

# A Mobile Cloudlet Assisted Architecture for Handling Multimedia Applications

Rajesh Kumar Verma<sup>1</sup>, Chhabi Rani Panigrahi<sup>2</sup>, Bibudhendu Pati<sup>2</sup>, Joy Lal Sarkar<sup>3</sup>, Abhishek Majumder<sup>3</sup>

<sup>1</sup> Biju Patnaik University of Technology,  
India

<sup>2</sup> Rama Devi Women's University,  
India

<sup>3</sup> Tripura University,  
India

rajeshverma\_chicago2004@yahoo.com, {panigrahichhabi, patibibudhendu,  
joylalsarkar, abhi2012}@gmail.com

**Abstract.** Multimedia comprises of different types of data formats from various sources. Handling the task of processing complex multimedia data is a big challenge and requires offloading to the cloud. The result is subsequently passed back to the Smart Mobile Device (SMD). This processing activity requires substantial amount of processing power and large network bandwidth of the cloud. As the output on the SMD needs to be of superior quality, there is a need for having a robust architecture which can run these resource intensive multimedia applications in an efficient manner. This work proposes an architecture which is assisted by the cloud, and helps in processing the multimedia applications in a fast and efficient manner. Also, the cloudlet assisted architecture has been proposed, where a mobile device will select a cloudlet based on the Borda Scores Method (BSM) and based on the request, the selected cloudlet then acknowledges the multimedia sensor nodes and the results are sent back to the respective mobile device.

**Keywords.** Cloudlet, mobile device, public cloud, SMD, BSM.

## 1 Introduction

The growing popularity of mobile devices in the present world is indicative of the fact that in the future, the world would expect almost all different

types of applications on the mobile devices like SMDs, tablets, laptops, etc. This will help to bring all the control in the hands of the user via the mobile devices [1, 20, 21, 22]. In today's world, we find that every person in all parts of the world carries a mobile in their hand. There are close to  $6.8 * 10^9$  mobile devices (connections / 100 citizens = 97 mobiles), and the connection speed will increase to 6.5 Mbps by 2020 [2].

Looking from the business perspective, it is estimated that the market of Mobile Cloud will touch around 46.90 Billion USD by 2019 [2]. Mobile devices come in very handy when you want to run any mobile application at any place and get instant service like m-commerce, checking price, paying for your services, GPS, medical or any other services [3, 23]. On the contrary, when using a mobile device, we need to compromise on the battery life, network connectivity quality using 3G / 4G and security concerns. Currently, we are experiencing a convergence of both the mobile devices and cloud computing. The cloudlet plays an important role in bringing both of these together [4].

Cloudlets comprise of a network of various mobile devices (example – SMD's, Tablets, laptops, etc.) which provide the user with the resources

(compute, storage and network) to enable the provisioning of his services to build applications at a lower cost [5]. The cloudlets have a unique property that they are deployed close to the users and also built on standard cloud architecture. Cloudlets possess sufficient computational and storage power and are secure enough as they mainly work in a hostile environment (in the absence of Internet) [6]. The main reason of preferring cloudlets is that they require less power compared to huge data centers in the cloud and can be deployed with ease in any location [5, 23]. Hence, cloudlet is a cheaper computational model and is gaining popularity. Cloudlets provide great advantage in case of real-time tasks as the close proximity to computational resources is a big advantage.

The offloading of task to the cloud is very often required when it more resource power to run the particular application as compared to the computational power that is delivered by the mobile devices. The total setup and running cost required in case of cloudlets is very negligible as compared to the computational cost to be paid to the cloud provider [6]. However, the different mobile devices connected together may experience network issues due to mobility reasons [28]. Multimedia has been gaining popularity since 1966, when the term was first coined in the movie world [7]. It involves the aggregation of varied type of content which is quite complex ranging from simple text files to rich media content. Several fields such as gaming, education and training, medical, etc. use different types of simulations using multimedia techniques extensively [8].

The advantage delivered by the use of multimedia is the superior quality of images and effects which gives a rich user experience and impact. The different areas that help drive multimedia are use of robust algorithms, graphics, cameras and SMD's [7, 24]. Although new models of mobile devices are being launched in the market at a very fast pace, they cannot cope up with the degree of advancement in multimedia applications which require relatively large computation power and storage capability. With a view to bridge the gap between the mobile devices and multimedia applications, the concept of Cloud Mobile Media

(CMM) is helping out as it is now possible to run the resource intensive tasks onto the cloud and render the results onto the mobile devices using the Software as a Service (SaaS) model, thereby boosting the revenue of the cloud providers [8, 26]. The different applications of CMM are the Audio/Video streaming based services which make use of the cloud resources in order to massage the content and deliver it via different devices, and secondly, the interactive services such as video conferencing, mobile remote desktops and others [10, 25].

### 1.1 Motivation

Multimedia is gaining popularity as it provides the capability to see and hear different formats of audio and video (like swf, .wav, .mp3, .mp4, .mpg, .wmv, and .avi) [8]. There are various applications such as executive presentations, training, advertising, marketing, network communication, entertainment, gaming and others which leverage multimedia features [7]. Also, the computation involved is quite complex and the resource usage is high. Under these circumstances, it is essential to utilize the services of the cloud by offloading the computation and resource intensive tasks, which causes latency and also incurs network costs [1].

Cloudlets incur negligible network costs as the resources can be easily found within the cloudlet present in the vicinity and hence low latency is experienced. However, due to the mobility factor of the mobile devices, we may encounter the problem of signal loss at some times. None the less, cloudlets are becoming very popular and especially in case of multimedia applications, they are being used extensively because of their capability to deliver using negligible network costs and good speed of delivery of content [5]. This has created an interest and in this work, we have proposed an architecture using cloudlets for handling of multimedia applications.

### 1.2 Contributions

In this section we have highlighted the major contributions of this work and are as follows:

1. Firstly, a system description has been given which provides an overall description about the mobile cloudlet system.
2. Secondly, a mobile cloudlet assisted architecture for handling multimedia applications has been proposed.
3. Finally, the simulation results of our proposed approach have been provided. Also, the comparison between the proposed approach and the three baseline algorithms is provided.

## 2 Related Work

In [6], the authors discussed about the Cloud Mobile Media (CMM) applications with respect to various parameters such as cost, response time, etc. [8]. In case of gaming applications, the tasks that require more computational power are run on the cloud and the output is displayed on the SMD [10]. In [11], the authors discussed about the integration of MCC and WSN in the “Always On” (AO) mode such that the sensor node is always active and keeps sending data to the cloud for computation, which is advantageous as the data gets accumulated through the use of WSN.

The mobile users will mainly request data of the nearby geographical regions only and also the sensors possess limited power. To this effect, the authors suggested two approaches, Collaborative Location-based Sleep Scheduling (CLSS1) where the sensor is turned on or off depending on the location. CLSS2 emphasizes on the performance factor of the integrated WSNs [13]. In [12], the authors used a cloudlets based algorithm for integration named as CLIW, which works with CLIW 1, and CLIW 2 and ensures that energy spent is minimum.

In [14], the physical sensor is virtualized on the cloud and subsequent provisioning can be done depending on the needs of the user and destroyed once the work has been completed. The architecture proposed in [7] illustrates the mechanism of sensor virtualization. In [8], an architecture has been proposed which depicts the use of sensors which are present in the human body for the monitoring of various body

parameters which will ultimately help the doctors to make an informed decision about the disease of the patient. In [15], proposed a robust integrated framework which reduces the storage requirements at the sensor end and the gateway takes care of performing various activities for further processing of the data which has been derived from the sensors. In [16], the authors proposed an approach named TPSS, which includes both Time and Priority Selective Data Transmission (TPSDT) and Priority-based Sleep Scheduling (PSS) approaches for energy saving. In TPSDT, the sensor data is transferred selectively based on the time and usefulness of the data to the user.

## 3 Definitions and Assumptions

In this section, we have described the mobility scenario of mobile users and cloudlets.

### 3.1 Mobility and Availability of Mobile Users

Let us consider that a mobile user  $M_a \in M$  is present at a particular location  $\mathbb{L}$ . It is possible that the user can move to another location  $1 - \mathbb{L}$ . In a real life scenario, the user spends his time at different locations and a good amount of time at other locations which the user visits frequently [17]. According to the concept of community, a large number of users are assembled in a specific region [17]. Let us consider that a user will spend his total time  $t(n)$  at a particular location and after sometime shift his location to a new place. Here,  $t(c)$  is the time spent in the community by the user and  $t(s)$  represents the time spending in a community. The user's arrival probability is  $P^A$  and can be expressed as follows [17]:

$$P^A = \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] = \lim_{n \rightarrow \infty} \mathbb{E} [\gamma(n)], \quad (1)$$

where,  $\gamma(n) = \frac{t(c)}{t(n)}$ .

Hence, the possible number of users available at a particular location  $\mathbb{L}$  can be expressed as follows:

$$\cup(M) = M \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right). \quad (2)$$

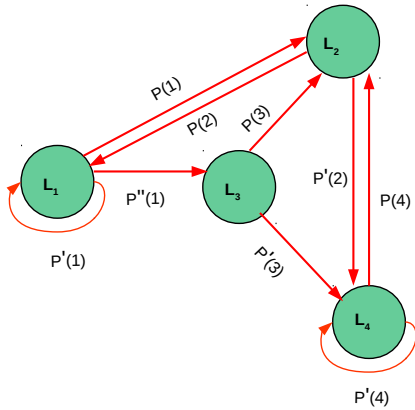


Fig. 1. Transition diagram of the mobile user's mobility

The transition probability of the mobile users can be expressed as follows:

$$P^T(M_i, M'_i) = e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M'_i} \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)^{M'_i}}{\left[ M'_i \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right) \right]^!}, \quad (3)$$

where, \$\forall M\_i \in \{0, 1, 2, 3, \dots, N\} \in M\$ and \$\forall M'\_i \in \{0, 1, 2, 3, \dots, N - 1\} \in M'\$ and \$\alpha\$ is the intensity of points and \$r\$ represents the radius of a circle.

The probability that the number of mobile users \$M\_i\$ are present in the range of a Base Station (BS) can be expressed using Eq. 4:

$$P^B(M_i, N') = e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_k} \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)^{M_k}}{\left[ M_k \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right) \right]^!}. \quad (4)$$

In the transition matrix the rows indicate that the number of mobile users available at time slot \$t\_a\$ and the columns indicate the number of mobile users \$M'\_i\$ who are likely to jump to the next community as given in Eq. 5.

The user's departure probability denoted as \$P^D\$ and can be expressed as follows [17]:

$$P^D = 1 - \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] = 1 - \lim_{n \rightarrow \infty} \mathbb{E} [\gamma(n)]. \quad (6)$$

Hence, the possible number of users who are departing from a particular location \$(1 - \mathbb{L})\$, can be expressed as follows:

$$\cup(M_i) = M_i \left( 1 - \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right). \quad (7)$$

The probability of a mobile user jumping from a location \$\mathbb{L}\$ to the adjacent location \$(\mathbb{L} + 1)\$ by making \$m\$ number of jumps in a given time \$t\$, can be expressed as follows:

$$P^l = P^l \{Y_{t+m} = \mathbb{L} | Y_t = \mathbb{L}\}. \quad (8)$$

Let us consider that a mobile user changes his location across different locations \$L\_1, L\_2, L\_3\$, and \$L\_4\$ as shown in Fig. 1. According to Fig. 1, the mobile users must satisfy the following conditions:

$$P(1) + P'(1) + P''(1) = 1, \quad (9)$$

$$P(2) + P'(2) = 1, \quad (10)$$

$$P(3) + P'(3) = 1, \quad (11)$$

$$P(4) + P'(4) = 1. \quad (12)$$

The transition state matrix \$T(G)\$ is depicted in Eq. 13-15.

$$T(G) = \begin{bmatrix} L_1 \rightarrow L_1 & L_1 \rightarrow L_2 & L_1 \rightarrow L_3 & L_1 \rightarrow L_4 \\ L_2 \rightarrow L_1 & L_2 \rightarrow L_2 & L_2 \rightarrow L_3 & L_2 \rightarrow L_4 \\ L_3 \rightarrow L_1 & L_3 \rightarrow L_2 & L_3 \rightarrow L_3 & L_3 \rightarrow L_4 \\ L_4 \rightarrow L_1 & L_4 \rightarrow L_2 & L_4 \rightarrow L_3 & L_4 \rightarrow L_4 \end{bmatrix}, \quad (13)$$

$$T(G) = \begin{bmatrix} P'(1) & P(1) & P''(1) & 0 \\ P(2) & 0 & 0 & P'(2) \\ 0 & 0 & P(3) & P''(3) \end{bmatrix}, \quad (14)$$

$$T(G) = \begin{bmatrix} P'(1) & P(1) & P''(1) & 0 \\ P(2) & 0 & 0 & P'(2) \\ 0 & 0 & P(3) & P''(3) \end{bmatrix} \equiv \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}. \quad (15)$$

$$A = \begin{bmatrix} e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_0} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} & e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_1} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} & \dots & e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_k} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_k \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_0} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} & e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_1} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} & \dots & e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{M_k} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right]}{[M_k \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t(c)}{t(n)} \right] \right)]!} \end{bmatrix}_{(N+1)^2} \quad (5)$$

### 3.2 Mobility and Availability of Cloudlets

In a particular time period, a cloudlet can only be connected with a unique mobile device from the list of all available mobile devices (as shown in Fig. 2). Let us consider that the cloudlets spend total time  $t_s^C(n)$  and also time spent by the cloudlets in a community is  $t_s^C(c)$ . So the cloudlets arrival probability is  $P_C^A$  and is expressed as follows:

$$P_C^A = \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right] = \lim_{n \rightarrow \infty} \mathbb{E} [\beta(n)] \quad (16)$$

Hence, the possibility of availability of the cloudlets  $C$  and a particular location  $\mathbb{L}$  is expressed as follows:

$$\cup(C) = C \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right] \right). \quad (17)$$

The transition probability of the number of cloudlets can be expressed as follows:

$$P_C^T(C_j, C'_j) = e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{C_j} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right]}{[C'_j \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right] \right)]!}, \quad (18)$$

where,  $\forall C_j \in \{0, 1, 2, 3, \dots, K\} \in C$  and  $\forall C'_j \in \{0, 1, 2, 3, \dots, K-1\} \in C'$ .

The probability of the mobile user is assign to the available cloudlets in a particular  $\mathbb{L}$  in a transmission range  $R$  and can be expressed as follows:

$$P^B(C_j, K') = e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{C_p} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right]}{[C'_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right] \right)]!}. \quad (19)$$

Therefore, the probability that a mobile user can not accessed the cloudlets due to the problem of

intermittent network connectivity, can be expressed as follows:

$$P'^B(C_j, K') = 1 - e^{-\alpha\pi r^2} \frac{(\alpha\pi r^2)^{C_p} \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right]}{[C'_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_s^C(c)}{t_s^C(n)} \right] \right)]!}. \quad (20)$$

The rows in a T transition matrix indicate the number of cloudlets available and the columns indicate that the number of cloudlets  $C_j$  which can jump to the next community as given in Eq. 21.

### 4 Mobile Cloudlets Assisted Architecture

The mobile cloudlets assisted architecture is shown in Fig. 3 where the mobile devices are connected to the multiple Radio Access Networks (RAN) for accessing the Global System for Mobile (GSM) or Universal Mobile Telecommunications System (UMTS) services [7]. The user executes the CMM applications via the mobile devices.

The architecture works in two ways:

- When a mobile device sends the request to the nearest cloudlet for receiving data from the Wireless Multimedia Sensor Networks (WMSNs).
- When a mobile device offloads the CMM application to the nearest cloudlet.

There are several cloud service providers like Google, Amazon, etc. which provide storage facility for storing video, music, or any other documents. There are so many categories of CMM applications available and these include audio/video streaming, interactive services, cloud based rendering, cloud storage and many others. In this work, we mainly focused on cloudlet based

$$A' = \begin{bmatrix} e^{-\alpha\pi r^2} \frac{C_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} & e^{-\alpha\pi r^2} \frac{C_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} & \dots & e^{-\alpha\pi r^2} \frac{C_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} \\ \vdots & \vdots & \ddots & \vdots \\ e^{-\alpha\pi r^2} \frac{C_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_0 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} & e^{-\alpha\pi r^2} \frac{C_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_1 \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} & \dots & e^{-\alpha\pi r^2} \frac{C_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right)}{\left[ C_p \left( \lim_{n \rightarrow \infty} \mathbb{E} \left[ \frac{t_S^C(c)}{t_S^C(n)} \right] \right) \right]!} \end{bmatrix} \quad (K+1)^2 \quad (21)$$

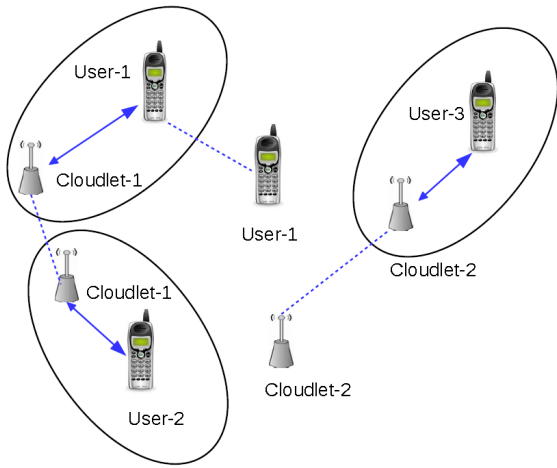


Fig. 2. Selection of cloudlet for mobile user's

rendering which involves on-line gaming. The mobile users offload some portion of the game to the cloudlet and during the game, the mobile users need to access cloudlets repeatedly.

The mobile device efficiently selects the best cloudlets by searching the nearby cloudlets availability. Generally a mobile device has to complete its large amount of tasks and also to initiate for sending the next task to the cloudlet once the previously sent task has been finished [17]. A mobile device has to maintain the cloudlets availability list which can be expressed as follows:

$$A(C) = \{C_1, C_2, C_3, \dots, C_k\} \in C. \quad (22)$$

The mobile device maintains the location string of all available cloudlets. Each mobile user determines the distance of all available  $C$ . Let us consider that a mobile device be present in the

Table 1. Task length of different events

Cloudlets	Locations(m)	Time Periods
Cloudlet-1	a	T(1)
Cloudlet-2	$\hat{a}$	T(1)
Cloudlet-3	b	T(2)
Cloudlet-4	$\hat{a}$	T(1)
Cloudlet-5	$\hat{b}$	T(2)

coordinate  $Q(x, y)$ . So the Euclidean distance can be computed using Eq. 23 [18]:

$$\Psi(i, Q) = \sqrt{(x - X(i))^2 + (y - Y(i))^2}. \quad (23)$$

Once the location string is prepared for any time period, a voting algorithm will be applied for selecting the best nearest cloudlet. From the cloudlet location string each mobile device assign a rank to each cloudlet based on the Borda Scores Method(BSM) of voting algorithm and is expressed as follows:

$$\lambda_M(i) = C - (C_i, R_{i,t}), \quad (24)$$

where  $(C_i, R_{i,t})$  represents the rank denoted as  $R_{i,t}$  of cloudlet  $C_i$ . Generally, a higher rank indicates the lower ordering in the preference list.

An example of cloudlets location list with cloudlets availability is given in Table.1 where two time periods have been taken. The cloudlets location string for respective time periods are maintained by the mobile devices and can be expressed as follows:

$$L_1^C = (C_1, a), (C_2, \hat{a}), (C_4, \hat{a}) \in C \text{ for } T(1) \in T, \quad (25)$$

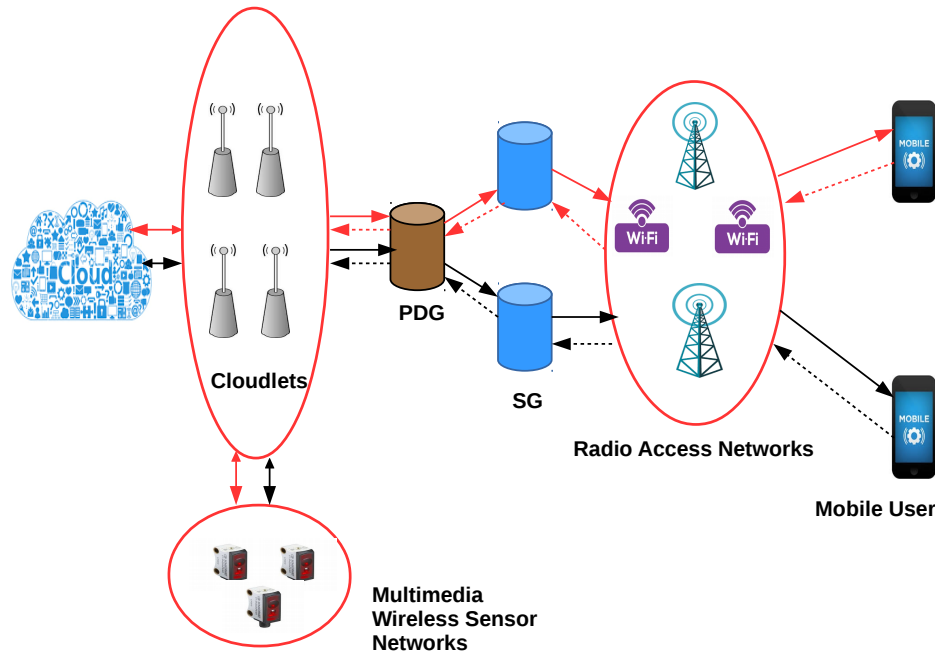


Fig. 3. Overview of mobile cloudlets assisted architecture

$$\mathbb{L}_2^C = (C_3, b), (C_5, \hat{b}) \in C \text{ for } T(2) \in T, \quad (26)$$

where T(1) and T(2) denotes as a respective time periods of T.

Let us consider that  $\hat{a} < \hat{a} < a$  for time period T(1). So the mobile device considers Cloudlet-4 as a winner and initiate further process like sending and receiving the tasks and so many. On the other hand, let us consider that  $b < \hat{b}$  for the time period of T(2). So the mobile device will select Cloudlet-3. The connection time between selected cloudlet and mobile device can be expressed as follows:

$$\mathbb{C}_C^M = \inf_{t>0} \|\sqrt{(x - X(i))^2 + (y - Y(i))^2}\| \leq \mathbb{R}, \quad (27)$$

where  $\|\cdot\|$  is the 2-D Euclidean norm and t is the time for available cloudlet and  $\mathbb{R}$  is the transmission range of the mobile device .

**Theorem 1.** *There is no transitive relation between mobile devices even if a cloudlet is in the range of two mobile devices.*

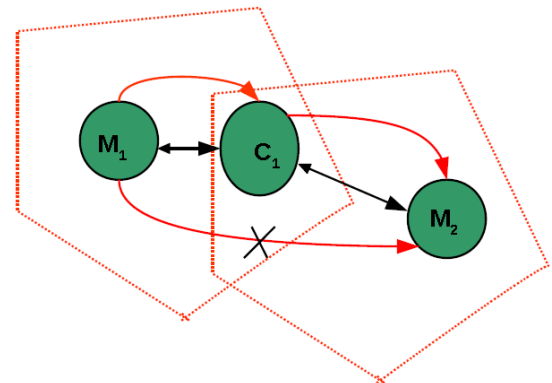


Fig. 4. Transition digram between mobile user and cloudlet

*Proof.* Let us consider that  $M_1 \prec C_1$  where  $\prec$  is the connection symbol and  $C_1 \prec M_2$  from Fig. 4. So  $M_1 \prec M_2$  contradicts the proof. Therefore, there is no transitive relation possible between mobile devices even if a cloudlet is in the range of two mobile devices.  $\square$

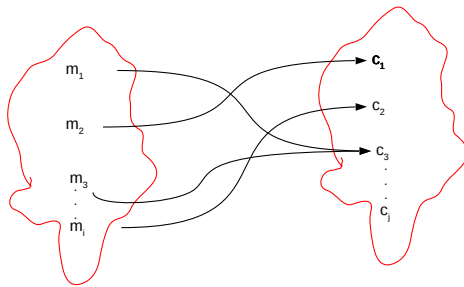


Fig. 5. Cloudlet selection of each mobile user

**Definition 4.1.** There is a characteristics function of  $f(M, C)$  which must satisfies the following conditions:

$$f(M, C) = \begin{cases} 0, & \text{if } (m_j \in M) \notin (c_k \in C) \\ 1, & \text{if } (m_j \in M) \in (c_k \in C). \end{cases}$$

**Theorem 2.** There is an injective relation among the set of mobile devices and the set of cloudlets under different timeslots.

*Proof.* Let us consider that a set of mobile devices  $m_1, m_2, m_3, \dots, m_i \in M$  and the set of cloudlets  $c_1, c_2, c_3, \dots, c_j \in C$ . Now according to the BSM every element of set  $\{m_1, m_2, m_3, \dots, m_j\}$  can select a unique element from set  $\{c_1, c_2, c_3, \dots, c_j\}$ . The system contradicts our assumption. So in the next time slot any element from  $\{m_1, m_2, m_3, \dots, m_i\}$  points to the same element of  $\{c_1, c_2, c_3, \dots, c_j\}$ . Consequently multiple mobile devices can access to the same cloudlet in a different time slot which is shown in Fig. 5. □

**Definition 4.2.** The mobile devices and cloudlets get close together if following condition must be satisfied:

$$R_d^l(\min) = \int_a \int_b \delta x \delta y \cap \int_m \int_n \delta p \delta z, \quad (28)$$

where, mobile devices and cloudlets are located in a 2-D space of  $\int_a \int_b \delta x \delta y$  and  $\int_m \int_n \delta p \delta z$ . The

limit  $a, b$  and  $m, n$  indicates that the available mobile devices and cloudlet in the corresponding axis.

**Theorem 3.** There may be efficient offloading if the cardinality of any set  $M$  is less then or equal to the cardinality of any set belongs to  $C$ .

*Proof.* Let us consider a set  $\{M_1, M_2, \dots, M_k\} \in M$  and a set  $\{C_1, C_2, \dots, C_l\} \in C$  means that each  $\{M_1, M_2, \dots, M_k\}$  can offload their applications to each  $\{C_1, C_2, \dots, C_l\}$ . The cardinality of  $\{M_1, M_2, \dots, M_k\}$  is  $|k|$  and for  $\{C_1, C_2, \dots, C_l\}$  is  $|l|$ . If  $|k| > |l|$  then at least some  $M_j \in M$  suffers for some  $C_j \in C$  but if  $|k| \leq |l|$  each  $M$  can assign to each  $C$ . Therefore, There may be efficient offloading if the cardinality of any set  $M$  is less then or equal to the cardinality of any set belongs to  $C$ . □

**Definition 4.3.** Each  $M \in PC_i$  if it satisfied the following condition:

$$Exec(ME, CE, PC) = \begin{cases} ME, & \text{if } E_c(M) \leq E_c(C) \\ CE, & \text{if } E_c(M), E_c(PC) > E_c(C) \\ PC, & \text{if } E_c(C) > E_c(PC). \end{cases}$$

where  $PC$ ,  $ME$ , and  $CE$  denotes as the cloud, mobile device, and cloudlet execution and  $E_c(C)$ ,  $E_c(M)$ , and  $E_c(PC)$  is the energy consumption for processing the data over cloudlet, mobile device, and cloud execution.

## 5 Performance Evaluation

In this section, we have described the experimental setup of our proposed approach and results along with the analysis of results.

### 5.1 Parameter Settings

For the experimentation of our work, we ran the simulation on the Dell machine having Intel Core I7-3770, 3.4 GHz Quad Core Processor. The disk storage was 500 GB and 4 GB memory was used with Java Development Kit (JDK) version 1.8. The simulation was run on the JAVA environment. The cloudlets were created with memory resources which varied from 10 to 20 GB.

Initially, we assumed that the cloudlets were moving between two areas namely A and B. The total time period of the movement of cloudlets was



divided into 5 time slots. In each of the time slot, the cloudlets were randomly assigned. The cloudlets were assigned to area A till the time its range was greater than the range of the area A. During the handoff period of the particular cloudlet from area A to area B, it is assumed that the cloudlets were assigned to area B by increasing the time slot by one. The sensor networks were created with a network size of  $500 \times 500 \text{ m}^2$ . The mobile user sends a request to the cloudlet and the cloudlet will take care of the request based on the state of the sensor node which is either in sleep or active state. The transmission and reception energy of the sensor nodes was taken as 0.021 mJ and 0.00601mJ with a transmission radius of 50 m.

For creating a public cloud, we used the Amazon Elastic Compute Cloud (EC2) setup [Amazon EC2(2016)]. The mobile devices have been created using the model as described in [Panigrahi et al. 2015]. The cloudlet list was maintained using the CreateCloudlet () method and the mobile devices selected the best cloudlet from the list of available cloudlets using the Java Comparator class. For deployment of the applications to the mobile devices, we used the characteristics of online gaming algorithm which can perform as a CMM application [Wang et al. (2014)]. Based on the application type and requirements, the proposed system can efficiently select the offloading strategy using the Definition 1.3.

## 5.2 Comparison with Baselines

In this work, a comparison of our proposed approach has been performed with three baseline algorithms, namely, AO [Zhu et al. (2014)], CLSS scheme [Zhu et al. (2014)] and CLIW scheme [Sarkar et al. (2016)]. The parameter chosen for comparison is the life time of the network.

## 5.3 Results and Discussions

The experimentation performed in this work, consists of generation of the cloudlets in a random manner in two different areas, which was having two separate cloudlet IDs. In Fig. 6 (a) and (b), the X axis represents the different cloudlet IDs with respect to the Euclidean metric from the mobile

user-1 and the mobile user-2. Fig. 7 (a) and (b), represents the Euclidean metric with respect to different cloudlet IDs. The mobile user selects the cloudlets based on the BSM as described in Equation (24). Fig. 8 shows the reverse preference list of the mobile user-1 and mobile user-2.

The value 0.45% indicates that the mobile device will give the highest preference where the distance between mobile user-2 and the cloudlet ID-7 is minimum with respect to the other available cloudlets. On the other hand, mobile user-2 gives lower preference to cloudlet ID-7. Fig. 9 represents the reverse preference percentage for mobile user-3 and 4. Fig. 10 represents the delay analysis by increasing the number of cloudlets for different mobile users.

The results shown in Fig. 10 (a) indicate that, even if the number of cloudlets increase, then the delay is maximum when the total number of cloudlets is 70. This is because sometimes the delay may decrease when there are sufficient number of cloudlets available and from the available cloudlets the mobile devices can choose the best cloudlet. The network lifetime of CLIW 1 and CLIW 2 scheme is described in [Sarkar et al. (2016)] and is expressed as follows:

$$Ne(life1)_{CLIW1} = \frac{I_o}{E^t \cdot (\psi \cdot \pi \cdot T^2) + E^r + \frac{P(E^t + E^r)}{2} - \omega(CC)} \cdot T. \quad (29)$$

here,  $E^t$  and  $E^r$  denote the consumption of energy for transmitting and receiving packets respectively.  $\omega(CC)$  denotes the extra communication cost which is considered as energy consumption due to high latency.

The network lifetime in CLIW2 is denoted as  $Ne(life2)_{CLIW2}$  and can be computed by using Eq. (30):

$$Ne(life2)_{CLIW2} = \frac{I_o}{E^t \cdot (\psi \cdot \pi \cdot T^2) + E^r + \frac{P(E^t + E^r)}{2} + \frac{(E^t + E^r)P^*}{2} - \omega(CC)} \cdot T. \quad (30)$$

The network lifetime of CLSS 1 scheme is expressed in [11] and is given in Eq. 31:

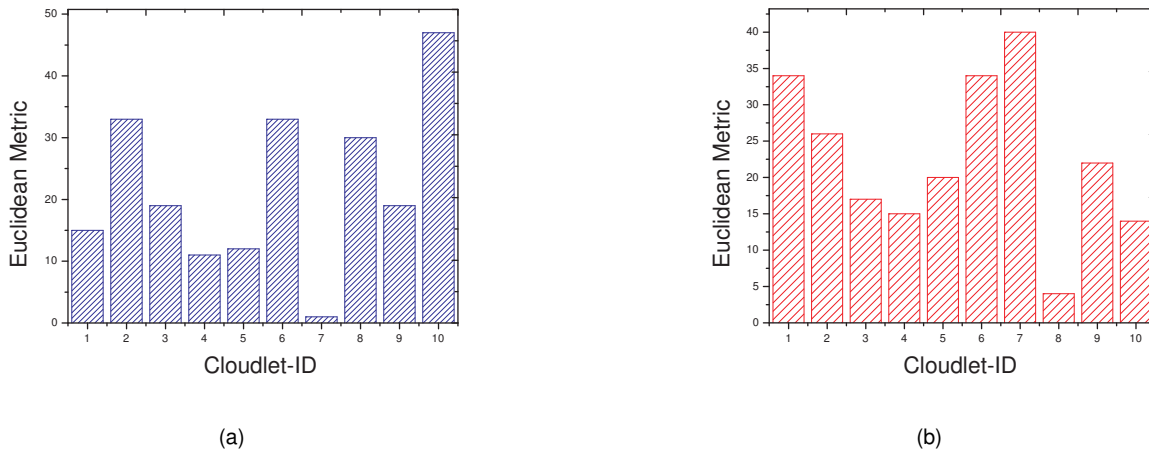


Fig. 6. Euclidean metric vs. Cloudlet ID

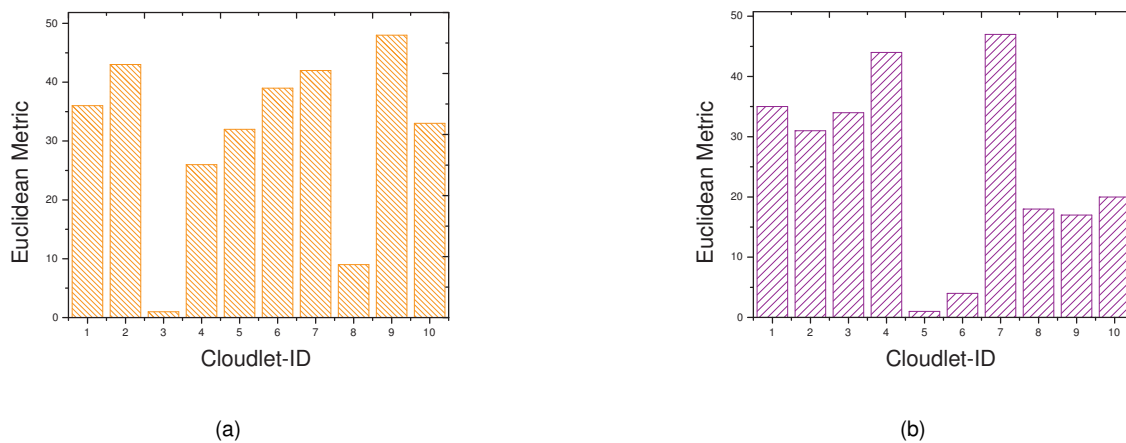


Fig. 7. Euclidean metric vs. Cloudlet ID

$$Ne(life1)_{CLSS1} = \frac{I_o}{E^t \cdot (\psi \cdot \pi \cdot T^2) + E^r + \frac{P(E^t + E^r)}{2}} \cdot T, \tag{31}$$

here,  $E^t$  and  $E^r$  denotes energy consumption that take place for transmitting and receiving the packets respectively.  $\omega(CC)$  denotes the cost of communication due to high latency, and is the overhead cost. The network lifetime in  $CLSS2$  is denoted as  $Ne(life2)_{CLSS2}$  and can be computed using Eqn. (32):

$$Ne(life2)_{CLSS2} = \frac{I_o}{E^t \cdot (\psi \cdot \pi \cdot T^2) + E^r + \frac{P(E^t + E^r)}{2} + \frac{(E^t + E^r)P^*}{2}} \cdot T. \tag{32}$$

The network lifetime of AO scheme is denoted as  $Ne(life)_{AO}$  and can be computed using [11]:

$$Ne(life)_{AO} = \frac{I_o}{E^t \cdot P + E^r \cdot P}. \tag{33}$$

Fig. 10(b) and Fig. 11 show the comparison of the proposed approach with respect to the three baselines algorithms. The results indicate that in

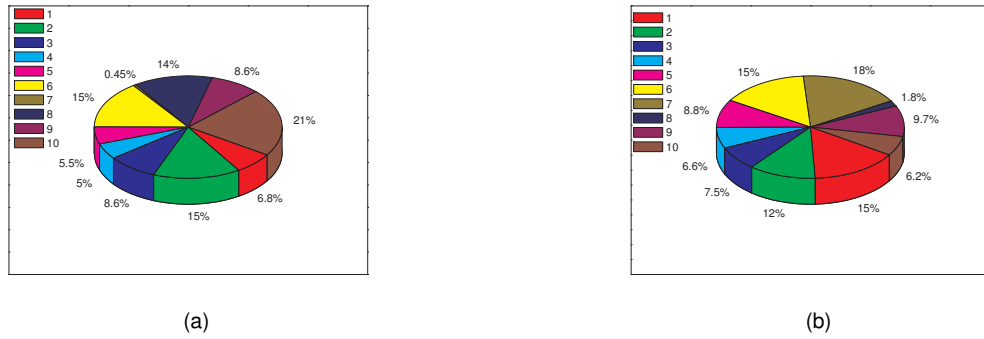


Fig. 8. Percentage of Cloudlet reverse preferences

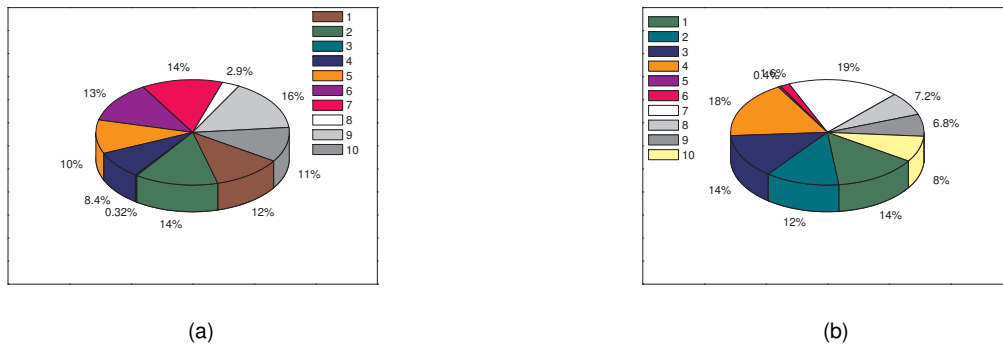


Fig. 9. Percentage of Cloudlet reverse preferences

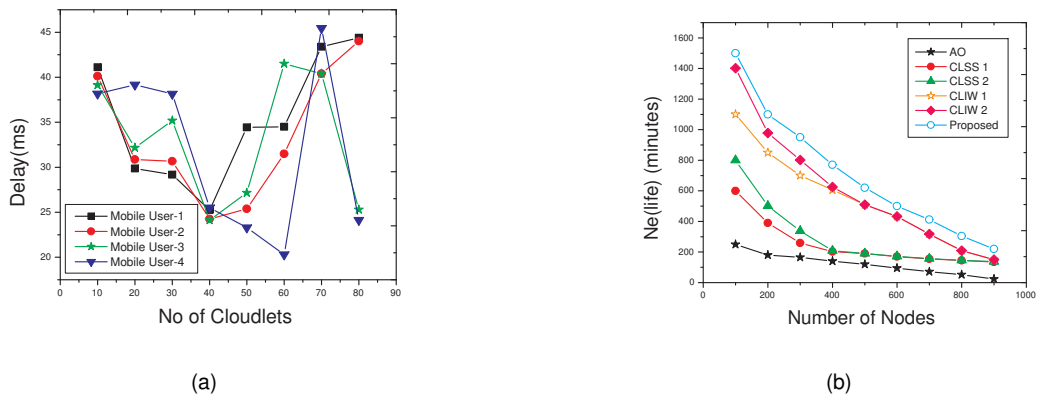
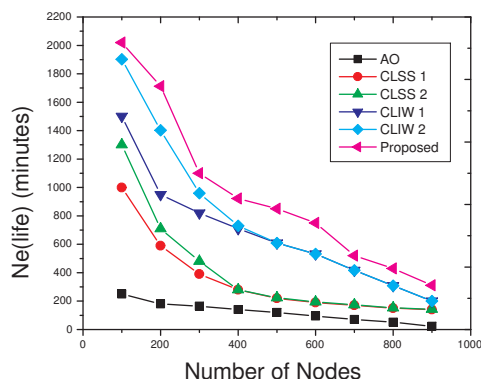


Fig. 10. (a) Analysis of delay with respect to the number of cloudlets, (b) Network lifetime of the proposed approach with respect to the baselines



**Fig. 11.** Network lifetime of the proposed approach with respect to the baselines

case of AO scheme, the network lifetime of sensor nodes is less with respect to CLSS scheme as it assumes that the sensor nodes are always on. On the contrary, the CLSS scheme considers the sleep scheduling technique which will reduce the energy consumption of sensor nodes. The CLIW scheme outperforms with respect to the CLSS scheme in terms computational speed.

On the other hand our proposed approach outperforms in comparison to the different baselines.

The proposed approach considers the optimum local cloud based on the BSM voting algorithm in which the mobile devices efficiently connect with the selected cloudlet and the result indicates a significant reduction in the communication overhead.

## 6 Conclusion

In this work, we have proposed a mobile cloudlet assisted architecture for running multimedia applications. The mobile device selects the best cloudlet based on the BSM voting algorithm. Based on the user's request, the selected cloudlet sends the request to the sensor nodes and the results are subsequently sent back to the mobile device via the cloudlet. In this paper, we mainly focus on the multimedia sensor nodes which are equipped with multimedia devices such as cameras, microphones, etc. The simulation

results clearly indicate that our proposed approach performs as compared to the three existing algorithms.

## References

1. Panigrahi, C.R., Pati, B., Tiwari, M., & Sarkar, J.L. (2015). Improving energy efficiency of mobile cloudlets using efficient offloading approach. *Proceedings of the 9th IEEE International Conference on Advanced Networks and Telecommunications Systems (IEEE ANTS)*, ISI Kolkata, pp. 1–6.
2. Cisco VNI (2019). <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/mobile-white-paper-c11-520862.html>, last accessed May 14, 2019.
3. Sanaei, Z., Abolfazli, S., Gani, A., & Buyya, R. (2014). Heterogeneity in mobile cloud computing: taxonomy and open challenges. *Communications Surveys & Tutorials*, 16(1), pp. 369–392.
4. Zhou, B., Dastjerdi, A.V., Calheiros, R., Srirama, S., & Buyya, R. (2015). mCloud: A Context-aware Offloading Framework for Heterogeneous Mobile Cloud. *IEEE Transactions on Services Computing*, 99, pp. 1–14.
5. Satyanarayanan, M., Bahl, R.C.P., & Davies, N. (2009). The case for VM based cloudlets in mobile computing. *IEEE Pervasive Computing*, 8(4), pp. 14–23.
6. Dong, H., Wang, P., & Niyato, D. (2012). A dynamic offloading algorithm for mobile computing. *IEEE Transaction on Wireless Communication*, 11(6), pp. 1991–1995.
7. Wang, S. & Dey, S. (2013). Adaptive Mobile Cloud Computing to Enable Rich Mobile Multimedia Applications. *IEEE Transactions on Multimedia*, 15(4), pp. 870–883.
8. Wang, S., Liu, Y., & Dey, S. (2012). Wireless Network Aware Cloud Scheduler for Scalable Cloud Mobile Gaming. *Proceedings of the IEEE International Conference on Communications (ICC)*, Ottawa, pp. 2082–2086.
9. Wang, S. & Dey, S. (2009). Modeling and Characterizing User Experience in a Cloud Server Based Mobile Gaming Approach. *Proceedings of the IEEE Global Telecommunications Conference (IEEE GLOBECOM)*, Hawaii, USA.

10. **Zhu, C., Leung, V.C.M., Yang, L.T., & Shu, L. (2014).** Collaborative Location-based Sleep Scheduling for Wireless Sensor Networks Integrated with Mobile Cloud Computing. *IEEE Transactions on Computers*, 20(20), pp. 1–14.
11. **Zhu, C., Sheng, Z., Leung, V.C.M., Shu, L., & Yang, L.T. (2015).** Toward Offering More Useful Data Reliably to Mobile Cloud From Wireless Sensor Network. *IEEE Transactions on Emerging Topics in Computing*, 3(1), pp. 84–94.
12. **Sarkar, J.L., Panigrahi, C.R., Pati, B., & Prasath, R. (2013).** MiW: An MCC-WMSN Integration Approach for Performing Multimedia Applications. *International Conference on Mining Intelligence and Knowledge Exploration*, pp. 83–92.
13. **Bi, H. & Gelenbe, E. (2014).** A cooperative emergency navigation framework using mobile cloud computing. *29th International Symposium on Computer and Information Sciences (ISCIS)*. Krakow, Poland, pp. 41–48, Springer International Publishing.
14. **Zhu, C., Wang, H., Liu, X., Shu, L., Yang, L.T., & Leung, V.C.M. (2014).** A novel sensory data processing framework to integrate sensor networks with mobile cloud. *IEEE Syst. J*, 99, pp. 1–12.
15. **Zhu, C., Sheng, Z., Leung, V.C.M., Shu, L., & Yang, L.T. (2014).** Towards Offering More Useful Data Reliably to Mobile Cloud From Wireless Sensor Network. *IEEE Trans. on Emerging Topic*, 3(1), pp. 84–94.
16. **Ge, Xiaohu, Ye, Junliang, Yang, Y., & Li, Qiang (2016).** User Mobility Evaluation for 5G Small Cell Networks Based on Individual Mobility Model. *IEEE Journal on Selected Areas in Communications*, 34(3), pp. 528–541.
17. **Misra, S. & Chatterjee, S. (2014).** Social choice considerations in cloud assisted WBAN architecture for post-disaster healthcare: Data aggregation and channelization. *Information Sciences*, 284, pp. 95–117.
18. **Amazon EC2 (2019).** <https://aws.amazon.com/ec2/>, Last accessed: 14/05/2019.
19. **Sarkar, J.L., Panigrahi, C.R., Pati, B., Saha, A.K., & Majumder, A. (2019).** MAAS: A mobile cloud assisted architecture for handling emergency situations. *Int J Commun Syst.*, e3950. DOI: doi.org/10.1002/dac.3950.
20. **Panigrahi, C.R., Verma, R.K., Sarkar, J.L., & Pati, B. (2018).** Energy-Aware Issues for Handling Big Data in Mobile Cloud Computing. In: **Skourletopoulos, G., Mastorakis, G., Mavromoustakis, C., Dobre, C., Pallis, E. (eds)** *Mobile Big Data. Lecture Notes on Data Engineering and Communications Technologies*, vol 10.
21. **Panigrahi, C.R., Sarkar, J.L., Pati, B., & Bakshi, S. (2016).** *E<sup>3</sup>M*: An Energy Efficient Emergency Management System using mobile cloud computing. *IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, Bangalore, pp. 1–6.
22. **Pati, B., Panigrahi, C.R., & Sarkar, J.L. (2018).** CETM: a conflict-free energy efficient transmission policy in mobile cloud computing. *International Journal of Communication Networks and Distributed System*, 20(2), pp. 129–142.
23. **Panigrahi, C.R., Sarkar, J.L., Tiwary, M., Pati, B., & Mohapatra, P. (2019).** DATALET: An approach to manage big volume of data in cyber foraged environment. *Journal of Parallel and Distributed Computing*, 131, pp. 14–28.
24. **Sarkar, J.L., Panigrahi, C.R., Pati, B., Trivedi, R., & Debbarma, S. (2018).** E2G: A Game Theory-Based Energy Efficient Transmission Policy for Mobile Cloud Computing. In: **Saeed, K., Chaki, N., Pati, B., Bakshi, S., & Mohapatra, D. (eds)** *Progress in Advanced Computing and Intelligent Engineering. Advances in Intelligent Systems and Computing*, vol 563.
25. **Panigrahi, C.R., Sarkar, J.L., & Pati, B. (2018).** Transmission in mobile cloudlet systems with intermittent connectivity in emergency areas, *Digital Communications and Networks*, 4(1), pp. 69–75.
26. **Pati, B., Sarkar, J.L., Panigrahi, C.R., & Debbarma, S. (2018).** eCloud: An Efficient Transmission Policy for Mobile Cloud Computing in Emergency Areas. In: **Sa, P., Sahoo, M., Murugappan, M., Wu, Y., & Majhi, B. (eds)** *Progress in Intelligent Computing Techniques: Theory, Practice, and Applications. Advances in Intelligent Systems and Computing*, vol 719, Springer, Singapore.
27. **Báez Esquivel, E., Rivero-Angeles, M.E., & Fernandez-Vazquez, A. (2018).** Priority Scheme for Enhancing the Capacity of Peer-to-Peer Networks in Mobile Environments. *Computación y Sistemas*, 22(4), pp. 1443–1454.

Article received on 14/01/2019; accepted on 03/05/2019.  
Corresponding author is Rajesh Kumar Verma.