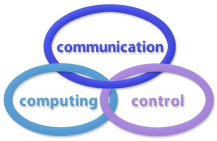
INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 19, Issue: 3, Month: June, Year: 2024
Article Number: 6143, https://doi.org/10.15837/ijccc.2024.3.6143




CCC Publications


# Contextual Information Based Scheduling for Service Migration in Mobile Edge Computing

Sanchari Saha, Iyappan Perumal, V.R.Niveditha, Mohamed Abbas, I.Manimozhi, C.Rohith Bhat

**Sanchari Saha**
CMR Institute of Technology, Bengaluru, India,
sanchari.s@cmrit.ac.in

**Iyappan Perumal***
School of Computer Science and Engineering (SCOPE)
Vellore Institute of Technology, Vellore, Tamil Nadu, 632014, India.
*Corresponding author: iyappan.perumal@vit.ac.in

**V.R. Niveditha**
Department of Computer Science and Engineering,
Sathyabama Institute of Science and Technology,
Chennai, 600119, Tamil Nadu , India,
niveditha.cse@sathyabama.ac.in

**Mohamed Abbas**
Electrical Engineering Department,
College of Engineering,
King Khalid University, Abha 61421, Saudi Arabia.
mabas@kku.edu.sa

**I. Manimozhi**
Professor and Head,
Department of Computer science and Engineering
East Point College of Engineering and Technology, Bangalore, India
drmanimozhi.i@eastpoint.ac.in

**C. Rohith Bhat**
Professor, Department of Computer Science and Engineering
Saveetha School of Engineering,
Saveetha Institute of Medical and Technical Sciences, (SIMATS), Chennai, Tamilnadu, India
rohithbhatc.sse@saveetha.com

**Abstract**

Mobile Edge Computing (MEC) is a distributed computing paradigm that delivers processing and data storage capabilities closer to the network edge, which is adjacent to mobile consumers and devices. MEC lowers latency, reduces data transmission times, and improves overall performance for mobile apps by relocating computing resources to the network's edge. But, due to higher average load and longer elapsed time, modern end devices such as smartphones and tablets cause major load challenges in mobile computing networks. Furthermore, if smartphones cause unpredictable traffic patterns, it becomes impossible to model and forecast the nature of communication. Such confusing traffic figures are caused not just by bursty Internet traffic, but also by multitasking operating systems that allow users to swiftly switch between active apps. Mobility of users and end devices impose a difficult challenge to provide continuous services in mobile computing. In this paper, this issue is addressed using the Contextual Information Based Scheduling (CIBS) technique to optimally allocate resources and provide seamless service to the users. The proposed method is implemented with NS-3, an open-source network simulator that provides a comprehensive set of modules for Mobile Edge Computing (MEC) simulations, including mobility modelling support. The experimental results show that CIBS offers migration time of 97512ms, delay time of 372115ms, execution time of 1061328ms and downtime of 98715ms. The results are compared with the existing Mobility-Aware Joint Task Scheduling (MATS) approach. The obtained results show that CIBS outperforms MATS with regard to migration time, latency, execution time and downtime.

**Keywords:** MEC, Scheduling, Migration, Priority, Contextual Information.

# 1 Introduction

In present days as a result of the expansion of mobile devices as well as mobile networks and Internet of Things (IoT) devices, numerous services with a requirement of low latency and high speed are emerging. Cloud computing services, such as Dropbox, Google Cloud Platform, and Amazon, have evolved to suit these requirements. Furthermore, with the advancement of numerous studies and technologies related to cloud computing, cloud computing may provide a well-established distribution model and application platform showun in (Figure 1). However, the massive amount of data transmitted between user devices and remote cloud servers produces a data overflow, resulting in saturation and the failure of backhaul networks [1].

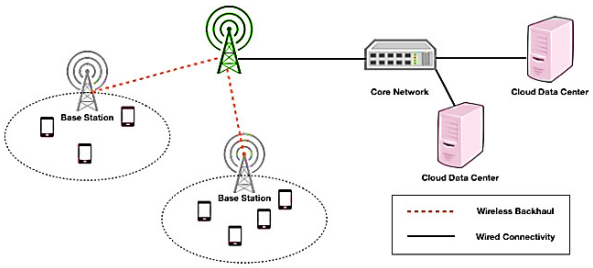



Figure 1: Cloud Computing Scenario

MEC is a new computing paradigm that shifts cloud-computing capabilities away from the centralized cloud and towards the wireless network border to satisfy the growing need for low-latency tasks. The European Telecommunications Standard Institute (ETSI) presented MEC as a new platform that enables cloud computing paradigm within a Radio Access Network (RAN) in close proximity to mobile users. In today's technological age, the use of tablets and smart phones plays an important role in every situation. As the number of mobile users grows, so do the challenges to Quality of Service (QoS) and Quality of Experience (QoE) [2]. Difficulties such as service disruption due to user mobility also do occur. As wireless devices profited from Moore's law, cellular mobile technologies are able to maintain their position as a focal point for introducing new and interesting features and benefits to the end user [3]. It can boost user computation capacity in applications such as augmented reality [4]. MEC enables users to discharge jobs to MEC servers situated at the network's edge. Given the variability of the offloaded activities' requirements and limited MEC capabilities, task offloading and scheduling is a difficult combinatorial problem [5]. MEC helps in minimizing congestion on mobile networks and

decrease latency by moving the strain of cloud computing to individual local servers, hence improving the QoS to the end users. The increasing use of digital mobile devices has boosted the data flows of various commercial applications significantly. MEC can utilize distributed computational resources to cover enormous network entities, allowing mobile edge processors to provide access to wide set of mobile users with multiple sources [6]. (Figure 2) [7] depicts MEC in 5G wireless network scenario. As illustrated in (Figure 2), MEC servers are built directly at Base Stations utilizing a generic wireless network-computing platform. It allows Apps to execute in close proximity to end users. With this role, MEC can assist in meeting the stringent low-latency requirements of high-speed wireless networks. The evolving MEC situation is surrounded by diverse terminals and devices, and also diverse content and traffic types. The ETSI Industry Specification Group (ISG) announces the MEC paradigm and reference architecture. It delivers cloud-computing facilities within the RAN. The environment in which these Apps along with QoE optimization systems operate is distinguished by low latency, high bandwidth, and access to dynamic wireless network information.

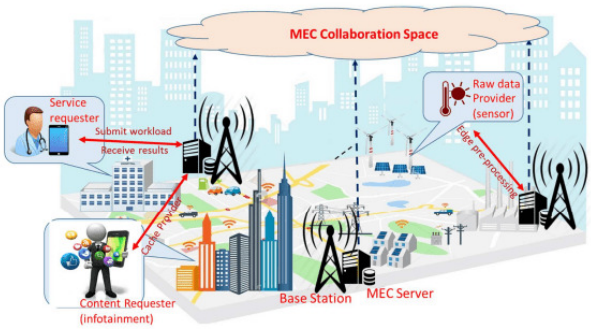



Figure 2: MEC in 5G Wireless Network Scenario

Despite several studies on the MEC environment has been carried out, the effective organization and structure between the central cloud server and the MEC server (MECS) has received little attention in recent publications [8-11]. A further issue is that the MECS and cloud computing resources cannot be utilized effectively because of insufficient attention to operations unique to the content and computing operation types in the MEC environment shown in (Figure 3). As a result, the user service is delayed. Mobility produces a transition in the Access Point (AP), which can cause a delay in the service hosted in the original Server Cluster [5]. When a mobile user switches APs, their data and current services should preferably be relocated or migrated to the Server Cluster at the new AP to reduce connection lag [6]. In one study [12], the migration decision was made by examining the Server Cluster inside a migration zone at the shortest distance from the AP. However, there is a constraint that the selected Server Cluster may occasionally impose extended delays in service delivery due to limited computation capability, bad network connectivity, or insufficient storage capacity to execute the services.

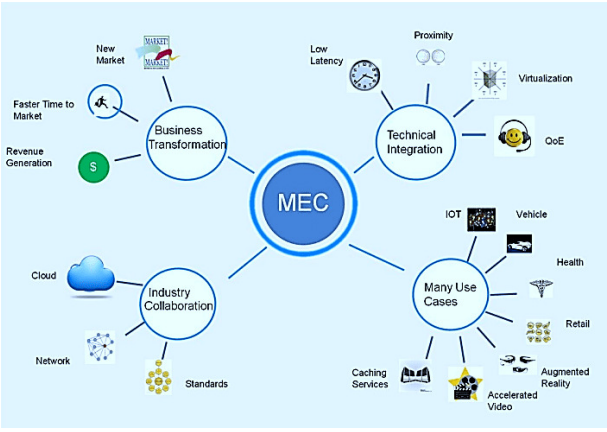



Figure 3: MEC Applications and Use cases

The majority of earlier works have only focused on quality optimization in dynamic service layouts. They, however, neglected the major influence of precise access network selection and service placement. In this paper, a contextual information-based scheduling technique is proposed to address this issue.

Major contributions of the paper are: - A contextual information-based scheduling method for service migration in mobility aware wireless network considering MEC scenario is proposed. - The proposed method is implemented in NS-3 simulator for performance analysis and is evaluated against the existing state-of-the-art scheduling technique. - A comparison of QoS characteristics such as delay, execution time, migration time and downtime are performed.

The remaining sections of this paper is structured as follows: MEC framework is explained in section 2. Various existing research works on resource allocation, mobility support, are mentioned in section 3. Service migration scheduling issues are discussed in section 4. Various scheduling strategies are discussed in section 5. The proposed Contextual Information Based Scheduling method is demonstrated in section VI. Simulation environment for implementing the proposed solution and the corresponding result with quantitative and qualitative performance analysis is demonstrated in section VII. Section 6 represents tentative compound annual growth rate of MEC in various application domains. Finally, the conclusion along with future research direction is mentioned in section 7.

## 2 MEC Framework

MEC defines the technology required to put and operate a mobile edge application on the wireless network edge. The MEC framework is depicted in (Figure 4) [14], where the top layer is the whole-system management layer, which has overall visibility of the system, and the bottom layer is the connectivity layer, which includes all possible connectivity protocols, ranging from local area networks (LAN) to cellular internet. The host level, the layer in between, includes the ME platform and applications module, as well as the virtualized environment on top of which the applications execute [15]. The ME platform is in charge of acquiring all of the resources and services required by the ME Apps. As shown in (Figure 5), associated entities described by the MEC architecture can be classified into following major levels.

The Mobile Edge System (MES) level entity defines the support system, the lifecycle management of user apps, and the mobile edge orchestrator [16]. They are in charge of orchestrating and managing the resources required for the execution of edge applications. The Mobile Edge Platform (MEP) is a set of essential functionalities, which is responsible for enabling mobile edge applications to identify, promote, deliver, and utilize ME services on a specific ME host [17]. The ME platform's essential foundation operations are necessary to guide traffic between applications, services, and wireless networks [18]. The ME platform accommodates rules governing forward operation of traffic related information from the mobile edge platform manager, mobile edge apps, and ME services. All these instructions are then sent to the forwarding unit [19].

## 3 Literature Survey

In the context of mobility aware wireless network, a MEC based distributed job offloading approach for low-load cell groups is proposed in [20]. Using game theory, the MEC node offloading quantity is established based on assessing offloading delay and cost. The research work in [21] offered a combination of power optimization technique and load balancing job-offloading notion based on SDN technology. The authors analyzed the problem framework and discovered a beneficial property, namely unimodular constraint. The authors of [22] used a greedy selection method to create a maximum power utility priority algorithm to achieve successful task offloading on mobile devices, but this strategy does not consider the QoS restrictions of task offloading.

This study makes a migration decision by calculating the shortest distance between Server Clusters inside a migration zone from the AP. MEC networks can also have restricted bandwidth, that may result in network congestion during service migrations. Addressing these problems need a combination of advanced algorithms, predictive modelling, real-time tracking, and coordination across edge nodes [23]. Mobility, latency, and edge node resources are all MEC-specific factors that form service

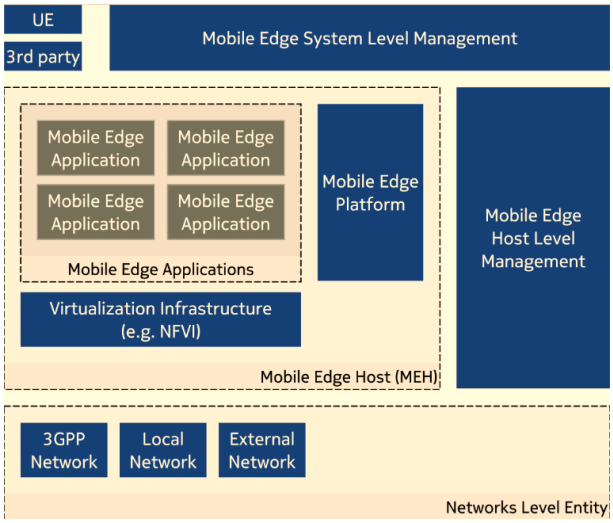



Figure 4: MEC Framework

Table 1: Comparison of state-of-the-art works relevant to the proposed method

| **Survey paper** | **Method applied** | **Limitation** |
|---|---|---|
| [25] | Offered a combination of power optimization technique and load balancing job-offloading notion based on SDN technology | Insufficient for high-load application systems |
| [26] | Greedy selection method | Does not consider the QoS restrictions of task offloading. |
| [27] | Dynamic Resource Allocation scheme for UAVs-assisted Mobile Edge Computing. | Did not consider the impact on the UAV's limited battery life. |
| [28] | Mobility-Aware Joint Task Scheduling (MATS) approach. Makes a migration decision by calculating the shortest distance between Server Clusters inside a migration zone from the AP. | Neglected the influence of access network selection and service placement. |

migration scheduling procedures. All the existing scheduling solutions address a different facet of resource allocation in MEC contexts. The technique chosen is determined by the application's specific requirements, the features of the edge infrastructure, and the intended performance goals. A mixture of these tactics is frequently employed to obtain optimal results in various contexts [24]. The Table 1 highlights comparison of state-of-the-art works relevant to the proposed method.

## 4 Methodology

Assumption: It is assumed that migrating services to a nearby potential server cluster can bring processing and storage resources closer to users, thereby minimizing service delays and increased user experience.

Objective: Based on the surveys and the mentioned challenges it is found that there is no available contextual information-based scheduling strategy that can seamlessly migrate the Virtual Machine that provides services to the user and control availability, latency, and delay. Due to the aforementioned problem, a contextual information-based scheduling method is desired in mobile network that can enhance service quality while the user is moving.

Mobility scenarios involving APs is a crucial factor in determining how services and resources are handled as and when mobile devices move within the network in MEC. A handover process occurs when a mobile device travels from one cell or access point to another to ensure ongoing connectivity

Table 2: List of symbols with description

| Symbols | Description |
|---|---|
| $S_{\text{Cluster}}$ | Server Cluster |
| $S_{\text{Cluster-list}}$ | List of Server Cluster |
| $S_{\text{Cluster-MIPS}}$ | PS values for each Server Cluster |
| $S_{\text{Cluster-RAM}}$ | List of memory sizes for each Server Cluster |
| $S_{\text{Cluster-BW}}$ | List of available bandwidths for each Server Cluster |
| Weight $_{MIPS}$ | Weight applied to each PS of the Server Cluster |
| Weight $_{RAM}$ | Weight applied to each RAM of the Server Cluster |
| Weight $_{BW}$ | Weight applied to each Bandwidth of Server Cluster |
| $S^{\text{Max}}_{Cluster-Pr}$ | Server Cluster with maximum priority |
| $MIPS_{\text{Max}}$ | Maximum PS value for Server Cluster |
| $RAM_{\text{Max}}$ | Maximum RAM size for Server Cluster |
| $BW_{\text{Max}}$ | Maximum Bandwidth for Server Cluster |
| Norm $^{\text{Weight}}_{MIPS}$ | Normalized weighted PS for each Server Cluster |
| Norm $^{\text{Weight}}_{RAM}$ | Normalized weighted RAM for each Server Cluster |
| Norm$^{\text{Weight }_{BW}}_{BW}$ | Normalized weighted BW for each Server Cluster |
| Tot$^{S_{\text{Cluster}}}_{Pr}$ | Total prioritized Server Cluster list |

in the network. The availability of resources changes when devices move closer or farther away from access points. MEC systems must dynamically assign resources to devices based on their proximity in order to ensure optimal performance and minimize latency. The context of devices such as location, network circumstances, and application needs change as they travel. It may be advantageous to move some of its active services to the new access point. To optimize user experiences, MEC systems can dynamically alter resource allocation based on the changing circumstances.

(Figure 5) is representing the mobility scenario across two access points. This scenario is anticipated while proposing the contextual information-based scheduling technique. In this situation, user 1 and 2 are going in opposite directions, at separate speeds and positions. Both the users are linked to AP1, and its VM services are kept on SC3. User 1 is in static position, while user 2 is in motion. When user 2 begins moving towards AP2 and attempts to access services from SC3, there may be a delay in acknowledging service requests, which may result in low latency, high network bandwidth, or an increase in execution time. To deal with this issue, the old SC is shifted to a new SC using the highest priority value. In this situation, the SCs near the AP2 that fall inside the migration area are also prioritized, and only the SC with the highest priority value is chosen for migration. As indicated in (Figure 5), SC6 has the highest priority value when compared to other SCs; hence, SC3 is relocated to SC6.

To understand the mobility scenario for implementing the proposed algorithm, imagine a group of students participating in an online course using their mobile devices or tablets. During the session, the seating area is free to walk around. Assume that the building has an MEC infrastructure and real-time operating systems are enabled. Note that if a student experiences poor service due to distance to access point, the MEC system must immediately move the meeting to a nearby access point. This process helps in reducing latency, and enabling cooperation even during movement. This research work proposes contextual information-based scheduling technique for server cluster selection. (Figure 6) represents pictorial flow of the following steps executed in the scheduling technique.

Step 1: Track user's location to assess their mobility Step 2: Determine whether the user's mobile device is in motion based on information obtained from step 1. Step 3: Locate the migration zone Step 4: Refer to the location where the choice to migrate is made. Step 5: Examine the availability of Server Clusters in the migration zone. Step 6: Use the matrix based on device contextual information to prioritize Server Cluster. Step 7: Select the Server Cluster with highest priority.

To perform prioritization totally three parameters namely Million instructions per second (MIPS),

Random Access Memory (RAM), and Bandwidth (BW) are assessed and aggregated for each Server Cluster. The proposed contextual information-based scheduling algorithm normalizes the values of these parameters by dividing each Server Cluster parameter value by its maximum value shown in Table 2. The algorithm for Contextual Information Based Scheduling is presented below. Input to this algorithm is list of Server Clusters and output from this algorithm is Server Cluster with maximum priority value. The algorithm is tailored to identify suitable server cluster based on its attributes and contextual information. An integral process of evaluating server clusters for diverse jobs is offered via this algorithm. For achieving a server cluster with the highest priority value, the algorithm examines each server cluster's metrics and then prioritizes them considering the determined total priority. The technique successfully calculates priority value for each server cluster and chooses the one with the greatest estimated priority value by taking MIPS, RAM, and BW into account in a normalized and weighted way.

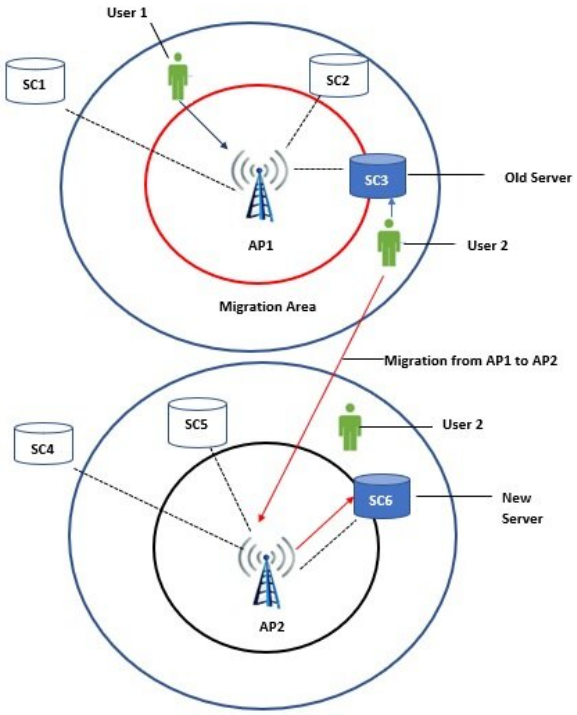



Figure 5: Mobility scenario across Access Points.

The algorithm is initialized with a list of server clusters identified as Scluster showun in (Figure 7). The algorithm iterates through each cluster represented by the index i, where n is the total server clusters in the list. The technique gathers MIPS, RAM, and BW values for each server cluster by using functions getMIPS (), getRAM (), and getBW (). Using Equations 1, 2, and 3, the algorithm computes maximum values of MIPS, RAM, and BW. These maximum values serve as normalization factors for subsequent calculations. The technique uses the estimated maximum values to normalize the MIPS, RAM, and BW numbers for each server cluster i. It then assigns weight values to each server cluster using Equations 4, 5, and 6.

These weight values describe the significance of the metrics of each server cluster. Using Equation 7, the technique computes the total priority value for each server cluster. This total priority value combines the weighted MIPS, RAM, and BW metrics.

$$MIPS_{\text{Max}} = \text{Max}\left(S_{\text{Cluster-MIPS}}\right) \tag{1}$$

$$RAM_{\text{Max}} = \text{Max}\left(S_{\text{Cluster-RAM}}\right) \tag{2}$$

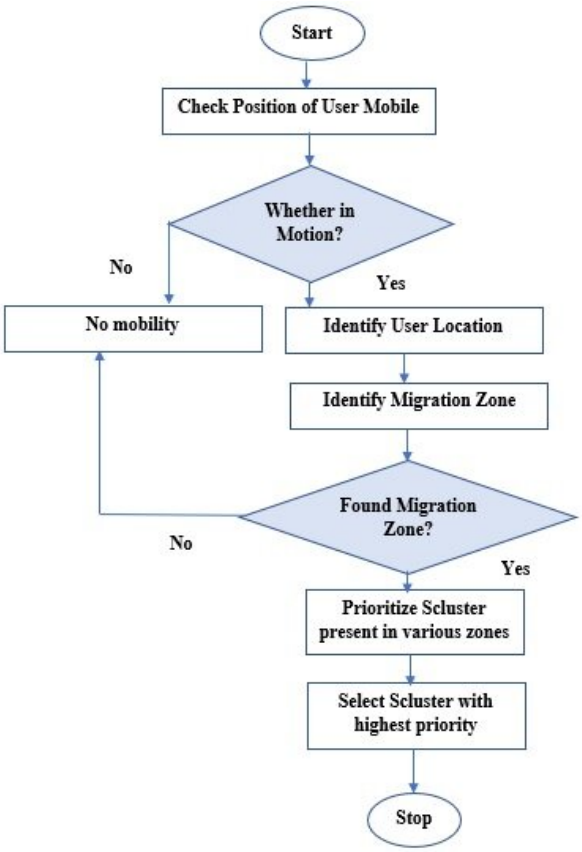



Figure 6: Mobility scenario across Access Points.

$$BW_{\text{Max}} = \text{Max}\left(S_{\text{Cluster}-BW}\right) \tag{3}$$

$$\text{Norm}_{\text{MIPS}}^{\text{Weight MIPS}}[i] \leftarrow \left(S_{\text{Cluster}}[i].getMIPS/MIPS_{\text{Max}}\right) * W_{\text{Weight}}\, t_{\text{MIPS}} \tag{4}$$

$$\text{Norm}_{RAM}^{\text{Weight } RAM}[i] \leftarrow \left(S_{\text{Cluster}}[i].getRAM/RAM_{\text{Max}}\right) * W_{\text{Weight}}\, RAM \tag{5}$$

$$\text{Norm}_{BW}^{\text{Weight } BW}[i] \leftarrow \left(S_{\text{Cluster}}[i].getBW/BW_{\text{Max}}\right) * Weight_{BW} \tag{6}$$

$$\text{Tot}_{Pr}^{\text{S } Cluster} \leftarrow \text{Norm}_{\text{MIPS}}^{\text{Weight MIPS}}[i] + \text{Norm}_{RAM}^{\text{Weight } RAM}[i] + \text{Norm}_{BW}^{\text{Weight } BW}[i] \tag{7}$$

Limitation: Presently, the proposed methodology does not consider the multi-dependence scenario within the mobile terminal application or the mobile terminal's residual battery power. This particular feature will be continued as part of the proposed method's extension effort.

## 5 Result Discussion

The proposed method is implemented using the Network Simulator-3(NS-3) simulation tool. NS-3 is an open-source network simulator that provides a comprehensive set of modules for wireless network simulations including mobility modelling support. It helps in the analysis of the influence of user mobility on system performance. Table 3 contains simulation settings along with values that are used to implement the proposed approach in MEC based wireless network scenario. We have used a variety of input values as part of the simulation setup environment. We considered a 10 x 10 km area with 145 server clusters distributed in the simulation environment. Each server cluster is linked to a single AP with a 1000-meter signal range. A simulation ends once the user finishes the migration

**Algorithm 1: Contextual Information Based Scheduling**

1. if $S_{\text{Cluster-list}} \neq 0$ // Input to the algorithm
2. for $(i = 0; i < n; i++)$ // $n$ = Total number of server clusters in the list
3. $\quad S_{\text{Cluster-MPS}}[i] = S_{\text{Cluster}}[i]$.getMIPS
4. $\quad S_{\text{Cluster-RAM}}[i] = S_{\text{Cluster}}[i]$.getRAM
5. $\quad S_{\text{Cluster-BW}}[i] = S_{\text{Cluster}}[i]$.getBW
6. end for
7. Calculate $MIPS_{\text{Max}}, RAM_{\text{Max}}, BW_{\text{Max}}$ using Eq. (1), (2), and (3)
8. for $(i = 0; i < n; i++)$ // $n$ = Total server clusters in the list
9. $\quad$ Perform normalization and allocate weight values to each $S_{\text{Cluster}}[i]$ using Eq. (4), (5), (6)
10. end for
11. Calculate $\text{Tot}_{pr}^{S_{\text{Cluster}}}$ using Eq. (7)
12. Find $S_{\text{Cluster}_{Max}^{Pr}}$ from $\text{Tot}_{Pr}^{S_{\text{Cluster}}}$ // Output from the algorithm

**Code snippet**

```
#include <iostream>
#include <vector>
struct ServerCluster {
   double getMIPS() const { /* for Implementation of getMIPS() and returning MIPS value */; }
   double getRAM() const {/* for Implementation of getRAM () and returning RAM value */; }
   double getBW() const { /* for Implementation of getBW() and returning BW value */; }};
void ComputeMaxValues(const std::vector<SCluster>& S_Cluster, int n)
{ if (n! = 0) {
     std::vector<double> S_Cluster_MIPS(n), S_Cluster_RAM(n), S_Cluster_BW(n);
     // Retrieve MIPS, RAM, and BW values for each server cluster
     for (int i = 0; i < n; ++i) {
        S_Cluster_MIPS[i] = S_Cluster[i]. getMIPS();
        S_Cluster_RAM[i] = S_Cluster[i]. getRAM();
        S_Cluster_BW[i] = S_Cluster[i]. getBW();}
     // Calculate Max values using Eq. (1), (2), and (3)
     double MIPS_Max, RAM_Max, BW_Max;
     // Calculation of MIPS_Max, RAM_Max, BW_Max using Eq. (1), (2), (3)
     // Perform normalization and allocate weight values to each S_Cluster[i]
     for (int i = 0; i < n; ++i) {
        // Perform normalization and allocation of weight values using Eq. (4), (5), (6)
        // Calculate Tot_Pr^S_Cluster using Eq. (7)
        double Tot_Pr_S_Cluster = /* Calculation using Eq. (7) */;}
     // Find S_Cluster_Max^Pr from Tot_Pr^S_Cluster
     // double S_Cluster_Max_Pr = /* Find S_Cluster_Max_Pr from Tot_Pr^S_Cluster */;
     // Output S_Cluster_Max_Pr}}
int main() {
   int n; // Total number of server clusters
   std::vector<SCluster> serverClusters(n);
   // Call the function with the server clusters list and the total number of clusters
   calculateMaxValues(serverClusters, n);
   return 0;}
```

Figure 7: Code Snippet

processes. The migration process starts once the user reaches the migration point, defined at 50 meters from its connected AP.

(Figure 8) shows the migration time. The timeframe required to transfer the data of a user

application from the source Server Cluster to the destination Server Cluster is referred to as migration time. The migration time in NS-3 is calculated by noting the timestamps when the migration operation began and ended. CIBS migration time is 97512ms, while MATS migration time is 99213ms. Lower migration times mean less service interruption and, as a result, higher QoS.

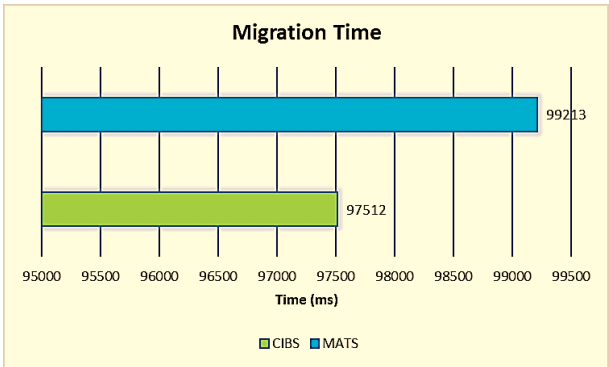



Figure 8: Migration Time (ms).

(Figure 9) depicts the delay between the user's mobile device and the Server Cluster. Delay is calculated as the time taken to process and finish a service request. When a service request is submitted to a MEC server and is properly processed and completed, the delay time is logged as a timestamp in NS-3. The delay time is derived using the time difference between each request. The CIBS delay time is 372115ms, whereas the MATS delay time is 374212ms. Lower CIBS latency suggests that it is better at delivering services to users.

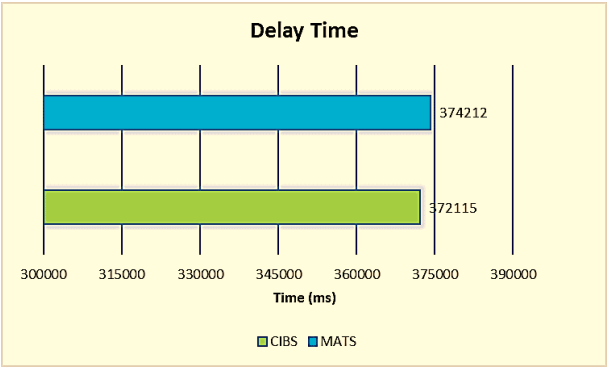



Figure 9: Delay Time (ms).

Execution time is defined as the time taken for a service to be executed on a MEC server after it is scheduled. The timestamps at which a task begins and ends are recorded in NS-3. To determine the execution time for service requests, the time difference is determined. As shown in (Figure 10), the execution time in CIBS is shorter than that in MATS. According to the results, the execution procedure takes 1061328ms in CIBS and 1097231ms in MATS.

The term "downtime" refers to the time when a device or service is unable to execute its intended duties. The start and end times of the downtime period are noted in NS3. The total downtime period is obtained by subtracting the start time from the end time. In (Figure 11), it can be seen that downtime of CIBS is 98715ms whereas for MATS it is 100213ms shown in Table 4. Low downtime in CIBS implies that it outperforms MATS in terms of offering superior QoS to users in motion.

## 6 Tentative compound annual growth rate of MEC in various application domains

In a MEC-based design, additional processing resources positioned at the cellular network's edge allow time-critical applications to meet stringent latency requirements. In the context of 5G, MEC

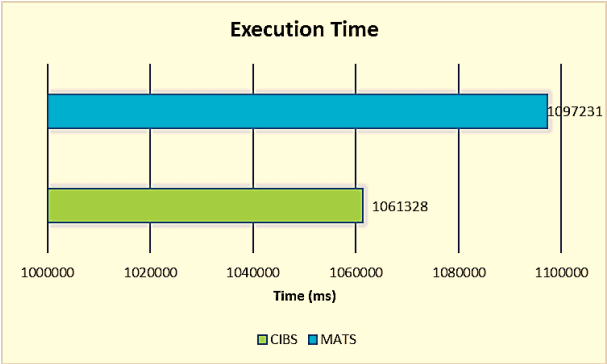



Figure 10: Execution Time (ms).

Table 3: Simulation settings

| Simulation Parameters | Parameter Values |
|---|---|
| Simulation Tool | NS3 |
| MIPS | 3500MIPS |
| RAM | 1024MB |
| BW | 1Mbps |
| Server Cluster count | 145 |
| User migration speed | 25kmph |
| Programming Language | C++ |
| Operating System | Ubuntu |

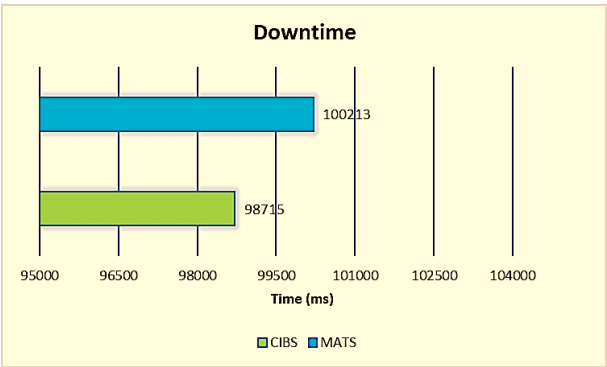



Figure 11: DownTime (ms).

enables wireless communications to exceed crucial latency requirements, allowing for the flexible and unrestricted use of productivity and quality-enhancing technologies such as augmented reality (AR), Virtual Reality (VR), robotics, machine vision, and so on. The top three use cases are estimated to account for majority of MEC spending by 2027: connected transportation, remote monitoring and maintenance, and augmented and virtual reality (AR/VR). The majority of new 5G-MEC transportation/automotive use cases include cellular-vehicle-to-everything technology and are primarily con-

Table 4: Statistical evidence of performance

| Performance Metrics | CIBS | MATS |
|---|---|---|
| Migration Time (ms) | 97512 ms | 99213 ms |
| Delay Time (ms) | 372115 ms | 374212 ms |
| Execution Time (ms) | 1061328 ms | 1097231 ms |
| Downtime (ms) | 98715 ms | 100213 ms |

cerned with occupant and pedestrian safety, improved driver assistance, and occupant convenience. Industry 4.0 applications such as anomaly detection and alert management, as well as machine health indexing, will be included in remote monitoring and maintenance. 5G-MEC would enable remote troubleshooting, remote quality assurance inspection, guided assembly, and training and knowledge transfer in the AR/VR sector. (Figure 12) depicts the mentioned scenario.

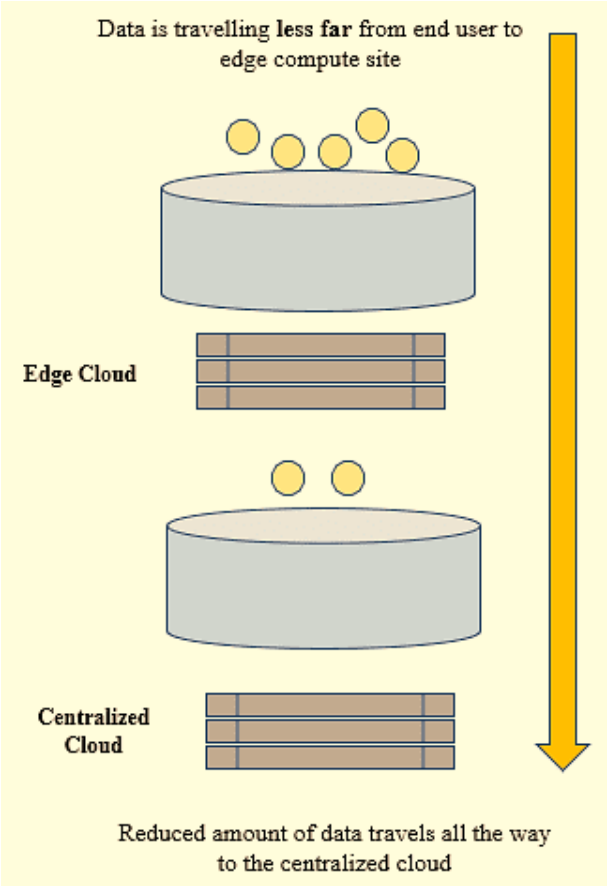



Figure 12: Shift to centralized cloud in MEC

(Figure 13, 14,and 15) depicts expected growth rate of MEC in various application domains for short timeline, medium timeline and long timeline [42].

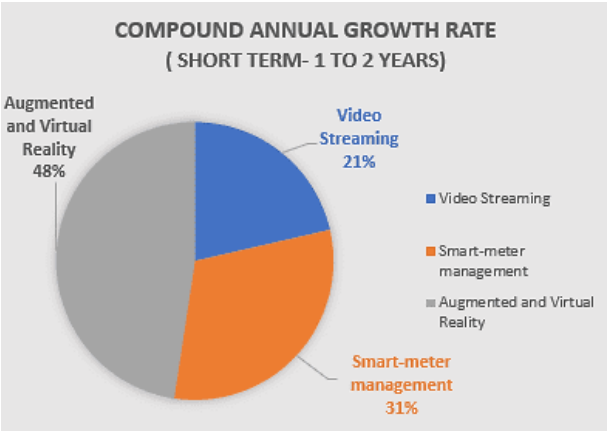



Figure 13: MEC Growth Rate for Short Timeline

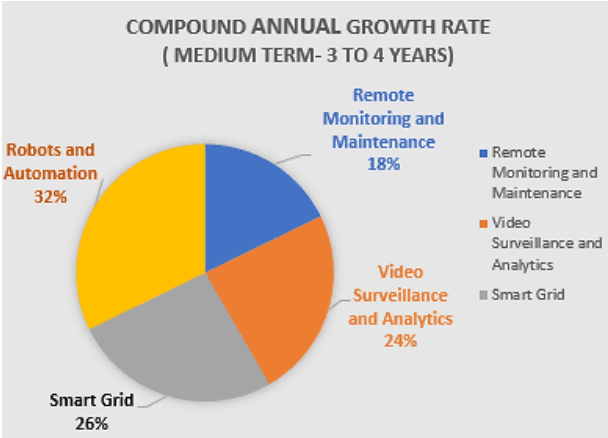



Figure 14: MEC Growth Rate for Medium Timeline

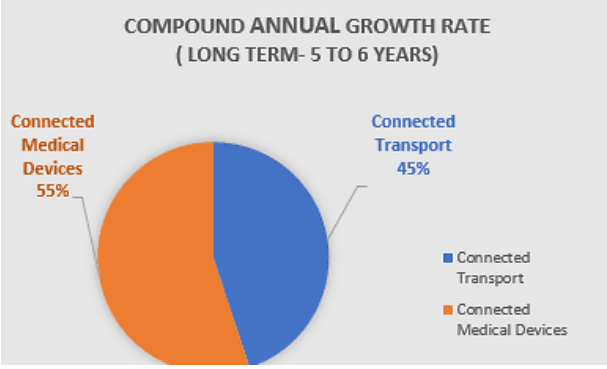



Figure 15: MEC Growth Rate for Long Timeline

## 7 Conclusion and future work

MEC has evolved as a new wireless network paradigm with mobility for assisting low latency applications by enabling computation offloading at the network edge. This paper offered a Contextual Information Based Scheduling approach for increasing service quality when a user moves from one access point to another. The proposed method assigns a priority value to each Server Cluster and then chooses the Server Cluster with the highest priority value to migrate the services. This study is experimented using the NS-3 simulator tool, and its performance is compared to the state- of-the-art MATS technique. The simulation findings show that CIBS surpasses the present MATS technique in terms of migration time, delay time, execution time and downtime. This research work adopted practical implications such as resource optimization, improved quality of service, load balancing across edge servers, reduced energy consumption by effectively utilizing resources closer to end-users, fault tolerance by enabling service rerouting in the event of server failures or network disruptions, and ensuring continuous service availability. The proposed work can be expanded in the future by considering the contextual information of numerous additional entities involved in mobile edge computing.

### Acknowledgements

The authors extend their appreciation to the Deanship of Scientific Research at King Khalid University (KKU) for funding this research through the Research Group Program Under the Grant Number:(R.G.P.2/283/44).

## References


[1] Juyong Lee and Jihoon Lee (2018). Hierarchical Mobile Edge Computing Architecture Based on Context Awareness, *Applied Sciences*, 8(7), 1160, 2018.

[2] Teja Sree B, G. P. S. Varma, Hemalatha Indukurib (2023) Mobile Edge Computing Architecture Challenges, Applications, and Future Directions, *International Journal of Grid and High-Performance Computing*, 15 2, 1–23 2023.

[3] Al-Badarneh, J., Jararweh, Y., Al-Ayyoub, M., Fontes, R., Al-Smadi, M., and Rothenberg, C. (2018). Cooperative mobile edge computing system for VANET based software-defined content delivery. *Computers and Electrical Engineering*, 71, 388–397, 2018.

[4] Joseph Boccuzzi (2018), Introduction to Cellular Mobile Communications, *Springer Book of Multiple access techniques for 5G wireless networks and beyond*,3-37, 2018.

[5] Alameddine, H. A., Sharafeddine, S., Sebbah, S., Ayoubi, S., and Assi,C. (2019) Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. *IEEE Journal on Selected Areas in Communications*, 37(3), 668–682, 2019.

[6] Huang J, Li S, Chen Y (2020) Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing. *Peer Network Application* 13(5), 1776–1787, 2020.

[7] Tuyen X. Tran, Abolfazl Hajisami, Parul Pandey, and Dario Pompili (2017). Collaborative Mobile Edge Computing in 5G Networks: New Paradigms, Scenarios, and Challenges, *IEEE Communications magazine, special issue on fog computing and networking.*, 55(4), 54-61, 2017.

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

[9] Muthukumaran, V., Kumar, V. V., Joseph, R. B., Munirathanam, M., and Jeyakumar, B. (2021). Improving network security based on trust-aware routing protocols using long short-term memory-queuing segment-routing algorithms. *International Journal of Information Technology Project Management (IJITPM)*, 12(4), 47-60, 2021.

[10] Chen, T., Matinmikko, M., Chen, X., Zhou, X., Ahokangas, P (2018). Software defined mobile networks: Concept, survey, and research directions. *IEEE Communications Magazine*, 53, no. 11, 126-133, 2018.

[11] Yixue, H., Min, C., Long, H., Hossain, M.; Ahmed, G. Energy Efficient Task Caching and Offloading for Mobile Edge Computing. *IEEE Access*, 6, 11365–11373, 2018.

[12] Puliafito Carlo, Goncalves Diogo M, Lopes Marcio M, Martins Leonardo L, Madeira Edmundo, Mingozzi Enzo, Rana Omer, Bittencourt Luiz F. Mobfogsim (2020) Simulation of mobility and migration for fog computing. *Simulation Model Practice Theory,* 101, 102062, 2020.

[13] [Online]. Available: https://5ghub.us/5g-mobile-edge-computing-mec-2/

[14] Hiroyuki Tanaka, Masahiro Yoshida, Koya Mori, Noriyuki Takahashi. (2018), Multi-access Edge Computing: A Survey, *Journal of Information Processing*,26,87-97, 2018.

[15] Chen M, Hao Y (2018) Task offloading for mobile edge computing in software defined ultra-dense network. *IEEE Journal on Selected Areas in Communications*, 36(3), 587–5979, 2018.

[16] Zhang J, Guo H, Liu J, Zhang Y (2019) Task offloading in vehicular edge computing networks: A load-balancing solution. *IEEE Transaction Vehicular Technology* 69(2), 2092–210410, 2019.

[17] Maithili, K., Vinothkumar, V., and Latha, P. (2018). Analyzing the security mechanisms to prevent unauthorized access in cloud and network security. Journal of Computational and Theoretical Nanoscience, 15(6-7), 2059-2063, 2018.

[18] Huang J, Li S, Chen Y (2020) Revenue-optimal task scheduling and resource management for IoT batch jobs in mobile edge computing. *Peer Network Application*, 13(5), 1776–1787, 2020.

[19] Shi J, Jun D, Wang J, Wang J, Yuan J (2020) Priority-aware task offloading in vehicular fog computing based on deep reinforcement learning. *IEEE Trans Vehicular Technology*, 69(12), 16067–16081, 2020.

[20] Yang J, Xiao S, Jiang B, Song H, Khan S, Ul Islam S (2020) Cache- enabled unmanned aerial vehicles for cooperative cognitive radio networks. *IEEE Wireless Communication* 27(2), 155–16126, 2020.

[21] Saha, S., Kumar, V. V., Niveditha, V. R., Kannan, V. A., Gunasekaran, K., and Venkatesan, K. (2023). Cluster-Based Protocol for Prioritized Message Communication in VANET. *IEEE Access*, 11, 67434–67442, 2023.

[22] Zaiwar Ali, Sadia Khaf, Ziaul Haq Abbas, Ghulam Abbas, Lei Jiao, Amna Irshad, Kyung Sup Kwak and Muhammad Bilal (2020) A Comprehensive Utility Function for Resource Allocation in Mobile Edge Computing, 2020.

[23] Cheng Y, Liang C, Chen Q, Yu R (2021) Energy-efficient D2D- assisted computation offloading in NOMA-enabled cognitive networks. *IEEE Transaction Vehicular Technology*, 70(12), 13441-13446, 2021.

[24] Kumar, V. D., Kumar, V. V., and Kandar, D. (2018). Data transmission between dedicated short range communication and WiMAX for efficient vehicular communication. *Journal of Computational and Theoretical Nanoscience*, 15(8), 2649-2654, 2018.

[25] Jie Lin, Lin Huang, Hanlin Zhang, Xinyu Yang, Peng Zhao (2022) A Novel Lyapunov based Dynamic Resource Allocation for UAVs-assisted Edge Computing, *Computer Networks*, 205, 108710, 2022.

[26] Kumar, Dr. V. V., Arvind, Dr. K. S., Umamaheswaran, Dr. S., and Suganya, K. S. (2019). Hierarchal Trust Certificate Distribution using Distributed CA in MANET. *International Journal of Innovative Technology and Exploring Engineering*, 8(10), 2521–2524, 2019.

[27] Tien Van Do, N.H. Do, H.T. Nguyen, Csaba Rotter, Attila Hegyi, Peter Hegyi, (2019). Comparison of scheduling algorithms for multiple mobile computing edge clouds, *Simulation Modelling Practice and Theory*, 93, 104-118, 2019.

[28] Zhang, M., Huang, H., Rui, L., Hui, G., Wang, Y., and Qiu, X. (2020). A service migration method based on dynamic awareness in mobile edge computing. *In NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium*. 1-7, 2020.


Journal's webpage: http://univagora.ro/jour/index.php/ijccc/


This journal is a member of, and subscribes to the principles of,
the Committee on Publication Ethics (COPE).
https://publicationethics.org/members/international-journal-computers-communications-and-control

*Cite this paper as:*

Sanchari Saha, Iyappan Perumal, V.R.Niveditha, Mohamed Abbas, I.Manimozhi, C.Rohith Bhat (2024). Contextual Information Based Scheduling for Service Migration in Mobile Edge Computing, *International Journal of Computers Communications & Control*, 19(3), 6143, 2024.
https://doi.org/10.15837/ijccc.2024.3.6143