A Decentralized Locality-preserving Context-aware Service Discovery Framework for the Internet of Things

Juan Li, Nazia Zaman
Computer Science Department
North Dakota State University
Fargo, USA
{j.li, nazia.zaman}@ndsu.edu

Honghui Li
College of Computer and Information Engineering
Inner Mongolia Agricultural University
Hohhot, China
lihh@imau.edu.cn

Abstract— **Today's Internet is shifting towards a larger and smarter scenario known as the Internet of Things (IoTs). The IoT envisions a multitude of heterogeneous objects and interactions with physical environments. In this environment, locating desirable services is challenging due to the considerable diversity, large number, dynamic behavior, and geographical distribution of the services provided by physical objects. In this paper, we propose a context-aware semantics-based service discovery mechanism – LOCA that can effectively locate services based on the context requirements. The proposed discovery framework is built on a fully distributed peer-to-peer (P2P) architecture which is scalable and robust. Moreover, a key feature of the discovery mechanism is its support of content and path locality. This feature can enhance the integrity of an organization and thus greatly improve the security of the organization. The effectiveness of the proposed framework is demonstrated through comprehensive simulation studies.**

Keywords- Internet of Things; discovery; context-aware; semantics

I. INTRODUCTION

As more objects are becoming embedded with sensors and gaining the ability to communicate, the resulting information networks – the Internet of Things (IoT) has emerged as a global Internet-based information architecture facilitating the exchange of goods and services. The IoT connects uniquely identifiable embedded computing devices within the existing Internet infrastructure to offer advanced connectivity of devices, systems, and services. IoT goes beyond machine-to-machine communications and covers a variety of protocols, domains, and applications [1]. The interconnection of these embedded devices, is expected to usher in automation in nearly all fields, covering wide range of applications such as healthcare, utilities, and transportation [2].

In order to effectively use services and data generated in the IoT, search and discovery mechanisms are crucial. These mechanisms should support locating resources and services related to an entity of interest in the physical world. Traditional web service discovery approaches are not suitable for service discovery in IoTs, due to the differences between real world services and traditional web services. As IoT resources/services are enabled by devices located in widely distributed, and heterogeneous information systems. There could be in high demand by business and end user applications. Designing an effective and efficient discovery mechanism faces many new challenges and requirements: Real-world services are provided by IoT devices in the physical world; normally, data of real world objects and events are available globally and in vast amounts. It is anticipated that the number of embedded devices connected to the Internet will be orders of magnitude larger than the number of computers connected to the Internet today [3]. This challenge requires the discovery mechanism to be scalable with respect to large number of objects. Moreover, due to the mobility of physical entities and devices and other dynamic changes, the relations between IoT Services, IoT devices, and the physical environments may also change over time. Most IoT discovery services should provide near real-time states of things in the physical world, which is quite different from traditional web services which are entirely virtual entities encapsulating enterprises business logics. Furthermore, real-world services are deployed in resource constrained devices, e.g., with limited battery, computing, communication and storage capabilities. This requires significant simplification, optimization, and adaptation of existing tools and standards [4]. Lastly, the IoT presents new security challenges in security, credentialing, identity management, and privacy [5].

To address the aforementioned challenges of discovery in IoTs, we propose a fully decentralized LOcation-preserving Context-Aware discovery framework – LOCA . Context awareness plays an important role in the discovery framework to enable services customization according to the current situation with minimal human intervention [22]. The proposed framework uses semantic web technologies, in particular ontology to encode the context information and match queries with services to select the most appropriate services. Ontology provides a common understanding of the context, and it can help the discovery service to infer relationships between entities and context. Therefore, the ontology model help LOCA achieves context interoperability in heterogeneous environments.

We propose a peer-to-peer (P2P) based overlay network as a flexible communication infrastructure of LOCA to

978-1-4673-7281-7/15 $31.00 © 2015 IEEE
DOI 10.1109/SCC.2015.51

implement scalable, robust, and locality-preserving distributed IoT service discovery. P2P architecture, especially distributed hash tables (DHTs), has been used for information exchange and resource discovery in many IoT applications, such as supply chain [6] and EPCglobal network [7]. However, DHTs have an inherent shortcoming which may obstruct their wide application in IoTs: DHTs cannot control where data (or metadata) is stored. This causes two problems: (1) (meta)data may be stored far from its frequent users; (2) it may be stored outside the administrative domain to which it belongs [8]. In IoTs, resources and services normally involve physical entities, therefore, it is important to guarantee secure access of data and device. For this purpose, we propose to use SkipNet [8] as the discovery infrastructure that allows for explicit control over content placement and guarantee access path locality. LOCA communication overlay adds content locality by explicitly placing data on specific overlay nodes. In addition, it adds path locality by guaranteeing that message traffic routed between two overlay nodes within the same organization stays within that organization. These properties provides advantages for availability, performance, manageability and security of the system.

The rest of the paper is organized as follows. In Section II, we survey existing work on resource discovery in the IoTs. In Section III, we present how we model and match context information to support context-awareness. In Section IV, we propose a decentralized locality preserving discovery communication model. In Section V, we evaluate the proposed methods and show the effectiveness of the proposed mechanism with a comprehensive set of simulations. Concluding remarks are provided in Sections VI.

II. Related Work

The development of the Electronic Product Code (EPC) to support the spread use of RFID in supply chain management (SCM) by Auto-ID [9] and EPCglobal [10] are one of the most important efforts in IoTs. As presented in [11], several research projects have implemented the discovery service for the EPC network. The first implementation was a Directory Look-up approach proposed by Beier et al. [12]. When an EPC event is first stored in a company's EPC information service (EPCIS), a discovery service is notified and related information is stored in the repository. Users can query the discovery service with an EPC. Related EPCIS URLs will be supplied to the user. The BRIDGE project [13], provides high-level descriptions and analysis of a number of approaches for implementing discovery services. In [13], eight discovery service approaches were evaluated and four were identified as promising candidates for large-scale EPC discovery services. These eight approaches are classified as two categories, namely, the *Directory Service* (DS) approaches and *Query Relay* (QR) approaches.

Recently, the use of P2P systems to implement scalable and robust distributed service discovery in IoTs has been investigated. To reduce the cost of broadcasting in the P2P network, various mechanism have been proposed. For example, the project described in [14] employed multicasting to replace broadcasting; the project presented in [15] performed selective forwarding to limit the number of search hops; and the work introduced in [16] proposed the probabilistic forwarding of queries. The aforementioned P2P networks are unstructured networks, which do not impose a particular structure on the overlay network by design, but rather are formed by nodes that randomly connecting to each other. On the other hand, there are IoT discovery frameworks based on structured P2P networks. For example, structured P2P architecture have been used for information exchange and resource discovery among participants of a supply chain [7]. Manzanares-Lopez et al. proposed a structured P2P discovery service for the EPCglobal network [17]. It offers item-level track and trace capabilities along the whole supply chain even when items are not directly visible. Most of the structured P2P systems use distributed hash tables (DHTs) as their underlining communication structure.

As pointed out by Paganelliet al., most of the aforementioned DHT-based discovery frameworks support queries by providing an object identifier (typically the EPC code) as input, but they do not support more flexible query schemes based on object attributes, though these types of information queries could become very important in the future IoT [18]. To support complex queries, in particular, multi-attribute and range queries, Paganelliet al. [18] proposed a layered functional architecture over DHTs. The proposed architecture used a Space Filling Curves (SFC) linearization technique for mapping a multidimensional domain into a one-dimensional one. It used a Prefix Hash Tree (PHT) search structure leveraging on a generic DHT interface. The DHT is implemented based on the Kademlia algorithm [19]. Another system, ERQ [20], used similar approach with Balanced Kautz (BK) tree to map the m-dimensional data space onto DHT nodes, and then used a distributed algorithm to process range queries. As mentioned, existing DHT approaches cannot control where the service metadata is placed, which is an disadvantages for system security.

Context-awareness plays an important role in IoT. Modeling context and selecting services based on the right context therefore is vital for IoT service discovery. As summarized by Perera et al. [21], context modeling can be classified as six categories: key-value, markup, graphical, object-based, logic-based, and ontology-based. Our context modeling belongs to the ontology-based category, we generalize and reuse some of the existing ontologies [22-24] and model the most fundamental context. In Section III, when we present our ontological modeling and reasoning schemes, we also provide detailed description of the most related existing research in these fields. Therefore, we do not explain the details of existing work in this section.

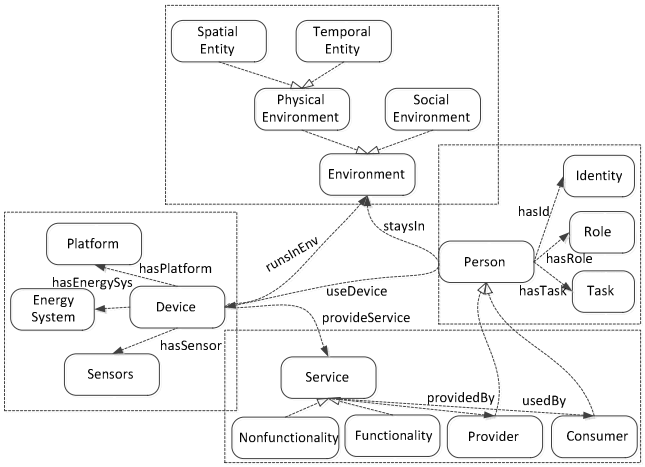

Fig. 1. Part of the defined high-level context ontology

III. Context Modelling and Reasoning

A. Context Modeling

Development of context-awareness in service discovery should be supported by adequate context modelling and reasoning techniques. In this paper, we propose to create a semantics-based mechanism to support context-aware service discovery. Previous work has shown that context modeling without a foundation of common ontologies have some deficits: "the use of ad hoc representation schemes lacking shared vocabularies can hinder the ability of independently developed agents to interoperate, the use of objects and data structures of low expressive power provides inadequate support for context reasoning" [22]. To overcome these problems, we propose an ontological model to represent context and automatically infer relationships between user, device, service, and environment. An ontology is an explicit specification of the conceptualization of a domain [22]. Ontology provides formal, machine-executable meaning on the concepts, and supports inference mechanisms that can be used to enhance semantic matchmaking. Therefore, a context ontology will greatly improve the interoperability between diverse devices and services, and enable automatic context matching without human intervention.

Different ontologies for context have been investigated in the past. In our project, we generalize and reuse existing IoT-related context ontologies such as [22-24]. In our ontological model, context information is described in an easily extensible way to facilitate sharing of collected information. We do not limit context information to a fixed set of attributes or properties. Instead we provide freedom of meaning choices to users in different applications. As specified by [23], completely formalizing all context information is likely to be an in-surmountable task, therefore we only model the most fundamental context, i.e., a set of upper-level entities, and provide flexible extensibility to add specific concepts in different application domains (e.g., home, office or vehicle). Common context concepts shared by different domains will be modeled as general context entities. Fig. 1 shows part of the high-level ontology defined in LOCA, in which context information includes both static and dynamical context such as function description, interface, running status and environment description.

B. Context Reasoning and Matchmaking

To automatically match and discover services based on the changing context, we propose two context filtering techniques: logic-based context reasoning and semantics-based context matching. To explain the role of context filtering, we present a sensor usage scenario as an example. In this scenario, a user interested in knowing the "*flood status in North Dakota"* would like to be notified of her interested subject. Through context filtering, this user will be notified when there's new sensing data about "*crest of Red River"*. The reasoning is based on the ontological knowledge: "*crest isRelatedTo flood"*, and "*RedRiver* is*LocatedIn NorthDakota*". As we use ontology to model context, context information can be processed with logical reasoning mechanisms. Ontology in OWL format is description logic (DL) in nature, which allows us to exploit the considerable existing body of DL reasoning. Besides supporting OWL-based ontology reasoning, we also provide user-defined reasoning by using first-order predicates. Users can create rules within the entailment of first order logic.

Another filtering technique is semantic matching. Compared with the simple keyword-based matching approaches, our matching mechanism overcomes differences in vocabularies. Existing methods on measuring the semantic similarity between two objects in the field of information retrieval and information integration are very comprehensive and computationally intensive. We take advantage of the context ontology and propose a simple mechanism to compute the semantic similarity. In particular, we extend and simplify our previously proposed distance-based approach [25], which converts the problem of measuring semantic distance of two entities to measuring the distance between them in an ontology graph. Most distance-based approaches assume that all edges in the graph have a uniform weight, which does not accurately reflect the edges' semantic variability. We proposed a weighted-distance measurement approach, which improves the accuracy by integrating factors including the depth of a node in the ontology hierarchy and the type of the links connecting nodes in the format as weight into the measurement metric. Specifically, when exploring semantic relations, we assign different types of links with different semantic distance-factor. For example, we assign the smallest distance-factor to the *equivalent-class* edge that denotes the two concepts at the end of edge are identical. Moreover, we generally consider that the distance-factor of *sub-class* (*super-class*) relation is shorter than that of other relations. To support flexibility, we allow applications and users to customize the distance-factor to reflect their special need. Another important factor affecting the semantic

distance is the depth of the nodes in the ontology hierarchical graph. In particular, concept nodes sharing ancestors (i.e., common more general concepts) at lower levels should be more similar than those whose common ancestors are at higher levels. To materialize the aforementioned principle, we formally define the similarity measurement as following. The semantic distance between two concepts C_a and C_b in a given ontology is defined as:

$$dis(C_a, C_b) = \frac{1}{2}\left(\frac{\sum_{i \in path(C_a \text{ to } C_p)} w_i dis(C_i, C_{i+1})}{\sum_{i \in path(C_a \text{ to } C_{root})} w_i dis(C_i, C_{i+1})} + \frac{\sum_{j \in path(C_b \text{ to } C_p)} w_j dis(C_j, C_{j+1})}{\sum_{j \in path(C_b \text{ to } C_{root})} w_j dis(C_j, C_{j+1})}\right),$$

where C_p is the common ancestor of C_a and C_b in the hierarchical ontology graph, C_{root} is the root of the ontology tree, C_{i+1} is C_i's parent, and w_i is the weight of edge presented as a distance factor (i.e., the closer relationship, the smaller the distance). The semantic distance above defines the distance as a "relative" distance to the distance between nodes and their common ancestors, thus it integrates the edge weight with the depth and the length of the shortest path.

IV. Decentralized Discovery Model

In order to efficiently locate desirable services, we propose a P2P-based fully decentralized discovery mechanism, LOCA. DHT-based P2P systems have been applied in IoTs to implement scalable and robust distributed discovery services. DHTs can provide good load balancing properties, but they cannot control where data is stored. Although data can be encrypted and digitally signed, storing them on an arbitrary overlay node outside the organization may be susceptible to attacks. Our discovery mechanism adopts the SkipNet [8] framework. SkipNet is a general purpose, scalable, fault tolerant overlay that allows for explicit control over content availability and placement. Content locality can improve security by allowing one to control the administrative domain in which data resides. Path locality provides additional security benefits to an overlay that supports content locality.

LOCA adopts a pure distributed SkipNet overlay to construct a decentralized index structure. Like DHTs, it offers better scalability and eliminates the bottleneck. SkipNet uses the concept of Skip List that maintains a sorted list of nodes as well as pointers that "skip" over varying number of nodes. In SkipNet overlay with *n* nodes, every node maintains connections to *O(log n)* other nodes, and any node can send a message to any other node while traversing only *O(log n)* intermediate hops. SkipNet nodes are organized into several doubly-linked lists. Each node is a member of several of these lists, but not all of them. Each of these lists is sorted using the nodes' string identifiers. Every pair of nodes that are adjacent in these lists forms a bidirectional network connection. Fig. 2 depicts a SkipNet overlay containing *8* peers, which are simultaneously interconnected at every level (i.e., from level 0-3). The pointers at level *0* points to neighboring peers. Pointers at level *1* point to peers that are two hops away. As a result the

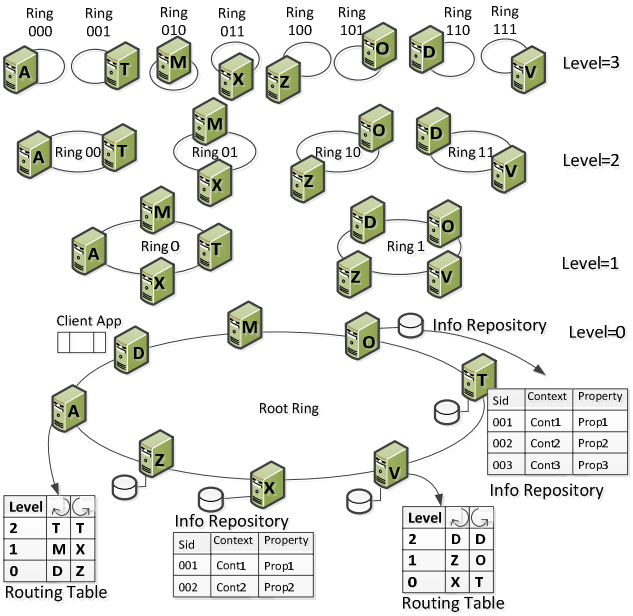

Fig. 2. The proposed service discovery and overlay routing infrastructure

overlay peers are divided into two disjointed rings. Similarly, nodes at level *2* form four disjointed rings, and for forth. In summary, rings at level *h+1* are obtained by splitting a ring at level *h* into two disjoint sets with each ring containing every second member of the level *h* ring.

Fig. 2 shows the high-level architecture of LOCA's communication overlay. Peers offered and maintained by respective service providers form a SkipNet overlay topology. Every peer in the discovery overlay provides an information repository. When new events/services are generated in the network, they are stored in the local information repository. Each peer exposes a set of discovery APIs that can be invoked by client applications to search for desirable services. At the same time, each node also maintains a routing table to enable efficient query routing. The discovery services returns the URIs of the repositories that handle information and services about those objects.

SkipNet supports two message routing algorithms: string-based name ID routing and numeric ID routing. To route a message by name ID, the routing would follow pointers that are closest to the intended destination name ID. At each peer, a message will be routed along the highest level pointer that does not point past the destination value. Routing terminates when the message arrives at a peer whose name ID is closest to the destination. The principle of routing by numeric ID is to follow rings that route closest to the intended destination numeric ID. The routing starts from the root ring until a node is found whose numeric ID matches the destination numeric ID in the first digit. Then the routing goes up to this node's level *1* ring and examines peers in this ring until a node is found whose numeric ID

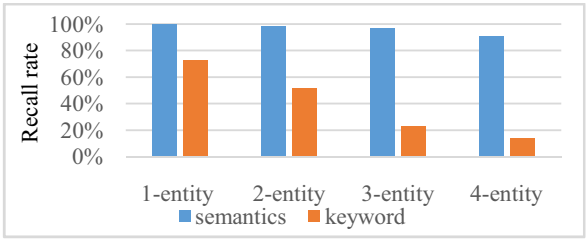

Fig. 3. Comparison of the recall rate of context matching based on semantics and keyword

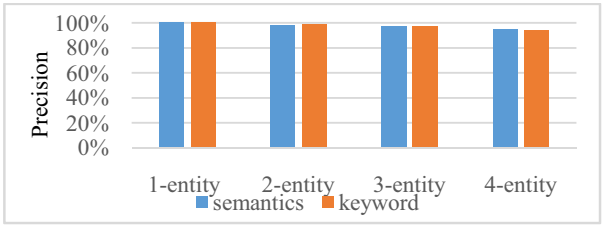

Fig. 4. Comparison of the precision of context matching based on semantics and keyword

matches the destination numeric ID in the second digit. The process repeats until routing cannot make any more progress at some level *h* such that none of the nodes in that ring share *h+1* digits with the destination numeric ID. The node whose numeric ID is numerically closest to the destination numeric ID in the ring at this level is the destination of the numeric ID routing.

Nodes in the overlay together provides a repository to store code resolution records. To publish a service record, the service provider will register the service metadata to the local information repository. The service metadata includes the service description, the context description represented with ontology, the identification of the related objects, and the URI of the service. We incorporate the provider's DNS domain name into the service registration code. For example, to register a service with identity *service0001* in the organization *sample.domain.org*, we name the service as *sample.domain.org/service0001*, which indicates that it is associated with and published by the provider "*sample.domain.org*". Then we can explicitly place data on specific overlay nodes or across nodes within an organization. Sub-organizations can be presented by adding their sub-organization names. For example, branch *subOrg1* in organization *sample.domain.org* can be represented as *sample.domain.org/subOrg1*. In operation, we reverse the components of the DNS name *sample.domain.org,* so that *sample.domain.org* becomes *org.domain.sample.* Consequently, all service within domain *sample.domain.org* share the prefix *sample.domain.org* in their name IDs. This yields path locality for organizations in which all services share a single DNS suffix (and hence share a single name ID prefix). To support SkipNet's Constraint Load Balancing (CLB), we divide a data object's name into two parts, namely the domain part over which load balancing should be performed and the name part that is used as input to the DHT's hash function.

To search for a data object that has been stored using CLB, we first search for any node within the CLB domain using search by name ID. To find the specific node within the domain that stores the data object, we perform search by numeric ID with the CLB domain for the hash of the CLB suffix. As peers are ordered by their name IDs along each ring of SkipNet and a routing message will not be forwarded past its destination. Therefore, all peers encountered during the routing process have name IDs between the source and the destination. If the source peer shares a common prefix with the destination peer, all peers traversed by the routing message will have name IDs that share the same prefix as well. This guarantees that message traffic routed between two overlay nodes within the same organization stays within it, i.e., routing path locality. Another nice property of our discovery system is it inherent support for range queries, as peers and data of SkipNet are stored in name ID order, data sharing common prefixes are stored over contiguous ring segments.

The property of content and path locality of LOCA's discovery structure is very important for IoTs. Content locality can explicitly specify where the metadata is placed; while the path locality guarantee that a routing message between any two peers within a single administrative domain always stays within that administrative domain's boundary. This isolation will effectively avoid malicious attacks by foreign machines outside the administrative domain, for example denial of service attacks, Sybil attack, eavesdropping, or traffic analysis.

V. Evaluation

In the first part of the experiments, we evaluate the effectiveness of our semantics-based context matching mechanism. In these experiments, we used a predefined context ontology containing 213 entities (23 classes, 6 properties and 184 instances). As illustrated in Fig. 1, under the root level, there are four first-level entities: *person*, *environment*, *device*, and *service*. Based on this ontology, we created services and their corresponding context instances by selecting and instantiating the context ontology. The distribution of services follow the Zipf distribution. Service query context is also specified using the same ontology. All service context are instantiated to the instance level, and the query context can be in either the class-level or the instance-level. We compare our semantics-based context matching with keyword-based context matching, which matches context only in the vocabulary level. The performances are compared in terms of recall and precision. The definition of recall and precisions are defined as follows:

$$recall = \frac{|\, relevantEntries \cap retrievedEntries \,|}{|\, relevantEntries \,|}$$

$$precision = \frac{|\, relevantEntries \cap retrievedEntries \,|}{|\, retrievedEntries \,|}$$

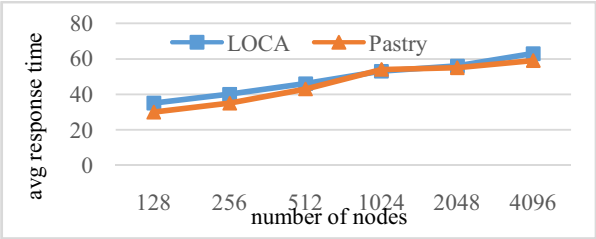

Fig. 5. Average query response time vs. network size

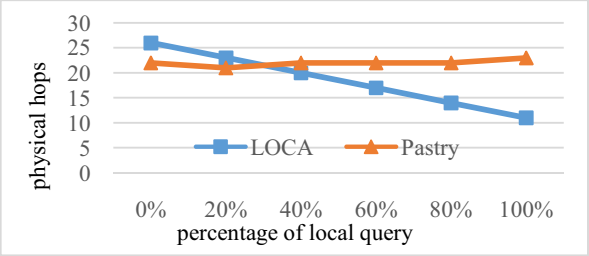

Fig. 6. Physical hops of discovery query routing vs. the percentage of forced local query

Fig. 3 shows the comparison of the recall rate of semantics-based context matching and keyword-based context matching. We vary the number of query context entity/criteria from 1-entity to 4-entity. We can see that semantics-based matching can dramatically improve the matching recall rate, by finding more context entities which are semantically related although literally may be irrelevant. Fig. 4 shows the comparison of precision of the two matching schemes. Again, we can see that using semantic match will not jeopardize the matching precision. Instead it may eliminate the semantic ambiguity problem such as polysemy and homonymy, thus improving the precision.

In order to evaluate LOCA's discovery performance, we implemented a massage-level, event-driven, discrete-time simulator. We did not model the queuing delay or packet loss in the network. The SkipNet overlay was implemented based on the source code from Microsoft Research [26]. Peers in the overlay network share a predefined context ontology. We generated the network topology using a popular topology generators GT-ITM [27]. In particular, we use GT-ITM to generate the Transit-Stub (TS) network model, which reproduces the hierarchical structure of the topology of the Internet. In this topology, a stub domain corresponds to an Autonomous System (AS) which carries only traffic that originates or terminates in the domain. While transit domains represent wide or metropolitan-area networks (WANs or MANs). For comparison, we also implemented a well-known DHT overlay Pastry [28] based on FreePastry [29]. The name of the peers in the network follows the format of organization-name/node-id, while the format of the service name uses the format of organization-name/node-id/service-id. Each experiment is performed ten times with different random initialization and the average value is presented.

Fig. 5 illustrates the response time as a function of the network size. In this figure, the time is measured in terms of simulation time units. We vary the number of peers in the network from 2^7 to 2^{12}. As can be seen from the figure, the response time using either of the overlay structure increases logarithmically as the growing network size. This demonstrates the good scalability of the LOCA overlay structure, as it scales well to a large network like a DHT network.

The most important property of LOCA discovery is its support of content and path locality. We performed a set of experiments to verify this property. Fig. 6 shows the result. In this experiment, we varied the percentage of local queries and examined the service discovery overhead in terms of physical hops traversed by the discovery query. Local queries are discovery queries searching for services which are provided by organizations where the query issuer belongs to. The network size in this experiment is 2^{12} and the number of organizations (ASs) is *25*. The query and the AS nodes follow a Zipf distribution. As illustrated in the figure, Pastry is insensitive to the query location and content location as their design ignores content locality. On the other hand, as the proportion of local query increases, LOCA's physical routing hops significantly reduce thanks to its locality preserving property.

VI. Conclusions

Service discovery is a very important task in IoTs. In this paper, we present LOCA, a context-aware and locality-preserving discovery framework. Like DHT-based discovery systems, LOCA is fully decentralized and robust. While unlike DHTs, LOCA can control where service and service metadata is placed. This improves the security and integrity of the discovery framework. To locate services based on context requirement, we proposed an ontological model to present and match context information. The proposed mechanism have been evaluated with simulations.

References

[1] Atzori, L., Lera, A., & Morabito, G. (2010). The internet of things: A survey. Computer Networks, 54(15), 2787–2805.

[2] Teixeira, T., Hachem, S., Issarny, V., & Georgantas, N. (2011). Service oriented middleware for the internet of things: a perspective. In Towards a Service-Based Internet (pp. 220-229). Springer Berlin Heidelberg.

[3] Mietz, R., Groppe, S., Römer, K., & Pfisterer, D. (2013). Semantic models for scalable search in the Internet of Things. Journal of Sensor and Actuator Networks, 2(2), 172-195.

[4] Guinard, D., Trifa, V., Karnouskos, S., Spiess, P., & Savio, D. (2010). Interacting with the soa-based internet of things: Discovery, query, selection, and on-demand provisioning of web services. Services Computing, IEEE Transactions on, 3(3), 223-235.

[5] Polk, T., & Turner, S. (2011). Security challenges for the internet of things. In Workshop on Interconnecting Smart Objects with the Internet.

[6] Shrestha, S., Kim, D. S., Lee, S., & Park, J. S. (2010). A peer-to-peer RFID resolution framework for supply chain network. In Future Networks, Second International Conference on (pp. 318-322). IEEE.

[7] Manzanares-Lopez, P., Muñoz-Gea, J. P., Malgosa-Sanahuja, J., & Sanchez-Aarnoutse, J. C. (2011). An efficient distributed discovery service for EPCglobal network in nested package scenarios. Journal of Network and Computer Applications, 34(3), 925-937.

[8] Harvey, N. J., Jones, M. B., Saroiu, S., Theimer, M., & Wolman, A. (2003). Skipnet: A scalable overlay network with practical locality properties. networks, 34, 38.

[9] Auto-ID Labs: http://autoidlabs.org/wordpress_website/

[10] EPCglobal: http://www.gs1.org/epcglobal

[11] Lorenz, M., Zeier, A., Plattner, H., Müller, J., & Schapranow, M. P. (2011). Discovery services in the EPC network. INTECH Open Access Publisher.

[12] Beier, S., Grandison, T., Kailing, K. & Rantzau, R. (2006). Discovery Services – Enabling RFID Traceability in EPCglobal Networks, Proc. of the 13th International Conference on Management of Data (COMAD).

[13] High Level Design for Discovery Services. BRIDGE project. (2007). Cambridge, A. U. & Research, S.

[14] Pasley, J. (2005). How BPEL and SOA are changing Web services development. IEEE Internet Computing, 9(3), 60-67.

[15] Chakraborty, D., Joshi, A., Yesha, Y., & Finin, T. (2006). Toward distributed service discovery in pervasive computing environments. Mobile Computing, IEEE Transactions on, 5(2), 97-112.

[16] Gao, Z. G., Yang, X. Z., Ma, T. Y., & Cai, S. B. (2004). RICFFP: an efficient service discovery protocol for MANETs. In Embedded and Ubiquitous Computing (pp. 786-795). Springer Berlin Heidelberg.

[17] Schoenemann, N., Fischbach, K., & Schoder, D. (2009). P2P architecture for ubiquitous supply chain systems.

[18] Paganelli, F., & Parlanti, D. (2012). A DHT-based discovery service for the Internet of Things. Journal of Computer Networks and Communications.

[19] Maymounkov, P., & Mazieres, D. (2002). Kademlia: A peer-to-peer information system based on the xor metric. In Peer-to-Peer Systems (pp. 53-65). Springer Berlin Heidelberg.

[20] Zhang, Y., Liu, L., Li, D., Liu, F., & Lu, X. (2009). DHT-based range query processing for web service discovery. In Web Services, 2009. ICWS 2009. IEEE International Conference on (pp. 477-484). IEEE.

[21] Perera, C., Zaslavsky, A., Christen, P., & Georgakopoulos, D. (2014). Context aware computing for the internet of things: A survey. Communications Surveys & Tutorials, IEEE, 16(1), 414-454.

[22] Chen, H., Finin, T., & Joshi, A. (2003). An ontology for context-aware pervasive computing environments. The Knowledge Engineering Review, 18(03), 197-207.

[23] Wang, X. H., Zhang, D. Q., Gu, T., & Pung, H. K. (2004). Ontology based context modeling and reasoning using OWL. In Pervasive Computing and Communications Workshops. Proceedings of the Second IEEE Annual Conference on (pp. 18-22).

[24] Strang, T., Linnhoff-Popien, C., & Frank, K. (2003). CoOL: A context ontology language to enable contextual interoperability. In Distributed applications and interoperable systems (pp. 236-247). Springer Berlin Heidelberg.

[25] Li, J., Wang, H., & Khan, S. U. (2012). A semantics-based approach to large-scale mobile social networking. Mobile Networks and Applications, 17(2), 192-205.

[26] SkipNet: http://research.microsoft.com/en-us/downloads/b7d19c09-87ba-426d-b2d2-4f978c76294e/

[27] GT-ITM: http://www.cc.gatech.edu/projects/gtitm/

[28] Rowstron, A., & Druschel, P. (2001). Pastry: Scalable, decentralized object location, and routing for large-scale peer-to-peer systems. In Middleware 2001 (pp. 329-350). Springer Berlin Heidelberg.

[29] FreePastry: http://www.freepastry.org/.