Journal of Electrical and Computer Engineering Innovations (JECEI)
Journal homepage: http://www.jecei.sru.ac.ir

Research paper

New Platform for IoT Application Management Based on Fog Computing

S. Kalantary[1], J. Akbari Torkestani[2,*], A. Shahidinejad[1]

[1]*Department of Computer Science, Qom Branch, Islamic Azad University, Qom, Iran.*

[2]*Department of Computer Science, Arak Branch, Islamic Azad University, Arak, Iran.*

Article Info

Article History:
Received 14 December 2022
Reviewed 25 January 2023
Revised 01 March 2023
Accepted 11 March 2023

Keywords:
Internet of things
fog computing
perfect difference graph
layered architecture

*Corresponding Author's Email Address:
ja.akbari@iau.ac.ir

Abstract

Background and Objectives: With the great growth of applications sensitive to latency, and efforts to reduce latency and cost and to improve the quality of service on the Internet of Things ecosystem, cloud computing and communication between things and the cloud are costly and inefficient; Therefore, fog computing has been proposed to prevent sending large volumes of data generated by things to cloud centers and, if possible, to process some requests. Today's advances in 5G networks and the Internet of Things show the benefits of fog computing more than ever before, so that services can be delivered with very little delay as resources and features of fog nodes approach the end user.
Methods: Since the cloud-fog paradigm is a layered architecture, to reduce the overall delay, the fog layer is divided into two sub-layers in this paper, including super nodes and ordinary nodes in order to use the coverage of super peer networks to use the connections between fog nodes in addition to taking advantage of the features of that network and improving the performance of large-scale systems. It causes fog nodes to interact with each other in processing requests and fewer data will be sent to the cloud, resulting in a reduction in overall latency. To reduce the cost of bandwidth used among fog nodes, we have organized a sub-layer of super nodes in the form of a Perfect Difference Graph (PDG). The new platform proposed for aggregation of fog computing and Internet of Things (FOT) is called the P2P-based Fog supported Platform (PFP).
Results: We evaluate the utility of our proposed method by applying ifogsim simulator and the results achieved are as follows: (1) power consumption parameter in our proposed method 24% and 38% have improved compared to the structure three-layer fog computing architecture and without fog layer respectively; (2) network usage parameter in our proposed method 26% and 32% have improved compared to the structure three-layer fog computing architecture and without fog layer respectively; (3) average response time parameter in our proposed method 17% and 58% have improved compared to the structure three-layer fog computing architecture and without fog layer respectively; and (4) delay parameter in our proposed method 1% and 0.4% have improved compared to the structure three-layer fog computing architecture and without fog layer respectively.
Conclusion: Numerical results obtained from the simulation show that the delay and cost parameters are significantly improved compared to the structure without fog layer and three-layer fog computing architecture. Also, the results show that increasing number of things has the same effect in all cases.

Introduction

Constrained systems are not able to interact with the vast amount of data on the Internet of Things (IoT); Therefore, cloud computing has been widely used and is an integral part of IoT. IoT creates unimaginable amounts of different types of data. This data is distributed throughout the environment, and is called big data. Some of the most important challenges to be considered for big data include processing, preparation and storage [1], [2], [3] and [6].

Cloud computing is a computing framework which is a good solution in this regard. Data is sent to cloud data centers for processing and storage and will be available after analysis and processing [7]. Using cloud computing in IoT applications has the following advantages: cost saving, reliability, manageability. The strength of IoT is that it permeates people's daily lives in terms of personal and home issues such as smart city, smart home, smart health, life assistance and work issues such as industry and factory automation and smart transportation. However, delays due to the cloud being away from end-users challenge the usefulness of IoT systems in many applications. Despite this, fog computing has been recommended to deal with many cloud processing problems such as unreliable latency, lack of proper mobility support and lack of location-awareness support, and reduced processing speed due to increased data transfer size and consequent reduced bandwidth. Fog processing is used to prevent the transmission of this large amount of data to cloud data centers and also to perform a series of necessary pre-processing on them. Fog computing is a distributed computing paradigm that acts as an intermediate layer between cloud data centers and IoT devices [4]. This concept was first defined by Cisco as the development of cloud computing, from the core to the edge of the network. In fact, fog calculations were introduced as mini clouds [5]. The fog computing environment consists of traditional network components such as routers, switches, proxy servers, base stations, and so on. These components enable fog computing to geographically distribute cloud-based services at the edge of the network. Therefore, fog computing can support data location, scalability, interoperability, and mobility. The architecture of cloud-fog-thing layer paradigm is shown in Fig. 1.

In this research, a new platform is proposed for fog of things (FOT), which defines a fog computing structure for IoT. The P2P-based Fog-supported Platform (PFP) utilizes the features of super peer networks to use communications and interactions among fog nodes in request processing as well as improving performance and quality of services (QoS); therefore, the nodes in the fog layer are organized into two sub-layers of super and ordinary nodes.

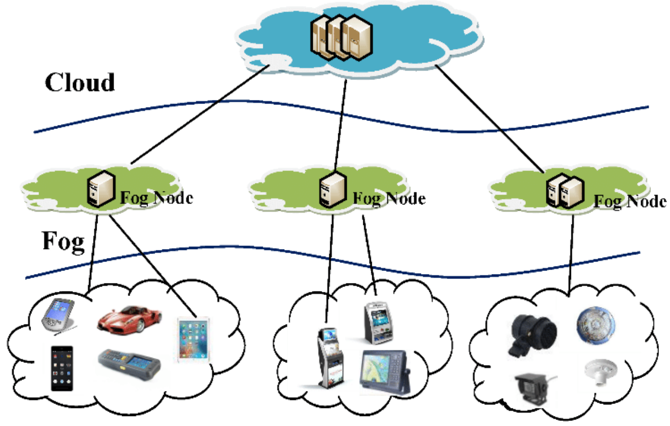

Fig. 1: Architecture of cloud-fog-thing layer.

Higher level nodes are called super fog nodes that are connected to each other in the form of a Perfect Difference Graph (PDG). Since the diameter parameter in these graphs is equal to 2, we expect a reduction in the communication delay and the volume of messages sent and consequently reduced available bandwidth. This paradigm is simulated by iFogSim simulator to manage IoT applications like smart city [29], [30]. Despite the numerous benefits of fog computing, research in this field is still emerging and immature, and many researchers are still studying it.

The rest of this article is organized as follows:

Section A provides an overview of research and studies on the smart city and the platforms provided by fog computing; Section B presents the proposed method; In the evaluation section, the results obtained from the simulation of the proposed method are stated along with its comparison with some of the performed projects, and finally the conclusion and suggestions for further research in this field are mentioned in the last section.

Technical Work Preparation

A. Related Work

Padova Smart City project, discussing the urban features of the IoT system, such as the services and items required to implement the Smart City [8]. In [9] the authors stated that coordinated distributed plans are required to create IoT applications in the smart city; Although, more attention has been paid to the integration of cloud computing and IoT in smart city applications. In [33] a cloud-based framework for creating a smart city through IoT capabilities is expressed. A framework called FOCAN (A Fog-supported Smart City Network Architecture for Management of Applications on the Internet of Everything Environments) in which components and services of the smart city communicate with each other and with fog computing [10]. The FOCAN architecture consists of two levels: IOE and fog nodes, which manage IoT applications. FOCAN is an efficient computing and communication structure which minimizes average energy consumption. A framework introduces in [11] for IoT in which data collected from medical sensors are stored on fog servers, and uses a centralized platform for end-to-end communication between end users and medical sensors. This structure has been able to support mobility well. In [12] the design of an open stack platform is proposed that considers scalability in smart city applications using fog computing. The authors in [13] have proposed a platform that uses fog computing to improve the performance of traffic and driving issues in VANET (Vehicular ad hoc network) networks to satisfy the need for location-awareness. In [14] the authors have designed a three-layer architecture for smart buildings based on fog computing, the results of which show that the fog layer is very effective in using network resources and reducing bottlenecks in cloud computing. In [15], a three-layer architecture is proposed for big data analysis in smart cities, including intermediate computational fog nodes, edge computing nodes, and fog nodes specifically with sensing power. Authors in [16] proposed a three-layer architecture is proposed for IoT applications called cloud, fog, and dew, in which the dew layer refers to edge devices such as sensors and video cameras. In [17], the authors have proposed the soft-IoT paradigm, in which they provide protocols using fog computing to facilitate the processing of local data and service delivery on small servers by virtual entities. In [18] a new architecture has been collected and updated, and has presented a process of real-time and heterogeneous information from different sources, and then it has been tested by providing a smart parking service in a smart city. In [19], a platform is provided based on fog data that uses fog computing to reduce cloud storage and transmission delays in smart health applications. The addressed issues in [20], include theoretical modeling of fog computing architecture, in particular, service delays, power consumption, and cost; But there is no specific policy to reduce service delays. In [21], the authors have suggested “task distribution” to minimize overall cost and service quality requirements in fog computing-based medical Cyber-Physical System (CPS). In [22] a service distribution strategy in the cloud-fog scenario is proposed so that the services are subdivided into sub-services and their parallel experiments are run on edge devices to minimize service delays. The optimization of data transfer from IoT sensors to the cloud by the enhanced learning technique to predict the data which will be transferred from the sensors in the future, then the amount of data transfer is reduced by determining the data which are not to be transferred, and the service quality is increased subsequently [23]. The distributed cloud storage replication method and the greedy exploratory method are proposed to minimize latency in order to counteract the amount of data transmitted between the sensor and the cloud [24]. However, it should be noted that fog computing has not been used in the last two cases. In [36] authors present a new platform integrating big data streaming processing with machine learning (ML)-based applications. And they provide a comprehensive IoT data processing workflow, including data access and transfer, big data processing, online ML, long-term storage, and monitoring. In [37] considers possible fog computing applications and potential enabling technologies towards sustainable smart cities in the IoT environments. In addition, different caching techniques and the use of Unmanned Aerial Vehicles (UAVs), and various Artificial Intelligence (AI) and Machine Learning (ML) techniques in caching data for fog-based IoT systems are comprehensively discussed. Finally, the potential and challenges of such systems are also highlighted. FogFrame is a framework for IoT application that use multi-tier fog computing and create fog colony [38]. A new delay tolerant network for IoT data processing introduces in [39] that uses multi-layer fog servers.

It is worth mentioning that in all the works reviewed in this section, the fog layer isn’t divided into two different layers of fog nodes, and they have a similar structure and the same capabilities and capacities. And a fog layer has been used and investigated in research.

B. Proposed Method

Since PDG is used in the proposed platform, before starting the details of the proposed method, we will briefly introduce and describe its features.

Definition 1: A Perfect Difference Set (PDS) is a set of residues $\{S0, S1, \dots., S\delta+1\}\ mod\ n$, so that any non-zero residue can be uniquely expressed in $\{Si - Sj\}$ format.

Definition 2: A PDG is a graph with n vertices where $n = \delta 2 + \delta + 1$, δ is the power of graph and at least equals 2. In PDG, node i is connected to $1 \le j \le \delta$, $(i \pm Sj)(mod\ n)$, and Sj is an element of PDS (Perfect Difference set) from δ order.

According to [25] and [26], 4 edges can be defined in each PDG as follows;

Ring edge: The edge connecting consecutive nodes i and $i + S1\ (mod\ n)$.

Chord edge: The edge connecting the non-consecutive nodes i and $i + Sj(mod\ n)$ where $2 \le j \le \delta$,

Forward edge: For node, including chord edges connecting nodes i and $i + Sj\ (mod\ n)$, and ring edge connecting nodes i and $i + S1\ (mod\ n)$.

Backward edge: For node i, including the chord edges connecting nodes i and $i - Sj\ (mod\ n)$, and the ring edge connecting nodes i and $i - S1\ (mod\ n)$.

Table 1 shows 10 initial values of PDS.

Table 1: Ten initial value of PDS

n	δ	Perfect difference sets
7	2	0, 1, 3
13	3	0, 1, 3, 9
21	4	0, 1, 4, 14, 16
31	5	0, 1, 3, 8, 12, 18
57	7	0, 1, 3, 13, 32, 36, 43, 52
73	8	0, 1, 3, 7, 15, 31, 36, 54, 63
91	9	0, 1, 3, 9, 27, 49, 56, 61, 77, 81
133	11	0, 1, 3, 12, 20, 34, 38, 81, 88, 94, 104, 109
183	13	0, 1, 3, 16, 23, 28, 42, 76, 82, 86, 119, 137, 154, 175
273	16	0, 1, 3, 7, 15, 63, 90, 116, 127, 136, 181, 194, 204, 233, 238, 255

a. Describing the Proposed Method

The fog node is the key component of fog computing used in this research, which provides the resources and activities requested by services and has the computing, storage, and networking ability necessary to run IoT applications [27]. On the other hand, each node must be able to communicate with the other fog node, cloud data centers, and things; because fog nodes are the intermediate layer between clouds and things [28]. Therefore, it must also have mechanisms to communicate with heterogeneous components, data collection, control, and analysis.

In this research, fog nodes are divided into two categories of super fog nodes and ordinary fog nodes based on performance, storage and computational capabilities, and location in the proposed PFP structure. Therefore, the proposed PFP architecture consists of three levels: the lowest layer includes the things in the smart city, which are connected to the Internet using various communication technologies such as 4.5G, Wi-Fi, or ZigBee, and need to be clustered according to their location; Fog nodes are in the middle level, and cloud data centers are located at the highest level. Fig. 2 shows a general framework of the proposed architecture. There are three layers in this architecture including thing, fog, and cloud. Cloud servers are located at the cloud layer and consist of several processing and storage units. The strength of this architecture goes back to the second layer, the fog layer. Fog nodes are divided into two categories based on processing, storage, and networking capabilities; therefore, the fog layer consists of two sub-layers: the upper layer is called Super Fog layer (SFL) and the bottom layer is called Ordinary Fog Layer (OLF).

In PFP, things in the things layer must be clustered; it is performed based on the location of things. In addition to distance, speed and direction of movement are also used as criteria in clustering, as some things have a very high mobility. Each thing must be associated with an ordinary fog node. Things clustered in a group will be associated with the nearest ordinary fog node. In this study, it is assumed that each active thing in IoT is associated with only one ordinary node and registers itself to only one ordinary fog.

Based on [32], things in the IoT can be classified into 3 categories. The first category will be things whose destruction causes severe irreparable physical, economic or social damage, such as a wireless pacemaker or car brake system controller. The second category includes things whose absence or breakdown has severe physical or economic effects, such as the misusing the air conditioning. Finally, the third category of things consist of those whose absence, deterioration or withdrawal from the system is not a serious threat to living beings and also economic or social conditions. Therefore, the type of each thing is determined based on this classification in the smart city. Because each fog node contains a local database, the information needed for the applications and the data in this classification can be stored in the fog nodes to store things information and the data they generate; storage and using different recovery policies to access the data also creates a prioritization to respond to requests. Storage can be used in analyzing and storage of big data.

In the past IoT platforms, all data obtained from different sources such as sensor devices, IoT devices, and websites were sent to the cloud, leading to reduced processing speed and using large quantity of bandwidth due to the high size of data transfer. Now, ordinary fog nodes perform an initial analysis of the received data, and an index of analytical data in ordinary fog nodes is stored in the super fog node instead of sending data to the cloud. Due to the mentioned storage and using the resources available in fog nodes, the need for communication with cloud data centers is minimized and there is less delay.

The upper sub layer, SFL, contains super fog nodes that are more powerful in terms of processing, storage, and networking capabilities than the nodes in the bottom layer, OFL; although they are still a long way from the capabilities of cloud servers, they can be considered mini-clouds. The nodes in this layer are connected with each other according to PDG, order δ. Each SFN is associated with several OFNs. For example, all OFNs defined in a settlement are associated with an SFN. In this structure, it must be noted that each thing is associated with only one ordinary fog node (of course, to maintain the information network, an alternative ordinary fog node is available is always available) and each OFN is connected to only one SFN and they are not in direct contact with each other.

As mentioned before, fog nodes, also have networking facilities and equipment in addition to storage. Therefore, part of the activities of ordinary fog nodes is intended to meet the immediate needs based on the demand for things and the provision of resources. Based on this, and considering the types of connections and how to send queries in PDG-based super peer-to-peer networks, 5 types of connections can be defined in this structure. Thing-to-thing relationship (t2t), thing-to-ordinary fog node relationship (t2o), super node connection to ordinary node (s2o), super node connection with another super node fog (s2s) and super node fog connection to cloud data centers (s2c) These connections are shown separately in Fig. 2, except for the t2t connection.

Requests that cannot be processed by things can be divided into 3 categories based on the amount of resources required and the estimated processing time, and like the traffic class field in the IPv6 header, the type of request can be specified, so that there is less delay in sending requests. ; Thus, the first category called low-res requests (which require low resource) that can be processed in ordinary fog nodes and are compatible with the resources and capabilities of ordinary fog nodes. The second categories (which require middle resource) includes requests that must be processed by super fog nodes, so they can be sent to super fog nodes immediately after reception to be processed by one of the super fog nodes. They can be introduced as semi-heavy processing; finally, the last category is called high-res requests (which require high resource) that must be processed by the cloud, so they are sent to the cloud immediately after being received by the super fog node, and are referred to as heavy-processing requests. Although, it should be noted that in situations such as non-acceptance in the queue, each request can be processed by another processing node, for example, low-res requests may also be processed in super fog nodes or clouds.

The strength of this method is in sending messages in the super layer of fog. Anyway, the message reaches this layer, the transmission of the message between the nodes follows the PDG algorithm. If the request is of mid-res type, it can be accepted and processed in super fog nodes; therefore, according to the PDG-based communications in SFL, requests are sent to other super fog nodes to find a suitable fog node to run. If no super fog node accepts the request at this step, it must be sent to the cloud. The following describes how PDG-based super fog nodes are connected [25] and [26]. According to [34] sending requests from fog nodes to other fog nodes (either ordinary fog nodes or super fog nodes) or sending them to the cloud is called request offloading or load sharing. In our proposed structure, deciding to offload a request to other fog nodes depends on the type of request, the fog node response time and, the conditions of the fog node in terms of available space. This will happen if the response time of the fog is longer than the maximum allowed delay of the request and the type of request also allows offload.

In this paper, our purpose is to examine whether the proposed structure is less delayed in responding to requests; Note that according to [35], we define a delay the time required to service a request sent from a thing, that is, the time interval between the moment a request is sent by an thing until a response is received.

b. Communicating and Sending Messages in PDG-Based SFL Layer

The system proposed in this paper is based on graph is called G, $G =< V, E >$where V is the total set of fog nodes including ordinary and super fog nodes. G ' is a subgraph from G, $G' =< V', E' >$can be considered as a directionless graph where V'⊂ V is the super fog node and E'⊂ E is the connections among the super fog node, called interconnection; It should be noted that the connections between ordinary fog nodes and super fog nodes are called intraconnection. Graph G' is a PDG of order δ and logical topology of super fog node is PDG. If the i∈ V' node wants to send a message to the system in order to find the appropriate node for accepting and then processing the request, a two-step process occurs (PDG-Algorithm):

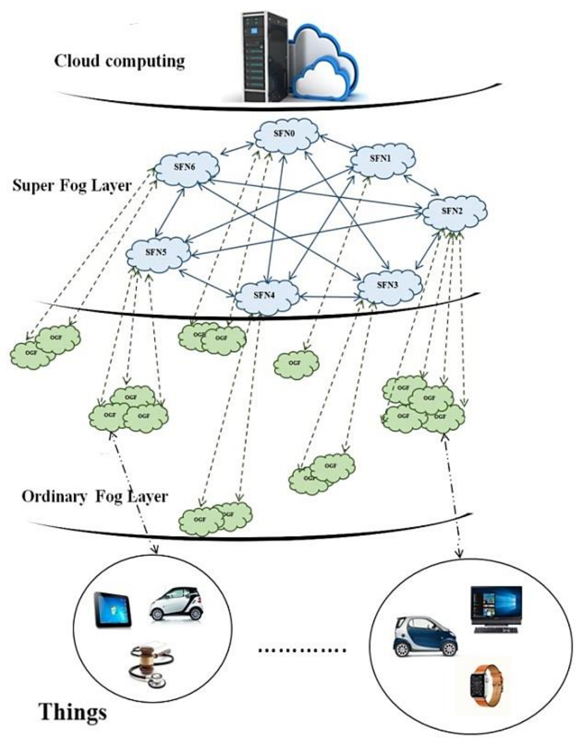

Fig. 2: Architecture of the proposed structure (PFP)
S2S connection ⟷, O2S connection ⇠----⇢
and T2O connection ⇠-·-·-⇢

Step 1: The message with the possibility of moving 2 steps (TTL = 2) is sent to the neighbors with whom it is connected according to the forward edge, and also the same message is sent to the neighbors with whom it is connected according to the backward edge (TTL = 1), and the receiving intermediate nodes reduce the TTL by one unit as soon as the message is received.

Step 2: If the intermediate node receives a message, it sends it to all nodes associated with the backward edge except the node from which it received the main message.

Since the diameter of the graph in this structure is 2, the communications are established with very little delay. Also, the number of messages moving and imposed on the super fog nodes equals $\delta 2 + \delta$, which is used in calculating the consumed bandwidth and makes the system cost less.

The procedure of the proposed method can be presented in Algorithm 1.

Algorithm 1: Pseudo code for proposed method

Input: rth request for uth user(Req{u,r})
Output: Assigned node for request processing
Begin
1. for all IoT users
2. for all IoT requests
3. Add Req{u,r} to fog-queue
4. end for
5. end for
6. while (Req{u,r} is in the fog-queue)
7. for all Req{u,r} in associated fog-queue according to Geographical clustering
8. if (possible assignment OFN) return OFN id
9. else
10. call(PDG-based SFL layer algorithm)
11.end of while
End

Fig. 3 presents proposed method in flowchart format.

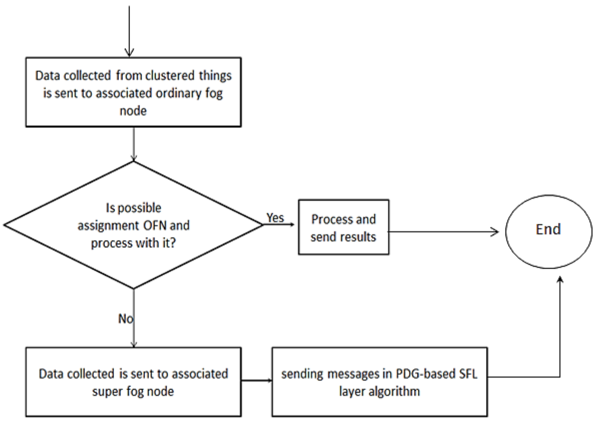

Fig. 3: Flowchart of proposed method.

Results and Discussion

In this section, we examine the proposed approach through simulation. In the following, we will explain how to set the simulator and performance criteria. Then, the simulation results will be discussed.

A. Simulation Settings

The simulations presented in this section are performed using the iFogSim library [29] and [30], which is an extension of CloudSim [31]. This simulator is used for modeling the cloud computing infrastructure and IoT services. The iFogSim toolkit allows the user to describe fog nodes and provide resource management mechanisms for performing IoT services, as well as the posibility to evaluate performance metrics related to fog environments. In the present study, the simulation was performed using the iFogSim library on a computer with an Intel Corei5 CPU, a 250 GB disk, a 4 GB RAM and Windows 10. The iFogSim emulator consists of a set of classes, including the FogDevice, Sensor, and Actuator classes for fog modeling, and a set of classes, including AppModule, AppEdge, and Tuple for modeling IoT services. Note that the FogDevice class is one of the most basic iFogSim simulator classes used to simulate fog nodes, which has memory, network, and computing resource features. This class specifies the hardware specifications of the fog nodes as well as their connections. In the proposed approach, FogDevices are designed on several levels. At the lowest level are things and IoT devices, and at the highest level are VMs, which connect to gateways using links. To configure cloud layer infrastructure and fog layers, fog devices are assumed with the characteristics listed in Table 2. The Sensor class is used to simulate IoT sensors and can be used to generate tuples that are equivalent to tasks in a cloud computing environment. The Actuator class is used to implement the output operation.

Table 2: cloud layer and fog layer specification

		MIPS	RAM	storage	Down-BW	Up-BW
Cloud Layer	Host	48800	60000	1000000	100	10000
	VM	3800	8000	100000	1000	10000
Fog Layer	SFN	200	4000	30000	1000	10000
	OFN	500	1000	10000	50	5000

B. Performance Criteria and Simulation Results

In this section, the results of the simulated scenario discussed above are presented and the results obtained are compared with the other two modes. In one mode, there is only one layer of fog in the system and all fog nodes have the same structure and are in the same state in response to requests.

In the other mode, the considered system employs a cloud layer but no fog layer. The efficiency parameters studied in the research include energy consumption, average response time and the amount of network consumption. The computing power of fog devices is given based on millions of instructions per second (MIPS). RAM and storage are specified in MB and bandwidth is measured in Mbps. In this section, we use notations that describe in Table 3.

In the present study, the experiments were repeated 4 times; each time we changed the number of things in the system; the number of sensors was considered 10, 20, 50 and 100. The number of ordinary and super fog nodes has not changed. Fig. 3 shows the amount of energy consumption that we compute it based on (1) for proposed algorithm. According to the obtained results and as we expected, the amount of energy consumption increased following an increase in the number of things. Since we used PDG network and two layers of fog with different characteristics in the proposed approach, the least amount of energy was consumed in all cases compared to the other two modes (the system with one layer of fog, and the system without fog layer). The information of fog devices is received more accurately and quickly due to the presence of PDG network; therefore, we face a reduction in energy consumption.

$$Energy\ consumption = CEC + (CT - UT) * LHP \quad (1)$$

Table 3: Notations and definitions

Notation	Definition
CEC	Current Energy Consumption
CT	Current Time
UT	Update Time
LHP	Last Host Power
TL	Total latency
TS	Total size of tuple
MST	Maximum simulation Time
EST	Estimated Service Time
EET	Ending Execution Time
N	Total number of executed tuple
ST	Service Time

In Fig. 4, network usage is investigated and compared in 3 modes: the system with only cloud layer and without the fog layer, the system with a fog layer with similar fog nodes, and the proposed model. As expected, the network consumption increases sharply as the number of things in the system increases in the absence of the fog layer. In the other two modes, we observe less network consumption than in the first mode. Comparing the conditions in the fog layer system, it can be stated that the network consumption in the proposed algorithm is of the lowest value in all conditions. This parameter for proposed method computes based on (2).

$$Network\ usage = \frac{TL*TS}{MST} \quad (2)$$

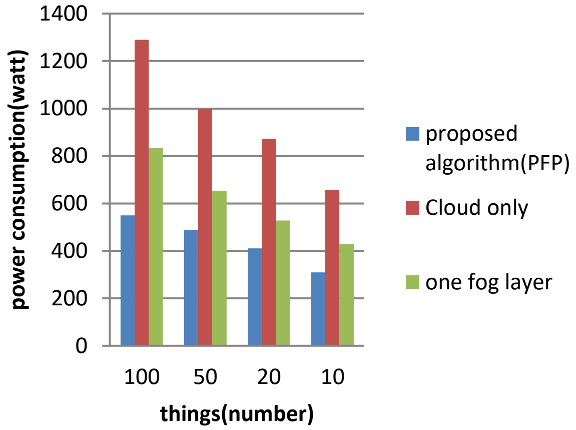

Fig. 3: Power consumption.

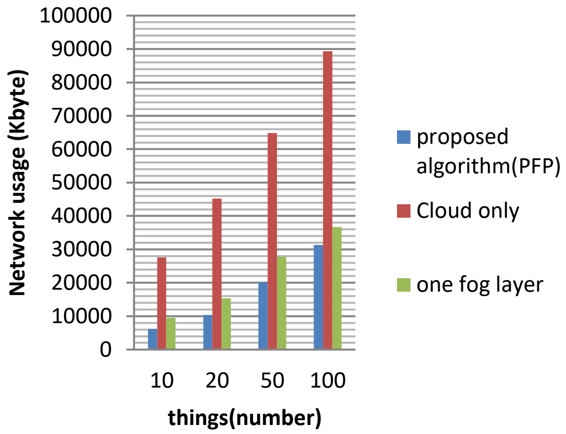

Fig. 4: Network usage.

Fig. 5 shows the average response time in the system. As shown in the figure, the average response time increases with increasing the number of sensors, which is predictable; because the number of requests in the system increases following the increase in the number of sensors and consequently there will be an increase in the average response time of the system. But what is noteworthy is that the average response time in the proposed algorithm has improved compared to the other two modes. Because request response management is more appropriate in the proposed algorithm.

Fig. 6 shows the delay in different time intervals. The proposed method has the least delay compared to other methods. The reason for that is the use of pdg communication, which leads to the selection of fog node with higher accuracy, and as a result, the response time will be reduced and the amount of delay will be less. We considered the time intervals as 10 second intervals and repeated the work up to 100 seconds. In the situation where the fog layer is a single layer, because the load distribution happens with a longer delay, the delay should be increased compared to the proposed method. This parameter is computed based on (3).

$$Delay = \frac{EST*N+(EET-ST)}{N} \quad (3)$$

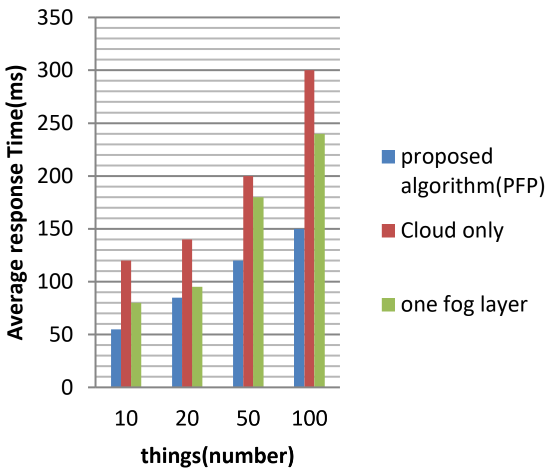

Fig. 5: Average response time.

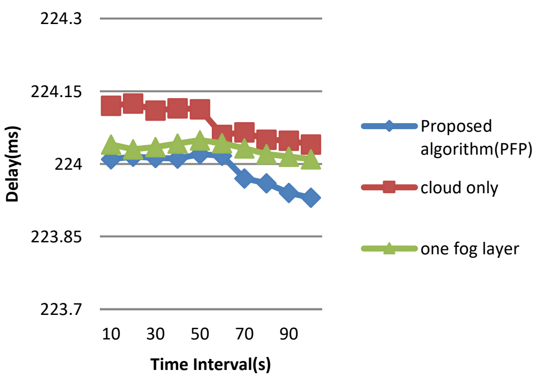

Fig. 6: Delay.

Conclusion

In this paper, a new platform is proposed for fog of things (FoT) that defines a fog computing structure for IoT. The P2P-based fog supported platform (PFP) utilizes the features of super peer networks to use communications and interactions between fog nodes in request processing as well as improving performance and QoS; therefore, it organizes the nodes in the fog layer into two sub-layers of super and ordinary nodes. Higher level nodes are super fog nodes that are connected to each other in the form of a PDG (Perfect Difference Graph). Since the diameter parameter in these graphs is equal to 2, examining the results, it was observed that the performance evaluation parameters including energy consumption, average response time, network consumption and delay have been significantly improved. This paradigm is simulated by ifogsim simulator for managing IoT applications like smart city. The numbers of super and ordinary fog nodes are constant and have not changed in this research. It is suggested for the future researches to investigate the effect of it. The structure of these nodes also follows the PDG; it is suggested to consider and compare other structures in the future research.

Author Contributions

S.Kalantary designed the experiments and collected the data. All authors carried out the data analysis. J. Akbari and A.Shahidinejad were supervisor. All authors interpreted the results and wrote the manuscript.

Acknowledgment

The authors would like to thank both the Editor and anonymous reviewers of JECEI for their valuable and constructive Comments and feedbacks.

Conflict of Interest

The authors declare no potential conflict of interest regarding the publication of this work. In addition, the ethical issues including plagiarism, informed consent, misconduct, data fabrication and, or falsification, double publication and, or submission, and redundancy have been completely witnessed by the authors.

Abbreviations

IoT	Internet of Things
PDG	Perfect Difference Graph
FoT	Fog of Things
QoS	Quality of Service
P2P	Peer to Peer
CPS	Cyber-Physical System
FoCAN	Fog-supported Smart City Network Architecture
UAV	Unmanned Aerial Vehicles
ML	Machine Learning
AI	Artificial Intelligence
SFL	Super Fog layer
OFL	Ordinary Fog layer
TTL	Time to Live

References

[1] Y. Hajjaji, W. Boulila, I. R. Farah, I. Romdhani, A Hussain, “Big data and IoT-based applications in smart environments: A systematic review,” Compu. Sci. Rev., 39: 100318, 2021.

[2] F. E. F. Samann. S. R. M. Zeebaree, S. Askar, “IoT provisioning QoS based on cloud and fog computing,” J. App. Sci. Tech. Trends, 2(01): 29-40, 2021.

[3] I. Lee, K. Lee, "The Internet of Things (IoT): Applications, investments, and challenges for enterprises," Bus. Horiz., 58(4): 431-440, 2015.

[4] T. Wang, Y. Liang, W. Jia, M. Arif, A. Liu, M. Xie, "Coupling resource management based on fog computing in smart city systems," J. Network Compu. Appl., 135: 11-19, 2019.

[5] M. Etemadi, M. Ghobaei-Arani, A. Shahidinejad, "Resource provisioning for IoT services in the fog computing environment: An autonomic approach," Comput. Commun., 161: 109-131, 2020.

[6] S. Kalantary, J. Akbari Torkestani, A. Shahidinejad, "Resource discovery in the Internet of Things integrated with fog computing using Markov learning model," J. Supercompu., 77: 13806-13827, 2021.

[7] M. Ghobaei, A. Shahidinejad, M. Torabi, "Resource elasticity management using fuzzy controller based on threshold changes in the cloud computing environment," Electron. Cyber Def., 8(3): 63-81, 2020.

[8] A. Cenedese, A. Zanella, L. Vangelista, M. Zorzi "Padova smart city: An urban internet of things experimentation," in Proc. IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks, 2014.

[9] H. A. Khattak, H. Farman, B. Jan, I. Ud Din, "Toward integrating vehicular clouds with IoT for smart city services," IEEE Network, 33(2): 65-71, 2019.

[10] P. G. V. Naranjo, Z. Pooranian, M. Shojafar, M. Conti, R. Buyya, "FOCAN: A Fog-supported smart city network architecture for management of applications in the Internet of Everything environments," J. Parallel Distrib. Comput., 132: 274-283, 2019.

[11] C. Thota, R. Sundarasekar, G. Manogaran, R. Varatharajan, M. K. Priyan, "Centralized fog computing security platform for IoT and cloud in healthcare system, in Fog computing: Breakthroughs in research and practice," IGI global: 365-378, 2018.

[12] D. Bruneo, S. Distefano, F. Longo, G. Merlino, A. Puliafito, V. D'Amico, M. Sapienza, G. Torrisi, "Stack4Things as a fog computing platform for Smart City applications," in 2016 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS), 2016.

[13] J. Grover, A. Jain, S. Singhal, A. Yadav, "Real-time vanet applications using fog computing," in Proc. First International Conference on Smart System, Innovations and Computing, 2018.

[14] F. J. Ferrández-Pastor, H. Mora, A. Jimeno-Morenilla, B. Volckaert, "Deployment of IoT edge and fog computing technologies to develop smart building services," Sustainability, 10(11): 3832, 2018.

[15] B. Tang, Z. Chen, G. Hefferman, T. Wei, H. He, Q. Yang, "A hierarchical distributed fog computing architecture for big data analysis in smart cities," in Proc. the ASE BigData & SocialInformatics: 1-6, 2015.

[16] K. Skala, D. Davidovic, E. Afghan, I. Sovic, Z. Šojat, "Scalable distributed computing hierarchy: Cloud, fog and dew computing," Open J. Cloud Compút. (OJCC), 2(1): 16-24, 2015.

[17] C. Prazeres, M. Serrano, "Soft-iot: Self-organizing fog of things," in Proc. 30th International Conference on Advanced Information Networking and Applications Workshops (WAINA), 2016.

[18] R. Tandon, P. Gupta, "Optimizing smart parking system by using fog computing," in Proc. International Conference on Advances in Computing and Data Sciences, 2019.

[19] A. M. Rahmani, T. Nguyen Gia, B. Negash, A. Anzanpour, I. Azimi, M. Jiang, P. Liljeberg, "Exploiting smart e-Health gateways at the edge of healthcare Internet-of-Things: A fog computing approach," Future Gener. Comput. Syst., 78: 641-658, 2018.

[20] X. Masip-Bruin, X. Masip-Bruin, E. Marín-Tordera; G. Tashakor, A. Jukan, G. J. Ren, "Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems," IEEE Wireless Commun., 23(5): 120-128, 2016.

[21] G. M. Dias, Prediction-based strategies for reducing data transmissions in the IoT, Universitat Pompeu Fabra, 2016.

[22] V. B. Souza Xavi Masip-Bruin, E. Marin-Tordera, W. Ramirez, S. Sanchez, "Towards distributed service allocation in fog-to-cloud (f2c) scenarios," in Proc. 2016 IEEE global communications conference (GLOBECOM), 2016.

[23] M. Ramachandra, "Optimization of the data transactions and computations in IoT sensors," in Proc. 2016 International Conference on Internet of Things and Applications (IOTA), 2016.

[24] A. Kumar, N. C. Narendra, U. Bellur, "Uploading and replicating internet of things (IoT) data on distributed cloud storage," in Proc. 2016 IEEE 9th International Conference on Cloud Computing (CLOUD), 2016.

[25] J. S. Li, C. H. Chao, "An efficient superpeer overlay construction and broadcasting scheme based on perfect difference graph," IEEE Trans. Parallel Distrib. Syst., 21(5): 594-606, 2009.

[26] N. Singh et al., Study of Topological Analogies of Perfect difference Network and Complete Graph, 2019.

[27] M. Ghobaei-Arani, A. Souri, A. A. Rahmanian, "Resource management approaches in fog computing: a comprehensive review," J. Grid Comput., 18: 1-42, 2019.

[28] E. M. Tordera, E. Marin Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G. Jie Ren, J. Zhu, J. Farre, "What is a fog node a tutorial on current concepts towards a common definition," arXiv preprint arXiv:1611.09193, 2016.

[29] H. Gupta, A. Vahid Dastjerdi, S. K. Ghosh, R. Buyya, "iFogSim: A toolkit for modeling and simulation of resource management techniques in the Internet of Things, Edge and Fog computing environments," Softw. Pract. Exper., 2017. 47(9): 1275-1296, 2017.

[30] R. Mahmud, R. Buyya, "Modelling and simulation of fog and edge computing environments using iFogSim toolkit," Fog and edge computing: Principles and paradigms: 433-465, 2019.

[31] R. N. Calheiros, R. N. Calheiros, R. Ranjan, A. Beloglazov, C. A. F. De Rose, R. Buyya, "CloudSim: a toolkit for modeling and simulation of cloud computing environments and evaluation of resource provisioning algorithms," Softw. Pract. Exper., 41(1): 23-50, 2011.

[32] P. Maiti, J. Shukla, B. Sahoo, A. Kumar Turuk, "Mathematical modeling of qos-aware fog computing architecture for iot services," in Proc. Emerging Technologies in Data Mining and Information Security: 13-21, 2019.

[33] T. H. Kim *et al.*, Smart city and IoT, Elsevier: 159-162, 2017.

[34] A. Yousefpour, C. Fung, T. Nguyen, K. Kadiyala, F. Jalali, A. Niakanlahiji, J. Kong, J. P. Jue, "All one needs to know about fog computing and related edge computing paradigms: A complete survey," J. Syst. Archit., 98: 289-330, 2019.

[35] A. Yousefpour, G. Ishigaki, J. P. Jue, "Fog computing: Towards minimizing delay in the internet of things," in Proc. 2017 IEEE international conference on edge computing (EDGE), 2017.

[36] Z. Wan, Z. Zhang, R. Yin, G. Yu, "KFIML: Kubernetes-based fog computing iot platform for online machine learning," IEEE Internet Things J., 9(19): 19463-19476, 2022.

[37] H. Zahmatkesh, F. Al-Turjman, "Fog computing for sustainable smart cities in the IoT era: Caching techniques and enabling technologies-an overview," Sustainable Cities Soc., 59: 102139, 2020.

[38] O. Skarlat, S. Schulte, "FogFrame: a framework for IoT application execution in the fog," PeerJ Comput. Sci., 7(22): e588, 2021.

[39] D. A. Chekired, L. Khoukhi, "Multi-tier fog architecture: A new delay-tolerant network for IoT data processing," in Proc. 2018 IEEE International Conference on Communications (ICC), 2018.

Biographies

Samira Kalantary received her B.S. and M.S. degree from Islamic Azad University, Arak, Iran, in 2003 and 2007 respectively, both in Computer Engineering. She is currently studying her Ph.D. in the Department of Computer Engineering, Islamic Azad University, Qom branch, Iran. Her research interests are in the area of Internet of things, fog computing, resource management and network.

- Email: kalantarysamira@gmail.com
- ORCID: 000-0002-1641-8626
- Web of Science Researcher ID: NA
- Scopus Author ID: NA
- Homepage: NA

Javad Akbari Torkestani received the Ph.D. degree in computer science from the Islamic Azad University in 2010. He is Associate professor in Islamic Azad university, Arak branch currently. His publication topics and his research interests are Intelligent system, Optimization, Grid computing and Web Engineering.

- Email: ja.akbari@iau.ac.ir
- ORCID: 0000-0002-6075-4889
- Web of Science Researcher ID:NA
- Scopus ID: 25222831700
- Homepage: NA

Ali Shahidinejad received the Ph.D. degree in computer science from the University Technology Malaysia, Johor Bahru, Malaysia, in 2015. He is currently working toward the second Ph.D. degree with Deakin University, Melbourne, Australia. His publication topics are Internet, Internet of Things, and distributed processing.

- Email: a.shahidinejad@gmail.com
- ORCID: 0000-0003-4856-9119
- Web of Science Researcher ID:NA
- Scopus ID: 54586030700
- Homepage: NA

How to cite this paper:
S. Kalantary, J. Akbari Torkestani, A. Shahidinejad, "New platform for IoT application management based on fog computing," J. Electr. Comput. Eng. Innovations, 11(2): 399-408, 2023.

DOI: 10.22061/jecei.2023.9489.626

URL: https://jecei.sru.ac.ir/article_1850.html