Improving the QoS of Fog Nodes Using Delay Sensitive Task Offloading

Asrat Mulatu Beyene(✉)

BDA and HPC Center of Excellence, Addis Ababa Science and Technology University, Addis Ababa, Ethiopia
asrat.mulatu@aastu.edu.et

Abstract. Fog computing is an extension of cloud computing where services are provided at the edge of a network. With the growth of the Internet of Things (IoTs) different applications are emerging. Many of these applications are delay sensitive that demand better reliability. Different task scheduling algorithms were proposed to manage such applications under fog computing environments. Still, it is difficult to meet the required latency requirements of applications with existing algorithms. Therefore, in this paper, a delay sensitive offloading algorithm is proposed that considers two different criteria during scheduling – the expected deadline of a given task and the computational capability of nodes. The effectiveness of the proposed algorithm is evaluated by using two typical working scenarios of fog nodes where comparative analysis with two popular task offloading algorithms have been made. The overall result shows that the proposed algorithm can provide as much as 37% improvement in the average response time of tasks in IoT applications.

Keywords: Fog computing · Task scheduling · Internet of Things · Delay sensitive · QoS requirements

1 Introduction

In the emerging 5G technology, ultra-low latency and high reliability are demanded from IoT applications [1]. Many research works tried to explore the possibility of fog computing by having the fog layer between the cloud and IoT devices [2]. And others modified some aspect of IoT architectures to minimize delay [3]. Fog computing tried to achieve such requirements by making computations at the edge of the network. But, due to resource constraint, uneven distribution of IoT devices in the domain, and natural variety of tasks it demanded extra effort to fulfill it. Minimizing delay of services, utilizing resources effectively, and minimizing power consumption are still existing challenges [4].

For many applications, another entity executes tasks on behalf of IoT devices and return results to those devices [5]. Some tasks demanding high computing devices and others resource-constrained ones in fog computing environment. To solve this contradiction a technique called offloading is used. It is used to outsource tasks and entities that

M. A. Delele et al. (Eds.): ICAST 2020, LNICST 384, pp. 417–431, 2021.
https://doi.org/10.1007/978-3-030-80621-7_30

work together to achieve the final computational goal of applications [5]. Task offloading is attractive for emerging IoT and cloud computing applications in that it can occur among IoT devices, edge devices, and fog nodes.

Many mobile services have been introduced with the emergence of different smart devices in the market. Mobile devices are resource-constrained. And, due to their mobility it is, usually, essential to offload part of their tasks to the cloud. Furthermore, integration of machine learning algorithms is needed to attain the required smartness of many services such as image processing and online gaming [6]. Efficient task execution is necessary, especially, when intelligence and sophistication of tasks become higher. But, the capability of IoT devices usually can fulfill the resource requirements of tasks. In these cases, the tasks are offloaded to the middleware [7]. This middleware is the fog layer that contains computing resources such as fog devices, fog servers, and gateways. In some situations, it might be required to offload some tasks to the cloud server [5]. However, tasks are offloaded to fog nodes for storage in addition to computation [5, 8].

2 Related Works

As presented in Table 1 there are many research efforts in the area of fog computing [9, 12, 14, 16, 19, 20, 22, 23, 28], edge computing [1, 21, 24, 25, 30, 31, 34], cloud computing [4, 10, 11, 15], of their combinations among them like fog-cloud [17, 18, 37], fog-cloud-edge [29], or of with IoT [5, 6, 13, 26, 33, 35, 36]. Some these proposed scheduling algorithms [1, 10, 11, 21, 34], analytical models [13, 14], heuristic algorithms [12, 16, 19, 20, 25, 28], optimizations [17, 18, 24, 28, 33, 35], architectures/frameworks [9, 23, 29, 36, 37], intelligent algorithms [6, 30, 31, 33], dynamic [1, 9, 10, 21, 33] and on-demand algorithms [16, 23, 24, 26]. Many worked on task offloading [1, 5, 6, 9–11, 13, 16–21, 23–26, 28, 29, 31–37], few included data [24, 30], and yet very few worked only on data [22]. Many of the research works involve optimizing computational ressources [1, 4, 6, 9–11, 13, 16–26, 28–31], and some considered both computational network resources [22, 32–37]. From the works considered in this review one can understand that the area is a hot research area where scenario based improvement on existing task offloading algorithms is demanded [4, 5, 12, 15, 27].

In this work, task offloading algorithm is proposed among fog nodes to enable tasks to meet their deadline requirements based on delay requirements of the tasks and computational capabilities of nodes.

3 Selected System Architecture

The system architecture considered in this work has four layers with IoT devices in the first, lower-level fog nodes in the second, aggregate fog nodes and cloud servers in the third and fourth layers, respectively.

3.1 IoT Devices

Data sources, from which various forms of data are generated, like Wi-Fi sensors, actuators, and smart devices, lay in this layer. This layer consists of physical and virtual sensors, where any data generation device could fall into any of these groups.

Table 1. Summary of related works

Author	System considered	Algorithm name/type	Improvement focus	Offloading type
Sladana Josilo et al. [16]	Fog Computing	Game theoretic analysis based Decentralization Algorithm	Task	Computation
Christine Fricker et al. [14]	Fog computing	Analytical model	-	Load Balancing
Tongxiang Wang et al. [10]	Mobile cloud computing	Cooperative Multi-tasks Scheduling on ACO algorithm (CMSACO)	Task	Computation
Mika Jia et al. [11]	Distributed Cloud computing	Scheduling algorithm	Task	Computation
Ashkan Yousefpour et al. [13]	IoT-Fog-Cloud Scenario	Analytical model	Task	Computation
Xiaohui Zhao et al. [17]	Fog-cloud computing	Optimal energy consumption algorithm	Task	Computation
Jianbo Du et al. [18]	Fog-cloud computing	Low-complexity suboptimal algorithm	Task	Computation offloading and resource allocation
Qiliang Zhu et al. [19]	Fog computing	Task offloading policy	Task	Computation
Te-Yi Kan et al. [20]	Fog computing	Heuristic load distribution algorithm	Task	Computation
Lingfang Gao [21]	Mobile edge computing	Task offloading strategy & priority-based task scheduling algorithm	Task	Computation
Mohammed Al-khafajiy et al. [36]	Fog-IoT Computing	Generic IoT-Fog based architecture	Task	Computation & network
Tomislav Shuminoski et al. [37]	Mobile cloud-fog computing	5G-based Fog-Cloud Computing Framework	Task	Computation & network
Quang Duy La et al. [6]	Fog computing & IoT apps	AI based algorithm	Task	Computation

(*continued*)

Table 1. (*continued*)

Author	System considered	Algorithm name/type	Improvement focus	Offloading type
Mithun Mukherjee et al. [22]	Fog computing	Quadratic-ally constraint quadratic programming (QCQP)	Data	Computation & network
Yeonjin Jin et al. [23]	Fog computing	On-demand offloading architecture	Task	Computation
Ibrahim Alghamdi et al. [24]	Mobile edge computing	Optimal offloading rule	Task & data	Computation
Jiuyun Xu et al. [25]	Mobile edge computing	Branch-and-Bound Algorithm	Task	Computation
Huaiying Sun et al. [26]	Generic IoT-Fog-Cloud system	Energy & time efficient computation offloading and resource allocation(ETCORA) algorithm	Task	Computation
Hyame Assem Alameddine et al. [1]	Edge Computing	Dynamic Task Offloading and Scheduling	Task	Computation
Ashkan Yousefpour et al.[9]	Fog Computing	QoS-aware Dynamic Fog Service Provisioning (QDFSP) Framework	Task	Computation
Yijin Pan et al. [29]	Hierarchical Fog-Cloud-Edge Computing System	Hierarchical Fog-Cloud-Edge Radio Access Networks (C-RANs) Architecture	Task	Computation
Jieun Kang et al. [30]	Edge computing	Intelligent offloading model	Task & data	Computation
Md Delowar Hossain et al.' [31]	Densely Deployed Small-Cell Networks with Multi-Access Edge Computing	Fuzzy Based Collaborative Offloading (FCTO) Scheme	Task	Computation

(*continued*)

Table 1. (*continued*)

Author	System considered	Algorithm name/type	Improvement focus	Offloading type
Sudip Misra et al. [33]	Cloud- Software Defined Fog-IoT	Detour - Dynamic Task Offloading	Task	Computation & network
Sladana Josilo [34]	Edge Computing	Resource allocation & scheduling scheme	Task	Computation & network
Mohamed K. Hussein et al. [35]	Fog-IoT Computing	Ant Colony Optimization	Task	Computation & network

3.2 Lower-Level Fog Nodes Layer

This layer contains fog nodes that are capable of performing computation and storage. It accepts requests generated form IoT devices that need better computational power.

3.3 Aggregate Fog Nodes Layer

This layer is composed of servers and gateways which control fog nodes located at the lower layer. These devices have more computational capability and storage capacity when compared to lower-level fog nodes. Those tasks that are deadline insensitive and computation-intensive are executed in this layer.

3.4 Cloud Layer

This layer contains cloud servers and data centers that provide networking, computational, and storage services. This layer executes or stores tasks received in two ways. The first is directly from IoT devices while the other is from aggregate fog nodes.

4 Proposed Task Offloading Algorithm

4.1 System Model

The proposed task offloading algorithm functions as follows. First, tasks generated from various IoT devices are sent to the nearest fog node for computation. Since tasks generated from IoT devices are heterogenous and fog node resources are limited (computation power, battery life and storage), all incoming tasks may not be served in that nearest fog node. So, the tasks are evaluated to choose the best place for execution. In the system model depicted in Fig. 2, task's deadline time is the first criterion used in scheduling. Based on the type of task and data transfer requirements, expected network delay is computed. The result is then used for driving the expected service time. By combining the two (task deadline and type), it is possible to get a better QoS and minimum overall response time. Referring to Fig. 1, tasks can be executed at one of the upper three layers.

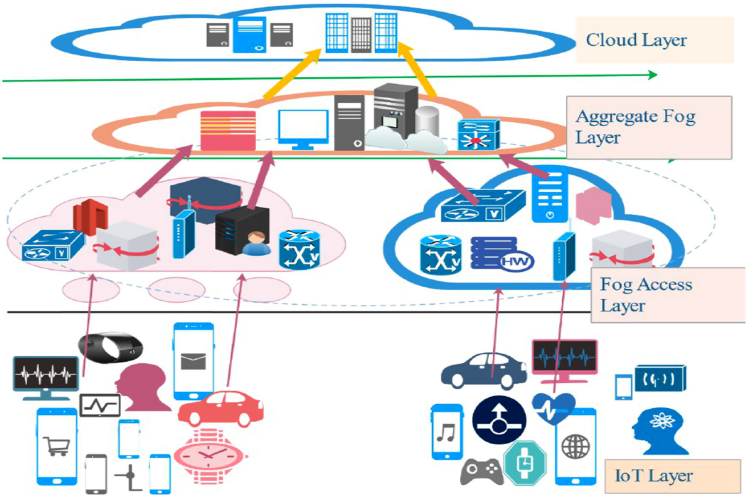

Fig. 1. System architecture

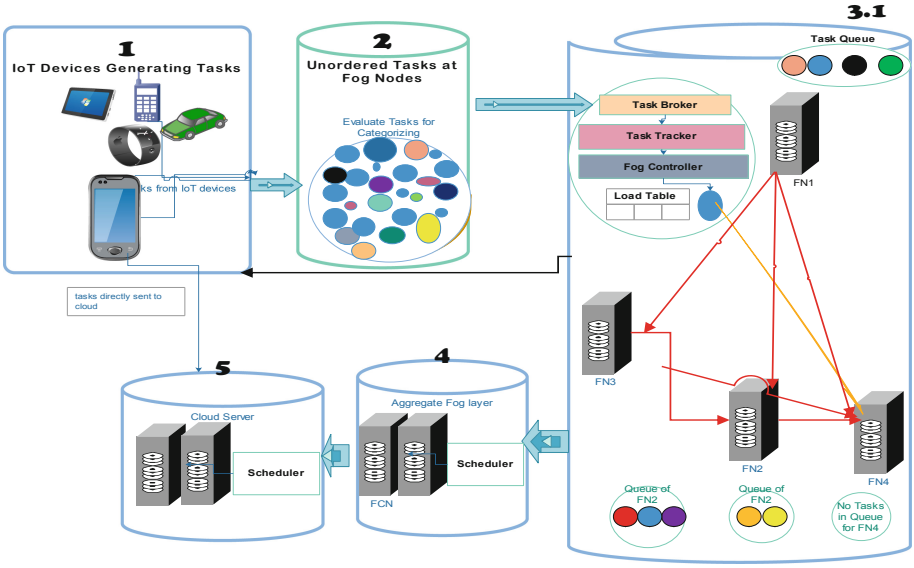

Fig. 2. Proposed system model to demonstrate the proposed algorithm

So, tasks that are not executed at the Fog Access (or lower-level fog nodes) layer can be executed at the aggregate or cloud layers. The aggregate layer stores or executes task when it gets a request from lower-level fog nodes. But, the cloud layer may get requests directly from the IoT devices or via the aggregate nodes.

4.2 Discerption of the Proposed Algorithm

Figure 3 and Fig. 4 show both the pseudocode and flowchart of the proposed task offloading algorithm. As it is depicted in the figures, tasks sent from IoT devices arrive at the nearest fog node in a Poisson rate. The tasks are classified according to the nature of the device that has generated the task. Then, the status of each fog node is checked whethe it fulfils the deadline or not. Based on which the task is offloaded to the best possible fog node in the cloud system.

5 Evaluation of the Proposed Offloading Algorithm

5.1 Simulation Parameters and Settings

For simulation purposes, the parameters of the processing capabilities of the fog nodes, aggregate fog nodes, and the cloud server are shown in Table 2 and 3. The CloudAnalyst tool is configured to generate periodic rapid simultaneous requests to create traffic based on 1) the number of online users at a given time, 2) the input data size of requests, 3) the instantaneous network characteristics, and 4) the hardware behavior of the fog nodes, aggregate fog nodes, and cloud servers. Twenty different IoT devices are used to generate different tasks. These includes, among others, smart speakers, smart watches, smartphones, TV sets, smart cane, lighting sets, refrigerators, smart clothes, and biosensors. Five fog nodes are used that have different computational power since fog nodes are heterogeneous in the real world. One aggregate fog node and one cloud server both with ten virtual machines are also considered.

IoT devices initiate requests, send it to the cloud server directly or to the nearest fog node. Based on the proposed algorithm, if it sends the request to a fog node the task would be processed at the receiver fog node or offloaded either to the best neighboring fog node (with minimal response time) or to the aggregate fog node. Since, transmission delay has a huge impact on response time; the bandwidth has been appropriately set between IoT devices and fog nodes, and among fog nodes themselves. Consequently, the link between IoT devices and fog nodes is subdivided into three categories. The bandwidth between IoT devices, that generates simple tasks (S), is assumed to be 250 Kbps. Those IoT devices that generate medium tasks (M) have 25 Mbps and those devices that generate large tasks (L) have a 54 Mbps bandwidth. The transmission rate among fog nodes is set to 100 Mbps as explained in [2, 9]. Depending on the above parameters, the proposed algorithm was compared with FCFS and Dynamic task offloading algorithms as they are put in [10] and [11], respectively.

Even though there are many recently proposed task offloading algorithms in cloud and fog computing systems the two algorithms are selected based on their popularity and regarding as the basic ones in area as discussed in [5] and [12].

First Come First Served Task Offloading Algorithm (FCFS): This algorithm works based on the arrival time of tasks. For time-sensitive applications it is less efficient. The reason behind this is the waiting time for tasks is higher. Large tasks that arrive early need much time for execution and force time-sensitive tasks to wait much longer.

Algorithm - 1: Offloading a Task

***Input**: Task, the deadline of a task, type of task and candidate nodes n, and data size*
***Output**: Target fog node, Response time*

1. *Request received from IoT device*
2. *Identify task type (S-small, M-medium, L-large)*
3. *If Fog node j is free*
4. *Assign request to the node for execution*
5. *End if*
6. *Else*
7. *If (Deadline > 100ms)*
8. *Offload task to the cloud server*
9. *End if*
10. *If (Deadline > 80 & Deadline < 100)*
11. *Offload task to the Aggregate fog node*
12. *End if*
13. *Else*
14. *Calculate execution time E_t of the task at node j*
15. *Calculate waiting time W_t of the task in the queue*
16. *Calculate transmission delay D_t for the task*
17. *Calculate Service time (Sojourn time) $S_t=W_t + E_t + D_t$*
18. *If (Deadline > S_t)*
19. *Assign task in the queue of node j*
20. *End if*
21. *Else*
22. *min = S_t*
23. *for each candidate node, k = 1 to n do*
24. *calculate transmission delay $D_t(j,k)$*
25. *calculate the execution time of task i at node k*
26. *calculate service time S_{tk} of task i at node k*
27. *if S_{tk} < min*
28. *min = S_{tk}*
29. *End if*
30. *End for*
31. *If Deadline < min*
32. *Offload task to that node k with min service delay*
33. *End if*
34. *Else*
35. *If Deadline < S_t < min*
36. *Assign the task to node j*
37. *End if*
38. *Else*
39. *Assign the task to the node k with min service time*
40. *End*

Fig. 3. Pseudocode of the proposed algorithm

Dynamic Task Offloading Algorithm: This algorithm operates based on the queue size of fog nodes. It offloads tasks to neighboring fog node if the queue size exceeds

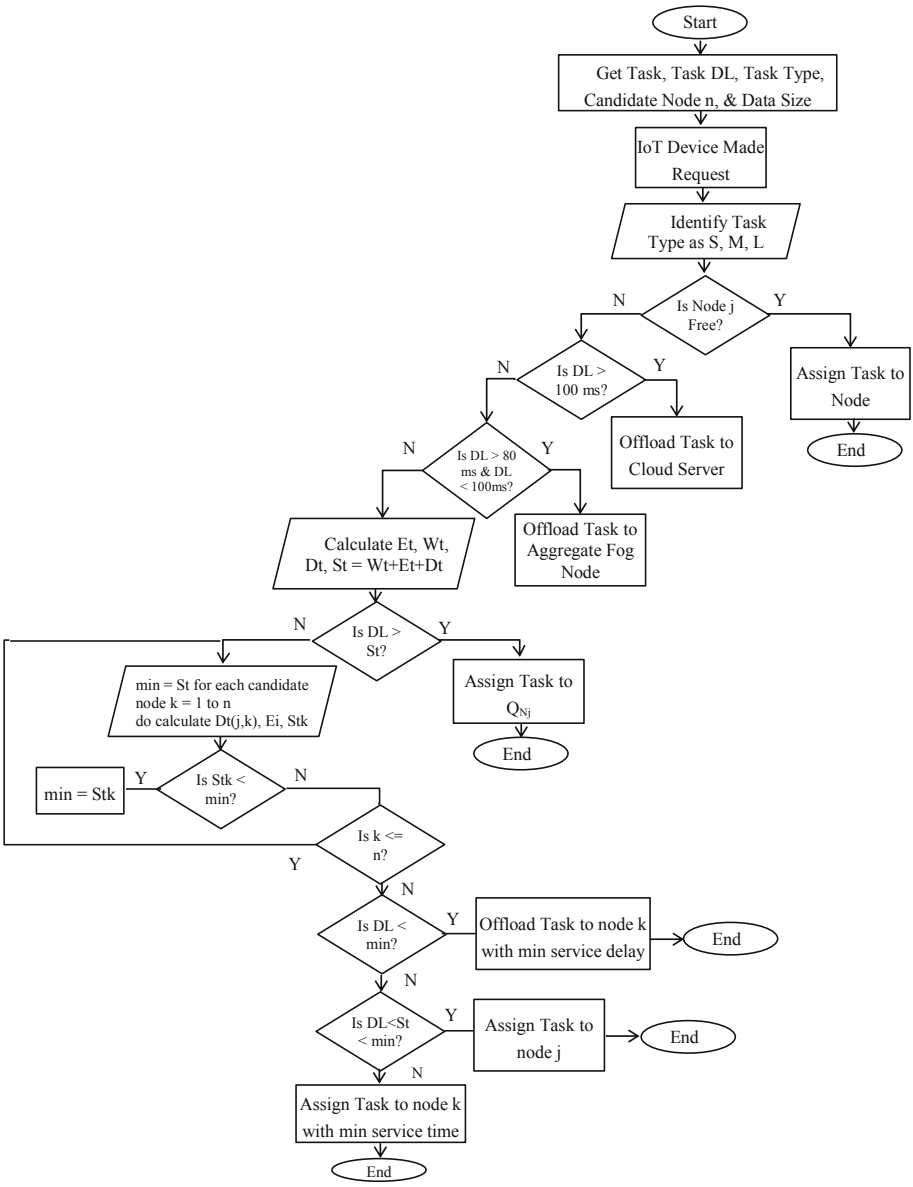

Fig. 4. Flowchart of the proposed algorithm

the threshold value of the fog node. Even though it treats all incoming tasks in the same way, it works better than the FCFS algorithm. Since it only considers the queue size of fog nodes to offload tasks it results in reduced overall response time. However, the requirements of individual tasks are not achieved, oftentimes [12].

Table 2. Processing capability of nodes

Parameter	FN1	FN2	FN3	FN4	FN5	AFN	Cloud Server
Storage (MB)	1000000	500000	500000	700000	1000000	10000000	1000000000
No. of Processors	2	2	2	2	2	4	4

Table 3. Network bandwidth parameters

Node/Node	FN1	FN2	FN3	FN4	FN5	AFN	Cloud
IoT Devices (Kbps/Mbps/Mbps)	250/25/54	250/25/54	250/25/54	250/25/54	250/25/54	-	250/25/54
FN1	100	100	100	100	100	300	-
FN2	100	100	100	100	100	300	-
FN3	100	100	100	100	100	300	-
FN4	100	100	100	100	100	300	-
FN5	100	100	100	100	100	300	-
AFN	300	300	300	300	300	-	1000
Cloud server	1000	1000	1000	1000	1000	1000	-

5.2 Scenario-Based Comparative Analysis

Experimentation has been done with two different scenarios. These scenarios are aimed to evaluate the effectiveness of the proposed algorithm with nodes processing capabilities. The first scenario considers fog nodes with identical and the second with those with varying processing capabilities.

Scenario 1: Average Delay of Tasks with Similar Processing Capability of Fog Nodes.

Table 4 and Fig. 5 presents the average delay of tasks in the three algorithms when Fog Nodes of identical processing capability is considered.

Table 4 and Fig. 5 shows the average delay of tasks executed on similar fog nodes. The result shows that the proposed algorithm has improved response times that go up to 34% and 22% when compared with FCFS and Dynamic task offloading algorithms.

Scenario 2: Response Time of Tasks with Different Processing Capability of Fog Nodes.

Table 5 and Fig. 6 shows the average delay of tasks in each algorithm by deploying fog nodes with different computational capabilities.

The result shows that the proposed algorithm has up to 37% and 25% improvements of response times when compared with FCFS and Dynamic task offloading algorithms.

The idea behind assigning different computation capability for fog nodes come from real-world scenarios. It tests the proposed algorithm to be useful for resource-constrained

Table 4. Summarized average delay of tasks with similar capability of fog nodes

Tasks	Algorithms		
	FCFS	Dynamic	Proposed
1	50.041	50.043	50.002
2	110.375	107.612	86.395
3	60.238	63.249	51.82
4	64.52	62.249	50.054
5	67.745	61.296	50.818
6	70.172	61.85	50.001
7	73.992	69.968	58.748
8	185.121	155.773	121.648
9	61.893	61.901	50.892
10	65.495	61.536	50.804
11	60.903	56.905	67.882
12	56.043	62.21	50.114
13	130.549	112.874	120.114
14	65.096	50.906	44.811
15	60.94	49.826	46.799
16	49.637	50.829	48.922
17	65.701	57.729	60.194
18	139.754	115.33	125.395
19	59.871	50.493	42.482
20	50.027	54.705	49.709

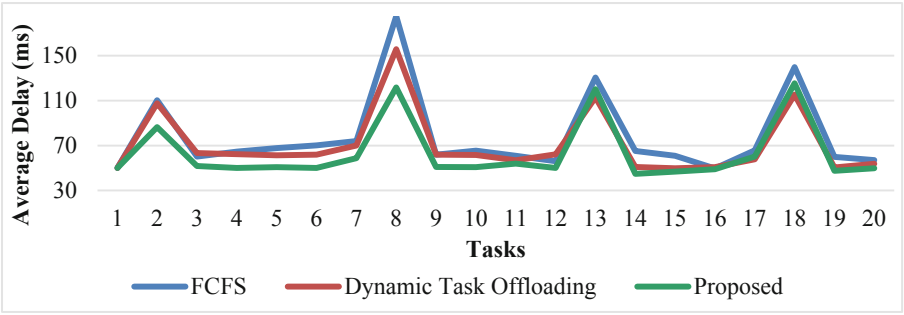

Fig. 5. Average delay of tasks with a similar processing capability of fog nodes

devices and nodes. Moreover, its ability to find and assign tasks to the most appropriate nodes during offloading is exhibited by the results obtained in Table 4 and Fig. 6.

Table 5. Summarized average delay of tasks with different capability of fog nodes

Tasks	Algorithms		
	FCFS	Dynamic	Proposed
1	50.041	50.043	50.03
2	110.375	107.612	83.68
3	60.238	63.249	51.765
4	64.52	62.249	49.935
5	67.745	61.296	50.818
6	70.172	61.85	60.703
7	73.992	69.968	57.33
8	185.121	155.773	116.213
9	61.893	61.901	50.79
10	65.495	61.536	58.712
11	60.903	56.905	46.951
12	56.043	62.21	50.791
13	130.549	112.874	125.536
14	65.096	50.906	43.837
15	60.94	53.826	47.797
16	49.637	50.829	41.861
17	65.701	61.729	54.698
18	139.754	115.33	121.523
19	59.871	50.493	41.473
20	50.027	54.705	45.906

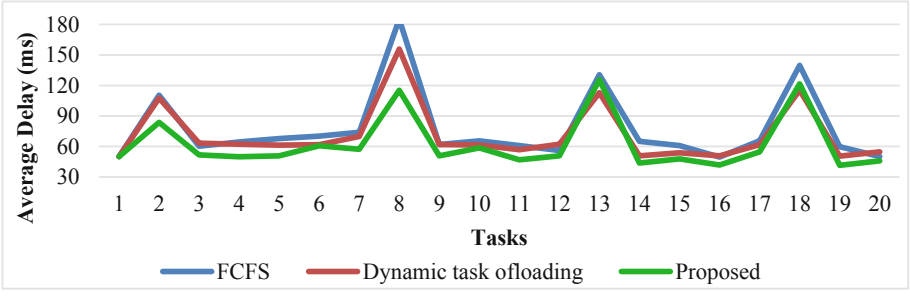

Fig. 6. Average delay of tasks with different processing capability of fog nodes

6 Conclusions

The growth of the Internet of Things (IoT) opens up new challenges and opportunities. Different task scheduling algorithms are proposed to serve those IoT applications. Existing task scheduling algorithms have gaps in treating delay-sensitive tasks arrived at fog nodes. In this paper, deadline-based offloading algorithm has been proposed that can reduce the average delay of tasks. It solves the problem in two ways, one, by prioritizing tasks based on their deadline and offload those tasks that are delay sensitive to neighboring fog nodes that can achieve the delay requirements. The proposed algorithm is evaluated with two other common task-offloading algorithms and using two different scenarios based on varying computational capacities. The results confirmed that it is possible to get up to 37% improvement of average response time of delay sensitive tasks.

Although this work improved the response time of tasks, some critical questions need to be assessed to further understand and improve the effectiveness of the proposed algorithm. The additional energy consumption of the proposed algorithm needs to be investigated to apply it on resource-constrained IoT devices. Moreover, the observed improvements in average response times are not consistent with varying number of tasks which deserve further investigations to know the underlying reasons.

References

1. Alameddine, H.A., Sharafeddine, S., Sebbah, S., Ayoubi, S., Assi, C.: Dynamic task offloading and scheduling for low-latency IoT services in multi-access edge computing. IEEE J. Sel. Areas Commun. **37**(3), 668–682 (2019)
2. Saurabh, S., Hassan, M.F., Khan, M.K., Jung, L.T., Awang, A.: An analytical model to minimize the latency in healthcare internet-of-things in fog computing environment. PLoS ONE **14**(11), e0224934 (2019)
3. Fricker, C., Guillemin, F., Robert, P., Thompson, G.: Analysis of an offloading scheme for data centers in the framework of fog computing. ACM Trans. Model. Perform. Eval. Comput. Syst. (TOMPECS). **1**(4), 1–8 (2016)
4. Ma, X., Zhao, Y., Zhang, L., Wang, H., Peng, L.: When mobile terminals meet the cloud: computation offloading as the bridge. IEEE Network **27**(5), 28–33 (2013)
5. Aazam, M., Zeadally, S., Harras, K.A.: Offloading in fog computing for IoT: review, enabling technologies, and research opportunities. Futur. Gener. Comput. Syst. **87**, 278–289 (2018)
6. La, Q.D., Ngo, M.V., Dinh, T.Q., Quek, T.Q., Shin, H.: Enabling intelligence in fog computing to achieve energy and latency reduction. Digit. Commun. Networks. **5**(1), 3–9 (2019)
7. Wang, S., et al.: Adaptive federated learning in resource constrained edge computing systems. IEEE J. Sel. Areas Commun. **37**(6), 1205–1221 (2019)
8. Elgazar, A., Harras, K., Aazam, M., Mtibaa, A.: Towards intelligent edge storage management: determining and predicting mobile file popularity. In: 2018 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering (MobileCloud), pp. 23–28. IEEE, Bamberg, Germany (2018)
9. Yousefpour, A., et al.: FogPlan: a lightweight QoS-aware dynamic fog service provisioning framework. IEEE Internet Things J. **6**(3), 5080–5096 (2019)
10. Wang, T., Wei, X., Tang, C., Fan, J.: Efficient multi-tasks scheduling algorithm in mobile cloud computing with time constraints. Peer-to-Peer Networking Appl. **11**(4), 793–807 (2017). https://doi.org/10.1007/s12083-017-0561-9

11. Jia, M., Liang, W., Xu, Z.: QoS-aware task offloading in distributed cloudlets with virtual network function services. In: Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems, pp. 109–116. Florida, USA (2017)
12. Ghobaei-Arani, M., Souri, A., Rahmanian, A.A.: Resource management approaches in fog computing: a comprehensive review. J. Grid Comput. **18**(1), 1–42 (2019). https://doi.org/10.1007/s10723-019-09491-1
13. Yousefpour, A., Ishigaki, G., Jue, J.P.: Fog computing: towards minimizing delay in the internet of things. In: 2017 IEEE International Conference on Edge Computing (EDGE), pp. 17–24). IEEE, Honolulu, USA (2017)
14. Kim, S.: New application task offloading algorithms for edge, fog, and cloud computing paradigms. Wirel. Commun. Mobile Comput. (2020)
15. Akherfi, K., Gerndt, M., Harroud, H.: Mobile cloud computing for computation offloading: issues and challenges. Appl. Comput. Inform. **14**(1), 1–6 (2018)
16. Jošilo, S., Dán, G.: Decentralized algorithm for randomized task allocation in fog computing systems. IEEE/ACM Trans. Networking **27**(1), 85–97 (2018)
17. Zhao, X., Zhao, L., Liang, K.: An energy consumption oriented offloading algorithm for fog computing. In: Lee, J.-H., Pack, S. (eds.) QShine 2016. LNICSSITE, vol. 199, pp. 293–301. Springer, Cham (2017). https://doi.org/10.1007/978-3-319-60717-7_29
18. Du, J., Zhao, L., Feng, J., Chu, X.: Computation offloading and resource allocation in mixed fog/cloud computing systems with min-max fairness guarantee. IEEE Trans. Commun. **66**(4), 1594–1608 (2018)
19. Zhu, Q., Si, B., Yang, F., Ma, Y.: Task offloading decision in fog computing system. China Commun. **14**(11), 59–68 (2017)
20. Kan, T.Y., Chiang, Y., Wei, H.Y.: QoS-aware mobile edge computing system: multi-server multi-user scenario. In: 2018 IEEE Globecom Workshops (GC Wokshops), pp. 1–6. IEEE, Abu Dhabi, UAE (2018)
21. Gao, L., Moh, M.: Joint computation offloading and prioritized scheduling in mobile edge computing. In: 2018 International Conference on High Performance Computing and Simulation (HPCS), pp. 1000–1007. IEEE, Orléans, France (2018)
22. Mukherjee, M., et al.: Task data offloading and resource allocation in fog computing with multi-task delay guarantee. IEEE Access. **7**, 152911–152918 (2019)
23. Jin, Y., Lee, H.: On-demand computation offloading architecture in fog networks. Electronics **8**(10), 1076 (2019)
24. Alghamdi, I., Anagnostopoulos, C., Pezaros, D.P.: Delay-tolerant sequential decision making for task offloading in mobile edge computing environments. Information **10**(10), 312 (2019)
25. Xu, J., Hao, Z., Sun, X.: Optimal offloading decision strategies and their influence analysis of mobile edge computing. Sensors. **19**(14), 3231 (2019)
26. Sun, H., Yu, H., Fan, G., Chen, L.: Energy and time efficient task offloading and resource allocation on the generic IoT-fog-cloud architecture. Peer-to-Peer Networking Appl. **13**(2), 548–563 (2019). https://doi.org/10.1007/s12083-019-00783-7
27. Lin, L., Liao, X., Jin, H., Li, P.: Computation offloading toward edge computing. Proc. IEEE **107**(8), 1584–1607 (2019)
28. Vu, T.T., Nguyen, D.N., Hoang, D.T., Dutkiewicz, E., Nguyen, T.V.: Optimal Energy Efficiency with Delay Constraints for Multi-layer Cooperative Fog Computing Networks. arXiv preprint arXiv:1906.03567 (2019)
29. Pan, Y., Jiang, H., Zhu, H., Wang, J.: Latency Minimization for Task Offloading in Hierarchical Fog-Computing C-RAN Networks. arXiv preprint arXiv:2003.11685 (2020)
30. Kang, J., Kim, S., Kim, J., Sung, N., Yoon, Y.: Dynamic offloading model for distributed collaboration in edge computing: a use case on forest fires management. Appl. Sci. **10**(7), 2334 (2020)

31. Hossain, M.D., Sultana, T., Nguyen, V., Nguyen, T.D., Huynh, L.N., Huh, E.N.: Fuzzy based collaborative task offloading scheme in the densely deployed small-cell networks with multi-access edge computing. Appl. Sci. **10**(9), 3115 (2020)
32. Bala, M.I., Chishti, M.A.: Optimizing the computational offloading decision in cloud-fog environment. In: 2020 International Conference on Innovative Trends in Information Technology (ICITIIT), pp. 1–5. IEEE, Kottayam, India (2020)
33. Misra, S., Saha, N.: Detour: dynamic task offloading in software-defined fog for IoT applications. IEEE J. Sel. Areas Commun. **37**(5), 1159–1166 (2019)
34. Josilo, S.: Task placement and resource allocation in edge computing systems. Ph.D. diss., KTH Royal Institute of Technology (2020)
35. Hussein, M.K., Mousa, M.H.: Efficient task offloading for IoT-based applications in fog computing using ant colony optimization. IEEE Access. **8**, 37191–37201 (2020)
36. Al-Khafajiy, M., Baker, T., Waraich, A., Al-Jumeily, D., Hussain, A.: IoT-fog optimal workload via fog offloading. In: 2018 IEEE/ACM International Conference on Utility and Cloud Computing Companion (UCC Companion), pp. 359–364. IEEE, Zurich, Switzerland (2018)
37. Shuminoski, T., Kitanov, S., Janevski, T.: Advanced QoS provisioning and mobile fog computing for 5G. Wirel. Commun. Mobile Comput. 2018 (2018)