IoT Technologies for Mobile Crowd Sensing in Smart Cities

Abbas M. Ali Al-muqarm and Furkan Rabee
Computer Science Department, Faculty of computer science and mathematics, University of Kufa, Iraq
Email: abbasm.almahmoud@student.uokufa.edu.iq; Furqan.rabee@uokufa.edu.iq

Abstract —Internet of Things is a sophisticated concept of the Internet, so all things in our lives can be connected to the Internet or to each other to send and receive data to perform specific functions through the network. The Mobile Crowd Sensing (MCS) technology is used by users who use a smartphone as a new direction in the development of the Internet of Things. The volunteers rely on collecting data from the environment by taking advantage of smartphone features such as (camera, temperature, GPS, Microphone, etc.).There are two challenges with MCS are Battery power consumption and the cost of data upload. This paper presented a new framework called Efficient power consumption (EPC - MCS) based on the use of one of the IoT protocols for the limited devices resources such as(power, memory, and processing ,etc.) which is the CoAP protocol after comparing its performance with the protocols (MQTT and XMPP) by simulation proved to be power efficient. This framework proposes dividing the MCS area into two areas , Global mobile node, and local mobile node, volunteers in the global area are collecting data by using their smartphone and sending data to the local area, that might be smartphone, computers or other existing devices at home or a health center. If a local nodes uploading the data to a switch, the switch will route the data to the gateway which has two choices: If there is a need to make a quick decision, should be switched to the fog layer, else uploading the data to the cloud via a Wi-Fi or a 3G connection with a piggyback. Thus, yield two advantages are Zero cost for uploading data and less energy consumption. In this paper, we presented an algorithm to route the data between the global nodes and local nodes with fog computing and without. Evaluate the performance of our framework through use cooja emulator show to be effective in efficient power consumption and data uploading, compared to previous articles in the same subject.

Index Terms—Internet of Things (IoT), Mobile crowd sensing (MCS), smart cities, fog computing, CoAP.

I. Introduction

Internet of things is the next technology revolution, Shorten the term is IoT, the first Internet of things term used by Kevin Ashton in 1999, in the Internet of things can communicate between things and data transfer from machine to machine (M2M) without direct intervention of the people ,IoT makes our life easier by receiving notices on the personal phone to tell us what we need in the refrigerator of food or to tell us about the roads or intersections crowded to change direction , home automation and smart home systems will be the largest market for the Internet of things in the consumer sector by the end of 2020. Internet of things is a network of various physical devices, household devices, agricultural devices and industrial, as well as health sensors such as monitoring The rate of heartbeat and oxygen , either in the environmental side includes the prediction of the weather and temperature, humidity, light By spreading a number of sensors in different places and this is called Wireless Sensor Networks (WSNs), a wireless sensor network is a wireless network consisting of various devices distributed using sensors to monitor physical or environmental conditions. A WSN system link to internet Across a gateway by using many protocols and internet protocol ipv4 or ipv6 for connecting sensor devices, or can be collect data by people who are the process of sensing the environment through their mobile phone and this is called Mobile Crowdsensing (MCS), atypical MCS scenario involves users carrying their smartphones with applications running in the background to continuously collect sensor readings, either from built-in sensors or wearables, such data acquisition activity requires minimal user involvement and is named opportunistic sensing in literature, in contrast to participatory sensing, which requires active user involvement to create sensor readings [39]. Internet of things uses different communication technologies such as radio-frequency identification (RFID), Bluetooth, Near Field Communication (NFC), Wireless networks (Wi-Fi and ZigBee). On the opposite side of the mobile Crowdsensing advantages, there are some challenges, most importantly the power consumption and the cost of data upload, in this paper we proposed a structure and algorithm to reduce the power consumption and cost of data upload.

Manuscript received December 25, 2018; revised July 8, 2019.
doi:10.12720/jcm.14.8.745-757

II. Related Works

IoT technologies for Mobile Crowd Sensing is already a large research area, where there are many studies lead to Energy-Efficient System design for IoT devices based role of emerging memory technologies and approximate computing [1]. Provide solutions for Energy-Conserving in using diverse wireless radio access technologies for IoT connectivity [2]. Efficient Power Consumption in Wireless Communication Techniques for the Internet of Things based on using low power wireless techniques and modules, for short range connectivity the candidate protocols are ZigBee, 6LoWPAN and low power Wi-Fi, For long connectivity, the candidate is LoRaWAN protocol [3].

A. Mobile Crowd Sensing (MCS)

There are many persons have sensing and computing devices collectively share data and extract information to measure and map phenomena of common interest. [4]. A propose a four-stage life cycle to characterize the mobile crowd sensing process [5]. Monitoring of environmental noise pollution in urban areas [6]. A series of techniques for optimizing the uploading process [7]. A propose new specific metrics for the analysis of MCS datasets [8]. Presented a simulation platform for MCS systems is tailored to assess sensing activities in large-scale realistic urban environments is a public street lighting [9]. Provide incentives for stimulating users to participate in mobile crowdsensing applications, such as (air pollution monitoring, noise monitoring, and traffic monitoring) [10].

B. Energy Conservation in Mobile Crowdsensing

Present a design framework for an Energy-Efficient Mobile Sensing System uses hierarchical sensor management strategy by powering only a minimum set of sensors and using appropriate sensor duty cycles [11]. Introduce a sensing approach which lowers the power requirement for motion sensing where described a technique that can use for body motion sensing using a novel approach which leverages static electric fields around the human body [12]. Presented a new framework called Efficient Transfer Route for Mobile Crowd Sensing to manage two challenges are power consumption and data uploading cost by clustering the participant users with two areas [13]. Reduce the overall energy consumption by split multiple modules of a continuous sensing application between the main processor and a low-power processor [14]. A propose framework to reduce energy consumption and data uploading cost, reduces data cost by loading data to Bluetooth/Wi-Fi gateways and reduces energy consumption by piggyback data on a call or using more energy-efficient networks rather than initiating new 3G connections [15]. A propose number of energy efficient data delivery strategies using real-time mobile data stream mining for data reduction this lead to benefits in energy efficiency [16]. Gets benefit by parallel data uploading with voice calls or Parallel Connections over Bluetooth or Wi-Fi [17]. In this paper presents an energy efficient Mobile Crowdsensing framework where sensing results are transferred in parallel with phone calls where The proposed framework embeds several mechanisms from existing work such as parallel transfer and cycle-based delay and propose an algorithm to avoid redundant task assignments [18]. Reduce data size before uploading such as compressing process for data [19], or uploading part of the data while assuming the rest [7].

C. Data Cost Conservation in Data Uploading

There are several previous studies in reducing the cost of data focused on reducing the size of data by compression before uploading [21]. A propose a framework for minimizes the cost of both sensing and reporting [20]. Moreover, in addition to reducing energy consumption can be reducing data cost by using Wi-Fi or piggyback with 3G communication with considering the priority to the message should be transferred or upload first such as [13]. In another paper a propose a delay tolerant data uploading framework, by introducing delay tolerant data uploading mechanisms.UnDP participants could relay PAYG participants sensed data to the server without additional cost; PAYG participants could also upload their sensed data via free charge Bluetooth/Wi-Fi gateways to reduce cost [22].

III. Problem Statements

In this section, we presented problems and challenges for MCS implementation. MCS depends on the volunteers to use their smartphone devices to collect information about the environment, health or traffic congestion, and send or share with interested parties. Running the application on their mobile smartphone and the sensing process leads to consume energy, the other problem is uploading data to servers locally or via the internet leads to increasing the cost of upload data, this is two important problems became the main interesting for researchers. Several methods have been proposed in order to reduce energy consumption and cost-effective data uploading. Perhaps the most popular of these are reducing the size of data by compression before uploading [21], or uploading part of the data while assuming the rest [7] this may lead to the problem of losing some important data. An article proposed a delay tolerant data uploading framework to minimize mobile data cost and decreased energy consumption [22]. From the above, we proposed a new structure based on the Constrained Application Protocol (CoAP) to reduce the energy consumption and reduce data uploading cost through uploading data during the call depending on the CoAP protocol characteristics (piggyback).

IV. Mobile Crowd Sensing (MCS) and Smart Home

At the moment, everyone began to use technology and modern devices to make their new life easier and more comfortable, especially with the development of mobile crowd sensing research. A smart home is one such application of IoT, where it became possible to control many of the home appliances as well as control of the system of home protection using mobile technology based on the signal sent from the mobile phone to perform monitoring tasks for the elderly at home, control of lighting, temperature, Smart TVs, health, Entertainment and smart cameras, where can residents can monitor their homes when they are away or an outhouse. Fig. 1 shows (Technical fields in smart homes), where used much wireless technologies such as (Wi-Fi, Bluetooth and Bluetooth LE, ZigBee, and Z-Wave to connect smart devices with each other’s and connect them to the internet over the gateway to allowing to control household smart devices by an application on their smartphone. There are many concepts in the smart home implementation with mobile crowd sensing:

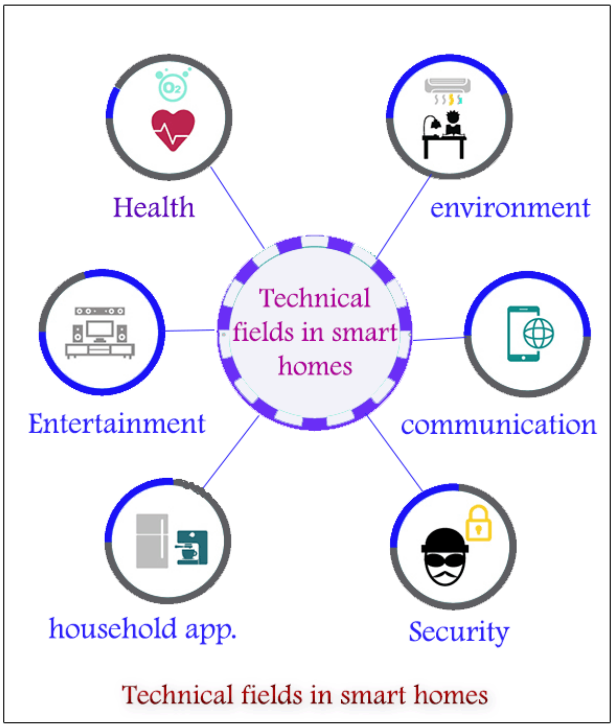

Fig. 1. Technical fields in smart homes

A. Sensing Data

Smart home systems consist of different sensors such as this in TVs, smart refrigerator, coffee maker, doors, temperature monitor and sound sensor. All these sensors collect data from the environment such as humidity or act as a control device for some smart devices such as a remote control to adjust the temperature of the room, Sensor data is the output of these devices, the sensors are connected to a central unit called (gateways). The devices communicate with the gate via M2M technology, where Low-power wireless communication technologies can be used to connect devices with a gateway such as Wi-Fi, ZigBee and Bluetooth low energy will, therefore, be our collection of data that must be sent through the gateways to guide a specific process.

B. Collecting Data

Designing a system for the processing and collecting of sensor data from smart home devices through design a software which takes an advantage of the combination of PaaS cloud services available in Microsoft Azure [23]. As mentioned above in Part A, the sensors are collecting continuous data this leads to a large size of raw data that needs processing and storage, Some sensor data are small in size and some are very large, so we see that small - sized data can be collecting on the sensor itself before sending it to the gateway and then to the cloud , either if the data is of large size, we suggest that it be sent in real time to the gateway which may be a smartphones to collect and process data and even reduce unnecessary data before uploading to the cloud and all of that because of the limitations of energy and memory in sensors. Data processing on the edge (gateway) is faster than sending data to the remote cloud in addition to reducing the cost of data processing.

C. Routing Data

After collecting the data, it should be route to the cloud. Routing process among sensor nodes, it's a very important point to reduce energy consumption and the burden of upload data to the cloud, so it's necessary to select suitable routing protocols and communication techniques. The smart home contains many devices are manufactured by different companies where the devices will be incompatible with each other so we need what is called (hub) or (bridge), the function of it translating the various communication protocols, e.g. , when the mobile device wants to communicate with another a smart home device uses different communication technology by using (hub) to serve as a translator between the two, in addition to its ability to data processing and management wireless communication. Smart devices are connected to each other using different communication technologies such as (Wi-Fi or Bluetooth) but we see that these two technologies consume more energy, we are trying to reduce energy consumption so we suggest using Z-Wave or ZigBee, where they have a lower rate of energy consumption. Smart devices network needs to connect to the (gateway) by using one of the low power protocol such as (MQTT, CoAP, etc.), that is designed to reduced power and then connect over the Internet to enable users to access and manage their devices remotely. Fig. 2 shows schema a illustrates the routing structure of a smart home.

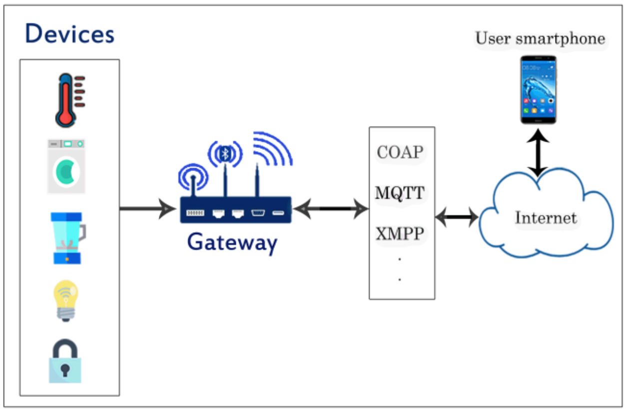

Fig. 2. Routing structure of a smart home

The sensor nodes at home can be on two types:

- **The first type** is fixed and Connected to a continuous electrical source, this type of node called the N_e^F .
- **The second type** is the mobile nodes that operate on the battery, this type of node called the N_b^M .We proposed that, N_e^F nodes be the responsibility of

transferring data to the gateway. The nodes N_b^M are collect data and send them to N_e^F contract to reduce the energy consumed depending on the algorithms of choosing the shortest path and the appropriate routing protocols such as (LEACH) algorithm see this in Fig. 3.

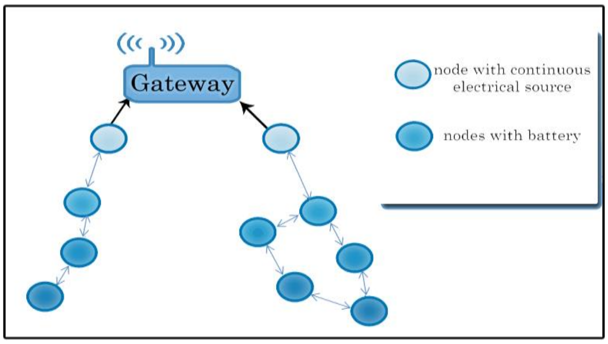

Fig. 3. Smart home network depending on the LEACH algorithm

D. Updating Data

There are many sensors deployed in the smart home such as (health care, temperature, light, lock, pressure, etc.). All these devices generate different data, we need to store them to be processed and analyzed as well to access them over the Internet, after the data collection by the sensors are routed through the gateway by using a suitable communication technique such as (ZigBee, Bluetooth or Wi-Fi). It is now necessary to look for a sturdy cloud to host our sensors data, to pass this data to the cloud we use a specific protocol such as (CoAP or MQTT), data is stored in the cloud in private (in the case of a smart home) or in general if we want to share data with the others, to reduce power consumption we proposed to send data in real time, or sent data at any time, also compress data or delete redundancy or delete unnecessary data to reduce cost of data uploading, as shown in Fig. 4.

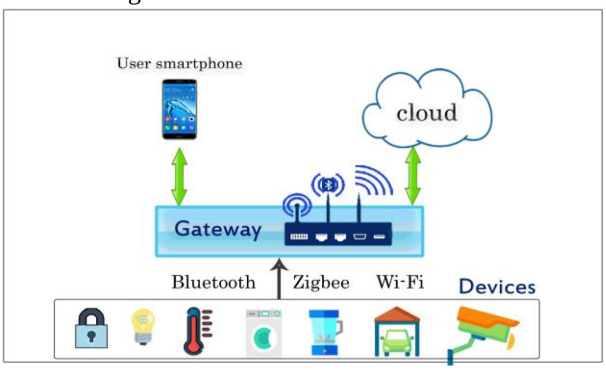

Fig. 4. Data upload in smart home

V. Internet of Things (IoT)

A. Structure of IoT (Protocols and Layers)

Internet of Things a combination of smart devices that communicate with each other's Without direct human intervention, the devices communicate with each other and connect through the gateway which may be (router, switch or smartphone) to connect to server in other side by using different communication technologies such as(ZigBee, Bluetooth , NFC, Wi-Fi , and Z Wave) which are used to reduce power consumption and data cost ,on the other hand, there are many non - physical protocols that are used to integrate with the communication technologies mentioned above for the same purpose, such as MQTT, CoAP, XMPP, AMQP, and DDS. IOT stack consist of application layer protocols (MQTT, CoAP, XMPP, AMQP, DDS), transport layer protocols (UDP or TCP), network layer protocols (RPL, IPv6 and IPv4), the Adaption layer (6LoWPAN), Data link layer (Ethernet, Wi-Fi) and physical layer (IEEE 802.15.4). All layers and protocols of IoT stack as shown in Fig. 5.

layer	Layer name	protocol
6	Application	MQTT, COAP, XMPP, AMQP, DDS
5	Transport	UDP / TCP
4	Network	RPL , IPv6 , IPv4
3	Adaption	6lowpan
2	Data link	Ethernet , Wi-Fi
1	Physical	IEEE 802.15.4

Fig. 5. IoT stack (layers and protocols)

B. Low Energy Protocols

In This section, we want to display the protocols in the application layer. Internet of things devices will usually operate on batteries, as most of the protocols are not designed to conserve energy, so it is necessary to design protocols that support the reduction of energy consumption, real-time performance, bandwidth, memory and battery consumption are essential to be considered when dealing with sensors or devices with limited power. Those protocols are:

1) Constrained Application Protocol (CoAP)

CoAP is a protocol designed to work on devices with limited resources for the transfer of internet content over the web as an alternative to the html protocol, which is not designed for the limited resources and it is complex, so CoAP protocol carefully designed to maintain the characteristics of the html protocol but in a simpler way, CoAP runs over UDP (User datagram protocol) that helps to avoid costly TCP handshake before data transmission [24], CoAP support request/response message also it works with many communication technologies such as Bluetooth, Wi-Fi , 2G, 3G, and 4G networks. One of the main reasons to use (CoAP) in our design is to reduce power consumption and cost of uploading data and CoAP stack shown in Fig. 6.

CoAP uses four types of messages to exchange the messages between client and server as in Fig. 7.

The types of messages are:
1) Confirmable (CON).
2) Non-Confirmable (NON) messages.
3) Acknowledgment (ACK) messages.
4) Reset (RST) messages.

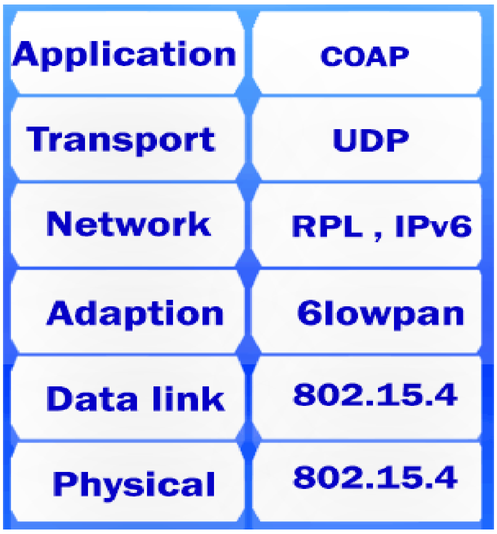

Fig. 6. CoAP stack

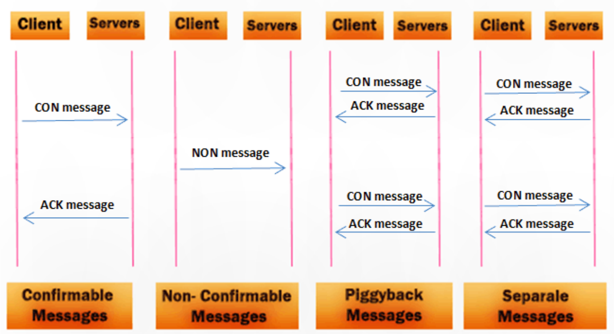

Fig. 7. Four types of messages to exchange the messages between client and server.

The following operations are performed by a CoAP:
1) GET: Used to retrieve the current status of the resource.
2) PUT: To update the resource status.
3) POST: use to create a new resource.
4) DELETE: to delete the resource.

2) CoAP protocol features

CoAP has many features that make it desirable in the Internet of things applications, the following points shows the main features:

- CoAP header size is 4 bytes.
- CoAP is a RESTFUL protocol.
- Support a piggyback message.
- CoAP is a one-to-one and multicast.
- Support send data messages over IPV4 and IPV6 networks.
- CoAP is a simple proxy and caching.
- CoAP is a machine-to-machine interaction (M2M).
- Support synchronous and a synchronous message.
- Use (DTLS) over UDP for security.

Fig. 8 shows CoAP message format

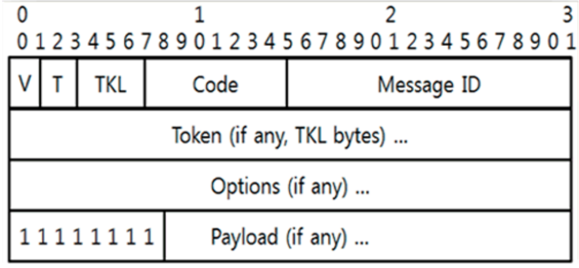

Fig. 8. CoAP message format

3) MQTT (Message Queue Telemetry Transport)

The MQTT protocol is one of the important of the Internet of things protocols. It is light, open, and designed so that it is easy to implement. The simplified design and lightness of this protocol made it an appropriate solution for embedded devices with limited and unlimited resources in both processing and storage capacity. The protocol offers benefits that reduce power consumption and bandwidth, and these are two very important factors in the Internet of things devices. MQTT protocol is based on the principle of (publish/subscribe) to send messages in a very lightweight, which is usually used in situations that require communication over mobile networks such as (GPRS, 3G and LTE) because of the possibility of data transfer using Bandwidth few. MQTT runs over TCP. In addition, MQTT protocol has three basic parts: the server, the publisher, and the subscriber as shown in Fig. 9.

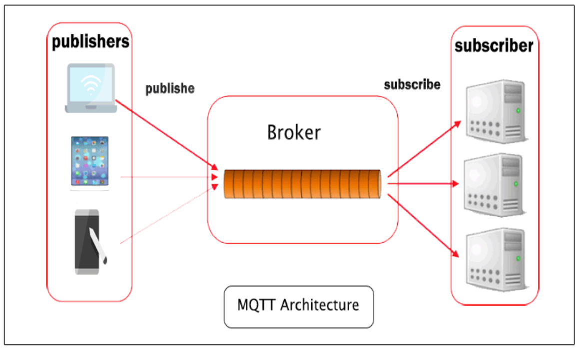

Fig. 9. MQTT architecture

- **MQTT protocol features:**
- MQTT protocol is based on TCP/IP and this lead to more security.
- MQTT header is 2 bytes.
- Support Quality of Service (QoS).
- Easy to integrate with new devices.
- MQTT is a many-to-many communication.

Fig. 10 shows (MQTT format message) and Fig. 11 shows (MQTT stack).

Bit	7	6	5	4	3	2	1	0
Byte 1	Message Type				DUP	QoS		Retain
Byte 2	Remaining Length (Length of options and payload)							
Byte 3	Options							
Byte n								
Byte n+1	Payload							
Byte m								

Fig. 10. MQTT format message

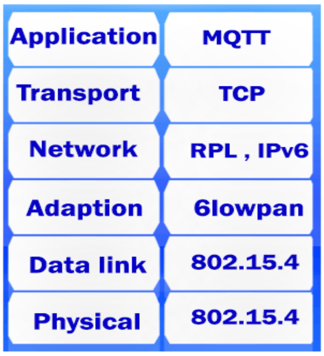

Fig. 11. MQTT stack

4) XMPP (Extensible Messaging and Presence Protocol)

XMPP is a real-time data transmission protocol that is ideal for chatting, audio, video and other applications that require instant data transfer, it uses XML which makes it easily extensible, but it uses for XML lead to more of overhead because much of headers and this increases the energy consumption, supports request/response and publish/ subscribe architecture, but it is not practical for M2M communications, also don't support quality of service (QoS). It should be noted that this technology will not replace HTTP technology but will be on its side. Intermediate servers. This direct contact eliminates unwanted or unauthorized messages. This is one of the security methods for XMPP. It also supports encrypted connections through the use of transport layer security (TLS). With the remarkable increase in the use of smartphones, the applications of the online chat will increase. The XMPP is an important instant messaging protocol. XMPP tests ping at regular intervals to check the current connection status to avoid server idle. The process of ping more battery power of portable devices [25]. It has been shown that XMPP can be used with different devices and networks, and XMPP can be used above UDP rather than TCP. Fig. 12 shows XMPP architecture.

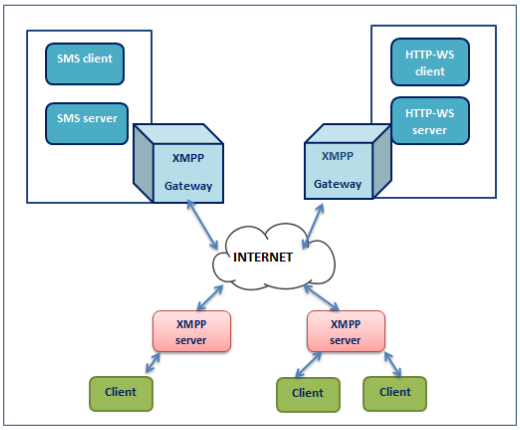

Fig. 12. XMPP architecture

C. Routing Protocols

Some of the features that are considered to be familiar to the Internet of things devices such as (the limitations of energy consumption, cost, memory limitations and the processing of data) these things that dominate all the design requirements. It is therefore, necessary to use the appropriate routing protocol to control the restrictions mentioned. There are many routing protocols in IoT, we discuss some of these protocols:

1) RPL routing protocol

RPL - IPv6 routing protocols for low power and loss network [26]. This protocols type is designed for a network comprising of constraint devices in power computation capability and memory. Thus the data transmission in this type of networks is unreliable and has a low data rate, but high loss rate [27]. The RPL protocol works in the IP layer and this allows routing across multiple layers. That has been standardized by IETF in 2011 to establish a common base low power and loss networks (LLNs). Because of the great similarity between (IoT) and (LLNs) and where ipv6 is an essential feature for (IoT), RPL had become is a protocol for routing in the internet of things networks. The RPL protocol allows for three types of traffic: (multi-peer to peer), (peer to peer) and (peer to multi-peer). RPL forms a tree called a Directed Acyclic Graph (DAGs). Each node in the RPL network has a preferred parent that acts as a gateway to that node, the node directs the packet to its preferred parent and so on until the goal reaches it, if the node has a routing table, the RPL protocol uses control packages (DIS, DIO, DAO) to maintain the tracks and build the tree, where these packages carry information different network [28], [29]. Fig. 13 shows IoT network and RPL. Fig. 14 shows RPL protocol stack.

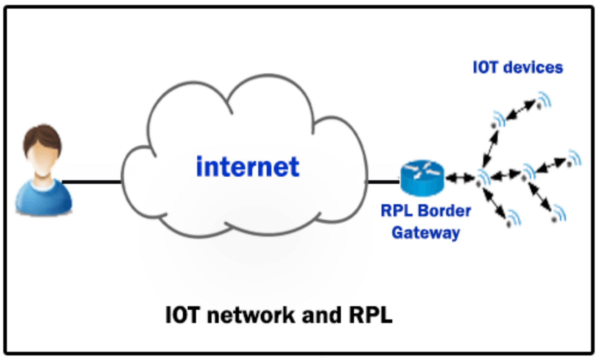

Fig. 13. IoT network and RPL

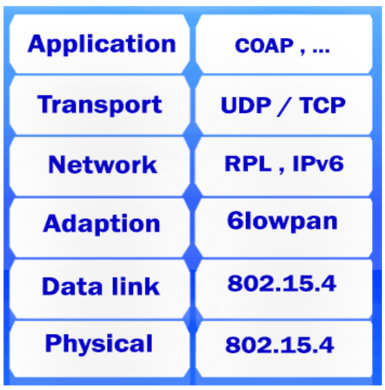

Fig. 14. RPL protocol stack

2) CORPL

Cognitive RPL (CORPL) is a protocol that extends RPL and uses the same DODAG technology, but with a couple of modifications to RPL. First, it introduces opportunistic forwarding which enables the packet to have multiple forwarders set but only the best next hop will be chosen to forward the packet. Then, each node will maintain a forwarding list instead of its parent only and updates its neighbor with its changes using DIO messages, based on the updated information, each node dynamically updates its neighbor priorities in order to construct the forwarders set [31]-[33], and [40].

3) CARP and E-CARP

Channel-aware routing protocol (CARP) is another routing protocol that is based on distributed networks and designed for underwater communication. It is a lightweight packet forwarding protocol and hence, can be applied to IoT systems. It considers historical link quality measurements to select the forwarding route. Network initialization and data forwarding are the two scenarios that should be considered in such protocols. In network initialization, a HELLO packet is broadcasted from the sink to all other nodes in the networks. In data forwarding the packet is routed from the sensor to sink in a hop-by-hop fashion, each next hop is determined independently. The main problem with CARP is that it does not support the reusability of previously collected data. In other words, if the application requires sensor data only when it changes significantly, then CARP data forwarding is not beneficial to that specific application [34]. In [35] an enhancement of CARP was done in E-CARP by allowing the sink node to save previously received sensory data, when new data are needed, E-CARP sends a ping packet which is replied with new data from the sensor nodes. Thus, E-CARP reduces the communication overhead drastically.

D. Why Did We Use These Protocols?

There are many important protocols that are integrated between them to build the Internet of things stack to work in limited energy conditions and processing capacity in addition to reducing the cost of data and these protocols are:

1) 6LoWPAN

6LoWPAN - IPv6 over 802.15.4 [30], is meant to extend IPv6 networks to IoT networks. The advantages of this approach are the possibility of re-using existing IPv6 technologies an infrastructures. However, this type of network is originally designed for computing devices with higher processing capability and memory resources which is not suitable for IoT network entities [27]. IPv6 over low power wireless personal area network (6LoWPAN) is one of the first and extensively used IETF standards in this category. It efficiently encapsulates IPv6 long headers in IEEE802.15.4 small MAC frames, which cannot exceed 128-byte length. 6LoWPAN specifications allow many features including different length addresses, different networking topologies, low bandwidth, low power consumption, cost-efficient, scalable networks, mobility, reliability, and long sleep times. Header compression is used in the standards to reduce transmission overhead, fragmentation to meet the 128-byte maximum frame length in IEEE802.15.4, and support of multi-hop delivery [34].

2) ZigBee

ZigBee It is a wireless protocol defined by layer 3 and above of IEEE 802.15.4 [36]. Is a networking technology designed specifically for applications that require a long-term network connection without the need to provide an electrical card between short periods. This technology has a long battery life, it is also designed for applications that require high availability of service so that there is a direct alternative in case one of the devices has been disabled. It has been developed to be used in smart homes to provide a network connection to cooling and heating devices and other smart home appliances. It consists of several elements: **Coordinator:** Responsible for starting and controlling the network, it also stores information about the network, which includes information about protection and the documented centers of broadcasting. **Routers:** are responsible for dynamically expanding the network, providing a copy of the router's settings and also providing fault tolerance technology, which does not stop other devices if turn off one of the devices. **Devices**: are the only devices that receive and transmit. ZigBee technology supports different methodologies (Star, Cluster-Tree and Mesh) as in Fig. 15.

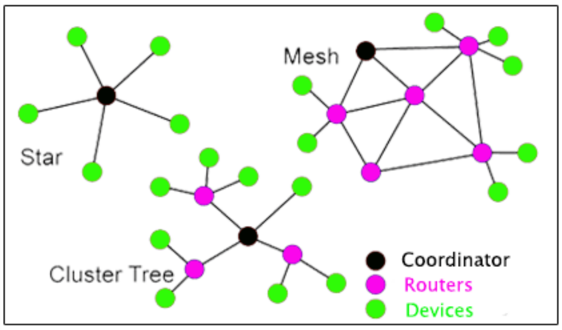

Fig. 15. ZigBee topologies

E. Application Layer Protocols Performance

A number of application layer protocols have been discussed. In this section, we will compare these protocols are in many ways, including the transport layer protocol. The comparison did through the architecture of the protocol, the quality of the service (QoS), security and others as in Table I.

TABLE I: A COMPARE OF IOT APPLICATION LAYER PROTOCOLS

Protocol	RESTful	Transport		Architecture		Security			Support to QoS	Header size (Byte)	Bandwidth needed	
		UDP	TCP	Req-Res	Pub-Sub	TLS	DTLS	SSL			Low	High
CoAP	✓	✓	✗	✓	✗	✗	✓	✗	✓	★★★★	✓	✗
MQTT	✗	✗	✓	✗	✓	✓	✗	✓	✓	★★	✓	✗
XMPP	✗	✗	✓	✓	✓	✗	✗	✓	✗	-	✓	✗

The Table I summarizes several important issues for the three protocols: CoAP, MQTT, and XMPP in terms of safety, transport and service quality.

The CoAP and MQTT protocols are used with devices that suffer from limited resources such as power, processing, and memory, if the REST is required, the CoAP protocol will be the best if not the only one, in addition, this protocol is used UDP, which reduces the delay time and it uses the datagram transport layer security (DTLS) to prevent breakthroughs in the sent data. If there is an application based on XML, the choice XMPP protocol is the best for this, so MQTT and XMPP use TCP and that means less security. MQTT uses (TLS) and (SSL) by a broker, XMPP also used (SSL) between server and client.

1) Fog computing

Is a highly virtualized platform and the Fog a non-trivial extension of the Cloud [37],[38]. The need for this model arose after the proliferation of internet of things IoT and these small devices generate a very large amount of data and the need for analysis capabilities and the response speed is very large by the cloud, in addition to the need for a very wide frequency range, these devices often send data to the cloud and wait for the response to make the decision in its mission. In the case of the fog computing model, these devices send this data to nearby devices, such as smartphones and communicate with them to receive the response in a much shorter time compared to the time required to send data to the cloud, because of the presence of these peripherals in a geographical area very close. They may also be used for security. In a fog computing, the processing is done on a smart device in a smart router or gateway. There are several advantages for fog computing, such as (real-time interactions, low latency, location-awareness, support for mobility, very low delay and Heterogeneity). Applications of fog computing are (Smart Home, Smart Grid, Smart Vehicle Management, Smart Data Management, etc.). Fig. 16 display the three layers (sensing devices of IoT, fog computing and cloud).

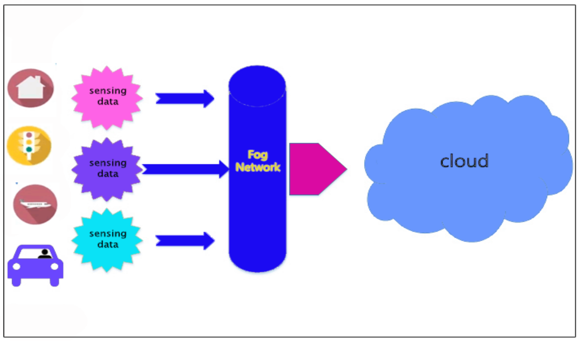

Fig. 16. The three layers (sensing devices of IoT, fog computing and cloud)

VI. EPC - MCS PLATFORM

Nowadays, modern cities have begun to move towards a new approach to improving the life of peoples and thus aim to model smart cities. The process of collecting data from the environment manually is a difficult process. An alternative is used to facilitate the process of data collection via the MCS model. In this paper, we proposed a crowd sensing platform to manage the various sensor operations. Volunteers use their smartphones to collect data from the environment in real time via the camera, GPS, microphone, etc. All this data would be sent to the server. The processes of sensing, collecting data, and sending it to the server lead to two problems: the power consumption of the phone's battery and the cost of data uploading, this article proposed some solutions to reduce these problems.

A. EPC - MCS Structure

In this part of the paper, we suggested the structure for MCS as shown in Figure 18. The proposed structure to reduce the energy consumption and the cost of uploading data. In this structure, we defined the user as node **Nu**, where this **Nu** is the main part of MCS, the nodes can move freely in any zone where supposed that any zone has many types of infrastructure and areas such as (houses, establishments, civilian places, etc.), therefore, we divided the nodes into two types:

Type I: In this type nodes are smartphone devices used by participants to collect various data from the environment through (camera, microphone, temperature, wearable devices or etc.) or through connecting the phone with fixed point sensors via Bluetooth. This type of node called the global mobile node $\mathbf{N_u^G}$. The job of this type to connect each other to moving the data to the type II node (as explained below), without uploading the data to the data center.

Type II: the devices in type II are the mobile devices, computer, tablet, as well as the smartphone or any other devices, can be recharged for presence this device at home, school, health center or any building where the energy is available to recharge the battery easily, and using the internet freely. This type of node called the local mobile node $\mathbf{N_u^L}$. The job of this node to uploading the data to the data center with no power consumption and no cost required to lose.

Routing mechanism: This part shows the mechanism for any data from the sensing to the uploading. The nodes $\mathbf{N_u^G}$ connect with others via Bluetooth and Wi-Fi if it is required, where they send the collecting data to the nearest node if there is $\mathbf{N_u^L}$ node in the neighbour, a routing protocol between the $\mathbf{N_u^G}$ to deliver the information to the nearest node based on the RPL protocol to choose the best route which contains the least number of hops, to reach to the $\mathbf{N_u^L}$ node. The $\mathbf{N_u^L}$ collects the information from $\mathbf{N_u^G}$ to uploading it to the data center using a 3G, 4G via peggybag or Wi-Fi .

B. Layers and Protocols

As mentioned above, general IoT stack is consist of a number of layers where each layer contains a set of protocols. In this part we suggested layers and protocols as shown in Fig. 17 and choice one protocol from each layer that supports low power consumption.

In the application layer, we chose a CoAP protocol for its high specifications in reducing the energy consumption in the devices that run on the battery as well as using the piggyback with the 3G connection, which reduces the cost of data transfer as detailed above. In the transport layer will be UDP protocol, where CoAP protocol compatible with UDP protocol well. UDP that helps to avoid handshake before data transmission and errors detection. The UDP header size is small compared with TCP all this lead to efficient power consumption. In the network layer, we will use ipv6 where the increased number of devices connected to the internet lead to design ipv6 as an alternative to ipv4, where it provides a very large number of addresses, it also offers new features that give the ability to install easier for devices, improve security and avoid barriers of (NAT). All of these features lead to reduced energy consumption. The physical layer is specifies the physical properties of the network such as (frequency, timing, and voltage). IEEE 802.15.4 is a transmission mechanism and standard radio technology for low power networks (Fig. 18).

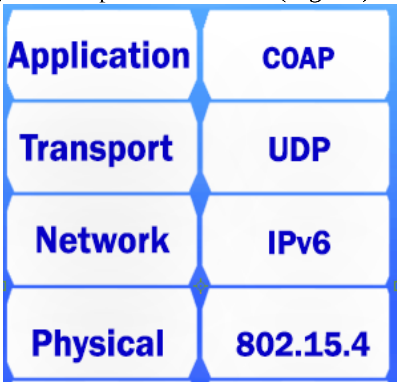

Fig. 17. Proposed layers and protocol

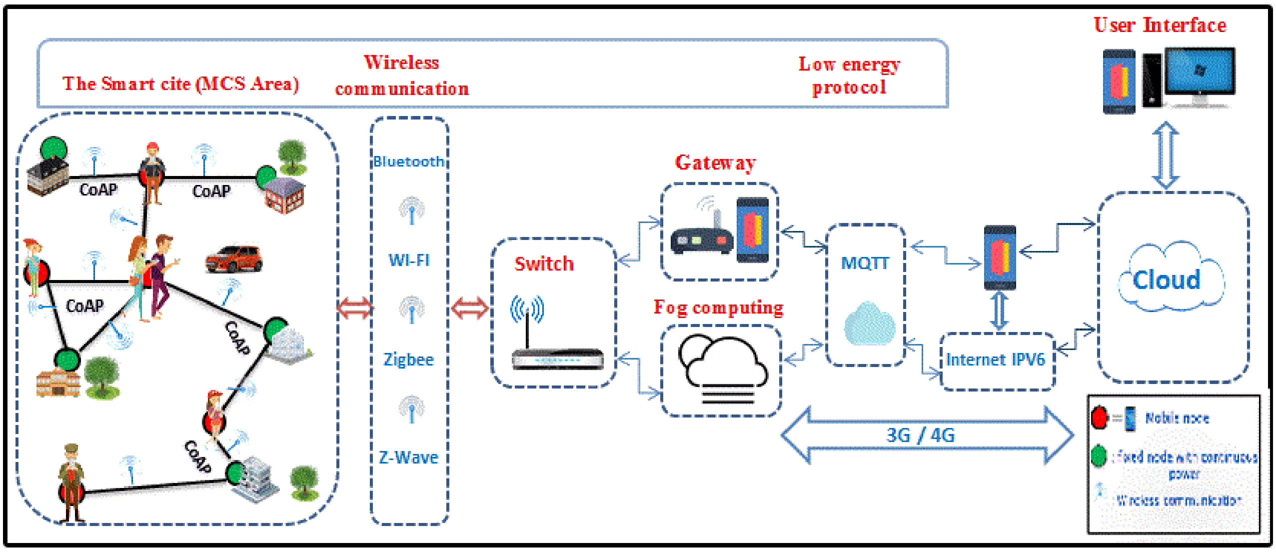

Fig. 18. Proposed structure (EPC)

C. *Proposed Algorithm 1: for Efficient Power Consumption and Routing Data in MCS Area (EPC and RD):*

Ni node = mobile node
Nj node = neighbor node of Ni
BS = base station = switch

a. Main Algorithm

begin
(x,y) detection
If the node is fixed then BS
else is Ni node
End

b. Node sensing (mobile node)

begin
{
Ni Node deployment
For node Ni do
{
Get (x,y) position
Get initial energy
Get the ID of Ni node
Active bluetooth
}
}

c. Sending information

Ni node begins sensing info. when BS request that (BS request data each 30 s)
for node Ni do
Nj ⟵ neighbor node of Ni
Calculate distance from Ni to (BS && Nj)
If Nj node nearest to Ni then
Ni sends information to Nj with(ID of Ni node ,energy remaining,position,time)
else

Ni sends information to BS with(ID of Ni node, energy remaining, position, time)
If Ni node energy less than 50% then
BS send request information each 60 s

d. switch data for FOG or Gateway

When BS received information from Ni node
If information with fast query then
BS route information to Fog
In Fog The reply includes in a packet of info. and send back through BS to same Ni node
else
Route information to Gateway
Gateway upload information Vie Wi-Fi or 3G call to cloud
end

VII. Protocols Implementation

This section we implemented the well-known IoT protocol to show the main difference between them. The implementation did by Cooja simulator under Contiki OS. Contiki is an open source system where it is designed for resource-constrained devices and the Internet of things. Cooja is a network simulator designed for the Contiki operating system, Cooja simulator, which supports using C language. This article used Cooja to simulate different protocols under different conditions and different topologies also it discussed the results.

TABLE II: GENERAL SIMULATION PARAMETERS

Parameter	value
Operating System	Contiki 2.7
Radio Environment	Unit Disk Graph Medium (UDGM)
Number of nodes	10
Node transmission range	50 m
Node carrier sensing range	100 m
TX/Rx ratio	100 %
Time	600 s

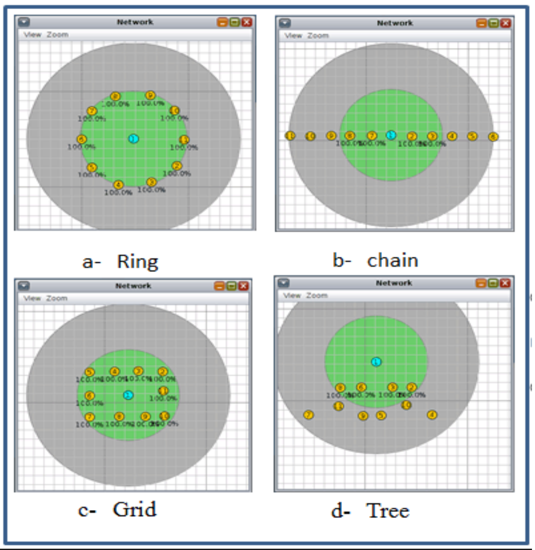

Fig. 19. The four network topologies (Ring, Grid, chain and Tree). The blue Node is a sink node, yellow nodes are sender nodes.

We have simulated the RPL protocol with different topologies such as (Ring, Grid, chain, and Tree) as shown in Fig. 19. In each of them, the network consists of 10 nodes and one sink, but the sink site is different for each one. The goal of this experiment is to know the variables such as energy consumption, radio duty cycle and other measures. Where the average energy consumption can be defined as: Average Power Consumption = (Transmit + Listen + CPU power +LPM.

In Fig. 20, we see more power consumption in which the sink is in the center (Ring topologies in Fig. 19, a) compared to other topologies, where there was the highest number of hops ranging from 1 to 6 hops, which increases the consumption of energy. In chain topologies (Chain topologies in Fig. 19, b) also we note that there is a high consumption of power, but less than the topology of the ring, where it has a hops rate of between 1 and 3. Either in Grid topologies (Grid topologies in Fig. 19, c), we find that the average of power consumption is balanced and better than the other topology's. Tree topologies (Tree topologies in Fig. 19, d), is the best power consumption, where the average of hops did not exceed tow hops. The nodes that consume a lot of energy are either to be far from the sink or be in the way of another nodes used to deliver the data to the sink.

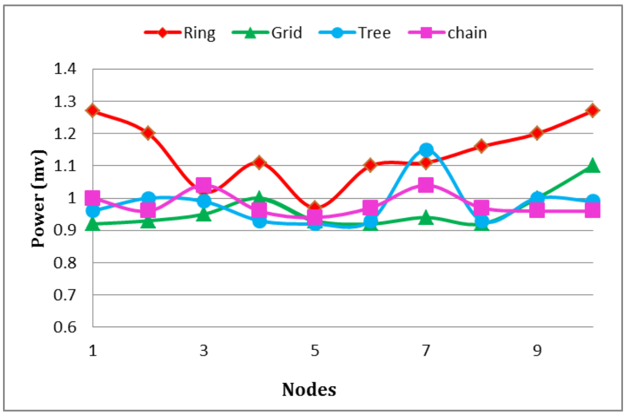

Fig. 20. Average power consumption

Power consumption is important for wireless sensor nodes in low-power sensor networks, radio transceiver must be turned off as much as possible to avoid work and this is done by Radio Duty Cycling (RDC), RDC is a power management tool depending on the operating cycle and to ensure long node and network lifetime. When the radio cycle is stopped, the nodes will not be able to transmit and receive from the neighbors so that they can set up a schedule in which the protocol will run the radio at a specified time and then the neighboring nodes can send the packets in a timely manner. Figure 21 shows the average radio duty cycle between the four topologies. The working period of the nodes in a ring topology is more than others topology where the nodes will not go to sleep only a little time, either in the grid topologies network the nodes will sleep and wake up when needed and this provides more energy.

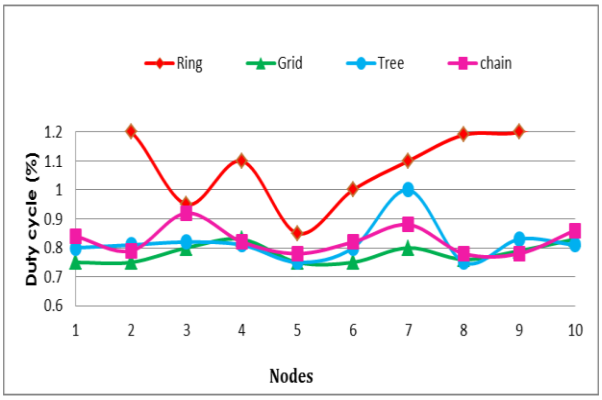

Fig. 21. Average duty cycle

The CoAP protocol is considered to be one of the most important protocols designed specifically for sensors devices with limited resources, especially in power, from its features is used(UDP) instead of (TCP) and small head size and more, 6lowpan technology has made it possible to connect the devices and small things to the Internet, IPv6 technology through 6LoWPAN and the IPv6 routing protocol for low power and loss networks (RPL) have made wireless sensor networks (WSNs) integrated with smart objects over the internet.

We performed a simulation of eight nodes. Fig. 22 shows a comparison of the average power consumption for the three protocols (CoAP, 6lowpan, and RPL), showing the superiority of the CoAP protocol on the rest of the protocols.

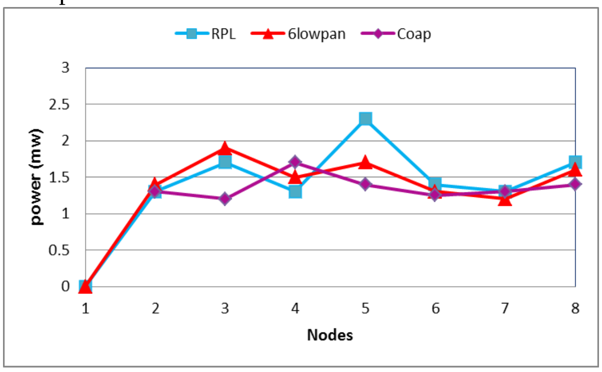

Fig. 22. Average power consumption For (CoAP, 6lowpan, and RPL)

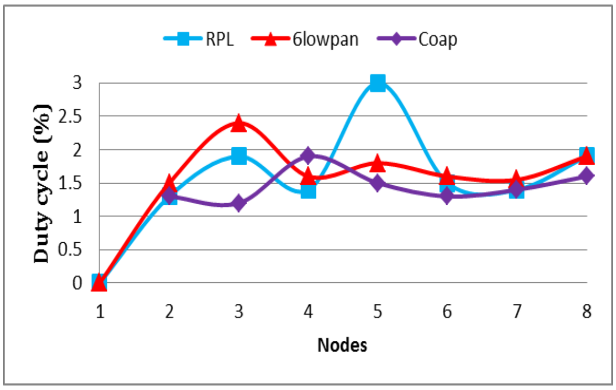

Fig. 23. Average duty cycle For (CoAP, 6lowpan, and RPL)

As for the average duty cycle, Fig. 23 shows a clear superiority of the CoAP protocol. This technique (duty cycle) makes the nodes sleep when they do not need to be sent or received data and wake up for any new operation, the aim of this is to save power.

VIII. Conclusion

The term Internet of things refers to millions of devices connected to the internet where it is considered one of the biggest developments in the current time. With the proliferation of smartphones are rich in sensors, MCS model having many applications in smart cities such as (weather, smart home, traffic congestion, etc.). On the other hand, there are two challenges in MCS, energy consumption and the cost of data upload. In this paper, we have presented (EPC - MCS) a framework looks at how to efficient power consumption and the cost of uploading data in the MCS, the proposed framework, it includes dividing the MCS into two areas, global and local, in the global area where volunteers collect data via their smartphones and send them to the nodes in the local area that existing in a home , health care , school or any building where the electric power is continuous to be transferred to the fog when need to make a quick decision or to the gateway before transferred to the cloud, where we used CoAP protocol, which proved its efficiency in energy consumption, this framework also proposed an algorithm to route data in the area of the MCS to reduce energy consumption and without the cost of data upload. The experiment results verified the efficiency and effectiveness of (EPC - MCS) for power consumption and data uploading in mobile crowdsensing.

References

[1] H. Jayakumar, A. Raha, Y. Kim, S. Sutar, W. S. Lee, and V. Raghunathan, "Energy-efficient system design for IoT devices," in *Proc. 21st Asia and South Pacific Design Automation Conference*, 2016, pp. 298–301.

[2] Z. Abbas and W. Yoon, "A survey on energy conserving mechanisms for the internet of things: Wireless networking aspects," *Sensors*, vol. 15, no. 10, pp. 24818–24847, 2015.

[3] M. S. Mahmoud and A. A. H. Mohamad, "A study of efficient power consumption wireless communication techniques/modules for internet of things (IoT) applications," 2016.

[4] R. K. Ganti, F. Ye, and H. Lei, "Mobile crowdsensing: current state and future challenges," *IEEE Communications Magazine*, vol. 49, no. 11, 2011.

[5] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4W1H in mobile crowd sensing," *IEEE Communications Magazine*, vol. 52, no. 8, pp. 42–48, 2014.

[6] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Ear-phone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE International Conference on Information Processing in Sensor Networks*, 2010, pp. 105–116.

[7] M. Musolesi, M. Piraccini, K. Fodor, A. Corradi, and A. T. Campbell, "Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones," in *Proc. International Conference on Pervasive Computing*, 2010, pp. 355–372.

[8] S. Chessa, M. Girolami, L. Foschini, R. Ianniello, A. Corradi, and P. Bellavista, "Mobile crowd sensing

management with the ParticipAct living lab," *Pervasive and Mobile Computing*, vol. 38, pp. 200–214, 2017.

[9] S. Chessa, M. Girolami, L. Foschini, R. Ianniello, A. Corradi, and P. Bellavista, "Mobile crowd sensing management with the ParticipAct living lab," *Pervasive and Mobile Computing*, vol. 38, pp. 200–214,

[10] X. Zhang *et al.*, "Incentives for mobile crowd sensing: A survey," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 54–67, 2016.

[11] Y. Wang *et al.*, "A framework of energy efficient mobile sensing for automatic user state recognition," in *Proc.e 7th International Conference on Mobile Systems, Applications, and Services*, 2009, pp. 179–192.

[12] G. Cohn, *et al.*, "An ultra-low-power human body motion sensor using static electric field sensing," in *Proc. ACM Conference on Ubiquitous Computing*, 2012, pp. 99–102.

[13] F. Rabee, "Efficient transfer route for mobile crowd sensing (ETR-MCS)," *JCM*, vol. 13, no. 4, pp. 162–168, 2018.

[14] M. R. Ra, B. Priyantha, A. Kansal, and J. Liu, "Improving energy efficiency of personal sensing applications with heterogeneous multi-processors," in *Proc. ACM Conference on Ubiquitous Computing*, 2012, pp. 1–10.

[15] L. Wang, D. Zhang, Z. Yan, H. Xiong, and B. Xie, "effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 45, no. 12, pp. 1549–1563, 2015.

[16] P. P. Jayaraman, J. B. Gomes, H. L. Nguyen, Z. S. Abdallah, S. Krishnaswamy, and A. Zaslavsky, "Cardap: A scalable energy-efficient context aware distributed mobile data analytics platform for the fog," in *Proc. East European Conference on Advances in Databases and Information Systems*, 2014, pp. 192–206.

[17] J. K. Nurminen, "Parallel connections and their effect on the battery consumption of a mobile phone," in *Proc. 7th IEEE, Consumer Communications and Networking Conference*, 2010, pp. 1–5.

[18] H. Xiong, D. Zhang, L. Wang, J. P. Gibson, and J. Zhu, "EEMC: Enabling energy-efficient mobile crowdsensing with anonymous participants," *ACM Transactions on Intelligent Systems and Technology*, vol. 6, no. 3, p. 39, 2015.

[19] E. Soroush, K. Wu, and J. Pei, "Fast and quality-guaranteed data streaming in resource-constrained sensor networks," in *Proc. 9th ACM International Symposium on Mobile ad Hoc Networking and Computing*, 2008, pp. 391–400.

[20] A. Capponi, C. Fiandrino, D. Kliazovich, P. Bouvry, and S. Giordano, "A cost-effective distributed framework for data collection in cloud-based mobile crowd sensing architectures," *IEEE Transactions on Sustainable Computing*, vol. 2, no. 1, pp. 3–16, 2017.

[21] W. Sherchan, P. P. Jayaraman, S. Krishnaswamy, A. Zaslavsky, S. Loke, and A. Sinha, "Using on-the-move mining for mobile crowdsensing," in *Proc. IEEE 13th International Conference on Mobile Data Management (MDM)*, 2012, pp. 115–124.

[22] 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, 2017.

[23] D. Gesvindr, J. Michalkova, and B. Buhnova, "System for collection and processing of smart home sensor data," in *Proc. IEEE International Conference on Software Architecture Workshops (ICSAW)*, 2017, pp. 247–250.

[24] M. Asim, "A survey on application layer protocols for Internet of Things (IoT)," *International Journal of Advanced Research in Computer Science*, vol. 8, no. 3, 2017.

[25] K. K. Guduru, S. Dev, and R. H. Naganur, "Mitigating power consumption in mobile devices with dynamic triggering of XMPP ping requests," in *Proc. IEEE 84th Vehicular Technology Conference (VTC-Fall)*, 2016, pp. 1–7.

[26] T. Winter, *et al.*, "RPL: IPv6 routing protocol for low-power and lossy networks," 2012.

[27] T. T. T. Thuy, "Routing protocols in Internet of Things," *Aalto University*.

[28] W. Xiao, J. Liu, N. Jiang, and H. Shi, "An optimization of the object function for routing protocol of low-power and lossy networks," in *Proc. 2nd International Conference on Systems and Informatics (ICSAI)*, 2014, pp. 515–519.

[29] H. S. Kim, J. Ko, D. E. Culler, and J. Paek, "Challenging the IPv6 routing protocol for low-power and lossy networks (RPL): A survey," *IEEE Commun. Surv. Tutor*, vol. 19, no. 4, pp. 2502–2525, 2017.

[30] Z. Shelby and C. Bormann, *6LoWPAN: The Wireless Embedded Internet*, vol. 43. John Wiley & Sons, 2011.

[31] Aijaz and A. Aghvami, "Cognitive machine-to-machine communications for internet-of-things: A protocol stack perspective," *IEEE Internet of Things Journal*, vol. 2, no. 2, 2015, pp. 103-112

[32] Aijaz, H. Su, and A. H. Aghvami, "CORPL: A routing protocol for cognitive radio enabled AMI networks," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 477–485, 2015.

[33] A. Aijaz, H. Su, and A. H. Aghvami, "CORPL: A routing protocol for cognitive radio enabled AMI networks," *IEEE Trans. Smart Grid*, vol. 6, no. 1, pp. 477–485, 2015.

[34] T. Salman and R. Jain, "A survey of protocols and standards for internet of things," *Advanced Computing and Communications*, vol. 1, no. 1, 2017.

[35] S. Basagni, C. Petrioli, R. Petroccia, and D. Spaccini, "CARP: A channel-aware routing protocol for underwater acoustic wireless networks," *Ad Hoc Networks*, vol. 34, pp. 92–104, 2015.

[36] N. N. Srinidhi, S. M. D. Kumar, and K. R. Venugopal, "Network optimizations in the Internet of Things: A review," *Engineering Science and Technology, an International Journal*, 2018.

[37] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proc. First Edition of the MCC Workshop on Mobile Cloud Computing*, 2012, pp. 13–16.

[38] S. Yi, Z. Hao, Z. Qin, and Q. Li, "Fog computing: Platform and applications," in *Proc. Third IEEE Workshop on Hot Topics in Web Systems and Technologies (HotWeb)*, 2015, pp. 73–78.

[39] N. D. Lane, E. Miluzzo, H. Lu, D. Peebles, T. Choudhury, and A. T. Campbell, "A survey of mobile phone sensing," *IEEE Communications Magazine*, vol. 48, no. 9, 2010.

[40] H. P. Alahari and S. B. Yalavarthi, "A survey on network routing protocols in Internet of Things (IOT)," *International Journal of Computer Applications*, vol. 160, no. 2, 2017.

Furkan Rabee is a one of staff member in Computer Science department / College of Computer Science and Mathematics/University of Kufa. He received his B.S. degree in computer engineering and M.S. degree in computer engineering with computer network major from AL- Nahrian University, in 2000 and 2008 respectively. He got his Phd from School of Computer Science and Engineering, UESTC, Chengdu, China 2015. His research interests include real-time scheduling algorithms, real-time locking protocols, real-time operating systems, Computer Network, and Wireless Sensor Network.

Abbas M. Ali Al-muqarm. He received his B.S. degree in computer science from University of Kufa, in 2013 .He is now a student of M.S. in Computer Science/College of Computer Science and Mathematics/University of Kufa . His research interests in,Internet of Things (IoT), Mobile Crowd Sensing (MCS), Computer Network and Wireless Sensor Network.