Efficient Transfer Route for Mobile Crowd Sensing (ETR-MCS)

Furkan Rabee
Computer Science Department, Faculty of Computer Science and Mathematics, University of Kufa, Iraq
Email: Furqan.rabee@uokufa.edu.iq

Abstract **—**The smartphone became a necessity in the personal life, and this device invoked in Mobile Crowd Sensing MCS to cover the smart cities requirements. There are two challenge factors with MCS are energy consumption and data uploading cost. This paper presented a new framework to manage these two challenge factors, this framework called "Efficient Transfer Route for Mobile Crowd Sensing (ETR-MCS)", this framework began with clustering the participant users with two areas, master area, and slave area, the users in slave area have two responsibilities, first, collect the data from the environment, second, the users would build a transfer route called "three-way handshaking route" to transfer the data to the users in master area. The users in Master area will be responsible to upload the data to the server with low energy and zero data uploading cost, this gain got from choosing the best clustering areas, where the master area prefer to choose the users how are at home or in work, therefore; they will upload the data using Wi-Fi or piggyback with 3G communication with considering the priority to the message should be transferred or upload first. This framework also presented precise equations for "time of sense", "time to transfer data" for the user in slave area, also give equations for energy consumption by the participants in both areas. The evaluations to this framework shows outperform of our proposed framework with compare to the previous articles in Mobile Crowd Sensing.

Index Terms**—**Wireless sensor network, mobile crowd sensing, clustering, three-way handshaking.

I. Introduction

As sensor-armed smartphones, that technique gets more popular [2], mobile crowd sensing [11], [3] has become an effective way to carry out various sensing tasks such as environmental monitoring [13] and social sensing [6]. To encourage users to participate in mobile crowd sensing tasks, should reducing the unconvinced afford for the users. In this regard, there are two critical concerns; energy consumption, and mobile data cost. While energy consumption is related to a mobile phone's battery life, mobile data cost is associated with the fees incurred, especially for the users who do not hold an unlimited data plan. Therefore, reducing energy consumption and data cost incurred can encourage more people to actively participate in crowd sensing tasks. Many papers have developed several energy-saving approaches for attracting engagements in mobile crowd-sensing. Some studies gave a solution by proposed four-stages life cycle to the mobile crowd sensing (i.e: task creation, task assignment, task execution, and crowd data integration)[3]. This paper proposes a solution by adding a priority to the task to reduce the cost and the network traffic congestion. A trade-off Making between local and remote computation [6], reducing data uploading frequency by predicting missing data on the server side [14], and splitting the task intelligently among users, etc. These existing works mostly assume that the sensed data should be sent to a central server as soon as the data is produced. in [2] supposed some mobile crowd-sensing tasks do not necessarily require the sensed data to be uploaded in real time [2]. So this paper presents a framework to send the data with a tolerant delay of time to reduce energy consumption, and mobile data cost.

Manuscript received December 22, 2017; revised April 3, 2018.
doi:10.12720/jcm.13.4.162-168

II. Related Works

A. Mobile Crowd Sensing

There are many studies for mobile crowd sensing which lead many applications such as application to sense and compute devices collectively share data to measure and map phenomena of common interest [11], proposed four-stage life cycle to characterize the mobile crowd sensing process[3], and urban noise monitoring [13]. This application enabled by the participatory management to crowd sensing frameworks [4], the strategies for data uploading [14], etc., have been studied.

B. Energy Conservation in Mobile Crowdsensing

An intelligent algorithm for task assignment to reduce the number of sensing participants could minimize the total energy consumption in a mobile crowd sensing task. For instance, to select the minimum number of participants to cover a restricted area, many mechanisms are proposed based on different task models [4], [1]. The individual devices in mobile crowd sensing, energy saving mechanisms have been produced by three stages: sensing; computing; and data uploading. The first two stages, sensing, and computing, are responsible for the process of acquiring and processing data from sensors. To reduce the energy consumption of sensing, existing approaches include adopting low power sensors [7]. To reduce the energy consumption of computing, mobile sensing systems can use low power processors [8], etc. While the proposed work focuses on the three phases

©2018 Journal of Communications

sensing, routing, and uploading, most of the above-mentioned approaches can be combined into with additional modification in our protocol, to make crowd sensing more energy efficient. To reduce the energy consumption in data uploading, we proposed using low power communication methods [5] where the protocol turn to get the benefit from Bluetooth, WiFi, or parallel data uploading with voice calls [15] as energy- efficient communication methods. This field made the strategy of this paper to dispensing the direct communication to the mobile stations by all participants, instead of that using the Wi-Fi and Bluetooth to route the date to specific uploader participators. In [2] using the carry and forward data (relays between sensing devices) to the server, e.g.

Compressing data before uploading [16], or uploading part of the data while assuming the rest [14]. These approaches consume extra energy in computation, so the decision to apply these techniques needs to be studied carefully to see the overall performance.

Data Cost Conservation in Data Uploading To decrease data cost in uploading, previous works focused mainly on reducing data size via additional computation on the local mobile devices to aggregate sensed data before uploading [6]. Moreover, the energy-saving work such as [16] also aims at reducing the amount of the data to upload. While these existing approaches can reduce data cost significantly when the data size is big, they cannot eliminate it completely. This paper, the process to upload the data doing by specific users who will be using Wi-Fi and Bluetooth, thus cost data uploading will be zero.

III. Problem Statements

In this section, we showed the problem statement which takes the interesting of the researchers, which is how to sense, compute, and upload the information with low cost and low energy consumption. The previous studies which designed a crowd sensing applications (reality mining [6] and environment monitoring [1]) do not need to upload the sensed data immediately after it is sensed and collected. Such applications allow some delay (a max tolerable amount of delay d_{max}) between collecting the data from sensors and uploading it to the server, i.e., the sensed data generated at t_0 on a user's phone can be uploaded during $[t_0, t_d]$ (where $t_d = t_0 + d_{max}$). Other researchers suppose there are two plan areas, non-data-plan users(NDP) and data plan users(DP), to design delay-tolerant data uploading schemes that can minimize mobile data cost for NDP and decreased energy consumption for DP [19]. From above articles, we proposed a new formwork which supposed two main principles, first; that should be a delay from sending to upload as period $[t_0, t_d]$ to depend more on zero-cost data, second; dividing the sensing areas into many areas to decrease the energy consumption. The next sections explained more detail about our framework.

IV. Efficient Transfer Route for Mobile Crowd Sensing (ETR-MCS)

To give a detail description to ETR-MCS , we divided the explanation to the objective parts, as follows:

A. Mobile Sensing area Description

This part shows the detail description to the mobile sensing area to the ETR-MCS, and all the relationships between the participant's users and the environment, and we described it by many definitions, figure 1 shows the mobile sensing area to the ETR-MCS.

Definition 1:

Supports different types of smart equipment can engage with this framework such as mobile phones, tablets, smart camera (with Wi-Fi, infrared or Bluetooth, etc…), to transfer different type of massages such as images, text, voices,..etc. All smart equipment, we called it **nodes**.

Definition 2:

ETR-MCS starting by clustering the nodes, where this framework clustering the nodes to two major areas, Master area, and slave area, where the clusters should organize as a star topology. There are two types of clusters:

Master cluster: This cluster M_C construct when the user is at home or work, thus; the user can upload with zero cost by Internet connection and can charge the battery easily. It usually is a unique cluster.

Slave cluster: To complete the design to the sensing area, must be a slave clusters, the number for such that cluster will be at least two clusters(S_C>=2). This design of clustering NP-problem avoided because it is use star topology. This design with Master area and Slave area to reduce the data cost uploading, and power consumption.

Definition 3: all users controlled by an application to make the users working by two modes:

Maser Mode: the application will turn the users to this mode just they enter to the M_C and they will be just Data Uploader Users (U_{mu}).

Slave mode: the users in this mode should be Data Sensing (U_{sc}).

Definition 4: All users' equipment should be identifying by GPS to know which area the user will be, thus; to turn to which mode should to be.

Definition 5: in this paper we will use concept "Node" for the participant user. There are two types of nodes; active nodes, and inactive nodes or sleep nodes.

B. The Life Cycle for =ETR-MCS

The life cycle for mobile crowd sensing MCS application was presented by researchers[3][10] ,one study presented three tasks(creating, assignment, and execution) to complete the life cycle for MCS[10], while another study presented four(creating task, assigning task, execution task, data integration task) tasks to complete the life cycle of MCS [3]. Inspire with these articles, we proposed our life cycle to MCS according to our design:

1. Task creation: The organizer for MCS creates an MCS task to the nodes by mobile sensing applications that supported to the users to their smartphones. It is such that application developed by MEDUSA framework [10].
2. Task assignment: after creating sensing task by MCS organizer, the second stage is task assignment. Employing the nodes and assigning them with individual sensing tasks that are supposed to run in each participant's mobile device [3]. In this framework choosing the participants to be as sensing user U_{sc} or upload U_{mu} user depending on their areas. Where the proposed algorithms for this paper design to be responsible to turn the user from mode to mode automatically according to the node location by GPS.
3. Task execution: Once receiving the assigned sensing task, a participant would try to finish it within a pre-defined MCS task duration in parallel with other tasks [3]. In our proposal, we have two types for task execution, each one complete the other. And they as follows :

1) Task preparation. (Sensing, Orientation, and Sending)

This task for U_{sc} node, where this type of nodes working in slave mode as sensing user, where this user does three sub-tasks:

Sensing: node here sensing the environment by using camera, infrared, …etc, and this user might connect with the outside sensors to sense the changing happen with the environment such as noise, pollution ..etc, this connection do by Bluetooth communication.

Orientation: after completing the sensing information, according to our proposal, the proposed system should make a complete route from U_{sc} to the U_{mu}, to upload information by Wi-Fi communication only. Where Wi-Fi communication doesn't use in sensing process. To build such that route, we proposed a three-way handshaking route to organize that communication, it explained later in this paper.

Sending: after make completing the route from U_{sc} to U_{mu}, sending the sensing information to upload under master cluster M_C.

2) Task computing. (Collecting, Uploading, Integration).

This task does under Master cluster by U_{mu} , this node should upload the data to the data server center by two ways: **first**, by Wi-Fi (because of the user in this cluster in home or work); where this way is zero-cost data uploading. **Second**, upload via piggy-back with 3G mobile communication. This way also free and zero-cost data uploading.

Fig. 2 shows an example to illustrate the routing data with ETR-MCS from sensing data to the uploading.

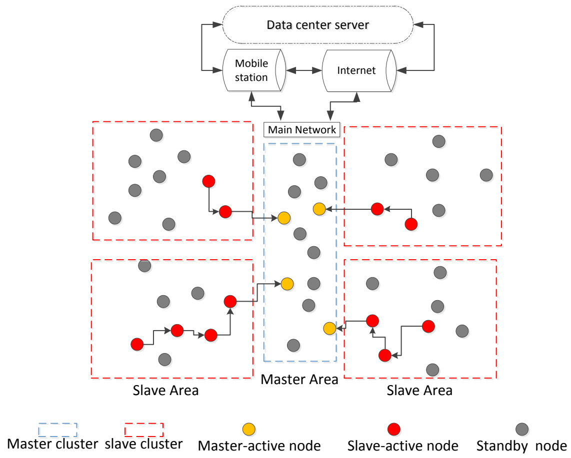

Fig. 1. Mobile sensing areas

3) The three-way handshaking route

Before explaining the three-way handshaking route, we explained the hidden terminal problem. The problem supposed that, If node A and node C both want to send to node B, but they cannot hear each other, they are both going to sense the carrier to be idle and start sending, which leads to a collision at B, This problem is addressed by the Medium Access with Collision Avoidance MACA protocol [20]. The MACA used to send information from node to node. In this article, we used it to open route to sending a message. Such that protocol using it with the nodes under slave cluster U_{sc} to send the data to the nodes under U_{mu} by depending on its neighbor under slave cluster. Starting the process by sending request from the sender to other nodes to open the route, and he will get the response from any node will receive the request in mobile crowd sensing. This protocol called **three-way handshaking route**. If we suppose that each node U_{sc} will send a request to each other in slave cluster to transfer data to upload node U_{mu} in the master cluster, it should be one route will be open from one node type U_{sc} to one node type U_{mu} in time. And this route will build by other nodes in slave cluster and master cluster as shown in Fig. 2. The node would get the route and will be ready to send the data we called it **occupied** the route. And to explain more about building the route and its requirement, we starting by the node which want to transfer own data. This node start sending the Request to Send RTC to any neighbor nodes, and with RTC should send another information:

Priority **P** : if the node wants to send an information must give the priority for this information according to the importance of this information. This P has two values: H: high , L: low.

Weight W: the distance from the requesting node to the node which has to make the connection with it. W: refer by integer number and it calculated by meters or could use the edge of the requested node.

After that each request will be in "First in First Serve" queue, Fig. 3 shows three-way handshaking route protocol. More detail about building the route and its equations would come later in this paper (Rule3).

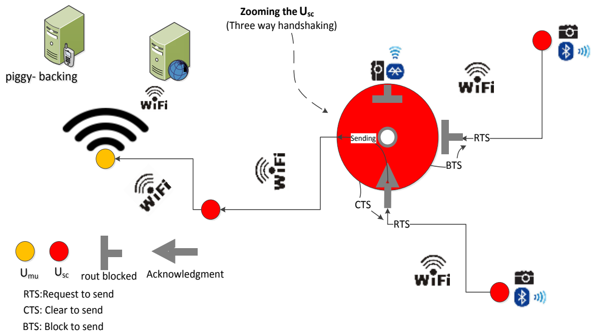

Fig. 2. TR-MCS routing data between nodes

4) Data upload schemes: In this section, we described the methods to upload data. From our topology, supposed the master cluster, will choose the users how are in working place or at home, where the users in this cluster will use the Wi-Fi.

The users in master cluster U_{mu} will be in Upload mode, it is mean he will upload the received information from U_{sc} to upload it , where this process doing by two ways:

a. Upload the data by Wi-Fi.
b. Upload the data parallel with voice call via 3G communication. So these two ways would be zero cost data uploading.
c. Some date is not important and its little big could upload with midnight because most mobile companies give a discount and free minutes after midnight, thus could use this minuets to upload the data.

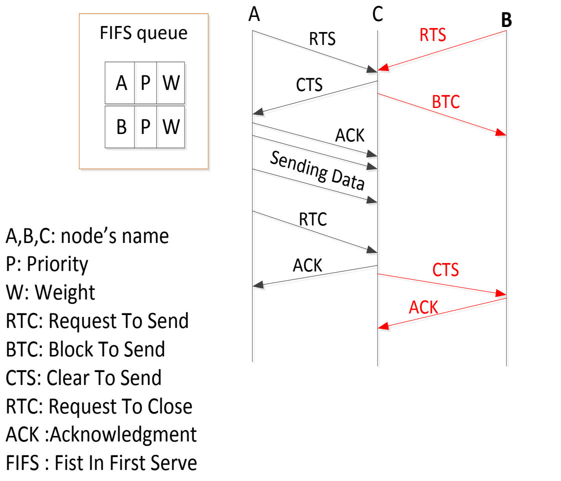

Fig. 3. Three way hand shaking rout

V. Data Cost Uploading and Energy Consumption Rules

To get a perfect performance for MCS, there are two problems should be handled, energy consumption and data cost uploading. In this paper we found a relationship between "time to transfer" and "energy consumption", so we classified the work as follows:

1. **The time consuming by transfer:** to drive a specific time for transferring the data, we submitted the following rules:

Rule1: to sense an information from the environment, there is a time for sensing that information, this time repeated frequently by MCS organizer, we called it *Time to sense* (T_s) this time may detect an event (E_s).

Rule2: as we mentioned before about the life cycle for **ETR-MCS** under task preparation, the node should do an orientation route before send; In this framework, all nodes (U_{sc}) before sensing the data (it mean before T_s), should to check the possibility for making the orientation route. Why do that? Because if such that checking happened, will know the active route A_{route} (the route from U_{sc} to U_{mu}), thus; the nodes (U_{sc}) included the A_{route} after sensing the information will be in active mode, otherwise, the nodes don't include A_{route} will be in inactive or sleep mode. So this time should happen before T_s by time t_o, and it is called pre-sensing time($T_{pre\text{-}s}$),

$$T_{pre\text{-}s} = T_s - t_o \tag{1}$$

Just t_o is finish T_s start with time =0.

Example: if T_s start every one hour, and if we suppose the initial sensing begin at 08:00:00 o'clock, t_o=20 second, so $T_{pre\text{-}s}$=07:59:40 o'clock. The next T_s will begin at 09:00:00 o'clock, and so on.

Rule3: time sharing: when the pre-sensing is initialized, all U_{sc} send RTS to all neighbored nodes to build the route to the U_{mu} , and waiting for the replay, if the replay is CTS, it means the route is open, else the replay is BTS and other information to share the time with other blocked node for sending the information to the U_{mu}. The period of time sharing should be between the current T_s to the next T_s depending on the priority of the task, after that each U_{sc} will turn to inactive mode until its own specific time-sharing. To calculate the time sharing between nodes, this is as follows:

The total time sharing

$$T\_Sh^{j}_{U_{sc} \in S_c^i \wedge rout_x \wedge E_s^j} = T_s^j / N_x^{U_{sc}} \tag{2}$$

N_x^{Usc} : Number of blocked nodes U_{sc} on the same route x in the same name i of cluster S_c .

j: frequent time for sense. For example, the **first** time to sense, the **second** time to sense and so on.

From the above, we can calculate the specific time to transfer the data for each node:

$$T\_U_{sc}^{j}\Big|_{S_c^i \wedge rout_x \wedge E_s^j} = \text{Local time} + j + T_s^j + P_{H,L}^k * T\_Sh^{j}_{U_{sc} \in S_c^i \wedge rout_x \wedge E_s^j} \tag{3}$$

where P: the K^{th} node's task priority

And we can calculate the time spend by each U_{sc} for each event

$$T_{sp}^{total}\_U_{sc}^{j}\Big|_{S_c^i \wedge rout_x \wedge E_s^j} = t_o + t_s^j + T\_transfer^j_{U_{sc}\in S_c^i \wedge rout_x \wedge E_s^j} \quad (4)$$

The time spend by each U_{mu} for each event

$$T_{sp}^{total}\_U_{mu}^{j}\Big|_{S_c^i \wedge rout_x \wedge E_s^j} = \text{uploading time} + \sum_{j=1}^{q} T_{sp}^{total}\_U_{sc}^{j}\Big|_{S_c^i \wedge rout_x \wedge E_s^j} \quad (5)$$

Rule 4: sensing and turn off: after the U_{sc} sensing the data then the Bluetooth and another sensing tool will turn off to prevent the redundancy sensing to reduce the energy consumption by the device. And open the Wi-Fi depend on the time $T\_U_{sc}^{j}$ for each node.

2. Energy handling

Rule5: Total energy:

As mentioned before the researchers calculated the energy by the following equation:

$$En_{user}^{total} = \min \sum En(t_o, t_d) \quad (6)$$

In this paper we derived the total energy depending on our time equations (4)and (5), so the total energy consumption by U_{sc} and U_{mu} is for the same event derived in the following equation

$$En^{total}_{(U_{sc}\wedge U_{mu})\in S_c^i \wedge rout_x \wedge E_s^j} = \sum_{j=1}^{q} En(T_{sp}^{total}\_U_{sc}^{j}) + En(T_{sp}^{total}\_U_{mu}^{j}) \quad (7)$$

where the deadline time t_d in our framework doesn't exceed the value in equation (2)

Rule5: the user U_{mu} will upload when he does any call to send the data parallel with a call by 3G piggyback facility.

VI. Evaluation Concepts

In this section, we showed the mobile phone energy evaluations. There are many studies which calculate the mobile phone actions, Table I shows [2] the important actions to different phones which are used by this paper in the experiment test where we depend on the energy consumption by Bluetooth scanning and 3G call from this table. Table II [15] shows the power consumption to the download and the upload actions to different communications, where all actions in this table are depended in our experiments. Table III [21] shows the time required via different communication. In this table we depend on the time delay which caused by the Round Trip Time RTT. Why were we depended this delay? We depended this delay because we used the three-way handshaking protocol, so we need to use 2*RTT.

TABLE I: ENERGY CONSUMPTION FOR DIFFERENT PHONE USAGES

Action	Power(W)	Time (s)	Energy (J)
Idle	0.15	60	0.9J
3G Call	1.265	60	75.9J
SMS	0.0583	60	3.5J
Bluetooth Scanning	0.225	20	4.5J

TABLE II: THE POWER USAGE TO DOWNLOAD AND UPLOAD BY MOBILE PHONE

Action	Power(W)	Speed (KB/s)
Wi-Fi download	1.1	143.1
Wi-Fi upload	1.1	115.3
Wi-Fi download/Upload	1.5	430
3G download +3G call	1.4	48
3G upload +3G call	1.4	43

TABLE III: THE TIME REQUIRED TO THE DIFFERENT COMMUNICATION

Link type	Bandwidth (typical)	One-way distance (typical)	Round-trip delay RTT
Dial-up	56Kbps	10km	87 µs
Wireless LAN	54 Mbps	50m	0.33 µs

VII. Experimental Setup

Download energy: In this section, we showed the experimental work did by this paper depending on our proposed rules and the information in evaluation concept section. Fig. 4 shows a comparison to the energy consumption by download via Wi-Fi, 3G, and our proposal, where this figure shows clear outperform to our proposal.

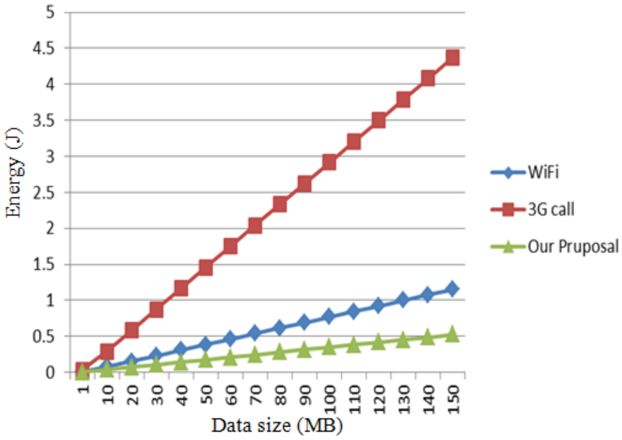

Fig. 4. Energy consuming via download

Upload energy: in this section, we tested our proposal to check the energy consuming by uploading the data, and we did the comparison via Wi-Fi , and 3G. Fig. 5 shows our proposal outperforms comparing with others.

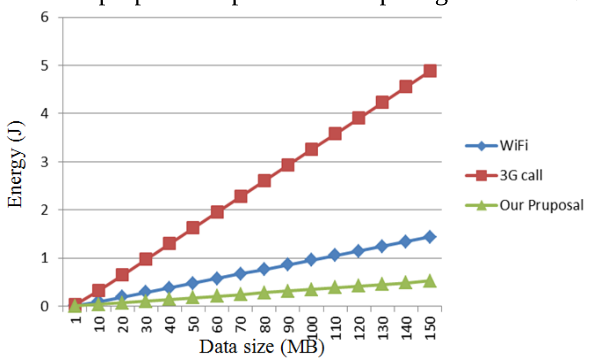

Fig. 5. Energy consuming via uploading

The energy consumption by U_{sc} and U_{mu}: in this part we showed the energy consuming by U_{sc} and U_{mu},

©2018 Journal of Communications

according to the evaluation concepts, and we re-represented equations (6) and (7) by practical evaluation with corresponding to the information in Table III. Here we summarized the practical representation by the following equations:

Energy consuming by the normal way (as previous researchers) for any user:

$$En_{user}^{total} = En_{scan} + En_{download} + En_{(3G\ call+upload)} \quad (8)$$

with our proposed, we need to divide the users to two parts U_{sc} and U_{mu}. The energy consuming by the user U_{sc} in slave cluster:

$$En_{U_{sc} \in S_c^i \wedge rout_x \wedge E_s^j}^{total} = En_{scan} + En_{download} + En_{(download+uplaod)*N} \quad (9)$$

where N, is the number of nodes in route x

And the energy consumed by user U_{mu} master cluster we used:

1. Upload by Wi-Fi:

$$En_{U_{mu} \in S_c^i \wedge rout_x \wedge E_s^j}^{total} = En_{(download+uplaod)*N} \quad (10)$$

2. Upload by 3G communication:

$$En_{U_{mu} \in S_c^i \wedge rout_x \wedge E_s^j}^{total} = En_{(3G\ call+uplaod)} \quad (11)$$

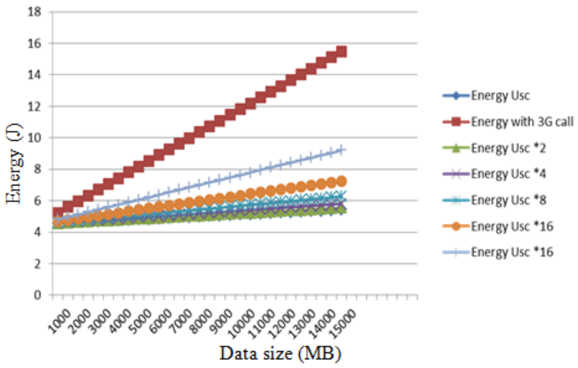

Fig. 6 Shows the energy consumption with increasing the data size to the users in slave cluster U_{sc} by different scenarios, where; Energy with 3G call: This the normal way did by previous researchers. Energy U_{sc} * number: refer to the route to transfer data with a number of nodes.

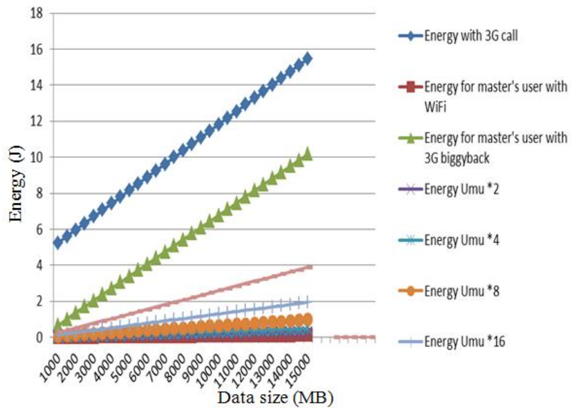

Fig. 7. Shows the energy consumption with increasing the data size for the user in slave cluster U_{mu} by different scenarios, where; Energy for master's user with 3G piggyback call: the update information by U_{mu} by 3G piggyback call with for free cost. Energy U_{mu} * number : refer to the number of nodes could share to upload the data. Energy for master's user with Wi-Fi: the uploading the data by Wi-Fi by U_{mu}

After applying the up equations, Fig. 6 shows the results to the energy consumption by U_{sc}, and it is clear outperform our proposed. Fig. 7 shows the energy consumption by U_{mu}.

VIII. CONCLUSION

This framework has many properties which make it very useful in MCS, where clustering the users to the master and slave, with depend the users in master area who are in home or work, would give a new facility in MCS to reduce the energy and the cost of data uploading. Besides that, the users how engage with MCS haven't effort about the energy and the cost, which lead him to be happy to support the MCS. This paper arrange the MCS by give a specific equation to "time of sense" and "time to transfer" and also "the time to upload" , depending on the priority of task, therefore; we get a real time framework with zero data upload cost and low energy consumption.

REFERENCES

[1] H. Xiong, D. Zhang, L. Wang, and H. Chaouchi, "EMC3: Energyefficient data transfer in mobile crowdsensing under full coverage constraint," *IEEE Trans. Mobile Comput.*, vol. 14, no. 7, pp. 1355–1368, Jul. 2015.

[2] 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.

[3] D. Zhang, L. Wang, H. Xiong, and B. Guo, "4w1h in mobile crowd sensing," *IEEE Commun. Mag.*, vol. 52, no. 8, pp. 42–48, Aug. 2014.

[4] G. Cardone, *et al*., "Fostering participaction in smart cities: A geosocial crowdsensing platform," *IEEE Commun. Mag.*, vol. 51, no. 6, pp. 112–119, Jun. 2013.

[5] N. D. Lane, *et al*., "Piggyback crowdsensing (PCS): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities," in *Proc. 11th ACM Conf. Embedded Netw. Sensor Syst.*, Rome, Italy, 2013, pp. 7:1–7:14.

[6] W. Sherchan, *et al*., "Using on-the-move mining for mobile crowdsensing," in *Proc. IEEE 13th Int. Conf. Mobile Data Manage.*, Bangalore, India, 2012, pp. 115–124.

[7] G. Cohn, *et al*., "An ultra-low-power human body motion sensor using static electric field sensing," in *Proc. ACM Conf. Ubiquit. Comput.*, Pittsburgh, PA, USA, 2012, pp. 99–102.

[8] M. R. Ra, B. Priyantha, A. Kansal, and J. Liu, "Improving energy efficiency of personal sensing applications with heterogeneous multiprocessors," in *Proc. ACM Conf. Ubiquit. Comput.*, Pittsburgh, PA, USA, 2012, pp. 1–10.

[9] C. Gomez, J. Oller, and J. Paradells, "Overview and evaluation of Bluetooth low energy: An emerging low-power wireless technology," *Sensors*, vol. 12, no. 9, pp. 11734–11753, 2012.

©2018 Journal of Communications

[10] M. R. Ra, B. Liu, T. F. La Porta, and R. Govindan, "Medusa: A programming framework for crowd-sensing applications," in *Proc. 10th Int. Conf. Mobile Syst. Appl. Serv.*, Low Wood Bay, U.K., 2012, pp. 337–350.

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

[12] G. Perrucci, F. Fitzek, and J. Widmer, "Survey on energy consumption entities on the smartphone platform," in *Proc. IEEE 73rd Veh. Technol. Conf.*, Yokohama, Japan, 2011, pp. 1–6.

[13] R. K. Rana, C. T. Chou, S. S. Kanhere, N. Bulusu, and W. Hu, "Earphone: An end-to-end participatory urban noise mapping system," in *Proc. 9th ACM/IEEE Int. Conf. Inf. Process. Sensor Netw.*, Stockholm, Sweden, 2010, pp. 105–116.

[14] 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. 8th Int. Conf. Pervas. Comput.*, Helsinki, Finland, 2010, pp. 355–372.

[15] J. Nurminen, "Parallel connections and their effect on the battery consumption of a mobile phone," in *Proc. 7th IEEE Consum. Commun. Netw. Conf.*, Las Vegas, NV, USA, 2010, pp. 1–5.

[16] E. Soroush, K. Wu, and J. Pei, "Fast and quality-guaranteed data streaming in resource-constrained sensor networks," in *Proc. ACM Symp. Mobile Ad Hoc Netw. Comput.*, Hong Kong, 2008, pp. 391–400.

[17] A. Kate, G. M. Zaverucha, and U. Hengartner, "Anonymity and security in delay tolerant networks," in *Proc. 3rd Int. Conf. Secur. Privacy Commun. Netw. Workshops*, Nice, France, 2007, pp. 504–513.

[18] B. Hull, *et al*., "CarTel: A distributed mobile sensor computing system," in *Proc. 4th Int. Conf. Embedded Netw. Sensor Syst.*, Boulder, CO, USA, 2006, pp. 125–138.

[19] T. Fawcett, "An introduction to ROC analysis," *Pattern Recognit. Lett.*, vol. 27, no. 8, pp. 861–874, 2006.

[20] P. Karn, "MACA- A new Channel Access Method for Packet Radio," in *Proc. ARRL/CRRL Amateur Radio 9th Computer Networking Conference*, 1990, pp. 134-140.

[21] L. L. Peterson and B. S. Davie, *Computer Networks a System Approach*, Fifth edition, Morgan Kaufmann Publishers is an Imprint of Elsevier, 2012

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 science and M.S. degree in computer network 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.