Multimedia Computational Offloading for 5G Mobile Edge Computing

Adhimuga Sivasakthi D (dasivasakthi0607@gmail.com)
Madras Institute of Technology https://orcid.org/0000-0001-5694-9547
Gunasekaran Raja
Anna University Chennai

Research Article

Keywords: Computation offloading, mobile edge computing, multihopmesh communication, edge server, energy efficiency

Posted Date: February 15th, 2021

DOI: https://doi.org/10.21203/rs.3.rs-221801/v1

Multimedia Computational Offloading for 5G Mobile Edge Computing

D Mr. D. Adhimuga Sivasakthi[*1] Dr. R. Gunasekaran[2],

[*1]*Research Scholar, Department of Computer Technology, Madras Institute Of Technology, Anna University, Chennai, India*

[*]*Email: dasivasakthi0607@gmail.com*

[2] *Professor & Head in Department of Computer Technology, Anna University, Chennai , Tamilnadu, India*

Abstract

Due to the tremendous growth of smartphone users with their massive usage of faster response, delay sensitive applications lead to high traffic demands, still not yet been met by current researchers. Thus, the ever-increasing computational demands are managed through computational offloading, which offload intensive workloads to the small cells having functionality similar to the resource-rich providers namely edge server. With the use of edge server, the heavily loaded computations get executed outside of the mobile device which results in minimization of mobile energy consumption and latency. Mobile edge computing is an emerging paradigm of commercial infrastructure for computational offloading, which enhances the power of smart mobile devices. Generally, edge provides cloud services and resources to the nearest proximity of users with radio access in fifth-generation (5G) networks for low latency, prompt response, and filtering. But, the frequent network failure automatically degrades the performance. Hence in this paper, we integrate the functionalities of Edge Server with Pico cells (ESP) for efficient traffic management and also propose a distributed Multi hop Mesh Middle layer (MMM) architecture for seamless communication with high resilience, minimal latency, and reduction of energy consumption. Generally, a pico cell is a distributed antenna system, an alternative to a repeater used to extend wireless

services to 100 users. We developed an Ant Social based Vector (ASV) optimization algorithm for an intelligent offload decision, which paves the way for backhaul routing conflicts. Simulation results of Cloud Sim show that offloading tiny devices integrated with significant additional computational capabilities would satisfy all the demands from each network within a 15ms delay.

Keywords: *Computation offloading; mobile edge computing; multihopmesh communication; edge server; energy efficiency*

1. Introduction

With the spectacular growth of the mobile devices, the massive demands of resource-hungry and delay sensitive critical mobile services, such as video format transformation, voice recognition, interactive gaming, virtual reality, augmented reality, real-time speech recognition and content-based image retrieval have attracted significant attention of future 5G networks. Since, these applications are delay sensitive [1] which resulted in an increasingly high computing demand, which address multiuser computation and heterogeneous network issues. For that, it is mandatory to have an exclusive virtual machine for each user which allows resource sharing by the assumption of peak workload demands as static. At the time of offloading, the demands for doing computations exceed the capacity of the mobile device. Due to the limitations of the computation capacity and cache size, resource-constrained applications cannot be executed efficiently on the mobile device. Most of the current work focuses on datasets with short data packets (below Kbits).The researchers of future wireless use cases e.g., Virtual Reality (VR), Augmented Reality (AR) and Tactile Internet (TI), demands not only transmission of a large packet but requires immense computing facilities. It is difficult for a mobile user with limited computation and storage resources to meet the requirements of current use cases. On the other hand, battery lifetime of mobile devices is the primary constraint, which indicates plethoric energy consumption [2] from local computing

which is not tolerable for the user to get better user experience. The file size of responses that comes back from the server is smaller when compared to the file size of computation or transmission. Thus, it is mandatory to keep the computational server close to the user's location.

Notice that the distance of cloud and cloudlets are farthest from a mobile device, so offloading to cloud server leads to delay in user request [3]–[5].Alternative techniques or adaptive algorithms are in need to offload intensive task to small cells. Small cell offloading brings new challenges such as computation workload, file size, wireless communication protocol, latency. But it subsidizes heterogeneity of computing, storage, communication resources, and diversity requirement of network services. Therefore, it is more preferable in situations where eNB's get overloaded in hotspots with high-density demand. For example, to prevent mobile users feeling as irritation, dizzy and nauseous, VR system has to guarantee the latency which should be less than the time humans start noticing the lag [6] (e.g., not more than 20ms).

By the year 2019, IDC reported that 45% of the IoT data generated by diverse requests gets processed, stored, and analyzed on edge [7]. Edge technology is first initiated by the European Telecommunications Standards Institute (ETSI). Edge is an emerging technology which promotes the necessity for communication and computation capabilities that were adaptively managed and shared among mobile devices. The access of small size nano datacenter in 4G and 5G network is termed as mobile edge. An edge device is an infrastructure of a heterogeneous router, gateway, switch, access points, or a base station and end-user devices. Generally, edge devices are required to have sufficient computational [8]-[12] and storage resources. It aid as a complement cloud to extend resources and services. In particular, response time is a primary factor for user satisfaction namely Quality of Experience (QoE) [13], [14]. Through Edge with Pico cell (ESP), swift response for the real-

time application and expansive resource demands of the end user gets satisfied with reduced data traffic and minimal delay. In order to satisfy end users, ESP scan be used as data delivery and sharing points, thereby computations have been performed with vast volumes of saved data. Caching at the pico cells may considerably reduce access latency and save a considerable amount of backhaul network traffic.

Hence, deploying computation and storage resources in ESP could enable a large number of diverse applications with instant response. Further, reducing the volume of downward traffic in the network core essentially brings energy efficiency. To offer distributed services with prompt response, a novel distributed Multi hop Mesh Middle layer (MMM) architecture has been proposed. As shown in Figure 1, MMM integrate the functionalities of ESP as a middle layer between user and data center. In this paper, we are considering ESPs as mesh topology, inter linked with multi hop communication for data filtration and seamless communication. The motivation of this paper is image processing, voice recognition and translation, m-gaming, cooperative spectrum sensing in cognitive radio cloud networks, because of resource-intensive nature of the application and intrinsic limitations of the mobile devices and communication medium.

For instance, we can transmit a high-resolution image simultaneously using ESP with a frame rate of 120 frames per second in a network for processing such as frame detection, accuracy in object recognition, virtual holograms modeling, etc. with short time span of less than 20ms.Our proposed MMM enables the applications and services to employ on top of multi-vendor computing platforms, also entitle the mobile operators with more number of customers. We consider multi-user computation offloading problem in 5G network with a scenario that 10% of 8 million people requested to view movies from Netflix at the same time. To handle this scenario, we treat all requests in parallel and access an infrastructure of Content Delivery Network (CDN) from ESP through backhaul as in distributed manner;

thereby the hosted content must not travel through various networks to reach the end users mobile.

Our main contributions are as follows:

- We design a novel architecture called Multi hop Mesh Middle layer(MMM) to support intensive computational offloading with self-healing framework.
- We formulate energy efficient handover and offloading in 5G with the assistance of SDN.
- We develop an Ant Social based Vector (ASV) optimization algorithm to minimize latency and overall network energy.

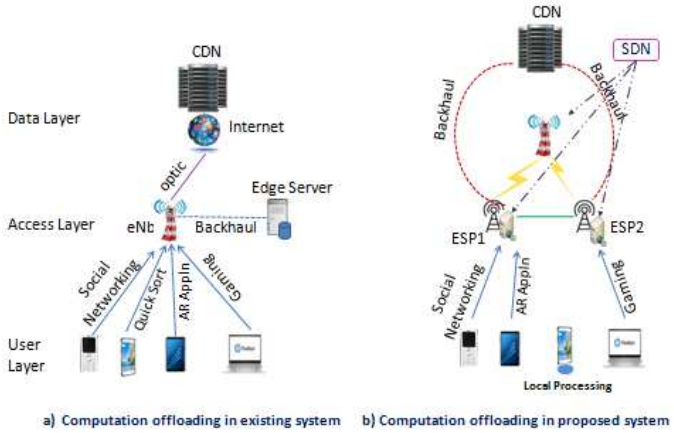

Fig. 1 Computation offloading Scenario a) Existing and b) Proposed Framework

The rest of this paper is organized as follows: Section II discusses related work, Section III covers system architecture, Section III.1 presents ASV in MMM Architecture, Section IV Section describes experimental analysis of the proposed MMM framework and Section V specifies the conclusion.

2. Related Work

Due to the bandwidth limitations in wireless networks there have been a few researches on efficient computation in 5G with 1) static assumption of offload parameters such as dynamic usage behavior, and network condition, 2) feasible elastic solutions. Generally, decision making is an essential aspect which is primarily based on the prediction of computing time.

So, in early 2000s, the main focusis to develop algorithms for offloading decisions to decide whether offloading would benefit mobile users or not [15]. Energy saving [16] is another crucial issue, hence complex and energy constrained computations could be offloaded to the cloud, which performs operations faster with retaining of mobile battery life [17],[18] but it leads to a signal overhead for a mobile user.

To overcome offloading problems, Lei et al. proposed a computation offloading mechanism where intensive application is executed completely at remote side, i.e., outside of the mobile device is termed as computation offloading. The method proposed in [19] addresses for a multi-client mobile environment in which virtual machine can be migrated from a cloud datacenter to mini-datacenter (edge), and can be managed through cloud surrogate for better mobile energy efficiency, but the mechanism is not guaranteed for IP continuity. To overcome the IP continuity issue, Wang et al. proposed a centralized improved particle swarm optimization algorithm for a single client application. The authors in [20] focus on minimization of mobile energy consumption, and optimal VM and CDN placement. The mechanism specified in this paper lacks composite services and other topological support except for LTE. Also, Lin et al. suggested a computation off loading framework for effective decision named Ternary Decision Maker (TDM) for a single client application. The scheme proposed in [21] dealt with reduction of both mobile energy consumption and response time. But this framework is suitable for LTE with intermittent connectivity.

To address the issues on the preservation of battery power through non-intermittent connectivity, Ragona et al. proposed a centralized framework (ECO-CLOUD) using mobile cloud computing through Near Field Communication (NFC) interfaces for a multi-client application [22].A distributed multisite computation offloading using genetic algorithm has been proposed by Zeinab et al. for a multi-user environment. The authors in [23] targets to optimize the overall energy consumption and response time by performing efficient

offloading. Still the framework does not support differences in access delays, proximities, computational capability or monetary cost. Besides, Roy et al. proposed a multi usermobile cloud computing algorithm for the optimum selection of cloudlet. The authors in [24] emphasis on the reduction of both battery power and latency time. This paper discusses about the rare availability of cloudlet, with complex offloading decision and time consuming selection scheme.

Mao et al. proposed a centralized Energy Efficient Computation Offloading (EECO) algorithm to solve the time-consuming issue which jointly optimizes partial offloading and radio resource allocation by incorporating the multi-access characteristics. The authors in [25] consider the utilization of energy for both computation and transmission. However, the theoretical analysis considers only priority-based channel allocation. Also, Yang et al. proposed a distributed joint optimization algorithm for both the problems of multiclass multichannel dimensional knapsack and performance function matrix. The authors in [26] provides mechanisms to manage competition among multi-users, effective computation partitioning. But it fails to addresses the maximization of average application throughput, decision making and consumption of mobile energy.

Jason et al. proposed architecture for centralized framework called AR edge computing for a multi-client application to perform partial remote live offloading. The authors in [27] addresses the issues of reduction in end-end latency and cost. However, the mathematical analysis is makes use of homogeneous network. To achieve efficient offloading, Demiriset al. presented a framework of demand based computational offloading to reduce monetary cost and to maximize energy savings. The authors in [28] dealt with balancing the competing demands and managing infrequent access of an individual user. Also, Ganz et al., presented a survey of computations on the heterogeneity of data for a single user, which need not be applied to only one particular information processing technique.

Besides, the technique in [29] is adaptive to changes in the input data and handling of multimodal data in eNB without the need for domain knowledge. Cao et al. discussed game theoretic offloading mechanisms in eNodeB (eNB)with the assistance of edge. The expectation of multichannel contention environment [30] is to balance the transmission and computation cost by which it can achieve least execution cost. But the mathematical model works better for homogeneity in terms of their computation ability and their computation tasks. In our paper, we propose a mesh multi hop self-healing ESP for 5G networks.

3. MMM Architecture for Computation Offloading

To improve the coverage of multiuser, pico cells are densely deployed in hotspots areas around 20 meters' distance such as shopping malls, retail spaces, stock exchanges, train stations, etc. which can support for a maximum of 100 users at a time. Due to fast, inexpensive and vast coverage with reduced outdoor interference, we use picocells for doing computations at the time of handover. So that, it allows operators to provide excellent services at dynamic environment, improving customer perception and satisfiction. Also, the higher signal quality of picocell leads to better throughput because it allows the network to use a more spectrally efficient transmission scheme where more bits can be transmitted at the same time.

Thus, by integrating the functionalities of Edge Server in Picocells (ESP) as a multihopin mesh network provides high resilience. i.e., any sort of fault or failure in a node will not be affected or make the entire network to be down. Moreover, connecting the operators of a core network (CN) is done via backhaul links, which could reduce CAPEX and OPEX. In our paper, we aim to achieve not only faster but an effective resource and backhaul sharing of computation for multi-users with tolerable delay. Besides, we develop a distributed Ant Social based Vector optimization algorithm (ASV) for seamless communication and computation with reduction of signal overhead.

To enhance backhaul routing, we use learning technique for the design of efficient user pattern. The user pattern comprised of detailed information: 1) user behavior such as distance, age, education, culture and type of information sharing, 2) event history such as frequent site access, time and usage and 3) social affiliation. Based on this information, eNB allocate ESPs and channels for user. For efficient computation offloading.

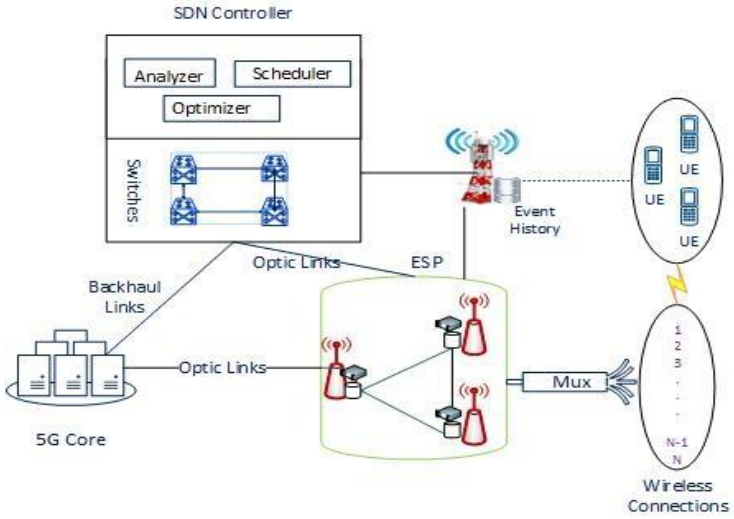

Fig.2 MMM Architecture for Computation Offloading

We handover computations of eNB to ESP with the assistance of Software Defined Networks (SDN).With the use of SDN, it is possible to segregate both data and control planes. Thereby, it brings mobile devices with high resilience, also overcome eNBs frequent feedback and signal overhead.

In our paper, to handle the mobile device with faster and effective computation, we consider eNB as a master and ESPs serves as workers. Each picocell is embedded with high-level edge cloud components namely transcoder, buffer, cache, and local database (Trained data). As shown in Figure 2, whenever mobile users are requesting for a social media, we handover the real-time demand of eNB to ESPs and so eNB can proceed with any other parallel operations for any user. Due to the non-intervention approach of eNB, energy overhead has been reduced.

We manage the entire communication and computation between eNB and ESP with the assistance of an SDN controller, which controls both eNB and ESP for seamless communication. Also, SDN monitors the inter-process communication between eNB and ESP. There is direct access between ESP to CDN till the corresponding response delivered to the user. The same content can be delivered (broadcast) through multiplexer via a frequency reusable wireless network. Thus, we introduce an architecture named Multihop Mesh Middle layer (MMM) to improve the total execution time, by making the time of computation in ESP as smaller than local. Initially, the classification of traffic can be carried out by an analyzer which segregates the user demands with an intensive and non-intensive task based on computation time or energy required. Then, scheduler can predict the user's complete usage behavior such as a type of network, type of frequent accesses of the site, timing, interest, mode of viewing (stream/online), number of accesses, and network usage using event history, which helps for prefetching of content from CDN. Internal cache of ESP is used to hold the snapshot of next task access while the current is in execution which helps to reduce latency and computation time.

Finally, optimizer supports optimal routing and backhaul conflicts through a distributed ASV algorithm. Various devices such as Apple, Samsung, Nokia etc., are escorting with different forms of connectivity, such as 4G, Wi-Fi or next-generation radio technologies. Besides, the messages of social media are small, encrypted and come in different forms of protocols such as http, xml, REST and JSON.

To manage various protocols, low latency aggregation points are required to distribute messages. In particular, the location of mobile aggregation sites can be either an indoor (e.g. hospital, airport, large corporate HQ), or an outdoor for a particular public coverage scenario (e.g. stadium, shopping mall, railway station) which will be controlled by SDN which provide radio coverage to the premises.

Thus, the integration of ESP in small cells empowers the direct delivery of locally-relevant, fast services frome NB. Besides, our MMM provides the capability to resolve multi-user computation offloading challenges.

To minimize the computational offloading problem in MMM Network formulated is given below

1. Service – oriented optimization problem
2. CN-Oriented Optimization Problem

1. Service – oriented optimization problem

In the service-oriented optimization problem to maximize the sum of all tasks' utility, and the problem can be formulated as

$$\max_{\alpha ij} \sum_{i=1}^{M} \mathrm{Y}_{task}^{i}$$

s.t.

$$\begin{aligned}
&C1: \wedge_i \le d_i - a_i, \forall i \\
&C2: \sum_{j=1}^{N} \rho ij(Tij + Gij) \le T_i, \forall_i \\
&C3: \mathrm{Y}_{task}^{i} \ge 0, \forall_i, \\
&C4: \mathrm{Y}_{CN}^{i} \ge 0, \forall_j, \\
&C5: \sum_i \rho ij \le 1 \forall_j \\
&C6: \sum_i \rho ij \le 1 \forall_i \\
&C7: \rho ij = \{0,1\}, \forall i, j.
\end{aligned} \tag{1}$$

here, C1 represents the actual waiting time constraint where Δi is the actual waiting time. Besides, C2 gives the service delay constraint. C3 ensures the minimum utility of each task and C4 restricts the minimum utility of every CN. C5 indicates that each CN can compute at most one task one time, and C6 shows that each task can be allocated to at most one CN. Finally, C7 represents the binary constraint.

2. CN-Oriented Optimization Problem:

Contrary to the service-oriented problem, the CN-oriented problem focuses on maximizing the utility of CNs, and the objective function can be represented as

$$\max_{\rho ij} \sum_{j=1}^{N} Y_{CN}^{i} \tag{2}$$

Subjects to constraints C1-C7. Obviously, the service-oriented and CN-oriented optimization problems presented above involve 0-1 programming, and the general solution for this kind of problem is branch and bound. The branch and bound is a centralized optimization method, which indicates that global information of all tasks and CNs is needed during the decision process. However, the information exchange will overload with the increase of network size. One effective way to solve this problem is the matching game, which performs in a distributed manner

For example, Image-Editing applications in Figure3, most of the stages are typically carried out in the ESP. Initially, the captured image from the mobile device can be uploaded over wireless networks. Then, image-edition procedure is completely carried out by ESP, and then the response is returned to the user from the ESP. Responses are in some user understandable media form (e.g., augmented reality). Such MMM architecture enhances the performance of overall network latency and computing time.

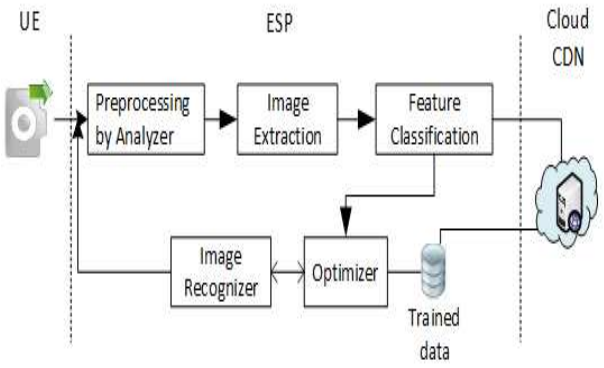

Fig. 3 An example of mobile image editing with ESP

Steps performed by ESP for image editing:

1. User uploaded the updated image captured from a mobile device

2. ESP performs preprocessing and then extracts the features from the image

3. Trained data in CDN will be fetched, and stored in the cache

4. Optimizer checks for the classified features with the trained data

5. If there is a match occurs, then ESP returns the edited image to user for approval and then updates CDN with the same. Otherwise sends the error message to user.

3.1. ASV in MMM Offloading

Generally, heuristic algorithms can be applied to a wide range of any optimization problems to yield near-optimal solutions with minimal execution time. Thus, we develop an ASV algorithm for optimal path selection. To do so, our proposed algorithm inspired the behavior of ants for seeking food between source locations to the repository. Generally, a pheromone trial is a liquid, lays down by ants during they travel. It acts as trial for further ants to proceed. By the trail values, it is easy to identify the shortest and optimal path. Since the value of trail gets evaporated over time, the only shorter path will attract all ants to deposit more pheromone values in that path.

These positive feedbacks of ants will be updated in ta bu (target) list for optimal selection. Finally, the shortest path with high pheromone value will be considered as optimal. When this artificial intelligence concept is applied to NP-hard problems, the term "trail value" or pheromone refers to the latency and its transmission cost from the user location to destination as CDN. The main idea is spreading the utility gained from finishing the computation throughout processor time it requires based on distance or latency.

For instance, consider a scenario of a single user requesting for social networking. User is a source node, and CDN is a destination. With each ESP has a list which will be updated by ants with their respective latency values to the destination. Ants will travel from sourceto destination by laying pheromone in their path with the constraint as ants won't revisit any

node twice. Now, associatively compare the collected value in the list of each ESP with the latency. Finally, the path with less latency will be chosen as optimal for processing the offload. Our proposed ASV algorithm is also applicable for multiuser diverse application needs. Also, ESPs are in the form of mesh network, which yields optimal routing. Generally, by offloading to shorter distance ESPs we can achieve less latency, power, and transmission or monetary cost.

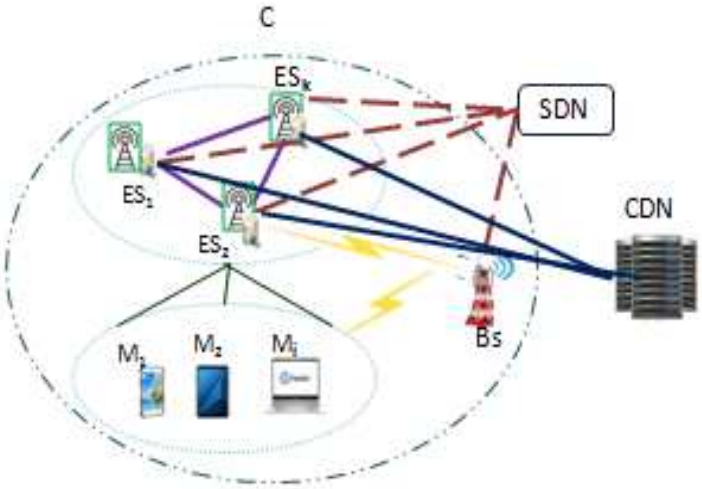

Fig. 4 System Model for computation

Hence, it avoids migration of ESP from one location to another based on user's proximity. Therefore, the ESP of a minimum latency path has been selected for offloading. Any ESP can execute and share the partial processing with its neighbor ESPs of another region more easily without transmission delay. Thus, high resilience with multi sharing is applicable readily and at least cost.

Assume a sample of an application as in Figure 4 has few numbers of clusters denoted as C_i, for $1 \geq i, k \leq n$, and each cluster is modeled as fourtuples $C_i = \{M_i, B_s, ES_k, \gamma_i\}$, where M_i, B_s denotes the number of users and base station respectively. Whereas ES_k represents a number of ESP sin a cluster, γ_i signifies the annotation of the component, indicating whether components can be executed as locally or on ESP. A mobile device of k user is modeled as eight tuples with $M_k = \{Z_{in}, Z_{out}, N, \alpha, P_x, P_{idle}, P_y, P_z\}$, where Z_{in}, Z_{out} denoted as size of

computation for input and output data, N represents maximum capacity of workload (maximum workload assigned which measured in Million Instructions), α denotes the demand of the user, P_x,P_y denotes the power consumption at local computation and communication by the mobile device, P_{idle}, P_z represents the power consumption at idle time and display of the device.

3.1.1. Local Computing

If γ_i= 0, then the component would be executed locally in a mobile device and thus the energy consumption E and latency L can be calculated using

$$E^{loc}_i = Z_{in} (T^{loc}_i P_x) + P_z \quad (1)$$

$$L^{loc}_i = T^{loc}_i + \xi \quad (2)$$

where T^{loc}_idenotes time taken for local computing, ξis a constant related to hardware architecture, and we have

$$P_x = \alpha / N.$$

3.1.2. ESP Computing

If computation energy of edge server for a component is less than E^{loc}i, then we have γi= 1, and so the component could be offloaded to ESP for computation. When the delay sensitive component is offloaded to ESP, then the overhead data rate due to the downlink data transmission T^{ov}ofM users via wireless channel can be represented as

$$T^{ov}_i = \log_2 (1 + t_n . g_{n,s} / P_0) \quad (3)$$

where, t_n, $g_{n,s}$ denotes transmission power of user and channel gain between user n and base station s, $P_0$is the power of spectrum density.

Let the mobile device with the computational ability of user as $C_u$and base station as C_b, which is quantified by the total number of CPU cycles per second. Then the execution latency $L^{ESP}$for the task can be computed as

$$L^{ESP}_i = T^{trans}_i * C_{ui}/C_{bi} \quad (4)$$

where C_{ui}, C_{bi} denotes the computation capability of user and base station via a wireless channel, and $T^{trans}$is the transmission delay, so we have $T^{trans}_i = Z_{in} / T^{ov}_i$.

Let n_i is dynamic power consumption of a CPU cycle, and we have $n_i = Z_{in} * C_u$, then the energy consumption at over head E^{ov} can be represented as

$$E^{ov}_i = t_n T^{trans}_i + \delta \quad (5)$$

where δ is constant energy due to a mobile device which could hold the channel for a while even after transmission of data. Thus, the computational energyof ESP with latency will be expressed as

$$E^{ESP}i = n_i (P_y + P_{idle} + P_z) + T^{ov}_i (P_y + P_{idle} + P_z) E^{ov}_i / B_i \quad (6)$$

where B_i is the spectral bandwidth, T^{ov}_i, E^{ov}_i is the over head time and energy for obtaining and releasing ESP. Computations can be done either locally or in ESP. In this paper, our proposed ASV is formulated as a mixed integer, and non-convex problem along with the objective of maximizing energy savings and latency is expressed as follows

$$F(x) = \max \sum_{i=1}^{n} Ei.\gamma i \quad (7)$$

$$\text{Such that } \sum_{i=1}^{n} \gamma i (T^{trans}_i + T^{ov}_i) + T^{loc}_i \gamma i \leq \alpha \quad (8)$$

$$\text{Along with } \sum_{i=1}^{n} \gamma i L^{ESP}_i \leq lag \quad (9)$$

where α is the total local execution time of the mobile application, lag represents the maximum tolerable delay for a sensitive application, E_i is the difference between a component local computation energy and edge computation energy. Thus, we have the choices of offloading decisions, and the computational complexity of M mobile users and T computation tasks for ESP is O (n^{MT}).

In our proposed, computations can be done by ESP, due to which long packet transmission and long thread of feedback would be reduced. Which in turn reduce the issues of battery drain, monetary cost, and latency, particularly low-speed connection. Our objective is minimizing overall power consumption with tolerable delay. In particular, delay is the

integration factor of communication, computation and network delay. Due to the integrated functionalities of ESP with a mesh network, 20% of above-mentioned delays will be reduced. In Demand Vector Classification (DVC) algorithm, based on the user's demand, traffic can be segregated as real time and non-real time. DVC is an effective technique which provides optimal and straightforward solutions under all classification techniques.

Algorithm 1: DVC algorithm

Input : cluster of user requests
Output : Decision Vector γ

```
Start demand_classification
Initialize threshold, n
for each request n do
compute Eloc , EESP
if Eloc> EESP&& EESP≤ threshold
γi = 1
Else
γi = 0
end if
end for
Stop demand_classification
```

Thus, analyzer will concurrently evaluate the energy consumed by the mobile device for the current task. Also, the max value in the energy level is assumed to be the threshold and idle value when mobile is inactive as 0.5. In order to take an effective decision, equation 1 and 6 evaluates the energy of the current task to be executed in device. Generally, handover will be associatively managed and processed. If the evaluated value is higher, then eNB will offload the task to ESPs, which is nearer to the user's location with the current value as $\gamma i = 1$; otherwise, eNB allows to perform computations locally with the value as $\gamma i = 0$.

The proposed algorithm has three phases

- Initiation phase
- Discovery phase and
- Handover phase

In the initiation phase, classification of real-time and non-real time computations can be taken place based on user demands. Discovery phase is also known as decision phase, which decide whether to offload computation to ESP or allow device to execute locally. Once the demand is identified, then discovery phase has to determine better ESP {n=n1...nx} with fair location based on the event history of a profiler which is represented as z. Since ESPs are in mesh type, it enables multi sharing and resilience with the entire satisfaction of users with n nodes and m ants.

The optimizer uses Dynamic Offloading Decision (DOD) algorithm to determine an optimal path from the possible nodes, by which user will get service either from ESP or locally. Also, a scheduler will get updated with the current status of ESP computations, which will be monitored by eNB. The path selection is purely based on proximity (distance) of users. Also, the forwarding nodes resolve their decision and routing conflicts with tolerant delay. Through proposed ASV based path selection algorithm, long communication delay due to a poor communication link will be effectively managed. Hence predicted pheromone (trial) could be updated in the optimal list.

Algorithm 2: DOD algorithm

Input: cluster of user requests, γi
Output: path ∂

Start path_selection
Initialize nodes as n, computations as m
for i= 1 to n do
if $\gamma i = 1$
Get z and predict user behavior
for j = 1: m do
if Eij> 0
Compute T^{trans}, T^{ov}, L^{ov}, T^{loc}
if min $\{T^{trans}ij + T^{ov}ij) + T^{loc}ij \} \leq \alpha$
$\partial = \partial \cup \{i\}$
Execute ASV algorithm
else
$\gamma i = 0$
Compute L^{loc}
end if
$\partial = \partial \cup \{j\}$

17 end if
18 end for
19 end if
20 end for
21 updated path ∂
22 Stop path_selection

Generally, each eNB maintains the details of event history of each cluster and their corresponding nodes. Through which the social relations of a user (user behavior) can be predicted, and so opportunistic network with close proximity for computation can be determined. Besides, an effective handover and communication could be taken place between eNB to ESP. Our proposed is a multihop mesh type; single node failure will not make the entire network to be down (resilience). In addition, the handover phase will receive requests of user, that will be segregated and assigned (hand over) to ESP so that communication between users and computations can be done by ESP directly, which in turn reduce delays. The non-intervention process of eNB is termed as handover, and all the activities between eNB and ESP can be controlled through SDN.

Algorithm 3: ASV algorithm

Input: cluster of user requests, path ∂
Output: Optimal path S

Start optimal_path_detection
Initialize t computations, n nodes and m ants for each
partial_solution=∂
while $t \neq 0$ do
for i= 1to n
for j=1 to m
Find available vertices v
Initialize j^{th} ant in the i^{th} computations
if available (v) > 1
compute L^{ESP}
partial_solution= min (L^{ESP})
else
Ant die
Return
endif
end for
sort partial_solution in ascending order
move to the next available node

```
19      if partial_solution ≤ lag
20      S = partial_solution
21      Add the solution and report to the Scheduler
22      for each ant that completed a solution do
23      update S for each edge that the ant versed
24      end for
25      endif
26      return S
27      end for
28      Stop optimal_path_detection
```

The ASV algorithm, initially make the optimal path S as an empty set. If the path has at least one node then traverse for t computations, till an optimal solution S is found successfully. If there is no next node, then the path of the current node is optimal. If there is more than one next node to process, then compute latency of each node. Finally, compare the latency values of each node in the path with other nodes. If the latency value is less than and equal to the specified lag time (i.e., 20ms), then mark that path as optimal S and assign the minimum latency value as the best trail value. The trial value is updated automatically like the ant tabu table repository process. If a node performs partial computations and it fails, remaining computations will be carried out by other ESPs with the assistance of mesh network to perform successful task completion.

3.2. Open Research Challenges

In our paper, we consider primary challenges of generalized architecture and optimal offloading. Our results show that it would be an alternative, reliable and an effective offloading for both static and dynamic environments than other existing works.

3.2.1. Architecture and Deployment

The specification of how to perform computation offloading and the deployment location of edge server within the network are not mentioned. So, investigating the optimal site selection is a problem in edge server. Most of the research articles theoretically addresses edge server concept, which can then be analyzed through the modern learning techniques along with

novel algorithms. However, the critical issues are to be validated by numerical analysis or by simulations, and also essential to validate key principles and findings through simulation under more complex realistic situations and scenarios. Besides, there is no clear picture about the type of simulators and the place where it will be best suited. At the same time, massive trials and further experiments in emulated networks or real networks are mandatory to move edge server concept closer to reality.

Thus, choosing the optimal placement by deploying picocells with the integration of edge (ESP) requires computational resource provisioning along with the deployment budget. Also, determining the ESP density to cope up with their demands isalso closely related to the infrastructure deployment cost. Caching at the ESP named Computation Caching (CC), supports sharing of multi-user to enhance the user experience and also handles the management of load efficiently by providing results to end users directly without fetching their required tasks beforehand. Unlike content caching, CC presents several new challenges.

First is a diverse type of computing tasks, which depends on the computing environment. Though few contents of frequently used data are cacheable for reuse by other devices, data belongs to personal computing is not cacheable and so it must often be executed in real time. Second, it is not practical to build user patterns locally at each server; instead, learning the intelligent techniques and methods over extensive sets can provide a broader view on the user patterns.

3.2.2. Offloading Decision

In green networking, the energy consumption at the ESP has to be taken into account during the decision. Most of the research articles dealing with the offloading decision with the assumption of static scenarios, i.e., the location of UEs are fixed before and during the offloading. Besides, the consumption of energy due to transmission will drastically change the offloaded data. Also, channel quality drops due to low movement or fading. Hence, it is

459 necessary to propose new advanced methods such as data analytics, machine learning
460 techniques, etc. for efficient offloading decisions. For instance, collaborating various
461 prediction techniques on the UEs mobility and channel quality during the offloading provides
462 better estimation of offloading for varying conditions. A major challenge in offload decision
463 belongs to the consideration of backhaul alleviation between the ESP and ability to reflect
464 varying load and parameters during the offloading decision. This paradigm motivates for
465 rethinking and reshaping research effort from single downlink to the mixed downlink and
466 uplink in the future.

467 *3.2.3. Mobility Management*

468 Mobility is a critical issue to ensure service continuity for dynamic mobile users. Moreover,
469 efficient management in the utilization of communication and computation services are also a
470 challenging issue from the service orchestration perspective. In order to manage the recent
471 trends of learning techniques optimally, analytical mechanisms can be used by ESP to
472 estimate, predict and manage the movement of users through personal preference
473 information, which in turn improves the user experience. Besides, to achieve better user
474 computation experience, proposed offloading techniques consider mobility-aware scheduling
475 policies at the ESP. This approach introduces a set of interesting research problems including
476 mobility-aware online prefetching of user computation data, server scheduling, and fault-
477 tolerance computation.

478 *3.2.4. Security and Privacy*

479 The distributed deployment feature leads to frequent site attacks hence requires stringent
480 security policies/implementation of trust management systems. In order to gain access to the
481 platform, derive the information needed regarding user proximity and network analytics.
482 Besides, service providers would like to attain user information to update their services; then
483 there is a great challenge to the development of privacy protection mechanisms.

4. Experimental Results

As a proof of concept, we evaluate our framework in Cloud Sim with 3 ESPs and components such as a cache for prefetch of trained data and high-end Xen hypervisor server for computation. The proposed ESP mesh can support around 250 to 300 nodes. For instance, we are taking image editing process with 250 nodes, in which each server of ESP with 1Gb RAM, 1.6 TB storage capacity, an average bandwidth of 20MHz and an average image size of 1000 KB are taken. Our evaluation metrics from Table I are the real time parameters based on load per cluster. The objective is to show how the amount of offloaded tasks from the mobile devices will impact the total power consumption and the incurred delay in the system. In this paper, ESPs are in the form of a mesh topology, to reduce power consumption and increase coverage. Thus, the computations for a multi-user can be opportunistically offloaded to ESPs through wifi. Besides, the tasks during handover can be controlled through SDN.

Table 1: Simulation Parameters

Description	Value
Number of pico cells	30
Bandwidth of pico cells	20 MHz
Distance between pico to User	50m
Distance between pico to ESP	1 Km
Computation capacity of ESP	30MFLOPS
Storage capacity of ESP	100MB
Coverage diameter per ESP	100m
Number of devices per ESP	[10-30]
Number of computations per device	4
Transmission power of user device	0.5w
Computation capacity per device	[1-100] MIPS
Storage capacity per device	[1-100] Mb
Inter edge round time trip	10 ms
Simulation Iterations	20
Latency	(15, 20, 70) ms
Interval Time	100 ms
Execution time	30 ms
Resource requests	[2, 5, 7] MFLOPS
Packet size	[280,410,700] KB

The effectiveness of the proposed scheme is evaluated through the experiments with varying numbers of requests. Simulation in CloudSim shows that ESP is serving most of the requests. In order to verify the effectiveness, we consider a scenario of a 5G heterogeneous ultra-dense network, where a set of 3 pico cells with a distance of 50m to users are distributed with a bandwidth of 20MHz. Each small cell is surrounded by three users with the transmission power of 0.5W, capacity of both computation and storage are 100MIPS and 100Mbs respectively. Also each user could be accomplished with at most four types of computations for 100 iterations.

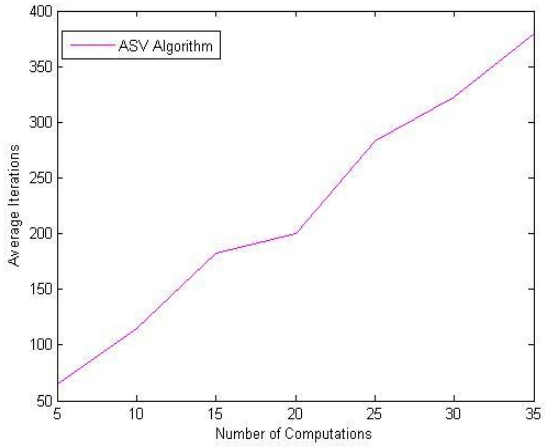

Fig. 5 Number of computations vs Average Iterations

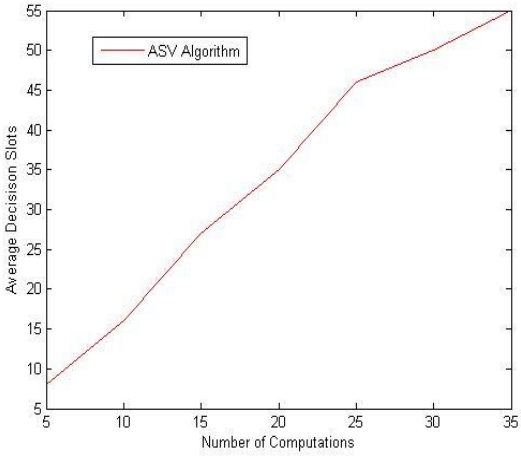

Fig. 6 Average Decision slots for convergence with different number of computations

Regarding the kind of computations, the size of an input data is set to be 250KB to 800 KB. Hence, we conclude that lengthier size of the input packet, then longer the computations, with high energy cost. In particular, our proposed DOD offloading scheme exhibits parallel associative computations with reduced latency and energy. Multi sharing of the core cloud is showing an increase with increasing number of requests to support the ubiquitous coverage for the services. Figure 5 and Figure 6 shows the consumption of energy and average decisions of offload sing ASV with respect to number of computations. In our scenario, we are taken 3 ESPs and three users per ESP with simultaneous access of 4 tasks. So, totally 36 computations can be done in ESP simultaneously with varied iterations. It is clear that the computations and its iterations are inversely proportional to each other. ESPs can perform computations simultaneously; thus energy consumption is less compared to mobile device execution.

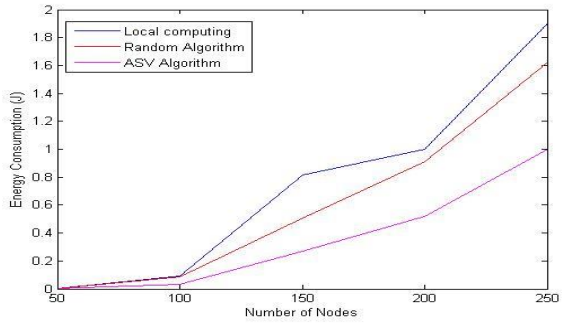

Fig. 7 Comparison of energy consumption using ASV with existing mechanisms

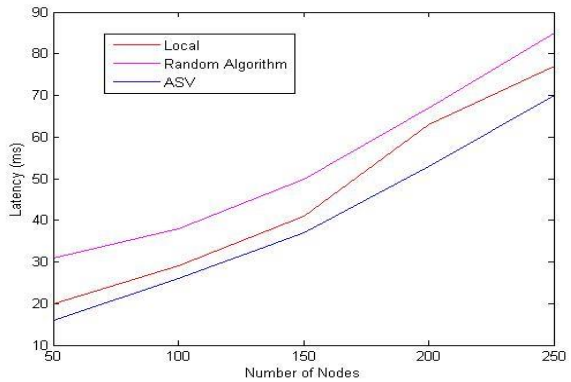

Fig. 8 Comparison of latency constraints in proposed ASV with existing mechanisms

Figure 7 and Figure 8 depicts energy consumption and latency of intensive computations that are carried out by ASV is less compared to the computations done by local and random algorithm. It is clear that the computation for up to 100 nodes of varied type's results in reduced latency by 2%, than with local and random computing; if it exceeds 120 nodes then local and random computing cannot effectively produce results, there by affects the Quality of Experience (QoE). But our proposed ASV provides a faster response with improved QoE. We can see from Figure 9 that local computations will drain the user's battery much faster than ESP. Since computations can be done in ESP in an effective manner, the battery can be prolonged. The user will get satisfied by the faster intensive computations with retaining power in their device.

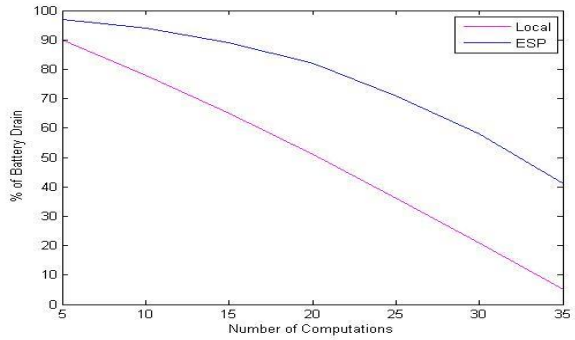

Fig.9 Battery drain in ASV based on the number of computations

Besides, increasing the percentage of requests by the mobile devices in turn will reduce total power consumption in contrast to the scenarios that need the 5G core to fulfill. However, application and content providers are challenged by the latency of the network when connecting to the CDN, which has to be resolved by today's emerging intelligent mechanism. Thus, in order to obtain a tolerable latency, combination of proximity and intelligence has been embedded in our proposed ESP.

5. Conclusions

Thus, the proposed MMM architecture for computation offloading provides more computational resources for faster execution. Moreover, it corroborates the need of ESP for

efficient computation offloading, where the distance of information-propagations are tens to hundreds of meters with higher computation capability. Our solution for effective offloading with ESP is unique, is not only bringing computation resources closer to mobile user's, but also decouples the problem of identifying user needs through the use of prediction and learning technique. Hence, our proposed MMM architecture efficiently handles both the offloading and handover. In addition, it efficiently sustains with the optimal routing through intelligent ASV algorithm. In particular, ESP is an emerging mechanism that can be best suited in emergency situations, battlefield surveillance, retail places, stadium, shopping malls, high-speed mobile video applications access to transport etc. Thus, our proposed offloading scheme is a self-healing network that could be intelligently managed by licensed reusable channels. As a future work, we plan to create a test bed for offloading scheme, and then to expand this concept to other areas such as trust management among others.

Acknowledgment

D. AdhimugaSivasakthi and Gunasekaran Raja gratefully acknowledge for the support of NGNLab, Department of Computer Technology, Anna University, MIT Campus, Chennai.

Compliance with Ethical Standards

Funding: No funding has been received.

Disclosure of potential conflicts of interest: Authors declare that they have no conflict of interest.

Ethical approval: This article does not contain any studies with human participants performed by any of the authors.

Consent to participate: Not applicable

Consent for publication: Not applicable

Code availability: Not applicable

Declaration of interests: The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper

References

[1] Golkarifard, M., Yang, J., Movaghar, A., & Hui, P. (2017). A Hitchhiker's guide to computation offloading: Opinions from practitioners. *IEEE Communications Magazine*, *55*(7), 193-199.

[2] Khoda, M. E., Razzaque, M. A., Almogren, A., Hassan, M. M., Alamri, A., & Alelaiwi, A. (2016). Efficient computation offloading decision in mobile cloud computing over 5G network. *Mobile Networks and Applications*, *21*(5), 777-792.

[3] Mach, P., & Becvar, Z. (2017). Mobile edge computing: A survey on architecture and computation offloading. *IEEE Communications Surveys & Tutorials*, *19*(3), 1628-1656.

[4] Wang, S., Zhang, X., Zhang, Y., Wang, L., Yang, J., & Wang, W. (2017). A survey on mobile edge networks: Convergence of computing, caching and communications. *Ieee Access*, *5*, 6757-6779.

[5] Mao, Y., You, C., Zhang, J., Huang, K., & Letaief, K. B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, *19*(4), 2322-2358.

[6] http://blogs.valvesoftware.com/abrash/latency-the-sine-qua-non-of-ar-and-vr/.

[7] https://www.bayshorenetworks.com/blog/breaking-down-idc-top-10-iot-predictions-for-2017

[8] Mao, Y., Zhang, J., & Letaief, K. B. (2016). Dynamic computation offloading for mobile-edge computing with energy harvesting devices. *IEEE Journal on Selected Areas in Communications*, *34*(12), 3590-3605.

[9] Bhattacharya, A., & De, P. (2017). A survey of adaptation techniques in computation offloading. *Journal of Network and Computer Applications*, *78*, 97-115.

[10] Chen, X., Jiao, L., Li, W., & Fu, X. (2015). Efficient multi-user computation offloading for mobile-edge cloud computing. *IEEE/ACM Transactions on Networking*, *24*(5), 2795-2808.

[11] Kottursamy, K., Raja, G., Padmanabhan, J., & Srinivasan, V. (2017). An improved database synchronization mechanism for mobile data using software-defined networking control. *Computers & Electrical Engineering*, *57*, 93-103.

[12] Kottursamy, K., Raja, G., & Saranya, K. (2016). A data activity-based server-side cache replacement for mobile devices. In *Artificial Intelligence and Evolutionary Computations in Engineering Systems* (pp. 579-589). Springer, New Delhi.

[13] Elgazzar, K., Martin, P., & Hassanein, H. S. (2014). Cloud-assisted computation offloading to support mobile services. *IEEE Transactions on Cloud Computing*, *4*(3), 279-292.

[14] Liu, W., Gong, W., Du, W., & Zou, C. (2017, February). Computation offloading strategy for multi user mobile data streaming applications. In *2017 19th International Conference on Advanced Communication Technology (ICACT)* (pp. 111-120). IEEE.

[15] Saguna and Intel, "Using mobile edge computing to improve mobile network performance and profitability," White paper, 2016

[16] Hu, Y. C., Patel, M., Sabella, D., Sprecher, N., & Young, V. (2015). Mobile edge computing—A key technology towards 5G. *ETSI white paper*, *11*(11), 1-16.

[17] Bharathi U., Mercy Faustina J., Elavarasi B & Gunasekaran R. (2017)Energy Efficient Data Offloading Mechanism,' in proc. IEEE International Conference on Telecommunication, Power Analysis and Computing Techniques(ICTPACT), Apr-6-8,

[18] Kumar, K., Liu, J., Lu, Y. H., & Bhargava, B. (2013). A survey of computation offloading for mobile systems. *Mobile networks and Applications*, *18*(1), 129-140.

[19] Jiao, L., Friedman, R., Fu, X., Secci, S., Smoreda, Z., & Tschofenig, H. (2013, July). Cloud-based computation offloading for mobile devices: State of the art, challenges and opportunities. In *2013 Future Network & Mobile Summit* (pp. 1-11). IEEE.

[20] Wang, S., Zhou, A., Hsu, C. H., Xiao, X., & Yang, F. (2015). Provision of data-intensive services through energy-and QoS-aware virtual machine placement in national cloud data centers. *IEEE Transactions on Emerging Topics in Computing*, *4*(2), 290-300.

[21] Lin, Y. D., Chu, E. T. H., Lai, Y. C., & Huang, T. J. (2013). Time-and-energy-aware computation offloading in handheld devices to coprocessors and clouds. *IEEE Systems Journal*, *9*(2), 393-405.

[22] Ragona, C., Granelli, F., Fiandrino, C., Kliazovich, D., & Bouvry, P. (2015, December). Energy-efficient computation offloading for wearable devices and smartphones in mobile cloud computing. In *2015 IEEE Global Communications Conference (GLOBECOM)* (pp. 1-6). IEEE.

[23] Goudarzi, M., Movahedi, Z., & Nazari, M. (2016, July). Efficient multisite computation offloading for mobile cloud computing. In *2016 Intl IEEE Conferences on Ubiquitous Intelligence & Computing, Advanced and Trusted Computing, Scalable Computing and Communications, Cloud and Big Data Computing, Internet of People, and Smart World Congress (UIC/ATC/ScalCom/CBDCom/IoP/SmartWorld)* (pp. 1131-1138). IEEE.

[24] Mukherjee, A., De, D., & Roy, D. G. (2016). A power and latency aware cloudlet selection strategy for multi-cloudlet environment. *IEEE Transactions on Cloud Computing*, *7*(1), 141-154.

643 [25] Zhang, K., Mao, Y., Leng, S., Zhao, Q., Li, L., Peng, X., ... & Zhang, Y. (2016).
644 Energy-efficient offloading for mobile edge computing in 5G heterogeneous
645 networks. *IEEE access*, *4*, 5896-5907.
646 [26] Yang, L., Cao, J., Wang, Z., & Wu, W. (2017, August). Network aware multi-user
647 computation partitioning in mobile edge clouds. In *2017 46th International Conference*
648 *on Parallel Processing (ICPP)* (pp. 302-311). IEEE.
649 [27] Schneider, M., Rambach, J., & Stricker, D. (2017, March). Augmented reality based on
650 edge computing using the example of remote live support. In *2017 IEEE International*
651 *Conference on Industrial Technology (ICIT)* (pp. 1277-1282). IEEE.
652 [28] Soh, H., & Demiris, Y. (2014). Incrementally learning objects by touch: Online
653 discriminative and generative models for tactile-based recognition. *IEEE transactions*
654 *on haptics*, *7*(4), 512-525.
655 [29] Ganz, F., Puschmann, D., Barnaghi, P., & Carrez, F. (2015). A practical evaluation of
656 information processing and abstraction techniques for the internet of things. *IEEE*
657 *Internet of Things journal*, *2*(4), 340-354.
658 [30] Cao, H., & Cai, J. (2017). Distributed multiuser computation offloading for cloudlet-
659 based mobile cloud computing: A game-theoretic machine learning approach. *IEEE*
660 *Transactions on Vehicular Technology*, *67*(1), 752-764.

Figures

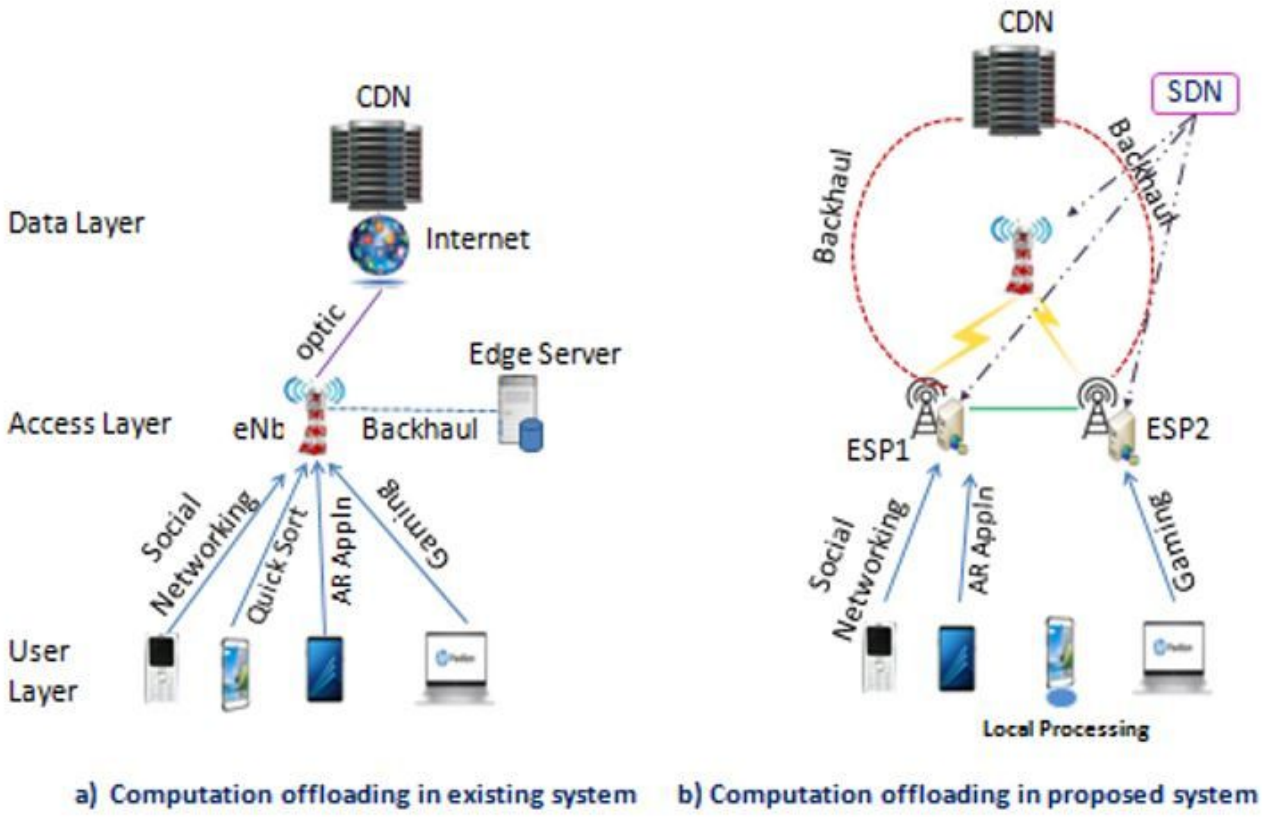

Figure 1

Computation offloading Scenario a) Existing and b) Proposed Framework

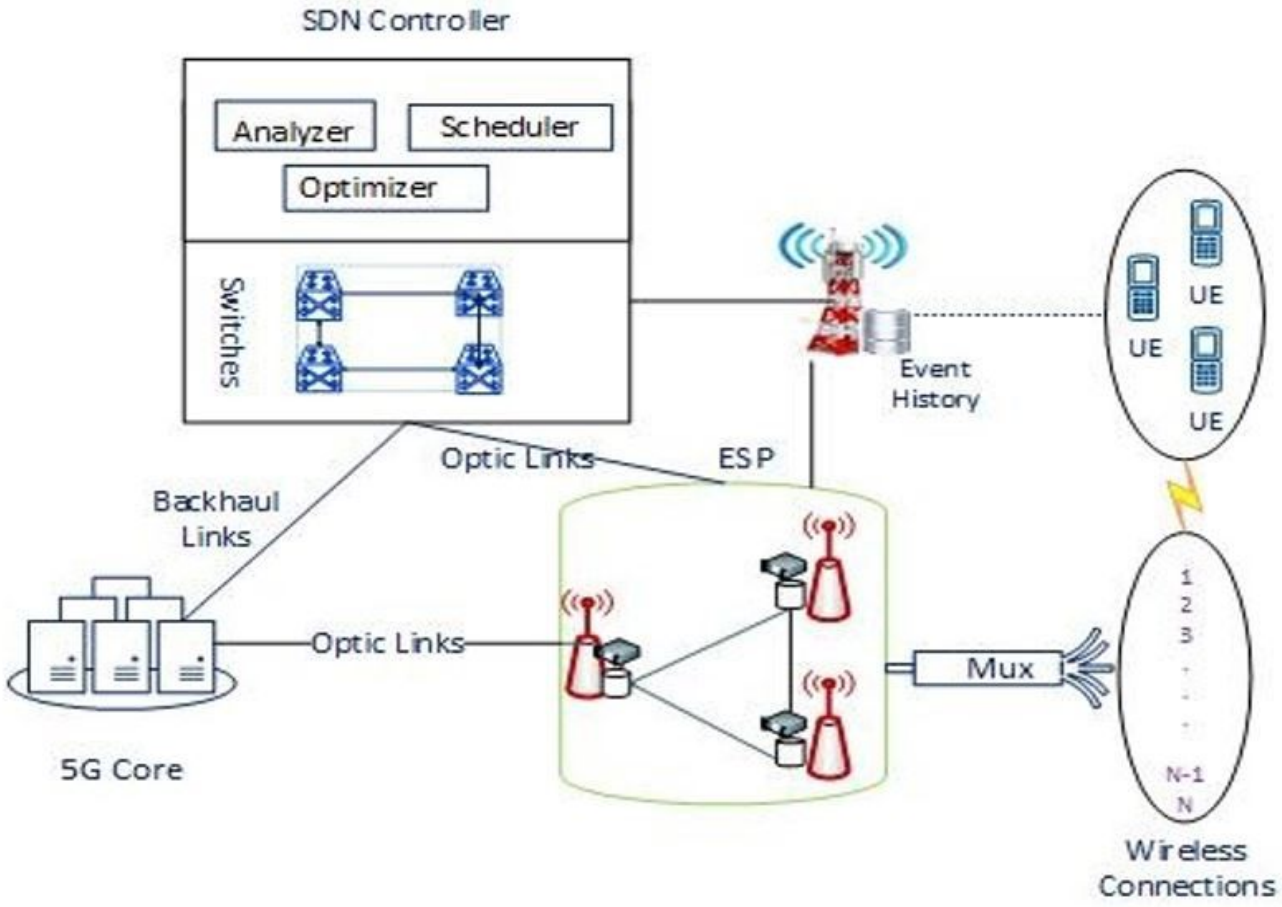

Figure 2

MMM Architecture for Computation Offloading

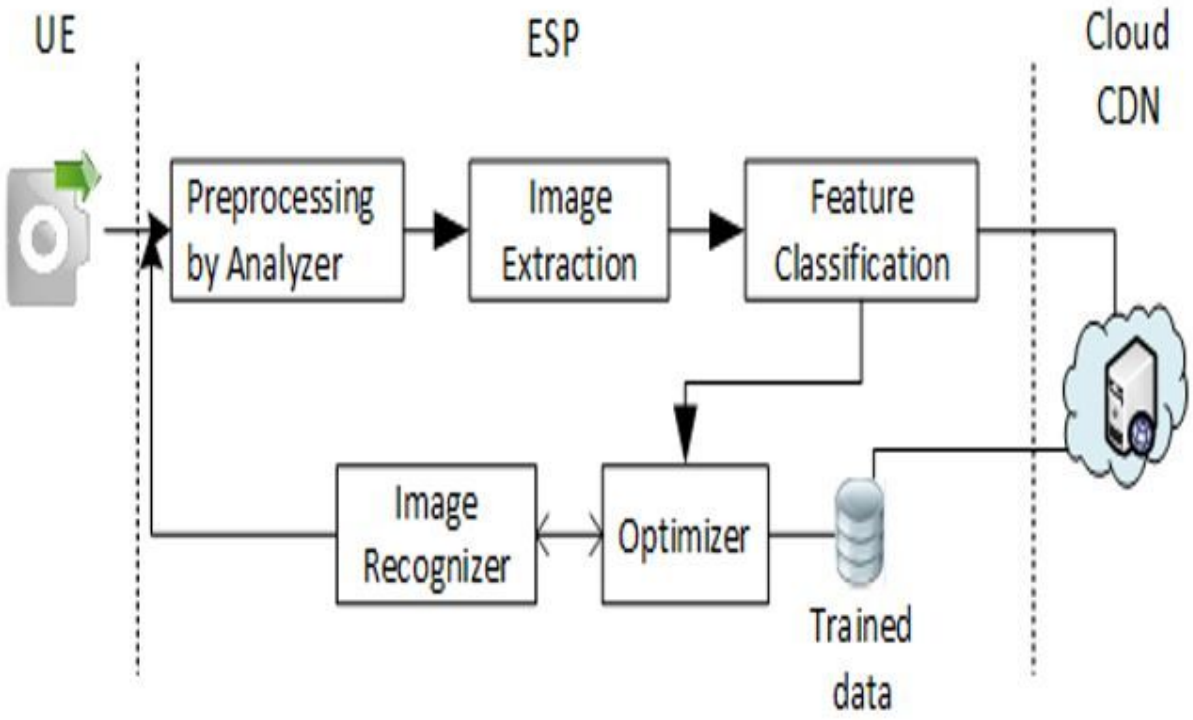

Figure 3

An example of mobile image editing with ESP

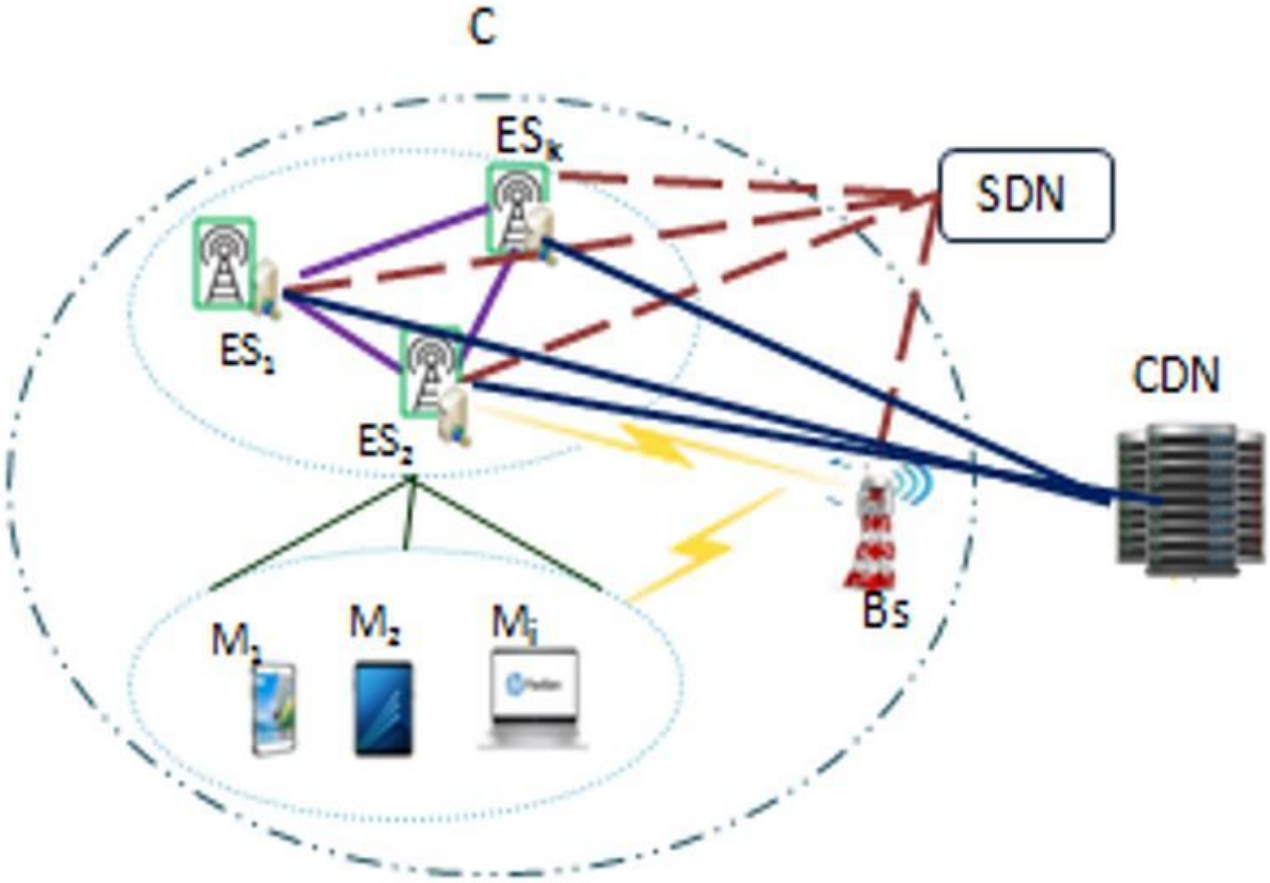

Figure 4

System Model for computation

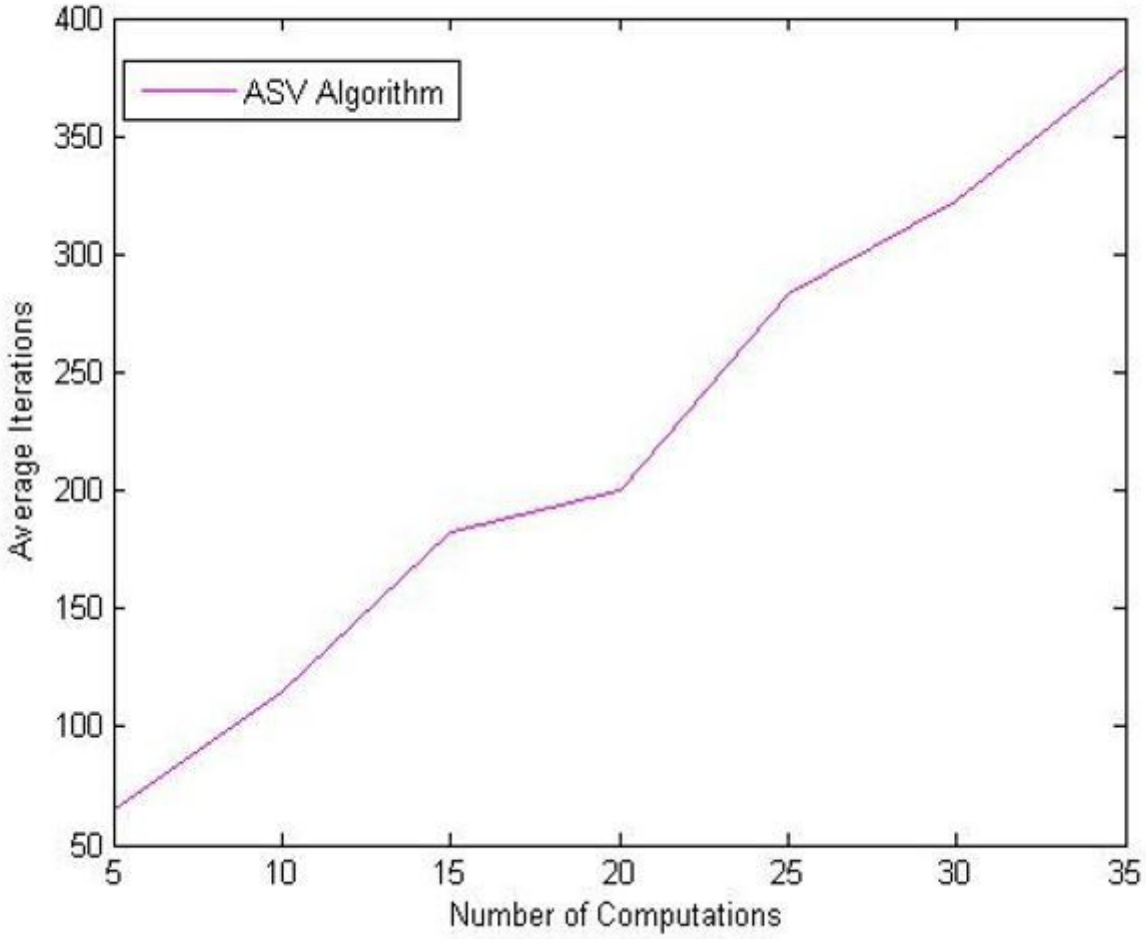

Figure 5

Number of computations vs Average Iterations

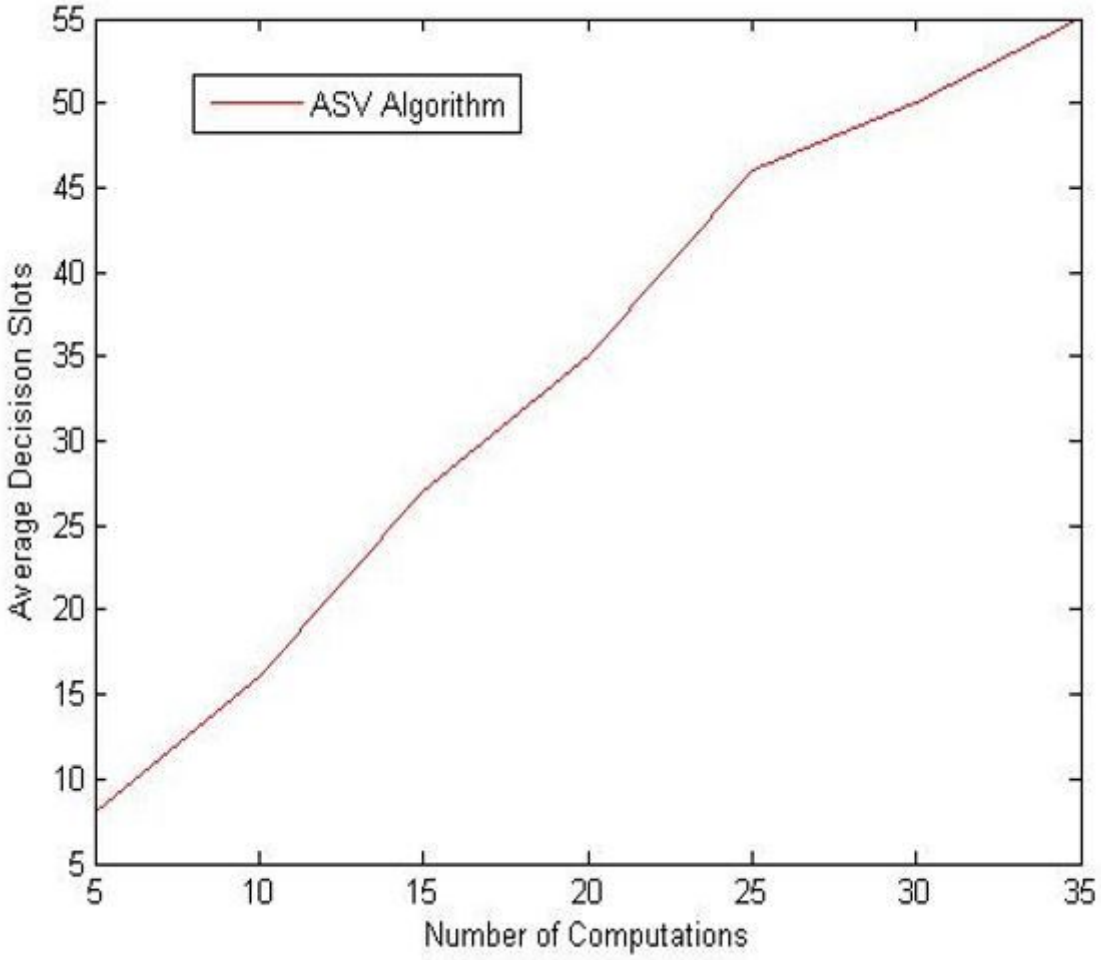

Figure 6

Average Decision slots for convergence with different number of computations

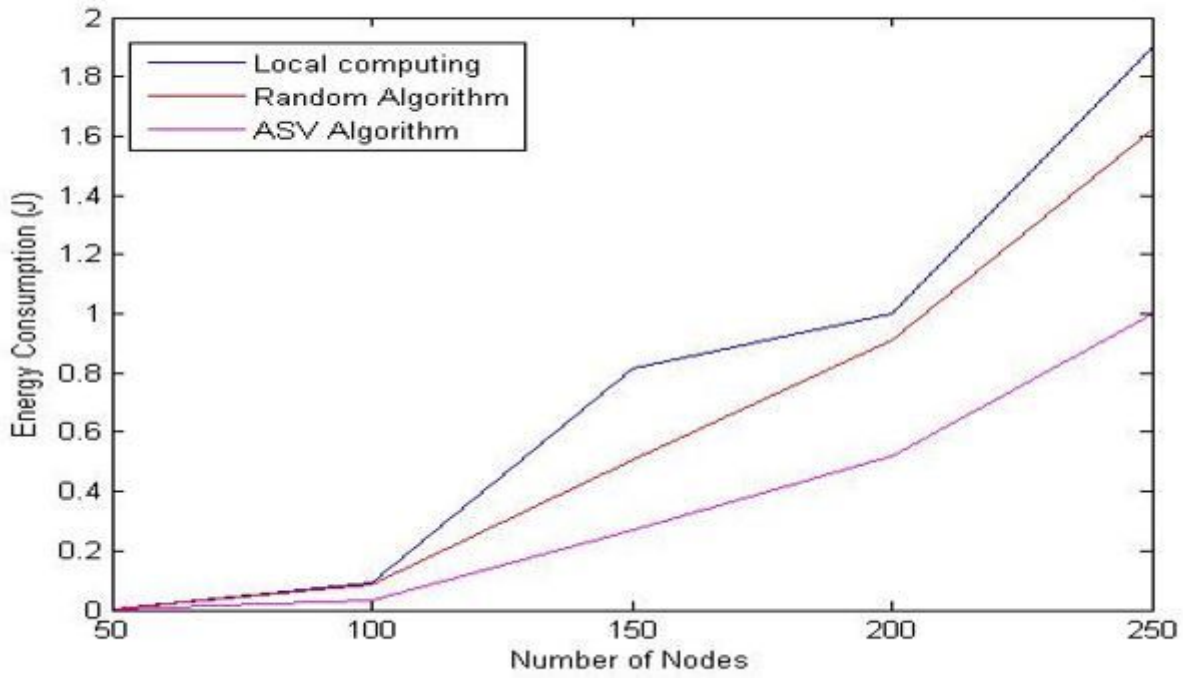

Figure 7

Comparison of energy consumption using ASV with existing mechanisms

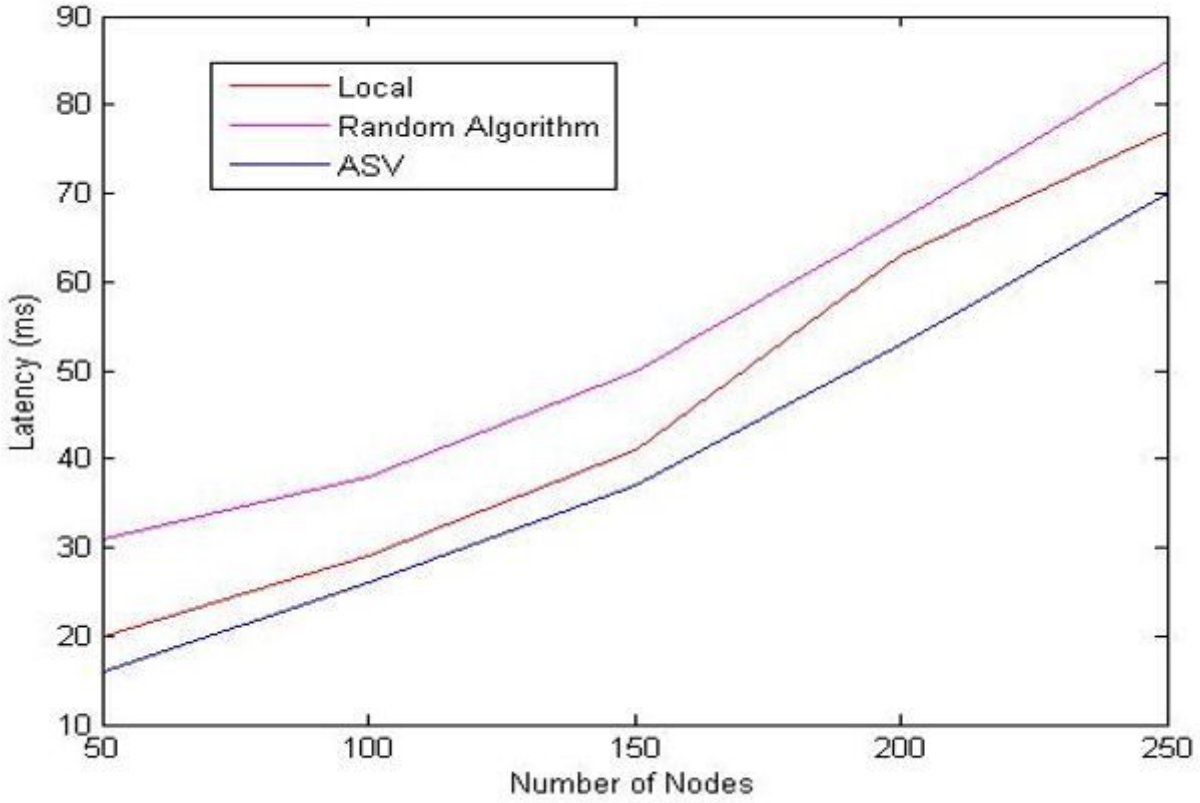

Figure 8

Comparison of latency constraints in proposed ASV with existing mechanisms

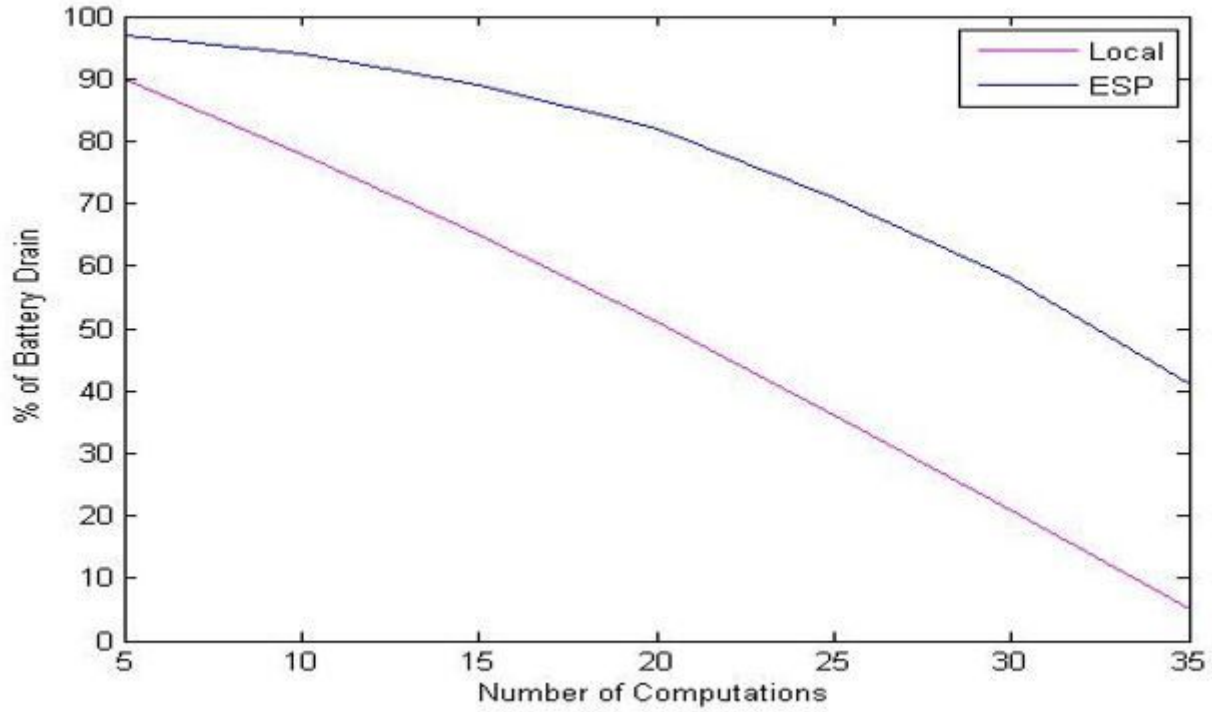

Figure 9

Battery drain in ASV based on the number of computations