Leveraging Cloud Inter-zone Architecture for Response Time Reduction

Birane Koundoul[1(✉)], Youssou Kasse[1], Fatoumata Balde[1], and Bamba Gueye[2]

[1] University of Bambey, Bambey, Senegal
{birane.koundoul,youssou.kasse,fatoumata.balde}@uadb.edu.sn
[2] University of Dakar, Dakar, Senegal
bamba.gueye@ucad.edu.sn

Abstract. Technology has undergone a rapid evolution in recent years through cluster, grids, cloud and IoT. The latter is leading to the proliferation of data across various domains such as transport, health, environment. In the process, the number of connected devices continues to grow, generating an extraordinary amount of data that the traditional cloud is struggling to manage. In such a situation, the problems encountered revolve around high latency, a decrease in the level of quality of service, high bandwidth, enormous energy consumption. This situation justifies the birth of Fog Computing whose role is not only to collect data from connected objects (phones, vehicles, tablets) in points of presence placed as close as possible to the users. This leads to a reduction in response time. Based on architecture models with Fog nodes in a zone, we propose a new architecture with interconnected zones. It allows us to distribute requests between zones to reduce access to the cloud to reduce latency. Our solution is to interconnect zones in a double ring mode with a set of Fog nodes in each zone. Communication between Fog nodes using the gossip protocol and the distributed hash table for inter-zone communication. We will also propose an algorithm to favour access to Fog Computing over the Cloud. We will detail in the following sections.

Keywords: Fog computing · IoT · Data access · Distributed hash table · Performance evaluation · RdP · Graph theory

1 Introduction

In recent years, the cloud, which was seen as a key infrastructure for on-demand user services encompassing several domains such as commerce, applications, has been challenged by connected objects. These connected objects have undergone rapid evolution and are producing huge amounts of data. In [1,2] the authors estimated that the number of connected devices could reach 50 billion by 2020. This becomes a challenge for the cloud to meet the demands with minimal time. By 2011, smartphone traffic had far surpassed PC traffic. In the US, [2] show that 80% of the population uses smartphones. This confirms that the average number of connected devices per person will reach 6.58 Cisco reported in 2020.

Y. Faye et al. (Eds.): CNRIA 2021, LNICST 400, pp. 87–97, 2021.
https://doi.org/10.1007/978-3-030-90556-9_8

With the Internet of Things (IoT), there is data where real-time responses are expected. Therefore, query processing at the cloud level will not be attractive. In 2012, Cisco proposed a technology called Fog Computing [3,4]. This technology was not born to replace the cloud but to complement it. Because the cloud is far from the users, Fog Computing tries to address the issues of latency, quality of service, mobility.

Fog Computing allows the collection of data from connected objects (phones, vehicles, tablets) in points of presence placed as close as possible to the users and finally send responses at a low time. Fog Computing can be defined as an infrastructure for storing and processing data from connected objects. Fog nodes are heterogeneous in terms of processing performance, storage capacity and latency of access to objects and users. In [5], Kuljeet Kaur et al. argue that the integration of cloud computing and IoT is necessary not only to process the stored data but also to reduce latency. The cloud can process, store a large volume of data but it is somewhat remote from the users. This is why Fog Computing (From Core to Edge) has recently emerged to enable seamless convergence between cloud and mobile for real-time content, data delivery and processing [3,6]. Fog computing is a distributed architecture where data collection points are placed at the edge of users. Nodes are placed in different locations depending on their storage capacity. Those with limited resources are placed closer to the users, while the others (large resources) are placed at the core of the network. There are several research works that focus on latency minimisation among which we can mention: [6,7], on latency reduction based on data placement at Fog nodes. Solutions have been proposed in terms of exact algorithms, heuristics and metaheuristics. To solve this problem, we propose an architecture with interconnected zones in double ring mode. Our approach differs from existing methods in two important ways. First, it easily reduces the access to the cloud, and second, it also facilitates the communication of the Fog nodes.

The rest of the paper is structured as follows: in Sect. 2, we will review related work. Our architecture is presented in Sect. 3 with a comparison of some existing architectures. In Sect. 4, we define an algorithm to privilege the access to Fog Computing over the cloud. Section 5 presents the experimental results to reduce the access to the cloud. Finally, in Sect. 6, we end the paper with a conclusion and future works.

2 Related Work

Today's networks are complex and contain a large number of heterogeneous components. In fact, the use of Fog Computing can drastically reduce overall network latency [8]. In Fog Computing, the processing and storage components, called Fog nodes, are heterogeneous in terms of performance and storage capacity. In [9], the authors show that Cloud and Fog Computing provide on-demand services but neither of them can guarantee the quality of service of IoT based on delay sensitive applications alone.

Abedi et al. used a delay-sensitive application as a case study, which aims to monitor the health status of people in maritime environments. In this paper, Abedi et al. discuss an artificial intelligence-based task distribution algorithm (AITDA) using a broker between users and servers. This broker is responsible for receiving computing tasks from users and assigning them to servers. The limitation with this architecture is that the broker is not able to distinguish the best resource between the Fog and the Cloud. Furthermore, if the number of tasks available in the broker increases (the number of data arriving), the allocation of tasks to servers will take time.

In [10], Mostafa et al. proposed an algorithm for automated selection and allocation of Fog Computing resources (FResS). In this paper, they also proposed a model for predicting the execution time of a task. The FResS technique stores historical user and device (IoT) data by creating execution logs that can be used for future tasks. In this model, a new layer is placed between the Fog layer and the connected objects which will increase the number of hops.

In [11], Vasileios et al. proposed two architectures to reduce latency. In their first model, the architecture consists of a set of interconnected Fog nodes to facilitate data exchange. They improved their architecture by proposing zones with a set of Fog nodes to approve their storage powers. The limitations with his model are that there is no interconnection of the zones. This avoids direct zone communication. As a result, a request that is not processed in the zone will be redirected to the cloud. This implies a high latency.

According to [12] four graph models are considered the most widely used: the regular graph model, the Erdös Renyi (ER) network model, the Barabasi-Albert (BA) model and the Watts-Strogatz (WS) model. The Erdös Renyi model is a graph that is generated by a random process. In this model, two variants exist and are closely related to the random graph model. ER is a simple yet very rich model, which allows a large number of results to be obtained on large graphs. The number of edges at the vertices can vary. For the Barabasi-Albert model, it is a model that generates a graph with property. This means that some may have more neighbours than others. This model incorporates two concepts, namely growth and preferential attachment. The latter does not exist in the ER model. Growth means that the number of connected nodes increases over time; preferential attachment means that the more connected a node is, the more likely it is to receive new links (edges). It is a network model of scale invariants identifiable by the degree distribution of their nodes (the number of neighbours of a node) [13]. For the Watts-Strogatz model, it is a random graph generation model producing graphs with the small world property. That is, each node is a short distance from all other nodes in terms of hops but the node is not connected to all other nodes [13]. The Watts and Strogatz model is able to address both limitations encountered in ER. It solves the clustering problem while managing the short path between two nodes.

In Sect. 3, we will discuss our model (architecture) of interconnected areas with a set of Fog nodes using the Barabasi-Albert and Watts-Strogatz graphical models to reduce the distance between Fog nodes.

3 Architecture

In Fig. 1, we have represented our architecture with a set of interconnected zones in double ring mode. The main objective is to reduce access at the cloud level. Recall that Vasileios et al. in [11] proposed an architecture with unconnected zones. The latter are connected directly to the cloud, unlike our architecture where the zones are connected to the cloud but inter-zone communication is noted. Also at the level of each zone, several Fog nodes are placed and interconnected. This makes it possible to manage a large amount of user data in a geographical area. A Fog node in a zone is connected to several other Fog nodes in the same zone. The number of neighbours of the nodes is different because with the use of BA and WS models, the number of links between the nodes increases as they are connected. In addition, the gossip protocol is applied at

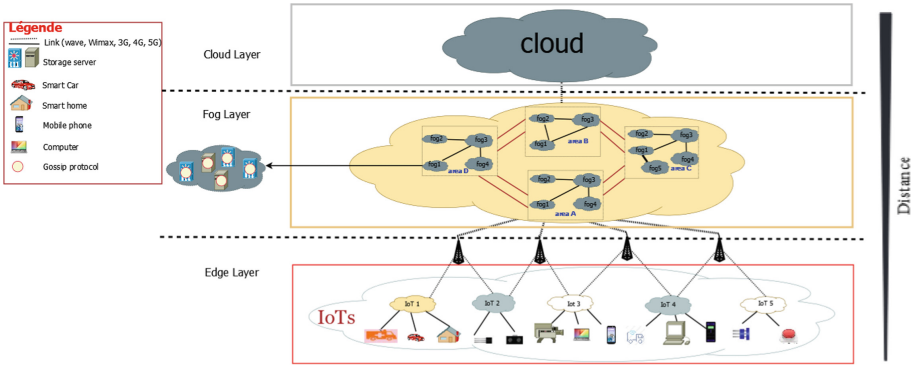

Fig. 1. Interconnection of zones in double ring mode.

each node. This protocol allows the Fog node to know all the objects stored by its neighbours. The neighbours of the Fog nodes communicate the stored information to each other. Each node has a key range which it stores and communicates with its neighbours. This facilitates access to the data. For example, a node A with its neighbours B and C, node A knows the objects stored in nodes B and C. This makes it possible to redirect the request directly to the node that can handle the request.

We have also defined an algorithm for zone switching. This algorithm is detailed in Sect. 4. The objective is to minimise access to the cloud. The redirection of the request in a zone is done by consulting the hash table (DHT). This consultation of the DHT makes it possible to target the zone likely to answer the request. The zones are connected in a double ring. This means that each zone has two neighbours and the advantage is that another link can be used if one of them fails. The principle is the same as for the ring topology, except that an extra ring is added as a backup in case the primary ring fails. At the level of each zone, a controller node is responsible for storing the set of keys (DHTs)

of the stored objects in the zone. This controller node communicates with its neighbours (the controller nodes of the neighbouring zones) the stored objects and an update message is sent by the nodes after each new addition or removal of data. Therefore, the controller node will update its hash table.

3.1 Model with Zone Interconnection in Double Ring Mode

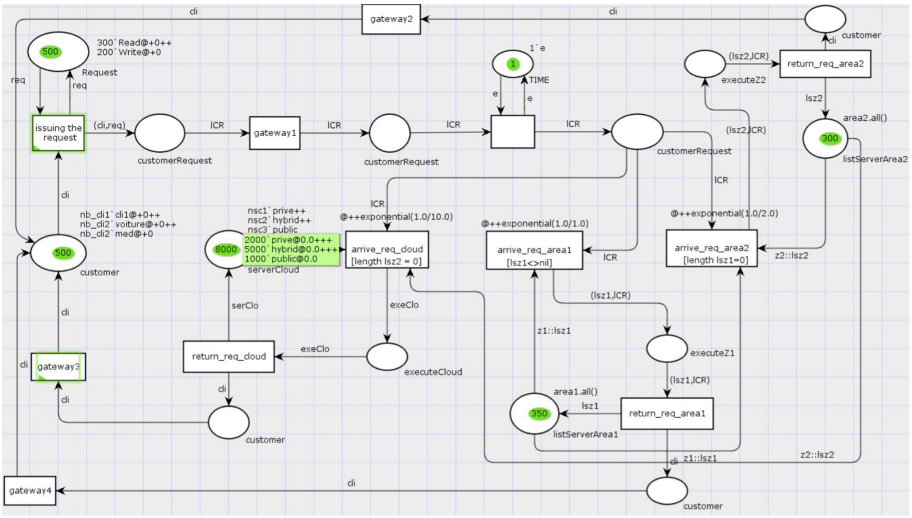

Fig. 2. Representation of the architecture with the petri nets.

Table 1. Number of servers per infrastructure

Infrastructure	Number of servers
Area 1	350
Area 2	300
Cloud	8000

In this article we have adopted the BA and WS graph models. This allows us to find the shortest path between two nodes but also to be able to increase the number of nodes in the network (in a zone) according to the number of connected objects in the zone. We have modelled the system infrastructure as an undirected graph with a set of vertices and edges connecting them. An undirected graph G is given by a section G = (S, A) with S a finite set of vertices and A a set of unordered pairs of vertices $\{s_i, s_j\} \in S^2$. The vertices of the graph are used to model the different Fog nodes existing in the system infrastructure and the edges are used to model the physical links between the different Fog nodes (see Fig. 2).

Table 2. number of requests by type of request

Type of request	Read	Write
Number of requests	300	200

Table 3. Comparative table of the result obtained with the two models

	Area 1	Area 2	Cloud	Lost	Maximum execution time (ms)
Model Vasileios et al. in [11]	200	161	139	0	96.44
Our model	202	191	60	47	56.85

The model we have proposed allows us to reduce access at the cloud level. With the number of requests sent by the connected objects, some of which require a minimum response time, it is preferable to reduce the access at the cloud level. Hence the interest of our model, as it consists of a set of zones with a set of Fog nodes for each zone. The interconnection of the Fog nodes is done according to the graphical models of Barabasi-Albert and Watts-Strogatz. These models make it possible to find the shortest path between two nodes but also to be able to increase the number of nodes in the network (in a site) according to the number of connected objects in the area. The table (Table 1) shows the number of servers used in each zone of the Fog and cloud computing infrastructure and the Table 2 shows the number of read and write requests sent by the client. We have a comparison of the results obtained after simulation in the Table 3.

4 Algorithm to Favour Fog Computing over Cloud

In our algorithm, the client after sending the request, a gateway check is done to determine if the request will be forwarded or returned to the client. The request is lost if it is returned to the client. If not, it will be forwarded and this will allow it to be sent to a Fog node or to the cloud. When transmitting the request, our algorithm favours Fog nodes over the cloud. This reduces the response time of the request. A Fog node is available if it has the resources to satisfy the request. However, each node is located in an area covered by a base station (BS). In addition, at the level of each node, the gossip protocol is applied which allows the nodes to know all the objects stored by its neighbours. Therefore, if the target node cannot process the request, it will forward the request to its neighbour that can process the request. This process remains in the same zone (same base station) because in each zone, there is a node that plays the role of controller. This allows it to know all the objects stored in the zone. After a consultation of its distributed hash table (DHT), this controller node will redirect the request to another zone (a susceptible zone). This change of zone implies the change of base station. At the time of the change, two cases can happen: either the request is lost if the change was not successful or the request is transferred to another zone

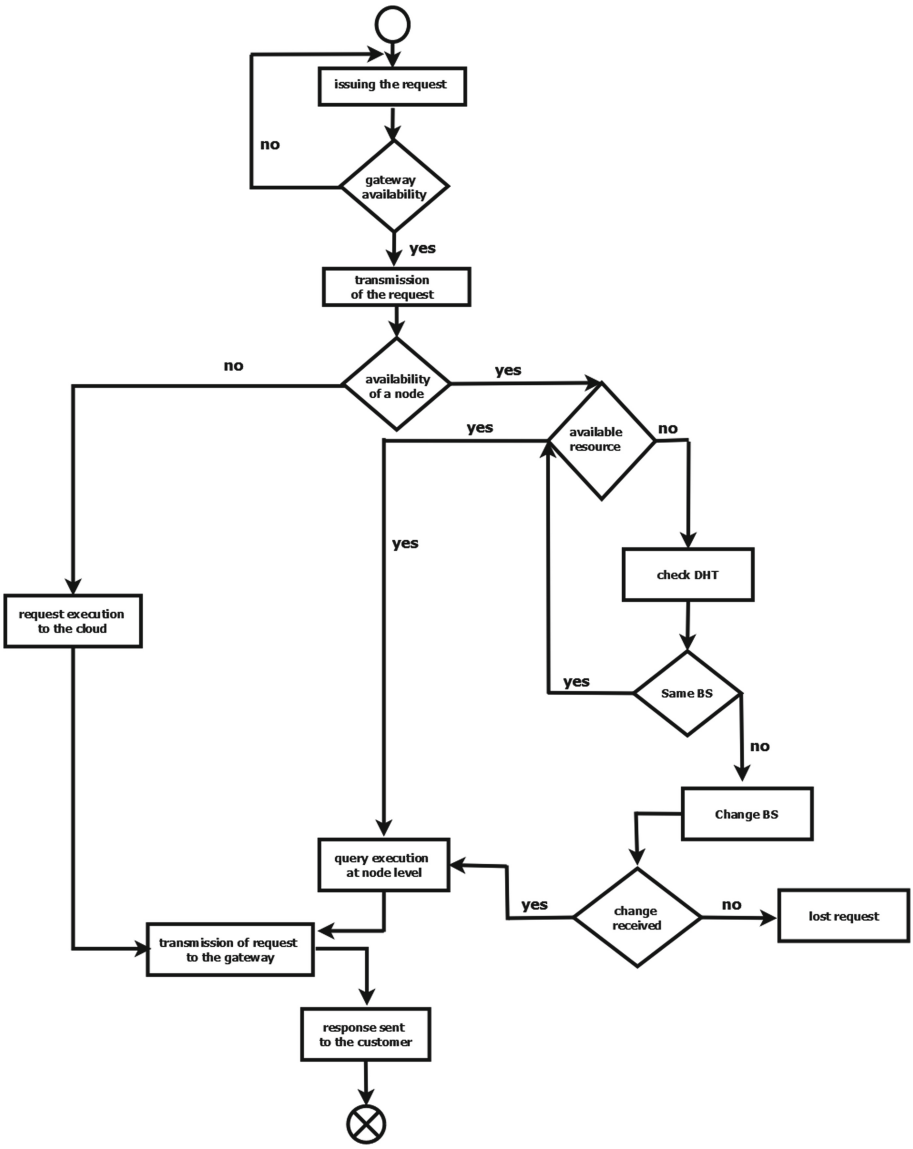

Fig. 3. Algorithm to prioritise access to Fog Computing over the Cloud

(successful change). At the moment the change of zone has been successful, the same process is applied as the processing of the incoming zone. After processing the request at a Fog or Cloud node, the response is transferred to the gateway so that the gateway can forward the response to the client. The advantage of our algorithm is that it reduces access to the cloud which is far away from the end users (Fig. 3).

5 Results

After simulation with the CPN tools, we compared the model of Vasileios et al. in [11] with our model. We obtained results that allow us to judge our model as interesting. We placed a limited number of servers in Fog zones (see Table 1) for a number of 500 requests of different types (see Table 2). Figure 4 shows the number of requests executed in the two zones and in the cloud. This shows that there is not a big difference in the number of requests executed between the zones and the cloud. A large number of requests are executed at the cloud level which implies the high response time (see Table 3). Figure 5 shows the results

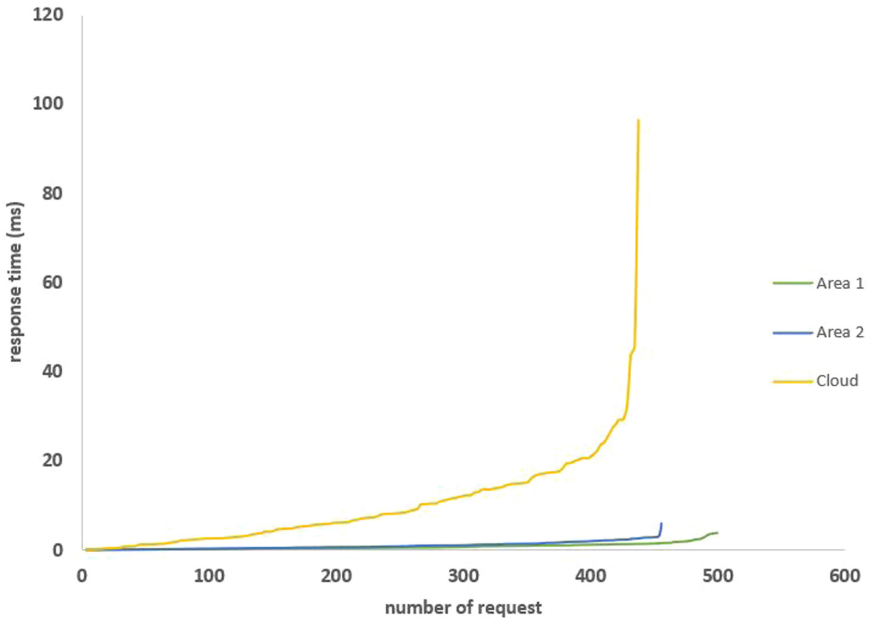

Fig. 4. Response time of 500 requests in an architecture without zone interconnection.

obtained in our model. We notice that compared to the model of Vasileios et al. in [11], a model without zone interconnection (MWHI), we were able to reduce the number of requests to the cloud. This implies the reduction of the response time (see Table 3). In contrast to our model where an interconnection of zones is noted (MWI). As our goal is to reduce the access to the cloud, our algorithm consists in prioritising the requests to the zones for processing. When a gateway is available, we check if a Fog node is ready to process the request, if so, the Fog node processes the request, if not we move to the next Fog node. This check is done as long as the zone contains a Fog node before moving on to another zone. We can say that we have achieved our goal of reducing access to the cloud.

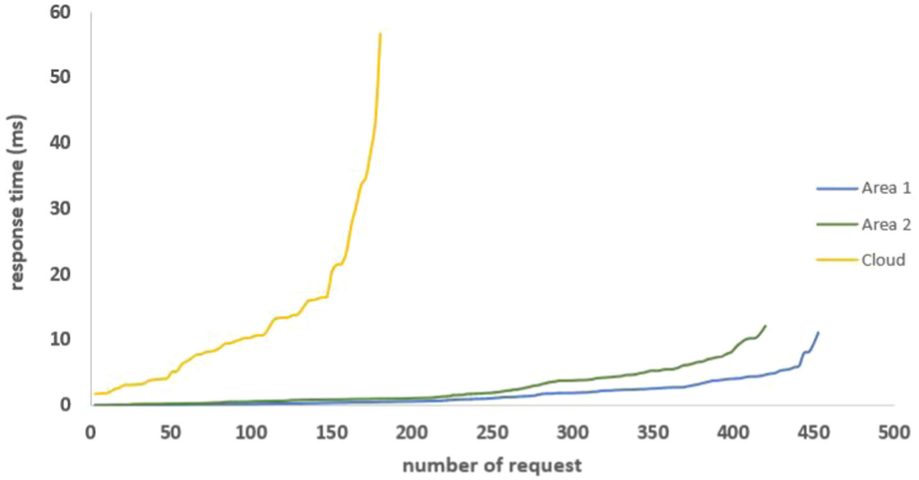

Fig. 5. Response time of 500 requests in an architecture with zone interconnection.

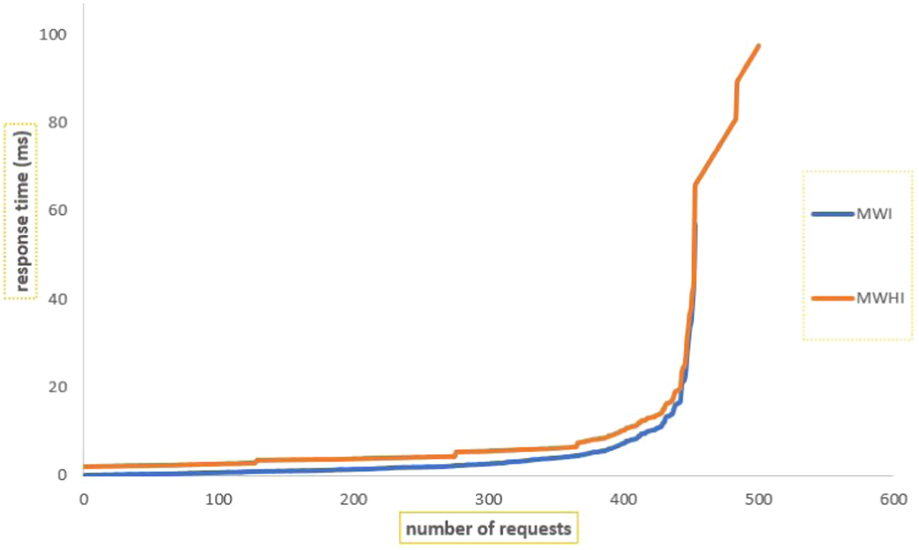

Fig. 6. Performance of both models in terms of response time.

This will allow us to reduce the response time of requests. In Fig. 6, we have the performance of the two models in terms of response time. We see that each model performs well but with some limitations. In terms of response time, our model is better because it allows to reduce the access to the cloud. This is not the case with the MWHI model. In this model, requests will be redirected to the cloud if the zone cannot handle the request.

5.1 Limitations of Our Model

After comparison with the model of Vasileios et al. in [11], we noted some limitations in our model. First, we found:

- lost requests in our model (see Table 3).
- a linear evolution of the response time at the cloud level.

The lost requests are due to the change of zones with a check of the available resources in the neighbouring zones.

6 Conclusion

In this paper, the main idea was to propose an architecture with a set of interconnected zones to solve latency problems. In addition, given the interconnectedness of the zones, an algorithm is proposed to reduce the access to the cloud. This allowed in this simulation to have, a limited number of requests sent to the cloud (60) compared to the model of [11] (139) and a decrease in response time (see Table 3). For the next step, we will try to see how to balance the loads between the Fog nodes and even the zones. In addition we will try to answer the question: how the data placement would be interesting to reduce the response time. Finally, we will try to use an extension of the coloured petri nets which is the hierarchical modular coloured petri nets to take back the zones in the form of modules. This will allow us to reproduce an existing module to create others.

References

1. Alam, F., Mehmood, R., Katib, I., Albogami, N.N., Albeshri, A.: Data fusion and IoT for smart ubiquitous environments: a survey., IEEE Access **5**, 9533–9554 (2017)
2. Luan, T.H., Gao, L., Li, Z., Xiang, Y., Wei, G., Sun, L.: Fog computing: focusing on mobile users at the edge. ArXiv150201815 Cs, mars 2016
3. Bonomi, F., Milito, R., Zhu, J., Addepalli, S.: Fog computing and its role in the internet of things. In: Proceedings of the first edition of the MCC workshop on Mobile cloud computing, Helsinki, Finland, août 2012, pp. 13–16 (2012)
4. Gupta, H., Dastjerdi, A.V., Ghosh, S.K., Buyya, R.: iFogSim: a toolkit for modeling and simulation of resource management techniques in Internet of Things, edge and fog computing environments., ArXiv160602007 Cs, juin 2016
5. Kaur, K., Garg, S., Kaddoum, G., Ahmed, S.H., Jayakody, D.N.K.: 'En-OsCo: energy-aware osmotic computing framework using hyper-heuristics. In: Proceedings of the ACM MobiHoc Workshop on Pervasive Systems in the IoT Era, Catania, Italy, juill. 2019, pp. 19–24 (2019)
6. Stojmenovic, I., Wen, S., Huang, X., Luan, H.: An overview of Fog computing and its security issues. Concurr. Calc. Prat. Exp. **28**(10), 2991–3005 (2016)
7. Taneja, M., Davy, A.: Resource aware placement of IoT application modules in Fog-Cloud Computing Paradigm. In: Conference: 2017 IFIP/IEEE Symposium on Integrated Network and Service Management (IM), pp. 1–7 (2017)

8. Naas, M.: iFogStor: an IoT data placement strategy for fog infrastructure. In: Conference: 2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC), pp. 1–8 (2017)
9. Abedi, M., Pourkiani, M.: Resource allocation in combined fog-cloud scenarios by using artificial intelligence. In: 2020 Fifth International Conference on Fog and Mobile Edge Computing (FMEC), avr. 2020, pp. 218–222 (2020)
10. Mostafa, N., Ridhawi, I.A., Aloqaily, M.: Fog resource selection using historical executions. In: 2018 Third International Conference on Fog and Mobile Edge Computing (FMEC), avr. 2018, pp. 272–276 (2018)
11. Moysiadis, V., Sarigiannidis, P., Moscholios, I.: Towards distributed data management in fog computing. Wirel. Commun. Mob. Comput. (2018)
12. Khan, B.S., Niazi, M.A.: Modeling and analysis of network dynamics in complex communication networks using social network methods. ArXiv170800186 Cs Math, août 2017
13. Albert, R., Barabási, A.-L.: Statistical mechanics of complex networks. Rev. Mod. Phys. **74**(1), 47–97 (2002). https://doi.org/10.1103/RevModPhys.74.47