DOI 10.12694/scpe.v19i4.1411
Scalable Computing: Practice and Experience ISSN 1895-1767
Volume 19, Number 4, pp. 309–337. http://www.scpe.org

A SURVEY ON MOBILE CLOUD COMPUTING: MOBILE COMPUTING + CLOUD COMPUTING (MCC = MC + CC)

RAMASUBBAREDDY SOMULA* AND SASIKALA R†

Abstract. In recent years, the mobile devices become popular for communication and running advanced real time applications such as face reorganization and online games. Although, mobile devices advanced for providing significant benefits for mobile users. But still, these devices suffers with limited recourses such as computation power, battery and storage space due to the portable size. However, The Cloud Technology overcome the limitations of mobile computing with better performance and recourses. The cloud technology provides enough computing recourses to run mobile applications as storage computing power on cloud platform. Therefore, the novel technology called mobile cloud computing (MCC) is introduced by integrating two technologies (Mobile Computing, Cloud Computing) in order to overcome the limitations(such as Battery life, Storage capacity, Processing capacity) of Mobile Devices by offloading application to recourse rich Remote server. This paper presents an overview of MCC, the advantages of MCC, the related concepts and the technology beyond various offloading frameworks, the architecture of the MCC, Cloudlet technology, security and privacy issues and limitations of mobile cloud computing. Finally, we conclude with feature research directions in MCC.

Key words: Mobile Computing, Cloud Computing, Mobile Cloud Computing, Cloudlet Selection, Computation offloading, Edge Computing, security.

AMS subject classifications. 15A15, 15A09, 15A23

1. Introduction. Over the past few decades, mobile devices have been playing an important role in our modern and virtual lifestyle. For instance, according to the survey by International Data Corporation (IDC) in 2016, the usage of mobile devices and tablets was increased by 1.6 billion units exponentially [1]. In recent years, mobile applications became popular in various categories such as news, entertainment, health, business and social networks. The mobile computing allows users to access all necessary applications from application centres such as Android play store and Apple iTunes etc., irrespective of location. Even mobile cloud computing provides high-end features for running various real time applications but still users demand for more computing resources. For mobile computing, the mobile devices are designed with limited battery life, storage capacity, processing capacity and communication capabilities. Mobility is important feature on pervasive computing environment where the user is able to perform his work without any interruption. The cloud computing is emerging technology which is formed with amalgamation of various technologies such as virtualization, distributed computing, SOA, web services etc. Cloud computing provides massive computing resources (such as hardware, software, storage) in order to improve the performance of application as well as reducing processing cost. It allows users to access data from any location on demand basis. The mobile device will perform high computational tasks on cloud platform which require more computing resources. The cloud computing paradigm can be represented through three different service models. Platform as a service (paas), Infrastructure as a service (Iaas), software as a service (saas) as shown in Fig 1.1The author in [2], presented the annual growth rate of cloud service models, Iaas is 41%, paas holds 26.6% and Saas holds 17.4%. The emerging technology, Mobile Cloud Computing has been introduced to overcome limitations of mobile devices. Recently, the mobile users demand for computing is being increased due to the development in mobile computing technology. Various studies define the importance and benefit of mobile cloud computing for mobile users and enterprisers. For example, according to the ABI, the usage of mobile devices reached to 280 million by 2015, the revenue of Mobile cloud computing reached to $ 5.2 billion [3]. Currently the growth of advanced mobile devices developed rapidly with sufficient resources such as battery life, storage, processing power. Nonetheless, it is still suffering from processing real time application such as image recognition,video streaming, language translation. Mobile devices are less compared to server systems and desktop computers in terms of computing power and storage. When mobile device runs resource intensive task put heavy load on processor and reduce battery life.

*School of Computer Science and Engineering (SCOPE), VIT University, Vellore, India (svramasubbareddy1219@gmail.com).
†School of Computer Science and Engineering (SCOPE), VIT University, Vellore,India.(sasikala.ra@vit.ac.in)

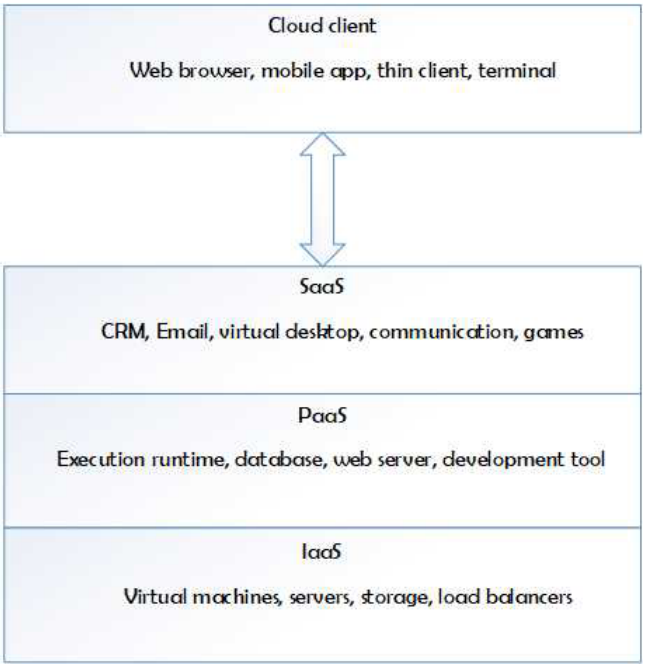

FIG. 1.1. *Cloud Computing Services*

Nowadays, the research work on cloud computing is aiming to enhance computing capabilities of mobile devices by allowing Mobile users to access various service based models such as software, infrastructure and computing services. Amazon is one of the cloud service provider which provides security to user personal data by various storage service models (S3) [4]. MCC promises to improve performance of the mobile application beyond mobile computing, with the help of cloud computing [5] [6]. Most of the data generated by the mobile device will be video content which is over 78% by 2021 forecast by cisco [7]. The concept of offloading fully or part of the application into remote cloud environment to address limitations of mobile computing through service providers other than the mobile can deploy application on cloud where both storage and computation can happen out of mobile device is known as Mobile Cloud.

In this paper, we aim to discuss Various categories of research areas in Mobile Cloud Computing such as computation offloading, cloudlet selection (or) edge computing, resource provisioning, security, privacy issues and VM migration techniques. Furthermore, we plan to discuss proposed research works and also upcoming novel solution for addressing Mobile Cloud Computing issues.

2. Mobile Cloud Computing Overview.

2.1. Definition of Mobile Cloud Computing. MCC is an emerging technology where it fill gap between limited resources of Mobile devices as well as resource intensive applications required to run a resource rich environment computing. According to the MCC forum definition: the execution of mobile application will happen outside of mobile device. The computation power and storage of mobile application more to cloud environment for processing. The MCC allows mobile users to access computing services, it is not restricted to particular mobile users [3]. In the second definition [8] [9]. The mobile cloud computing provides computing resources for mobile devices remotely. In the third destination [10] [11], the cloud server does not need to act as powerful server. But enhancing mobile devices configuration setup in terms of storage and processing capacity.

2.2. Related concepts and technology.

2.2.1. Mobile Computing. Nowadays, Mobile Devices (MD) became an essential equipment for communication in everyone life. Even though mobile devices are able to support real time resource hungry applications but still they are limited in terms of storage, processor and power consumption. In order to address this issues,

the emerging cloud computing technology provides enough resources to optimize performance of the mobile applications. Generally, mobile computing is a process of executing applications in mobile device and transferring result to one (or) more devices. Mobile communication is able to make use of centrally located application (or) data with help of small (or) little portable computing devices. This technology make every application to be executed in single devices. The usage of mobile devices is increasing day by day, the requirement is to provide better services for low cost and power consumption also increases. The list of issues in mobile computing is represented in following Fig 2.1.

2.2.2. Mobile Network Architecture. The classification of mobile network can be represented as following Fig 2.2.

2.2.3. Cellular Architecture. Earlier, the mobile networks were intended to cover huge geographical area by using single transmitter with high power consumption. Even conventional architecture covers huge geographical area, but it does not support frequency reuse technology.

In order to facilitate frequency reuse as well as large coverage, the cellular architecture was brought into mobile networks. This cellular architecture replaces high power consumption transmitter with low power transmitters. The large geographical area splits into number of hexagonal cells which are served by base station.

Each cell in cellular network is surrounded by number of independent cell. Each adjacent cell boundary touches each other. The hexagonal cell covers certain area in geographical location. Each cell is served by nearby base station. The base station which serves each cell is allotted with certain portion of frequency. The base station of adjacent cell is allotted with different frequency ranges to overcome interruptions in communication.

The following formula depicts the frequency reuse distance:

$$d = r\sqrt{(3 * n)} \tag{2.1}$$

Here, r represents distance between cell center and cell boundary
n represents adjacent cells around concerned cell

2.2.4. Mobile Ad Hoc Network Architecture (MANET). In MANET's network, nodes, routers and switches position are not fixed. It consists of mobile devices which communicate with each other through wireless network. In such network, the node is able to service and send or receive response from nearby neighboring node. The positions of nodes in MANET can organize the network.

The following Fig 2.3 illustrates the behavior of the MANET architecture. It consists of 5 nodes, two are mobile nodes (mobile node 1& mobile node 2). One is to handle pc and the other is sensor node. The base station acts as a router which routes messages to all involved nodes to MANET network.

Each node in MANET behaves like router to communicate with other neighboring nodes. It is also known as self-organized network [12].

2.2.5. Mobile Wireless Sensor Network Architecture (MWSN). MWSN is similar to the MANET network, except sensor nodes involvement. In MWSN, the sensor nodes having computing and communication abilities [12].

In MWSN, the sensor nodes acts as routes to pass messages to neighbour nodes as well as to communicate with other networks such as MANET, cellular network Fig 2.4 depicts the MWSN architecture.

The main advantage of MWSN over static sensor network is the expansion of no. of applications. It is used in many real time applications such as health care to monitor blood pressure and heart rate [12].

2.3. Cloud Computing. Cloud Computing (CC) is an advanced technology that provides computing resources for Information Technology (IT) to increase capability and capacity over network [6]. Cloud is a collection of virtualized computers which provides resources dynamically on basis of pay as you go (or) pay-per use model [12]. Cloud computing allows users to access application (or) data anytime from anywhere on demand basis [6]. It is mainly focusing on development of advanced applications, computing models and using existing services for developing new software [13]. Cloud computing technology is composed of various technologies such as grid computing, SOA, virtualization, web services.

Various applications with client as a model. We can say Amazon web services (AWS) and Microsoft Azure cloud as example of public cloud. Azure cloud open and provide services to build, deploy and run applications

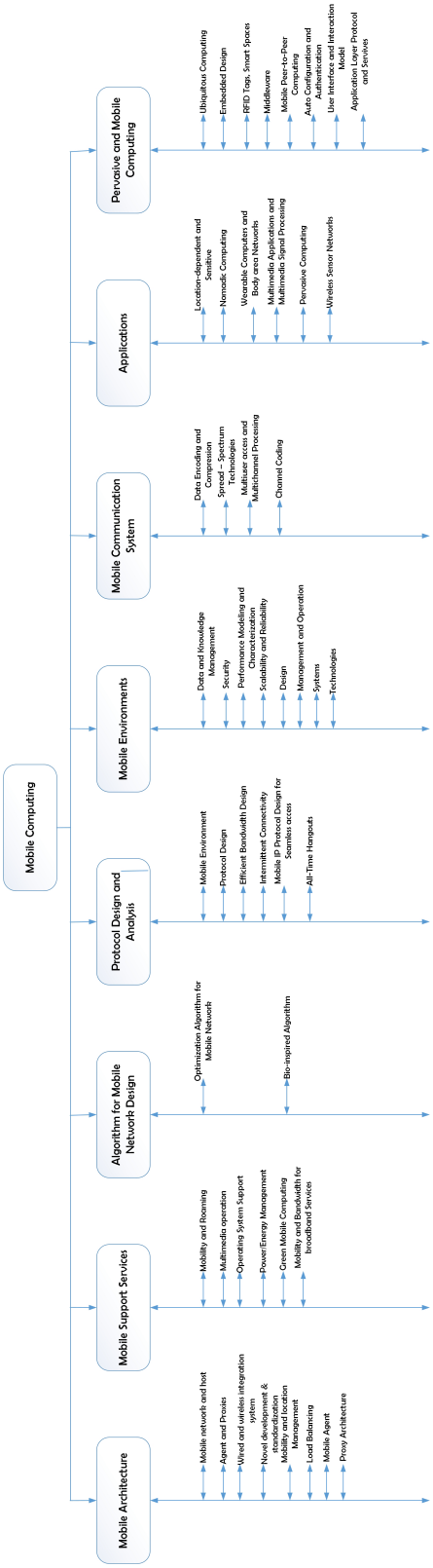

FIG. 2.1. *Mobile Computing Challenges*

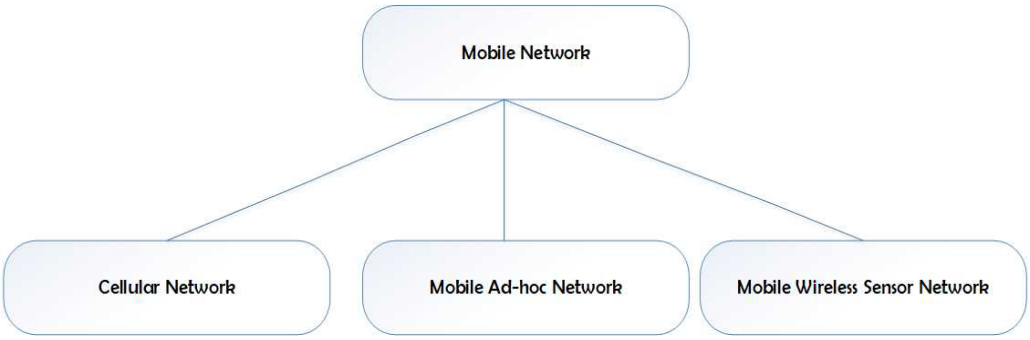

FIG. 2.2. *Classification of Mobile Network*

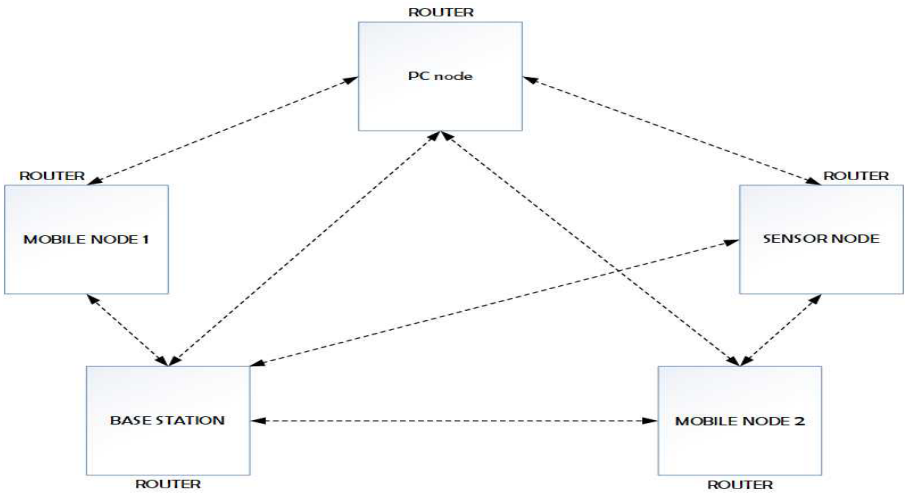

FIG. 2.3. *Architecture of MANET*

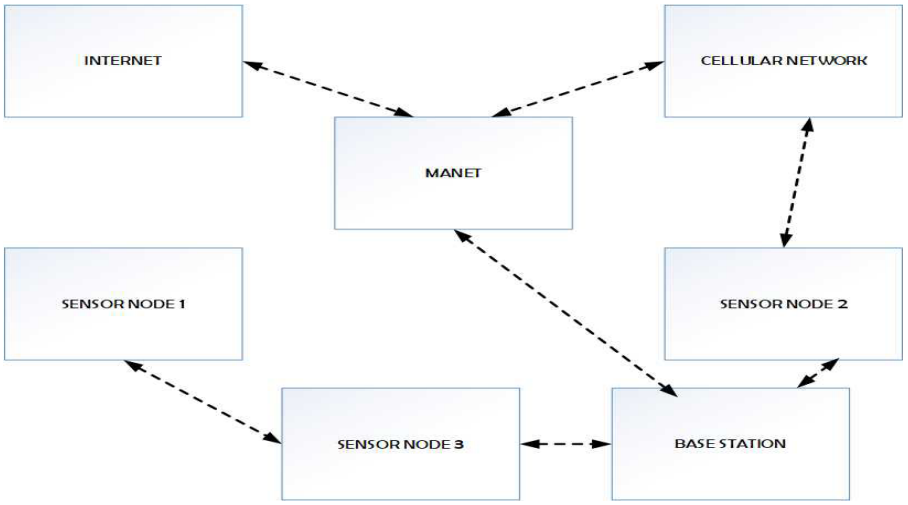

FIG. 2.4. *Wireless Sensor Network Architecture*

on servicers [4]. AWS cloud provides services via two models, Infrastructure as a Service (Iaas) and software as a service (saas). The client can directly access applications without installing any software's in local system [14].

2.3.1. Characteristics of cloud computing.

On demand self-service. Whenever user client require services such as virtual machine for processing and storage is being leveraged without any interaction between users and service providers.

Broad Network Access. Client can access services at anytime from anywhere through powerful devices such as laptops, smart phones and tablets.

Resource Polling. Multiple users can access computing resources (processing power, storage, bandwidth, memory) in multi-tenancy model. The user have no information about from where the services are provided by service provider.

Rapid Elasticity. Based on subscriber demand, the resources are rapidly increased Automatically. The user thinks that cloud resources are limited and scalable at any time.

Measured Services. The service provider offer resources on pay-per-use Manner. the transparency is to be maintained between users and service providers.

2.4. Cloud Computing Deployment Models. Cloud models can be used to deploy cloud services. The deployment models are classified into four types [15] [14] [16].

1. Private cloud;
2. Public cloud;
3. Hybrid cloud;
4. Community cloud.

Private Cloud. In this model, data center is owned by particular organization and managed by either organization or third party. Private cloud is restricted to particular users [17].

Public cloud. Public cloud is not restricted to particular user. It can be used by all kinds of cloud users such as Research, Industry and Company [17].

Hybrid cloud. In this model, one or more deployment models are integrated to design single data centre. This model can be used to overcome issues arises by private cloud during accessing [14] [18] [19] [16] [17].

Community cloud. In this model, the data centre is owned by one or more organizations and managed either by third party or only one of the community organization.

2.5. Architecture of Mobile cloud computing. Nowadays mobile devices became part of our daily life style and can be connected to any cloud server at anytime from anywhere via wireless infrastructure. Cloud computing introduced concept: Bring Your Own Device(BYOD), which allows employees to leverage privileged organization content and applications deployed in cloud server. The virtualization technology in cloud computing enables multiple VMs (Virtual Machines) or operating systems to run on smart phone devices including tablets, smart devices and laptops. That is cloud computing provides services in multi-tenant manner to subscribers via mobile virtualization. The cloud computing offers task oriented services with virtualization on mobile devices to provide unlimited computing power and storage on demand basis. Cloud computing build MCC applications which are enhanced in terms of computing power and storage comparing with traditional mobile computing applications.

Limited Battery Life. The battery capacity of mobile device is limited to run high-end application. It is not possible to depend on other external power sources while moving (mobility). The charge of battery will be lost in few hours.

Limited storage capacity. Every smart phone or mobile device is configured with 8 GB and laptop is configured with 500 GB. It can be expanded with external memory. It cannot support more than configured storage, when back is required.

Limited processing capacity. The smart phone having ARM processor, it can only run small and very few applications. In case of laptops with various processor (i3, i5 and i7) are available but not affordable due to high cost. The processor in mobile device cannot be upgraded if anyone want to upgrade.

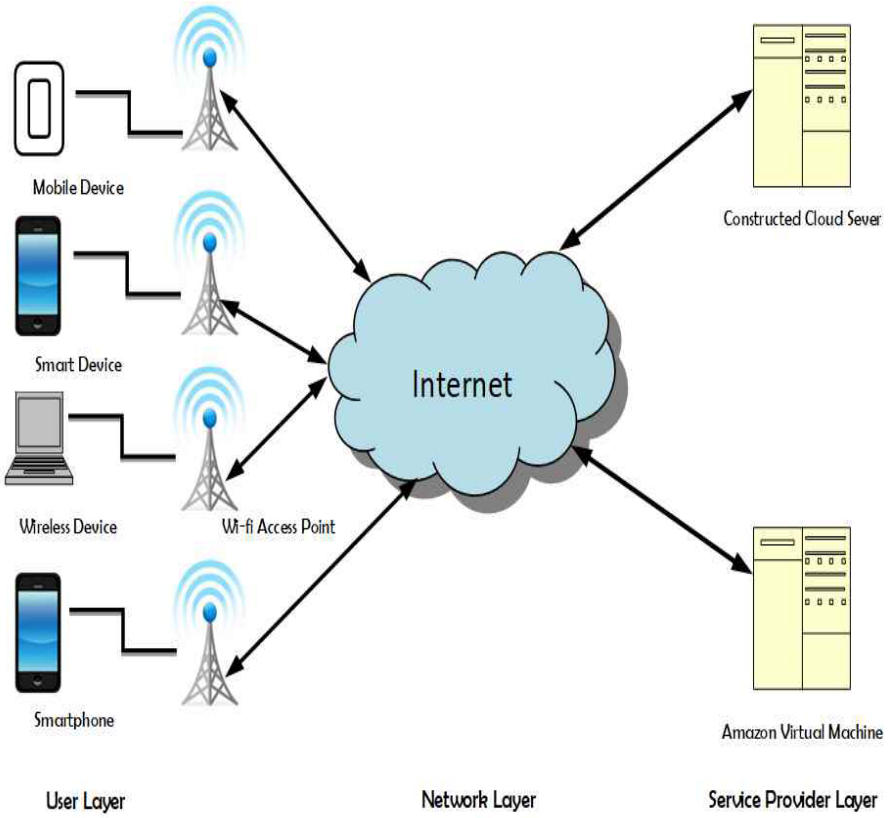

FIG. 2.5. *Architecture of MCC*

Low Bandwidth. The conventional technologies such as EDGE, GPRS and GSM provide low bandwidth. The advanced technologies 3G and 4G provides high bandwidth, but they are available only in developed cities/towns. Fig 2.5 depicts the architecture of MCC which is categorized into three different layers:

1. *User layer;*
2. *Network layer;*
3. *Service provider layer.*

2.6. Benefits of Mobile Cloud Computing.

2.6.1. Extended battery lifetime. Battery consumption became a serious issue in mobile computing. There have been many proposed models in order to address battery issue. But they are focused on hardware design. MCC provided solution for preserving battery life by offloading resource intensive applications onto cloud and then the entire process will be done at cloud side after that result is sent back to mobile device.

2.6.2. Data Storage. According to user perspective, MCC provides unlimited storage on demand basis. Cloud storage permits users to store and access data from anywhere at any time. The data stored in cloud could be any multimedia data. The cloud storage stores data in an encryption format if any change causes to mobile, the data will remain safe in cloud as a backup.

2.6.3. Increasing processing power. The computing power could be saved in mobile device by processing applications on cloud and result depicted in device. The mobile user does not feel of having limited processing power because cloud provides unlimited resources. the MCC allows users run complex resource intensive applications without any resource restrictions.

2.6.4. Dynamic provisioning. Mobile users can access required resources on demand basis, dynamic provisioning that permits users to have access resources without any advanced reservation by creating virtual machines (VMs) with appropriate configuration. Whenever user occurs cloud services, the no. of CPU cores and storage dynamically increased based on requirement. The self service provisioning is more beneficial compared to hardware configuration enhancement.

2.6.5. Scalability. Scalability is one of the significant characteristics of cloud computing. The resources allocated to user will be increased or decreased based on user requirement. The cloud service provider ensure to manage resource requirement of mobile application.

2.6.6. Reliability. Cloud is always reliable compared to mobile device. Cloud renders provide security application such as virus scanning and malicious code detection being executed in cloud. In order to save user from installing in local systems, MCC provides various authentication mechanisms for preventing unauthorized user from access cloud resources or confidential data.

2.6.7. Ease of integration. In mobile computing environment, the user cannot access resources or services. In MCC, the user can access all kind of services due to integration of various services into cloud. The emerging advanced technologies such as Big Data and IOT can be easily integrated with MCC technology to enhance the Quality of Services (QOS).

3. Offloading Approach. The concept of offloading can be done by offloading resource intensive application partly or fully from mobile device (MD) to cloud. Offloading classified into two ways namely code offloading and state offloading. Code offloading is achieved through sending part of the application to remote cloud for executing. On other hand, state offloading means transferring entire application to remote cloud. The process of offloading can be achieved by following three steps:
1. Partitioning
2. Preparation
3. Offloading decision

Partitioning. Partition of an application is an initial step in which the entire application is divided into various components. These components are affordable and non-affordable which means the components run on local device or run at remote cloud server. Based on different information the component can be considered either affordable or non-affordable. While designing application the programmer annotate local or remote execution through an API as affordable. The intensive part of application can be identified through code analysis and performance prediction (application profiling). It is not efficient approach partitioning application at designing time, because both techniques are not considering real-time execution context. So that the accuracy is very less.

Preparation. In this step, the actions which are required for execution of mobile application at remote server. This action may be selection of server, installation of code and execution on account of mobile device. Both data and code needed for remote execution.

Offloading decision. This is final step in offloading, before offloading component onto remote server. When mobile device uses offloading component then it is not necessarily to depend on execution. The decision is based on run time, then the real time information available such as battery consumption for sending data to remote server, wireless connection strength. Comparatively the runtime includes more overhead than decision during design time.

3.1. Types of Frameworks. The offloading frameworks can be classified into two categories. The first category is static offloading frameworks. In which all discussed steps in above section can be achieved at design time on other hand, the dynamic offloading framework can be achieved at runtime. It means the decision is taken at runtime whether to offload or not.

3.2. Offloading Mechanism. There have been various proposed works on offloading mechanism for offloading resource intensive application into cloud. This can be classified into two offloading mechanisms:
1. VMs offloading
2. Code offloading

In code offloading, the computational intensive component is sent to remote server by invoking Remote Procedure Call (RPC) with the help of notations, compilers and binary code modifiers where as VMs offloading can be achieved capturing mobile state and storing into cloud. During offloading the execution is stopped at mobile device and VM clone sends to cloud.

3.3. Comparison among various offloading mechanisms in MCC. The existing popular offloading mechanisms would be discussed in this section. Each mechanism properties and offloading process concludes at end of the section [20].

3.3.1. Clonecloud Framework. The motivation behind clone cloud [21] is to reducing power consumption on mobile device by offloading computation intensive application into remote server. In clone cloud, the application partitioning can be achieved by integrating program profile with program partitioning in order to obtain constrains, for example the component which depends on local mobile device integrates parts like sensor , camera and speakers can be executed locally. The clone cloud mechanism will use threads functionalities at application portioning. The programming analysis aim to obtain possible migration points, in other hand the profile is aimed to produce cost of migrating and processing at server.

In preparation step, the application of the mobile device is captured and stored in cloud server.

In decision step, the decision is taken place at runtime which means all running threads on mobile device are suspended and transferred to cloud server then all threads resumed in clone cloud to offload computation. The execution process in clone cloud is to create duplicate mobile software in cloud server. Then computation is offloaded to server and result s back later once the execution process is done. The distributed mechanism in clone cloud aims to implement partitioning process for given application in application layer virtual machine (VM).

The author Chan et al [21], tested clone cloud with different applications such as virus scanner, image search and privacy preserving applications in various scenarios for example, clone cloud with Wi-Fi and clone cloud with 3G environment.

3.3.2. MAUI Framework. The MAUI [21]framework focusing on energy optimization by executing complex components at server in cloud. The execution of components in MAUI is done dynamically because continuous profiling process. The MAUI tries to hide the difficulties of execution at remote server from mobile user in order to make an impression that the entire application execution is done at local device. The developer of application can decide the annotations which component is to execute in mobile device or which is to execute at remote server.

In order to achieve MAUI partitioning framework the following conditions must be installed in both mobile device and remote server side. One is application binaries and other one is proxies, profilers and solvers.

The profile maintain information about network conditions which is helpful for MAUI to take appropriate decision otherwise leads to wrong decision. The profiler keep on updating the information during whole execution of the application.

In MAUI, the profiler will collect information and gives it to the MAUI solver which can make decision at runtime whether to offload or run it locally. Author has conducted various experiments using three various applications such as face recognition, chess and video. In first comparison, the author has composed energy consumption of application on stand-alone mobile device and with MAUI framework. The energy consumption is optimized with MAUI framework by offload nearby server. On other hand, the energy consumption is reduced by executing chess and video game respectively 45% and 25% with MAUI framework by offloading to nearby server.

3.3.3. Cloudlet Framework. Offloading mechanism is not always optimum solution because of network failure and long processing delay. The cloudlet is a cloud in box, which is situated nearby mobile device. We can say that cloudlet brings cloud closer to mobile devices.

The cloudlet reduces response time in milliseconds by executing application in nearby cloudlet that is comparatively better than executing on remote cloud server. Satyanarayan et al in [22], introduced VM based cloudlet framework in which, cloudlet for hosting offloaded task that is run on remote server for storage and processing purpose the cloudlet is not as same as cloud and any other parallel system. The cloudlet based VM supports scalability, mobility and elasticity.

In preparation step, the cloudlet framework require mobile device application processing environment at remote server then offload complete application to remote server through VM which is based on dynamic VM synthesis. The mobile device act as interface and the entire application execution can be achieved at cloudlet infrastructure. The user mobility is a primary challenge while processing application in cloudlet.

The cloudlets are distributed in geographical area, the users can easily access storage resources and computing cycles via internet infrastructure. In order to avoid long delays, generally cloudlets located at population areas such as bus stops, coffee shops and colleges. Users can access distant cloud via cloudlet if the user must offload resource intensive application, then the application has to discover and send application to cloudlet [22], otherwise the application can select optimal cloudlet based on network status.

3.3.4. Jade Framework. Jade framework [23] is similar to other framework, but different perspective. In jade, the system consider both application and device status to make appropriate decision where the application must be executed. This framework aims to reduce energy consumption for mobile devices as well as minimize burden on application developer.

The application is partitioned into various classes based on available information. In partition step, the system verifies both application and device status through other information such as load variation, energy level communication cost. The jade designed with enough number of APIS, this minimize the burden on developer to control the partition of application and remote server interact with local code.

In jade, the offloading decision is taken at run time whether to execute locally or remote server. Jade can be performed on two different servers.

1. Android server
2. Non-android server

The non-android server must maintain installation of java platform. Jade runs as normal java program at non-android server. The decision of offloading can be changed based on device status in result, energy consumption is reduced. Jade can easily transform computational task from mobile device to available remote server to optimize energy consumption.

The author conducted experiment using face recognition application. The experiment was done on 50 pictures each with 200 kb size. In result, jade has outperformed on existing frameworks by reducing 34% average power consumption.

3.3.5. Mirror Server Framework. This mirror framework [24] use Telecommunication Service Provider (TSP) at remote server. TSP would provide services to landline mobile users. The mirror server can advance the mobile devices by providing required resources storage, computation offloading and security on computation infrastructure. The mirror server can maintain VM instances for various mobile devices. In mirror framework, the entire application is offloaded to remote server so the partition of application not necessary.

In preparation step, the new virtual machines (VMs) created, this VMs are managed and deployed by Mirror server. The application execution is done at Mirror VM instance under control of mirror server. Mirror server optimize offload mechanism.

The mirror server is not specially designed for data analysis and provide limited services (i.e. file sharing and file scanning) are included. The author conducted experiments by installation of file scanner at mirror server. The applications are trying to access mirror. The energy is reduced considerably, execution time also increased running scanner on mirrors.

3.3.6. Cuckoo Framework. The author in [25], has introduced new framework called as cuckoo, which offload resource-intensive code to remote server for mobile device. In this model, offloading can be achieved through java stub model. Cuckoo was designed to advances the performance and reduce battery utilization. The partition of application is adapted from existing android model, which separates affordable and non-affordable components of the application. This process represents through user interface. The affordable components are offloaded into any JVM resource. In preparation step, the application developer is required to write code two times, one for local execution and other is for remote execution. For this the require programming model which is useful when connection is dropped support execution. This both codes combined to form single package. Cuckoo framework is dynamic offloading model and offload only well-identified components of application. If remote resource is not available for offloading task then execution will takes place in local device.

3.3.7. Phone2cloud Framework. In phone2cloud [26], the author has focused on energy efficiency and application performance by conducting quantitative experiments on various scenarios. This framework is not fully automatic offloading framework, if application needs to be executed over cloud then it requires to modify manually at preparation stage. The delay-tolerance threshold and static analysis are required to make offloading decision. This threshold can be formed based on prediction of time taken for transferring data to remote cloud via Wi-Fi network. This framework waits until Wi-Fi available rather than sending data directly.

The phone2cloud framework aimed to reduce power consumption and execution time while offloading. It was implemented on Android and Hadoop environment to analysis experiment results. There are various components involved in phone 2cloud framework, which would be helpful to make appropriate offloading decision to run either locally or remotely. Components like bandwidth monitor, resource monitor, offload decision manager, remote execution manager, offloading predictor, local execution manager and offloading proxy.

The framework consider two parameters before offloading takes place. First one is execution time of application on mobile device and delay tolerance threshold. Second one is power consumption of application to run on both mobile device as well as cloud environment. If the user waiting time is (delay tolerance threshold) is less than average execution time on mobile device then application offload to cloud. The power consumption is also considered as another constrain before application is offloaded to cloud. If power consumption is less for executing in mobile device, then application run locally.

The author examined phone2cloud by conducting experiments with various applications such as word count, path finder and sort application. In result, framework reduced energy consumption, improves preference of application and experience of the user.

3.3.8. Thinkair Framework. ThinkAir framework aims to address issues raised by existing frameworks such as MAUI does not address issue of scalability while executing application over cloud. The clonecloud framework tries to extend binary pieces of process to make overall execution faster on cloud.

However, this approach will not support if any drastically changes happen to input or execution environment. In order to address these two challenges such as elasticity or scalability and power consumption in mobile device, the author in [27] introduced thinkair framework.

In preparation step, the methods are annotated as remote which are received to offload cloud server. Thinkair approach provide simple programmer API to reduce burden on application developers. The execution component can detect whether method is affordable or not and handle all other necessary tasks such as decision making and communication with remote server without developer involvement.

The execution encounter makes decision for the first method based on environment parameters such as Wi-Fi signal, available sources. For example if the Wi-Fi signal is good then the method offloads to remote cloud otherwise executes on local mobile device. In profiling step, thinkair aims to predict make more accurate decision with the help of variant profilers such as hardware profiler, software profiler and network profiler. These three profilers together feed to estimate power consumption more accurately.

Hardware profiler.

- The hardware profiler collects information related to hardware such as CPU, screen, 3G and Wi-Fi interface.
- The CPU utilization is measured from 1 to 100 in different frequencies.
- Screen can be measured through brightness from 0 to 255.
- The power consumption of Wi-Fi interface either low or high.
- The 3G interface is either idle on shared with other channel.

Software profiler. The software profile collects information related to execution of program. This profile record following information:

- The total time required for executing method.
- The total CPU time of the method.
- No. of statements exist in method.
- No. of times the method invokes.
- The size of the thread.

Network profiler. Network profile involves overhead because it considers other profiles and parameters as well. In previous model, we used to consider only RTT on network. Now, thinkair brings other parameters such

as number of packets sent/received per second, other parameters related to 3G/Wi-Fi interface, for example uplink and downlink rate for transferring and receiving data. These all measurements feeds to achieve better estimation in offloading method.

In thinkair, the partition is done manually by providing programmer API. The offloading decision can be made by considering variant profiles data. The author evaluated thinkair framework using four different applications such as face detection, N-queens Problem, virus scanning application and image merging application. The thinkair outperform in each experiment and reduced energy consumption and improved application performance using accurate prediction model.

3.4. Comparison Table among Different Offloading Frameworks. Table 5.1 presents our comparison between different offloading frameworks.

4. Cloudlet: Bringing cloud closer [28]. Nowadays mobile devices gaining popularity for computation and storage capabilities. The applications on mobile devices require more resources to process, but mobile devices due to lack of resources unable to provide required resources for resource-intensive applications. In Mobile Cloud Computing (MCC), computing offloading mechanism address resource hungry application by executing partly or entire application on remote cloud server. The offloading approach also faces challenge such as low bandwidth and high latencies. The computation offloading approach is not appropriate for real-time application such as face recognition, navigation and online video games. When network connectivity is poor then performance of application is affected. In order to address this problem, the cloudlet concept has been proposed by satyanarayan [22] [29] [30]. Cloudlet aims to bring cloud closer to mobile users [31]. Cloudlet is a sort of mini cloud which is formed by connecting various nearby mobile devices via Wi-Fi or Bluetooth. The mobile devices which are involved in cloudlet termed as cloudlet nodes. The cloudlet nodes could be laptops, mobile devices, PDAs, tablets and palmtops. Cloudlet allows nearby mobile users to leverage available computational resources via Wi-Fi network. Therefore, the execution time of application is reduced to milliseconds comparatively less execution on remote cloud server. The cloudlet is dynamic in nature, it can move and join at any point of time from network [32].

Nearby mobile users leverage cloudlet resources by running all resource-rich application and reduces end-to-end response time [33]. Cloudlet can act as static cloudlet and dynamic cloudlet. The static cloud is termed as cooperative cloudlet because established by cooperative organization. Besides, cloudlet can be formed with nearby mobile devices such as device connected each other in railway station. Cloudlet is a novel emerging technology for latency-sensitive application and computation intensive application to improve application performance and user experience with application [34] [35]. Fig 4.1 represents basic process of cloudlet concept.

4.1. Cloudlet characteristic. The purpose of mobile cloud is discussed in above section that brings cloud resources close to mobile users. The functioning of cloudlet can be represented through following four characteristics briefly.

Soft-state. Generally, soft state is represented for efficiency in computer science, which can be replaced at any point of time. Soft-state is self-managing. It is completely different from hard-stated and holds catch state for cloud. Soft state store all mobile users data in buffer for security concerns before transmitting to remote cloud. Soft-state implementation is much more efficient in network environment when compared with hard-state.

Close at hand. Cloudlet is available very close to mobile users in order to provide high bandwidth and low latency in network.

Well Connected. Mobile Cloud Computing enhance battery power utilization by providing sufficient computational resources to processor offloaded resource rich mobile application over cloud.

Cloud Standards. Cloudlet functions as similar as remote cloud. The only difference is in bringing cloud resources to mobile user for reducing battery consumption and high latency issues. The offloaded task is executed on VMs running in cloud infrastructure.

4.2. Classification of cloudlet. Cloudlet can be classified into two types:

1. Ad-hoc cloudlet;
2. Elastic cloudlet.

TABLE 3.1
Comparison Table among Different Offloading Frameworks

Framework	**Partitioning**	**Preparation**	**Offloading Decision**	**Objective**	**Partly/ Entire App**	**Mechanization**
VM Cloudlet	The Entire Application is offloaded in the form of image	Mobile device processing Environment is required at remote server	NO Decision	Cloudlet-Based Mobile Cloud Computing	Entire Application	Not Available
Phone2Cloud	Partly /Entire Application Can Happen	The Developer has to annotate application to Execute in cloud.	Static	Reducing Power Consumption Improving Performance of application	Part/Entire Application	Partly Mechanization
MAUI	Each Method Lable either Local or Remote	Each application is Required to create twice, one is for Mobile and another for cloud.	Dynamic	offloading can be achieved based on energy consumption.	Method Level	Manual
Mirror Server	The entire application offloaded	Mirror Required for Smart phones	Not Available	the development of application can be achieved easily and energy consumption optimized	Entire Application	Not Available
Cuckoo	Partition can be done using existing activity model in android	Remote Server require Java Environment to run application	Dynamic	Reducing Power Consumption Increasing speed of dynamic intensive operation	Method Level	Manual
CloneCloud	Partition is done using Program analysis and Profiles	The Mobile OS requires to host on remote server	Dynamic	Providing Resources on demand basis while executing application on cloud	Thread	Automatic
Jade	Class-level Partitioning is done	it consider work load, communication cost, energy status.	Dynamic	energy consumption optimized	Class Level	Automatic
ThinkAir	Method -level Partitioning is done	it focuses on hardware, software and network profiles to archive accurate offloading,	Dynamic	scalability of cloud enhances power of mobile computing. parallelizing execute application on multiple VM's concurrently, on-demand resource allocation.	Method Level	Automatic

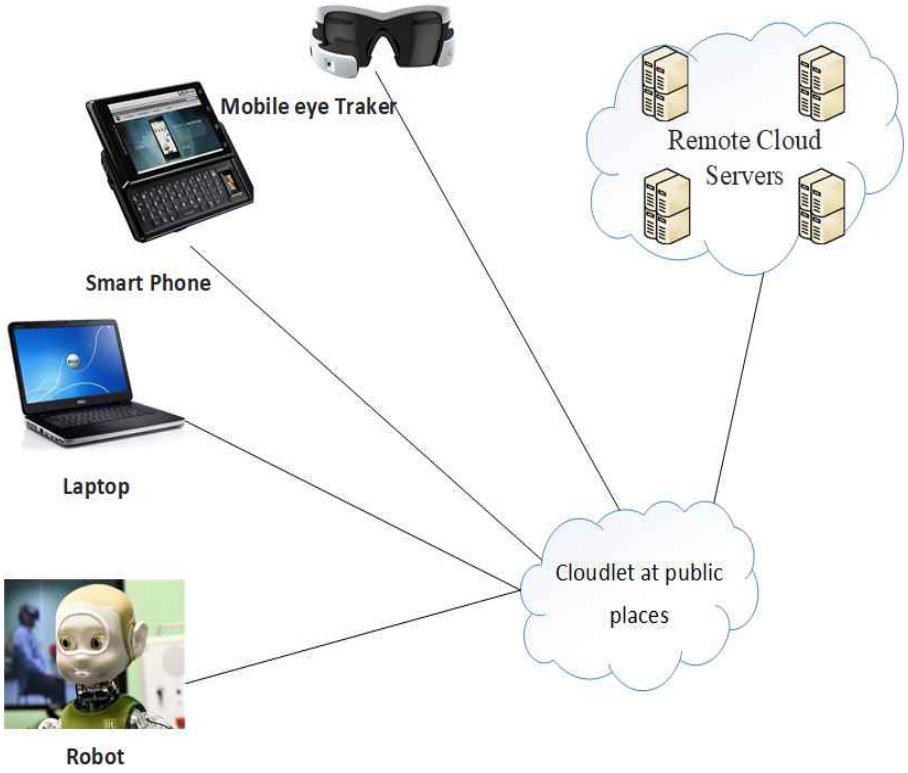

FIG. 4.1. *Basic Cloudlet View*

Ad-hoc cloudlet can be formed with accumulation of mobile node [36]. This mobile node can join and leave at any point of time. All mobile nodes help to run agent, the agent is responsible for recreating migration and deployment component whenever mobile nodes leave or join. Cloudlet helps to migrate a task from one cloudlet to another cloudlet based on cloudlet configuration and vicinity in case of elastic cloudlet, the mobile nodes are allowed to run on VMs in virtual environment. The node agent can perform dynamic spawing for mobile nodes based on available resources. The concept of elastic cloudlet is comparable with the VM-based cloudlet proposed by Sathyanarayan [22]. It solves the problem of lack of resources by offering pre-configured VM to cloudlet. Elastic cloudlet is formed through public cloud. The only one difference is that makes the both models different with extra layer, which exist in elastic cloudlet to handle mobile user applications.

4.3. Architecture of cloudlet. Cloudlet architecture Fig 4.2 formed with additional layer between mobile devices and cloud. Cloudlets are distributed in geographical area as Wi-Fi access points. The performance of the cloudlet can be calculated based on following three properties:

1. cloudlet size;
2. lifetime of cloudlet node;
3. reachable time.

Cloudlet size. The size of cloudlet is defined based on number of mobile nodes connected to that master node (initiator).

Life-time of cloudlet node. The lifetime of cloudlet node can be calculated based on how much time spent for processing task with initiator node.

Reachable time. The amount of time both mobile node and initiator (master node) in T. the cloudlet architecture can be represented with combination of both Ad-hoc cloudlet and elastic cloudlet. The architecture

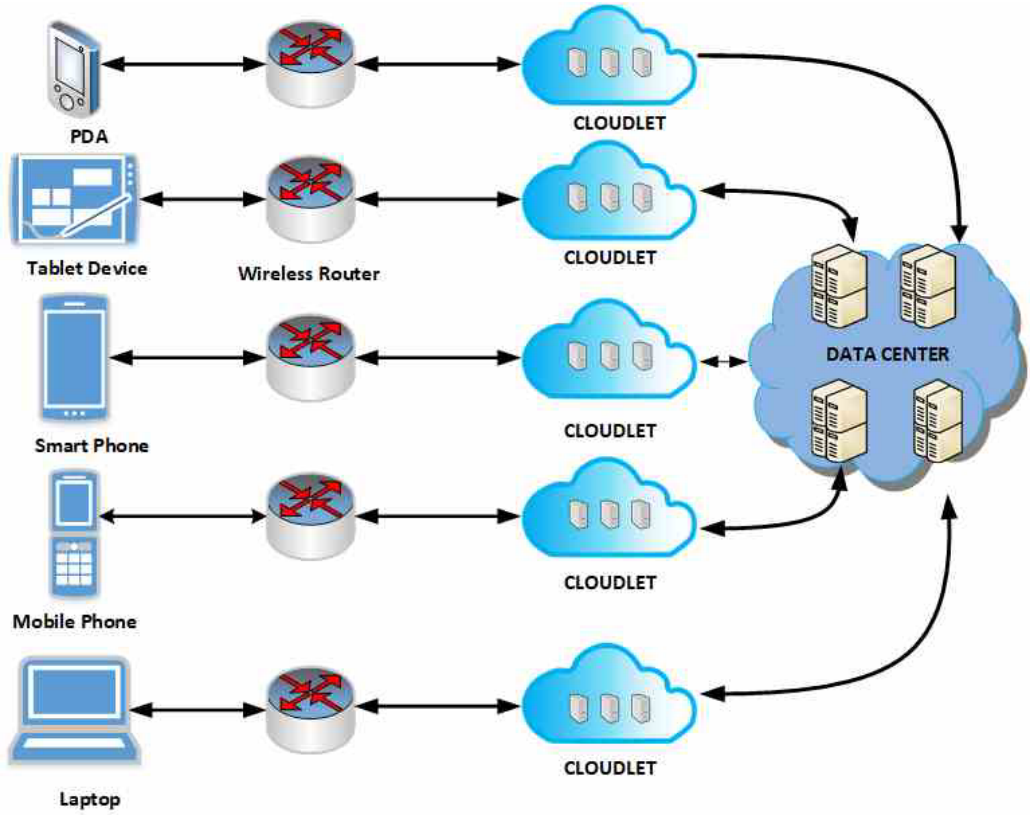

FIG. 4.2. *Architecture of cloudlet*

discuss about how mobile devices interact and communicate each other.

The architecture is categorized into three layers.

Component layer. In component layer, the number of components (mobile device) together can form deployment environment. Every component is handled by execution environment(EE) which decides whether the component is to run or stop. The components are distributed in area which can be facilitated by employing more than one EE. Component can discover issues related to performance and disclose configuration details to EE. The EE can detect performance issues and provide appropriate solutions such as resource provisioning for offloaded resource hungry tasks.

Node layer. The cloud environment can be formed with no. of servers each server is partitioned by running VMs operating system resides on each Virtual Machine. EE will hold more than one VM for execution. The node can be formed with combination of both hardware and O.S. it is Node Agent [NA] responsibility to manage and monitor all running nodes in cloudlet. NA will also take decision to start or stop any EE. The resource provisioning among all nodes in cloudlet is done by node agent.

Cloudlet layer. The no. of nodes together can be formulated as cloudlet. The cloudlet agent (CA) is responsible to manage all cloudlets and maintain communication with all node agents in cloudlet the node agent of one cloudlet can communication with other cloudlet in order to migrate resource hungry task for execution. The node can be set as cloudlet agent by considering maximum amount resources availability [37].

The following section describes briefly about categories of cloudlet. Fig 4.4 represents categories of cloudlet architecture.

4.3.1. Network Based Architecture. The mobile devices can communicate with nearby cloudlet or other devices with help of network enhance as which is being used among servers in cloud. Mobile devices always connect to near cloudlet via popular networks 3G, 4G and Wi-Fi. The cloudlet distribute computational

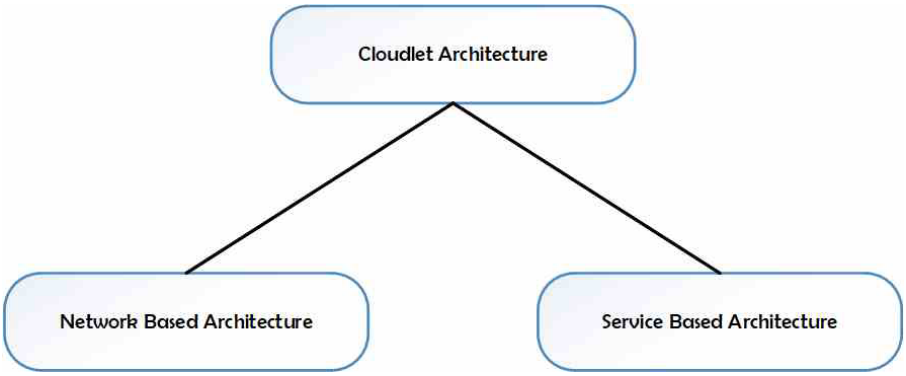

FIG. 4.3. *Classification of Cloudlet Architecture*

tasks to available cloudlets for executing and send results back to them. The data among cloudlet can be sent and received through routing algorithms.Two popular algorithms can be used to make communication among cloudlet in network i.e. distributed routing algorithm and centralized routing algorithm.

The distributed scheme uses peer-to-peer communication among cloudlets. The cloudlet distributes its present location to all nodes nearby, the node can receive and connect to the cloudlet. The mobile device maintains cloudlet table for storing ID of cloudlet whenever it receives presence of cloudlet information for future use. The cloudlet also maintain mobile table, which stores all mobile IDs that can be connected to cloudlet. The cloudlet broadcast mobile IDs table to other cloudlet to know. Each cloudlet shares mobile node information to ensure to make easy for resource allocation in future.

In centralized scheme, one server is established called as centralized server. The task of the centralized server is to store IDs of all available cloudlet. The broadcasting is done once all cloudlet gets registered with centralized server by sending cloudlet ID whenever the mobile node connects to cloudlet, the cloudlet has to store ID of cloudlet and all attached mobile nodes of cloudlet. It is centralized server task to maintain huge table for storing cloudlet IDs and mobile node IDs. Whenever mobile node wants communicate with other node in another cloudlet, the cloudlet has to send details of cloudlet Id and mobile node Id. The server acts as proxy between cloudlet to send and receive data from one cloudlet to another cloudlet.

4.3.2. Service based architecture. This architecture aims to disclose how data is managed and shared among nodes in cloudlet and among cloudlet as well. The behaviour of service base architecture discussed through following two services.

File editing.. The file can be edited directly in remote cloud otherwise the whole file can be downloaded into local cloudlet where mobile users can edit it. Once the editing is done that file send back to cloud through wireless network. It is cloudlet agent responsibility to maintain synchronization between cloud and cloudlet. When multiple users are allowed to edit file in cloudlet [38].

The steps for file editing in service architecture are:

- The node looks for nearby cloudlet and connect to it after successful connections. The node would calculate the round trip cost from its location.
- After successful connection with remote cloud server request for file editing then, node calculations the round trip cost from its location.
- If the cost of the cloudlet file editing is less than remote cloud file editing, then file editing is done at cloudlet itself, otherwise in remote cloud server.
- The cloudlet update file after successful editing.

Video streaming. The node does video streaming available in remote server by means of cloudlet nodes to save time and energy instead of streaming directly from remote server.

The steps for video streaming in service architecture are:

- The node looks for nearby cloudlet and connects to it. After successful connections, the node calculates round trip cost from its location.

TABLE 4.1
Mobile Configuration

Year	Memory	Device Type
2018	1TB	High-end devices
2018	1TB	Low-end devices

- After successful connection with remote servers the node requests for video streaming and then calculates the round trip cost from its place.
- If the round trip cost of cloudlet is less than remote server then video streaming would take place at cloudlet otherwise from remote server.
- The synchronization concept is not here in video streaming unlike file editing, the video being while downloading into cloudlet.

4.4. Pocket cloudlet. The development of internet has been raising from last few decades by introducing various mobile devices such as smart phones, tablets and other PDAs. The mobile users are able to leverage cloud services by means of advent of internet. The communication channels help to mobile devices to access cloud services. These challenges serves request and bring issues such as energy overhead and latency issues. Two major constrains are mainly raising from radio link i.e. network availability and energy consumption. The mobile communication is not able to serve increasing demands of mobile users the mobile cloud computing provides solutions to address issues. The configuration of mobile devices can be developed in terms of processor and memory size. For example nowadays mobiles are manufactured with extension of memory about 64 GB of non-volatile. The expansion of mobile storage with nominal restrictions can be observed by researches.

Mobile devices having enough storage to store large amount of data locally. The storage availability is specified in Table 4.1. The advent feature hash technology provides more storage irrespective of local storage of mobile device. Most of storage space in mobile devices remain unused. The availability of storage space in mobile device can be used for storing some cloud services locally.

In cloud, specific services are being used often by mobile users. The major usage of specific resources causes data to be downloaded over and over when user download data recursively which causes high latency and more energy consumption. These problems can be solved by means of internal mobile storage to store cloud services which are accessed often. The concept of pocket cloud is formed by storing frequently used data [39]. Pocket cloud can reduce power consumption, high latency and other overhead issues. Storing part of service or entire cloud services into mobile devices. By increasing storage capacity of mobile device, more number of cloud services can be stored in device. The pocket cloud make mobile device more efficient in every possible way.

The pocket cloud provides advantages to mobile users are:

- Pocket cloud enhance user experience by storing cloud services locally.
- Mobile users can access data at any time without delay.
- Pocket cloudlet minimizes burden on cloud servers and radio links in network.
- Since every user have individual mobile device, it is easy to identify usage pattern, storing cloud services in mobile devices on demand basis.
- Security levels has been enhanced for storing sensitive in mobile storage.

The data stored on mobile devices are updated in regular intervals sensitive data is updated frequently on other hand less sensitive data are updated when resource is not constrained i.e. when mobile is getting charged, when network connection is very high. Pocket cloud follows certain protocols before storing services:

- The size of storing data varies each service.
- Security mechanism should be followed before storing data.
- Efficient architecture should be used for storing and retrieving huge amount of data.
- Updating of data by means of proper mechanism on real-time basis.

4.4.1. Pocket cloud architecture (PCA). Pocket cloud architecture Fig 4.4 provides the way in which cloud services are accessed by mobile devices. According to research study, more than 90% user visit less than 1000 URLs in the specific time period. Cloud services can be transferred from cloud to mobile devices by means

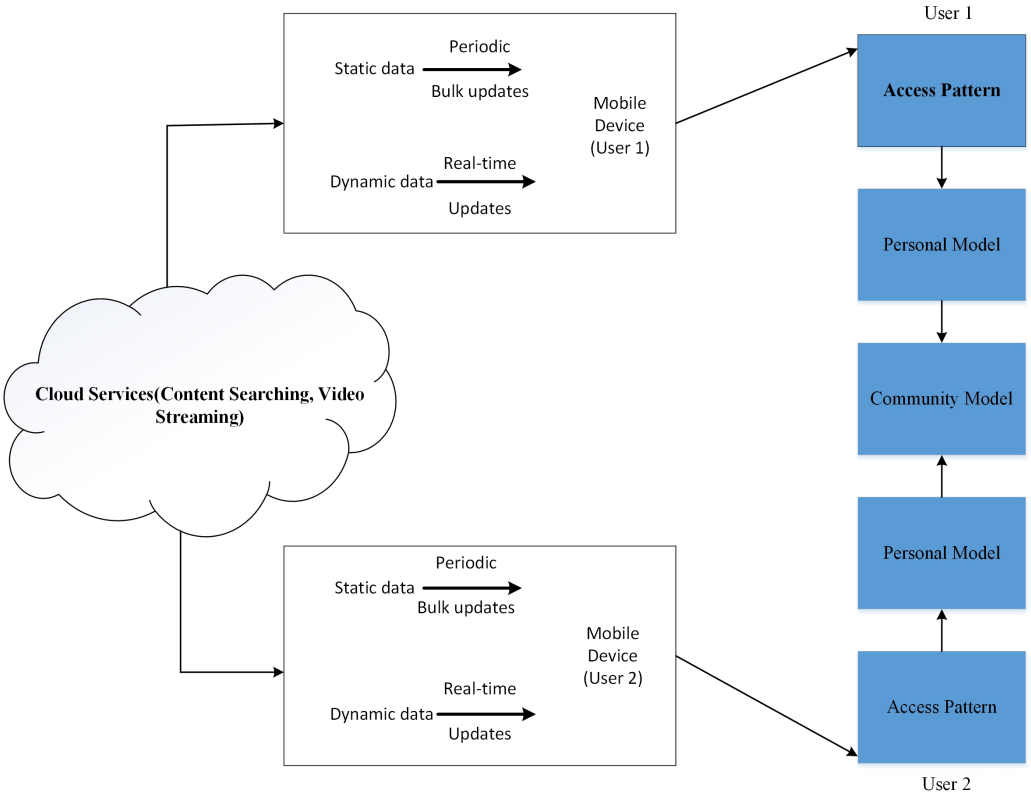

FIG. 4.4. *Architecture of PCA*

of 3G, 4G and Wi-Fi long range networks. Mobile devices store data for future use. Data can be classified into two categories static data and dynamic data.

Static data is updated periodically whenever network availability is high with more battery life for example at night time when mobile getting charged. Dynamic data is updated on real-time basis, which require high network availability.

Mobile users access data stored in devices based on patterns which are formed by cloud. These patterns also known as access pattern. Access pattern can be formed as personal model to maximize usage of cloud services. By combining all individual personal models to be formed as community model.

4.5. Comparision between cloud and cloudlet. Many existing research work have mistaken by mentioning that both cloud & cloudlet are same. But, it is not true, each of them have their own architecture, nature of functioning. There are various parameters to prove both technologies are different paradigms. Table 4.3 shows comparison between cloud and cloudlet.

4.6. Summary of Literature Review on Cloudlet. Modh et al [40]have characterized the idea of blending of two new technologies which are mobile computing and cloud computing into one known as Mobile Cloud Computing (MCC). Mobile Cloud Computing (MCC) helps in providing rich benefits of both the combined technologies. Cloud computing helps to overcome the problem of storage limit as well as increasing the computational power, processing power and storage of various mobile applications. Mobile computing helps in easy access and retrieval of any data stored in our mobile device. Still there are a few difficulties identified with Mobile Cloud Computing. In this paper they have presented different difficulties of systems like Internet availability, data transmission, dormancy, access speed and so on for MCC. They additionally discuss about the one cloudlet solution system for the fundamental system issue of idleness that influences the upgrade of MCC.

As mobile devices are being used widely they are playing an significant role in every individuals life. In any

case, mobile devices have the limitations, for example, low computational power and quick depletion of power from their batteries. Loai et al in [41], have found the solution that the services of mobile cloud computing can be utilized to run specific assignments at the cloud and send the results to end user devices addition memory and then handles the power. This model of mobile computing is effective with the view of cloudlet scheme. This mobile cloud computing model reduces the expensive technologies such as Wi-Fi, 3G/4G, networks by communicating with the cloudlet directly rather than being in contact with venture cloud server. In addition to this model, a plan which involves the interaction of cloudlet with each other. This plan is certainly known as ace cloudlet administration plan. Colleges, institutions and healthcare centre widely make use of effective Mobile Cloud Computing (MCC) where it necessary to store and access the large amount of information. The MCC model in which the results of non-cloudlet are outperformed is discussed in this model.

Mobile computing is restricted with limitations such as battery, memory and capacity etc. By overwhelming these restrictions the Mobile Cloud Computing has become familiar by offloading the tasks that are beyond the range of end user device capacity to the cloud and later processing the those tasks in cloud are sent back to the user device. Utilization of mobile cloud computing tends to reduce the power consumption and time consumption to process the tasks that are offloaded to the cloud. MCC can be utilized efficiently to reduce the power consumption and time consumption with the help of cloudlet based MCC framework proposed by Jaraweh et al in [42].Experiment outcomes have demonstrated that utilizing the proposed system lessens the power utilization from the cell phone, in addition to decreasing the correspondence inactivity when the cell phone asks a task to occur remotely while keeping high calibre of administration stander.

Offloading of tasks with high intensity in the mobile into the cloud server by rising innovation mobile cloud computing. Raei et al in [42] proposed analytical based performance model to overcome the problems occurred in expecting results due to the MCC attributes like portability, unsteadiness of 3G/Wi-Fi and virtualization that cannot be predicted. A technique called fixed point iteration technique sets the cyclic reliance between the problematic sub-models. Physical Machine (PM) acts as piece of cloudlet otherwise an open cloud when virtual machine (VM) is maintained on physical machine. This type of MCC is executed based on parameters like network failure and workload. The effects caused due to this parameters are measured based on two measures: request dismissal likelihood and mean reaction delay. This model is understood by the use of SHARPE programming bundle.

Although mobile devices are increasing rapidly in our day to day life. They are limited with certain constraints. Assets in mobile device can be reduced by offloading the high intensity tasks into the nearby cloud with the help of mobile distributed computing. Cloudlet is an essential part for the customer cloud system in focalizing advancement in cloud registering and mobile computing. In this paper, Pang et al [43]exhibits a broad review of examines on cloudlet based. They initially hindsight the development of cloudlet based mobile computing. From that point forward, they audited the current research on the cloudlet based processing offloading and information offloading. Two cases regarding the cloudlet are presented and examined the present scenario, endeavours and upcoming bearings of this field.

According to Dinh et al in [3], MCC has been enlightened with the potential innovation for cloud administrators along with the increase in mobile application and cloud computing idea. Whenever the tasks received by mobile cannot be processed by the mobile, MCC organizes the cloud computing into mobile condition such that cloud computing process the task by overcoming the limitation of mobile device such as battery life, stockpiling and transmission capacity including condition i.e. how heterogeneous, versable and accessible it is and security. We discuss the basic outline of MCC with definition and how it is useful in engineering and its application.

Gai et al [44] stated that utilizing Mobile Cloud Computing (MCC) to empower cloud clients to procure advantages of cloud computing by an ecological amicable technique is an effective procedure for taking care of current modern requests. However, the limitations of remote data transmission and gadget limit have brought different impediments, for example, additional vitality waste and idleness delay, while conveying MCC. A dynamic energy aware cloudlet-based mobile cloud computing model (DECM) has been proposed to overcome the limitation such as additional vitality waste and idleness of remote data transmission and device limit. This model DECM makes use of extra vitality during the interchanging of remote data by dynamic cloudlet-based model (DCL). In this paper, they inspect their model by a recreation of functional situation and give strong outcomes to the assessments. The principle commitments of this paper are twofold. In the first place, this paper

is the primary investigation in taking care of vitality squander issues inside the dynamic systems administration condition. Second, the proposed display furnishes future research with a rule and hypothetical backings.

According to Sanaei et al in [45], MCC is the resultant of rapid and repeated research exercises that are performed in favour of increasing various mobile devices with the help of different cloud advantages. Encouragement if interoperability, transportability and incorporation between the different stages is important in the middle of such different condition. The facilitators in MCC helps in examining the heterogeneity to understand and also difficulties. The successful MCC undergoes literary struggles when cloud computing and cloud figuring is integrated. In the present paper, we discuss about the characterization of MCC, how to illuminate the important endeavours, testing diversification in figuring. Heterogeneity is classified as equipment, stage, highlight, API and system after the base of heterogeneity is explored. The improvement of cross-stage cloud applications is blocked due to the multi-dimensional heterogeneity in MCC which develops application and code discontinuity issues. Difficulties due to the effects of diversification are recognized through the research and we discuss about the methodologies like virtualization, middleware and service oriented architecture (SOA) is taken care by overcoming heterogeneity.

The Table 4.2 points towards several papers on cloudlet.

5. Security, Privacy And Challenges In Mobile Cloud Computing (MCC).

5.1. Layers of cloud computing.

Data Centers Layer. It provides hardware facility and infrastructure for cloud. in which, numerous servers are connect via internet to provide services to users [56].

Infrastructure as a Service (IaaS). It provides hardware, storage, servers, networking component for users, and users will pay as you go [56].

Platform as a Service (PaaS). It provides advanced environment for application developing, deploying, and testing [57].

Software as a Service (SaaS). It shares available applications and information remotely via internet with multiple users and pay only for they use [57].

5.2. Security Breaches And Issues.

5.2.1. Data ownership. Cloud computing provide facility to user to store purchased data such as video files, audio files, e-books remotely. There can be a chance that user will not be able to access bought data from server and should be aware of access permission of bought data. Mobile cloud computing solve this kind of breaches by using context information like location, capabilities of device, user profile [56].

5.2.2. Privacy. Privacy is one of the significant challenge in mobile cloud computing. Some mobile applications store users personal data in cloud by hiring storage. Third party companies share users sensitive data with government agencies without users permission [45].

5.2.3. Security Issues. Mobile devices are venerable to attacks and chances of stolen data because mobile devices are unprotected. An unauthorized user easily gets access of authorized users. Few security issues mentioned as follows [58]:

- Data loss from loss/stolen devices.
- Information stealing from mobile malware.
- Data leakage happens with untrusted third party.
- Insecure network access and unreliable access points.
- Vulnerabilities with in devices, operating system, design and third party application.
- Near field communication (NFC) proximity based hacking.

The concept of security breach is that the unauthorized user access sensitive data of other user without corresponding user permission. Many organizations treat their data as voluble asset of their company. It is well known fact that ever user knows that it is impossible to avoid loss of data in network world. There are many ways that data could get lost [59].

TABLE 4.2
Summary of Literature Review on Cloudlet

Ref	Cloud	Cloudlet	Problem	Solution
[41]		CL	Optimizing power consumption and high latency.	MCCSIM
[42]		CL	Minimizing power consumption and latency.	Cloudlet based MCC
[46]		CL	Optimizing performance of cloudlet by considering workload, recourse capacity, connection failure rate, request rejection probability, mean response delay.	Fixed-point iteration algorithm
[47]		CL	Optimizing cloudlet selection and resource provisioning.	Round Robin with load-degree algorithm.
[44]	C	CL	Reducing power usage for cloud selection.	Dynamic cloudlet selection model
[48]	C	CL	Developing hybrid application by optimizing power and latency issues.	Automation Script with Exhaustive Search algorithm
[49]	C	CL	Cloudlet selection and processing with low power consumption and latency .	Recourse allocation using centralized proxy server.
[50]	C	CL	Optimizing power consumption and latency by distributing tasks among cloudlets.	MILP linear programming model
[51]	C	CL	Optimizing bandwidth and resource in cloudlet based MCC.	triple-stage Stackelberg game using backward method.
[52]	C	CL	Minimizing CPU execution time and memory usage.	Bee's life algorithm
[53]		CL	Load balancing among fog nodes to optimize power and recourse usage.	Optimal Multi-User Small Cell Clustering
[54]		CL	Optimizing user access mode selection.	Evolution Algorithm.
[55]		CL	Distribution of load among nodes to least latency.	Matching theory.

TABLE 4.3
Comparision Between Cloud computing and cloudlet

Parameters	**Cloud Computing(CC)**	**Cloudlet**
State	Hard and Soft State	soft-state
Management	Professionally Managed	Self Managed
Environment	Large space Required for maintaining servers	Established at organization
ownership	Centralized Management	Decentralized Management
Network	Internet	LAN
Sharing	Unlimited devices Communicate and share data	Limited Devices only share data and communicate
cost	Investment is high	Investment is Low
security	More secure and Reliable	Less secure

5.3. Mobile cloud computing suffers from following risks.

- User does not know where exactly mobile data is stored in mobile cloud computing environment which leads user does not have control over stored data.
- Physical damage of cloud server, loss of encoding key and due to malicious insider, risk of data loss may arise.
- Customer may intentionally plant virus of phishing attack in to cloud server which may lead to loss of other users data, and cloud provide is unable to do anything because violation of privacy policy of company.
- When cloud provider services number of users, flaw in encryption may lead to unauthorized encryption.
- As per service level agreement cloud provider should maintain security level Security risk may rise in Iaas due to lack of isolation among hosted virtual machine in single server.
- Most users share their sensitive and personal data through mobile application for instance online transaction that can be attacker main target.

5.4. Security Issues of Mobile Cloud Computing. This section describe different possible attacks in mobile cloud computing.

SQL Injection Attack. The attacker adds malicious code in standard SQL so that attacker get unauthorized access of database, is able to access sensitive data [60].

Browser Security. Every user use browser to transmit data over network. Browser uses SSL technology to provide protection to user authentication details. But attacker always tries to break user credentials by using sniffing package which is installed on intermediary host.

Denial of service. The attacker prevents user accessing services from cloud [61].

Cookie poisoning. The attacker changes content of cookie to have illegal access of application [62].

Flooding attacks. Attacker continuously sends resource required request to cloud server so that cloud get flooded with ample requests. cloud has feature called scalability based on number of requests given send by users but intruder stop server from serving actual users by sending requests rapidly [63].

Incomplete data deletion. When data is deleted, it does not remove copy of that data from backup server until the operation system of the server is commanded specially by network service provider. Precise data deletion is impossible because replicas stored in backup server [63].

Usually user is able to connect cloud server by using web browser or web services [70]. Web service attacks also effect cloud computing. In spite of cloud security uses XML signature for protecting an element name, attribute, value from attackers.

Table 5.1: Comparison among different security models of addressing security and privacy issues in MCC

Authors/year	**Approach**	**Trusted level**	**Security attribute provided**	**Benefits**	**Drawbacks**	**Conclusion**
1	**2**	**3**	**4**	**5**	**6**	**7**
J.Oberheide et al. Virtualized in-cloud security services for mobile devices [62] (2008)	CloudAV	Fully trusted	Antivirus, Security as a Service	Reduced On Device software complexity and power consumption	Disconnected operation and privacy loss	By moving the detection capabilities to a network service, we gain numerous benefits including increased detection coverage, less complex mobile software, and reduced resource consumption.
Zhang et al. ACM workshop on Cloud computing security [64] (2009)	Cloudlet	Semi-trusted	Task partitioning	Good tradeoffs between processing overhead and communication cost	Security of Weblet can be improved with other techniques.	support flexible and efficient ways to augment computing, storage, and communication capabilities of applications for resource-constrained Devices.
Xiao and Gong et al. International Conference on Mobile Data Management [65] (2010)	Lightweight algorithm	Semi trusted	Authorization of users data in cloud	Automatic Dynamic updating of credential information	More processing and energy burden on mobile device	merit of dynamic credential is that the attackers difficulty to fake the credential grows with time.

1	2	3	4	5	6	7
Wang and Wang et al. 11th International Conference on Mobile Data Management, MDM [66] (2011)	Top down spatial cloaking	Distrusted	Privacy preserving framework in location based Scheme	Reduced communication cost by doing spatial cloaking based on the historical data in cloud.	More energy consumption and processing burden on mobile device	top-down spatial cloaking algorithm,and devised optimization are proposed to reduce the communication cost.
Huang et al. MobiCloud: building secure cloud framework for mobile computing and communication [58] (2010)	MobiCloud	Distrusted	Security in Storage as a Service in MANET	Secured data while using Public Cloud	Increased cost due to two cloud providers	The mobicloud framework will enhance communication by addressing trust management, secure routing, risk management issues in network.
G. Portokalidis et al. Annual Computer Security Application Conference (ACSAC) [67] (2010)	Threat detection in Smartphone based on CloudAV	Fully trusted	Security as a Service	Reduced transmission overhead and energy consumption	More Cloud usage cost. battery life is consumed more.	it offers more comprehensive security than possible with alternative models.

1	2	3	4	5	6	7
R.Chow et al. ACM Cloud Computing Security Workshop [68](2010)	Policy based cloud authentication platform	Fully Trusted	Authentication of user.	Authentication based on behavioral data of user	Privacy threat	our proposed authentication approach potentially improves security and usability.
Jia et al. IEEE Conference on Computer Communications Workshops, INFOCOM WKSHPS [59](2011)	Proxy reencryption (PRE) scheme and Identity based encryption (IDE) scheme	Semi trusted	Secure data Service	Reduced cost of updating of access policy and communication cost	Reduced cost of updating of access policy and communication cost	identity based proxy re-encryption scheme to make mobile users easily implement fine-grained access control of data and also guarantee the data privacy in the cloud
Yang et al. Provable data possession of resource constrained mobile devices in cloud computing [69] (2011)	extended the public provable data possession scheme	Distrusted	ensures privacy, confidentiality and integrity of user data stored on cloud	Reduced energy and processing requirement on mobile device	Degradation of performance with the increase in no. of users in Trusted Party Agent (TPA). Cost also increases due to two cloud service providers.	

1	2	3	4	5	6	7
Saman Zonouz et al. Science Direct journal of Computers and security [63] (2013)	Secloud for Smart phones	Trusted	cloud based comprehensive and lightweight security for smart phones	Reduced energy and processing requirement on mobile device for providing security in mobile device	Cloud assumes to be fully trusted which needs to be reconsidered. The personal data of users accessed to the cloud can affect the privacy issues	Secloud provides a powerful, yet resource-friendly, protection for smart phones by performing the security analysis on an emulated version of the devices, running inside a cloud
Vijay Varadharajan et al. IEEE Transactions On Network and Service Management [70](2014)	Security as a Service Model	Trusted	Virtualization technology and VMM security functionalities	Offers a baseline security to the provider to protect its own cloud infrastructure	Insider attack from Tenant Domain and Cloud Service Provider	security as a service model that a cloud provider can offer to its multiple tenants and customers of its tenants
Qiao Yan and F. Richard Yu et al. IEEE Communications Surveys & Tutorials [61](2016)	Softwaredefined networking (SDN)	Trusted	Networking-asa-service (NaaS), control and data planes are decoupled	software-based traffic analysis, centralized control, global view of the network, dynamic updating of forwarding rules	Security of SDN itself remains to be addressed and potential DDoS vulnerabilities exist across SDN platforms	Defending DDOS attack by making full use of SDN-base cloud advantages in cloud environment.

6. Conclusion. In this article, we have discussed about various concepts in mobile cloud computing: (1) Mobile computing, (2) cloud computing, (3) mobile cloud computing,(4) offloading approach, (5) cloudlet approach and (6)security and privacy. We have given extensive survey on existing frameworks on computation offloading. Even through, there are various frameworks presented, but the objective of each framework is to improving mobile performance by reducing energy consumption and response time.

We have provided a survey on emerging cloudlet technology and challenges. the concept of cloudlet aims to address high latency and response time issues by brining cloud recourses closer to mobile user. We have analyzed comparison among popular cloudlet architecture and selection techniques; the secure routing protocol used to protect communication channel between devices and cloud as well as need to be addressed several issues such as data integrity, authentication, authorization and access control. We believe that by introducing edge computing architecture for optimizing mobile performance in MCC will take place in near feature.

REFERENCES

[1] R. Meulen, "Gartner Says Global Smartphone Sales to Only Grow 7 Per Cent in 2016."
[2] B. Butler, "Gartner: Cloud Putting Crimp in Traditional Software, Hardware Sales," *Network World*, 2012.
[3] H. T. Dinh, C. Lee, D. Niyato, and P. Wang, "A survey of mobile cloud computing: architecture, applications, and approaches," *Wireless communications and mobile computing*, vol. 13, no. 18, pp. 1587–1611, 2013.
[4] M. Tulloch, *Introducing Windows Azure for IT Professionals*. Microsoft Press, 2013.
[5] R. Buyya, "Introduction to the ieee transactions on cloud computing," *IEEE Transactions on Cloud Computing*, vol. 1, no. 1, pp. 3–21, 2013.
[6] P. Mell, T. Grance, and Others, "The NIST definition of cloud computing," 2011.
[7] C. V. N. Index, "Global mobile data traffic forecast update 2014–2019 white paper, feb 2015," *See: http://www. cisco. com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862. html*, 2015.
[8] L. Yang, J. Cao, Y. Yuan, T. Li, A. Han, and A. Chan, "A framework for partitioning and execution of data stream applications in mobile cloud computing," *ACM SIGMETRICS Performance Evaluation Review*, vol. 40, no. 4, pp. 23–32, 2013.
[9] A. Klein, C. Mannweiler, J. Schneider, and H. D. Schotten, "Access schemes for mobile cloud computing," in *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010, pp. 387–392.
[10] P. Bahl, R. Y. Han, L. E. Li, and M. Satyanarayanan, "Advancing the state of mobile cloud computing," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 2012, pp. 21–28.
[11] E. Miluzzo, R. Cáceres, and Y.-F. Chen, "Vision: mClouds-computing on clouds of mobile devices," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 2012, pp. 9–14.
[12] R. Kamal, *Mobile Computing*. Oxford University Press, Inc., 2008.
[13] L. S. Ashiho, "Mobile technology: Evolution from 1G to 4G," *Electronics for you*, pp. 94–98, 2003.
[14] M. Armbrust, A. Fox, R. Griffith, A. D. Joseph, R. Katz, A. Konwinski, G. Lee, D. Patterson, A. Rabkin, I. Stoica, and Others, "A view of cloud computing," *Communications of the ACM*, vol. 53, no. 4, pp. 50–58, 2010.
[15] R. Buyya, J. Broberg, and A. M. Goscinski, *Cloud computing: Principles and paradigms*. John Wiley & Sons, 2010, vol. 87.
[16] D. Durkee, "Why cloud computing will never be free," *Queue*, vol. 8, no. 4, p. 20, 2010.
[17] M. D. Dikaiakos, D. Katsaros, P. Mehra, G. Pallis, and A. Vakali, "Cloud computing: Distributed internet computing for IT and scientific research," *IEEE Internet computing*, vol. 13, no. 5, 2009.
[18] K. Chard, S. Caton, O. Rana, and K. Bubendorfer, "Social cloud: Cloud computing in social networks," in *Cloud Computing (CLOUD), 2010 IEEE 3rd International Conference on*. IEEE, 2010, pp. 99–106.
[19] B.-G. Chun, S. Ihm, P. Maniatis, M. Naik, and A. Patti, "Clonecloud: elastic execution between mobile device and cloud," in *Proceedings of the sixth conference on Computer systems*. ACM, 2011, pp. 301–314.
[20] K. Akherfi, M. Gerndt, and H. Harroud, "Mobile cloud computing for computation offloading: Issues and challenges," *Applied Computing and Informatics*, 2016.
[21] E. Cuervo, A. Balasubramanian, D.-k. Cho, A. Wolman, S. Saroiu, R. Chandra, and P. Bahl, "MAUI: making smartphones last longer with code offload," in *Proceedings of the 8th international conference on Mobile systems, applications, and services*. ACM, 2010, pp. 49–62.
[22] M. Satyanarayanan, P. Bahl, R. Caceres, and N. Davies, "The case for vm-based cloudlets in mobile computing," *IEEE pervasive Computing*, vol. 8, no. 4, 2009.
[23] H. Qian and D. Andresen, "Jade: Reducing energy consumption of android app," *the International Journal of Networked and Distributed Computing (IJNDC), Atlantis press*, vol. 3, no. 3, pp. 150–158, 2015.
[24] B. Zhao, Z. Xu, C. Chi, S. Zhu, and G. Cao, "Mirroring smartphones for good: A feasibility study," in *International Conference on Mobile and Ubiquitous Systems: Computing, Networking, and Services*. Springer, 2010, pp. 26–38.
[25] R. Kemp, N. Palmer, T. Kielmann, and H. Bal, "Cuckoo: a computation offloading framework for smartphones," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2010, pp. 59–79.
[26] F. Xia, F. Ding, J. Li, X. Kong, L. T. Yang, and J. Ma, "Phone2Cloud: Exploiting computation offloading for energy saving on smartphones in mobile cloud computing," *Information Systems Frontiers*, vol. 16, no. 1, pp. 95–111, 2014.
[27] S. Kosta, A. Aucinas, P. Hui, R. Mortier, and X. Zhang, "Thinkair: Dynamic resource allocation and parallel execution in the cloud for mobile code offloading," in *Infocom, 2012 Proceedings IEEE*. IEEE, 2012, pp. 945–953.

[28] T. Verbelen, P. Simoens, F. De Turck, and B. Dhoedt, "Cloudlets: Bringing the cloud to the mobile user," in *Proceedings of the third ACM workshop on Mobile cloud computing and services*. ACM, 2012, pp. 29–36.
[29] M. Satyanarayanan, Z. Chen, K. Ha, W. Hu, W. Richter, and P. Pillai, "Cloudlets: at the leading edge of mobile-cloud convergence," in *Mobile Computing, Applications and Services (MobiCASE), 2014 6th International Conference on*. IEEE, 2014, pp. 1–9.
[30] S. Simanta, K. Ha, G. Lewis, E. Morris, and M. Satyanarayanan, "A reference architecture for mobile code offload in hostile environments," in *International Conference on Mobile Computing, Applications, and Services*. Springer, 2012, pp. 274–293.
[31] S. Clinch, J. Harkes, A. Friday, N. Davies, and M. Satyanarayanan, "How close is close enough? Understanding the role of cloudlets in supporting display appropriation by mobile users," in *Pervasive Computing and Communications (PerCom), 2012 IEEE International Conference on*. IEEE, 2012, pp. 122–127.
[32] Z. Dou, "Benefits of utilizing an edge server (cloudlet) in the mocha architecture," Ph.D. dissertation, University of Rochester. Department of Electrical and Computer Engineering, 2013.
[33] D. Fesehaye, Y. Gao, K. Nahrstedt, and G. Wang, "Impact of cloudlets on interactive mobile cloud applications," in *Enterprise Distributed Object Computing Conference (EDOC), 2012 IEEE 16th International*. IEEE, 2012, pp. 123–132.
[34] Y. Li and W. Wang, "Can mobile cloudlets support mobile applications?" in *Infocom, 2014 proceedings ieee*. IEEE, 2014, pp. 1060–1068.
[35] J. Flinn and M. Satyanarayanan, *Energy-aware adaptation for mobile applications*. ACM, 1999, vol. 33, no. 5.
[36] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," *Wireless networks*, vol. 8, no. 2-3, pp. 153–167, 2002.
[37] H. Wang, "Accelerating mobile-cloud computing using a cloudlet," Ph.D. dissertation, University of Rochester. Department of Electrical and Computer Engineering, 2013.
[38] J. Wu and H. Li, "On calculating connected dominating set for efficient routing in ad hoc wireless networks," in *Proceedings of the 3rd international workshop on Discrete algorithms and methods for mobile computing and communications*. ACM, 1999, pp. 7–14.
[39] E. Koukoumidis, D. Lymberopoulos, K. Strauss, J. Liu, and D. Burger, "Pocket cloudlets," in *ACM SIGPLAN Notices*, vol. 46, no. 3. ACM, 2011, pp. 171–184.
[40] R. M. Modh and J. M. Patel, "A study on improving challenges of network with cloudlets for mobile cloud computing," *International Journal of Global Research in Computer Science (UGC Approved Journal)*, vol. 4, no. 4, pp. 143–146, 2013.
[41] A. T. Lo'ai, W. Bakheder, and H. Song, "A mobile cloud computing model using the cloudlet scheme for big data applications," in *Connected Health: Applications, Systems and Engineering Technologies (CHASE), 2016 IEEE First International Conference on*, 2016, pp. 73–77.
[42] Y. Jararweh, L. Tawalbeh, F. Ababneh, and F. Dosari, "Resource efficient mobile computing using cloudlet infrastructure," in *Mobile Ad-hoc and Sensor Networks (MSN), 2013 IEEE Ninth International Conference on*. IEEE, 2013, pp. 373–377.
[43] Z. Pang, L. Sun, Z. Wang, E. Tian, and S. Yang, "A survey of cloudlet based mobile computing," in *Cloud Computing and Big Data (CCBD), 2015 International Conference on*. IEEE, 2015, pp. 268–275.
[44] K. Gai, M. Qiu, H. Zhao, L. Tao, and Z. Zong, "Dynamic energy-aware cloudlet-based mobile cloud computing model for green computing," *Journal of Network and Computer Applications*, vol. 59, pp. 46–54, 2016.
[45] Z. Sanaei, S. Abolfazli, A. Gani, and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 1, pp. 369–392, 2014.
[46] H. Raei, N. Yazdani, and R. Shojaee, "Modeling and performance analysis of cloudlet in Mobile Cloud Computing," *Performance Evaluation*, vol. 107, pp. 34–53, 2017.
[47] R. Somula and R. Sasikala, "Round robin with load degree: An algorithm for optimal cloudlet discovery in mobile cloud computing," *Scalable Computing: Practice and Experience*, vol. 19, no. 1, pp. 39–52, 2018.
[48] A. Akbar and P. R. Lewis, "Towards the optimization of power and bandwidth consumption in mobile-cloud hybrid applications," in *Fog and Mobile Edge Computing (FMEC), 2017 Second International Conference on*. IEEE, 2017, pp. 213–218.
[49] A. Mukherjee, D. De, and D. G. Roy, "A power and latency aware cloudlet selection strategy for multi-cloudlet environment," *IEEE Transactions on Cloud Computing*, 2016.
[50] Y. Jararweh, M. Al-Ayyoub, M. Al-Quraan, A. T. Loai, and E. Benkhelifa, "Delay-aware power optimization model for mobile edge computing systems," *Personal and Ubiquitous Computing*, vol. 21, no. 6, pp. 1067–1077, 2017.
[51] S. Meng, Y. Wang, Z. Miao, and K. Sun, "Joint optimization of wireless bandwidth and computing resource in cloudlet-based mobile cloud computing environment," *Peer-to-Peer Networking and Applications*, vol. 11, no. 3, pp. 462–472, 2018.
[52] S. Bitam, S. Zeadally, and A. Mellouk, "Fog computing job scheduling optimization based on bees swarm," *Enterprise Information Systems*, vol. 12, no. 4, pp. 373–397, 2018.
[53] J. Oueis, E. C. Strinati, and S. Barbarossa, "The fog balancing: Load distribution for small cell cloud computing," in *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*. IEEE, 2015, pp. 1–6.
[54] S. Yan, M. Peng, M. A. Abana, and W. Wang, "An evolutionary game for user access mode selection in fog radio access networks," *IEEE Access*, vol. 5, pp. 2200–2210, 2017.
[55] M. S. Elbamby, M. Bennis, and W. Saad, "Proactive edge computing in latency-constrained fog networks," in *Networks and Communications (EuCNC), 2017 European Conference on*. IEEE, 2017, pp. 1–6.
[56] H. Wang, S. Wu, M. Chen, and W. Wang, "Security protection between users and the mobile media cloud," *IEEE Communications Magazine*, vol. 52, no. 3, pp. 73–79, 2014.
[57] D. Dev and K. L. Baishnab, "Notice of Violation of IEEE Publication Principles A Review and Research Towards Mobile

Cloud Computing," in *Mobile Cloud Computing, Services, and Engineering (MobileCloud), 2014 2nd IEEE International Conference on*. IEEE, 2014, pp. 252–256.

[58] D. Huang, X. Zhang, M. Kang, and J. Luo, "MobiCloud: building secure cloud framework for mobile computing and communication," in *Service Oriented System Engineering (SOSE), 2010 Fifth IEEE International Symposium on*. Ieee, 2010, pp. 27–34.

[59] W. Jia, H. Zhu, Z. Cao, L. Wei, and X. Lin, "SDSM: a secure data service mechanism in mobile cloud computing," in *Computer Communications Workshops (INFOCOM WKSHPS), 2011 IEEE Conference on*. IEEE, 2011, pp. 1060–1065.

[60] P. Garg and V. Sharma, "An efficient and secure data storage in Mobile Cloud Computing through RSA and Hash function," in *Issues and Challenges in Intelligent Computing Techniques (ICICT), 2014 International Conference on*. IEEE, 2014, pp. 334–339.

[61] Q. Yan, F. R. Yu, Q. Gong, and J. Li, "Software-defined networking (SDN) and distributed denial of service (DDoS) attacks in cloud computing environments: A survey, some research issues, and challenges," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 1, pp. 602–622, 2016.

[62] J. Oberheide, K. Veeraraghavan, E. Cooke, J. Flinn, and F. Jahanian, "Virtualized in-cloud security services for mobile devices," in *Proceedings of the first workshop on virtualization in mobile computing*. ACM, 2008, pp. 31–35.

[63] S. Zonouz, A. Houmansadr, R. Berthier, N. Borisov, and W. Sanders, "Secloud: A cloud-based comprehensive and lightweight security solution for smartphones," *Computers & Security*, vol. 37, pp. 215–227, 2013.

[64] X. Zhang, J. Schiffman, S. Gibbs, A. Kunjithapatham, and S. Jeong, "Securing elastic applications on mobile devices for cloud computing," in *Proceedings of the 2009 ACM workshop on Cloud computing security*. ACM, 2009, pp. 127–134.

[65] S. Xiao and W. Gong, "Mobility can help: protect user identity with dynamic credential," in *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010, pp. 378–380.

[66] S. Wang and X. S. Wang, "In-device spatial cloaking for mobile user privacy assisted by the cloud," in *Mobile Data Management (MDM), 2010 Eleventh International Conference on*. IEEE, 2010, pp. 381–386.

[67] G. Portokalidis, P. Homburg, K. Anagnostakis, and H. Bos, "Paranoid android: versatile protection for smartphones," in *Proceedings of the 26th Annual Computer Security Applications Conference*. ACM, 2010, pp. 347–356.

[68] R. Chow, M. Jakobsson, R. Masuoka, J. Molina, Y. Niu, E. Shi, and Z. Song, "Authentication in the clouds: a framework and its application to mobile users," in *Proceedings of the 2010 ACM workshop on Cloud computing security workshop*. ACM, 2010, pp. 1–6.

[69] J. Yang, H. Wang, J. Wang, C. Tan, and D. Yu, "Provable data possession of resource-constrained mobile devices in cloud computing." *JNW*, vol. 6, no. 7, pp. 1033–1040, 2011.

[70] V. Varadharajan and U. Tupakula, "Security as a service model for cloud environment," *ieee transactions on network and service management*, vol. 11, no. 1, pp. 60–75, 2014.

Edited by: Rajkumar Rajasekaran
Received: Jun 29, 2018
Accepted: Oct 25, 2018