Journal of Engineering and Applied Sciences 14 (10): 3200-3209, 2019
ISSN: 1816-949X

# Parallel Three-Way Handshaking Route in Mobile Crowd Sensing (PT-MCS)

Furkan Rabee, Ali S.A. Al-Haboobi and Mohammed Ridha Nsaif
College of Computer Science and Mathematics, University of Kufa, Najaf, Iraq

**Abstract:** The energy consumption by mobile phone is a big challenge to enroll the volunteers in mobile crowd sensing tasks, besides that, there is another problem, the cost of uploading data to the data center. This study presents a protocol to manage the routing in mobile crowd sensing to reduce the power consumption and data uploading cost. The proposed research starts by classifying the users into two modes, collector users and up-loader users where collector users collect data without uploading it to the data center whereas the up-loader users have to upload the data to the data center after receiving the data from the collector users. The collector users are responsible for two tasks, first, collect the data from the environment in different connections such as WiFi, Bluetooth. Second, collect the data from the other collector users to deliver it to another user until reach to any up-loader user. This study proposes a protocol to manage the routing path from user to a user called Parallel Three-way handshaking in Mobile Crowd Sensing PT-MCS. This protocol has more benefits are reducing the power consumption and data cost for uploading where it provides an equation for delay time and power consumption. The proposed research showed it is out perform compared with other researches.

**Key words:** Wireless sensor network, mobile crowd sensing, three-way handshaking, power consumption, protocol, WiFi

## INTRODUCTION

Over the recent years, sensor-armed smartphones (Wang *et al*., 2015) have become an important instrument in Mobile Crowd Sensing (MSC) (Zhang *et al*., 2014; Ganti *et al*., 2011). These sensors are capable of performing different MSC tasks such as social sensing (Sherchan *et al*., 2012) and environmental monitoring (Rana *et al*., 2010). In order to attract more users for participating in MSC applications, it is important forreducing energy consumption and mobile data cost for user's devices. The energy consumption is associated with mobile device's battery while mobile data cost is related to fees incurred. For attracting more users in engagements effectively in MSC tasks, it should prevent battery drain excessively and decreasing data cost incurred.

Several papers have developed many energy-saving methods for encouraging participation in MSC applications. However, energy consumption and mobile data costin MSC are still needed for more minimizing. Some studies proposed solutions by including implementing dynamic sensing task cycle (Rachuri *et al*., 2011) doing a trade-off between remote and local computation (Sherchan *et al*., 2012). In addition, the task is intelligently dividing among users (Sheng *et al*., 2012). Most these solutions researches by sending the sensed data to acentral server as soon as the data is generated (Wang *et al*., 2015). However, other MSC tasks do not immediately need sending the sensed data to a central server but it provides a tolerable amount of delay to upload it (Wang *et al*., 2015).

In this study, we presented a new method to send data with a tolerant delay of time to reduce energy consumption and mobile data cost. In this framework, the user's devices are divided into two modes: collect mode and upload mode. In the collect mode, users are responsible for sensing the data and sending it to an upload user that is responsible for uploading the sensed data to the data center with zero cost via. WiFi communication.

### Literature review

**MSC framework:** In recent years, there has been an increasing number of studies on mobile crowd sensing that has many applications. For example, urban noise monitoring (Rana *et al*., 2010) and an application to sense and compute devices collectively share data to measure and map phenomena of common interest (Ganti *et al*., 2011). By Zhang *et al*. (2014) proposed a MSC life cycle in four-phases to characterize the MSC process. These applications are enabled by the participatory management

**Corresponding Author:** Furkan Rabee, College of Computer Science and Mathematics, University of Kufa, Najaf, Iraq to crowd sensing frameworks (Cardone *et al*., 2013). There are many strategies for data uploading (Musolesi *et al*., 2010) have been studied.

**Energy conservation in MSC:** In order to reduce the energy consumption in a MSC task an intelligent algorithm has been designed for task assignment by minimizing the number of participants to cover a specific area. Several approaches have been presented depending on various task models by Xiong *et al*. (2015) and Cardone *et al*. (2013). For each user's device in MSC, three stages have been studied for energy-saving mechanisms which are sensing, computing and data uploading. Sensing and computing stages are responsible for the process of obtaining and processing data from sensors. For minimizing the energy consumption of sensing, many methods have adopted low power sensors (Cohn *et al*., 2012). Low power processors have been used in mobile sensing systems for reducing the energy consumption of computing (Ra *et al*., 2012a, b). The aforementioned methods focused on the first two stages while the proposed method focused on the three phases 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.

For reducing the energy consumption of data uploading by Lane *et al*. (2013) used low power communication methods. For example, Bluetooth, Wi-Fi or parallel data uploading with voice calls (Nurminen, 2010) where these communications are used as energy-efficient communications instead of immediately using 3G. Moreover, participant's devices used for carrying and forwarding data between devices in order to upload data by specified participants who have a better chance for energy-efficient approaches (Hull *et al*., 2006). The previous research of data uploading process is used in our protocol with an important modification.

## MATERIALS AND METHODS

Compressing data before uploading (Soroush *et al*., 2008) or uploading part of the data while assuming the rest (Musolesi *et al*., 2010). These methods consume additional energy in computation and therefore, the choice to adopt these approaches requires being researched carefully to find the overall performance.

**Data cost conservation in data uploading:** To reduce data cost in uploading, the previous study focused essentially on minimizing data size with extra computation on the local mobile devices to aggregate sensed data before uploading (Sherchan *et al*., 2012). In addition, the energy-saving method such as Soroush *et al*. (2008) aimed for minimizing the size of the data to upload. Whereas these existing methods can significantly decrease data cost when the data size is large which cannot entirely remove it. In this study, the process to upload the data is doing by specific users who will be using WiFi and Bluetooth and therefore, the cost of data uploading will be zero.

**Problem statements:** Recently, researchers have shown an increasing interest in reducing the energy consumption and the cost of data uploading for mobile crowd sensing. Many crowd sensing applications have been designed by many researchers using low energy consumption and low-cost uploading. However, these two critical concerns are still needed further reducing to encourage more participants in crowd sensing tasks. Much of the existing studies that designed MCS applications such as environmental monitoring (Xiong *et al*., 2015) and reality mining (Sherchan *et al*., 2012) which sense data and upload it to the server after some delay (a maximal tolerable amount of delay $d_{max}$) in order to decrease data cost incurred and energy consumption. The sensed data produced on a user's device at $t_0$ and then uploaded to the server during ($t_d$, $t_d$) where $t_d = t_0 + d_{max}$. In a recent study by Wang *et al*. (2015) divided users to Non-Data-Plan (NDP) users and Data-Plan (DP) users. This is designing for delay-tolerant uploading plans for decreasing the cost of data uploading for NDP users and minimizing the energy consumption for DP users. From previous studies, a new framework has been proposed which is divided the environment into two users: collector users and uploader users where collector users are sensed the data and orient it to other users until reach any uploader user. While the uploader user is responsible for uploading the data to the data center with zero cost and low energy consumption. The next study will illustrate the proposed framework in detail.

**PT-MCS:** To explain the SCR-MSC framework, this study will illustrate the main parts as follows.

**Mobile sensing area description:** The detail description of PT-MSC will be explained for the mobile sensing area and all the connections between participants and the environment. In the crowd sensing process, the following definitions have been assumed as follows.

**Definition 1:** All smart devices will be named nodes and the PT-MSC supports various types of devices. They can be a tablet computer, mobile phone, a smart camera with WiFi, Bluetooth or infrared. These devices can send various messages, for example, texts, images, voices.

**Definition 2:** PT-MSC starts with classifying the users into two groups of users: uploader user and collector user. The users communicate between them as star topology where the uploader user should be in the center of the star.

**Upload area:** $A_{up}$ this user becomes uploader user when he is at home or work, thus, the user can upload with zero cost via. an internet connection and can charge the battery easily. The number of users should usually be less than the collector user.

**Collect area:** $A_{col}$ the other important part of the sensing area, the collect area and the user will be outside the home or work. Using the star topology between the uploader user and the collector user, NP-problem avoided. This design supposed to reduce the data cost uploading and power consumption as we discussed in detail later in this study.

**D3:** All users controlled by an application to make the users working by two modes.

**Upload mode:** The application will turn the users into this mode just they enter to the $A_{up}$ and they will be just data uploader Users ($U_{up}$).

**Collect mode:** The users in this mode should be data sensing ($U_{col}$).

**D4:** All user's equipment should be identifying by GPS to know which area the user will be thus to turn to which mode should be.

**The life cycle of PT-MSC:** There are different stages to complete the life cycle of MSC. By Zhang *et al*. (2014), it was designed three stages: creating, assignment and execution while another study designed four stages (creating, assigning, executing and data integration) tasks (Ra *et al*., 2012a, b). By using these two studies, we designed a new life cycle of MSC as follows.

**Task creation:** Mobile sensing applications create a MSC task on each user's node by the organizer of MSC. MEDUSA framework developed such that application (Ra *et al*., 2012a, b).

**Task assignment:** It is the second stage after task creating. In this stage, participant's nodes are employing and assigning with separate sensing tasks on each node (Zhang *et al*., 2014). In this framework, the participant can be a collector User $U_{col}$ or an uploader User $U_{up}$ based on their area.

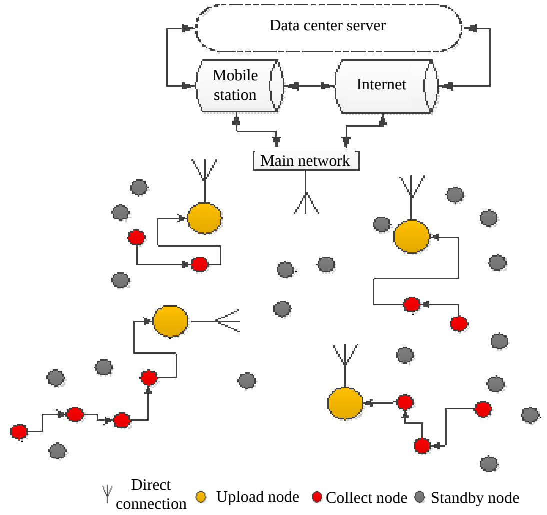

Fig. 1: Sensing area

**Task execution:** When assigned sensing task is received, the participant is trying to complete it during a predefined period in parallel with other tasks (Zhang *et al*., 2014). There are two types of task execution.

**Task preparing (sensing, orienting and sending):** The task of collector user $U_{col}$ is completing in three subtasks.

**Sensing:** This sub-task is achieving in the $U_{col}$ node by sensing the environment through different sensors such as camera and infrared. This user's sensors are connected with the outside world to sense different changes to the environment such as pollution and noise. The process of sensing can be done by Bluetooth communication.

**Orienting:** When the sensed data is generated, the application on $U_{col}$ node should build a full route from the $U_{col}$ to the $U_{up}$ in order to upload the sensed data to the data center by the $U_{up}$ through the WiFi network only. The route can be built by using three protocols that will be explained later in the next study.

**Sending:** When the route is completed, the sensed data should be sent in order to delivered to $U_{up}$ and consequently uploading it under this user.

**Task computing (collecting, uploading, integration):** After receiving the $U_{up}$ node the sensed data, this user should upload it to the server by two methods. The first is by using the WiFi communication and the second is by piggy-back with 3G mobile communication. These methods are zero-cost data uploading. Figure 1 shows an example to illustrate the routing data with PT-MCS from sensing data to the uploading.

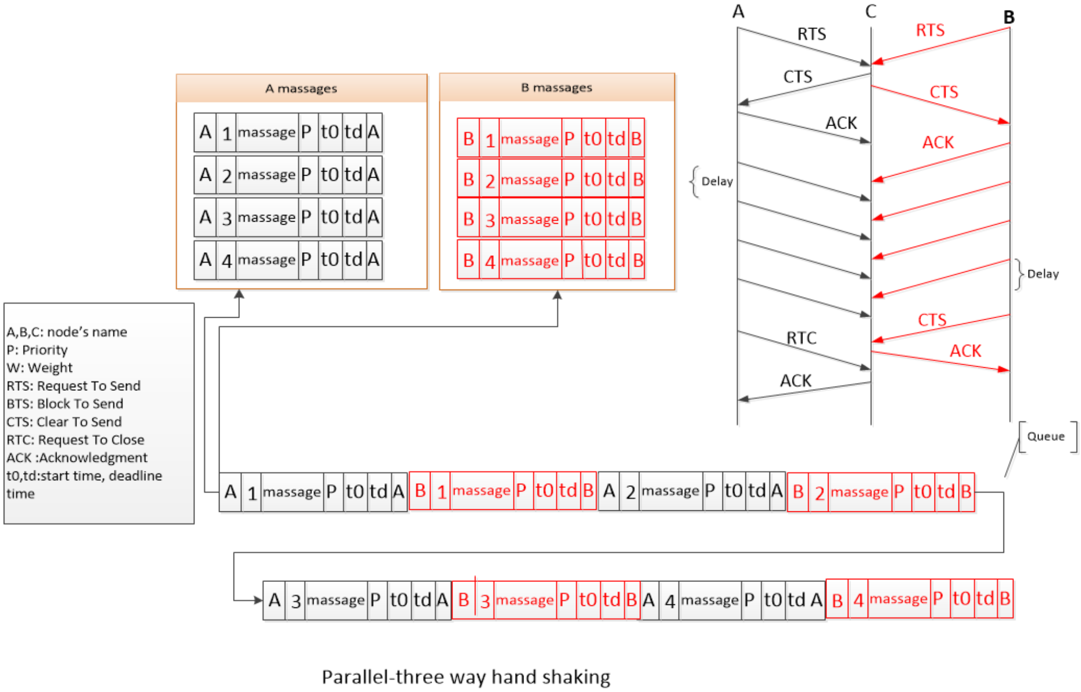

Fig. 2: Capsulation the data via. parallel-three way handshaking

**Parallel-three way handshaking route:** In this study, we present our proposed approach to build a route path to transfer data in mobile crowd sensing. This parallel-three way handshaking route contained two parts.

**The communication between the nodes:** The communication between the nodes presented to recover the hidden terminal problem. The hidden terminal problem supposed that, if node (A) wants to send to node (C) but there is another node (B) wants to send to node (C) where (A) and (B) cannot hear each other. They are both going to sense the carrier to be idle and start sending which leads to a collision, this problem is solved by MACA protocol (Karn, 1990). MACA used to transfer information from node to node. In this study, we present new protocol called parallel three-way handshaking to build transfer route in mobile crowd sensing. This protocol used another solution to the hidden terminal problem to build the route. Suppose that there are three nodes A-C where A needs to send to C and B needs to send to C and both A and B cannot lesson to each other. MACA protocol (Karn, 1990) supposed that A will send a request to send RTS to C, if C is free, it will send a clear to send CTS to A, then, start to transmit the data and block the others. The parallel three-way handshaking supposed that node C can accept the connection from two nodes by building a route can synchronize between A and B called capsulation. Figure 2 shows the capsulation of the data by parallel-three way handshaking protocol. Where C node accepts the RTS from both A and B and starts receiving the data from A or B as messages where the data already partitioned as messages at the source A or B where, C receives the message one, for example, from A in a time and receiving the message one from B in a delay time between receiving message one and message two from A. After C finished receiving the messages, it will capsulate the data with adding an address, priority and transfer time, then, it provides a header and a trailer for each message. Such that formatting is doing to the data to be easy for the $U_{up}$ to de-capsulate the data. A and B are $U_{col}$ and C could be $U_{col}$ to pass the data to another node or could be $U_{up}$ to uploading the data to data center.

The route will build when each $U_{col}$ sends a request to each other to transfer data for uploading to the data center, each $U_{up}$ will receive the request from $U_{col}$, thus, there are several routes will construct to transfer data, after the route will construct, the parallel three-way handshaking will apply between the nodes (the parallel three-way handshaking will be included in the route to transfer the data between the nodes), Fig. 3 shows the transfer route between the nodes.

The nodes that will send the data such as A or B, it will partition its data to messages as we mentioned before, then these nodes will send another information with each message as shown in Fig. 2, the format as follows: P: this is a priority factor where there is a priority for each

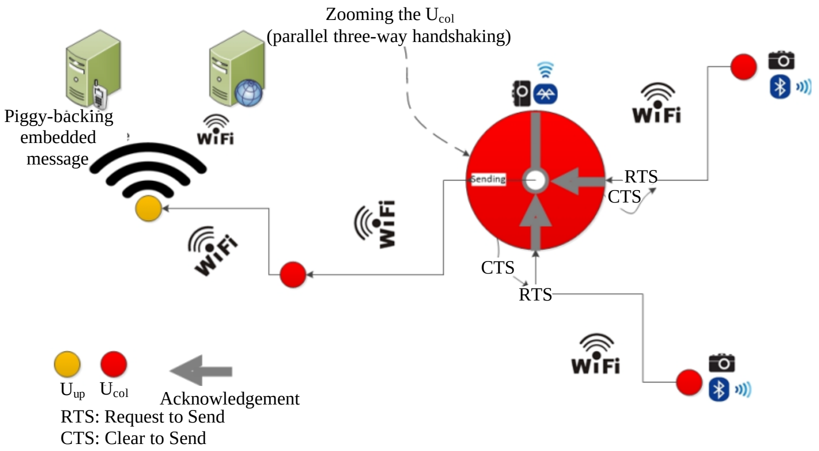

Fig. 3: PT-MCS routing data

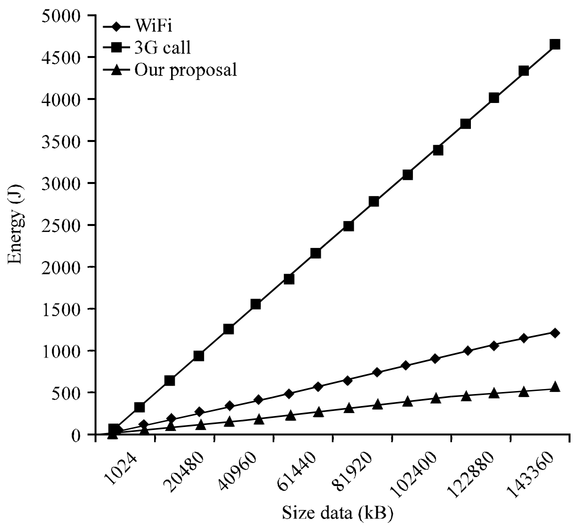

Fig. 4: Energy consuming via. download

message, if it is considered, there are two values: H: High and L: Low. $T_0$, $T_d$: these values show the starting time that a message has been sent and the deadline for the arrival time, these factors are very important in real time situations.

**The message numbers and sequence:** This factor shows the number on the messages of data to avoid missing information and may use for ordering the sequence of data in case there is no priority for messages. It could show by 1 or 2, … , etc. or 1-5, 2-5, … , etc.

**Header or trailer:** This factor refers to the start and end for each message, it could show by any symbol to refer the name of the original data.

**Building the route:** After partitioning the data by A or B, then, the message will send to C after receiving the CLS. The C node would capsulate all the messages and sending them to the next node to reach the destination node $U_{up}$ through the transition route as shown in Fig. 3. There are many algorithms to build the route, Fig. 4 shows these algorithms.

**Algorithm 1; Main algorithm (routing algorithm):**

```
Began
Location detection
If the node is collector then
    { call $U_{col}$
else call $U_{up}$}
End
```

**Algorithm 2; Collector node ($U_{col}$) (routing algorithm):**

```
Began
{
Input the sense time
Input the orientation time
Turn on the Bluetooth
Sensing information for 20 sec
If t<20 then
{
Collect data
} end if
turn off the Bluetooth
turn on the WiFi
building the route
send $RTC_{remote}$
xx: If $CTS_{remote}$ = true then //the local send clear to send to remote
{
  If CTS $_{local}$=true // the local ready to accept the data
    a. {
    b. Partition the LOC data into messages
    c. Adding the properties for each message
    d. Capsulate the LOC data with REM data
    e. Sending data
    f. }
Else
    a. {
    b. Partition the LOC data into messages
    c. Adding the properties for each message
    d. Sending data as messages
    e. }
  Else
  {waiting …
Go to: xx
}
turn off WiFi
}
End
```

**Algorithm 3; Up-loader node algorithm (routing algorithm):**

```
Began
{
Detect the connection
YY: If RTS remote = true then
{
CTSlocol = ture
Receive the data (download)
Decapsulate the data
Else
{waiting go to YY: }
Uploading with WiFi or via. voice call (Parallel uploading)
}
End
```

## RESULTS AND DISCUSSION

**Uploading data design:** This study shows the rules to upload the data to the data center. Depending on our topology which is there are selective nodes to upload the data in crowd sensing and the location of them at home or in work where they use WiFi for uploading or upload the data parallel with voice call with zero cost. The $U_{up}$ is responsible for uploading the data and he will be in the center, thus, the topology is a star topology, the other user such $U_{col}$ should be collectors to the data from the environment as sensor or orient the information to the $U_{up}$. The uploading process is doing by the following rules: uploading the data via. WiFi. This is zero cost data uploading. Uploading the data parallel to voice call through 3G communication. This is another way to upload the data with zero cost of data uploading.

The data does not have a high priority or data is large such that cases, the data could be uploaded after midnight because most mobile companies offer a discount and free minutes after midnight, thus, uploading the data with this service.

**Energy consumption rules related to data cost uploading:** The performance for MCS isaffected by two problems, energy consumption and cost of data uploading. There is a relationship has been found between "Time to Transfer" and "Energy Consumption". To describe this relationship, we divided the work into two parts as follows.

**Consuming time by transfer:** To conduct a specific time for transferring the data, we depending the following rules to control on time-consuming.

**R1:** To detect information from the environment, the time to sense the information has been depended, this time repeated frequently by mobile crowd sensing organizer, we called it time to Sense ($S_t$) this time may detect an event ($S_e$).

**R2:** As mentioned in PT-MCS life cycle under task preparation, the node should do an orientation route before sending; in this protocol, the users $U_{col}$ should check the possibility to make the orientation route before data Sensing (before $S_t$), the reason to do this checking that to know the Active route $A_{route}$ (the route from $U_{col}$ to $U_{up}$), therefore, the user $U_{col}$ included the $A_{route}$ 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 $S_t$ by time $t_0$ and it is called pre-sensing Time ($T_{pre\text{-}s}$) Eq. 1:

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

where, just $t_0$ is finished $S_t$ Start with time = 0. For example, we suppose the initial sensing begin at 06:00:00 o'clock, $t_0$ = 20 sec. If $S_t$ Starts every 1 h, $T_{pre\text{-}s}$= 05:59:40, the next $S_t$ will begin at 07:00:00 o'clock and so on.

**R3:** When the pre-sensing began, all $U_{col}$ send RTS to the neighbored users to build the route reaching to $U_{up}$ and waiting for the reply. If the reply is CTS, it means the route is open, else the reply will be BTS, thus, the node will block. According to parallel-three way handshaking, most nodes will receive CTS and the time period to capsulate the node's data should be between the current $S_t$ to the next $S_t$ and arrange the node's data depending on the priority for each node's task. When any node delivers its data, it will turn to inactive mode to reduce the power consumption. To calculate the consumed time for each node through capsulation and $A_{route}$, we need to describe some parameters to reach to the final equation, called time for capsulation T_C:

**Round trip delay RTT:** This time for open and close the connection between the nodes, so, its need 2RTT for each node. The time to transfer the data to each node, $T_{ransfer.}$ Eq. 2:

$$T\_C^{j}_{U_{col}\in rout_x \wedge E_s^j} = \sum_{n=1}^{2} T^{n}_{transfer} + 2RTT \tag{2}$$

**J:** Frequent time for sense, for example, j = 1 is the first time to sense, j = 4 is the 4th time to sense. n = the node number which transfers data through $U_{col}$ in the route x. Now, we could write an equation to show the total spent time for an event by $U_{colas}$ follows Eq. 3:

$$T^{total}_{sp}\_U^{j}_{col}\Big|_{rout_x \wedge E_s^j} = t_o + T\_C^{j}_{U_{col}\in rout_x \wedge E_s^j} \tag{3}$$

The spenttime by uploading, the following equation shows the total time spend by $U_{up}$ to upload the data to the data center Eq. 4:

$$T_{sp}^{total}\_U_{up}^{j}\Big|_{rout_x \wedge E_s^j} = \text{Uploading time}+\sum_{k=1}^{q} T_{sp}^{total}\_U_{col}^{j}\Big|_{rout_x \wedge E_s^j} \quad (4)$$

The uploading time in this equation has shown in specific because it depends on the type of the device and network.

**R4:** When the $U_{up}$ finished sensing the data, the Bluetooth and other sensing equipment will turn off to prevent the redundancy sensing and to reduce the energy consumption by the device. While activating the WiFi depends on the spending time for each node.

**Energy handling**
**R5:** The total energy has been calculated by a researcher Wang *et al*. (2015) as follows (Eq. 5):

$$En_{user}^{total} = \min En\sum(t_o, t_d) \quad (5)$$

where, the deadline time ($t_d$) in this protocol should not exceed the value in Eq. 2. The total energy in this study is derived from the relationship between the time Eq. 3 and 4, thus, the total energy consumption by $U_{col}$ and $U_{up}$ showed by the following (Eq. 6):

$$En_{(U_{col} \wedge U_{up}) \equiv rout_x \wedge E_s^j}^{total} = \sum_{j=1}^{m} En(T_{sp}^{total}\_U_{col}^{j})+En(T_{sp}^{total}\_U_{up}^{j}) \quad (6)$$

Where:
J = Frequently time to sense
m = The maximum number of sense

**R6:** $U_{up}$ would upload the data when he makes any call to send the data by paralleling way to the call via. 3G piggyback facility.

**Evaluation concepts:** This part described the energy evaluation for mobile phone. Many studies which calculated the actions for the mobile phone, Table 1 shows (Wang *et al*., 2015) the main actions to different types phones which used in this study in the experiment test. We depended on the energy consumption caused by Bluetooth scanning and 3G call from this table. Table 2 (Nurminen, 2010) shows the action for mobile phone (download and upload) and its power consumption with different communications. The actions in this table were depended on our experiments. Table 3 Larry and Bruce (2012) shows the required time for different communications. In this table were depended on the delay time that produced by the Round Trip Time (RTT).

Table 1: Energy consumption for different phone usages

| Actions | Power (W) | Time (sec) | Energy (J) |
|---|---|---|---|
| Idle | 0.1500 | 60 | 0.9 |
| 3G call | 1.2650 | 60 | 75.9 |
| SMS | 0.0583 | 60 | 3.5 |
| Bluetooth scanning | 0.2250 | 20 | 4.5 |

Table 2: The power usage to download and upload by mobile phone

| Actions | Power (W) | Speed (kB/sec) |
|---|---|---|
| WiFi download | 1.1 | 143.1 |
| WiFi upload | 1.1 | 115.3 |
| WiFi download/Upload | 1.5 | 430.0 |
| 3G download+3G call | 1.4 | 48.0 |
| 3G upload+3G call | 1.4 | 43.0 |

Table 3: The time required for the different communication

| Link types | Bandwidth (typical) | One-way distance (typical) | Round-Trip Delay time DRT (μsec) |
|---|---|---|---|
| Dial-up | 56 kBps | 10 km | 87 |
| Wireless LAN | 54 MBps | 50 m | 0.33 |

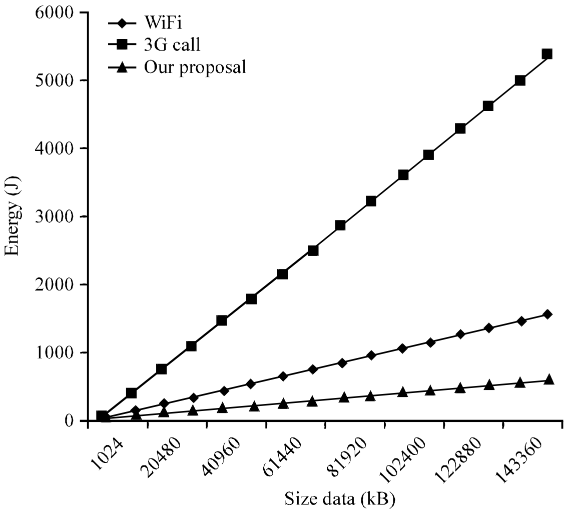

Fig. 5: Energy consuming via. uploading

**Experimental setup:** In this study, we presented the experimental results which based on our suggested rules and the information in energy evaluations study.

**The energy for downloading:** Figure 4 shows the compared result of the consumed energy by download using WiFi, 3G and our proposal where our proposal out performs the other proposals in the energy consumption.

**The energy for uploading:** Figure 5 shows the status of data upload and comparison with other proposals (WiFi and 3G), It shows the advantage of our proposal on the rest of the proposals.

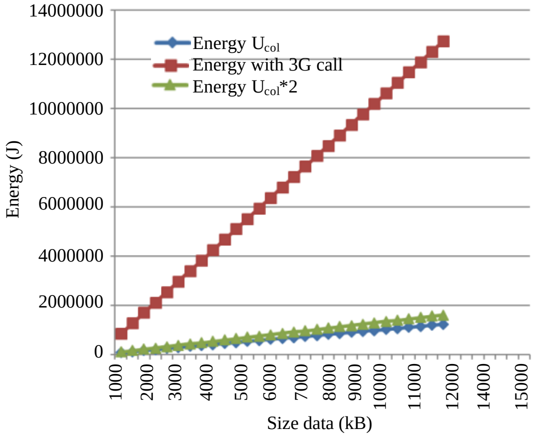

Fig. 6: The energy consumption with increasing the data size to the users $U_{col}$ by different scenarios where energy with the 3G call: this the normal way did by previous researchers. Energy $U_{col}$* number: refer to the route to transfer data with a number of nodes

The energy consumption by $U_{col}$ and $U_{up}$. Figure 6 shows the consumed energy by $U_{col}$ and $U_{up}$ according to the energy evaluations and we reformulated the Eq. 5 and 6 by experimental evaluation with consistent to the information in Table 3. Here, we summarized the practical representation according to the following Eq. 8. As previous researchers calculated the consumed energy using the normal method for any user:

$$En_{user}^{total} = En_{scan} + En_{download} + En_{(3G\ call+upload)} \tag{7}$$

With our proposed, we need to divide the users into two parts $U_{col}$ and $U_{up}$. The energy consuming by the user $U_{col}$:

$$En^{total}_{U_{cole} \in rout_x \wedge E_s^j} = En_{scan} + En_{download} + En_{(download+uplaod)*N} \tag{8}$$

where, N is the Number of nodes in route x. And the energy consumed by user $U_{up}$we used:

Upload by WiFi:

$$En^{total}_{U_{up} \in rout_x \wedge E_s^j} = En_{(download+uplaod)*N} \tag{9}$$

Upload by 3G communication:

$$En^{total}_{U_{up} \in rout_x \wedge E_s^j} = En_{(3G\ call+uplaod)} \tag{10}$$

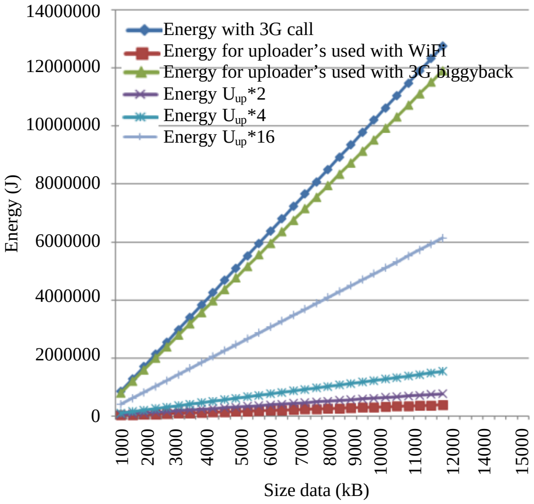

Fig. 7: The energy consumption with increasing the data size for the user $U_{up}$ by different scenarios where energy for master's user with 3G piggyback call: the update information by $U_{up}$ by 3G piggyback calls with for free cost. Energy $U_{up}$* number: refer to the number of nodes could share to upload the data. Energy for master's user with WiFi: the uploading the data by WiFi by $U_{up}$

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

## CONCLUSION

The purpose of the current study was to minimize the energy consumption and the cost of data uploading in mobile crowdsensing. We presented a parallel three-way handshaking protocol to manage the routing path from user to user. This protocol divides the users into two types: collector users and uploader users. The uploader users are responsible for the process of uploading the data after receiving it from the collector users. While the collector users are responsible for the process of sensing the data from the surrounding environment as well as collecting the data from other collector users to deliver it to any uploader user. Moreover, this protocol handles the hidden terminal problem and let the nodes to accept all the request from other nodes to deliver the data at the end to the uploader. In addition, the protocol makes the users who enrolled with MCS need not concern to the battery charge and the inconvenience incurred because this protocol reduces energy consumption and cost of data uploading. The experimental results confirmed our proposal outperforming compared with other proposals in the energy consumption and the cost of data uploading in mobile crowdsensing tasks.

## REFERENCES

Cardone, G., L. Foschini, P. Bellavista, A. Corradi and C. Borcea *et al.*, 2013. Fostering participaction in smart cities: A geo-social crowdsensing platform. IEEE. Commun. Mag., 51: 112-119.

Cohn, G., S. Gupta, T.J. Lee, D. Morris and J.R. Smith *et al.*, 2012. An ultra-low-power human body motion sensor using static electric field sensing. Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp' 12), September 05-08, 2012, ACM, Pittsburgh, Pennsylvania, USA., ISBN:978-1-4503-1224-0, pp: 99-102.

Ganti, R.K., F. Ye and H. Lei, 2011. Mobile crowdsensing: Current state and future challenges. IEEE. Commun. Mag., 49: 32-39.

Hull, B., V. Bychkovsky, Y. Zhang, K. Chen and M. Goraczko *et al.*, 2006. CarTel: A distributed mobile sensor computing system. Proceedings of the 4th International Conference on Embedded Networked Sensor Systems, (ICENSS'06), ACM SenSys, pp: 125-138.

Karn, P., 1990. MACA-a new channel access method for packet radio. Proceedings of the 9th International Conference on ARRL/CRRL Amateur Radio Computer Networking Conference (Vol. 140), September 22, 1990, London, UK., pp: 134-140.

Lane, N.D., Y. Chon, L. Zhou, Y. Zhang and F. Li *et al.*, 2013. November Piggyback Crowd Sensing (PCS): Energy efficient crowdsourcing of mobile sensor data by exploiting smartphone app opportunities. Proceedings of the 11th ACM Conference on Embedded Networked Sensor Systems (SenSys'13), November 11-15, 2013, ACM, New York, USA., ISBN:978-1-4503-2027-6, pp: 1-7.

Larry L.P. and S.D. Bruce, 2012. Computer Networks a System Approach. 5th Edn., Morgan Kaufmann Publishers, Burlington, Massachusetts, USA., ISBN:9780123850591, Pages: 888.

Musolesi, M., M. Piraccini, K. Fodor, A. Corradi and A.T. Campbell, 2010. Supporting energy-efficient uploading strategies for continuous sensing applications on mobile phones. Proceedings of the International Conference on Pervasive Computing (Pervasive 2010), May 17-20, 2010, Springer, Berlin, Heidelberg, ISBN:978-3-642-12653-6, pp: 355-372.

Nurminen, J.K., 2010. Parallel connections and their effect on the battery consumption of a mobile phone. Proceedings of the 7th IEEE International Conferences on Consumer Communications and Networking Conference, January 9-12, 2010, IEEE, Las Vegas, NV, USA., ISBN:978-1-4244-5175-3, pp: 1-5.

Ra, M.R., B. Liu, T.F. La Porta and R. Govindan, 2012b. Medusa: A programming framework for crowd-sensing applications. Proceedings of the 10th International Conference on Mobile Systems, Applications and Services (MobiSys'12), June 25-29, 2012, ACM, Low Wood Bay, Lake District, UK., ISBN:978-1-4503-1301-8, pp: 337-350.

Ra, M.R., B. Priyantha, A. Kansal and J. Liu, 2012a. Improving energy efficiency of personal sensing applications with heterogeneous multi-processors. Proceedings of the 2012 ACM Conference on Ubiquitous Computing (UbiComp'12), September 05-08, 2012, ACM, Pittsburgh, PA, USA., ISBN:978-1-4503-1224-0, pp: 1-10.

Rachuri, K.K., C. Mascolo, M. Musolesi and P.J. Rentfrow, 2011. Sociablesense: Exploring the trade-offs of adaptive sampling and computation offloading for social sensing. Proceedings of the 17th Annual International Conference on Mobile Computing and Networking (MobiCom'11), September 19-23, 2011, ACM, Las Vegas, Nevada, USA., ISBN:978-1-4503-0492-4, pp: 73-84.

Rana, R.K., C.T. Chou, S.S. Kanhere, N. Bulusu and W. Hu, 2010. Earphone: An end-to-end participatory urban noise mapping system. Proceedings of the 9th ACM/IEEE International Conference on Information Process Sensor Network (IPSN'10), April 12-16, 2010, ACM, Stockholm, Sweden, ISBN:978-1-60558-955-8/10/04, pp: 105-116.

Sheng, X., J. Tang and W. Zhang, 2012. Energy-efficient collaborative sensing with mobile phones. Proceedings of the IEEE International Conference on INFOCOM, March 25-30, 2012, IEEE, Orlando, FL, USA., ISBN:978-1-4673-0773-4, pp: 1916-1924.

Sherchan, W., P.P. Jayaraman, S. Krishnaswamy, A. Zaslavsky and S. Loke *et al*., 2012. Using on-the-move mining for mobile crowdsensing. Proceedings of the 2012 IEEE 13th International Conference on Mobile Data Management, July 23-26, 2012, IEEE, Bengaluru, Karnataka, India, ISBN:978-1-4673-1796-2, pp: 115-124.

Soroush, E., K. Wu and J. Pei, 2008. Fast and quality-guaranteed data streaming in resource-constrained sensor networks. Proceedings of the 9th ACM International Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc'08), May 26-30, 2008, ACM, Hong Kong, Hong Kong, China, ISBN:978-1-60558-073-9, pp: 391-400.

Wang, L., D. Zhang, Z. Yan, H. Xiong and B. Xie, 2015. effSense: A novel mobile crowd-sensing framework for energy-efficient and cost-effective data uploading. IEEE. Trans. Syst. Man Cybern. Syst., 45: 1549-1563.

Xiong, H., D. Zhang, L. Wang and H. Chaouchi, 2015. EMC 3: Energy-efficient data transfer in mobile crowdsensing under full coverage constraint. IEEE. Trans, Mob. Comput., 14: 1355-1368.

Zhang, D., L. Wang, H. Xiong and B. Guo, 2014. 4W1H in mobile crowd sensing. IEEE. Commun. Mag., 52: 42-48.