Check for updates

Received: 17 October 2019 | Revised: 17 March 2020 | Accepted: 30 April 2020

DOI: 10.1002/spe.2850

SPECIAL ISSUE PAPER

WILEY

EFFORT: Energy efficient framework for offload communication in mobile cloud computing

Saif U. R. Malik[1,2] | **Hina Akram**[3] | **Sukhpal Singh Gill**[4] | **Haris Pervaiz**[5] | **Hassan Malik**[6]

[1]Cybernetica AS, Tallinn, Estonia

[2]Department of Computer Science, COMSATS University, Islamabad, Pakistan

[3]COMSATS Institute of Information Technology, Islamabad, Pakistan

[4]School of Electronic Engineering and Computer Science, Queen Mary University of London, London, UK

[5]School of Computing and Communications, Lancaster University, Lancaster, UK

[6]Department of Computer Science, Edge Hill University, Ormskirk, UK

Correspondence
Haris Pervaiz, School of Computing and Communications, Lancaster University, Lancaster, UK.
Email: h.b.pervaiz@lancaster.ac.uk

Summary

There is an abundant expansion in the race of technology, specifically in the production of data, because of the smart devices, such as mobile phones, smart cards, sensors, and Internet of Things (IoT). Smart phones and devices have undergone an enormous evolution in a way that they can be used. More and more new applications, such as face recognition, augmented reality, online interactive gaming, and natural language processing are emerging and attracting the users. Such applications are generally data intensive or compute intensive, which demands high resource and energy consumption. Mobile devices are known for the resource scarcity, having limited computational power and battery life. The tension between compute/data intensive application and resource constrained mobile devices hinders the successful adaption of emerging paradigms. In the said perspective, the objective of this article is to study the role of computation offloading in mobile cloud computing to supplement mobile platforms ability in executing complex applications. This article proposes a systematic approach (EFFORT) for offload communication in the cloud. The proposed approach provides a promising solution to partially solve energy consumption issue for communication-intensive applications in a smartphone. The experimental study shows that our proposed approach outperforms its counterparts in terms of energy consumption and fast processing of smartphone devices. The battery consumption was reduced to 19% and the data usage was reduced to 16%.

KEYWORDS

cloud computing, cloud service providers, energy consumption, IoT, Internet spoofing, offload communication

1 | INTRODUCTION

Technological advancements in the field of information and communications technology (ICT) cause an explosive growth in the number of mobile devices accessing the wireless network. The development in Cloud Computing (CC) and wireless

This is an open access article under the terms of the Creative Commons Attribution License, which permits use, distribution and reproduction in any medium, provided the original work is properly cited.
© 2020 The Authors. *Software: Practice and Experience* published by John Wiley & Sons, Ltd.

communication technology has been the driving factor for such an exponential growth. It is anticipated that the total number of mobile devices in 2020 will reach 75 billion, while the volume of data would exceed 24.3 exabytes.[1] Recent reports have highlighted the impact and significance of Mobile CC (MCC). For example, according to a study by ABI Research, more than 240 million businesses will use services of cloud through mobile devices by 2015 and this will push the revenue of the MCC to $5.2 billion.[2] Furthermore, the use of smartphones has rapidly increased in numerous domains such as gaming, healthcare, enterprise, e-learning, and management of information systems. Even though the prophecy of mobile devices ruling the future computing devices; the smartphones alongside with their applications are still constrained by few limitations, such as memory, processor, and battery life. Nowadays, modern mobile devices have adequate resources such as fast processors, large memory, and sharp screens. However, it is still not enough to help with computing intensive tasks such as natural language processing, image recognition, and decision making.

To overcome the resource scarcity of smart devices, offloading computations is the demand in need.[3] MCC is a combination of CC and mobile computing that is generally a feasible solution to offload computations. However, the key barriers to offloading computation in MCC are the network bandwidth and latency.[4] If too many simultaneous offloading requests are made through the wireless access to the cloud, then severe interference may be created, resulting in a reduced data rates for data transmission.[3,5] Besides latency, since mobile devices contain enormous amount of personal data, there are concerns related to the data offloaded to the mobile cloud, such as trust, authentication, secure communication, privacy, and malicious attacks (like DoS).[6] Moreover, considering the "Mobile" nature of mobile devices, questions regarding what to offload, when to offload, and efficiency of offloading is still a research in demand.

Due to the advancement in the telecom industry, computing hardware has allowed the programmers to design complex applications in smartphones such as gaming apps, multimedia streaming, communication apps, and m-commerce. The goal is to facilitate the users to communicate effectively and efficiently. To address such communication intensive nature of applications, various solutions are proposed at different abstraction levels, from energy efficient hardware component (low-level) to user related solutions (high-level).[3] Some of the most prominent solutions are the one discussed in Reference 6, mobiByte,[7] Clone Cloud,[8] and MAUI system.[9] The total energy savings in the case of computation offloading is limited because it reduces the energy footprint of computationally intensive applications, which are not yet frequently used.[10,11] Considering smart phones having continuous and reliable connectivity, some interesting questions are which surrogate server to choose to offload data, when to decide for the offloading, and what data should be offloaded to meet the Quality of Experience (QoE) of the users.[12] There is a growing trend that service-oriented attributes like Quality of Service (QoS) does not cover every aspect of the application performance-related perspective.[8] As a consequence, the concept of QoE has gained strong interest in the research community.[13] Apart from the offloading decisions, another concern involved in MCC is network congestion control.[14] Mobile traffic is tremendously increasing, and network oper ators have to adopt new mechanisms to deliver smooth communication between users and cloud endpoints.[15]

In this research, we propose an ***E**nergy e**FF**icient framew**OR**k for offload communica**T**ion (**EFFORT**)* in MCC for communication offloading used for communication intensive applications that we believe will significantly reduce energy footprint for two main reasons. The first reason is, these applications are widely used by most smartphone users; and second, these applications are continuously or periodically run as a background process either as home screen widget or as background service and monitor some source of information on the Internet. Some examples include Really Simple Syndication (RSS) readers, social networking apps, weather information widgets, traffic information widgets, sports score services, and so on. To demonstrate the effectiveness and efficiency of our proposed approach, we developed a prototype application and benchmark it is using different Android devices in a real mobile cloud environment. The experiments were aimed to identify the network and battery consumption of the devices when the offloading mechanism is adopted.

Our Contributions: The main contributions of this research article are as follows:

a. Introducing a novel mechanism for efficient communication intensive applications on smartphones and analyzing them by identifying its requirements and crucial issues.
b. Implementation of communication offloading framework in the CC environment along with its evaluation with real-life applications.
c. Performance of proposed approach has been evaluated in cloud environment.
d. Addressing the challenges in MCC that require more investigation and elaboration.

The rest of the article is organized as follows: Section 2 explains the critical background concepts and terminology, including CC, the MCC concept, computation offloading and issues Smartphone's users are facing. Section 3 presents the MCC concept in detail. A comparison between the different offloading frameworks and their critical issues along

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

with the methodology and implementation overview of communication offloading framework is discussed in Section 4. Experimental validation is discussed in Section 5 and article is concluded in Section 6 along with some future directions in the area of CC.

2 | RELATED WORK

Even from the days of Simian Operating System (OS), people started using smartphones to perform several day-to-day activities that were earlier performed on Personal Computers (PC).[16] Due to the extensive range of smartphone applications, they are becoming the prime platform for computing for mobile users. Every year the number of smartphone shipments is increasing with the record growth rate of 16%. With the introduction of applications like 3G/4G Internet, people are not bound to sit in front of the machines (PC/Laptop) to use the Internet, but they can use it on a smartphone directly. Mobile computing is a promising technology that is gaining esteem as a mean to enlarge the potential of smartphone that is one of the resource-constrained devices. The models proposed for mobile cloud application development have been divided into different categories based on their efficiency, performance, memory, execution of an application that are the core objectives of the applications along with the offloading technique.

The term offloading is defined as a mechanism of shifting the specific computing tasks to an external platform, such as cloud, cluster, or grid, and so on. It may be essential due to the limitations of hardware in a computer system handling a specific task on its own. There are two types of offloading mechanisms; the first one is Computation offloading and the second one is Communication offloading. In computation offloading a clone image of the smartphone is formed which is comprised of the data and all the installed applications. Then the clone image is shifted to the cloud, where it dwells and executes on a virtual machine(s). The clone periodically synchronizes with the smartphone to keep the clone execution consistent and to facilitate computation offloading. Whereas, in communication offloading a clone method of communication is placed on the cloud, once a message is sent from the application to the clone method on the cloud, the clone method will forward that communication to the service on a cloud environment. There are two ends of that service, one resides on Cloud and other will be running on a mobile device.[17] The service on the cloud will fetch the data from clone methods on cloud and then push the data to the listener service to desired Smartphone. The service on the cloud is also responsible to identify the correct communication to correct mobile device. The listener service on a mobile device will identify the correct application and forward the data to the application. By doing this there will be only service running in mobile instead of each application running its on service. Table 1 shows the comparison between cloud and MCC in terms of the factors supported by respective technology. Moreover, in Table 2 a detailed comparison of existing offloading approaches is provided. We have precisely compared the existing offloading framework based on certain metrics. The "Cuckoo" framework is proposed in Reference 18 that provides an offloading mframework using Java Virtual Machines. The Ibis high programming system[24] provides the foundation for the communication components in Cuckoo. To use the framework, all of the applications have to be re-written to support local and remote processing. Although the authors in Reference 18 claimed to gain a considerable speedup, no specific figures are mentioned. Moreover, the framework is not adaptive, as soon as the mobile is connected to the cloud, the computations are offloaded. The other model is CloneCloud,[8] which is based on distributed execution by offloading certain parts of the applications. Unlike Cuckoo, the CloneCloud does not require programmer support for application execution. There are five basic distributed execution mode supported

TABLE 1 Mobile cloud and cloud computing comparison

Factors	Cloud computing	Mobile cloud computing
Context awareness	×	✓
Device energy	×	✓
Bandwidth utilization cost	×	✓
Network connectivity	×	✓
Mobility	×	✓
Location awareness	×	✓
Bandwidth	×	✓
Security	✓	✓

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

TABLE 2 Offloading techniques

Model	Context awareness	Latency	Bandwidth utilization	Scalability	Privacy
18	L	M	L	H	L
8	M	All	H	L	L
19	M	M	L	H	M
20	L	L	L	L	L
21	L	H	M & H	L	L
22	L	M & H	M	H	M
23	H	L	L	M	L
9	H	L	L	L	L

Note: L (low), M (medium), and H (high), cloud (C), nearby infrastructure (NI).

by CloneCloud, whose details can be found in Reference 25. The main advantage of CloneCloud is that it is adaptive and the offloading decisions are made by considering the offloading cost and other constraints. The offloading mechanisms proposed in References 19,20 are mainly focused to support flexibility and scalability. Both of the aforesaid model support heterogeneous components execution, where the component can either execute on the cloud or on the phone. Similarly, the models explained in References 21-23 and 9 support computational offloading. However, none of the aforesaid models, discussed or provided any framework to support communication offloading for data intensive applications. Therefore, in this article, we made an attempt to propose a framework that can facilitate communication offloading to achieve energy efficiency in mobile cloud. From the computation offloading solutions discussed in this section, following observations can be made:

1. There is a tradeoff between different factors in all of these solutions. The optimum solution will be the one that will not only lower the bandwidth usage but also address the processor usage, battery usage, privacy, and so on.
2. The optimum solution also needs to be, first, workable for developers. Considering the current trend creating any mobile application and making it available for App user is as easy as it can be. The first and most obvious question should be "Why a developer follows any of the purposed models for their application?"
3. Considering all the above facts we also need to re-consider our hypothesis that mobile devices need to offload communication/computation. We need to decide either if this is a problem in real or not. If so, then the solution provided above does not address all the areas.

The literature mainly focused on approaches that address the issues regarding computation offloading to overcome the limitations of mobile phones in terms of computation, memory, and battery. However, communication offloading is an under studied topic in the research community and we believe it can significantly improve the performance within the MCC environment.

3 | SYSTEM MODEL

Computation and communication offloading at a lower power and latency is an emerging area of research in the domain of MCC.[17] As shown in Figure 1, there are three main components in our proposed system. The first one is the *View*, which provides a UI for the Android application (communication or data intensive), cloud, and push notifications. Second is the *Controller*, which controls the clone cloud and native mobile application. Third is the *Model*, which defines the interaction between the client and the server.

Whenever a task has to be offloaded, there is a cost associated to it. The cost is usually computed in terms of power consumption or time required to perform the offloading. The power consumption of a task when its executed locally on the mobile device is computed as in (1):

$$\rho_{\mathrm{l}} = \rho_{\mathrm{m}} * \frac{i}{s_{\mathrm{m}}} \tag{1}$$

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

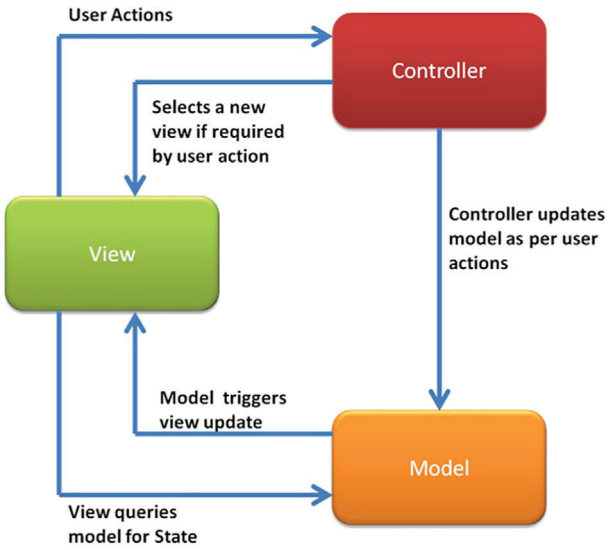

FIGURE 1 Basic components of proposed framework

where ρ_m is the power consumed by mobile device on each instruction, i is the total number of instructions to be executed, and s_m is the processing speed of the mobile. In contrast, the power consumption, in case of offloading the task to the cloud is computed as[26,27]:

$$\rho_o = \rho_m(I) + \rho_m(S) + \rho_m(R), \tag{2}$$

where

$$\rho_m(I) = i_\rho \left[\frac{i}{s_c} + \frac{D_{m,c}}{p}\right], \tag{3}$$

$$\rho_m(S) = s_\rho * \frac{\varphi_s}{\text{ר}_s}, \tag{4}$$

$$\rho_m(R) = r_\rho * \frac{\varphi_r}{\text{ר}_r}. \tag{5}$$

The power consumption for offloading (ρ_o) is mainly computed by adding up the power when the mobile is in idle state ($\rho_m(I)$), sending state ($\rho_m(S)$), and in receiving state ($\rho_m(R)$). The $\rho_m(I)$ is then further computed as the idle state power consumption of the device (i_ρ), speed of the cloud (s_c), distance between mobile device and the cloud ($D_{m,c}$), and the propagation delay (p). The $\rho_m(S)$ and $\rho_m(R)$ are computed in a similar passion, using the power consumption in respective states s_ρ and r_ρ, and the amount of data send and received (φ_s, φ_r) over the respective transmission rate (ר_s, ר_r).

In remote processing, the turnaround time is one of the most important parameters, especially in case of communication or data intensive applications. There are several phases, where the processing is conducted, which cause delays at several instances. In offloading process, several delays are encountered that makes up the turnaround time for the remote processing. The turnaround time includes[28]: (a) time taken to maintain the states of running instances of offloaded applications, (b) time to transfer the clone from a mobile device to the cloud, (c) time to download the clone at server, (d) time taken to set the preferences of mobile application to server node, (e) time to configure the clone to the server node, and (f) time to send the result to the mobile device.

4 | PROPOSED FRAMEWORK

This section presents the proposed solution for communication offloading. The proposed framework exploits the advantages of offloading to devise a relatively new technique of offloading communication in MCC. The main usage of Smartphone devices is communication, and some applications are communication expensive. We have identified a class of frequently used applications, information monitoring applications, requiring significant communication. The aforesaid

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

applications are continuously or periodically run as a background process, either as home screen widget or as background service and monitoring some source of information on the Internet.[16] Few examples of such applications are RSS readers, Social networking apps, weather information widgets, traffic information widgets, sports score services, and much more. This research targets these applications to offload communication, thus, enhancing the performance and battery life of Smartphone devices.

To test the effectiveness of the proposed communication offloading framework, experiments are conducted; writing applications using Software as a Service (SaaS) and placing its clone on the cloud component that is, Infrastructure as a Service (IaaS). Details of the experiment, along with its results will be presented in the following section.

4.1 | Communication offloading framework

The architecture of communication offloading framework is depicted in Figure 2. In the framework, for each application, there will be a clone method of communication placed on the cloud. Once a message is sent from the application to the clone method on the cloud, the clone method will forward that communication to the service on cloud environment.[17] There are two ends of that service, one resides on Cloud and other will be running on a mobile device.[17] The service on the cloud will fetch the data from clone methods on cloud and then push the data to the listener service to desired Smartphone. The service on the cloud is also responsible to identify the correct communication to correct mobile device. The listener service on a mobile device will identify the correct application and forward the data to the application. By doing this there will be only service running in mobile instead of each application running its own service.

To send communication from mobile using any application, there will be no need to involve cloud as this will cause an overhead. The applications in mobile can communicate directly with the live applications when it comes to sending data. The live application will create the object of the data needed to be sent, and forward it to its clone on the cloud by encrypting it with the Mobile Device ID that is authorized to receive the data. The clone that will add the application ID and send it to the service over the cloud that will push the object to the desired device based on Device ID. Once the object is received on the Smartphone, the service will push the data to right Application using the Application ID and finally, the data will be decrypted using the current mobile Device ID. In this scenario, every mobile has its Unique ID that will be used to encrypt and decrypt data for security. A unique application ID will be assigned to each application to ensure the data are sent to the correct application.

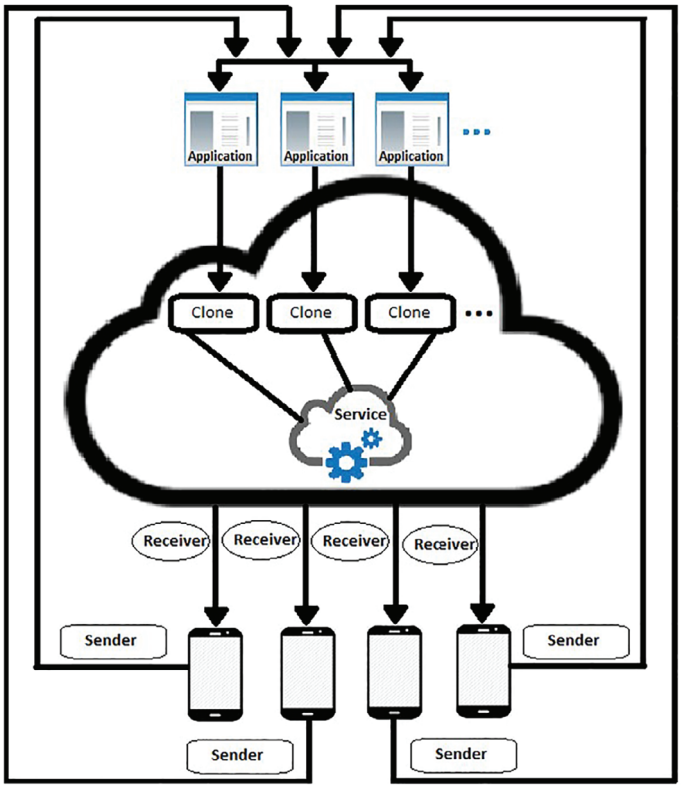

FIGURE 2 Communication offloading framework

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

4.2 | Implementation of communication offloading framework

There are three main modules of a mobile application regardless of its host operating system which are as follows:

Cloud Application: By using this module a listener method of application can be placed which is unique to all the applications and the purpose of this method is to listen to the messages sent from the application on the server. Meanwhile, a unique application ID along the message object is sent from the application server and afterward pushed to the cloud with the help of push service. *Push Service on Cloud*: Figure 3 depicts the sequence diagram of push service notification. This service, upon receiving the communication object from the above module will send to the designated mobile device by using the device ID.

The app on Mobile Phone: The actual application on Smartphone device will receive the message object from the CloneCloud, as shown in Figure 4. Upon receiving it the steps that will be performed are as follows:

1. By checking the Application ID, the object will be sent to the respective mobile application.

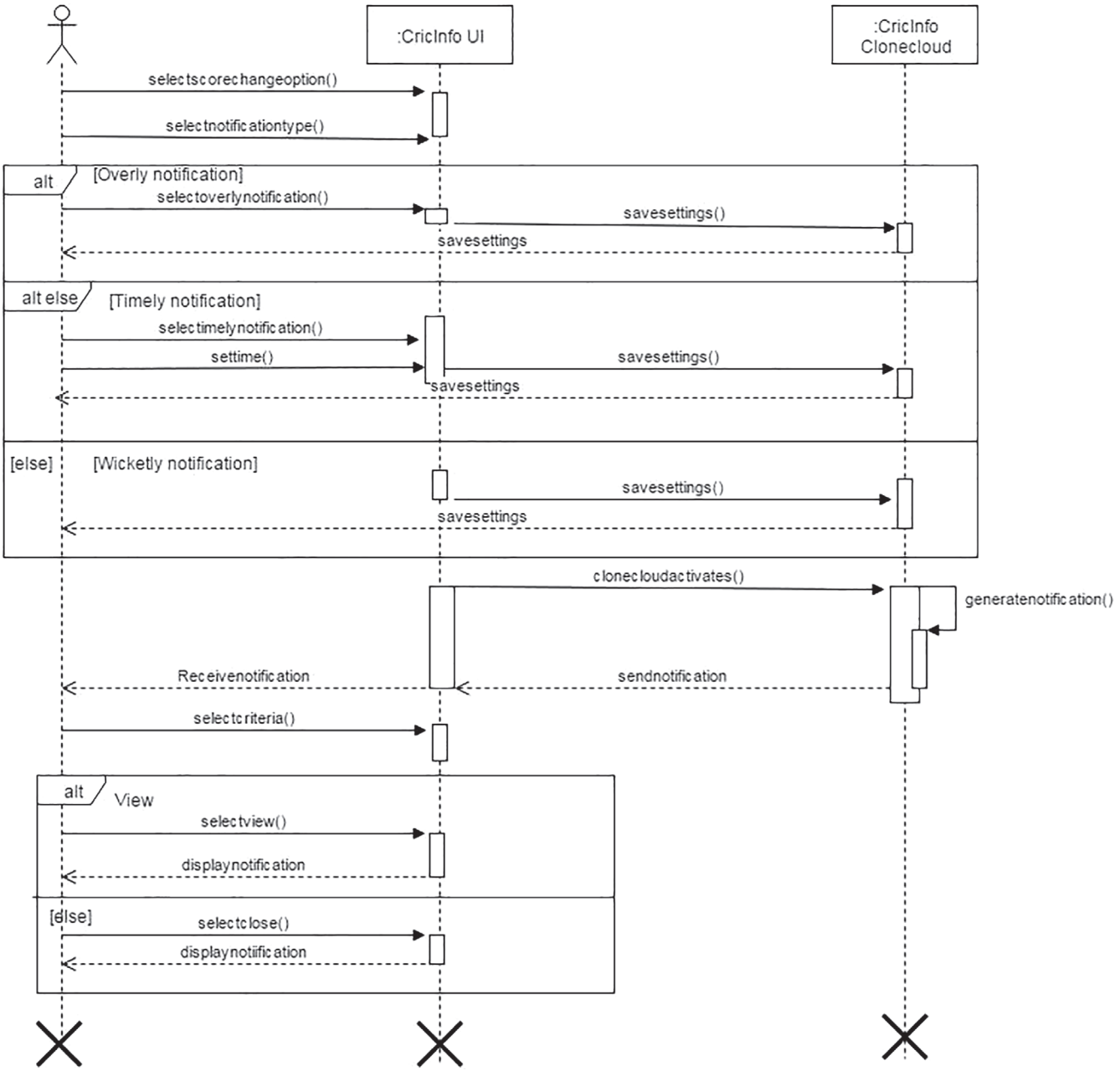

FIGURE 3 Sequence diagram of notification

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

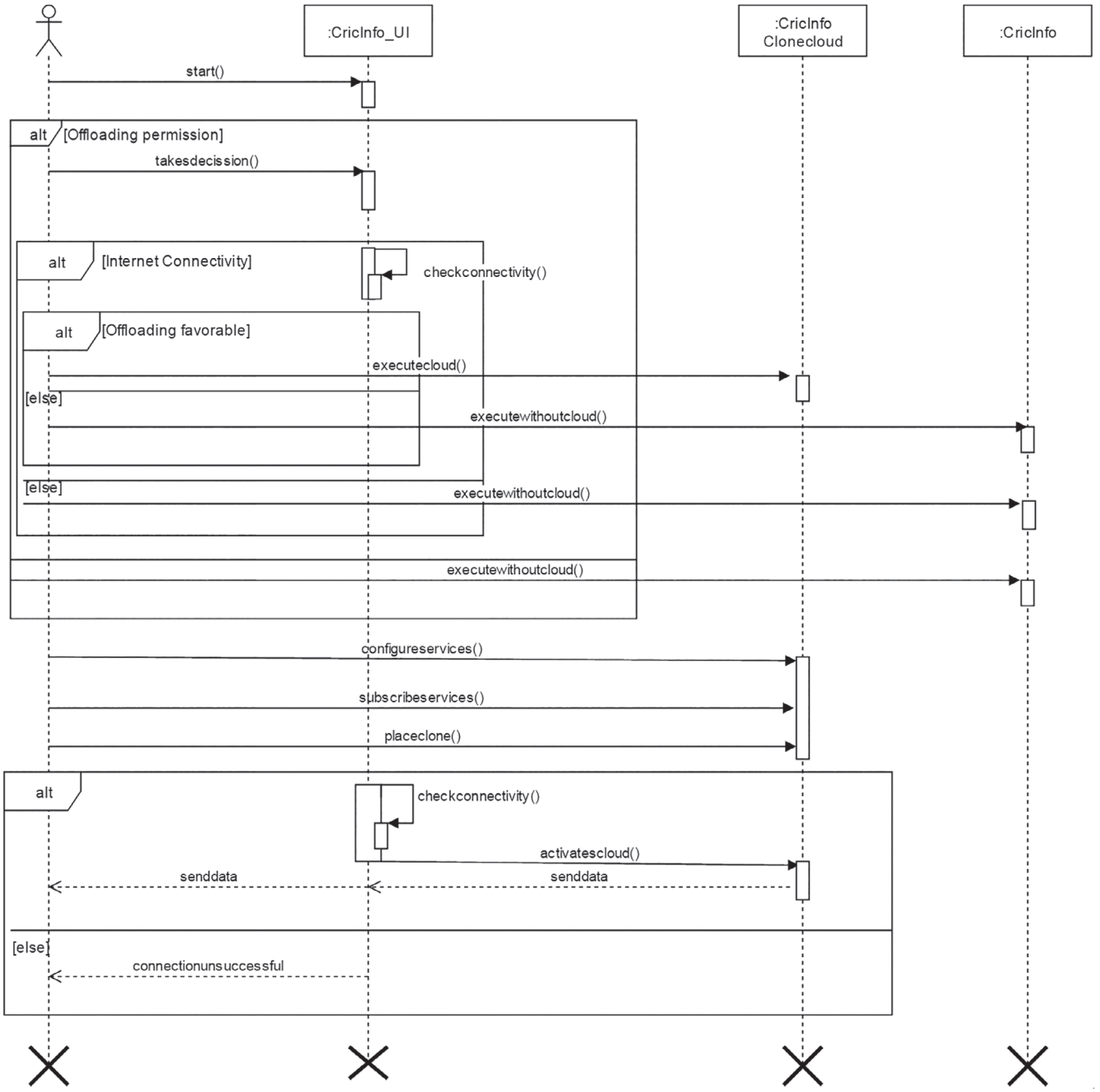

FIGURE 4 Sequence diagram for CloneCloud (Code of data sending request before implementing cloud)

2. Mobile Application will decrypt the message by taking the current Device ID. If the message is decrypted properly, it will be displayed in the app otherwise the package will be disregarded. This will ensure the message is sent to correct device and application.
3. A service needs to run on mobile that will receive all the objects coming from cloud service and assign them to the appropriate device.

5 | PERFORMANCE EVALUATION AND EXPERIMENTAL RESULTS

The experiments are performed on four different mobile vendors, namely Sony, Samsung, LG, and Q-Mobile. We developed an application in the Android platform for the validation of communication offloading. We choose Android platform

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

as it is compatible with the solution of communication offloading also it is an open source OS and the majority of Smartphone users uses it. To demonstrate the effectiveness and efficiency of our proposed approach, we developed a prototype application and benchmark it using different Android devices in a real mobile cloud environment. Several Android devices, shown in Table 3, (whose details are mentioned in the later section) were part of the experiment. As seen in Table 3, we chose different mobile devices from different vendors with different specifications, so as to test our prototype application on a diverse set of machines. Moreover, the experimental setup was consisting of a server node to run virtual Android device and an Internet connection (Wifi or 3G). The server machine was used to deploy the virtual device to offload the execution of the service mechanism. Java Development Toolkit was used to develop the prototype application. Several third-party tools, such as OS Monitor by EOLWRAL, were used to measure and monitor the resource utilization of the system. Furthermore, for battery consumption Power Tutor[28] Tool is used.

The prototype application that we build for our experimental analysis and evaluation is a sport score update app, which interacts with server and update the score board periodically. For intensive communication, we developed a smart phone application of CricInfo that interacts with the server and provide periodic updates. When the device is connected to the network the updates are sent using push notifications automatically. We developed two prototype of the application, one that uses offloading and the other that does not. As the mobile devices are resource constrained, our goal was to analyze the effect of offloading on the battery and other resources of the mobile devices. For the evaluation purpose, we have used the following devices:

5.1 | Battery consumption in apps with and without cloud

The battery was charged 100% at the beginning of the test. The normal processes were running in the background to analyze the overall status of the battery consumption. The results of using these applications for continuously for 2 hours are listed in Table 4. It can be clearly observed from the results depicted in Table 4 that the batteries of the devices are consumed more when the apps are executed without cloud. In case of cloud, the battery consumption is relatively less than the previous case. The results imply the energy efficiency can be achieved through offloading.

5.2 | Network usage test

The network test was conducted to check how much network is utilized while apps are executed on cloud and without cloud. Since, the network usage is directly related to the energy consumption of the device, the larger is the consumption the higher will be the energy footprints. For the test, we used the Apps on Wifi and 3G. We sent the same number of data

TABLE 3 Devices used in evaluation

Device name	OS	Battery
Sony Xperia T	4.3	1850 mAh
Samsung S5	5.0	2800 mAh
Q-Mobile Noir Z9	5.1.1	3000 mAh
Sony Xperia Z3	5.1.1	3100 mAh
LG Nexus 5	5.0	2300 mAh

TABLE 4 Battery consumption in apps

Device name	OS	Battery	Without cloud test end %age	With cloud test end %age
Sony Xperia T	4.3	1850 mAh	68	71
Samsung S5	5	2800 mAh	76	79
Q-Mobile Z9	5.1.1	3000 mAh	71	73
Sony Xperia Z3	5.1.1	3100 mAh	83	88
LG Nexus 5	5	2300 mAh	75	81

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

on both networks, namely Wifi and 3G for apps with and without cloud implementation. The results are shown in Table 5 and it is evident that the apps with the cloud have less network consumption, in both scenarios and on all devices, than the apps without cloud.

5.3 | Performance test

For performance, we analyzed our framework with non-cloud-based implementation alongside Wifi and 3G. For performance monitoring, we have used a free third-party tool from Android marketplace that is, OS Monitor by EOLWRAL. The goal of this test was to observe the overall effect of the offloading mechanism on the device. As we can see in Table 6, the performance of the devices with cloud execution is in all of the devices is relatively better than the without cloud environment.

TABLE 5 Network usage in apps

		Without cloud		With cloud	
Device name	**OS**	**Wifi**	**3G**	**Wifi**	**3G**
Sony Xperia T	4.3	1000 KB	998 KB	860 KB	890 KB
Samsung S5	5	990 KB	1008 KB	858 KB	901 KB
Q-Mobile Noir Z9	5.1.1	989 KB	992 KB	862 KB	888 KB
Sony Xperia Z3	5.1.1	999 KB	999 KB	866 KB	892 KB
LG Nexus 5	5	964.1 KB	1002 KB	858 KB	895 KB

TABLE 6 Performance monitoring in apps with and without cloud implementation

		Without cloud		With cloud	
Device	**OS**	**WiFiCPU usage**	**3GCPU usage**	**WiFiCPU usage**	**3GCPU usage**
Sony Xperia T	4.3	App1: 11.6	App1: 12.1	App1: 5.3	App1: 5.5
		App2: 11.6	App2: 12.1	App2: 5.3	App2: 5.5
		App3: 11.6	App3: 12.1	App3: 5.3	App3: 5.5
		App4: 11.6	App4: 12.1	App4: 5.3	App4: 5.5
Samsung S5	5	App1: 9.7	App1: 10.0	App1: 4.4	App1: 4.6
		App2: 9.7	App2: 10.0	App2: 4.4	App2: 4.6
		App3: 9.7	App3: 10.0	App3: 4.4	App3: 4.6
		App4: 9.7	App4: 10.0	App4: 4.4	App4: 4.6
Q-Mobile Noir Z9	5.1.1	App1: 13.3	App1: 13.9	App1: 6.9	App1: 7.1
		App2: 13.3	App2: 13.9	App2: 6.9	App2: 7.1
		App3: 13.3	App3: 13.9	App3: 6.9	App3: 7.1
		App4: 13.3	App4: 13.9	App4: 6.9	App4: 7.1
Sony Xperia Z3	5.1.1	App1: 8.6	App1: 9.4	App1: 3.9	App1: 4.3
		App2: 8.6	App2: 9.4	App2: 3.9	App2: 4.3
		App3: 8.6	App3: 9.4	App3: 3.9	App3: 4.3
		App4: 8.6	App4: 9.4	App4: 3.9	App4: 4.3
LG Nexus 5	5	App1: 10.3	App1: 10.7	App1: 5.8	App1: 6.2
		App2: 10.3	App2: 10.7	App2: 5.8	App2: 6.2
		App3: 10.3	App3: 10.7	App3: 5.8	App3: 6.2
		App4: 10.3	App4: 10.7	App4: 5.8	App4: 6.2

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

6 | DISCUSSION

We have conducted a series of experiments on several Android-based systems with different applications. The goal was to evaluate the significance of offloading communications from the mobile devices to the cloud and how much gains can be achieved. The experiments results revealed that the battery consumption drops when we adopt cloud model as described earlier. On average, with the entire devices that we have tested, we found that about 19% of battery is saved within 2 hours of testing, as shown in Figures 5 and 6. Both of the aforesaid figures indicate a drop in the energy consumption of the mobile devices.

In terms of network usage, we analyzed the connectivity of both, the Wifi and 3G, as shown in Figure 7. The devices we had were not all 4G compatible. However, we assume the results do not significantly differ in 4G as well. The data that are sent and received is quite evident that the proposed model helps in saving data as well. On average, from the test results, we can see that the data usage decreases by 16%.

From the performance perspective, we considered a worst-case scenario by executing all of the applications at once. Although, the said is a rare scenario, but it could still possibly happen. We noticed a slight hike in the cloud-based approach. However, even with the hike in performance, it is only 6.5% more usage of the processor.

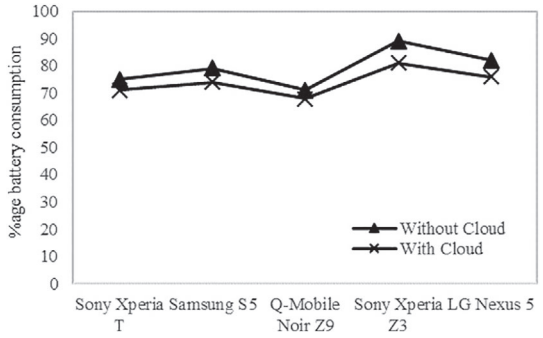

FIGURE 5 Battery consumption of device (Performance evaluation over 3G)

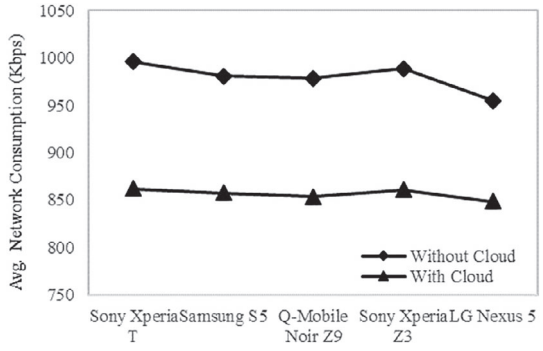

FIGURE 6 Difference in the battery consumption of the devices

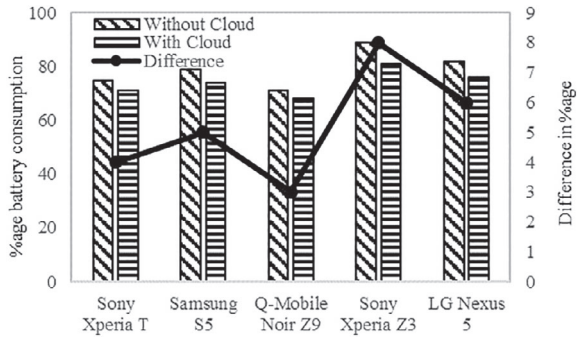

FIGURE 7 Average network consumption on Wifi and 3G using cloud offloading and without cloud

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

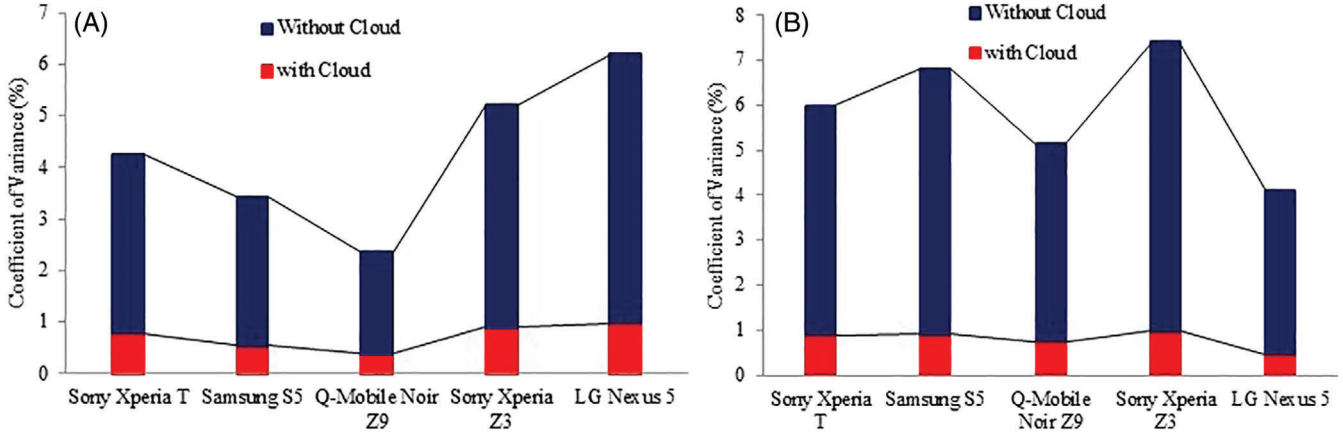

FIGURE 8 Coefficient of variation for cloud offloading and without cloud for (A) energy consumption and (B) network consumption

6.1 | Constraints and limitation

After thorough evaluation and experiments, we have observed the following limitations and constraints in our proposed approach.

- The true benefit of the proposed model is possible when there are several applications that use communication offloading. We believe that a single application may not get the best results out of the model.
- Since, we have focused on communication offloading only, the model is not suitable for computational offloading or may require significant changes in the model to support other operations.
- A centralized server is required, preferably by OS marketplace that will host all the application methods and has the service of cloud to Push data to relevant devices.

7 | STATISTICAL ANALYSIS

Statistical significance of the results has been examined by Coefficient of Variation (CoV), a statistical method which is used to equate to different means and moreover provide an overall study of performance of the proposed approach used for generating the statistics. It states the deviation of the data as a proportion of its average value, and is calculated as follows (Equation (6)):

$$\text{CoV} = \frac{\text{SD}}{M} \times 100 \tag{6}$$

where SD is a SD and M is a mean. CoV of energy consumption with cloud offloading and without cloud is shown in Figure 8A. Range of COV is (0.39%-0.97%) for energy consumption approves the stability of proposed framework. CoV of network consumption with cloud offloading and without cloud is shown in Figure 8B. Range of COV is (0.46%-0.99%) for network consumption approves the stability of proposed framework. Small value of CoV signifies proposed framework is more efficient and stable for offload communication in MCC.

8 | CONCLUSIONS AND FUTURE DIRECTIONS

In recent times CC has emerged as a new and attractive standard for distributing services over the Internet. Though, in spite of the fact that this standard provides numerous prospects to the various industries such as IT, business, and so on there are many issues in this area that still needs to be focused. In this article, we present the issue of battery consumption in Smartphones due to its performance of applications, storage, and computation these days. We have

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

conducted experiments which show that our proposed approach is an adaptive optimization framework for offload communication in CC to reduce the energy consumption in Smartphones along with increasing the speed of its intensive operations by a significant level. Furthermore, the aim of this article is to provide a better understanding of the challenges and discover major research directions in MCC as the recent technologies that exist are not adequate to comprehend its full potential. In future, our plan is to study the security and privacy attacks that can possibly be applied to cache, such as Cache Usage Attacks,[29,30] where the attacker monitors the CPU activities through caches on the physical machines. Moreover, to improve the cache hit ratio, a speculative mechanism can be developed that pre-fetches the data within the cache based on certain predictive parameter.

ACKNOWLEDGEMENTS

We would like to thank the editor, area editor, and anonymous reviewers for their valuable comments and suggestions to help and improve our research article.

ORCID

Sukhpal Singh Gill https://orcid.org/0000-0002-3913-0369
Haris Pervaiz https://orcid.org/0000-0002-8364-4682

REFERENCES

1. Wu J, Zeng M, Chen X, Li Y, Jin D. Characterizing and predicting individual traffic usage of Mobile application in cellular network. Paper presented at: Proceedings of the ACM International Symposium on Pervasive and Ubiquitous Computing and Wearable Computers; 2018:852-861; ACM.
2. Akherfi K, Gerndt M, Harroud H. Mobile cloud computing for computation offloading: issues and challenges. *Appl Comput Inform*. 2017.14:1–16.
3. Ghosh A, Khalid O, Rais RN, Rehman A, Malik SUR, Khan IA. Data offloading in IoT environments: modeling, analysis, and verification. *EURASIP J Wirel Commun Netw*. 2019;53(1):1-53.
4. Mach P, Becvar Z. Mobile edge computing: a survey on architecture and computation offloading. *IEEE Commun Surv Tutor*. 2017;19(3):1628-1656.
5. Chen X. Decentralized computation offloading game for mobile cloud computing. *IEEE Trans Parall Distrib Syst*. 2015;26(4):974-983.
6. Mollah MB, Azad MAK, Vasilakos A. Security and privacy challenges in mobile cloud computing: survey and way ahead. *J Netw Comput Appl*. 2017;84:38-54.
7. Khan AR, Othman M, Akhtar S, Khan AN, Madani S. MobiByte: an application development model for Mobile cloud computing. *J Grid Comput*. 2015;13 (1572–9184):1-24.
8. Chun BG, Ihm S, Maniatis P, and Naik M. Clonecloud: boosting mobile device applications through cloud clone execution; 2010. arXiv preprint arXiv:1009.3088.
9. Cuervo E, Balasubramanian A, Cho D, Wolman A, Saroiu S, Chandra R, Bahl P. MAUI: making smartphones last longer with code offload. Paper presented at: Proceedings of the 8th International Conference on Mobile Systems; 2010:49-62.
10. Han B. Mobile data offloading through opportunistic communications and social participation. *IEEE Explorer*. 2012;11:821-834.
11. Zhou Q, Xu M, Gill SS, Gao C, Tian W, Xu C, Buyya R. Energy efficient algorithms based on VM consolidation for cloud computing: comparisons and evaluations. Paper presented at: Proceedings of the 20th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing (CCGRID 2020); 2020:1-10.
12. Gill SS, Tuli S, Xu M, et al. Transformative effects of IoT, Blockchain and artificial intelligence on cloud computing: evolution, vision, trends and open challenges. *Internet of Things*. 2019;8:100118.
13. Reichl P. From charging for quality of service to charging for quality of experience. *Ann Telecommun*. 2010;65(3–4):189-199.
14. Gill SS, Garraghan P, Stankovski V, et al. Holistic resource management for sustainable and reliable cloud computing: an innovative solution to global challenge. *J Syst Softw*. 2019;155:104-129.
15. Mukherjee M, Kumar S, Shojafar M, Zhang Q, & Mavromoustakis CX. Joint task offloading and resource allocation for delay-sensitive fog networks. Paper presented at: Proceedings of the ICC 2019-2019 IEEE International Conference on Communications (ICC); 2019:1-7; IEEE.
16. Buyyaa R, Shin Yeo C. Cloud computing and emerging IT platforms: vision, hype, and reality for delivering computing as the 5th utility. *Future Generat Comput Syst*. 2009;25(06):599-616.
17. Ahmed E, Gani A, Khan MK, Buyya R, Khan SU. Seamless application execution in mobile cloud computing: motivation, taxonomy, and open challenges. *J Netw Comput Appl*. 2015;52:154-172.
18. Kemp R, Palmer N, Kielmann T, Bal H. Cuckoo: a computation offloading framework for smartphones. In: Gris M, Yang G, eds. *Mobile Computing, Applications, and Services. MobiCASE 2010. Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering* (Vol. 76). Berlin, Heidelberg: Springer; 2012:59-79.
19. Zhou B, Srirama S, Buyya R. An auction-based incentive mechanism for heterogeneous Mobile clouds. *J Syst Softw (JSS)*. 2019;152: 151-164.

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License

20. Bowen Zhou, Amir Vahid Dastjerdi, Rodrigo N. Calheiros, Satish Narayana Srirama, and Rajkumar Buyya,mCloud: a context-aware offloading framework for heterogeneous Mobile cloud *IEEE Trans Serv Comput (TSC)*, Volume 10, Number 5, Pages: 797–810, IEEE Computer Society Press, USA, 2017.
21. Satyanarayanan M, Kozuch MA, Helfrich CJ, OHallaron DR. Towards seamless mobility on pervasive hardware. *Pervasive Mob Comput*. 2005;1(2):157-189.
22. Ma RK, Lam KT, Wang CL. eXCloud: transparent runtime support for scaling mobile applications in cloud. Paper presented at: Proceedings of the IEEE International Conference on Cloud and Service Computing (CSC); 2011:103-110.
23. Kosta S, Aucinas A, Hui P, Mortier R, Zhang X. Unleashing the power of Mobile cloud computing using ThinkAir; 2011. arXiv preprint arXiv:1105.3232.
24. Van Nieuwpoort R, Maassen J, Wrzesiska G, et al. Ibis:aflexibleandefficientjavabasedgridprogramming environment. *Concurr Comput Pract Exp*. 2005;17:1079-1107.
25. Othman M, Madani SA, Khan SU. A survey of mobile cloud computing application models. *IEEE Commun Surv Tutor*. 2013;16(1):393-413.
26. Roy DG, De D, Mukherjee A, Buyya R. Application-aware cloudlet selection for computation offloading in multi-cloudlet environment. *J Supercomput*. 2017 Apr 1;73(4):1672-1690.
27. Mukherjee A, De D, Roy DG. A power and latency aware cloudlet selection strategy for multi-cloudlet environment. *IEEE Trans Cloud Comput*. 2016;7(1):141-154.
28. PowerTutor. http://ziyang.eecs.umich.edu/projects/powertutor/. Accessed September 05, 2019.
29. Ristenpart T, Tromer E, Shacham H, Savage S. Hey, you, get off of my cloud: exploring information leakage in third-party compute clouds. Paper presented at: Proceedings of the 16th ACM Conference on Computer and Communications Security; 2009:199-212; Chicago, IL.
30. Standaert FX. Introduction to side-channel attacks. *Secure Integrated Circuits and Systems*. New York, NY: Springer; 2010:27-42.

How to cite this article: Malik SUR, Akram H, Gill SS, Pervaiz H, Malik H. EFFORT: Energy efficient framework for offload communication in mobile cloud computing. *Softw: Pract Exper*. 2021;51:1896–1909. https://doi.org/10.1002/spe.2850

1097024x, 2021, 9, Downloaded from https://onlinelibrary.wiley.com/doi/10.1002/spe.2850 by UNIVERSITY OF ESSEX ALBERT SLOMAN LIBRARY, Wiley Online Library on [25/11/2026]. See the Terms and Conditions (https://onlinelibrary.wiley.com/terms-and-conditions) on Wiley Online Library for rules of use; OA articles are governed by the applicable Creative Commons License