# On the Influence of Fog Colonies Partitioning in Fog Application Makespan

Carlos Guerrero
Computer Science Department
University of Balearic Islands
Crta. Valldemossa km 7.5.
E07122 Illes Balears (SPAIN)
Email: carlos.guerrero@uib.es

Isaac Lera
Computer Science Department
University of Balearic Islands
Crta. Valldemossa km 7.5.
E07122 Illes Balears (SPAIN)
Email: isaac.lera@uib.es

Carlos Juiz
Computer Science Department
University of Balearic Islands
Crta. Valldemossa km 7.5.
E07122 Illes Balears (SPAIN)
Email: cjuiz@uib.es

***Abstract*—This paper presents a study of the use of network centrality indices as suitable indicators to determine the partitioning of fog colonies. Fog colonies have been used previously to enable a two level (inter and intra colony) resource management for the fog service placement problem. We propose selecting the nodes with highest values, which indicate the node importance, as the colony controllers. The remaining devices are subordinated to the closest controllers. We studied six centrality indices in three network topologies and two architecture sizes. The fog applications are designed as a set of interoperated services which makespan is increased when services are allocated in different colonies, or colonies have highest intra or inter distances. Consequently, we considered the network distance as indicator of the application makespan. The results showed that the smaller network distances was obtained with the Betweenness centrality index and a Barabasi-Albert network topology. It was also observed that the network distance only has significant differences when the colony size is varied between 1 and 20.**

***Index Terms*—Fog computing, fog colonies, complex networks, resource optimization.**

## I. Introduction

The number of applications developed for the Internet of Things (IoT) has been increased significantly in the last few years. In those applications, any device, however small, is able to connect to the Internet and collect, process, and share data from physical elements. As these environments have become popular, their processing requirements have increased rapidly, making necessary the use of more powerful resources.

A first solution to provide higher capacities to the IoT applications was to integrate cloud services by storing and processing the data of the IoT devices in the cloud providers, and sending the results back to the IoT devices. This earlier solution allowed to develop applications that interact with real devices and that have unlimited storage and processing capabilities. But important drawbacks emerged related with communication delays, network usages, and service costs to send high quantities of data between the IoT devices and the cloud providers.

Meanwhile, the network communication devices have increased their computational capacities, and a new paradigm has been explored, the Fog Computing [1]. A fog architecture provides computational and storage capacities in in-network devices, also called fog devices. By this, the communication nodes (fog devices) are able to allocate services to reduce the network latency of the applications, since the services are closer to the IoT devices. Important challenges arise in the management of those services across the fog devices [2].

Some preliminary works have proposed two-level architecture for the management of the services. For example, fog colonies have been proposed as a fog device organization to improve the service placement [3]. A fog colony is a set of fog devices with one device in charge of the coordination of the colony. All the devices are associated to one disjoint fog colony, i.e., is managed by on fog colony coordinator. In those architectures, there are two service management levels: a higher level where the services are allocated to fog colonies, conducted by a central manager, and a lower level where the services are allocated to the fog devices within the colony, conducted by the colony coordinator.

Several efforts have been addressed to improve the policies that allocate the services to the colonies or to the fog devices. But to the best of our knowledge, there is not any previous study that deals with the partition of fog colonies. We consider that it is important to study how the organization, number, and size of fog colonies influence on the system performance.

We propose the study of the benefits of using complex network metrics to create and define the fog colonies. Among all topological measures that can be analyzed from complex network theory [4], we have weighted centrality and clustering metrics as suitable ones. Thus, we consider six centrality and clustering approaches (Betweenness, Degree, Generalized degree, Closeness, Eigenvector and Clustering) and we study the application makespan for different colony sizes. We conduct the study in three network topologies (Lobster, Random Euclidean and Barabasi-Albert).

The main contributions of our work are: (a) The study of the influence of the fog colony structure in the application makespan (b) A preliminary proposal for fog colony creation based on clustering and centrality indices of complex networks as suitable indicators for the device distribution; (c) A colony partitioning aware optimization of the fog application makespans.

The makespan of the applications that are deployed in a fog architecture can be damaged by the spread of the services

978-1-5386-7503-8/18/$31.00 ©2018 IEEE
DOI 10.1109/FiCloud.2018.00061

across different fog devices. Moreover, the communication pattern of the fog colonies, where all the messages between the services are routed across the colony controllers, increases this risk. Consequently, the distance between the devices and their fog controller (intra colony distance) and the distance between the colonies, measured in our case in terms of the distance to the closest colony, influence the makespan of the application. Therefore, we have based our study in these both indicators.

Our exploratory study is addressed to answer three main research questions: RQ1. Do the size of the architecture or of the colonies influence in the intra colony and closest neighbor distances?; RQ2. Is there any difference in the intra colony and closest neighbor distance between different centrality indices? Does anyone obtain better results than the other indices?; RQ3. Does the network topology influence in the distance indicators? Does anyone obtain better results than the other indicators?

The structure of the paper is as follows: Section II summarizes some related researches; we formally establish a model of our scenario in Section III; Section IV explains the details of our partitioning algorithm based on the use of centrality indices for complex networks; Section V includes the experiments we carried out to evaluate our proposal; finally, the conclusion and future works are included in Section VI.

## II. Related Work

Previous studies have mainly focused the optimization of fog architectures on defining policies for service placement across the fog devices. Those placement strategies have been implemented with several optimization algorithms. For example, linear programming, evolutionary algorithms and greedy approaches are common solutions.

Huang et al. [5] presented a quadratic programming formulation to reduce the power consumption. Souza et al. [6] proposed an integer linear programming (ILP) to reduce latency and to guarantee the fulfillment of capacity requirements. Skarlat et al. [7] studied that an ILP optimization obtained improved times and cost while satisfying the QoS requirements.

Some other proposals have solved the optimization process using genetic algorithms (GA). For example, Wen et al. [8] presented a preliminary proposal based on a parallel GA to reduce response time and increase the QoS. Yang et al. [9] studied service placement and load dispatching optimization in mobile cloud systems. They proposed three optimization algorithms based on a greedy heuristic, a linear programming and a genetic algorithm. Colistra et al. [10] adapted the consensus algorithm to allow devices to cooperate in a distributed resource allocation management. Deng et al. [11] studied the trade-off between power consumption and delay in fog computing. They decomposed the initial allocation problem into three subproblems independently solved with convex optimization, ILP and Hungarian method. Bittencourt et al. [12] compared three service allocation algorithms to illustrate that these strategies depend on the demand coming from mobile users and can take advantages of fog proximity and cloud elasticity.

Most of those previous works were modeled as a centralized broker or orchestrator that needs information from all the components in the system (fog devices, clients, cloud, services) and takes global decisions to optimize the service placement. Problems with the scalability and the computational complexity of the algorithm are clear when the number of elements is very high such as, for example, in smart cities. It is necessary to define solutions that deal with a smaller number of elements. To solve this problem, several researches have proposed hierarchical organization of fog devices by defining subsets of devices that are self-managed.

Skarlat et al. [3] proposed to organize the fog devices as colonies, and each colony was in charge of deciding the services or applications modules that were placed inside the colonies and which ones were propagated to neighbor colonies. The optimization algorithm was implemented as a GA that maximized the utilization of the fog landscape while the deadlines of the application execution times were satisfied. They used the iFogSim to compare their proposal with placement strategies based on first-fit greedy algorithm, ILP optimization and a deterministic strategy that placed all the modules in the cloud. Their results showed a more effective use of the fog resources. Similar set-based organizations of the fog devices have been also proposed, such as micro-clouds [13], Foglets [14], or fog domains [15].

In those types of architectural organization, not only the placement policy has a direct influence in the performance of the system, but also the own organization of the elements. The number and size of the device sets, and the partition criteria have an important influence in the optimization of the system.

We propose an exploratory study to analyze topological and organization influence for the case of the fog colonies. We study the suitability of several complex network centrality and clustering indices as the metric to determine the most important and influent nodes to be defined as controllers of the fog colonies. To the best of our knowledge, this is the first study that addresses the optimization of the service placement based on the partitioning of the fog colonies instead of the placement policies. We have previously studied the suitability of those centrality indices to optimize data placement in fog architectures [16].

## III. System Model

Our architecture is based on the fog computing framework presented in Skarlat et al. [3]. The elements of the architecture are[1]: (a) fog devices $f^j$, network devices with processing capacities; (b) controller device $F$, a fog devices that is in charge of managing a set of fog devices; and (c) fog colony, $Res(F)$, a disjoint set of fog devices that has a controller and several fog devices. The controller device represents the head of a colony and it distributes the services among the subordinated fog devices. The framework consists on twofold

[1]We have used the same notations that in the article of Skarlat et al. [3]

resource management, a first global service allocation between the fog colonies, and a second allocation between the devices in a colony. If a service cannot be deployed in the designated colony, it is migrated to the closest neighbor colony.

Those elements are connected through a network. The devices are modeled as nodes of a graph structure, and the edges of the graph represent a direct network connection between two devices, $l^{f^j - f^{j'}}$, which is characterized by its network latency $d^{f^j - f^{j'}}$.

The network latency $D(f^a, f^b)$ between two any devices in the network can be calculated as the summation of the single network latencies of each link in the shortest path between them:

$$D(f^a, f^b) = \sum d^{f^j - f^{j'}}, \forall\ l^{f^j - f^{j'}} \in ShortestPath(f^a, f^b) \tag{1}$$

Skarlat et al. proposed to optimize the response time of an application. They modeled the applications following the distributed dataflow model [17], where the applications are designed as a set of interoperated services. Those services can be allocated in different fog devices. They proposed a policy that determines the service allocation to reduce the response time while the device resource limits are guaranteed. They calculated the response time of an application as the summation of the service execution times and the communication times between the devices that allocate the services. The communication time of services allocated inside the same fog colony is calculated as the distance of the shortest path between them. The communication between services in different colonies is routed across the controllers and, consequently, the communication time is calculated as the summation of (i) the shortest path between the fog device with the origin service and its controller, (ii) the shortest path between the controller devices of both colonies, and (iii) the shortest path between the fog device of the target service and its controller. An example of the communication is represented in Fig. 1. This example corresponds to an architecture with three colonies and one application of three services deployed in two of those colonies. The first service interoperates with a service in the second colony and this second service request a last service in the first colony. By this, two inter colony comunications (d2 and d5) and 4 intra comunications happens, with the following communication sequence $d1 + d2 + d3 + d4 + d5 + d6$, resulting in a total network distance of $1.93{+}2.5{+}1.95{+}1.95{+}2.5{+}1.8 = 12.63$.

They addressed the optimization as a service allocation problem, considering a static organization of the fog colonies. We propose to conduct a infrastructure optimization previous to the deployment of the fog architecture. This optimization process is addressed to determine the hierarchy of the colonies: number, size, controller devices and subordinated devices of each controller.

The organization of the fog colonies directly influences in three main aspects: (a) the network communication time between the controllers and their subordinated devices; (b) the network communication time between controller devices; (c) the resource capacity of the fog colonies. Since our approach is focused on a previous phase to the architecture deployment, the resource consumption of the applications are unknown at that phase. Consequently, we limit our study to the first two aspects.

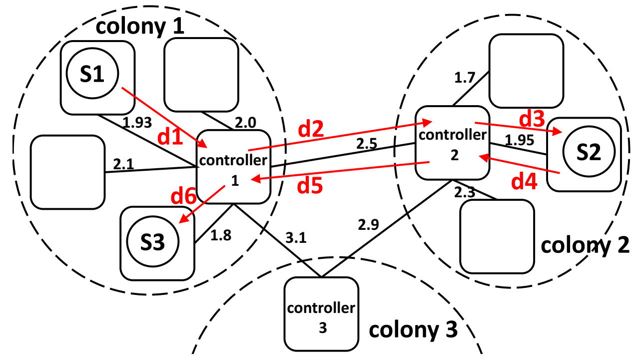



Fig. 1. Example of communication sequence for an application with 3 services.

The metric that we propose for the first aspect is the average of the inter-cluster distance of each fog colony, $\overline{IntraDist}$. We first measure the average of the distances between the controller device and the subordinated devices $IntraDist_{Res(F)}$ for each colony, Eq. 2, and finally calculate the average of those inter-cluster distances, Eq. 3.

$$IntraDist_{Res(F)} = \sum^{f^j \in\ Res(F)} \frac{D(f^j, F(f^i))}{|Res(F)|} \tag{2}$$

$$\overline{IntraDist} = \sum^{\forall Res(F)} \frac{IntraDist_{Res(F)}}{colony\ number} \tag{3}$$

where $F(f^i)$ is the controller device of the fog colony where $f^i$ is associated to.

For instance, if we consider the example in Fig. 1, the results of the intra distances for the two first colonies are calculated as $IntraDist_{colony1} = \frac{1.0+1.93+2.1+1.8}{4} = 1.73$, $IntraDist_{colony2} = \frac{1.7+1.95+2.3}{3} = 1.98$, and for the third colony we supposed $IntraDist_{colony1} = 1.3$ since the subordinated devices are not included in the figure. Finally, the average of the intra distances is $\overline{IntraDist} = \frac{1.73+1.98+1.3}{3} = 1.67$.

For the second aspect, we consider the average value of the distances between a controller and its closest controller, $\overline{ClosestNeighDist}$. We select this metric because the fog framework commonly migrates the non-allocated services to the closest neighbor colony. This is represented as:

$$\overline{ClosestNeighDist} = \sum^{\forall Res(F)} \frac{D(F, neighbor(F))}{colony\ number} \tag{4}$$

where $neighbor(F)$ is the controller device of the closest neighbor colony.

**Algorithm 1** Pseudo-code of the fog colony partition algorithm.

**Function:** ColonyPartition()
**Input:** network topology G, number of colonies K
**Output:** fog colonies partitions Res()

nodesValues = CalculateCentralityIndex(G)
ordered = sort(nodesValues)
controllers = ordered[0..K]
**for all** $f^i$ **do**
  closestF = null
  **for all** $F \in$ controllers **do**
    **if** $D(f^i, F) < D(f^i, closestF)$ **then**
      closestF = F
    **end if**
  **end for**
  Res(closestF) = Res(closestF) $\cup$ $f^i$
**end for**
**return** Res()

Considering the example in Fig. 1, the closest neighbor distances for each controller are 2.5 for the first and second controller, since both of them are their own closest neighbors, and 2.9 for the third controller, whose closest neighbor is controller 2. Consequently, the $\overline{ClosestNeighDist} = \frac{2.5+2.5+2.9}{3} = 2.63$.

We base the study of the fog architecture on the analysis of the intra colony distance and the closest colony distance, considering the best cases when these metrics are minimized: $minimize \quad \overline{IntraDist} + \overline{ClosestNeighDist}$.

## IV. Fog Colony Partitioning Algorithm

We conduct an exploratory research comparing several centrality and clustering indices, in three network topologies, to determine the controller devices and the partition of the fog devices in fog colonies. The objective is to evaluate how the intra colony distance and the closest neighbor distance vary between the studied scenarios, to find the index and the topology that reduce the most those metrics.

We propose to use the centrality indices to select the controller devices of the fog colonies. Thus, the phases to partition the devices into colonies would be: (i) model the network topology as a complex weighted network; (ii) calculate the value of the centrality/clustering index of each node; (iii) select the first $k$ nodes with the highest indices, where $k$ is the number of fog colonies to be created; (iv) partition the fog devices into colonies by subordinating each devices to its closest controller device. The pseudocode of the algorithm is presented in Algorithm 1. The number of fog colonies needs to be known before the creation of the colonies.

We have considered six standard and generalized indices to implement the function *CalculateCentralityIndex(G)* in Algorithm 1. Centrality and clustering indices denote an order of importance on the nodes of a graph by assigning a value, and they only depend on the structure of the graph. They are commonly used to detect the most central and important nodes in the graph [18].

The six indices we consider are briefly described as: (i) *Degree centrality* is based on the number of links incident upon a node. This is interpreted as an indicator of the probability that a communication process would pass through it; (ii) *Closeness centrality* of a node is the average length of the shortest path between the node and all other nodes in the graph. It is an indicator of the nodes that are closer to other nodes; (iii) *Eingenvector centrality* measures the influence of the node in a network using the concept that high-connected nodes contribute more to the score of the index. (iv) *Betweenness centrality* quantifies the number of times a node acts as a bridge along the shortest path between two other nodes. It measures the control of the node on the communication between other two nodes; (v) *Generalized degree* shows how many edges of given triangle multiplicity the node is connected to. The triangle multiplicity of an edge is the number of triangles an edge participates in; and finally, (vi) the *Clustering coefficient*, that is the fraction of possible triangles through that node that exist, for unweighted graphs, and the geometric average of the subgraph edge weights, for weighted graphs.

TABLE I
CHARACTERISTICS OF THE NETWORK TOPOLOGIES

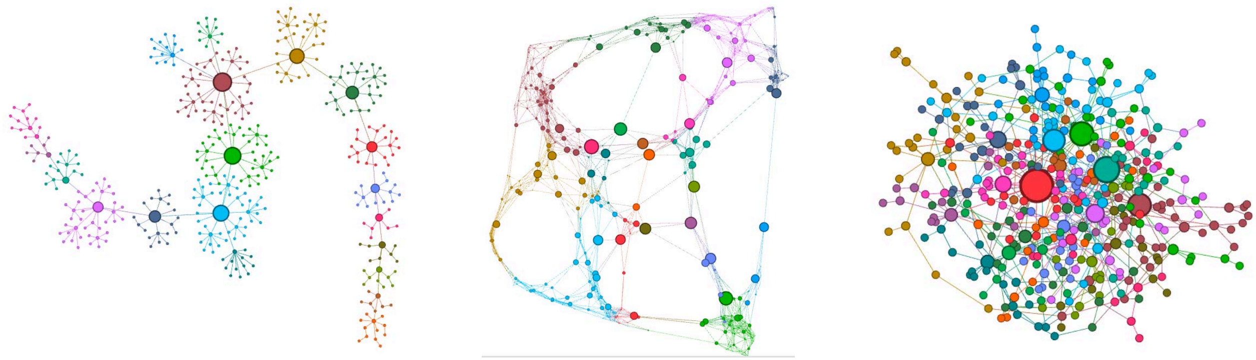
| Topology | Number of nodes | Number of edges | Average shortest path |
|---|---|---|---|
| Euclidean | 400 / 1000 | 2242 / 14402 | 7.33 / 6.30 |
| Lobster | 400 / 1000 | 400 / 1000 | 7.31 / 7.76 |
| Bar.-Alb. | 400 / 1000 | 596 / 1514 | 4.27 / 4.67 |

## V. Experimental Evaluation

To the best of our knowledge, there is any previous study that characterizes the network topology of a fog domain. Consequently, we considered several network topologies. They were selected from common topologies in the field of complex networks [19]. The experiments were designed as several scenarios that were varied in their network topology (Lobster, Euclidean, and Barabasi-Albert) and number of devices (400 and 1000 fog devices). For each of the experiment scenarios, the intra colony and closest neighbor distances were measured as application makespan indicators. The number of colonies were varied from maximum (the number of devices) to minimum (one colony) and the resulting distances for those number of colonies were measured. The experiments were repeated for the six centrality and clustering indices we considered. Those centrality indices were computed using the NetworkX library [20], and our partitioning algorithm was implemented in Python 2.7.

The network topologies were randomly generated using the GraphStream library[2]. Three network topologies were considered: Barabasi-Albert, Random Euclidean and Lobster. The default values of the libraries were used for the case of the

[2]Available on-line in http://graphstream-project.org/



(a) Lobster topology (b) Random Euclidean topology (c) Barabasi-Albert topology

Fig. 2. Network topologies for the experiment scenarios with 400 nodes. The size of the nodes is directly proportional to the value of the Betweenness centrality index. The colonies obtained for the case of size 20 are represented with colors.

Random Euclidean (a dimension of 2 for the graph's space) and the Lobster (a maximum degree of 10 and a maximum distance of 2 from any node of the root path). The parameter *maxLinksPerStep* of the Barabasi-Albert generator was set to 2. Table I shows some features of the resulting network topologies. Additionaly, Fig. 2 shows the topologies obtained for the case of 400 fog devices to have a deeper idea of the network topology organization.

As we were interested in comparing the suitability of the centrality indices, we defined the experiments as independent of other factors as possible. Consequently, the network distance between two directly connected nodes $(f^j, f^{j'})$ were fixed in $d^{f^j - f^{j'}} = 1.0$ generic network distance units.

### A. Results

The results are presented to compare the different elements that influence in the network distances between the devices in the fog colonies. Fig. 3 and Fig. 4 show the evolution of the network distance along the colony sizes for the cases of, respectively, 400 and 1000 fog devices. These figures include three plots, one for each network topology, and the results of the distances can be easily compared between the indices.

The y-axis of the plots represent the network distance, i.e., the summation of the intra and the closest neighbor distances, that we used as an indicator of the application makespan. Note that smaller values of the network distance are desirable. The x-axis are the mean size of the colonies. The experiments were varied by chaging the total number of colonies, i.e. controller devices. But for a better understanding of the results, these are presented in terms of the mean colony size, that is easily obtained with $\overline{Colony\ Size} = \frac{Total\ Devices}{Number\ of\ Colonies}$.

Fig. 5 is presented for a more clear comparison of the results between network topologies and total number of devices in the architecture. For the sake of clarity, only the index with the previous best results (Betweenness centrality) is represented in those figures.

For better comparison between the results, some specific values from the results have been numerically represented in TABLE II, mean colony sizes of 20, 8, and 4 devices per colony. The results are desegregated for the closest neighbor and intra colony distances for each centrality index. Note that the smaller the results are, the better for the application makespan.

Finally, we have also graphically represented the results of the fog colony partitioning for the cases of 400 fog devices and the Betweenness centrality in Figure 2. The size of the nodes is directly proportional to the value of the centrality index. The colors represent the fog colonies in the case of 20 colonies and consequently a mean colony size of 20.

All those results helped us to answer several research questions. They also helped us to determine some general rules when the partitioning into fog colonies is done.

### B. Discussion

The first research question we dealt with was if the size of the architecture influenced in the distance results. By the analysis of Fig. 3 and Fig. 4, the first conclusion was that the total number of fog devices in the architecture had a low influence in the distance results since the results for 400 devices were quite similar to the results with 1000 devices. For example, in the case of the Betweeness centrality (Fig. 5), the differences of the network distances between experiment sizes of the same topology resulted in around 2% for the Lobster, 5.5% for the Random Euclidean, and 1.5% for the Barabasi-Albert. Meanwhile, the differences between topologies range between 15% and 35%.

The general behavior of the distance metric is reflected in Fig. 3, 4, and 5, where it is observed that the colony size had an important influence on the results: the network distance was increased as the colony size was also increased. For example, in the case of the Betweeness centrality (Fig. 5), the differences in the network distance between small and big colony sizes were higher than 150%. This could make us to think that the application makespan is reduced by reducing the number of devices per colony, obtaining the optimized case with only one device per colony. But this conclusion is wrong because, as we mentioned in Section III, the resource capacity

TABLE II
RESULTS FOR THE NETWORK TRANSMISSION METRICS FOR COLONY SIZES OF 20, 8, AND 4.

| **Colony size** | **Betweenness** Cl.[1] / In.[2] / Tot.[3] | **Degree** Cl.[1] / In.[2] / Tot.[3] | **General. Deg.** Cl.[1] / In.[2] / Tot.[3] | **Closeness** Cl.[1] / In.[2] / Tot.[3] | **Eigenvector** Cl.[1] / In.[2] / Tot.[3] | **Clustering** Cl.[1] / In.[2] / Tot.[3] |
|---|---|---|---|---|---|---|
| Lobster - 400 nodes | | | | | | |
| 20 | 1.00 / 1.54 / 2.54 | 1.23 / 1.75 / 2.98 | 1.23 / 1.75 / 2.98 | 1.00 / 2.47 / 3.47 | 1.23 / 1.75 / 2.98 | 3.27 / 2.50 / 5.77 |
| 8 | 1.00 / 1.07 / 2.07 | 1.00 / 1.07 / 2.07 | 1.00 / 1.07 / 2.07 | 1.00 / 1.78 / 2.78 | 1.00 / 1.07 / 2.07 | 2.42 / 1.88 / 4.30 |
| 4 | 1.00 / 0.75 / 1.75 | 1.00 / 0.75 / 1.75 | 1.00 / 0.75 / 1.75 | 1.00 / 1.62 / 2.62 | 1.00 / 0.75 / 1.75 | 1.80 / 1.33 / 3.14 |
| Random Euclidean - 400 nodes | | | | | | |
| 20 | 1.05 / 2.28 / 3.33 | 1.09 / 2.53 / 3.62 | 1.23 / 2.68 / 3.90 | 1.00 / 3.52 / 4.52 | 1.09 / 2.53 / 3.62 | 1.91 / 2.63 / 4.54 |
| 8 | 1.02 / 1.54 / 2.56 | 1.08 / 1.71 / 2.78 | 1.06 / 1.77 / 2.82 | 1.00 / 3.11 / 4.11 | 1.08 / 1.71 / 2.78 | 1.25 / 1.68 / 2.93 |
| 4 | 1.01 / 1.02 / 2.03 | 1.00 / 1.14 / 2.14 | 1.00 / 1.15 / 2.15 | 1.00 / 1.85 / 2.85 | 1.00 / 1.14 / 2.14 | 1.09 / 0.95 / 2.04 |
| BarabasiAlbert - 400 nodes | | | | | | |
| 20 | 1.00 / 1.40 / 2.40 | 1.05 / 1.38 / 2.43 | 1.00 / 1.62 / 2.62 | 1.00 / 1.48 / 2.48 | 1.05 / 1.38 / 2.43 | 1.00 / 1.62 / 2.62 |
| 8 | 1.02 / 1.08 / 2.10 | 1.04 / 1.07 / 2.11 | 1.02 / 1.10 / 2.12 | 1.00 / 1.29 / 2.29 | 1.04 / 1.07 / 2.11 | 1.48 / 1.37 / 2.85 |
| 4 | 1.00 / 0.81 / 1.81 | 1.02 / 0.82 / 1.84 | 1.02 / 0.83 / 1.85 | 1.00 / 0.99 / 1.99 | 1.02 / 0.82 / 1.84 | 1.33 / 1.00 / 2.33 |
| Lobster - 1000 nodes | | | | | | |
| 20 | 1.00 / 1.07 / 2.07 | 1.00 / 1.07 / 2.07 | 1.00 / 1.07 / 2.07 | 1.00 / 1.78 / 2.78 | 1.00 / 1.07 / 2.07 | 2.42 / 1.88 / 4.30 |
| 8 | 1.00 / 0.68 / 1.68 | 1.00 / 0.68 / 1.68 | 1.00 / 0.68 / 1.68 | 1.00 / 1.56 / 2.56 | 1.00 / 0.68 / 1.68 | 1.72 / 1.14 / 2.86 |
| 4 | 1.00 / 0.37 / 1.37 | 1.00 / 0.37 / 1.37 | 1.00 / 0.37 / 1.37 | 1.00 / 0.92 / 1.92 | 1.00 / 0.37 / 1.37 | 1.26 / 0.47 / 1.73 |
| Random Euclidean - 1000 nodes | | | | | | |
| 20 | 1.02 / 1.54 / 2.56 | 1.08 / 1.71 / 2.78 | 1.06 / 1.77 / 2.82 | 1.00 / 3.11 / 4.11 | 1.08 / 1.71 / 2.78 | 1.25 / 1.68 / 2.93 |
| 8 | 1.02 / 0.84 / 1.86 | 1.00 / 0.97 / 1.97 | 1.01 / 0.96 / 1.97 | 1.00 / 1.55 / 2.55 | 1.00 / 0.97 / 1.97 | 1.07 / 0.79 / 1.86 |
| 4 | 1.00 / 0.38 / 1.38 | 1.00 / 0.42 / 1.42 | 1.00 / 0.40 / 1.40 | 1.00 / 0.59 / 1.59 | 1.00 / 0.42 / 1.42 | 1.02 / 0.37 / 1.39 |
| BarabasiAlbert - 1000 nodes | | | | | | |
| 20 | 1.02 / 1.08 / 2.10 | 1.04 / 1.07 / 2.11 | 1.02 / 1.10 / 2.12 | 1.00 / 1.29 / 2.29 | 1.04 / 1.07 / 2.11 | 1.48 / 1.37 / 2.85 |
| 8 | 1.00 / 0.70 / 1.70 | 1.02 / 0.72 / 1.74 | 1.02 / 0.73 / 1.75 | 1.00 / 0.88 / 1.88 | 1.02 / 0.72 / 1.74 | 1.28 / 0.89 / 2.18 |
| 4 | 1.00 / 0.37 / 1.37 | 1.00 / 0.37 / 1.37 | 1.00 / 0.37 / 1.37 | 1.00 / 0.43 / 1.43 | 1.00 / 0.37 / 1.37 | 1.13 / 0.42 / 1.55 |

*Note:* The results are represented as generic network distance units and the three values separated by slashes correspond respectively to: [1]Closest Neighbor Distance / [2]Intra Distance / [3]Total.

of the fog colony should be also considered. Fog colonies with less devices will migrate services to neighbor colonies with higher probability and consequently the communication between services will be increased. The application makespan will result in an increase of the time corresponding to the closest neighbor distances.

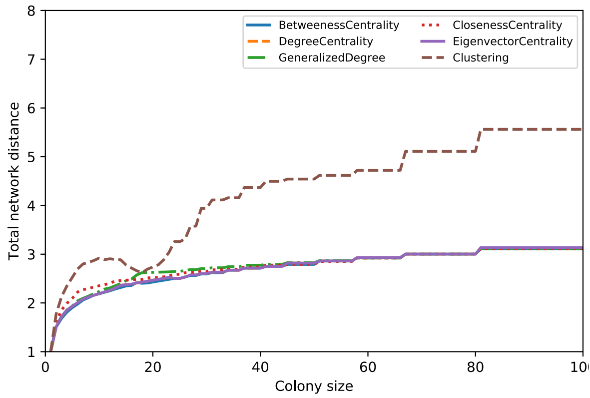
By the analysis of Fig. 5, it is observed that the network distance is stabilized for colony sizes higher than 10 for the Barabasi-Albert topology and higher than 40 for the other topologies, all of them for the Betweenness centrality. Therefore, we can conclude that once that the colony size is higher than those sizes we can increase it without damaging the network distance.

On the contrary, the improvement of the network distance is higher for small colony sizes. This is explained by the reduction of the intra colony distance as it can be observed in TABLE II. The first value of each column corresponds to the closest neighbor distance, the second one to the intra distance, and the third one to the summation of both of them. It is observed that the closest neighbor distance is more stable between colony sizes than the intra distance. The differences between sizes are then mainly generated by the intra distance. As the number of devices in the colonies is increased, further devices are included in the colony and the average distance in relation to the controller device is increased. On the other hand, distances between controllers are not increased with the colony size, as those controllers correspond to central and well-connected nodes, and the distances between them are smaller than the distances between controllers and subordinated devices.

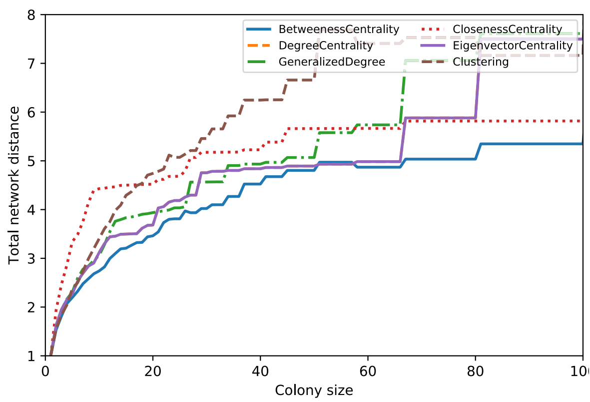
A second research question was related with exploring if any of the centrality indices offered better results for the network distance. We observed in Fig. 3 and Fig. 4 that the indices had a significant influence in the network distance,with important differences between them for all the cases. Moreover, Betweenness centrality resulted in the index with smallest network distances, independently of the number of colonies and the network topology.

Finally, the third research question was if network topology influenced in the results. By the analysis of Fig. 5 we concluded that the results differed for each network topology. The conclusion was that Barabasi-Albert topology obtained the lowest network distances and, on the contrary, Random Euclidean obtained the worst results.

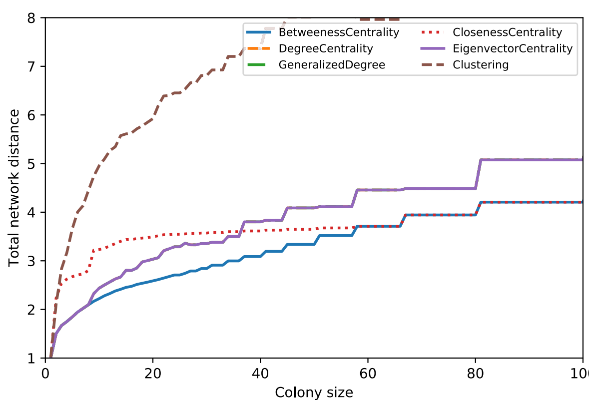
To sum up, Barabasi-Albert was the best topology, and Betweenness the centrality with the best behavior. Additionally,




(a) Barabasi-Albert network topology

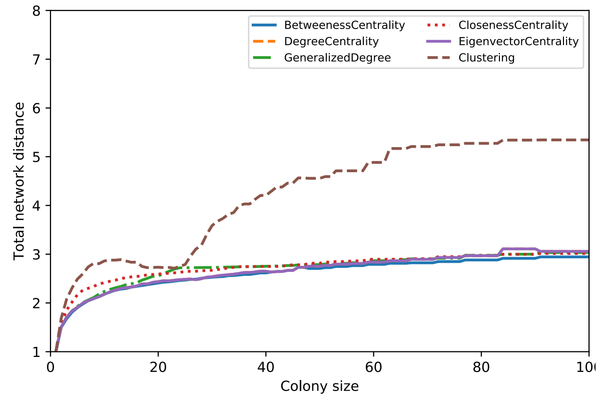



(b) Random Euclidean network topology

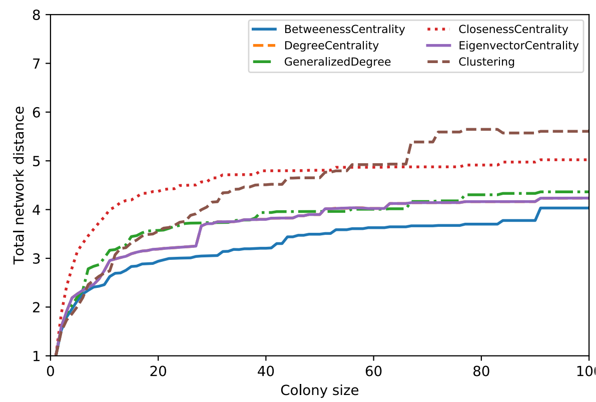



(c) Lobster network topology

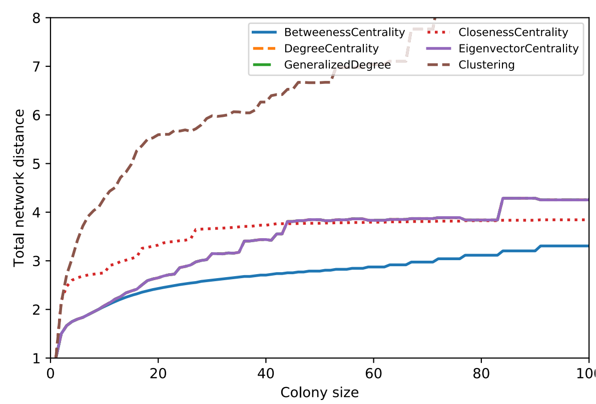
Fig. 3. Comparison of the results between centrality metrics with 400 fog devices.




(a) Barabasi-Albert network topology




(b) Random Euclidean network topology




(c) Lobster network topology

Fig. 4. Comparison of the results between centrality metrics with 1000 fog devices.

the increase of the network distance is negligible for colony sizes of more than 10 devices. The differences in the smallest colonies are due to the intra colony distance, but it is important to highlight that small colonies generate more migrations of application services to neighboring colonies.

## VI. Conclusion

We have presented an exploratory study of the suitability of centrality indices as indicators to be used for the partitioning of fog devices into fog colonies. Our objective was to study how the application makespan is influenced by the organization of the devices and the colonies. We selected the intra colony distance (average distance between the devices in a colony and

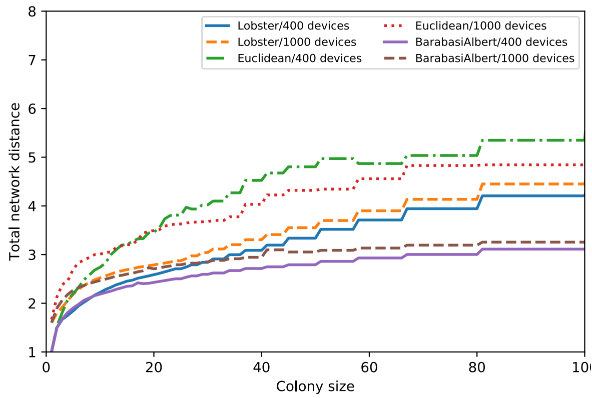



Fig. 5. Comparison of the results considering the colony size between network topologies and architecture size for Betweenness centrality.

the colony controller device) and the closest neighbor distance (distance with the controller of the closest colony) as indicators of the application makespan.

We measured the intra colony and closest neighbor distances for three network topologies (Barabasi-Alber, Lobster, and Random Euclidean), two architecture sizes (400 and 1000 fog devices) and six indices (Betweenness, Degree, Generalized Degree, Closeness, Eigenvector and Clustering).

From the analysis of the results, we concluded that Barabasi-Albert was the topology that obtained smaller network distances, and Betweenness centrality was the best index to determine the controller devices. We also concluded that once that a boundary of colony sizes of 10 devices (or 20 depending on the network topology) is exceeded, the increase of the network distance is negligible. Finally, the important differences in total network distance for small colonies is mostly due to the increase in the intra colony distance. It is important to highlight that although the results showed the lowest distances for the smallest colonies, we can not conclude that the best solution is to create colonies of sizes of 2-3 devices, because the resource capacity of the colony also influences in the application makespan since more application services are migrated to neighboring colonies, increasing the closest neighbor distance of the application.

As future works, the inclusion of additional indicators to also consider the resource capacity of the colonies are required. Additionally, fog service placement optimization policies and colony partitioning strategies could be combined to study if higher improvements are obtained when they are managed simultaneously. This could also open new research opportunities by also defining dynamic colony partitioning framework as the workload and the applications in the system evolve.


## Acknowledgment

This research was supported by the Spanish Government (Agencia Estatal de Investigación) and the European Commission (Fondo Europeo de Desarrollo Regional) through grant number TIN2017-88547-P (MINECO/AEI/FEDER, UE).


## References


[1] R. Mahmud, R. Kotagiri, and R. Buyya, *Fog Computing: A Taxonomy, Survey and Future Directions*. Singapore: Springer Singapore, 2018, pp. 103–130. [Online]. Available: https://doi.org/10.1007/978-981-10-5861-5_5

[2] C. Mouradian, D. Naboulsi, S. Yangui, R. H. Glitho, M. J. Morrow, and P. A. Polakos, "A comprehensive survey on fog computing: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. PP, no. 99, pp. 1–1, 2017.

[3] O. Skarlat, M. Nardelli, S. Schulte, M. Borkowski, and P. Leitner, "Optimized iot service placement in the fog," *Service Oriented Computing and Applications*, Oct 2017. [Online]. Available: https://doi.org/10.1007/s11761-017-0219-8

[4] S. Boccaletti, V. Latora, Y. Moreno, M. Chavez, and D.-U. Hwang, "Complex networks: Structure and dynamics," *Physics Reports*, vol. 424, no. 4, pp. 175 – 308, 2006. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S037015730500462X

[5] Z. Huang, K.-J. Lin, S.-Y. Yu, and J. Y. jen Hsu, "Co-locating services in iot systems to minimize the communication energy cost," *Journal of Innovation in Digital Ecosystems*, vol. 1, no. 1, pp. 47 – 57, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S2352664515000061

[6] V. B. C. Souza, W. Ramrez, X. Masip-Bruin, E. Marn-Tordera, G. Ren, and G. Tashakor, "Handling service allocation in combined fog-cloud scenarios," in *2016 IEEE International Conference on Communications (ICC)*, May 2016, pp. 1–5.

[7] O. Skarlat, M. Nardelli, S. Schulte, and S. Dustdar, "Towards qos-aware fog service placement," in *2017 IEEE 1st International Conference on Fog and Edge Computing (ICFEC)*, May 2017, pp. 89–96.

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

[9] L. Yang, J. Cao, G. Liang, and X. Han, "Cost aware service placement and load dispatching in mobile cloud systems," *IEEE Transactions on Computers*, vol. 65, no. 5, pp. 1440–1452, May 2016.

[10] G. Colistra, V. Pilloni, and L. Atzori, "The problem of task allocation in the internet of things and the consensus-based approach," *Computer Networks*, vol. 73, no. Supplement C, pp. 98 – 111, 2014. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1389128614002655

[11] R. Deng, R. Lu, C. Lai, and T. H. Luan, "Towards power consumption-delay tradeoff by workload allocation in cloud-fog computing," in *2015 IEEE International Conference on Communications (ICC)*, June 2015, pp. 3909–3914.

[12] L. F. Bittencourt, J. Diaz-Montes, R. Buyya, O. F. Rana, and M. Parashar, "Mobility-aware application scheduling in fog computing," *IEEE Cloud Computing*, vol. 4, no. 2, pp. 26–35, March 2017.

[13] Y. Elkhatib, B. Porter, H. B. Ribeiro, M. F. Zhani, J. Qadir, and E. Rivire, "On using micro-clouds to deliver the fog," *IEEE Internet Computing*, vol. 21, no. 2, pp. 8–15, Mar 2017.

[14] F. Bonomi, R. Milito, P. Natarajan, and J. Zhu, *Fog Computing: A Platform for Internet of Things and Analytics*. Cham: Springer International Publishing, 2014, pp. 169–186.

[15] A. Yousefpour, G. Ishigaki, R. Gour, and J. P. Jue, "On reducing iot service delay via fog offloading," *IEEE Internet of Things Journal*, vol. PP, no. 99, pp. 1–1, 2018.

[16] I. Lera, C. Guerrero, and C. Juiz, "Comparing centrality indices for network usage optimization of data placement policies in fog devices," in *2018 Third International Conference on Fog and Mobile Edge Computing (FMEC)*, May 2018.

[17] N. K. Giang, M. Blackstock, R. Lea, and V. C. M. Leung, "Developing iot applications in the fog: A distributed dataflow approach," in *2015 5th International Conference on the Internet of Things (IOT)*, Oct 2015, pp. 155–162.

[18] D. Koschutzki, K. A. Lehmann, L. Peeters, S. Richter, D. Tenfelde-Podehl, and O. Zlotowski, *Centrality Indices*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2005, pp. 16–61.

[19] M. Dehmer, *Structural Analysis of Complex Networks*. Birkhäuser Boston, 2010. [Online]. Available: https://doi.org/10.1007/978-0-8176-4789-6

[20] A. A. Hagberg, D. A. Schult, and P. J. Swart, "Exploring network structure, dynamics, and function using NetworkX," in *Proceedings of the 7th Python in Science Conference (SciPy2008)*, Pasadena, CA USA, Aug. 2008, pp. 11–15.