information

Article

Distributed Edge Computing for Resource Allocation in Smart Cities Based on the IoT

Omar Abdulkareem Mahmood [1], Ali R. Abdellah [2,*], Ammar Muthanna [3] and Andrey Koucheryavy [3]

[1] Department of Communications Engineering, College of Engineering, University of Diyala, Baquba 32001, Iraq; omar_abdulkareem@uodiyala.edu.iq
[2] Electrical Engineering Department, Al-Azhar University, Qena 83513, Egypt
[3] Department of Telecommunication Networks and Data Transmission, the Bonch-Bruevich Saint-Petersburg State University of Telecommunications, 193232 St. Petersburg, Russia; muthanna.asa@spbgut.ru (A.M.); akouch@mail.ru (A.K.)
* Correspondence: alirefaee@azhar.edu.eg

Abstract: *Smart cities using the Internet of Things (IoT) can operate various IoT systems with better services that provide intelligent and efficient solutions for various aspects of urban life. With the rapidly growing number of IoT systems, the many smart city services, and their various quality of service (QoS) constraints, servers face the challenge of allocating limited resources across all Internet-based applications to achieve an efficient per-formance*. The presence of a cloud in the IoT system of a smart city results in high energy con-sumption and delays in the network. Edge computing is based on a cloud computing framework where computation, storage, and network resources are moved close to the data source. The IoT framework is identical to cloud computing. The critical issue in edge computing when executing tasks generated by IoT systems is the efficient use of energy while maintaining delay limitations. In this paper, we study *a multicriteria optimization approach for resource allocation with distributed edge computing in IoT-based smart cities*. We present a three-layer network architecture for IoT-based smart cities. *An edge resource allocation scheme based on an auctionable approach is proposed to ensure efficient resource computation* for delay-sensitive tasks.

Keywords: 5G; IoT; edge computing; auctionable approach; resource allocation; smart city

Citation: Mahmood, O.A.; Abdellah, A.R.; Muthanna, A.; Koucheryavy, A. Distributed Edge Computing for Resource Allocation in Smart Cities Based on the IoT. *Information* **2022**, *13*, 328. https://doi.org/10.3390/info13070328

Academic Editors: Lorenzo Mucchi, Stefano Caputo and Lorenzo Biotti

Received: 7 June 2022
Accepted: 4 July 2022
Published: 7 July 2022

1. Introduction

Smart cities are technologically advanced metropolitan areas that use information and communication technology to solve their particular challenges and to help implement sustainable social, economic, or environmental development. In recent years, transforming urban areas into smart cities has become an objective for many cities worldwide, which have developed specific strategies that can be used in different ways to achieve this goal. Deploying 5G technology as part of a smart city approach enhances productivity and the standard of living and reduces costs and resource consumption. The significant advantages of intelligent services are that they provide more accommodation, shared infrastructure, resources, and flexibility, improved quality of service (QoS), and quality of experience (QoE) methods, depending on the required service and infrastructure location. These services improve citizens' quality of life and provide a better user experience [1,2].

IoT systems at home benefit energy demand and make life more comfortable and enjoyable. In an IoT-based smart city, many facilities are controlled by IoT devices, and the various resources required to run the services are deployed using distant cloud services. The QoS conditions for deployment of diverse, intelligent city services include excellent computation capability, exact delay limitations, efficient power use, etc.; i.e., there is a hard limit on the tolerable delay in virtual reality. In contrast, in video monitoring, the unavoidable delay is adaptable. Moreover, if the processing of a task in an intelligent traffic application exceeds the desired time, an accident may occur. IoT-based smart cities comprise

a variety of use cases, such as smart home IoT systems that have beneficial effects on energy consumption and provide a more relaxed and enjoyable life. Smart cities use many IoT systems to control many facilities, and the various resources required to run the facilities are stored on remote cloud servers. Intelligent buildings help improve occupant comfort and reduce unnecessary energy consumption, e.g., in banks, businesses, shopping malls, restaurants, universities, etc. Intelligent medical systems permit physicians to observe diseases by utilizing smart devices anywhere. This process lowers the cost of medical treatment and enhances the physician and patient experience. Intelligent implementation of transportation, vehicle identification, and route monitoring improve cities' transportation network security and capability [3–6].

The growth of IoT systems and the variety of intelligent city-based IoT applications, with their diverse QoS requirements, make it difficult to operate these systems efficiently [7]. The large amounts of information produced by these systems make withdrawing money from a remote data center costly and inefficient. The information produced by IoT systems and transmitted over the internet to a distant cloud requires an abundance of bandwidth and power in the IoT network. Furthermore, this leads to an increased load on the cellular network as these systems access cloud servers. Using core cloud services over a cellular network can result in restrictions for IoT application areas requiring lower latency and higher efficiency. This highlights the significance of edge computing patterns with MEC quality. The basic idea of MEC is to increase cloud capacity at the edges of mobile networks. Theoretically, this can be achieved by locating storage and computing resources at the edge of the radio access network (RAN) on demand; some of the capabilities provided by the cloud are shifted to these extra resources at the edge. Standard features of MEC technology include extremely low latency, location information, and background information about the network. The typical cloud computing model encounters difficulties, such as communication costs and undesirable delays, because of the restricted computing capacity of intelligent IoT systems and the remotely accessed cloud server. These challenges affect delayed operations and applications involving IoT sensing devices. One primary concern is to avoid long delays during the execution of IoT systems due to the problem of information congestion in IoT-enabled smart cities [8–10].

The RAN is an essential element of a wireless telecommunications system connecting various devices to other network parts through a radio link. The RAN connects client devices, such as cell phones, computers, or remotely controlled machines, via a fiber-optic or wireless backhaul link. This connection links to the core network, which manages subscriber information and location data, among other things [11]. Modern computational techniques are an appropriate solution in this context, as they allow storage and computational resources to be moved closer to IoT systems, as represented by base stations (BSs). The edge computing used in UAVs [12], a fundamental level in the IoT framework, reduces the delay between IoT systems and the computational backend infrastructure (mini-cloud). The UAV level adjacent to the IoT system offers more minor delays; however, it offers fewer resources than the use of BSs and a mini-cloud. Therefore, an effective, modern edge resource allocation system ensuring QoS for various smart city use cases is required. Auctionable methods are appropriate for dealing with the difficulty of allocating multiple conditional resources. Allocating resources in environmental IoT systems requires numerous entities, such as virtual machines (VMs), to complete the most significant tasks and undertake VM distribution. Moreover, the balance point of the simulation game may be superior resource allocation for specific tasks and VMs, where original resource allocation optimization occurs locally at each edge's site, balancing the minimization of energy consumption and delay for QoS assurance. A distributed technique can be applied to forward the overflow workload, thus balancing resource usage and achieving superior resource management [13,14].

Edge computing techniques are one prospective solution as they allow computing and storage resources to be moved closer to IoT devices. The middle layer in IoT systems is edge computing, which fills the delay gap between IoT systems and the bottom-layer cloud computing framework. The edge layer connects to the IoT devices and offers lower service

delay, but it has fewer resources than a remote cloud. Therefore, an efficient edge resource allocation method is needed to guarantee the QoS of various innovative city applications. Auction-based methods are one potential approach when investigating the complexity of multicriteria resource allocation [15].

The contributions of this article can be summarized as follows:

1. We used a three-layer network structure for IoT-based smart cities in this work. The first layer was the IoT layer, consisting of IoT systems with various smart city implementations, and the second layer contained the edge layer. The third layer was the cloud layer.
   (1) Controllers were deployed at the center or the edge of UAVs to obtain resource allocation results for tasks and VMs.
   (2) For delay-sensitive tasks in smart cities relying on the IoT, an auction-based approach for allocating edge resources is proposed to minimize the energy consumed and computation delay for tasks.
   (3) A comparative study of the proposed method and existing techniques was undertaken to illustrate the efficiency of the suggested approach.
   (4) Simulation results indicated that the proposed method outperformed current techniques.
2. The paper is structured as follows: Section 2 reviews the related literature, Section 3 presents the system structure and problem formulation, Section 4 introduces the proposed approach, and the simulation results are discussed in Section 5. Finally, Section 6 concludes the paper.

2. Literature Review

Many researchers have proposed edge solutions for resource allocation problems, including energy consumption and service delays, in smart cities based on IoT systems to make people's lives comfortable. Some of these solutions are described below.

Yang et al. [16] studied the effective use of energy for resource distribution in a UAV-based MEC network and solved the network's energy reduction issues. Energy efficiency in a UAV-based MEC was investigated in [17] through a functional migration method and route planning. In [18], edge resources were allocated to minimize average latency while numerous IoT systems powered multiple smart city facilities and met the edge server capacity limits. Another article [19] proposed an algorithm for offloading tasks with low complexity and balancing energy efficiency based on two-sided correspondence in an IoT system; the suggested method is innovative in dynamically balancing energy efficiency and delay and stably offloading tasks with low complexity.

Abdullah et al. [20] investigated the influence of the edge network's computation offloading framework on original applications and proposed a light-effect migration-based framework for computation offloading. Another article [21] proposed a resource-based task allocation method and executed and examined it to obtain enhanced efficiency in a heterogeneous multi-cloud network. The suggested task distribution method reduced the consumed energy and decreased the time between tasks. A joint algorithm for consumer choice and resource allocation in MEC has been proposed to optimize energy capacity [22]. MEC offers cloud services at the edge of the cellular network. With ultra-low latency and significant bandwidth, the use of edge computing to support IoT systems is the backbone for advancing intelligent systems and use cases, such as smart homes, competent healthcare, intelligent traffic control, intelligent farming, and smart cities [23].

Anagnostopoulos et al. [24] examined specific techniques for offloading computational power at the network edge to self-computing nodes, a context characterized by insufficient resources and many restrictions that can be violated when performing analytical tasks. Chen et al. [25] proposed a model that could improve the allocation of system resources and efficiently optimize the objective function and customer satisfaction for computation offloading methods in edge computing networks, considering UAV transmission and clean energy.

Attiya et al. [26] suggested another task scheduler for managing IoT application tasks using a CCE. Specifically, they suggested a new hybrid swarm intelligence method, based on a modified manta ray foraging optimization (MRFO) algorithm and the salp swarm algorithm (SSA), to process the scheduling of IoT tasks in cloud computing. Sahraoui et al. [27] discussed the role of IoT systems in social relationship management, including the issue of the bursting of social relationships in IoT systems, and reviewed the suggested ASI-based solutions, such as social-oriented machine-learning and deep-learning techniques. A modified Harris hawks optimization (HHO) algorithm using simulated annealing (SA) for scheduling jobs in the cloud environment was presented in [28]. Iman El-Dessouki et al. [29] focused on combining a smart grid with smart cities with 5G technology and demonstrated the benefits of network slicing and of adopting virtual MPN systems for smart cities. Aamir Anwar et al. [30] proposed a framework for achieving an optimal solution for smart, secure parking that has broader implications for applying 5G in smart cities.

3. System Structure and Problem Formulation

This study investigated a multicriteria optimization issue in an IoT-based smart city to minimize energy and delay. Here, we first introduce a three-layer network structure for IoT-based smart cities. Then, we develop an auctionable edge resource allocation method to guarantee low computational delay with energy-saving benefits for delay-sensitive applications in IoT systems.

3.1. System Structure

Figure 1 shows a three-layer network structure in an IoT-enabled smart city.

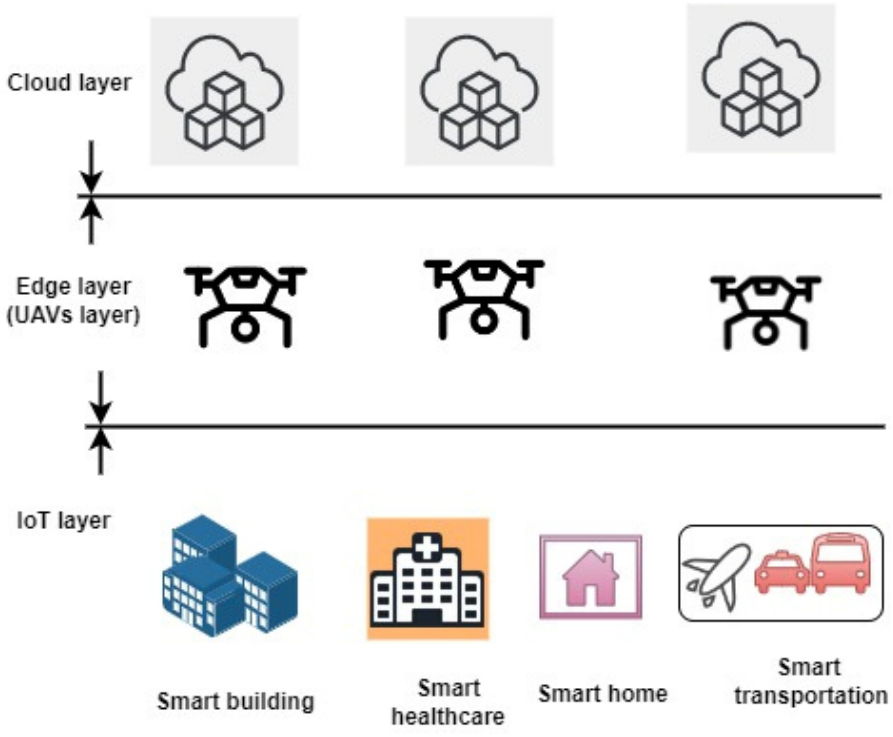

Figure 1. The network structure for IoT-enabled smart city.

The bottom layer comprises IoT systems that manage ongoing tasks created by the IoT systems' changing sizes. The IoT systems transmit QoS requests and task characteristics, such as the size of the task and the time limit, to the UAVs. In addition, the restricted potential computation power and short operating period of the IoT systems result in the tasks being offloaded to the UAVs to minimize energy consumption and computational time.

The UAV layer is near the IoT layer and includes VMs and controllers responsible for resource allocation, energy and delay analysis, and decision making.

The top layer is the cloud layer (mini-cloud), which consists of VMs, and the controller comprises three subcomponents: resource distribution, energy and delays analysis, and decision making.

Figure 2 depicts the resource allocator and evaluates the accessible resources at the (drone) edge and in the cloud layers for allocating tasks created via an IoT system to suitable VMs. The power and delay analysis unit assess whether the desired task has been completed with the lowest power consumption and delay ewhile satisfying the time limit. The decision-maker can utilize the suggested auction-enabled method to choose a suitable VM for tasks.

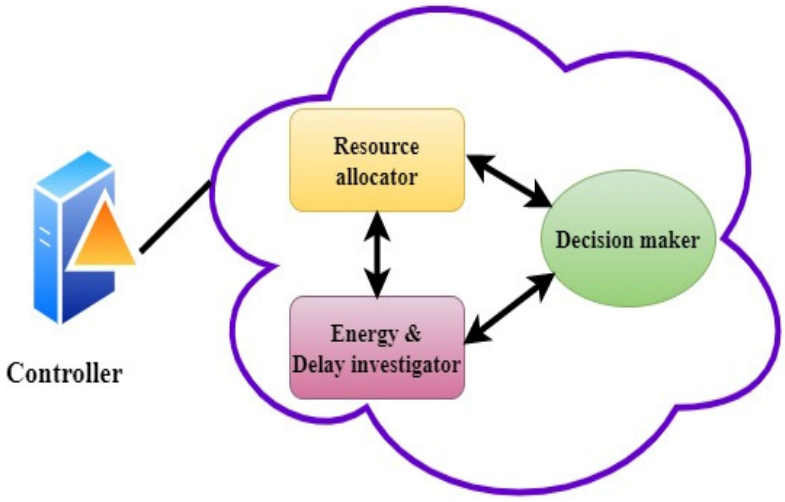

Figure 2. Controller elements.

The drone-edge and cloud layers send the controller VM data, for example, computation capabilities and power consumption. The controller utilizes the information non-task and the VM data to allocate the task to a suitable VM. Every controller in an IoT system gathers information via its insurance region and operates through drone or cloud layers. After obtaining the task, the controller takes the subsequent steps: the task will be rejected if the delay for the task outcomes does not satisfy the time conditions; otherwise, the controller transmits the task to the appropriate preference queue according to its preference layer; i.e., the deadline. For each task, the controller checks whether the accessible resources at the edge level are adequate to perform the task or otherwise checks other preference queues. The task is accomplished if the requirement is satisfied, considering the delays and consumed power for the corresponding VM. If not, the task is transmitted to the cloud level for implementation.

3.2. Problem Formulation

All intelligent-city use cases involve massive quantities of IoT systems. Assume that there are n IoT devices and suppose that numerous delay-sensitive tasks at the IoT level must be addressed locally (i.e., in the IoT system) or at the UAV or cloud level. Table 1 lists the different symbols used in this article.

A task (U_i) produced in an IoT system (D_i), where i = 1, 2,..., n, has the following characteristics: task size (U_{szi}), ($E_{i.l}$) energy consumed/CPU cycle, computational capacity (f_i) in terms of CPU cycles/second, and deadline (dl_i). This task information is sent to the controller, which undertakes resource allocation to reduce energy consumption and delay simultaneously.

Table 1. List of symbols.

Symbol	Description
U_i	Task created by device Di
U_{szi}	Task size of Ui
U_{Ei}	Energy consumption of Di/CPU cycle
f_i	Computational capacity of Di
T_i^l	Execution time of Ui (local)
$d_{i.l}$	Calculation delay of Ui (local)
$E_{i.l}$	Energy consumption of Ui (local)
dl_i	Deadline of Ui
$T_{tr.i}$	The transmission time of Ui
$f_{i.link}$	Calculation of link capacities in network
f_j	Calculation of VMj capacity
$T_{i.j}$	Ui execution time through VMj
$edge\_d_{i.j}$	Computational delay of Ui (edge)
E_j	Energy consumption of VMj/CPU cycle
$E_{i.j}$	Energy consumption of VMj for Ui
$E_{tr.i}$	Transmission energy for Ui
E_{max}	Maximum acceptable consumed energy
d_{max}	Maximum delay

There are three possible cases for calculation, depending on where the calculation occurs: the local case (in the IoT layer), edge case (in the UAV layer), or cloud case (in the mini-cloud layer). The controller maintains two main queues: Q_E for the UAV level and Q_C for the cloud level. Let (L_{dl}) and (H_{dl}) be the lowest and highest thresholds through to the deadline.

Assume VM_{edge} and VM_{cloud} are virtual machines (VMs) in the edge and cloud layers. The computation capability describes each VMj $\in$ {VM_{edge} or VM_{cloud}} f_j in terms of CPU cycles/second and consumed energy E_{ij} for each CPU cycle. The calculation of delay and the consumed power is explained below for each case.

With the local method, the IoT device performs the task calculation. The completion time $T_{i.l}$ of U_i is

$$T_{i.l} = \frac{U_{szi}}{f_i} \tag{1}$$

Now, the computing delay $d_{i.l}$ for device D_i is

$$d_{i.l} = ST_{i.l} + T_{i.l} \tag{2}$$

$ST_{i.l}$ represents the beginning time of U_i for device D_i. The consumed energy $E_{i.l}$ for D_i due to U_i is:

$$E_{i.l} = U_{szi} \times U_{Ei} \tag{3}$$

In UAV or cloud mode, the computations are performed on a single VM at the edge or cloud level. Suppose that the task (U_i) is offloaded toward the UAVs at the edge level. The transmission time ($T_{tr.i}$) of the task (U_i) is:

$$T_{tr.i} = \frac{U_{size}}{f_{i.link}} \tag{4}$$

$f_{i.link}$ is the transmission capability of the communication channel, and the implementation time $T_{i.j}$ of U_i on $VMj \in$ ES is

$$T_{i.j} = \frac{U_{szi}}{f_j} \tag{5}$$

Therefore, the computational delay $d_{i.j}$ of Ui due to task offloading is

$$d_{i.j} = T_{t.i} + \left(ST_{i.j} + T_{i.j} \right) \tag{6}$$

Here, $ST_{i.j}$ represents the starting time of U_i in VMj, supposing E_t to be the energy used to transmit information for each CPU cycle. The transmitted energy $E_{t.i}$ due to Ui is calculated as follows:

$$E_{tr.i} = E_t + U_{szi} \tag{7}$$

The computational energy $E_{i.j}$ of U_i in VMj is:

$$E_{i.j} = E_j + U_{szi} \tag{8}$$

Therefore, the overall used energy tot. $E_{i.j}$ as a result of task offloading to the edge level is:

$$tot\_ E_{i.j} = E_{tr.i} + E_{i.j} \tag{9}$$

Assume that h_i is the binary predictor that states whether a task is performed locally or by offloading to UAVs or the cloud level:

$$h_i = \begin{cases} 0 \text{ Local case} \\ 1 \text{ Edge case or Cloud case} \end{cases} \tag{10}$$

The delay $edge\_d_{i.j}$ and the expended energy $E_{i.j}$ of task U_i executed on VM_j are:

$$d_{i.j} = \left(1 - h_i \right) \times d_{i.l} + h_i \times edge\_d_{i.j} \tag{11}$$

$$E_{i.j} = \left(1 - h_i \right) \times E_{i.l} + h_i \times tot\_E_{i.j} \tag{12}$$

Further, the following restriction is supposed to be met:

$$d_{i.j} \leq dl_i \; and \; E_{i.j} \leq E_{max} \tag{13}$$

where E_{max} is the maximum power usage for the technique. The multiple criteria optimization issue is computed as one object ($y_{i.j}$) to minimize power and delays. It is developed as follows:

$$y_{i.j} = \propto \times \frac{d_{i.j}}{d_{max}} + \left(1 - \propto \right) \times \frac{E_{i.j}}{E_{max}} \tag{14}$$

where d_{max} represents the maximum delay, and α is the weight element related to the delay and consumed power. Likewise, for n devices, the number of loss functions y_t is computed as:

$$y_t = \sum_{vmj \in Q_E} \sum_{i=1}^{n} y_{i.j} \times x_{i.j} \tag{15}$$

in which $x_{i.j}$ is a binary variable that demonstrates whether U_i is performed on VMj or not. The purpose of this operation is to minimize y_t as follows:

$$Minimize \; y_t \tag{16}$$

4. Proposed Approach

The auctionable method will be one of the primary methods for resource allocation with a multiagent approach in future generations. The main advantage of auctionable methods is that they are not based on the utility of the nodes. Thus, they can achieve the de-

sired resource allocation without knowing the utility functions of the entities involved. The auctionable approach can help address significant research challenges in IoT environments, such as network entity obstacles, economic issues, decentralized resource allocation, and limited or missing information about network status and benefits. In auctionable methods, a central controller decides on the distribution based on the recommendations submitted by the agents participating in the auctionable system. In this study, we developed a resource allocation approach using an auction-based method to minimize delays and consumed power in IoT systems.

We suppose that every VM_{edge} or VM_{cloud} operates as a proxy and can offer a proposal to carry out a task or represent an object. Each agent provides suggestions for tasks at the head of the queue. It is assumed that only one VM can assign a task. A loss function or proposal value determines the priority of a VM within a group of VMs that are eligible for allocation. Every VM receives a prize based on the speed with which it completes the task before the deadline. The controller decider can use the suggested auctionable approach to choose a group of capable VMs, and the winner performs the task to meet the goal.

A VM's capacity to perform a task Ui depends on the objective value developed in Equation (14). The VMs in VM_{edge} meet the deadline (dl_i) limitations for the task U_i and make offers in the auction. The offer value is similar to the loss value. For the VMs that make the offers, the reward profit received from performing the calculations for U_i is determined by the following equation:

$$R_{i.j} = 1/\beta^{dl_i - d_{i.j}} \tag{17}$$

Here, $\beta \neq 0$. The victor VM_j among the group of possible VMs that contribute to the auction is determined using the following equation:

$$sel\_VMj = \frac{R_{i.j}}{bids_{i.j}} \tag{18}$$

5. Simulation Result

In this study, we simulated the system models using MATLAB R2020a on an Intel® Core™ i7-10700K processor running at 5.10 GHz, with 16 GB RAM, GPU/NVIDIA Gtx 1080, and Windows 10/64 bit. Table 2 shows the detailed simulation parameters used in this article. The suggested auctionable method for distributing edge resources was compared in terms of two primary forms:

1. **Local processing**: task calculation is performed over the IoT system to produce the task.
2. **Edge offloading**: complete tasks created with the IoT systems are offloaded to the UAV level, utilizing a greedy method to implement tasks.

Table 2. Simulation parameters.

Parameter	Value
CPU cycles required by U_i	[10–30] M cycles
Energy consumption of D_i	10^{-7} joules/cycle
Computation capacity of D_i	[150–600] M cycles per second
Energy consumption of VMj	10^{-8} joules/cycle
Computation capacity of VMj	$[10^4$–$3 \times 10^4]$ M cycles per second
Computation capacity of the transmission link	$[10^2$–$2 \times 10^2]$ M cycles per second
Transmission energy	10^{-7} joules/cycle

The metrics used to test the system's operation included power usage and average delay. To calculate the capacity of the suggested method, we compared it with its competitors and studied different schemes, such as switching various tasks and VMs.

We supposed that the task arrival would support a Poisson distribution with an expected value of $\lambda = 0.8$. The execution as a function of the tasks implemented is illustrated in Figures 3 and 4. The number of VMs was fixed at about 30.

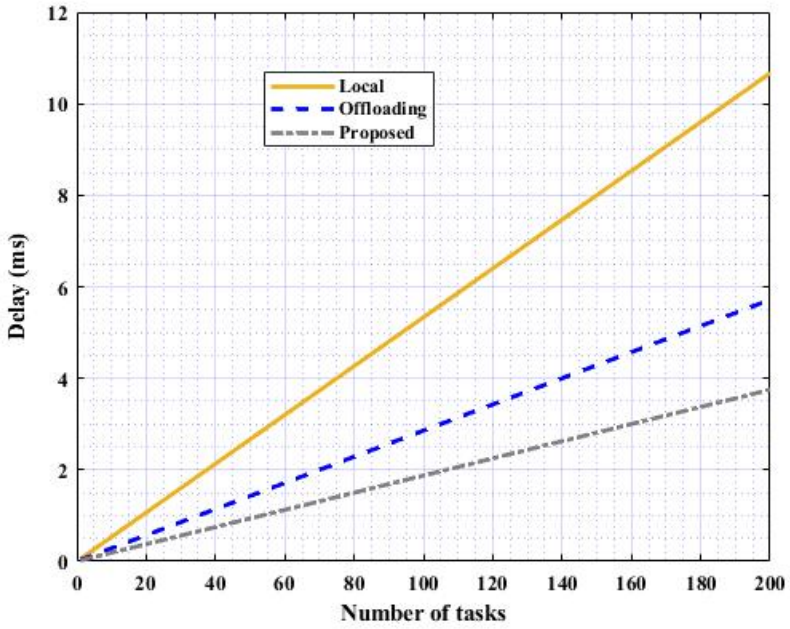

Figure 3. Performance vs. tasks—average delay.

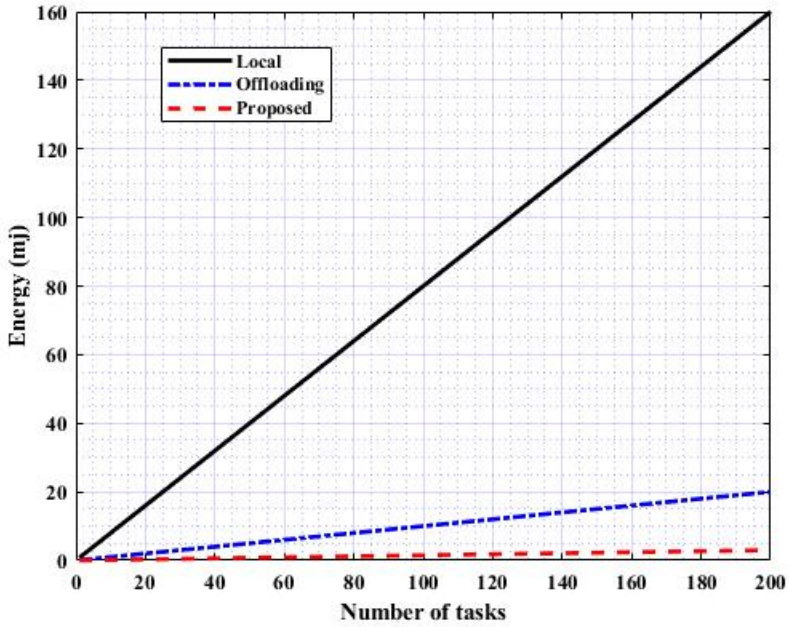

Figure 4. Performance vs. tasks—energy consumption.

Figure 3 displays the typical computational delays when the tasks were increased. It can be observed in Figure 3 that the delays also increased as the number of tasks increased. The delay in the local case was more significant than for the edge (UAV offloading) case and the suggested auction-enabled approach. The device's processing power was weak, so it was not easy to process many tasks.

The delay was low with the edge method (UAV offloading), but there was a bottleneck in the transmission delay. The number of VMs remained unchanged as many tasks were added, incrementally increasing the computation delay. However, the delay was lower in the suggested auction-based approach than in the greedy method (UAV offloading). The equilibrium between offloading tasks to the UAV level and local computation in the auction-based form maintained the same computational delay.

Figure 4 shows that the consumed energy increased with the number of tasks. The device had to consume a lot of energy to perform the tasks. The number of tasks increased the VM's operation time, thus increasing the consumed energy. Moreover, the VMs' timeout also entailed energy utilization. The proposed method consumed less energy in comparison to the other algorithms. Choosing a VM by relying on the target (or bid) value and reward value worked well in the auctionable approach.

Figures 5 and 6 show the operating system for the case with VMs at the edge level. At this point, the number of tasks was fixed at 80. The typical delay relating to the multiple criteria optimization issue is demonstrated in Figure 5.

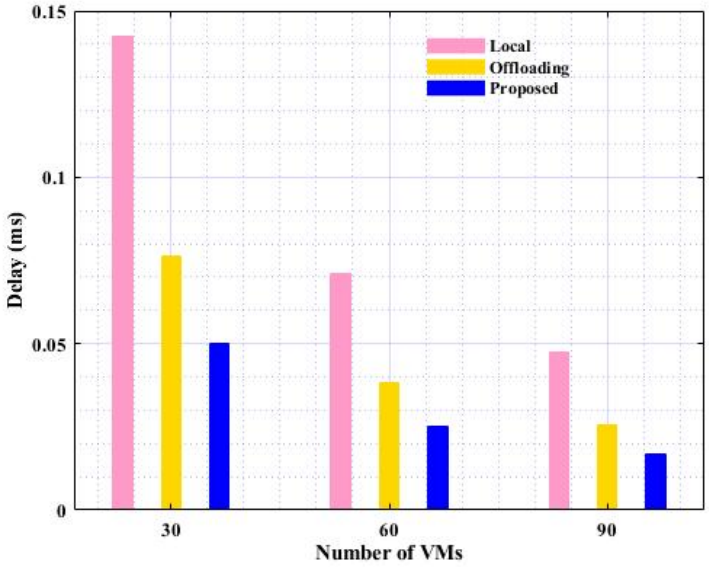

Figure 5. Performance vs. VMs—average delay.

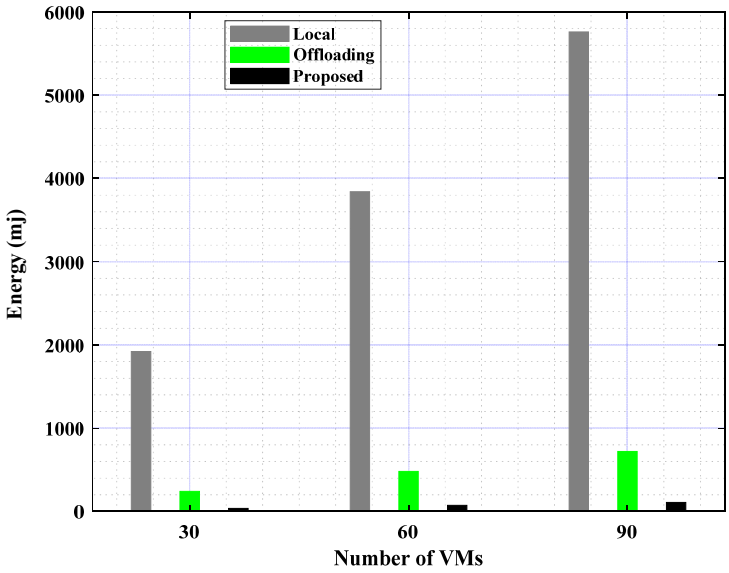

Figure 6. Performance vs. VMs—energy consumption.

It can be noted from Figure 5 that increasing the number of VMs decreased the computational delay. Nevertheless, the auctionable method with the greedy approach had a lower computational delay than the edge method (UAV offloading). Choosing a VM with an offer value defined by considering the system's purpose improved the suggested approach's efficiency.

Figure 6 shows the consumed energy based on the VMs. VMs increase their power usage as the effective task executions increase. Moreover, the energy spent by the significant number of inactive VMs contributes to the consumed energy. However, the suggested auctionable method consumed less power than the edge offloading method (UAVs) when using the greedy approach.

6. Conclusions

Delay-sensitive applications in the IoT era, such as smart grids and intelligent medicine, require a low computational delay. In addition, the energy needed to execute the diverse tasks created through IoT systems is an important problem for IoT networks. The edge computing represented by UAVs solves these problems in IoT networks, bringing the computations close to the information provider. This study described a three-layered network architecture (IoT, edge, and cloud) for IoT-based smart cities. Furthermore, an auctionable mechanism to overcome the problem of allocating IoT-based edge resources was developed. The auctionable tool was developed by considering the energy consumption and delay associated with VMs in task execution. The test results demonstrated that the suggested edge resource allocation outperformed the other methods. The average improvements for the proposed model compared to other methods in terms of the number of VMs vs. computational delay were 3.1% with 90 VMs and 9.2% with 30 VMs. Furthermore, the average improvement in terms of the number of tasks vs. the computational delay was 7% with 200 tasks. Moreover, the average improvement when using energy vs. the number of tasks was 157% with 200 tasks, but the average improvements in terms of energy vs. the number of VMs were 188% with 30 VMs and 565% with 90 VMs.

7. Future Work

In the future, this simulation can be further developed by adding air offloading algorithms based on the load and volume of the UAV's engine, its height above the ground, and the area it can cover. A networked group of UAVs could also be created, where a central UAV and four secondary drones connected to it undertake data offloading using the functions of the 5G network.

Author Contributions: Conceptualization, A.R.A. and O.A.M.; methodology, A.R.A.; software, A.R.A. and O.A.M.; validation, A.M., A.K. and A.R.A.; formal analysis, O.A.M. and A.R.A.; investigation, A.M.; resources, A.K.; data curation, A.R.A. and A.K.; writing—original draft preparation, A.R.A. and O.A.M.; writing—review and editing, A.R.A. and O.A.M.; visualization, A.R.A. and A.K.; supervision, A.K.; project administration, A.M. and A.K.; funding acquisition, A.K. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The article contains the data, which are also available from the corresponding author upon reasonable request.

Acknowledgments: This research was supported by Applied Scientific Research under the SPbSUT state assignment 2022.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Beştepe, F.; Yildirim, S.Ö. Acceptance of IoT-based and sustainability-oriented smart city services: A mixed methods study. *Sustain. Cities Soc.* **2022**, *80*, 103794. [CrossRef]
2. Kuru, K. Planning the Future of Smart Cities with Swarms of Fully Autonomous Unmanned Aerial Vehicles Using a Novel Framework. *IEEE Access* **2021**, *9*, 6571–6595. [CrossRef]
3. Alam, T. Cloud-Based IoT Applications and Their Roles in Smart Cities. *Smart Cities* **2021**, *4*, 64. [CrossRef]
4. Khedkar, S.P.; Canessane, R.A.; Najafi, M.L. Prediction of Traffic Generated by IoT Devices Using Statistical Learning Time Series Algorithms. *Wirel. Commun. Mob. Comput.* **2021**, *2021*, 5366222. [CrossRef]
5. Syed, A.; Sierra-Sosa, D.; Kumar, A.; Elmaghraby, A. IoT in Smart Cities: A Survey of Technologies, Practices and Challenges. *Smart Cities* **2021**, *4*, 24. [CrossRef]
6. Asad, M.; Basit, A.; Qaisar, S.; Ali, M. Beyond 5G: Hybrid End-to-End Quality of Service Provisioning in Heterogeneous IoT Networks. *IEEE Access* **2020**, *8*, 192320–192338. [CrossRef]
7. Abdellah, A.R.; Mahmood, O.A.; Kirichek, R.; Paramonov, A.; Koucheryavy, A. Machine Learning Algorithm for Delay Prediction in IoT and Tactile Internet. *Future Internet* **2021**, *13*, 304. [CrossRef]
8. Yar, H.; Imran, A.; Khan, Z.; Sajjad, M.; Kastrati, Z. Towards Smart Home Automation Using IoT-Enabled Edge-Computing Paradigm. *Sensors* **2022**, *21*, 4932. [CrossRef]
9. Huda, S.A.; Moh, S. Survey on computation offloading in UAV-Enabled mobile edge computing. *J. Netw. Comput. Appl.* **2022**, *201*, 103341. [CrossRef]
10. Abdellah, A.R.; Mahmood, O.A.; Koucheryavy, A. Delay prediction in IoT using Machine Learning Approach. In Proceedings of the 2020 12th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT), Brno, Czech Republic, 5–7 October 2020; pp. 275–279. [CrossRef]
11. Ibrahim, K.; Sadkhan, S.B. Radio Access Network Techniques Beyond 5G Network: A brief Overview. In Proceedings of the 2021 International Conference on Advanced Computer Applications (ACA), Maysan, Iraq, 25–26 July 2021; pp. 96–100. [CrossRef]
12. Alzaghir, A.; Abdellah, A.R.; Koucheryav, A. Predicting energy consumption for UAV-enabled MEC using Machine Learning Algorithm. In *Internet of Things, Smart Spaces, and Next Generation Networks and Systems, NEW2AN 2021, ruSMART 2021, LNCS*; Koucheryavy, Y., Balandin, S., Andreev, S., Eds.; Springer: Cham, Switzerland, 2022; Volume 13158, pp. 297–309. [CrossRef]
13. Xu, J.; Liu, X.; Li, X.; Zhang, L.; Jin, J.; Yang, Y. Energy aware Computation Management Strategy for Smart Logistic System with MEC. *IEEE Internet Things J.* **2021**, *9*, 8544–8559. [CrossRef]
14. Liyanage, M.; Porambage, P.; Ding, A.Y.; Kalla, A. Driving forces for Multi-Access Edge Computing (MEC) IoT integration in 5G. *ICT Express* **2021**, *7*, 127–137. [CrossRef]
15. Middya, A.; Ray, B.; Roy, S. Auction-Based Resource Allocation Mechanism in Federated Cloud Environment: TARA. *IEEE Trans. Serv. Comput.* **2022**, *15*, 470–483. [CrossRef]
16. Yang, Z.; Pan, C.; Wang, K.; Shikh-Bahaei, M. Energy Efficient Resource Allocation in UAV-Enabled Mobile Edge Computing Networks. *IEEE Trans. Wirel. Commun.* **2019**, *18*, 4576–4589. [CrossRef]
17. Gong, C.; Wei, L.; Gong, D.; Li, T.; Feng, F. Energy-Efficient Task Migration and Path Planning in UAV-Enabled Mobile Edge Computing System. *Complexity* **2022**, *2022*, 4269102. [CrossRef]
18. Zhao, L.; Wang, J.; Liu, J.; Kato, N. Optimal Edge Resource Allocation in IoT-Based Smart Cities. *IEEE Netw.* **2019**, *33*, 30–35. [CrossRef]
19. Wei, Y.; Yang, H.; Wang, J.; Chen, X.; Li, J.; Zhang, S.; Huang, B. Delay and Energy-Efficiency-Balanced Task Offloading for Electric Internet of Things. *Electronics* **2022**, *11*, 839. [CrossRef]
20. Yousafzai, A.; Yaqoob, I.; Imran, M.; Gani, A.; Noor, R.M. Process Migration-Based Computational Offloading Framework for IoT-Supported Mobile Edge/Cloud Computing. *IEEE Internet Things J.* **2020**, *7*, 4171–4182. [CrossRef]
21. Mishra, S.K.; Mishra, S.; Alsayat, A.; Jhanjhi, N.Z.; Humayun, M.; Sahoo, K.S.; Luhach, A.K. Energy-Aware Task Allocation for Multi-Cloud Networks. *IEEE Access* **2020**, *8*, 178825–178834. [CrossRef]
22. Feng, H.; Guo, S.; Zhu, A.; Wang, Q.; Liu, D. Energy-efficient user selection and resource allocation in mobile edge computing. *Ad Hoc Netw.* **2020**, *107*, 102202. [CrossRef]
23. Mukherjee, A.; Ghosh, D.D.S.K.; Buyya, R. *Mobile Edge Computing*, 1st ed.; Springer: Berlin/Heidelberg, Germany, 2021; 616p.
24. Anagnostopoulos, C.; Aladwani, T.; Alghamdi, I.; Kolomvatsos, K. Data-Driven Analytics Task Management Reasoning Mechanism in Edge Computing. *Smart Cities* **2022**, *5*, 30. [CrossRef]
25. Chen, Z.; Xiao, N.; Han, D. Multilevel Task Offloading and Resource Optimization of Edge Computing Networks Considering UAV Relay and Green Energy. *Appl. Sci.* **2020**, *10*, 2592. [CrossRef]
26. Attiya, I.A.; Elaziz, M.A.; Abualigah, L.; Nguyen, T.N.; El-Latif, A.A.A. An Improved Hybrid Swarm Intelligence for Scheduling IoT Application Tasks in the Cloud. *IEEE Trans. Ind. Inform.* **2022**, *18*, 6264–6272. [CrossRef]
27. Dhelim, S.; Ning, H.; Farha, F.; Chen, L.; Atzori, L.; Daneshmand, M. IoT-Enabled Social Relationships Meet Artificial Social Intelligence. *IEEE Internet Things J.* **2021**, *8*, 17817–17828. [CrossRef]
28. Attiya, I.; Elaziz, M.A.; Xiong, S. Job Scheduling in Cloud Computing Using a Modified Harris Hawks Optimization and Simulated Annealing Algorithm. *Comput. Intell. Neurosci.* **2020**, *2020*, 3504642. [CrossRef]
29. El-Dessouki, I.; Saeed, N. Smart Grid Integration into Smart Cities. In Proceedings of the 2021 IEEE International Smart Cities Conference (ISC2), Manchester, UK, 7–10 September 2021; pp. 1–4. [CrossRef]
30. Anwar, A.; Saeed, N.; Saadati, P. Smart Parking: Novel Framework of Secure Smart Parking Solution using 5G Technology. In Proceedings of the 2021 IEEE International Smart Cities Conference (ISC2), Manchester, UK, 7–10 September 2021; pp. 1–4. [CrossRef]