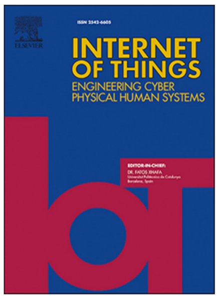
Journal Pre-proof



Intelligent context-aware fog node discovery

Afnan Abdulrahman Bukhari, Farookh Khadeer Hussain, Omar Khadeer Hussain

PII: S2542-6605(22)00089-0
DOI: https://doi.org/10.1016/j.iot.2022.100607
Reference: IOT 100607

To appear in: *Internet of Things*

Received date : 11 July 2022
Revised date : 16 August 2022
Accepted date : 24 August 2022

Please cite this article as: A.A. Bukhari, F.K. Hussain and O.K. Hussain, Intelligent context-aware fog node discovery, *Internet of Things* (2022), doi: https://doi.org/10.1016/j.iot.2022.100607.

Intelligent context-aware fog node discovery

Abstract

Fog computing has been proposed as a mechanism to address certain issues in cloud computing such as latency, storage, network bandwidth, etc. Fog computing brings the processing, storage, and networking to the edge of the network near the edge devices, which we called fog consumers. This decreases latency, network bandwidth, and response time. Discovering the most relevant fog node, the nearest one to the fog consumers, is a critical challenge that is yet to be addressed by the research. In this study, we present the Intelligent and Distributed Fog node Discovery mechanism (IDFD) which is an intelligent approach to enable fog consumers to discover appropriate fog nodes in a context-aware manner. The proposed approach is based on the distributed fog registries between fog consumers and fog nodes that can facilitate the discovery process of fog nodes. In this study, the KNN, K-d tree, and brute force algorithms are used to discover fog nodes based on the context-aware criteria of fog nodes and fog consumers. The proposed framework is simulated using OMNET++, and the performance of the proposed algorithms is compared based on performance metrics and execution time. The accuracy and execution time are the major points of consideration in the selection of an optimal fog search algorithm. The experiment results show that the KNN and K-d tree algorithms achieve the same accuracy results of 95 %. However, the K-d tree method takes less time to find the nearest fog nodes than KNN and brute force. Thus, the K-d tree is selected as the fog search algorithm in the IDFD to discover the nearest fog nodes very efficiently and quickly.

Keywords: Fog node, Discovery, Context-aware, Intelligent, Fog node discovery

1. Introduction

Edge devices are resource and power constrained as they have minimal capabilities for computing and storing their sensed and generated data. Edge devices need to offload their tasks to cloud services for processing and storage. As a result, a vast number of cloud services are provided to support these edge devices. However, cloud services suffer from bottleneck problems and latency [1]. For example, processing a large amount of data in the cloud, which is located far from edge devices, will increase the network bandwidth and response time significantly. To overcome these issues, a new paradigm known as fog computing has emerged. Merging fog computing with cloud computing will help these resource- and power-constrained devices process their tasks using fog nodes. Fog computing brings cloud services include computing, storage and networking to the edge of the network close to the fog consumer's vicinity, thereby decreasing latency and the response time significantly [2] [3]. Data processing will take place in the fog nodes in between the centralized cloud and edge devices. The fog computing paradigm is in its infancy stage which presents several challenges that need more investigation and solutions [4]. Discovering relevant fog nodes from the available ones that satisfy the fog consumer's requirements is a critical problem and it is an important stage of the fog node. To ensure that the tasks from the fog consumers are processed in a timely manner, one of the crucial aspects to consider for fog node discovery is the geographic distance between the fog node and the fog consumers as this directly impacts latency, response time, and bandwidth usage [5]. As we demonstrated in [6], context awareness in fog computing assists in the correct use of the requirements of the user that will assist in providing information on the node that has the best ability to provide these requirements [7]. Thus, it should be an important criterion to consider in fog node discovery.

In this paper, we propose an intelligent system to help fog consumers discover fog nodes based on the context-aware information of fog nodes and fog consumers. Fog node discovery is a crucial problem which needs further investigation by researchers.

The structure of this paper is as follows: Section 2 summarizes the existing approaches that have been proposed for fog node discovery. Section 3 presents the architecture of the proposed solution - IDFD. Section 4 details the performance evaluation of the proposed solution - IDFD and the simulation. Section 5 discusses in detail the experiment results. Finally, Section 6 concludes the paper.

2. Related works

The discovery of services has been well-studied in the areas of cloud computing [8], web services [9] and the IoT environment [10]. However, the discovery of services in fog computing is still in its infancy and none of the existing literature considers the discovery process in a context-aware manner. Most of the existing research considers that fog nodes are located close to the fog consumer and assumes that fog consumers have already discovered the fog nodes and are thus focusing on the next phase of their operation, i.e., processing the communication between the devices. They fail to consider the context-aware mechanism characteristics through which they should be discovered. A few studies focus on the actual discovery of fog nodes and the approach they utilize to achieve this is intensively discussed in [6]

Some existing approaches have been proposed fog node discovery based on an 802.11 Wi-Fi beacon to find appropriate fog nodes. Rejiba et al. [11] [12] proposed the discovery approach which focuses on reducing the energy consumption of edge devices based on the discovery of a wireless network signal range using the 802.11 beacon technique. Rejiba et al. presented the F2C (fog-to-cloud) system to discover fog nodes. The proposed approach embeds the 802.11 beacon technique to discover fog nodes that are close to the vicinity of the edge devices using Wi-Fi technology. While the proposed approaches detect and connect to the fog nodes using Wi-Fi and the 802.11 beacon technique, they fail to consider location-awareness and context-awareness requirements, such as but not limited to user location, user identity, etc. in the discovery process.

Venanzi et al. [13] [14] proposed a Bluetooth-based MQTT-driven node discovery solution in an IoT-fog environment. The proposed approach is termed power efficient node discovery and aims to ensure sustainability, energy efficiency, discoverability, and reliability while engaging in service discovery. The goal of this approach is to reduce the energy consumed by the edge device during the discovery process and to ensure IoT devices are 100% discoverable. Although the proposed approaches using the MQTT technique to find IoT devices are applied on a real-world platform, they do not focus on the use of intelligent algorithms to consider the context-aware, location-aware and energy-constraint requirements of fog nodes.

Gedeon et al. [15] proposed an approach called sunstone for the joint discovery and orchestration of fog nodes (fog computing resources). Sunstone combines three discovery mechanisms, namely (i) the snooping of traceroute packets, (ii) DNS NAPTR record information, and (iii) BGP community string advertisement. The actual discovery work takes place at the cloud side in a Kubernetes environment. The proposed approach is evaluated in a real-world testbed. While the results show that this approach helps to reduce end-to-end latencies, the authors do not use intelligent algorithms for a smart fog nodes search and also, they do not take into account the context-aware and location-aware characteristics of fog nodes for discovery.

Soo et al. [16] proposed a framework for fog node discovery based on the proximity of a mobile ad hoc social network (MASN) in a decentralized peer-to-peer manner. The mobile edge devices are connected to each other through a MASN and they monitor and gather information from MASN peers in the distributed hash table. When the edge device moves, the application from MASN gathers information about the fog nodes that are close to them and uses this information to access them. The GPS locations of the edge devices are used to find the nearest fog nodes. To reduce latency during the communication, the authors proposed using fog nodes that are within the location proximity. While researchers have proposed different methods for fog node discovery, they fail to consider the context-aware mechanism characteristic through which fog nodes should be discovered. They do not consider an intelligent mechanism to carry out the discovery that incorporates other specifics such as context-aware requirements. Based on the above literature review, we identify that the primary gap in the existing literature is the lack of mechanisms for discovering fog nodes in a context-aware manner. To solve this challenge, in this paper we propose the Intelligent and Distributed Fog node Discovery mechanism (IDFD) for storing information on fog nodes, searching and discovering optimal fog nodes. Given the complexity of the approach, in this paper we limit ourselves to single criterion-based approaches to compute the 'proximity' between a fog consumer and fog node.

3. Proposed solution - IDFD

In this research, we propose an intelligent framework on top of fog computing for context-aware fog node discovery which we called IDFD. IDFD is a framework which contains a novel concept of the fog registries consortium (FRC). We regard the FRC as an intermediary between the fog consumer (edge node) and fog nodes that acts as a broker for various activities such as, but not limited to, fog discovery, fog selection, etc.

3.1. Fog Registries Consortium (FRC)

In this research, we propose and use the notion of FRC for fog node discovery. In the future, the role of FRC can be extended beyond this. In this research, we define the notion of FRC which is a finite collection of fog registries (FR) that maintain information on fog nodes. Conceptually, FRC may be defined as follows: FRC = {FR1, FR2, FR3, FR4, … FRn}. The FRC is responsible for the following activities:

- managing the membership of the FRC;
- ensuring the integrity of the FRC process;

- ensuring concurrency in the FRC by synchronizing data between distributed fog registries;
- distributing the distributed fog registries across the globe in a uniform manner;
- controlling the connection between the fog consumer and the distributed registry in their geographical limitation.

In our proposed framework shown in Figure 1, the FRC consists of two types of fog registries as follows:

- **The main central fog registry (CFR):** The CFR controls and coordinates all the distributed fog registries in the consortium, ensures an equal and uniform spread of distributed fog registries, manages the membership of the distributed fog registry in the FRC, ensures the integrity of the FRC, and ensures concurrency by synchronizing fog node data between all other fog registries to keep them up to date.
- **The distributed fog registries (DFR):** In contrast to the main CFR, there are multiple geographically dispersed registries that synchronize with the main CFR. The DFRs are located in different remote regions and are uniformly spread across the globe. Each DFR stores information on all fog nodes in the network. All DFRs have the same data on all fog nodes and are synchronized. The DFRs are the first point of contact for fog consumers. The DFR has the following modules:
  - Fog Node Discovery Engine (FNDE)
  - Fog repository.

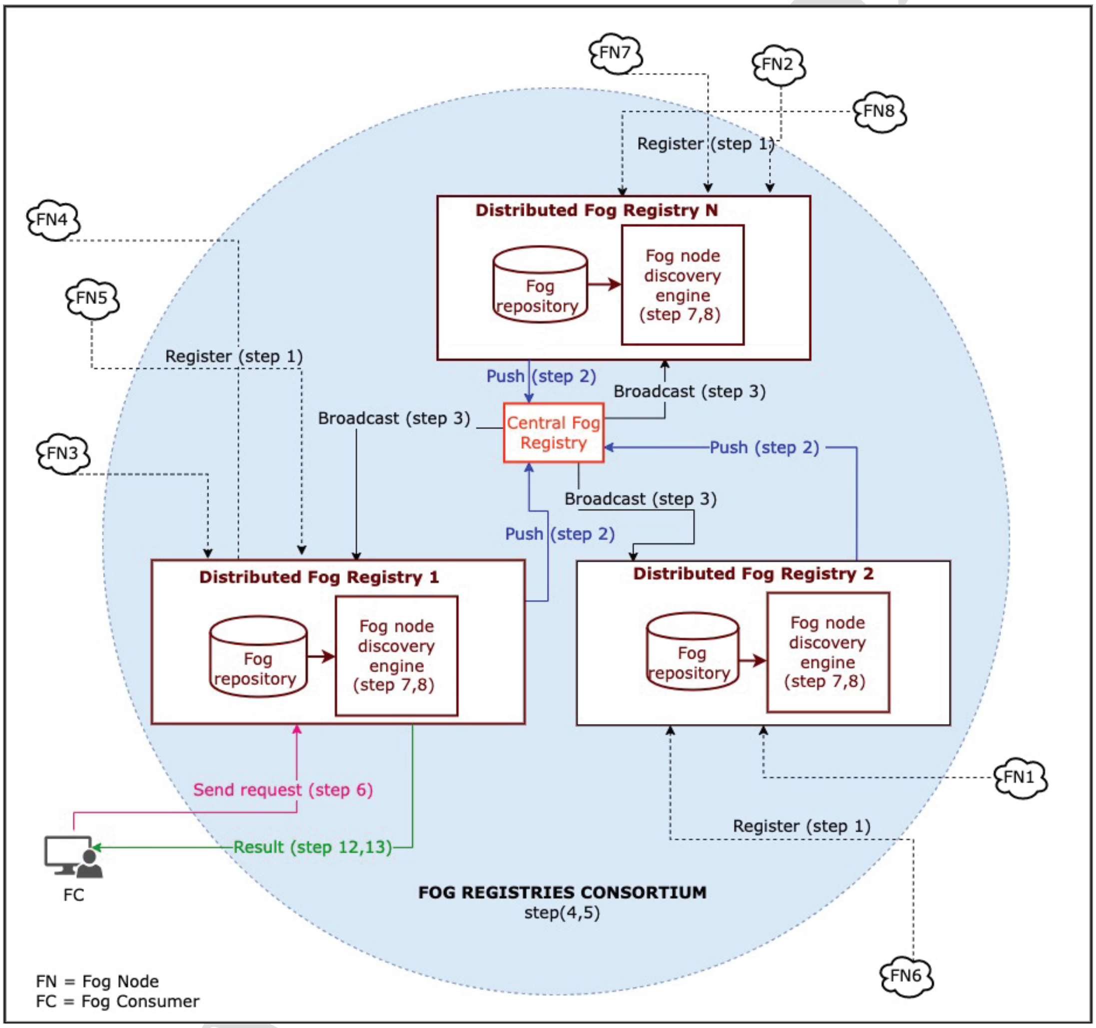


Figure 1: Conceptual overview of the proposed solution – IDFD

Membership of the FRC: The membership of the fog registries within the FRC could either be static or dynamic. In a fog environment, the membership of the FRC is transient and dynamic. However, the membership of the FRC could also be static with proper checks and balances in place to ensure the integrity of the process activity carried out by the FRC. In our research, we use a static membership function for the FRC. In the following, we describe how fog nodes can become members of the FRC and also how the integrity of the activity carried out by the FRC can be maintained (particularly under static membership).The job of the FRC is to store an exhaustive list of all the fog nodes in the network with their bespoke attributes such as but not limited to fog node name, fog node location etc. It is critical to note that this list of fog nodes stored in the FRC is not static but an evolving list where new fog nodes may join the fog network and existing ones drop out of the fog network. The membership criteria for the FRC are grounded on the factors that enable the fog registry (FR) to maintain and update the list of the fog nodes in a dynamic manner. As such, we use the following membership functions to join our FRC:

- Storage Capacity: The FR should have a very large storage space to maintain the extensive and expansive list of fog nodes with their bespoke attributes to minimize retrieval time.
- Processing capacity: The FR should have a fast processing capacity to minimize retrieval time.
- Reputation Value: The FR should be a trustworthy fog node as evidenced by its reputation value.

  Whilst in this research, we propose the above three factors for selecting the FRs, in a practical implementation, these factors can be different. The CFR initiates the selection process. It is critical to note that while the CFR initiates the process, the activities within the selection process are carried out in a manner that is visible to all the participants.

Ensuring the integrity of the FRC process: To ensure that no single entity or group of entities within the FRC has control of the entire activities in the FRC to the extent that they are able to manipulate its activities in an unfair manner, we propose the notion of checkpointing and benchmarking. Checkpointing is carried out at regular periodically recurring intervals of time (n). The working of checkpointing and benchmarking is as follows:

- During the checkpointing process, each member of the FRC is asked to submit a one-way hash of its repository, using a dynamically generated key. The CFR generates this key dynamically and every member of the FRC is required to submit a one-way hash outcome to the FRC which is shared with all members of the consortium.

- During the benchmarking process, led by the CFR, the hash outcomes of all the 'N' FRs are compared to identify whether there are any erroneous FRs. These erroneous FRs, whose hash outcomes do not match those of the majority ones, are asked to rectify the content of their respective registries so that there is no discrepancy in the one-way hash outcomes. If the CFR finds a FR whose one-way hash outcome repeatedly does not match those of the majority, then that fog node may be expelled from the FRC.

It is critical to note that both the checkpointing and the benchmarking process is carried out in a manner that is visible to all members of the consortium. A consortium-visible blackboard is used to systematically display all the activities of the checkpointing process (such as a call for one-way hash functions and the submission of one-way hash functions) and also all the activities of the benchmarking process (outcome of the comparison process, identification of non-compliant one-way hash functions, rectification of the repositories of non-compliant nodes and expulsion of repeating non-compliant nodes).

Ensuring concurrency between DFRs in the FRC: Each DFR stores information on all the fog nodes in the network. The FRC achieves consistency and integrity of information by ensuring that the information is stored and synchronized in all the DFRs. In this research, we use the push synchronization mechanism to synchronize fog node data between the DFRs in the FRC. It is an event-based process and is carried out when new information is added or when information is updated. The redundant data in the DFRs helps detect any malicious node if data is changed. Also, if one of the DFRs fails or dies, the data can be recovered from another registry.

3.2. Fog Node Discovery Engine (FNDE)

The IDFD framework is encapsulated in the FNDE in the DFR. We propose the FNDE as the means for discovering fog nodes. The FNDE collects the context-aware data of the fog consumer. In the scope of this research, the word 'context' refers to the identity and location of the fog consumer. In the future, other research may choose to enrich context-aware discovery by adding additional context-related parameters. The following steps demonstrate the methodological working of IDFD based on the FRC module and FNDE:

1. **Step 1:** The fog node provider publishes the context fog node data to the DFR when the fog node joins the network. The context fog node data includes the identity and the current physical location of the fog node which

includes fog node name, fog node ID, fog node location and fog node descriptions. The location of the fog node in this research is static and must be determined in its geographical range. Fog nodes may be located at various locations such as shopping malls, schools, hospitals, libraries, train stations etc.

2. **Step 2:** Data is stored in the fog repository of the DFR.
3. **Step 3:** Synchronize the fog node data between the DFRs in the FRC to ensure the concurrency of the information between them. When the data of the fog node is registered (new data) or modified (changed) in one of the DFRs, the DFR will push this new data (or changed data) to the CFR using the push synchronization mechanism. The data source which is the DFR notifies the data sink which is the CFR of new data or changed data. The push is an event-based process which starts when the new data arrive or the data are changed. This event-based push mechanism ensures data consistency between all DFRs.
4. **Step 4:** The CFR broadcasts the new data or the changed data to all the other DFRs. As a result of Step 4, all the DFRs have the same up-to-date data on all fog nodes, which maintains the consistency of data and the concurrency between all the DFRs in the FRC.
5. **Step 5:** When the fog consumer wants to connect to the fog node, the fog consumer will be prompted on their edge device (such as a mobile device) about the availability of the nearest fog node. The discovery of the fog node is based on the location-based context information of the fog consumer and also the fog node provider. The fog consumer initiates the node discovery process by providing their login credentials (usually comprising username and password; however, other authentication mechanisms such as finger printing may also be used). Developing reliable and foolproof authentication measures has been a longstanding research question. In this research, we do not intend to address this question and focus on developing reliable authentication mechanisms. On the other hand, we use proven authentication mechanisms such as the use of authentication service providers coupled with a single sign-on. Fog consumers authenticate their identity using an external third-party authentication service (such as Google or Microsoft). On successful authentication, they are redirected to the DFR in the consortium. The external party communicates the authenticated identity to the central fog node provider.
6. **Step 6:**The FNDE autonomously collects the fog consumer's physical location (latitude and longitude values from the GPS receiver) to carry out fog

node discovery.

7. **Step 7:** The FNDE applies the selected fog search algorithm, namely KNN [17], k-d tree [18], or brute force [19]. These search algorithms are commonly and successfully used in nearest neighbor searching. In our research, we implement four variants of the fog discovery algorithms (with each variant comprising one algorithm). It intelligently matches the nearest fog nodes on context parameters such as identity and location.
8. **Step 8:** The FNDE provides a list of the nearest fog nodes. The fog consumer is presented with these available fog nodes and they can select one of them.

Figure 2 illustrates the working of the proposed solution in details.

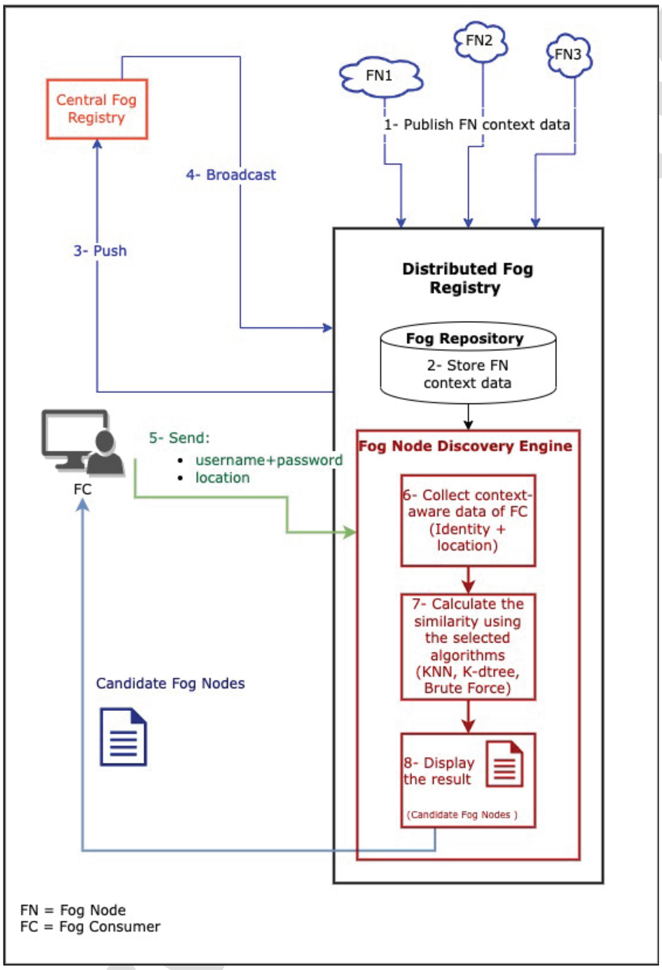


Figure 2: Overview of the fog node discovery workflow

Algorithm 1 summarizes the working of the the proposed solution – IDFD.

Algorithm 1 IDFD

Require: FN_Identity, FN_Location, FC_Identity, FC_Location
Ensure: K fog nodes
1: Registering FN_Identity and FN_Location, in the DFR
2: Store FN_Identity FN_Location in the fog repository of the DFR
3: DFR pushes FN_Identity FN_Location to the CFR by using the push synchronization mechanism
4: The CFR broadcasts the new or changed FN_Identity FN_Location to all the other DFRs
5: The FRC manages the connection between the FC and one of DFRs based on the FC's geographical location.
6: The FC provides his/her FC_Identity login credentials (username and password)
7: If the authentication is successful, then
8: The FNDE in the DFR autonomously collects the FC_Location (latitude and longitude values from the GPS receiver)
9: The FNDE finds the nearest fog nodes to the FC by using one of fog search algorithms.
10: The FNDE provides a list of the K-nearest fog nodes

4. Performance evaluation

To find the optimal nearest fog nodes, the FRC should be implemented first. We implemented the framework of FRC in a simulation environment using the OMNET++ platform. OMNET++ is an open source simulation tool coded in the C++ language [20]. We used OMNET++ version 5.6.2. OMNET++ contains three components which are network, NED language, and configuration file. In the network, we define four modules CFR, DFR, FN, and FC, as shown in Figure 3. . The NED creates the network topology. The configuration file called omnetpp.ini runs the network simulation.

Figure 3: Simulation modules

Firstly, we build a wireless network and build four modules for the FRC. The first two modules are the CFR and DFR which contain the fog repository that has information on all the registered fog nodes which are pushed from DFR. The fog repository schema of CFR and DFR is detailed in Table 1.

Table 1: The schema of the fog repository in CFR and DFR.

FN_ID	FN_name	LATITUDE	LONGITUDE
1000	Fort Hill Wharf DARWIN	-12.471947	130.845073
10000	Cnr Castlereagh & Lethbri PENRITH	-33.756158	150.698182
10000002	Optus 50m Lattice Tower 71 Eastward Road Utakarra	-28.77766	114.63426

The third module is the fog node module which is represented with four parameters, namely identification, name and physical location (latitude, longitude). The network is designed with up to 2000 fog nodes with their physical locations. The simulated latitude and longitude of fog nodes are edge servers' locations from the EUA dataset [21]. Table 2 provides a snapshot of the fog nodes dataset. The fourth module is the fog consumer module which represents each fog consumer in the simulated network with fog consumer ID, fog consumer IP, and physical location (latitude, longitude). We use the end user dataset from the EUA dataset [21] to represent fog consumers in the network. Table 3 shows a snapshot of the fog consumer dataset. These modules communicate via channels to send and receive messages. Secondly, the FNDE is implemented and written in the C++ language in the DFR module. We implemented the FNDE with four different nearest neighbors search algorithms: KNN with Euclidean distance [17], KNN with Manhattan distance [17], k-d tree [18], and the brute force algorithm with haversine distance [19]. Figure 4 shows the simulated network with four modules.

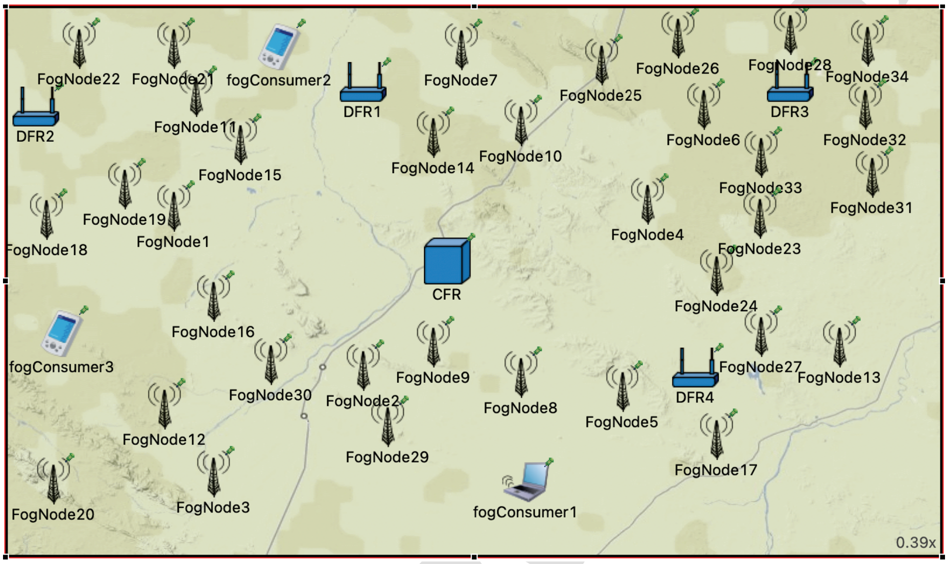


Figure 4: Simulation of the proposed framework

Table 2: A snapshot of the fog nodes dataset

SITE_ID	Name	LATITUDE	LONGITUDE
1000	Fort Hill Wharf DARWIN	-12.471947	130.845073
10000	Cnr Castlereagh and Lethbri PENRITH	-33.756158	150.698182
10000002	Optus 50m Lattice Tower 71 Eastward Road Utakarra	-28.77766	114.63426
10000003	6 Knuckey Street Darwin	-12.464597	130.840708
10000004	Cape Wickham Links Clubhouse KING ISLAND	-39.5964	143.9339

Table 3: A snapshot of the fog consumer dataset

FC_ID	IP	LATITUDE	LONGITUDE
FC1	1.120.2.1	-37.8833	145.3333
FC2	1.120.0.1	-30.5083	151.6712
FC3	1.120.163.1	-21.0405	149.1849
FC4	1.122.32.1	-31.9344	115.8716
FC5	1.123.11.1	-34.8333	138.6333

We conducted several experiments on the simulated network with a different number of fog nodes and different fog consumers. The experiments include finding the 10-nearest fog nodes for five different fog consumers.We selected five different fog consumers as detailed in Table 3. For each fog consumer, we find the 10-nearest fog nodes using the four proposed methods with a different number of fog nodes. We used 5 iterations. The first iteration has 100 fog nodes. The second iteration has 500 fog nodes. The third iteration has 1000 fog nodes. The fourth iteration has 1500 fog nodes. The fifth iteration has 2000 fog nodes. Then for each fog consumer, the 10-nearest fog nodes will be obtained using:

1. KNN with Euclidean distance
2. KNN with Manhattan distance
3. K-d tree
4. Brute force with haversine distance

Consequently, the first iteration includes 100 fog nodes and 1 fog registry. The results of the four methods to find the 10-nearest fog nodes for each fog consumer are presented in Table 4:

Table 4: The results of the four methods to find the 10-nearest fog services for one fog consumer

Fog consumer ID	Brute Force algorithm based on Haversine Distance	KNN with Euclidean Distance	KNN with Manhattan Distance	K-d tree with Euclidean Distance
FC1 = (-37.4886, 145.3333)	ServiceID = 33, distance = 4.15166	ServiceID = 33, distance = 0.0381813	ServiceID =33, distance = 0.0489	ServiceID = 33, distance = 0.04
	ServiceID = 65, distance = 7.207	ServiceID = 65, distance = 0.0757346	ServiceID = 65, distance = 0.104591	ServiceID = 65 , distance = 0.08
	ServiceID = 67, distance = 20.1157	ServiceID = 83, distance = 0.196166	ServiceID = 83, distance = 0.27459	ServiceID = 83 , distance = 0.20
	ServiceID = 83, distance = 20.2772	ServiceID = 84, distance = 0.202355	ServiceID = 84, distance = 0.28559	ServiceID = 84 , distance = 0.20
	ServiceID = 84, distance = 20.5597	ServiceID = 67, distance = 0.220727	ServiceID = 67, distance = 0.285641	ServiceID = 67 , distance = 0.22
	ServiceID = 31, distance = 26.1495	ServiceID = 31, distance = 0.296903	ServiceID = 31, distance = 0.328253	ServiceID = 31 , distance = 0.30
	ServiceID = 25, distance = 32.4289	ServiceID = 34, distance = 0.363791	ServiceID = 25, distance = 0.427566	ServiceID = 34 , distance = 0.36
	ServiceID = 96, distance = 32.8004	ServiceID = 25, distance = 0.365472	ServiceID = 90, distance = 0.4283	ServiceID = 25 , distance = 0.37
	ServiceID = 90, distance = 32.9477	ServiceID = 96, distance = 0.369949	ServiceID = 96, distance = 0.430675	ServiceID = 96 , distance = 0.37
	ServiceID = 34, distance = 34.5623	ServiceID = 90, distance = 0.372249	ServiceID = 30, distance = 0.4686	ServiceID = 90 , distance = 0.37

When a fog node is added to the network, it communicates with the nearest DFR and sends a *cMessage* which includes the fog node's context date. Then, when the fog consumer establishes the session to discover the nearest fog nodes, the latitude and longitude of the fog consumer's current location is sent to the DFR as a *cMessage*. The FNDE in the DFR will find the nearest fog nodes using the search algorithm and then sends the list to the fog consumer. We undertook the same processing for all iterations of 100, 500, 1000, 1500, and 2000 fog nodes for each fog consumer. Figures 5, 6, 7, 8, and 9 show the simulation of 100, 500, 1000, 1500, 2000 fog nodes respectively.

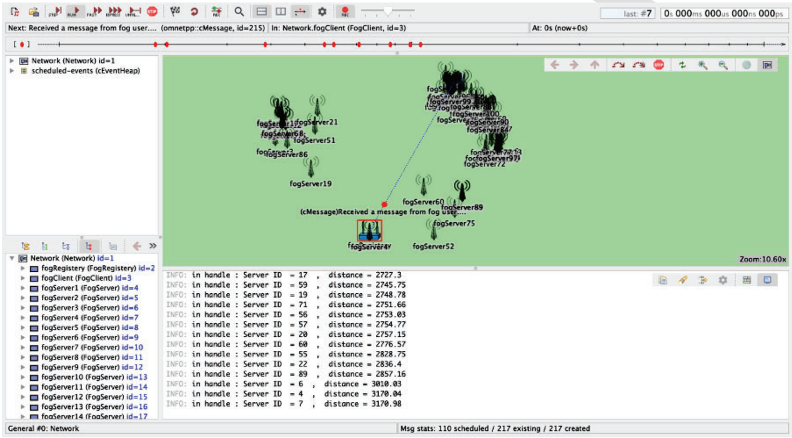


Figure 5: Simulating 100 fog nodes using OMNET++

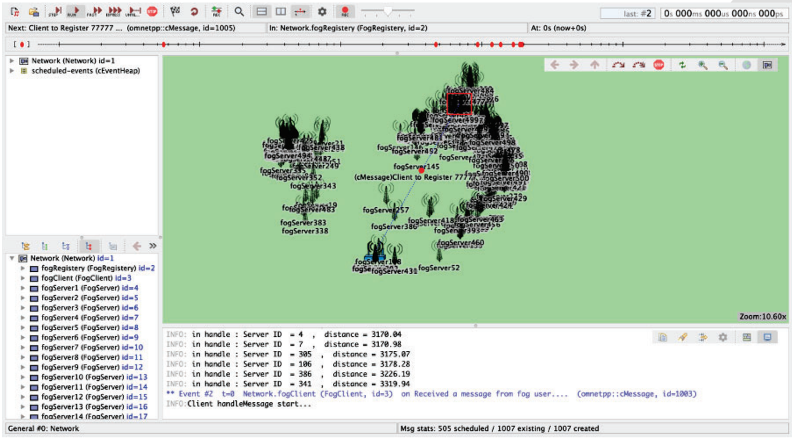


Figure 6: Simulating 500 fog nodes using OMNET++

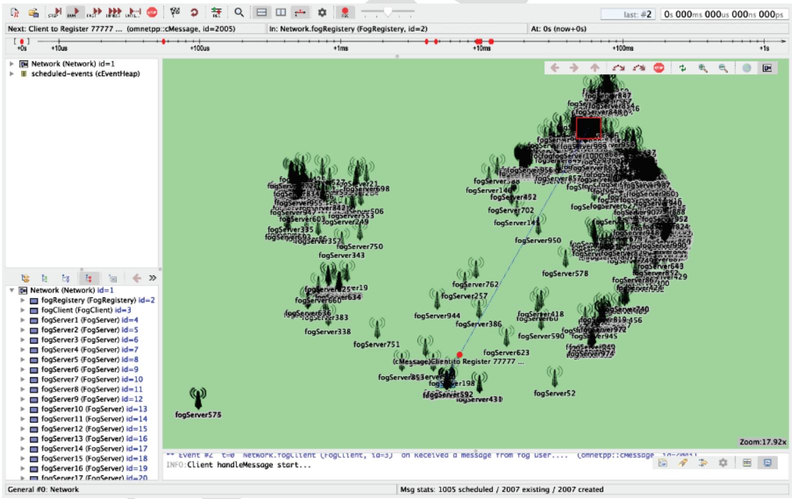


Figure 7: Simulating 1000 fog nodes using OMNET++

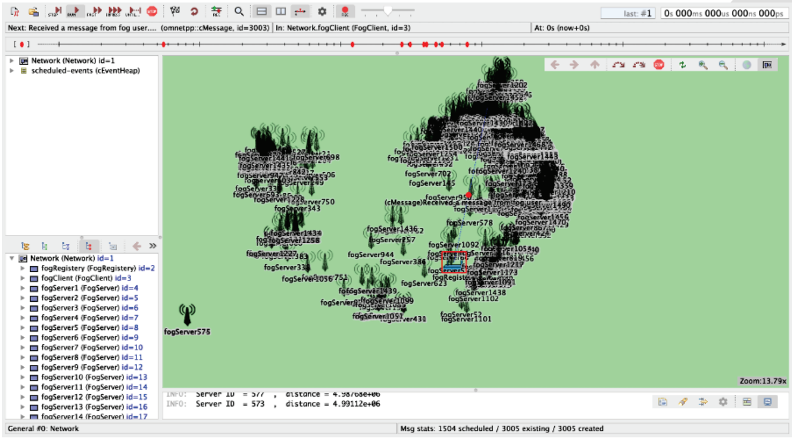


Figure 8: Simulating 1500 fog nodes using OMNET++

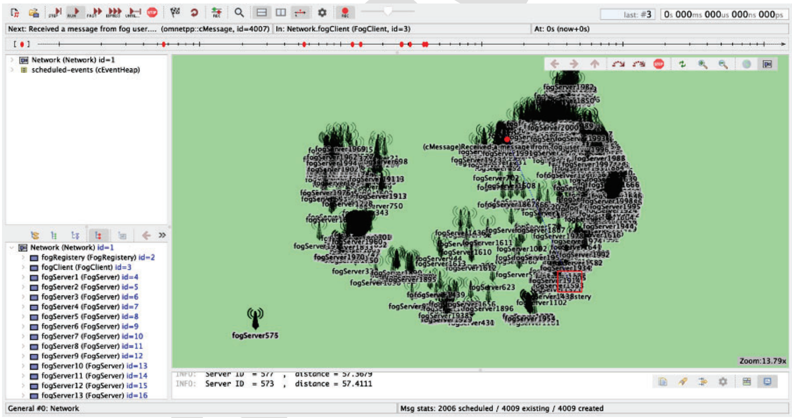


Figure 9: Simulating 2000 fog services using OMNET++

We evaluated the results of the nearest neighbours' fog nodes by comparing the four methods. We validated the results by calculating the evaluation metrices of the four methods using well-known and accepted metrices, namely precision, recall, F1 score and accuracy to obtain the optimal method of finding the accurate nearest neighbours fog nodes.

4.1. The validation process is as follows:

1. **Step 1:** Initialization process: The initialization process comprises the following steps:
   (a) Specify the number of fog nodes, starting with 100 fog nodes. The number of fog nodes in each iteration is increased by 500.
   (b) Determine the number of FRs, starting with 10 registries.
   (c) Determine the number of iterations = n times. The value of the parameters for (b) and (c) vary from one iteration to the next. The number of iterations = 5 and the number of fog nodes in each iteration is increased by 500.
      - First iteration: fog nodes =100
      - Second iteration: fog nodes =500
      - Third iteration: fog nodes =1000
      - Fourth iteration: fog nodes =1500
      - Fifth iteration: fog nodes =2000
2. **Step 2:** Implement the four nearest neighbor algorithms (KNNs, K-d tree, brute force) for fog node discovery in the DFR module.
3. **Step 3:** The system administrator selects a random fog consumer A and asks it to carry out context-aware fog node discovery. The system administrator knows the closest context-aware fog nodes for the selected fog consumer A; however, this information is unknown to fog consumer A.
4. **Step 4:** Use well-known and accepted metrices such as precision, recall and F1 score which are defined as:
   $Precision = \frac{TP}{TP+FP}$
   $Recall = \frac{TP}{TP+FN}$
   $F1Score = \frac{2*(Recall*Precision)}{(Recall+Precision)}$
   Then, we compute and compare the accuracy of the four methods.
   $Accuracy = \frac{TP+TN}{TP+FN+TN+FP}$ where a:
   - True Positive (TP): means the discovered fog node is relevant to the actual output.
   - True Negative (TN): means the undiscovered fog node is not relevant to the actual output.
   - False Positive (FP): means the discovered fog node is irrelevant to the actual output.

- False Negative (FN): means the undiscovered fog node is relevant to the actual output.

5. Step 5: Repeat steps 3 and 4 n times.

Figures 10, 11, 12, 13, and 14 illustrate the results of the four methods with a varying number of fog nodes.

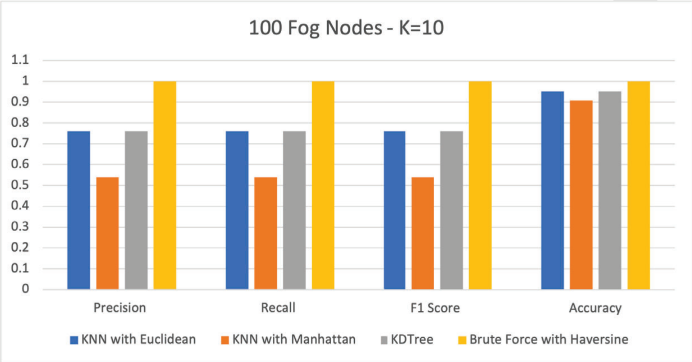


Figure 10: Evaluation results of the network with 100 fog nodes

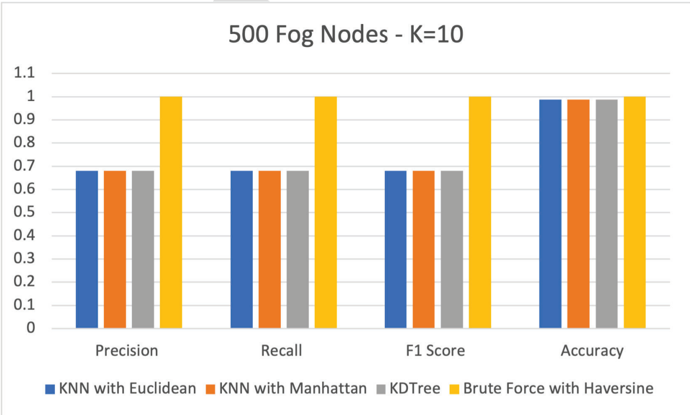


Figure 11: Evaluation results of the network with 500 fog nodes

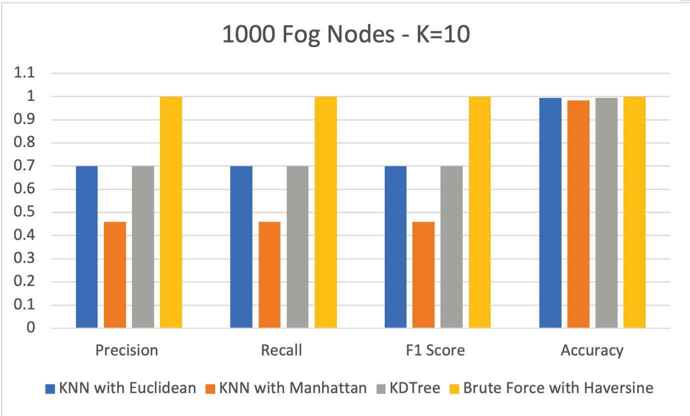


Figure 12: Evaluation results of the network with 1000 fog nodes

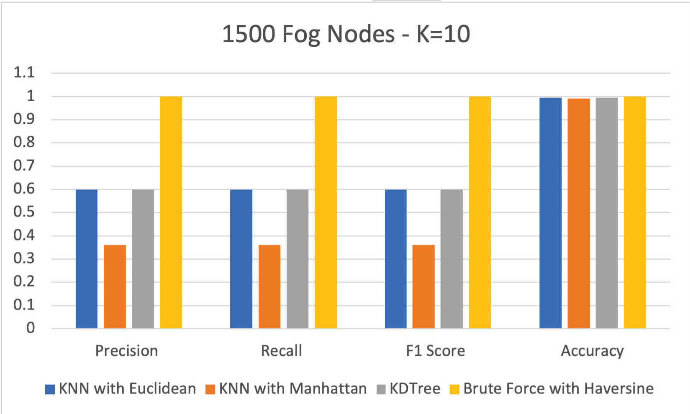


Figure 13: Evaluation results of the network with 1500 fog nodes

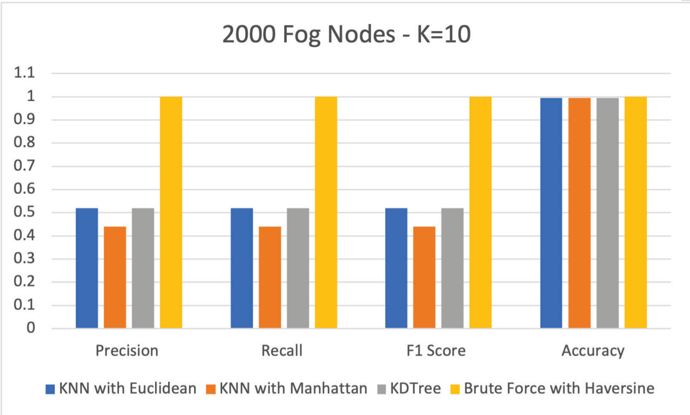


Figure 14: Evaluation results of the network with 2000 fog nodes

The overall accuracy of the four methods is shown in Figure 15.

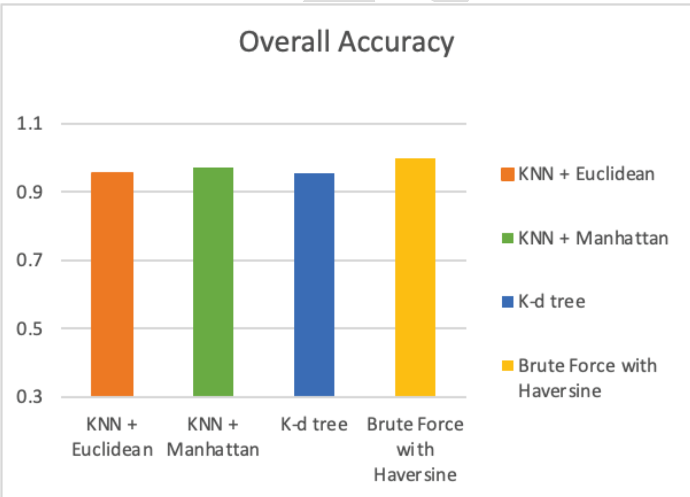


Figure 15: Average evaluation results

5. Discussion

Figures 10 - 15 show that the brute force algorithm with haversine distance has the highest accuracy of all the methods, outperforming K-d tree and KNN and also obtains a value of 1 for precision, recall, and F1 score. However, KNN and K-d tree have equal precision and recall values and their accuracy is also close to 1 which means these two approaches give optimal results as well because both methods use Euclidean distance to find the nearest neighbours. As can be seen from Figures 10 - 15, the brute force with haversine distance outperforms KNN and K-d tree, although the results of all methods are very similar.

A key point of consideration in the selection of an optimal fog search algorithm is the complexity of the algorithm itself. We use algorithm complexity as a key input to its selection. The DFR will carry out node discovery on multiple occasions for consumers. Even a very minor incremental reduction in the complexity of the search algorithm makes a huge difference to the load on the fog node server. The complexity of brute force with haversine distance is O(n^m). However, using this linear search method is not suitable for the approach proposed in this paper because when the data increases (number of fog nodes), the complexity of finding the nearest locations also increases. The complexity of the KNN method is O(n). The downsides are that KNN is very sensitive to the curse of dimensionality and expensive to compute with a O(n) calculation. In contrast, the complexity of the K-d tree method is O(long(n)). K-d tree is guaranteed log2 n depth where n is the number of points in the set.

The execution time of the three methods is calculated to evaluate their complexity. We computed the execution time of the three methods with a different number of fog nodes (100, 500, 1000, 1500, and 2000). We ran the simulation five times then the average of the execution times is obtained. Figures 16 - 20 show the average execution times of the three proposed methods with a different number of fog nodes.

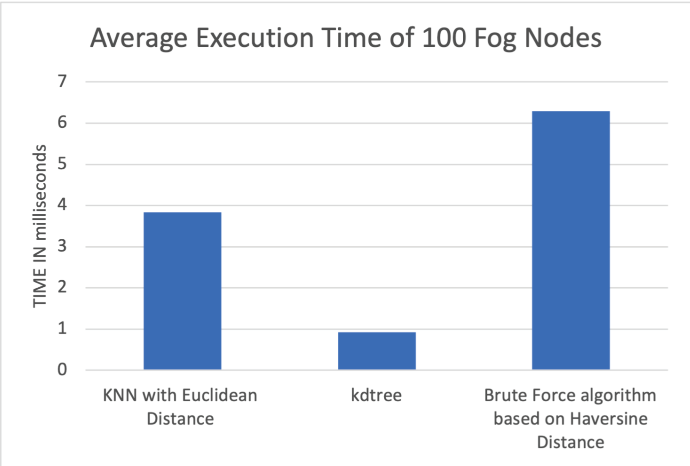


Figure 16: The average execution times of the three methods when fog nodes = 100

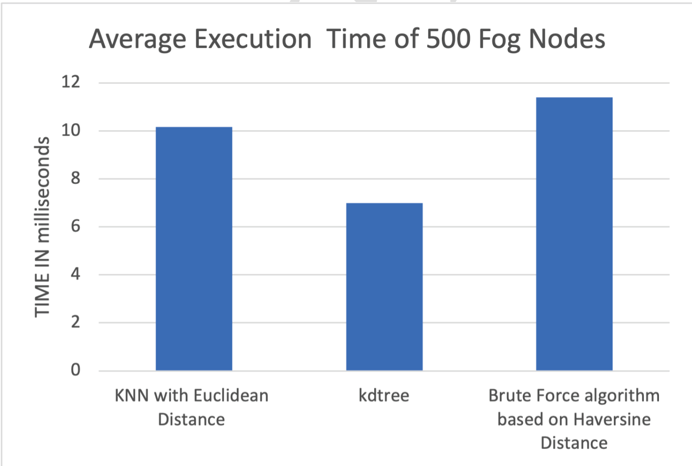


Figure 17: The average execution times of the three methods when fog nodes = 500

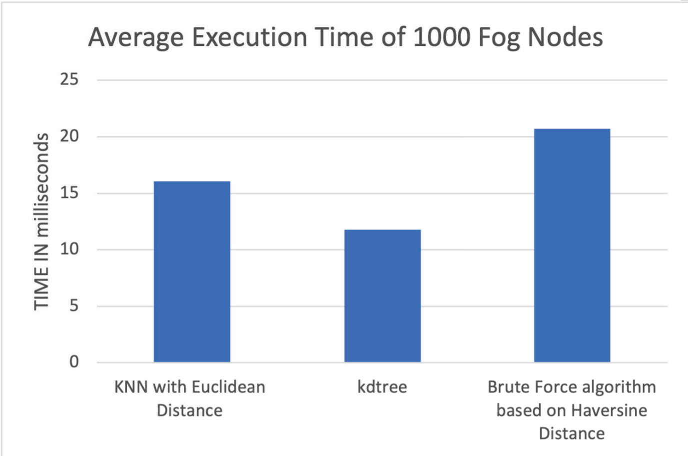


Figure 18: The average execution times of the three methods when fog nodes = 1000

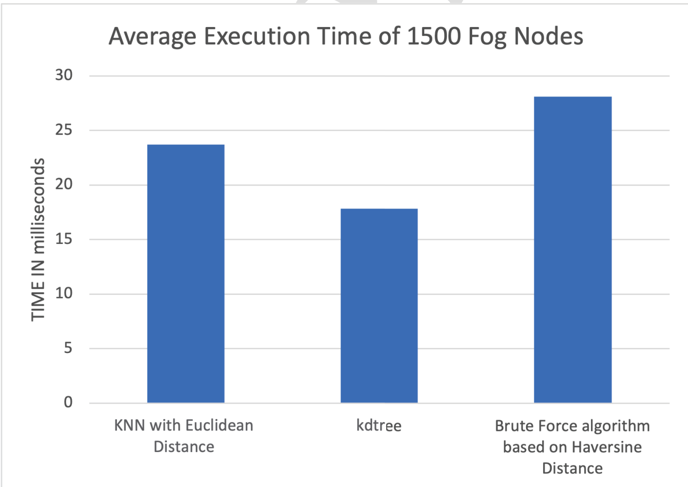


Figure 19: The average execution times of the three methods when fog nodes = 1500

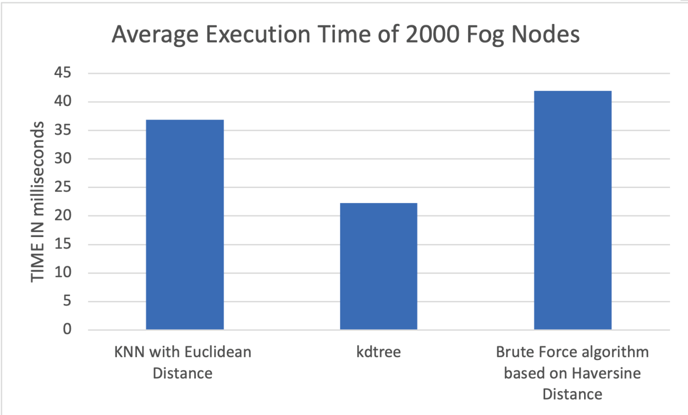


Figure 20: The average execution times of the three methods when fog nodes = 2000

The average of the overall execution times is shown in Figure 21.

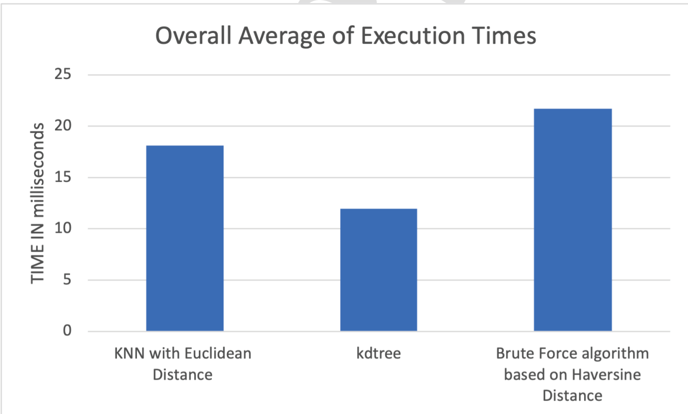


Figure 21: The overall average of the execution times of the three methods

Figures 16 - 21 show that the execution time of K-d tree is less than KNN and brute force with haversine distance. When the number of fog nodes is 100, the execution time of K-d tree is less than 1 millisecond, but the execution time of KNN is more than 3 milliseconds and brute force with haversine distance is more than 6 milliseconds. So, the K-d tree method takes less time to find the nearest fog services than KNN and brute force with haversine distance. Furthermore, the execution time of the three methods increases when the number of fog nodes increases. For example, when the number of fog nodes is 100, the execution time of K-d tree is about 0.92 milliseconds. However, when the number of fog nodes is 1000, the execution time of K-d tree is about 11.8 milliseconds. So, to select the optimal method to use in the FNDE, we took into consideration the incremental reduction in the execution time of the method. In the approach proposed in this paper, a very minor incremental reduction in the execution time of the method makes a huge difference to the load on the fog discovery module at FNDE. For this reason, the K-d tree method performs faster and needs less time to find the nearest fog nodes. Based on the experiment results, it is clear that the K-d tree is very efficient in terms of the proposed approach. Moreover, the K-d tree in FNDE will discover the nearest fog nodes very efficiently and quickly.

6. Conclusions and future work

Fog computing is a promising solution for critical cloud issues such as latency, network bandwidth, storage, etc. Discovering optimal fog nodes helps fog consumers process their services efficiently and quickly. In this paper, we proposed an intelligent mechanism for fog node discovery based on the context-aware data of fog nodes and fog consumers. We proposed IDFD to enable fog consumers to discover appropriate fog nodes in a context-aware manner. The IDFD contains the fog registries consortium (FRC) that has distributed fog registries. The working of the framework is encapsulated in the fog node discovery engine (FNDE) in the distributed fog registry. FNDE must use the optimal fog search algorithm to reduce time and increase accuracy. Four fog search algorithms, namely KNN with Euclidean distance, KNN with Manhattan distance, K-d tree, and brute force with haversine distance were implemented and evaluated in the FNDE using the OMENT++ platform. Several simulation experiments were conducted and the results show that the K-d tree search algorithm improves the overall system performance. The K-d tree algorithm achieves high accuracy result of 95% and requires less time to find the nearest fog nodes than the KNN methods and brute force with haversine distance. The thrust of this research is to build an intelligent single criterion approach for computing the ‘proximity’ between a fog node consumer and provider. Our approach is agnostic of the parameter used to quantify proximity. In our future work, we will propose intelligent multi-criteria-driven node discovery approaches based on diverse parameters that manifest proximity, such as bandwidth, physical distance, latency etc.

References

[1] S. P. Singh, A. Nayyar, R. Kumar, A. Sharma, Fog computing: from architecture to edge computing and big data processing, Journal of Supercomputing 75 (4) (2019) 2070–2105. doi:10.1007/s11227-018-2701-2.
URL https://doi.org/10.1007/s11227-018-2701-2

[2] R. K. Naha, S. Garg, D. Georgakopoulos, P. P. Jayaraman, L. Gao, Y. Xiang, R. Ranjan, Fog computing: Survey of trends, architectures, requirements, and research directions, IEEE Access (2018). doi:10.1109/ACCESS.2018.2866491.

[3] E. M. Tordera, X. Masip-Bruin, J. Garcia-Alminana, A. Jukan, G.-J. Ren, J. Zhu, J. Farre, What is a Fog Node A Tutorial on Current Concepts towards a Common Definition (2016).
URL http://arxiv.org/abs/1611.09193

[4] Q. Wei, Z. Jin, Service discovery for internet of things (2013) 1–6doi:10.1145/2430475.2430500.

[5] N. Mostafa, Cooperative fog communications using a multi-level load balancing, 2019 4th International Conference on Fog and Mobile Edge Computing, FMEC 2019 (2019) 45–51doi:10.1109/FMEC.2019.8795325.

[6] A. Bukhari, F. K. Hussain, O. K. Hussain, Fog node discovery and selection: A Systematic literature review, Future Generation Computer Systems 135 (2022) 114–128. doi:https://doi.org/10.1016/j.future.2022.04.034.
URL https://www.sciencedirect.com/science/article/pii/S0167739X22001625

[7] G. D. Abowd, A. K. Dey, P. J. Brown, N. Davies, M. Smith, P. Steggles, Towards a Better Understanding of Context and Moderators, International Symposium on Handheld and Ubiquitous Computing HUC 1999 (1999) 304–307.

[8] M. M. Al-Sayed, H. A. Hassan, F. A. Omara, An intelligent cloud service discovery framework, Future Generation Computer Systems 106 (2020) 438–466. doi:10.1016/j.future.2019.12.027.
URL https://doi.org/10.1016/j.future.2019.12.027

[9] A. Bukhari, X. Liu, A Web service search engine for large-scale Web service discovery based on the probabilistic topic modeling and clustering, Service Oriented Computing and Applications 12 (2) (2018) 169–182. doi:10.1007/s11761-018-0232-6.
URL https://doi.org/10.1007/s11761-018-0232-6

[10] A. Zaslavsky, P. P. Jayaraman, Discovery in the Internet of Things, Ubiquity 2015 (October) (2015) 1–10. doi:10.1145/2822529.

[11] Z. Rejiba, X. Masip-Bruin, A. Jurnet, E. Marin-Tordera, G. J. Ren, F2C-Aware: Enabling Discovery in Wi-Fi-Powered Fog-to-Cloud (F2C) Systems, in: Proceedings - 6th IEEE International Conference on Mobile Cloud Computing, Services, and Engineering, MobileCloud 2018, Vol. 2018-Janua, 2018, pp. 113–116. doi:10.1109/MobileCloud.2018.00025.

[12] Z. Rejiba, X. Masip-Bruin, E. Marin-Tordera, Analyzing the Deployment Challenges of Beacon Stuffing as a Discovery Enabler in Fog-to-Cloud Systems, 2018 European Conference on Networks and Communications, EuCNC 2018 (978) (2018) 271–276. doi:10.1109/EuCNC.2018.8442471.

[13] R. Venanzi, B. Kantarci, L. Foschini, P. Bellavista, MQTT-Driven sustainable node discovery for internet of things-fog environments, IEEE International Conference on Communications 2018-May (2018) 1–6. doi:10.1109/ICC.2018.8422200.

[14] R. Venanzi, B. Kantarci, L. Foschini, P. Bellavista, MQTT-Driven Node Discovery for Integrated IoT-Fog Settings Revisited: The Impact of Advertiser Dynamicity, Proceedings - 12th IEEE International Symposium on Service-Oriented System Engineering, SOSE 2018 and 9th International Workshop on Joint Cloud Computing, JCC 2018 (2018) 31–39doi:10.1109/SOSE.2018.00013.

[15] J. Gedeon, S. Zengerle, S. Alles, F. Brandherm, M. Muhlhauser, Sunstone: Navigating the Way through the Fog, Proceedings - 4th IEEE International Conference on Fog and Edge Computing, ICFEC 2020 (2020) 49–58doi:10.1109/ICFEC50348.2020.00013.

[16] S. Soo, C. Chang, S. N. Srirama, Proactive Service Discovery in Fog Computing Using Mobile Ad Hoc Social Network in Proximity, in: Proceedings - 2016 IEEE International Conference on Internet of Things; IEEE Green

Computing and Communications; IEEE Cyber, Physical, and Social Computing; IEEE Smart Data, iThings-GreenCom-CPSCom-Smart Data 2016, IEEE, 2017, pp. 561–566. doi:10.1109/iThings-GreenCom-CPSCom-SmartData.2016.126.

[17] T. M. Cover, P. E. Hart, Nearest Neighbor Pattern Classification, IEEE Transactions on Information Theory (1967). doi:10.1109/TIT.1967.1053964.

[18] M. RezaAbbasifard, B. Ghahremani, H. Naderi, A Survey on Nearest Neighbor Search Methods, International Journal of Computer Applications 95 (25) (2014) 39–52. doi:10.5120/16754-7073.

[19] O. Enqvist, F. Jiang, F. Kahl, A brute-force algorithm for reconstructing a scene from two projections, Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2011) 2961–2968doi:10.1109/CVPR.2011.5995669.

[20] OMNeT++, OMNeT++ Network Simulation Framework (2013).

[21] P. Lai, Q. He, M. Abdelrazek, F. Chen, J. Hosking, J. Grundy, Y. Yang, Optimal edge user allocation in edge computing with variable sized vector bin packing, in: Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics), 2018. doi:10.1007/978-3-030-03596-9{_}15.

Author 1:

Name: Afnan Abdulrahman Bukhari

Email: afabukhari@gmail.com , afnan.bukhari@student.uts.edu.au

Affiliation:

College of Computers and Information Technology, Taif university, Taif, KSA

School of Computer Science, University of Technology Sydney, Sydney, NSW, Australia

Author 2:

Farookh Khadeer Hussain

Head of Discipline (Software Engineering)

School of Computer Science and Australian Artificial Intelligence Institute

Faculty of Engineering and Information Technology

University of Technology Sydney Ultimo, NSW, 2007

Email: Farookh.Hussain@uts.edu.au

Author 3:

Name: Omar Khadeer Hussain

Email: O.Hussain@adfa.edu.au

Affiliation:

School of Business, University of New South Wales, Canberra, NSW, Australia

Conflict of Interest Statement

Declaration of interests

☒The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

☐The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: