Location-aware IoT Search Framework based on Data Messaging and Aggregation Techniques

Santosh Pattar*, Lakshmi K N*, Darshil Vala*, Venkatesh*,
Rajkumar Buyya†, Venugopal K R‡, S S Iyengar§, and L M Patnaik¶
* IoT Lab, University Visvesvaraya College of Engineering, Bengaluru, India.
† University of Melbourne, Melbourne, Australia.
‡ Bangalore University, Bengaluru, India.
§ Florida International University, Miami, USA.
¶ National Institute of Advanced Studies, Bengaluru, India.
e-mail: santoshpattar01@gmail.com

***Abstract*—With the deeper penetration of the Internet of Things (IoT) devices into the physical infrastructure and wider acceptance of IoT technologies by the community has created a tremendous opportunity for designers and developers to put forth applications that aim to improve the present state-of-the-art solutions. Location-based Service (LBS) provisioning is one such area where location data is utilized to offer user-centric services and thus improve personalized user experience. In this paper, we propose a search framework for the IoT ecosystem that offers location-aware services based on data messaging and aggregation techniques. We design a taxonomy for the classification of the IoT devices based on their mobility frequency and leverage it to design a priority scheme to address the co-located devices that offer similar services. Experimental results show that our proposed LBS Provisioning System is more effective in term of query resolution and storage requirements when compared to several existing works.**

***Index Terms*—Location based services, push/pull approach, data aggregation, priority assignment, query operation, smart museum.**

I. Introduction

With the rapid adaption of Internet of Things (IoT) technology into business, governmental, industrial and societal sectors today, several applications are being developed and offered (like smart home, healthcare, smart city, *etc*). The penetration of IoT devices into the physical world that are connected to the IoT infrastructure through the powerful cloud and edge computing platforms have made it possible to decentralize the computing processes required by the application [1], [2]. One of the primary services that is to be supported by these applications, called as Location Based Service (LBS), should cater to the needs of the end-users' based on their physical location or proximity [3].

The potency of an IoT application resides in the process of extraction and filtering of the data generated by the IoT devices. Location data provided by such a device is one of the meritorious feature among others that is utilized to offer user-centric services by the application (*i.e.*, LBS). The effective results of LBS are used to offer services such as the discovery of near-by services (like a coffee vending machine), suggest social events in the vicinity, turn-by-turn navigation, *etc*. However, due to the tremendous size of the IoT network, where the number of IoT devices getting connected to the IoT infrastructure is steadily increasing, it becomes an overwhelming challenge to choose a particular IoT device by the application to support LBS [4].

A search system casts a primary requisite for LBS applications to search and remit information for the demands of a user. It gathers the data from the IoT devices and performs operations like storing, indexing, ranking, securing, inspecting and supervising on the related features that are adduced with the location attribute. Over the past few years several techniques that support location-based services for an IoT ecosystem have been proposed, like Snoogle [5], MAX [6], Object Calling Home (OCH) [7]. Snoogle has sensor nodes embedded with every device in the IoT ecosystem. Sensor's data is annotated with textual descriptions using keywords and thus the system is only meant for static search only. Whereas, MAX is for exploring real-world entities that act as a user median agent. But, it is not suitable for a large scale network as the query has to be sent to every entity in the IoT network setup. OCH is a system used to locate the rampant locus of the lost things. It assigns the involved devices with identification tags. Nonetheless, the setup concerned with module computation causes communication overhead.

In this paper, we propose an LBS discovery method using data acquisition techniques (through Push and Pull methods). Here, the nearest IoT device providing the service is discovered in the network for a given user query. Our approach is composed of a pragmatic parallel system for searching IoT devices that offer LBS through a multi-tier fog based architecture. The devices are classified and arranged into layers based on their mobility frequency to facilitate the query resolution process. Furthermore, we develop a priority algorithm to rank and process the messages sent/received by the IoT devices and thus aids in prioritizing the services offered by similar devices. The priority is assigned for the request and response messages, based on some prior definition and further, the query is resolved by calculating the distance between the requested service and available services.

The rest of this paper is organized as follows. Section II confers survey of the relevant work. In Section III, we present the problem definition of the proposed work, its objectives, and assumptions adapted to accomplish the search process. Section IV details on the system model and incorporates the description of the background techniques, system architecture, and search algorithms. Section V encompasses implementation details with the experiments and obtained results. Finally, Section VI concludes the paper.

II. Literature Survey

IoT domain applications often include location-based services like tracking, managing IoT system with each device attached with different sensors and actuators. Various aspects of these works are discussed below.

Teran *et al.* [8] designed an IoT based localization indoor system using bluetooth low energy technology. The proposed system embeds two subsystems under the client-server paradigm and the IoT philosophy, called acquisition system and a central server. It is related to a simple location algorithm from received signal strength footprinting method that detects reference zones within closed environments. However, the physical object can abrupt bluetooth signal propagation. Bak *et al.* [9] presented a multi-tier cloud-based microservices provisioning for the advancement of location-based practices. The microservices provided by the proposed system includes contextual triggering, visualization, anomaly detection, and root cause analysis. But, the enormous amount of the raw data coming from several applications cannot be handled by the system.

Ma *et al.* [10] introduced a methodology to provide precise and effortless localization solution, which includes a foglight comprising of off-the-shelf light sensors integrated with IoT resources. This system enhances the easy location of the IoT devices associated with sensors. However, it has its flaws in case of cost issues as it is expensive to have off-the-shelf sensors when there are too many devices. Tao *et al.* [11] proposed a great-alternative-region strategy based on the proposition for partitioning the various data resources in a smart city. An improved genetic algorithm is used to obtain a prime positioning pattern of the devices. The proposed system supports effective identification of resources. Despite this, it does not work for divergent network topologies.

Zhang *et al.* [12] induced a secure location of things framework to solve the ultimatum of malignant attacks. This approach has two algorithms in order to locate a node/resource, the first algorithm is based on the probabilistic model and the second algorithm is based on differential-time-of-arrival. The proposed framework provides considerable upgradation of localization time execution. Yuan *et al.* [13] introduced an enhanced fleet search and destiny peak search technique based node location strategy concerning to find the locale of a node in an edge computing environment by employing a method of clustering. The aspired scheme is vigorous, pliant, has low time complexity and supports in avoiding NP-hard problems of the familial server placement schemes. Although the dynamicity and mobility complexion of the devices cascaded in fog computing has not been addressed here.

Ikebe *et al.* [14] developed a live data search architecture for the IoT as most of the available architectures work better only for the static data. The induced mechanism discovers various resources generating data to deal with dynamic nature. The sated work improves the services offered in the IoT by enhancing accessible exploration of resources. Yet, the proposed architecture is designed by assuming simple queries and does not work for the complicated requirements like analysis of images, videos, sound *etc*. It also lacks consideration of load balancing and scaling methods. Miao *et al.* [15] solicited searchable encryption technologies to encrypt data collected by multiple fog nodes based on attributes. The set forth system averts the arrival of unrelated search results and is feasible through minimal storage load. However, the system does not provide an apt solution for vague keyword searches and secure channel in the system leads to a large transmission burden.

Kamilaris *et al.* [16] introduced a search engine for the semantic web of things that includes physical devices, services, data *etc.*, in order to identify devices universally linked to the web and also designed an evaluation procedure resembling execution and performance over the web. However, the proposed scheme does not identify relevant data endpoints and results in the discovery of inappropriate resources. Guo *et al.* [17] asserted the development of the IoT search service by identifying leading data to construct a real-time IoT data warehouse. An incentive scheme is designed to overcome the problems of IoT search (which is different from the web search as it depends on dynamic and heterogeneous data). But, the work leads to privacy and security concerns and thus misuse of resources, as the user restriction is not taken into consideration.

III. Problem Statement

In this section, we describe the problem under consideration and list the objectives for the proposed solution and assumptions that we make for it.

A. Problem Statement

Consider an IoT ecosystem with a large number of IoT devices connected to it. Various attributes, like location, ID, battery life *etc*., quantify these devices and the services offered by these devices is sought out by the user through a search application developed for the said IoT ecosystem [18]. Our aim in this work is to effectively list the services that match the user demands based on his/her location in the best possible way by considering the possibility that several devices which offer similar services may be co-located.

B. Objectives

The goal of our current work is to provide effective location-based service in an IoT ecosystem by considering a multi-tire hierarchical data messaging architecture that makes use of push and pull aggregation techniques, as mentioned below.

1) Minimize communication overhead so as to facilitate effective query resolution.
2) Prioritizing the services of similar devices that are correlated and thus offer efficient search results.

C. Assumptions

We make the following assumptions to construct the proposed LBS search architecture system.

1) The IoT ecosystem is already set up that embeds search middleware, fog devices, and IoT resources. Users can utilize the services as well as act as a mediator through IoT devices to route the query.
2) We assume that the location of each device is automatically retrieved and updated at the repositories.

IV. Proposed System Model

In this section, we describe the proposed search system for the provision of LBS in an IoT ecosystem by leveraging the concepts of data messaging and aggregation techniques in an IoT network. In the first subsection, we illustrate an application scenario where LBS provisioning plays a pivotal role to offer user-centric services. In the next subsection, we give a background on data messaging and aggregation techniques. Finally, we discuss the system architecture, message prioritization, data aggregation and query resolution algorithms in the last subsection.

A. LBS Application Scenario

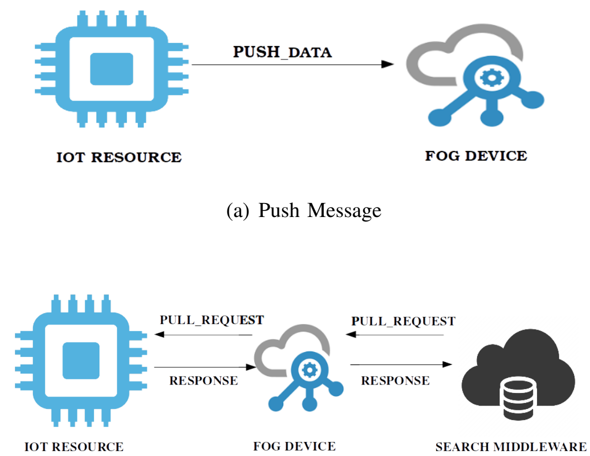
Museums are one of the thriving centers of tourism that have stood the test of time. They offer a holistic view of the art, culture, lifestyle, *etc.* from several standpoints and are often point for social events and information exchange. Today, due to digitization efforts an IoT based digital system can be positioned in the museum to support several operations and thus this system also supports to generate more revenue by incorporating the personalized experience of the user. Here, an LBS provisioning search system caters to the user demands by considering their physical proximity and location to recommend personalized services. Few such scenarios are listed as follows, (i) various artworks belonging to diverse categories that are housed at different locales of the museum can be navigated using map services offered by LBS search system through user preferences (ii) notification on the devices carried by the visitors regarding which location is less/more crowded, contextual information about the facilities like cafeteria, porchways, toilets, souvenir's shop *etc.*

B. IoT Data Messaging and Aggregation Techniques

In an IoT ecosystem, there exist several components like cloud and fog system, IoT devices and search facilitators that collaborate with each other to achieve the given complex task. Data has to flow through them at several levels and there exist two common techniques, know as PUSH and PULL, that are discussed below.

a) PUSH Approach: It is an IoT messaging approach where the data sensed by an IoT device is sent to the nearest fog device continuously. Fig. 1(a) shows the structure and path of a push message. Push operation is generally performed by the devices that have minimal storage capacity. It is the process of passing data from mobile IoT devices to the data accumulating device. In our use case scenario (smart museum) devices like an audio guide, virtual reality, various user wearables acts as IoT devices which continuously pushes data to their nearest fog device.

b) PULL Approach: It is an IoT messaging approach where the data is retrieved from an IoT device when requested. Fig. 1(b) illustrates the structure and path of a pull message. It in-turn has two message types pull-request and pull-response. If the required result is not found in search middleware the query is passed as a pull request message. The result fetched by an IoT resource for a pull request message is considered as a pull response message. Initially, the search is begun from the search middleware and the result is passed to a requested user if derived.



(a) Push Message

(b) Pull Message

Fig. 1. An Overview of PUSH/PULL Messaging Techniques

Both push and pull approaches vary in terms of speed of search, efficiency in performing an operation, result accuracy, traffic in data transmission, the capacity of the database *etc.* The push operation works faster compared to pull since there is no process of searching as the data is previously pushed into the respective fog devices. When mobility nature of devices is considered the efficiency of an approach reduces and leads to less accurate results to the user query. Sometimes the traffic inflow of data can enormously increase being data-pushed perpetually and requires a huge database resulting in excessive resource cost. The pull operation is slow since it embeds a process of searching among several levels starting from search middleware through fog devices and IoT resources, thus has a low response rate. Whereas this approach is more efficient, gives accurate results, cost-efficient as the resources consumed is less due to less usage of storage devices.

C. Taxonomy of the LBS Search System

We propose a taxonomy to classify the components of an LBS search system based on their mobility patterns. The Table I describes the list of resources of a smart museum and its characteristics, like categorization of different services

TABLE I
TAXONOMY OF THE COMPONENTS IN THE PROPOSED LBS PROVISIONING SCHEME FOR SMART MUSEUM

Service	Device	Mobility	Message	Architectural Component
Security	CCTV Fire Extinguisher	Static Pseudo Dynamic	Push/Pull Push/Pull	Fog Device Fog Device
Smart-guided Tour	Audio Virtual Reality	Dynamic Dynamic	Push Push	IoT Resource IoT Resorce
Amenity	Wheel Chair Vending Machine	Pseudo Dynamic Static	Push/Pull Push/Pull	Fog Device Search Middleware
Information Service	Bulletin Board Ticket Vending Machine	Static Static	Push/Pull Push/Pull	Search Middleware Search Middleware

available in the smart museum, devices involved in each service, the type of the mobility they belong to, kind of messages it sends/receives and the sort of an architectural component under which it is grouped, which can be one among an IoT resource, a fog device or a search middleware.

The mobility of the devices is classified based on movement frequency. It is divided into three categories as (i) static, (ii) pseudo-dynamic, and (iii) dynamic. The devices that are immobile fall under static. The devices that are mostly static but can be moved based on the need are grouped into pseudo-dynamic. Devices that are in continuous motion are considered dynamic.

D. LBS Provisioning Scheme

The flow of the proposed LBS scheme is briefed in this sub section by taking the smart museum as an use case scenario. As shown in the Fig. 2 the components of the LBS search system are categorized under three levels namely (i) search middleware, (ii) fog devices, and (iii) IoT devices. Search middleware acts a facilitator that accepts, forwards, aggregates and returns the messages in the search system. While, fog devices manage a set of IoT devices found under their vicinity.

User asks for a service in the form a query. Once a query is received, the search-middleware generates a pull request message for a requested service and broadcasts it to all fog devices. Two search operations are handled at the fog device,

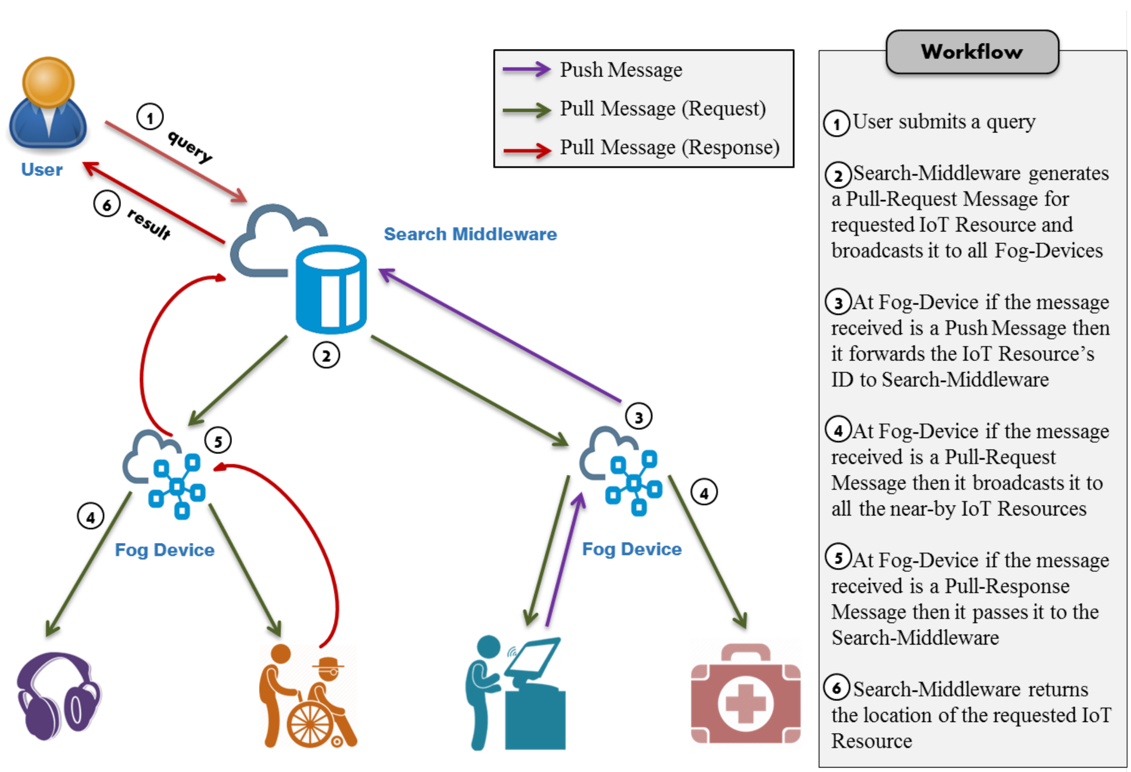


Fig. 2. Overall Flow of the Proposed LBS Provisioning Scheme through the Smart Museum Scenario.

one is to broadcast the location of a new found device through (Pull message) and other is to supply the results of the query to the search-middleware if found.

Algorithm 1: Prioritize Message

Input: PUSH or PULL Message from an IoT Resource/Search Middleware.
Output: Priority of the message.

```
Consider the IoT Ecosystem.
Consider the Taxonomy of IoT Resource.
if message = PULL then
    /* Priority Assignment for Query Request and
       Response Messages */
    if message = PULL.request then
        message.priority ← HIGH (4);
    end
    else if message = PULL.response then
        message.priority ← HIGH (4);
    end
    else
        return;
    end
end
else
    /* message is of type PUSH */
    /* Priority Assignment for Messages from IoT
       Resources based on Mobility */
    if resource.type = dynamic then
        message.priority ← VERY HIGH (5);
    end
    else if resource.type = pseudo-dynamic then
        message.priority ← HIGH (4);
    end
    else
        /* IoT resource is of static type */
        message.priority ← LOW (3);
    end
end
```

Algorithm 1 shows the priority assignment. In pull messages both request and response is assigned with high priority. In case of dynamic devices, push message is assigned with very high priority. whereas, pseudo dynamic devices of message type push is assigned with high priority and the push message of a static device is assigned with low priority.

The processing of query is performed after priority assignment. If the message has a priority less than or equal to three, the message having labels of an IoT resource is pushed in to a local array. For the messages with a priority greater than three, user request message is transmitted to all IoT resources. Finally, the response is sent to the respective search middleware. Once the query is processed, the location of a matched device is returned, if the message is push. The request is transmitted for the next IoT resources for pull messages.

The Algorithm 2 renders the operation of an user query. When an user implores a query to locate a desired item, the

Algorithm 2: Query Operation

Input: User Query to Search an Item.
Output: Location of an Item (L) with its Distance from an User (D).

Consider the IoT Ecosystem (Fig. **??**).
Let $\mathrm{S_M}$ be the set of search middlewares, $\mathrm{S_M} = (S_1, S_2, \ldots, S_n)$
Let $\mathrm{F_N}$ be the number of fog devices, $\mathrm{F_N} = (F_1, F_2, \ldots, F_n)$
foreach $S_i \in \mathcal{S}_M$ **do**
  L ← retrieve the list of items under $\mathrm{S_i}$
  if *Item found in Middleware* **then**
    $\mathrm{loc_i}$ ← $\mathrm{loc_i}$.location;
    $\mathrm{d_{su}}$ ← Calculate Distance between user and search middleware using equation 1;
    return $\mathrm{loc_i}$ and $\mathrm{d_{su}}$;
  end
  else
    foreach $F_i \in \mathcal{F}_N$ **do**
      if *Item found in fog device* **then**
        $\mathrm{loc_i}$ ← item.location;
        $\mathrm{d_{fs}}$ ← Calculate Distance between fog device and search middleware using equation 2;
        return $\mathrm{loc_i}$ and $\mathrm{d_{fs}}$;
      end
      else
        $\mathrm{loc_i}$ ← item.location;
        $\mathrm{d_{if}}$ ← Calculate Distance between IoT Resource and fog device using equation 3;
        return $\mathrm{loc_i}$ and $\mathrm{d_{if}}$;
      end
    end
  end
end

quest is initiated with the basis of search middleware, where its database is explored for the result. Once the query is solved, the location of the item is derived and the distance is calculated between user and an item with the help of equations 1-4. If the result is not found then the pull request is sent to fog devices that are registered under the search middleware. The location of each object is tracked based on some location tracker sensor and stored in the respective databases which will be in the form of latitude and longitude co-ordinates.

$$\mathrm{d_{(su)}} = (s_{\mathrm{y}} - u_{\mathrm{x}})^2 + (u_{\mathrm{y}} - s_{\mathrm{x}})^2$$
$$d_{\mathrm{su}} = \sqrt{d_{\mathrm{(su)}}} \tag{1}$$

$$\mathrm{d_{(fs)}} = (f_{\mathrm{y}} - s_{\mathrm{x}})^2 + (s_{\mathrm{y}} - f_{\mathrm{x}})^2$$
$$d_{\mathrm{fs}} = d_{\mathrm{su}} + \sqrt{d_{\mathrm{(fs)}}} \tag{2}$$

$$\mathrm{d_{(if)}} = (f_{\mathrm{y}} - s_{\mathrm{x}})^2 + (s_{\mathrm{y}} - f_{\mathrm{x}})^2$$
$$d_{\mathrm{if}} = d_{\mathrm{fs}} + \sqrt{d_{\mathrm{(if)}}} \tag{3}$$

Here, (s_x, s_y), is the location of an IoT resource. d_{su} (equation 1), is the distance between user and search middleware, d_{fs} (equation 2), is the distance between fog device and search middleware, d_{if} (equation 3), is the distance between IoT Resource and fog device.

V. Implementation and Performance Analysis

Our work follows a streamline approach, since all the IoT resources are connected to the respective fog devices and it is in turn connected to a search middleware. Despite of type of the message (Push/Pull) the data being generated by an IoT resource is stored in the corresponding databases and always passes through a search middleware to reach to a requested user. There are various metrics which has to be taken under analysis for evaluation, that are elucidated below.

A. Query Latency

The time taken to resolve a query is described using query latency. It is the time between an user issuing a query requesting for the service and return of the requested match to the user. In our approach, the response is quite immediate in case of PUSH messages when compared with PULL messages thus the query latency varies, as a whole the query latency is less since the search process occurs at faster rate due to hierarchical structure. Figure 3 shows the comparison of query latency between the distributive indexing scheme [14] and the location search using push/pull approaches.

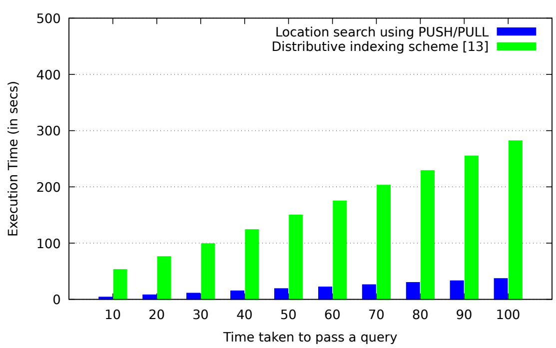


Fig. 3. Comparison between Query Latency of Traditional Search and Location search using Push/Pull approaches

B. Query Participation

The current work includes the participation of devices (nodes) in resolving a query. It estimates the number of which takes part in resolving a particular query to obtain results. Most of the static devices pushes the data to its next level, when the same is requested result can be directly fetched from a search middleware or a fog device. The proposed scheme minimizes the redundant nodes as once the result is found in any level it is immediately returned to user. Less the number of nodes traversed more will be the time efficiency, which has been followed in the current work. Figure 4 shows the comparison between query participation of snoogle [5] and location search using Push/Pull approaches, which depicts the processing time of a user query and the number of nodes traversed in finding the result.

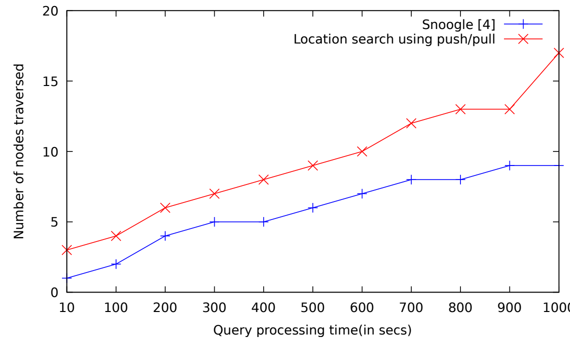


Fig. 4. Comparison between Query Participation of Traditional Search and Location search using Push/Pull approaches

C. Storage Requirement

The databases placed at various levels meant for storage. Search middleware always has a high storage capability since the dynamic devices which generates data continuously and maintains own database which is later pulled if requested. The data of static devices are sent to database of fog devices and search middleware. Since the storage devices are aptly used leads to the efficient usage of storage devices. Figure 5 shows the comparison between storage requirement of snoogle [5] and location search using Push/Pull approaches.

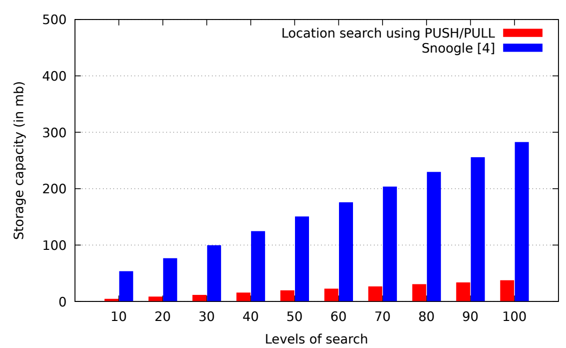


Fig. 5. Comparison between Storage Requirement of Traditional Search and Location search using Push/Pull approaches

VI. Conclusions and Future Work

Location data plays an important role in the provisioning of user-centric services in an IoT ecosystem. In the recent past several works have addressed the issue of searching IoT devices and the services that they offer, however they do not consider the mobility issue of the IoT devices and also incur huge communication and storage overhead. In this work, we devise a location-aware search system for the IoT through multi-tire fog based architecture. We classify the components of the IoT ecosystem based on their mobility frequency and propose a priority scheme to resolve conflicts between co-located devices that offer similar services. The proposed LBS Provisioning Scheme outperforms the existing techniques with respect to the number of nodes contacted, query resolution time and storage requirement. It also achieves comparatively less overhead in communication owing to the Push/Pull based data aggregation technique coupled with the priority scheme and mobility taxonomy. Although, we have addressed the communication overhead along with the prioritization of messages in an IoT ecosystem, the issue of trust and privacy and trust of devices and users with the ecosystem is a major concern and in future we would address these challenges through the use of inference rules to enhance privacy and incorporate trust through mobility behaviors of users or IoT devices.

REFERENCES

[1] S. Pattar, S. K. Dwaraka, V. Darshill, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Progressive Search Algorithm for Service Discovery in an IoT Ecosystem," *in Proceedings of the 2019 International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 1041–1048, 2019.

[2] S. Pattar, C. R. Sandhya, V. Darshill, D. Chouhan, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Context-oriented User-centric Search System for the IoT based on Fuzzy Clustering," *in Proceedings of the International Conference on Computational Intelligence,Security & IoT (ICCISIoT)*, pp. 1–14, 2019.

[3] J. Zhou, Z. Cao, Z. Qin, X. Dong, and K. Ren, "LPPA: Lightweight Privacy-preserving Authentication from Efficient Multi-key Secure Outsourced Computation for Location-based Services in VANETs," *IEEE Transactions on Information Forensics and Security*, pp. 1–13, 2019.

[4] S. Pattar, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "Searching for the IoT Resources: Fundamentals, Requirements, Comprehensive Review, and Future Directions," *IEEE Communications Surveys & Tutorials*, vol. 20, no. 3, pp. 2101–2132, 2018.

[5] H. Wang, C. C. Tan, and Q. Li, "Snoogle: A search engine for pervasive environments," *IEEE Transactions on Parallel and Distributed Systems*, vol. 21, no. 8, pp. 1188–1202, 2009.

[6] K.-K. Yap, V. Srinivasan, and M. Motani, "MAX: Human-Centric Search of the Physical World," *in Proceedings of the 3rd International Conference on Embedded Networked Sensor Systems*, pp. 166–179, 2005.

[7] C. Frank, P. Bolliger, F. Mattern, and W. Kellerer, "The Sensor Internet at Work: Locating Everyday Items using Mobile Phones," *Pervasive and Mobile Computing*, vol. 4, no. 3, pp. 421–447, 2008.

[8] M. Terán, J. Aranda, H. Carrillo, D. Mendez, and C. Parra, "Iot-based System for Indoor Location Using Bluetooth Low Energy ," *in Proceedings of the 2017 IEEE Colombian Conference on Communications and Computing (COLCOM)*, pp. 1–6, 2017.

[9] P. Bak, R. Melamed, D. Moshkovich, Y. Nardi, H. Ship, and A. Yaeli, "Location and Context-based Microservices for Mobile and Internet of Things Workloads," *in Proceedings of the 2015 IEEE International Conference on Mobile Services*, pp. 1–8, 2015.

[10] S. Ma, Q. Liu, and P. C.-Y. Sheu, "Foglight: Visible Light-Enabled Indoor Localization System for Low-Power IoT Devices," *IEEE Internet of Things Journal*, vol. 5, no. 1, pp. 175–185, 2018.

[11] M. Tao, K. Ota, and M. Dong, "Locating Compromised Data Sources in IoT-Enabled Smart Cities: a Great-Alternative-Region-Based Approach," *IEEE Transactions on Industrial Informatics*, vol. 14, no. 6, pp. 2579–2587, 2018.

[12] P. Zhang, S. G. Nagarajan, and I. Nevat, "Secure Location of Things (SLOT): Mitigating Localization Spoofing Attacks in the Internet of Things," *IEEE Internet of Things Journal*, vol. 4, no. 6, pp. 2199–2206, 2017.

[13] X. Yuan, Y. He, Q. Fang, X. Tong, C. Du, and Y. Ding, "An Improved Fast Search and Find of Density Peaks-Based Fog Node Location of Fog Computing System," *in Proceedings of the 2017 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData)*, pp. 635–642, 2017.

[14] T. Ikebe, H. Noguchi, and N. Hoshikawa, "Distributed Live Data Search Architecture for Resource Discovery on Internet of Things," *in Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 591–596, 2016.

[15] Y. Miao, J. Ma, X. Liu, J. Weng, H. Li, and H. Li, "Lightweight Fine-grained Search over Encrypted Data in Fog Computing," *IEEE Transactions on Services Computing*, vol. 12, no. 5, pp. 772–785, 2019.

[16] A. Kamilaris, S. Yumusak, and M. I. Ali, "WoTS2E: A Search Engine for a Semantic Web of Things ," *in Proceedings of the 2016 IEEE 3rd World Forum on Internet of Things (WF-IoT)*, pp. 436–441, 2016.

[17] Y. Guo, L. Fang, K. Geng, L. Yin, F. Li, and L. Chen, "Real-time Data Incentives for Iot Searches ," *in Proceedings of the 2018 IEEE International Conference on Communications (ICC)*, pp. 1–6, 2018.

[18] S. Pattar, C. R. Sandhya, V. Darshill, R. Buyya, K. R. Venugopal, S. S. Iyengar, and L. M. Patnaik, "SoCo-ITS: Service Oriented Context Ontology for Intelligent Transport System," *in Proceedings of the 2019 The 7th International Conference on Information Technology: IoT and Smart City (ICIT-2019)*, pp. 1–6, 2019.