Cognitive Edge Computing based Resource Allocation Framework for Internet of Things

Anas Amjad[§], Fazle Rabby[*], Shaima Sadia[**], Mohammad Patwary[†] and Elhadj Benkhelifa[‡]

[§]School of Creative Arts and Engineering, Staffordshire University, Stoke-on-Trent, UK
[*]Department of Software Engineering, Daffodil International University, Dhaka, Bangladesh
[**]Department of ETE, Daffodil International University, Dhaka, Bangladesh
[†]School of Computing and Digital Technology, Birmingham City University, Birmingham, UK
[‡]School of Computing and Digital Technologies, Staffordshire University, Stoke-on-Trent, UK
Email: anasamjad@ieee.org, {fazle35-871, sadia1493}@diu.edu.bd,
mohammad.patwary@bcu.ac.uk, e.benkhelifa@staffs.ac.uk

***Abstract*—Due to the inherent property of the processing resource request from mobile active or passive devices as part of internet of things (IoT), processing capacity as well as latency become major optimization criteria. To achieve overall optimized uses of cloud resources - having dynamic tracking, monitoring as well as orchestration framework is one of the key challenges to overcome. In the same context, enhanced uses of computing devices at distributed location is predicted to facilitate the success of IoT; subsequently the success of fifth generation (5G) of Wireless technologies. This opens enormous potential to integrate the unused resources of such distributed computed devices within the conventional cloudlet or cloud federation. However, this requires an efficient micro-level distributed computing resource tracking, monitoring and orchestration; where resources are distributed in geo-location as well as the availability of unused resources are time variant in nature. In this paper, we have proposed a cognitive edge-computing based framework solution for these requirements in order to achieve an efficient use of these distributed resources. This provides the end-user with a dynamic soft extension of computing facilities of cloudlet and cloud federation, as well as a revenue generation avenue to end-user. The simulation results show that such extension can be an exponential function of the number of local processing platforms agreed to participate in the proposed cognitive resource sharing.**

***Index Terms*—Distributed computing, edge computing, internet of things, merchant mode, resource sharing.**

I. Introduction

Over the last decade, the exponential growth in the number of connected devices has led the research community and the industry to focus on issues that arise due to the increasing network size. Internet of Things (IoT) has emerged as a paradigm to provide such connectivity over dynamic and global network infrastructures using a set of hardware and software components [1, 2]. In general, IoT comprises of devices that are widely distributed in nature and may have limited local storage and processing capabilities; consequently, posing challenges for IoT design. In order to tackle the issues faced in IoT, cloud computing has gained significant attention. Within modern telecommunication systems, cloud acts as a backbone component of the IoT and provides access to a shared pool of resources [3], where the storage and processing capabilities are assumed to be virtually unlimited [1]. The integration of IoT with cloud computing provides flexibility, scalability and energy-efficiency to the modern telecommunication systems. Many research studies have considered such integration to develop solutions for a diverse range of applications. Considering the drive towards the digitization of health care systems, an infrastructure is proposed in [4] for real-time health monitoring of patients based on such integration. Authors in [5] have developed a vehicular data cloud services model for intelligent parking and vehicle warranty analysis. In [6], a cloud computing based model is proposed to meet the computational requirements for smart grid applications. Since, data storage is a critical issue in IoT, a cloud computing based framework for the storage of massive IoT data is proposed in [7]. To provide services with IoT, clouds are required to meet particular quality requirements depending on the customers' expectations. In order to provide service perspective, a model is developed in [8], while considering different quality dimensions and metrics. The effectiveness of the developed model is demonstrated by evaluating the quality of multiple storage clouds.

Nevertheless, with the continuously growing demands for resources from cloud, service providers are expected to experience a performance bottleneck. Therefore, the assumption of unlimited cloud capability may not be feasible [9, 10]. Furthermore, cloud computing causes problems for latency-sensitive applications as it results in unreliable latency due to geo-distributed devices in IoT [10–12]. In future generation of wireless networks, two of the key elements for the realization of 5G are connectivity and latency. The global success of 5G is possible only if IoT can facilitate services that meet users' expectations in terms of connectivity and latency [13, 14]. Edge computing has emerged as a promising solution to the problems faced in cloud computing and to comply with the 5G communication standards [15, 16]. Edge computing extends the existing cloud computing paradigm to the network edge in order to meet the requirements of latency-critical and computation-intensive IoT applications. Complying with the 5G vision, edge computing reduces the overall latency

978-1-5386-2859-1/17/$31.00 ©2017 IEEE

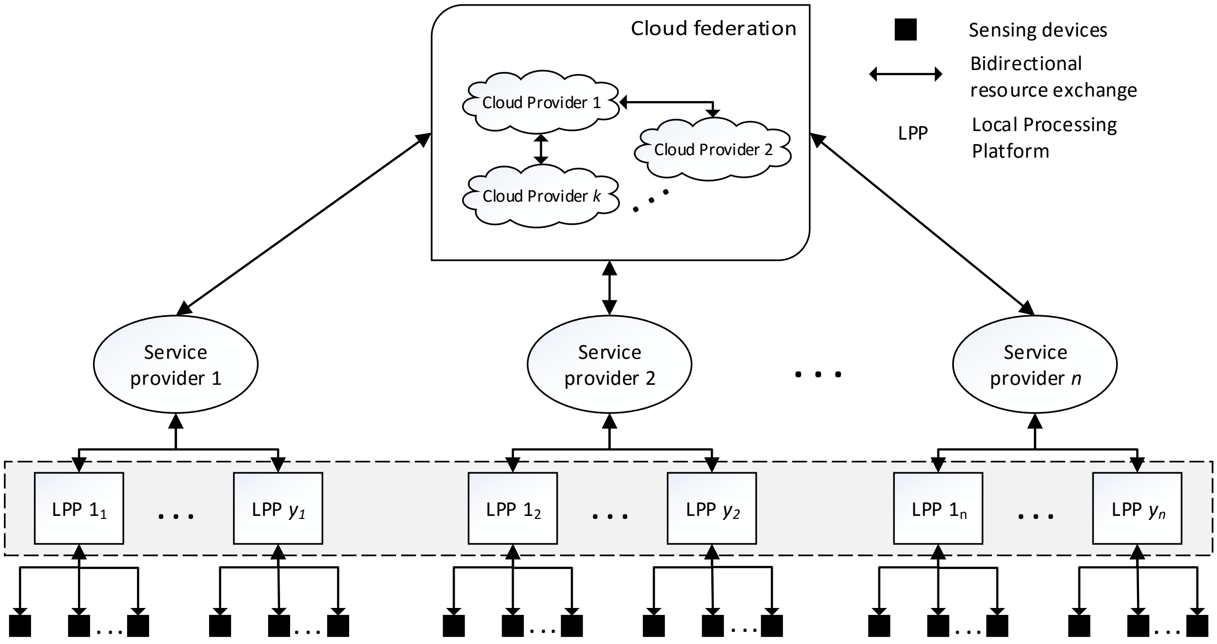

Fig. 1. Proposed framework for tracking, monitoring and orchestration of distributed cloud resources

and provides an energy efficient solution [9]. Nevertheless, in order to make the resource allocation to be more efficient, the tracking, monitoring and orchestration of cloud resources is required. One of the possible candidates to track the system usage metrics is ATOM, which is proposed in [17]. However, in resource saturated circumstances, ATOM may not provide an optimal solution to the resource allocation problem.

Motivated by the significance of IoT in future generation of wireless networks, this paper considers edge computing as well as connectivity and latency to develop an efficient solution for dynamic tracking, monitoring and orchestration of cloud resources in IoT. The contributions of this paper are summarised as follows:

- A framework for dynamic tracking, monitoring and orchestration (DTMO) of cloud resources is proposed. The proposed framework incorporates softness in the decision making process to provide an efficient solution for dynamic resource allocation.
- Considering the exponential growth in the number of connected devices, an algorithm is proposed for computation offloading to tackle the issue of processing capacity shortage.
- In order to reduce the latency in future generation of wireless networks, a strategy is devised for computation offloading.

The advantages of the proposed framework are two-fold. Firstly, the request for resource is fulfilled in an efficient manner and secondly, the device that provides its resources can generate revenue.

The rest of this paper is organised as follows: Section II presents the system model and proposed methodology. Section III evaluates the proposed framework. Finally, the conclusions are made in Section IV.

II. System Model and Proposed Methodology

The proposed framework for distributed tracking, monitoring and orchestration of cloud resources is shown in Fig. 1. Considering the IoT environment, various sensing devices are present that are capable of generating different types of data such as temperature, pressure, visual information etc. A number of sensing devices are connected to each local processing platform (LPP), which is capable of processing the data obtained from the sensing devices. In order to enhance the energy efficiency, each LPP can classify sensing tasks between its sensing devices [18]. The proposed model comprises of n number of service providers, where each service provider can be represented by $\mathcal{S}_e$, and it can support y_e number of static/mobile LPP, such that $e = 1, 2, 3, \ldots, n$. A cloud federation exists in the proposed model [19], which provides core functionality by utilising the contributed resources from k number of cloud providers. Once the data is processed by a LPP, it can be sent to the cloud for storage. Moreover, the data collected from different LPP can be collaborated at the cloud for efficient decision making. Since LPP may have limited energy, they may have to offload their tasks by requesting for a computational resource. In that case, the respective service provider will send the request to the cloud federation.

Considering computation offloading [20] and the shared resources of the cloud providers, the cloud federation looks for available resources that can be allocated to LPP in order to perform a particular task. Although, this approach facilitates the sharing of cloud resources with LPP, it is inefficient in

TABLE I
KEY SYMBOLS AND THEIR DEFINITION

Symbol	Definition
k	the number of cloud providers
n	the number of service providers
C_i^t	the total capacity of ith cloud provider
C_u^t	the currently occupied capacity of ith cloud provider
n_i	the number of data centers of ith cloud provider
$\mathbf{D}$	the respective cloud controllers
$d_{i,j}$	the distance between ith cloud controller and jth data center
R_r	the amount of required resource
R_t	the type of resource requests considered in the proposed model
ρ_w	the available capacity of each LPP
s	the number of LPP that agreed to connect with cloud in case of available capacity
$\mathbf{c}$	the available capacity of s LPP
$\mathbf{p}$	the cost of resources
β_t	the acceptable distance threshold
g	the grading of resources
t_s	the desired grade of requested resource
$\mathcal{F}(\cdot)$	the function to find a cloud provider that can fulfill the resource request
$\mathcal{P}(\cdot)$	the function for purchasing a resource using the merchant mode
ξ	the percentage of LPP that agreed to participate in resource sharing

terms of latency and leads to a performance bottleneck for exponentially growing number of requests [10].

In order to overcome the aforementioned problems, two types of resource requests are considered in the proposed model i.e. (a) computational capacity for processing and storing data; and (b) low-latency computational capacity. The LPP in IoT are expected to be heterogeneous i.e. having different computational capabilities. The LPP with higher computational capability may not utilize their resources fully and may have available resources at a given instance of time. The tracking and monitoring of tasks is performed using ATOM [17] for profiling of the resources. In the proposed model, if the cloud providers cannot fulfill a new resource request, the LPP that have agreed to share their available resources with cloud are considered to acknowledge the request. This approach enhances the efficiency of the system in terms of resource allocation and generates revenue for the LPP that share their resources. The proposed methodology for dynamic tracking, monitoring and orchestration of resources is described below in detail. The list of key symbols used in this paper along with their definition is given in Table I.

Let C_i^t denote the total capacity of the ith cloud provider, where that $i = \{1, 2, 3, \ldots, k\}$. Assume each cloud provider has n_i number of data centers; the distances of data centers from their respective cloud controllers can be represented by $\mathbf{D}$ as,

$$\mathbf{D} = \begin{bmatrix} d_{1,1} & d_{1,2} & \ldots & d_{1,l} \\ d_{2,1} & d_{2,2} & \ldots & d_{2,l} \\ \vdots & \vdots & \ddots & \vdots \\ d_{k,1} & d_{k,2} & \ldots & d_{k,l} \end{bmatrix} \tag{1}$$

where $l = \max\{n_i | i = 1, 2, 3, \ldots, k\}$ and $d_{i,j} \in \mathbf{D}$ is given by,

$$d_{i,j} = \begin{cases} \text{Distance between } ith \text{ cloud} & \\ \text{controller and } jth \text{ data cente,} & j \leq n_i \\ -1, & \text{otherwise} \end{cases} \tag{2}$$

Suppose a LPP requests for a resource, where R_r denotes the amount of required resource. The type of resource requests considered in the proposed model is represented by R_t, such that $R_t \in \{0, 1\}$, where $R_t = 0$ refers to a resource request for computational capability and $R_t = 1$ refers to a resource request for latency-sensitive application. Let C_i^u denote the currently occupied capacity of the ith cloud provider. In order to accommodate a new request for a resource, the type of resource request has to be considered as well as the available resources. Algorithm 1 presents the approach for resource tracking and allocation (RTA). In the proposed algorithm, four different scenarios are considered, as shown in Table II, where $\lambda = \sum_{i=1}^{k} C_i^u + R_r$.

TABLE II
SUMMARY OF THE SCENARIOS CONSIDERED IN THE PROPOSED SYSTEM

Scenario	Resource Request	Condition	Algorithm
1	computational capability	$\lambda \leq \sum_{i=1}^{k} C_i^t \cap R_t = 0$	1
2	computational capability	$\lambda > \sum_{i=1}^{k} C_i^t \cap R_t = 0$	1 and 2
3	low-latency resource	$\lambda \leq \sum_{i=1}^{k} C_i^t \cap R_t = 1$	1 and 3
4	low-latency resource	$\lambda > \sum_{i=1}^{k} C_i^t \cap R_t = 1$	1 and 3

In the first scenario, if a computational resource is requested by a particular LPP, Algorithm 1 checks whether or not a cloud provider can acknowledge the request. ATOM [17] is utilized which tracks the tasks being performed for profiling of the resources and provides information about the available cloud resources. If cloud resources are available, function $\mathcal{F}(\cdot)$ utilizes the profiling information and considers the amount of requested resource to find the cloud provider $\hat{i}$ that can fulfill the request.

As in scenario 1, a computational resource request is considered in the second scenario. However, in contrast to scenario 1, scenario 2 handles the condition when the cloud provider cannot acknowledge the request and a LPP within

Algorithm 1 Proposed resource tracking and allocation scheme

```
Initialisation:
    The amount of requested resource $R_r$, the type of requested resource $R_t$, total capacity of each cloud provider $C_i^t$, currently occupied cloud capacity $C_i^u$, the number of data centers for each cloud provider $n_i$, the distance of each data center from its respective cloud controller $d_{i,j}$ and the acceptable distance threshold $\beta_t$.
$\lambda \leftarrow \sum_{i=1}^{k} C_i^u + R_r$
if $\lambda \leq \sum_{i=1}^{k} C_i^t \cap R_t = 0$ then
    $\hat{i} \leftarrow \mathcal{F}(R_r)$
    $C_{\hat{i}}^u \leftarrow C_{\hat{i}}^u + R_r$
else if $\lambda > \sum_{i=1}^{k} C_i^t \cap R_t = 0$ then
    Go to Algorithm 2
    Obtain LPP ID $\mathcal{I}$ and allocate resource
else if $\lambda \leq \sum_{i=1}^{k} C_i^t \cap R_t = 1$ then
    $\hat{i} \leftarrow \mathcal{F}(R_r)$
    $\mathbf{t} \leftarrow d_{\hat{i},1:n_{\hat{i}}}$
    $\gamma_1 \leftarrow \arg\min_j [t_{1,j} - \beta_t]$
    if $\gamma_1 \leq 0$ then
        $C_{\hat{i}}^u \leftarrow C_{\hat{i}}^u + R_r$
    else
        Go to Algorithm 3
        Obtain LPP ID $\mathcal{I}$ and allocate resource
    end if
else if $\lambda > \sum_{i=1}^{k} C_i^t \cap R_t = 1$ then
    Go to Algorithm 3
    Obtain LPP ID $\mathcal{I}$ and allocate resource
end if
return
```

the IoT has to be searched that has agreed to contribute its available space to be a part of the cloud.

Suppose s denote the number of LPP that have agreed to be connected to be a part of the cloud in case of available capacity. Let ρ_w denote the used capacity of each LPP, such that $w = 1, 2, 3, \ldots, s$; where $\rho_w \in [0, 100]$. Let $\mathbf{c}$ with dimension $1 \times s$ denote the available resources of s LPP and $\mathbf{p}$ with dimension $1 \times s$ denote the respective cost of resources, where the available capacity of wth LPP $c_w \in \mathbf{c}$ is given by $(1 - \rho_w)$. Considering the condition provided in scenario 2, Algorithm 2 is proposed to fulfill the resource request by dynamically finding an optimal solution in terms of computational capability and resource cost. Firstly, several LPP with available resources that are sufficient to handle the resource request are considered along with their respective costs. Merchant mode [21] is considered to find the LPP that can offer suitable amount of resource with minimum possible cost. Let $\mathcal{P}(\cdot)$ be a function for purchasing a resource using the merchant mode. It is assumed that the price of per unit capacity is fixed and it is set by the regulatory authority. The aim of the proposed algorithm is to provide one single LPP that fulfils the request. However, if a single LPP cannot provide enough resources, a combination of LPP $\mathbf{C}$ can be considered to purchase resources with minimum possible cost using the merchant mode. Finally, Algorithm 2 provides ID $\mathcal{I}$ of the LPP which will allocate its resources to fulfill the resource request.

Algorithm 2 Proposed computation offloading algorithm for processing capacity shortage

```
Initialisation:
    The amount of requested resource $R_r$, the number of LPP $s$ that are considered for resource allocation, the used capacity of each LPP $\rho_w$ and the cost of resources $\mathbf{p}$.
$\hat{\mathbf{c}} \leftarrow \varnothing$
for $w \leftarrow 1$ to $s$ do
    $c_w \leftarrow 100 - \rho_w$
    $p_w \leftarrow$ the respective cost of resource
    if $\lfloor c_w - R_r \rfloor = 0$ then
        $\hat{c}_w \leftarrow c_w$
        $\hat{p}_w \leftarrow p_w$
    end if
end for
if $\hat{\mathbf{c}} \neq \varnothing$ then
    $\mathcal{I} \leftarrow \mathcal{P}(\hat{\mathbf{c}}, \hat{\mathbf{p}})$
else
    $\mathbf{C} \leftarrow$ smallest possible combinations of LPP
    $\mathbf{P} \leftarrow$ the respective cost of resources
    $\mathcal{I} \leftarrow \mathcal{P}(\mathbf{C}, \mathbf{P})$
end if
return $\mathcal{I}$
```

In the third scenario, if a resource is requested for a latency-sensitive application, the algorithm performs a check to find if the request can be acknowledged by a cloud provider. Similar to the first scenario, ATOM [17] performs the profiling of the resources available, and cloud provider $\hat{i}$ is found that can fulfill the request. In this case, since the application is latency-sensitive, the intelligence of the proposed system is enhanced by performing additional checks. The distances of the data centers from $\hat{i}th$ cloud controller are considered and denoted by $\mathbf{t}$. Let β_t denote the acceptable distance threshold for a particular application. Algorithm 1 finds if a data center of $\hat{i}th$ cloud provider can fulfill the resource request by considering the distance threshold β_t. In case if the data centers of $\hat{i}th$ cloud provider cannot acknowledge the request, the proposed system provides an efficient approach, given in Algorithm 3, to searches for a LPP in the IoT with available resources which can be allocated to support the LPP that has requested for a low-latency resource.

In contrast to scenario 3, the fourth scenario handles the condition when the resources of the cloud provider are occupied and it cannot acknowledge the request for a low-latency resource. In this scenario, similar to scenario 3, Algorithm 3 is used to find the LPP that can provide its available resources for a particular cost to fulfill the resource request.

Assume $\hat{\mathbf{d}}$ with dimension $1 \times s$ represent the distances of the LPP from the cloud controller that have available resources. Suppose t_s denote the desired grade of requested resource, such that $t_s \in \{1, 2, 3\}$; where $t_s = 1$, $t_s = 2$ and $t_s = 3$ refer to the suitability of a resource for real time, quasi-real time and non-real time tasks. Let $\mathbf{g}$ with dimension $1 \times s$ denote the grading of s resources. In the proposed approach, α_1, α_2 and α_3 are application-specific tunable parameters; for example, if a particular application has acceptable tolerance of $\pm 10\%$, $\alpha_{(\cdot)}$ can be set to 10. Initially, Algorithm 3 classifies the LPP into three categories based on the grading. Subsequently, depending on the desired grade of requested resource t_s, the LPP that are suitable for the task to be performed are chosen. Assume $\tilde{\mathbf{d}}$ denote the distance of the chosen LPP from the cloud controller, $\tilde{\mathbf{c}}$ represents their respective available capacity and the respective resource cost is given by $\tilde{\mathbf{p}}$. Algorithm 3 considers the merchant mode for choosing a suitable resource belonging to LPP $\mathcal{I}$ in order to perform the desired task with optimal cost.

The flow chart of the proposed framework is shown in Fig. 2. In the proposed framework, DTMO tracks the available and blocked resources. Suppose P_w denote the overall available capacity of cloud providers and participating LPP. Let l_t be the target latency. When a new resource request comes into the proposed framework, it goes to the umbrella Algorithm 1. Tracking Monitoring and Orchestration (TMO) context will check if the request is for only computational resource or for low-latency computational resource. If it is only computational resource, then the availability of the resource in the cloud federation's resource pool will be checked. If available, it will allocate the resource according to the request and update the resource monitor of the DTMO framework. If the resource is not available in the cloud federation's resource pool, child Algorithm 2 will be used. Through task tracking, Algorithm 2 will check available computational resource for offloading. By allocation criteria optimization, Algorithm 2 will find the most efficient resource to satisfy the request and takes into account the resource pricing. Once the resource is allocated for a particular request, the resource monitor of the DTMO framework will be updated.

If TMO context finds that the request is asking for low-latency resource, it will check if the data centers of the cloud federation's resource pool can meet the latency threshold. If yes, then Algorithm 1 will allocate the resource and update the resource monitor. If not, then the child Algorithm 3 will be used. Algorithm 3 will check for the availability of the resource for offloading through task tracking. It will classify the candidate resources according to latency threshold. Afterwards, Algorithm 3 will take resource pricing into account to fulfill the resource request. The resource monitor will be updated after resource allocation.

Algorithm 3 Proposed computation offloading algorithm for latency reduction

```
Initialisation:
    The number of LPP $s$ that are considered for resource allocation, the acceptable distance threshold $\beta_t$, the desired grade of requested resource $t_s$; for $s$ LPP: the distance $\hat{\mathbf{d}}$ from the cloud controller, the grading of resources $\mathbf{g}$, the available capacity $\mathbf{c}$ and the cost of resources $\mathbf{p}$.
$\mathbf{g} \leftarrow \varnothing$
Sort $\hat{\mathbf{d}}$ in ascending order
$\mathbf{p} \leftarrow$ the respective cost of resources
for $w \leftarrow 1$ to $s$ do
    if $(\beta_t - \beta_t/\alpha_1) \le \hat{d}_w \le (\beta_t + \beta_t/\alpha_1)$ then
        $g_w \leftarrow 1$
    else if $(\beta_t - \beta_t/\alpha_2) \le \hat{d}_w \le (\beta_t + \beta_t/\alpha_2)$ then
        $g_w \leftarrow 2$
    else if $(\beta_t - \beta_t/\alpha_3) \le \hat{d}_w \le (\beta_t + \beta_t/\alpha_3)$ then
        $g_w \leftarrow 3$
    end if
end for
for $\hat{j} \leftarrow 1$ to $3$ do
    $h_{\hat{j}} \leftarrow s - \sum_{w=1}^{s} \Big[ \operatorname{sgn}(g_w - t_s) \Big]^2$
end for
if $t_s = 1$ then
    $\gamma_2 \leftarrow 1$
    $\gamma_3 \leftarrow h_1$
else if $t_s = 2$ then
    $\gamma_2 \leftarrow h_1 + 1$
    $\gamma_3 \leftarrow h_1 + h_2$
else if $t_s = 3$ then
    $\gamma_2 \leftarrow h_1 + h_2 + 1$
    $\gamma_3 \leftarrow h_1 + h_2 + h_3$
end if
$\tilde{\mathbf{d}} \leftarrow \mathbf{d}_{\gamma_2:\gamma_3}$
$\tilde{\mathbf{p}} \leftarrow \mathbf{p}_{\gamma_2:\gamma_3}$
$\tilde{\mathbf{c}} \leftarrow \mathbf{c}_{\gamma_2:\gamma_3}$
$\mathcal{I} \leftarrow \mathcal{P}(\tilde{\mathbf{d}}, \tilde{\mathbf{c}}, \tilde{\mathbf{p}})$
return $\mathcal{I}$
```

III. Evaluation

The evaluation investigates the feasibility of the proposed framework for efficient use of distributed resources. The standard workload scenario for simulating cloud resource allocation is considered [22, 23]. Exponentially growing number of heterogeneous LPP within the range of 1000 to 10,000 is considered. The activity factor of LPP that agree to participate in the proposed cognitive resource sharing is assumed to be 50%. Depending on the percentage of available resources that the participating LPP have agreed to share (denoted by ξ) and the type of resource request, different scenarios are considered to demonstrate the effectiveness of proposed RTA algorithm. The simulation is modelled by considering $\xi \in \{15\%, 30\%, 45\%\}$.

Fig. III shows a comparison of the proposed scheme with

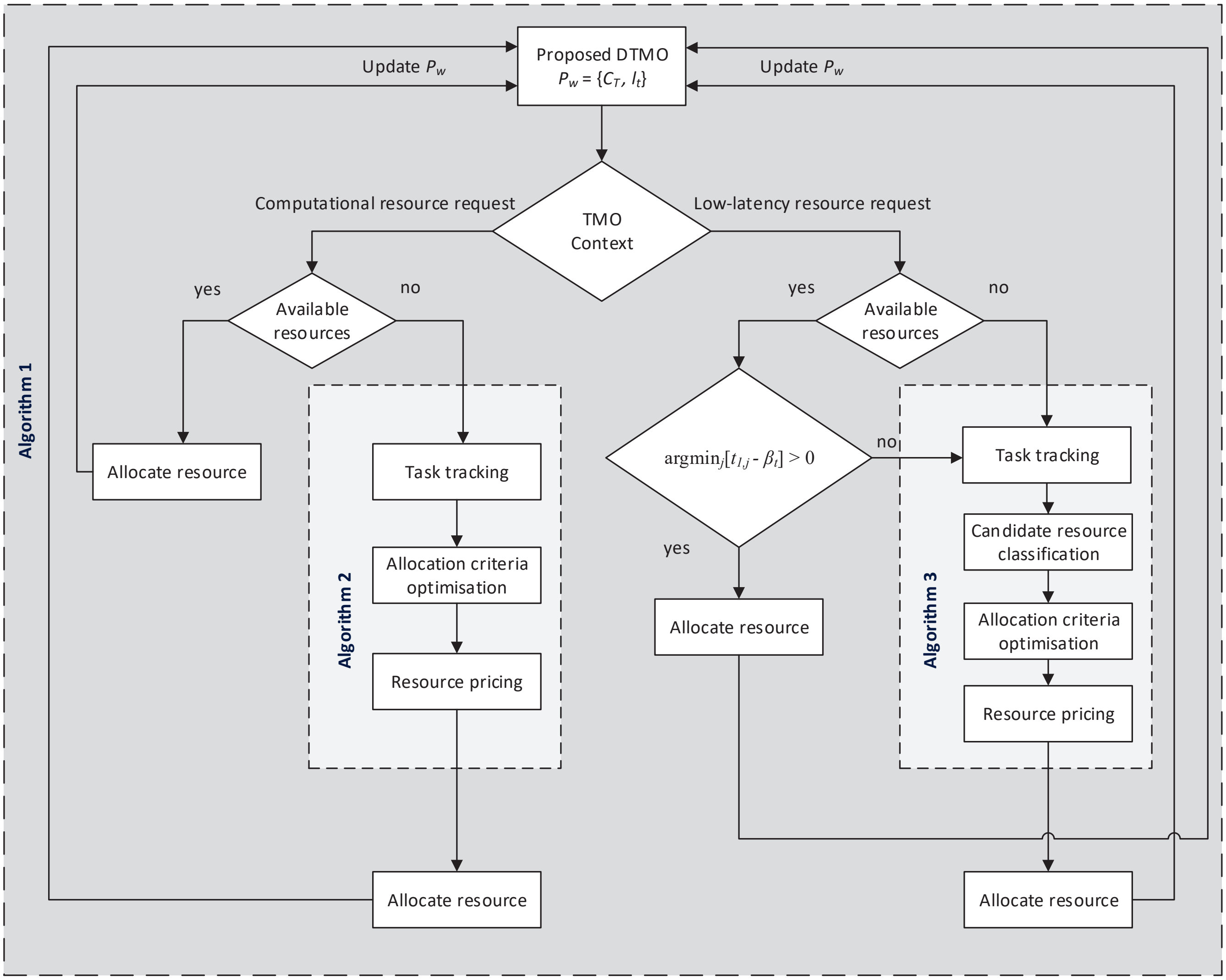

Fig. 2. The flowchart of the proposed DTMO framework

the conventional scheme in terms of the percentage of cloud requests fulfilled with increasing number of LPP. The conventional scheme utilises traditional cloud computing paradigm to acknowledge the requests for resource allocation, whereas, the proposed scheme utilises cognitive resource sharing model for distributed resource allocation. It can be observed from the results that with exponentially growing number of LPP, the efficiency of the conventional scheme is drastically affected. Therefore, its utilisation in 5G networks is expected to result in performance bottleneck and latency related issues. In contrast, the proposed scheme outperforms the conventional scheme and provides softness in decision making to allocate resources efficiently. It is observed from the results that as the percentage of available resources that the participating LPP have agreed to share increases, the proposed scheme significantly enhances the efficiency of fulfilment of resource allocation requests. Hence, the proposed framework provides an efficient model for resource sharing in future generation of wireless networks.

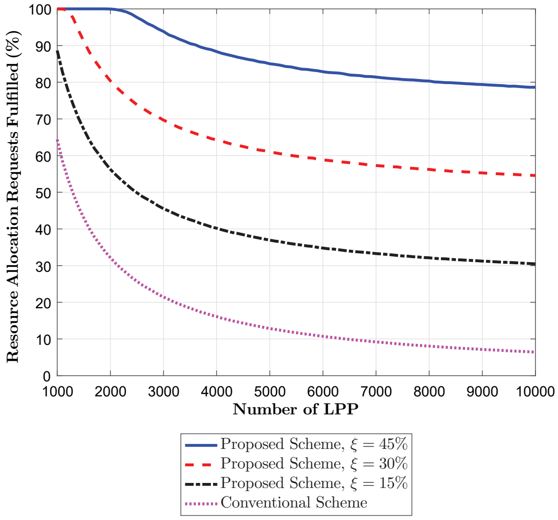

Fig. 3. Percentage of cloud requests fulfilled by the proposed and conventional schemes with increasing number of local processing platforms (LPP)

IV. Conclusion

In any network of connected devices, processing capacity and latency have always been major concerns for researchers and practitioners. The number of connected devices are expected to grow exponentially in the next coming decades, creating a new paradigm shift; where the aforementioned concerns will become major optimization criteria. Due to the growing demands for resources, depending on the enterprise cloud, is no longer a feasible solution. Edge Computing is a promising paradigm is this context, and complies with future developments such as 5G technologies. The framework proposed in this paper intended to integrate the advancement of edge computing resource requirement schemes as well as the resource allocation schemes found in the literature for enterprise cloud; to attain a universal resource allocation framework for IoT . The assumption has taken into consideration that the enterprise cloud operating system supports bi-directional resource sharing from local processing platform with heterogeneous device properties. To obtain the proposed resource allocation framework to be more cost-effective, auction mode of resource sharing agreement will be considered for future works, alongside investigating the feasibility of a unified operating system compatible with cloud resource sharing within heterogeneous computing devices.

References

[1] A. Botta, W. De Donato, V. Persico, and A. Pescapé, "Integration of cloud computing and internet of things: a survey," *Future Generation Computer Systems*, vol. 56, pp. 684–700, 2016.

[2] Z. Wen, R. Yang, P. Garraghan, T. Lin, J. Xu, and M. Rovatsos, "Fog orchestration for internet of things services," *IEEE Internet Computing*, vol. 21, no. 2, pp. 16–24, Mar 2017.

[3] L. Wang, G. Von Laszewski, A. Younge, X. He, M. Kunze, J. Tao, and C. Fu, "Cloud computing: a perspective study," *New Generation Computing*, vol. 28, no. 2, pp. 137–146, 2010.

[4] J. H. Abawajy and M. M. Hassan, "Federated internet of things and cloud computing pervasive patient health monitoring system," *IEEE Communications Magazine*, vol. 55, no. 1, pp. 48–53, 2017.

[5] W. He, G. Yan, and L. Da Xu, "Developing vehicular data cloud services in the IoT environment," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1587–1595, 2014.

[6] S. Rusitschka, K. Eger, and C. Gerdes, "Smart grid data cloud: A model for utilizing cloud computing in the smart grid domain," in *2010 First IEEE International Conference on Smart Grid Communications (SmartGridComm)*. IEEE, 2010, pp. 483–488.

[7] L. Jiang, L. Da Xu, H. Cai, Z. Jiang, F. Bu, and B. Xu, "An IoT-oriented data storage framework in cloud computing platform," *IEEE Transactions on Industrial Informatics*, vol. 10, no. 2, pp. 1443–1451, 2014.

[8] X. Zheng, P. Martin, K. Brohman, and L. Da Xu, "CLOUDQUAL: a quality model for cloud services," *IEEE transactions on industrial informatics*, vol. 10, no. 2, pp. 1527–1536, 2014.

[9] C. You, K. Huang, H. Chae, and B. H. Kim, "Energy-efficient resource allocation for mobile-edge computation offloading," *IEEE Transactions on Wireless Communications*, vol. PP, no. 99, pp. 1–1, 2016.

[10] A. Ahmed and E. Ahmed, "A survey on mobile edge computing," in *2016 10th International Conference on Intelligent Systems and Control (ISCO)*. IEEE, 2016, pp. 1–8.

[11] F. Bonomi, R. Milito, J. Zhu, and S. Addepalli, "Fog computing and its role in the internet of things," in *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. ACM, 2012, pp. 13–16.

[12] S. Yi, C. Li, and Q. Li, "A survey of fog computing: concepts, applications and issues," in *Proceedings of the 2015 Workshop on Mobile Big Data*. ACM, 2015, pp. 37–42.

[13] Department for Culture, Media & Sport, "Next Generation Mobile Technologies: A 5G strategy for the UK," https://www.gov.uk/government/publications/next-generation-mobile-technologies-a-5g-strategy-for-the-uk, Policy paper, 2017.

[14] Department for Culture, Media & Sport, "UK Digital Strategy," https://www.gov.uk/government/publications/uk-digital-strategy, Policy paper, 2017.

[15] Y. Jararweh, A. Doulat, O. AlQudah, E. Ahmed, M. Al-Ayyoub, and E. Benkhelifa, "The future of mobile cloud computing: integrating cloudlets and mobile edge computing," in *2016 23rd International Conference on Telecommunications (ICT)*. IEEE, 2016, pp. 1–5.

[16] Y. C. Hu, M. Patel, D. Sabella, N. Sprecher, and V. Young, "Mobile edge computing-a key technology towards 5G," *ETSI White Paper 11, ETSI*, 2015.

[17] M. Du and F. Li, "ATOM: Efficient tracking, monitoring, and orchestration of cloud resources," *IEEE Transactions on Parallel and Distributed Systems*, vol. PP, no. 99, pp. 1–1, 2017.

[18] A. Amjad, M. Patwary, A. Griffiths, and A.-H. Soliman, "Characterization of field-of-view for energy efficient application-aware visual sensor networks," *IEEE Sensors Journal*, vol. 16, no. 9, pp. 3109–3122, 2016.

[19] K. Chard and K. Bubendorfer, "Co-operative resource allocation: Building an open cloud market using shared infrastructure," *IEEE Transactions on Cloud Computing*, 2016.

[20] K. Kumar and Y.-H. Lu, "Cloud computing for mobile users: Can offloading computation save energy?" *Computer*, vol. 43, no. 4, pp. 51–56, 2010.

[21] M. Asaduzzaman, R. Abozariba, and M. Patwary, "Spectrum sharing optimization and analysis in cellular networks under target performance and budget restriction," in *2017 IEEE 85th Vehicular Technology Conference (VTC Spring)*, June 2017, pp. 1–7.

[22] K. Chard and K. Bubendorfer, "High performance resource allocation strategies for computational economies," *IEEE Transactions on Parallel and Distributed Systems*, vol. 24, no. 1, pp. 72–84, 2013.

[23] K. Chard, K. Bubendorfer, and P. Komisarczuk, "High occupancy resource allocation for grid and cloud systems, a study with drive," in *Proceedings of the 19th ACM International Symposium on High Performance Distributed Computing*. ACM, 2010, pp. 73–84.