobility aware routing protocol for low power and lossy networks," Wireless Networks, vol. 25, no. 4, pp. 1641-1655, 2019.
- [22] Y. Al Sawafi, A. Touzene, K. Day, and N. Alzeidi, "Toward hybrid RPL based IoT sensing for smart city," in Information Networking (ICOIN), 2018 International Conference on, 2018: IEEE, pp. 599-604.
- [23] A. Capponi, C. Fiandrino, B. Kantarci, L. Foschini, D. Kliazovich, and P. Bouvry, "A Survey on Mobile Crowdsensing Systems: Challenges, Solutions and Opportunities," IEEE Communications Surveys & Tutorials, 2019.
- [24] B. Guo, Z. Yu, X. Zhou, and D. Zhang, "From participatory sensing to mobile crowd sensing," in Pervasive Computing and Communications Workshops (PERCOM Workshops), 2014 IEEE International Conference on, 2014: IEEE, pp. 593-598.
- [25] J. Dutta, F. Gazi, S. Roy, and C. Chowdhury, "AirSense: Opportunistic crowd-sensing based air quality monitoring system for smart city," in 2016 IEEE SENSORS, 2016: IEEE, pp. 1-3.
- [26] L. Liu, W. Liu, Y. Zheng, H. Ma, and C. Zhang, "Third-eye: a mobilephone-enabled crowdsensing system for air quality monitoring," Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies, vol. 2, no. 1, p. 20, 2018.
- [27] F. Montori, L. Bedogni, and L. Bononi, "A collaborative internet of things architecture for smart cities and environmental monitoring," IEEE Internet of Things Journal, vol. 5, no. 2, pp. 592-605, 2017.
- [28] A. Janecek, D. Valerio, K. A. Hummel, F. Ricciato, and H. Hlavacs, "The cellular network as a sensor: From mobile phone data to real-time road traffic monitoring," IEEE transactions on intelligent transportation systems, vol. 16, no. 5, pp. 2551-2572, 2015.
- [29] A. Allouch, A. Koubâa, T. Abbes, and A. Ammar, "Roadsense: Smartphone application to estimate road conditions using accelerometer and gyroscope," IEEE Sensors Journal, vol. 17, no. 13, pp. 4231-4238, 2017.
- [30] D. Shin et al., "Urban sensing: Using smartphones for transportation mode classification," Computers, Environment and Urban Systems, vol. 53, pp. 76-86, 2015.
- [31] F. J. Villanueva, D. Villa, M. J. Santofimia, J. Barba, and J. C. Lopez, "Crowdsensing smart city parking monitoring," in 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT), 2015: IEEE, pp. 751-756.

- [32] X. Chen and N. Liu, "Smart parking by mobile crowdsensing," International Journal of Smart Home, vol. 10, no. 2, pp. 219-234, 2016.
- [33] M. Pouryazdan and B. Kantarci, "The smart citizen factor in trustworthy smart city crowdsensing," IT Professional, vol. 18, no. 4, pp. 26-33, 2016.
- [34] P. Fraga-Lamas, T. Fernández-Caramés, M. Suárez-Albela, L. Castedo, and M. González-López, "A review on internet of things for defense and public safety," Sensors, vol. 16, no. 10, p. 1644, 2016.
- [35] R. Pryss, W. Schlee, B. Langguth, and M. Reichert, "Mobile crowdsensing services for tinnitus assessment and patient feedback," in 2017 IEEE International Conference on AI & Mobile Services (AIMS), 2017: IEEE, pp. 22-29.
- [36] R. Pryss, M. Reichert, W. Schlee, M. Spiliopoulou, B. Langguth, and T. Probst, "Differences between android and ios users of the trackyourtinnitus mobile crowdsensingmhealth platform," in 2018 IEEE 31st International Symposium on Computer-Based Medical Systems (CBMS), 2018: IEEE, pp. 411-416.
- [37] L. A. Kalogiros, K. Lagouvardos, S. Nikoletseas, N. Papadopoulos, and P. Tzamalis, "Allergymap: A Hybrid mHealth Mobile Crowdsensing System for Allergic Diseases Epidemiology: a multidisciplinary case study," in 2018 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops), 2018: IEEE, pp. 597-602.
- [38] S. Jovanovic et al., "A Mobile Crowd Sensing Application for Hypertensive Patients," Sensors, vol. 19, p. 400, 2019.
- [39] P. Levis and T. H. Clausen, "The trickle algorithm," 2011.
- [40] T. Higuchi, H. Yamaguchi, T. Higashino, and M. Takai, "A neighbor collaboration mechanism for mobile crowd sensing in opportunistic networks," in Communications (ICC), 2014 IEEE International Conference on, 2014: IEEE, pp. 42-47.
- [41] M. Tamai and A. Hasegawa, "Data aggregation among mobile devices for upload traffic reduction in crowdsensing systems," in Wireless Personal Multimedia Communications (WPMC), 2017 20th International Symposium on, 2017: IEEE, pp. 554-560.
- [42] F. Shi, U. Adeel, E. Theodoridis, M. Haghighi, and J. McCann, "OppNet: Enabling citizen-centric urban IoT data collection through opportunistic connectivity service," in ,Internet of Things (WF-IoT), 2016 IEEE 3rd World Forum on, 2016: IEEE, pp. 723-728.
- [43] V. F. Mota, T. H. Silva, D. F. Macedo, Y. Ghamri-Doudane, and J. M. Nogueira, "Towards scalable mobile crowdsensing through device-to-device communication," Journal of Network and Computer Applications, vol. 122, pp. 99-106, 2018.
- [44] L. Wang, Z. Yu, D. Yang, T. Ku, B. Guo, and H. Ma, "Collaborative Mobile Crowdsensing in Opportunistic D2D Networks: A Graph-based Approach," ACM Transactions on Sensor Networks (TOSN), vol. 15, no. 3, p. 30, 2019.
- [45] R. Pradhan, S. Rakshit, and T. De, "Performance Evaluation of RPL under Mobility for VANETs," in 2018 5th International Conference on Signal Processing and Integrated Networks (SPIN), 2018: IEEE, pp. 739-744.
- [46] O. Ltd. "https://www.omnetpp.org/." (accessed.

- [47] X. Liu, Z. Sheng, C. Yin, F. Ali, and D. Roggen, "Performance analysis of Routing Protocol for Low power and Lossy Networks (RPL) in large scale networks," IEEE Internet of Things Journal, vol. 4, no. 6, pp. 2172-2185, 2017.
- [48] C. Bettstetter, G. Resta, and P. Santi, "The node distribution of the random waypoint mobility model for wireless ad hoc networks," IEEE Transactions on mobile computing, vol. 2, no. 3, pp. 257-269, 2003.
- [49] S. Shankar and N. G. S. Varaprasad, "jayanthi,"Designing Energy routing protocol with power consumption optimization in MANET"," IEEE Transaction, 2013.
- [50] J. Schandy, L. Steinfeld, and F. Silveira, "Average power consumption breakdown of Wireless Sensor Network nodes using IPv6 over LLNs," in Distributed Computing in Sensor Systems (DCOSS), 2015 International Conference on, 2015: IEEE, pp. 242-247.

#### AUTHORS

Yahya Alsawafi receive his BSc degree in computer science from Sultan Qaboos University, Oman in 2001 and MSc in computer science from University of Bedfordshire, UK in 2015. He is currently a PhD student at computer science department at Sultan Qaboos University. His research interests include Computer Network, Computer Security, Internet of Things, programming, Artificial intelligence.

**AbderezakTouzene** received PhD degree in, computer science from INPG France in 1992. Now he is a professor in Sultan Qaboos University, Department of Computer Science. His research, interests include mobile networks, network security, and distributed computing

**Khaled Day** received the PhD degree in computer, science from University of Minnesota, USA in 1992, now he is a professor in Sultan Qaboos University, Department of Computer Science. His research, interests include parallel and distributed computing,, and networks.

**Nasser Alzeidi** received the PhD degree in computer, science from University of Glasgow, UK in 2007. Now he is an associate professor in Sultan Qaboos, University Department of Computer Science. His, research interests include embedded networked and, distributed systems, performance evaluation, and, mobile Ad hoc networks.







