Cooperative Task Offloading Using Cloud And Edge Computing

Dr. S. Lilly Sheeba, Associate Professor, SRM Institute of Science and Technology, Dakshin. D, Lingaraj Gopalakrishnan, Dheepak. G, Students, SRM Institute of Science and Technology

Abstract - Recently, we've seen a paradigm shift in mobile computing, moving from centralised cloud computing to mobile edge computing, from mobile Internet access to the new fifth-generation cellular network connectivities are the most noteworthy developments for cloud computing (MEC). MEC's main characteristic would be that it pushes cloud applications, network management, and memory onto network edges (e.g., area network and network devices) so that resource-constrained mobile devices won't be running computation-intensive applications. A single variable antenna design has the potential to slash the overall network delay and cellular energy consumption by a considerable amount, enabling the 5G ambition to become a reality. MEC's provided a strong incentive for industry and academic exploration of the conduct of extensive research & development efforts. One result of MEC research is that fuses wireless networking and computing to merge various technologies, which have led to the development of designs ranging from processing offloading methods to networking architectures. MEC has brought a detailed treatment of computer and field resources to the latest innovations under intense light, and ongoing research attention. In this paper, the debate of which emphasises the importance of continued advancement in these areas. To better understand these problems and investigate solutions, we also study MEC device implementation, with cache enabled MEC, solve mobility management issues, and assess potential problems, and look into avenues of further privacy improvements. Such progress as we make in the development of MEC will be needed for MEC to transition to real life. Now that we've learned the definition of simple expansive, we'll broaden on a few more popular MEC concepts and MEC efforts.

Keywords - Mobile edge computing, mobile cloud computing, computation offloading, resource management, Collaborative cloud.

I. INTRODUCTION

An ongoing shift in computing models from the central cloud to distributed among individual devices is rapidly manifesting itself in the coming future: fueled by 5G and Internet of Things, computing paradigms are evolving in today's mobile computing (MEC). MEC's primary feature is that it pushes mobile computing, network control, and storage to the network edges (e.g., base stations and access points), allowing for computation-intensive and latency-sensitive applications to run in cloud servers or MECs which have more processing resources.

Mobile edge computing (MEC) brings the potential to enable applications requiring low latency with high bandwidth utilisation. Even so, achieving the required low latency can be difficult, especially in a “Internet of Everything” which is becoming a reality thanks to the advanced IoT systems. Furthermore, while energy efficiency is a critical consideration when developing MEC solutions, a lack of adequate coordination among greedy smart devices for offloading can result in higher energy costs and latencies. Energy-efficient offloading in MEC has been the subject of a few studies. Via joint resource allocation, the energy demand of MEC-based mobile applications was reduced. For fifth-generation (5G) heterogeneous networks, an energy-efficient computing offloading method was also proposed.

With all of the above in mind, MEC offers numerous capabilities, including mobile resource optimization that places resources closer to the end users, the data prior to sending (or some of it, extracting) for (or during) processing at the cloud boundary. The MEC and 'fogulstaf's Concept of Fog Computing' expand on the term "fog computing" in some ways, though both are utilised by some firms and in their work. While MEC, as a whole, in opposition to cloud computing is providing processing and storage capability to the edge nodes, is primarily focused on bringing computing and resource augmentation to the node level, can have the focus on extending the current capabilities and interface layers between BSs with new functions of splitting and splitting to layer. Fog computing is most commonly comes in to gateways that are under the control of organisations, but MEC infrastructure is provided and owned by network operators.

We provide a detailed overview of MEC technology along with evolving MEC applications in Smart devices in this article. Then we concentrate on using MEC for handheld smart devices, and we suggest a low-latency
environmental MEC offloading scheme. This article's primary elements could be summed up as follows. We look at the most recent MEC applications for a variety of IoT applications, including their different features, technical criteria, and strategic approaches. We design a cross-layer computing resource sharing mechanism between edge computing servers and propose a mobility-aware hierarchical edge computing architecture. Finally, we propose an incentive-based cooperative task offloading strategy that maximises the service provider's utility while lowering smart device energy demand and task completion time.

II. LITERATURE SURVEY

[1] Jinke Ren and all have used collaborative cloud and edge computing. Thinking about working with the cloud Sharing and co-ordinating files through cloud computing involves the ability to upload documents to a cloud where they are stored and available to multiple users. Cloud collaboration capabilities allow for users to add and edit content and modifications, which makes it possible for others to upload, comment, and work with the documents. More businesses are opting for cloud collaboration has been found to be far more popular and preferred in the past few years.

[2] All of Mr. Molina's colleagues use the MCC server scheduling. Think of MCC as giving your mobile devices the capability to push their workloads to the cloud, which enhances the varieties of apps you can use on your devices, and simultaneously enhancing their features. There are a number of ways in which researchers have recommended the use of MCC (may differ significantly from the use of the others), each with its own MCC architecture. Therefore, a new approach is required for the efficient use of idle device capacities in which a decrease in response time does not compromise the service levels.

[3] K.Kumar and everyone have considered using binary offloading. More efficient use of resources means that computation is offloaded to the user or it's own channels, rather than using the user's or the system's channel resources when available. Additionally, beamforming and MIMO techniques can be used to free up-load less energy for cellular networks with high bandwidth requirements. For a long response time-dependent feature, it is better to offload computation from the MEC rather than to the user's end system with generous provisioning of large computation capacity.

[4] S. E. Mahmoodi and all have used partial offloading. Partial offloading allows components partitioning to be modified, which may or may not hold all of the information. It is possible to achieve a larger energy saving by offloading by giving computationally intensive tasks to MEC (or time-consuming) consuming ones and less demanding ones to MEC servers, which results in a more effective implementation latency with regard to the work done with fewer sub-tasks. Graph theory is a useful for identifying off-loaded processes, in that it provides a way to precisely defined layout for the scheduling according to the dependency network.

[5] Y. Li and all have used co-operative computing which improves sharing results of all edge nodes. The extra-large amount of computational resources available when conducting distributed processing eliminates network congestion also boosts the capacity and brings information processing into the reach of all parties. As I have previously described, this project can be carried out using a cooperative peer-to-peer approach. To name a few, short-range transmission using D2D techniques, as well as the ability to provide computational resources and toilance and sharphs of result provide.

[6] T. Zhao and all have used server selection to get successful offloading probability. When you're looking at MEC systems, one of the significant challenges is providing multiple computation tasks and allowing for the diverse requirements of servers to work together. Wherever there is latency sensitive computation, the solution is to offload both the problem and less critical functions to cloud servers and hard to deploy functions to the edge servers.

[7] M. S. Elbamby and all have used server co-operation to minimize latency. When MEC-servers servers work together, the resource and computation utilisation are enhanced for the good of both. More importantly, it distributes the computation load evenly across the networks to reduce computation latency while improving resource utilisation. Additionally, the server design should also take into consideration temporal and spatial arrivals and computational capabilities, as well as revenue varying on the servers and channels over time.
[8] M-H. Chen and all have used computation migration to reduce costs. MEC systems have become effective when it comes to mobility management, which was previously unachievable, through the use of computation other things being equal, migrating or nonessential workers depends on the costs of migration, how far apart the users and servers are, and how much computation power the group has. Generally, it is preferred that a long- that the use computer movement (a long distance) away from its original MEC migration destination should be used to move the computation to nearby destinations.

[9] J. Liu and all have used stochastic model to minimize latency. In the stochastic task model, tasks' arrival and departure times may be correlated. This allows designers to design computation offloading policies that are tuned to maximise the temporal arrival and temporal departure of tasks. Another very important consideration is to keep the task and MEC server utilisation within bounds at the user and server by doing offloading at the user level.

[10] C. You and all have used joint radio-and-computational resource allocation. With finite radio and computational resources, a multi-user MEC will suffice e. This is an example of system-level objective like reducing the overall energy consumption, and improving channel gains, where the system may or should focus on prioritising the clients with high channel utilisation and low local compute energy consumption, who could potentially help save more energy. At a certain maximum, this will reduce the system’s income because there will be too many offloading users that impede on communications and computations among the offloading users.

III. EMERGING MEC APPLICATIONS

A great number of connected smart devices often necessitate the transmission of information. Relating to the computing capacity on the MEC servers makes it possible for them to sense and regulate interactions. Thus, according to MEC, the main characteristics are as follows -

1) Proximity - MEC is best used in relation to key features in IoT data generated by closeness to the intelligent devices.

2) Low Latency - Shifting large amounts of data off the network to a mobile Edge server reduces application load time, decreases the potential for network congestion, and makes for a better experience for the end user.

3) Semantic Sense of Position and Platform - When your device connects to the MEC, the information that you give it will help the system discover where the other MEC devices are in the local area network and help the system determine where the other devices are. Applications on the devices have the capability to keep track of the current network states by using the real-time information, and thus they can predict what devices may need additional maintenance.

4) Flexible Deployment - For agencies that use high- and non- third-party software developers, MEC is widely utilised for novel mission-critical projects, which are deployed by both fixed-mobile operators and both legal and illegal third-party programmes.

5) Synergy of Diverse Resources - To handle a large number of computing task demands, both cloud computing and edge computing must be used in tandem. In order to meet the diverse performance requirements of IoT applications, it is also important to synergize heterogeneous resources such as storage, caching, and communication.

6) Mobility Awareness - The freedom to switch is a major aspect of today's widespread mobile devices. Tasks can be outsourced to many MEC servers as devices progress. The necessity for a continuous task offloading service requires a computing infrastructure that is seamlessly incorporated.

7) Protection of security and privacy - When you are dealing with a MEC in an IoT network, your systems are vulnerable to DDoS (denial of service) attacks such as DDoS (distributed denial of service (denial of service). In addition, the disclosure of confidential data that occurs during deploying applications may be disclosed to unauthorised parties. The data centre security and privacy balance of an arduous task is that
computer network trying to achieve is difficult to maintain while allowing its cost effectiveness to be protected at the same time.

8) **Respond in Real-Time** - The need for a quick and immediate response is the key for Cloud devices. Designing a computation offloading scheme that will be able to meet low latency requirements is not just a matter of giving each resource a static time window is not only part of its effort and its also an strategic approach.

We will describe six distinct approaches that can be taken with regard to each of MEC, then talk about the different possibilities and new complexities for tackling those challenges.

![Figure 1: MEC Applications](image)

### 3.1 MEC For Intelligent Transportation

Connected to transportation networks and enable smart systems to become more environmentally aware and interconnected, with information processing technologies, which forms the basis for a new idea of an interactive transportation platform.

There are many good applications for a smart cities, but in this one the self driving system appears to be particularly attractive. Providing service can aid the organisation of path-oriented tasks, which can allow them to be offloaded to RSU or other nodes that can provide more up-to-date information.

Specifically, fast mobility, which is found only in vehicles, poses complex problems for smart transport systems because of its inherent character, is something that is unprecedented in its performance. It is possible that during fast vehicle movement for vehicles to come into contact with many wireless access points that have been deployed in 5G networks, preferably not; due to the limited specific machine capacity of MEC servers or to the significant computational prerequisites for vehicle-related tasks, the operation of the vehicle should not happen on a single machine until it is fully outside the vehicle. As a result, efficient task absorption optimizations and collective resource allocation among all these MEC servers are needed.
Other things being equal, a severe task-case workloads may result in loss of effectiveness for client machines communicating with the MEC. Vehicle-to-Vehicle and Vehicle-to-Infrastructure are both considered to be communication in vehicle networks. When vehicles are moving in a V2 route, handover occurs between two roadside points with the transmission of files, this handover expansion technique will restrict the utility task offloading solution’s performance.

Offloading tasks in Vehicle-to-Vehicle mode is an acceptable and viable solution to the problem. Vehicular platform can has a computing ability. It is thus possible to have a vehicle-independent mobile task-free cloud infrastructure, through which a number of vehicles can work together. More predictability among vehicles' and less complicated vehicle-to-computing is anticipated, given the fact that the vehicle speed is much slower than the vehicle-to-RSU relation. There are handover costs in task offloading, as long as there are no other people involved in the process. The overall higher connectivity and lower latency of V2V delay allow for the task of sending files from vehicles to vehicles to be handled uniquely by V2V technology.

3.2 Smart Grid Using MEC -

A smart grid is a new electrical energy grid system which can produce and deliver power to its customers more efficiently, versatile, and reliably. To monitor and maintain the grid condition, a huge number of internet-connected sensing devices are being used. Such devices are constantly producing large quantities of data.

MEC enables data collection and analysis at cloud based servers near the power supply chain, lowering distribution costs and increasing energy management efficiency. However, before implementing MEC for smart grid, a variety of issues must be resolved, which will be discussed briefly next.

Offloading data processing activities to MEC servers can be very burdensome, even for MEC servers, unless they are implemented on a massive scale, due to the grid's large scale and the huge and varied types of sensory data in various subsystems of the grid. Furthermore, owing to the intrinsic stochasticity and associated uncertainties introduced into the smart grid by renewable sources of energy, efficient energy use necessitates more complex control systems as well as more computational resources. Furthermore, as electric vehicles become more common, its charging / discharging processes will have an effect on the grid. Electric vehicles and the grid's energy sharing process must also be tracked and controlled.

Using the computing power of electric vehicles to exchange functions to MEC servers is really a promising approach to alleviate the burden. Since the charging / discharging by electric vehicles takes a long time, the vehicles are left in facilities or charging points during this time. The fixed connection assists in the control and scheduling of vehicular computational power. In addition, by using power line communication method, the function file can be sent to the vehicles through charging cables, potentially lowering offloading costs and latency.

3.3 MEC for Agriculture and Rural Development -

Iot advances have had a significant impact on modern farming, where various sensors and automated systems are used to replace vast quantities of human labour. Environmental factors such as climate, humidity, soil moisture, and plant growth status are collected during the crop cultivation process. These factors' continuous monitoring data must be evaluated in order to provide real-time input on appropriate cultivating adjustments. The data collected can be sent to and analysed in network edge servers nearby farm fields using MEC.

The climate in a particular region involves a systematic pattern. Artificial intelligence (AI) built into MEC servers to know and use trends will increase accuracy rate and assist farmers in better planning agriculture sector. Furthermore, we find a close connection between the natural crops of neighboring regions. For example, rainfall water in one area can carry moisture to nearby areas. Multiple MEC servers in several neighbouring areas may form an agricultural data processing cooperative to share the research outputs in response to such observations. This allows for more precise and efficient farm planting while also enhancing MEC server resource utilisation.
3.4 Intelligent Healthcare Using MEC -

Smart connected gadgets are now expected to play an active role in enabling medical care at any time and from any place. Temperature, blood pressure, heart rate, or other measurements, as well as diagnosis and treatment recommendations, would be performed by such systems in both a reactive and constructive manner. Such services necessitate broad data collection, cross-domain cooperation, and in-depth data analysis, both of which are dependent upon robust transmission of data as well as processing capabilities. In this situation, resource-constrained smart devices could become bottlenecks, resulting in data processing delays and a degradation of the end-user experience. The use of MEC servers to aggregate and evaluate gathered health data in close vicinity to consumers opens the possibility for resolving the issues listed above.

It is important to spread health statistics analysis to several MEC servers in order to use MEC for medical management in a timely manner (e.g., daily). This necessarily involves server collaboration in order to process medical data from different areas and communities at the same time. In this sense, data protection is a major concern, making it possible to disclose only authorized details when it relates to confidential information such as wellness information. Health data that is sensitive to individual privacy can only be exchanged with pre-arranged approvals.

MEC revolutionises patient care for physicians by allowing them to quickly access patient data and use it to generate diagnosis reviews from a health data platform at the edges. MEC technology can increase care efficiency significantly, but effective treatment choices for emergency medical conditions, such as with a patient in the Intensive care unit or a patient with heart problems, cannot wait too long. To improve MEC functionality while guaranteeing that patients are handled in live time, more sophisticated methods are needed. The combination of MEC plus cache systems is one effective option to minimise care latency. When a patient is admitted at a hospital, his medical information should be collected and cached on MEC servers that would be included in the recovery process. In an emergency, the cached data can be provided to MEC processing without having to wait for data to be acquired.

3.5 MEC for Smart Buildings

A Smart Framework is a structure which uses sensors,cams and actuators to collect environmental building information and control the operations of the building automatically in which temperature, light, gas and humidity can be controlled in accordance with customer preference. The collected information is shared and processed locally by implementing MEC servers in smart buildings, improving the efficiency of building control and adapting the decision-making for new situations.

Other than Edge servers coordination within buildings, smart buildings may integrate a range of functions, including protection, if MEC servers can link to and coordinate with control systems or servers outside. The coordination of MEC servers for building fire alarms and a fire department's dispatch centre, for example, enhances firefighting performance. The link between MEC servers in building automation and a power system control centre allows the systems to dynamically change power usage in the buildings based on real-time grid conditions and power prices, culminating in savings on the buildings' energy bills. Since the control centers might be beyond the building's access network, long-distance communication system may have a significant impact on collaboration amongst MEC servers in the building and remotely controlled centres. As a result, in order to enable such applications, it is important to collectively layout interaction and collaboration amongst the servers.

3.6 MEC for Smart Retail

Smartly managed Internet of Things (IoT) and network devices have significantly altered the business strategy, and opened the door to the concept of intelligent retail. We were able to utilise the collected information on the cargo/shopping habits and user preferences to the MEC servers in a large retail environment to fine-tune our customer service. An analysis service provided by MEC will additionally help retail stores provide for rapid responsiveness to client needs and to market conditions.
Retailers are constantly collect massive data: There are literally billions of customers, and billions of records, which can result in a huge data volume. Since a rapid accumulation of massive amounts of large datasets is so much, it’s more appealing for companies to investigate customer behaviour and constantly fine-tune their strategy according to what has already been learned.

New challenges for IOT (“everything from daily activities to highly diverse electronic devices, sensor-based tasks are poised to grow”) as well as numerous mobile applications arise, with IOT a bigger issue than ever, data-driven tasks as well as exponential amounts of data generated from these various devices is predicted to proliferate. By having a better understanding of how the relationships and patterns and how the data operates, we may be able to perform these operations better, rapidly and massively complicated data has shifted from being seen as simply abundant to being employed with data statistics to be analyzed with a powerful tool. This refers to efforts to incorporating the MEC with predictive analytics, where plenty of study is expected to help the enormous developments in IoT and other new technology endeavours.

IV. ENERGY EFFICIENT - LOW LATENCY STRATEGY IN MEC SYSTEMS

The computational power of mobile devices is limited, so we offer an offloading method that uses only a minimum amount. Highly dynamic topology is proposed to help manage the tasks in intelligent transportation systems. We think this mechanism we have for cross-layer resource sharing between Mobile nodes can also relieve the stress on individual MEC systems as well as IoT applications when it comes to heavy computation.

4.1 The Agility Conscious Hierarchical Approach

In the suggested hierarchical MEC system that has been suggested. We define two-layered MEC resource allocation, with green energy powering the computation servers. These virtual machines have a higher processing energy savings than mobile devices. The first layer assets are n MEC servers, each with a computation capacity of \( \{\phi_{1,n}, \phi_{2,n}, \ldots, \phi_{n,n}\} \) where \( n = \{1,2,3, \ldots\} \). Such MEC servers are installed on \( n \) base stations (BSs) that are spaced along a lane. A backup computing server creates a second layer of the resources, sharing computing tasks with MEC servers whenever their computing resources are inadequate to satisfy the requirements of smart devices. MEC servers must pay costs \( c \) to the remote cloud server for using its component system resources when they want to transfer their functions to it.

At the reference source of the lane, there can be \( N \) smart mobile devices. Both of the devices move at the same pace \( p \). An individual smart device has a specific computing mission, such as machine learning or algorithmic processing which is formulated as \( \phi_{m,n} = \{\phi_{m,n,0}, \phi_{m,n,1}, \ldots, \phi_{m,n,k}\} \), \( \phi_{m,n} \ (1,2,3, \ldots) \). The scale of the function data input and the amount of computational resources needed, respectively, are \( \phi_{m,n,0} \) and \( \phi_{m,n,1} \). For such a mission, the maximum delay is \( \phi_{m,n}^{max} \).

Each smart device's task can be completed remotely on a MEC server using work offloading or locally using the device’s own computational power. Let \( \phi_{m,n,j} \) represent the device’s preference, where \( \phi_{m,n,j} = \phi_{m,n} \) represents that the device prefers to transfer process \( \phi_{m,n,j} \) to MEC machine \( k \), and \( \phi_{m,n,0} = \phi_{m,n,j} \) implies that the client prefers not to transfer function \( \phi_{m,n,j} \) to edge servers \( k \). The task execution time is provided as \( \phi_{m,n,j}^{s} \) when system \( i \) selects localized computing. When the system transfers function \( \phi_{m,n,j} \) to MEC server \( k \), however, the overall time cost \( \phi_{m,n,j}^{s} \) is divided into 3 parts, specifically the device’s access time to the MEC server \( k \) the time it takes for input data to be transmitted and for the calculation to be completed. Consequently, \( \phi_{m,n,j}^{s} \) can be seen as following

\[
\phi_{m,n,j}^{s} = \sum_{j=1}^{s} \frac{\phi_{m,n,j}^{s}}{p} + \frac{\phi_{m,n,j}^{s}}{TR_{k}} + \phi_{m,n,j}^{s} / C_{l,k}
\]

where \( \phi_{m,n,j}^{s} \) is the road section duration capable of being interpreted by BS \( j \). \( \phi_{m,n,j}^{s} \) is the device data transfer rate for particular device communicating with BS \( k \) and \( \phi_{m,n,j}^{s} \) is the capacity of computational resources partitioned by MEC server \( k \) for computing mission \( \phi_{m,n,j}^{s} \).
4.2 Formulation Of The Problem: Ideal Energy Conducive Offloading With Minimal Latency

The functionality of system \(i\) acquired by transferring process \(\square\square\square\square\square\) to MEC edge server \(k\), which is primarily determined by the task completion delay, energy conservation improvement, and offloading transaction costs. Since smart device users are logical, they prefer offloading to targeted MEC servers to optimise their resources while remaining only within latency constraints. The optimization algorithm for system \(i\) is given set \(\{\square\square\square\square\square\}\), which is the cost to use a component system resources of MEC server \(k\) is

\[
\text{max}_{\{co,di,k,CC,i\}} = \sum_{k=1}^{n} dp_{i,k} (\lambda(tek_{i,0} - tek_{i,k}) + \xi (eg_{i,0} - er_{i,k}) - y_{k} CC_{i,k})
\]

\[
s.t \sum_{i=1}^{x} dp_{i,k} tek_{i,k} \leq d_{i}^{max}
\]

\[
\sum_{i=1}^{x} dp_{i,k} CC_{i,k} \leq g_{k}^{max} + g^{b},
\]

\[
dp_{i,k} = \{0,1\}, \sum_{k=1}^{n} dp_{i,k} \leq 1
\]

here \(\square\square\square\square\square\) and \(\square\square\square\square\square\) become the time and energy required by the smart device to complete the task locally, respectively. The energy cost of transferring work \(\square\square\square\square\square\) to server \(k\) is \(\square\square\square\square\square\). where \(\lambda\) and \(\xi\) are coefficients. The sum of computing resources purchased by MEC server \(k\) from the cloud server is \(\square\square\square\).

Since each MEC server has a limited computing power, devices can need to compete for task offloading target servers and server resources, making a non-cooperative game an intelligent way to simulate the server selection procedure. Since the mission offloading task between both the systems is a concave bi game, the game has a Balance point. The MEC servers, as offloading network operators, tend to optimise their usefulness by selling virtualized resources to devices. Since the selling price has such a strong influence on smart device offloading decisions, pricing strategy is one of the MEC servers’ incentive strategies for attracting more customers. In addition to price adjustments, a resource-depleted MEC server can purchase additional computational power from the backup server in order to satisfy more tasks at a higher utility.

As a result, in the anti-cooperative match between MEC servers during the offloading process, server \(k\) strategy set can be written as \(\{\square\square\square\square\square\}\). For MEC server \(k\), the utility objective function can be written as

\[
\text{max}_{x_{k},y_{k}} Z_{k} = x_{k} \sum_{i=1}^{N} CC_{i,k} - \beta_{k} \min \left( \sum_{i=1}^{N} CC_{i,k}, g_{k}^{max} \right) - y g^{b}_{k}
\]

\[
s.t \ x_{k} > 0, g^{b}_{k} \geq 0
\]

here, \(\beta_{k}\) is the expense of running server \(k\), which provides a measure of virtual resources. The match seen between MEC servers has a Balance point because the value function \(\square\square\square\square\square\) is persistent and quasi-concave in view of \(\square\square\square\square\square\) and \(\square\square\square\square\square\).

4.3 Solutions: A Stackelberg Game Approach

The computing resource price of MEC servers determines the offloading strategies chosen by smart devices during the task offloading process. Furthermore, as service providers, the MEC servers are indirectly coupled by resource prices in the service rivalry with one another. As a response, the price could be used as a link between devices and MEC servers to connect these two non-cooperative games.
A Stackelberg game is an enticing method to model and fix the two-layer offloading issue, provided that the devices' offloading decisions are in answer to the prices offered by the MEC servers. Since the individual games between smart devices and MEC servers have Nash equilibrium, it is evident that the Stackelberg game has a Stackelberg equilibrium. The Stackelberg equilibrium is a solution to the optimal offloading problem that maximises the utility of MEC servers while lowering task delay and power consumption. This problem can be solved using a heuristic iterative approach. The smart devices allow the correct reply to the prices announced by the MEC servers in each iteration, and the MEC servers draw their optimal policies based on the devices' known responses. The iteration comes to a halt when no plan changes adjust the utilities on both sides.

V. ILLUSTRATIVE RESULTS

We evaluate the performances of the proposed offloading schemes in this section. We imagine a situation in which mobile smart devices are installed on vehicles travelling at 120 km/h. A computation task is assigned to each unit. The sum of computational resources needed by such activities is a random number between (20 and 70).

Figure 2 illustrates the MEC servers' total usefulness using various offloading systems. Those that use the optimum pricing scheme have greater benefits than those who use the fixed price scheme. The utility obtained by the servers in the fixed price system is purely proportional to the number of capital they use at their disposal. The systems with optimal price, as calculated by Stackelberg analysis of the game, can, on the other hand, dynamically change resource price. If there are more work offloading queries, the price may be raised to obtain more utility. The usefulness of the appropriate price scheme without remote backup cannot increase as the range of phones grows, particularly when the range of phones exceeds 200.

When faced with a large amount of job tasks, the MEC servers' necessary resources under the task delay constraint can be depleted. No more offloading services are provided without the backup server. As a result, the utility's maximum bound has been reached. The two-layer resource model with backup server, on the other hand,
might be able to change this upper limit and gain high utility even with high offloading demands, proving the efficacy of our proposed hierarchical offloading system.

Figure 3 depicts the energy usage of tasks completed using various schemes. When compared to the cost of a scheme without any offloading service, the 3 offloading schemes save a considerable amount of energy by executing tasks on server with increased energy efficiency, particularly when there are a large number of devices. Furthermore, as shown in Fig. 4, our suggested mobility-aware optimal offloading schemes consume less energy than the picture solution, especially for the backup server scheme. The explanation for this is that the matching approach only considers computing and communication capabilities in the devices' current position, and ignores the fact that MEC servers that are actually outside of this scope can be reached by the devices as they travel. Our suggested mobility-aware offloading schemes, on the other hand, effectively leverage these remote tools while consuming less energy during the offloading process. Although the backup server offers little value when the number of devices is 80, the decrease in energy becomes more pronounced when the number of devices rises in our proposed two offloading schemes. This indicates that the proposed hierarchical offloading architecture, in conjunction with a backup server, will reduce the MEC servers' computing resource requirements and make the edge computing network more energy efficient.
Figure 4. Average task latency reduction rates with different system speeds are compared.

Figure 4 represents the typical latency reduction rates of tasks using our planned mobility-aware offloading schemes at different system speeds. Faster speeds mean less time invested on system travel and reaching MEC servers located further away. Under the constraints of delay, our proposed system allows activities to be offloaded to more distant servers at a lower cost. As a consequence, further MEC servers can be used, lowering execution time latency. In any of the cases with different system numbers, the rates go up as the speed increases in Fig. 5. With more machines, the impact of speed on latency reduction becomes more important, as the MEC servers’ resources near the starting point are exhausted.

VI. CONCLUSION

We look at MEC technologies in the sense of developing IoT applications in this article. Both critical features and underlying challenges, as well as methods for dealing with them, are examined and presented. Then we concentrate on smart mobile device computation offloading. We present a mobility-aware hierarchical architecture for mobile smart device computation offloading, as well as an energy-efficient offloading system with low task latency. Numerical results were used to demonstrate the efficacy of our suggested approach.

Despite recent promising work in the field of MEC-assisted IoT systems, comprehensive and powerful computational requests, as well as diverse and evolving offloading, remain significant challenges. For example, a question that has yet to be answered is how to increase task execution efficiency by leveraging the importance of applications. One potential research path when using IoT in a hostile environment is maintaining user privacy and defending MEC servers from attack. Furthermore, with developments in artificial intelligence (AI), the issue of how to incorporate AI into the MEC offloading process is a high priority.
REFERENCES


