Secured Service Discovery Technique in IoT

Ali H. Ahmed, Nagwa M. Omar, and Hosny M. Ibrahim
Deptartment of Information Technology, Assiut University, Egypt
Email: ali.hussein@aun.edu.eg, n omar@aun.edu.eg, hibrahim@aun.edu.eg

Abstract —Service Discovery is a process of automatically finding appropriate services and their providers in IoT taking into consideration requests' context and QoS. In IoT, many heterogeneous objects offer different services so it is challenging to locate desirable services due to the considerable diversity and large scale. An algorithm for securely locating, delivering, and matching of services is a must for providers and consumers. The use of complex algorithms and powerful fixed infrastructure is infeasible due to the limited resources of most IoT devices. This paper introduces a secure and broker based service discovery technique that deliver services to consumers taking into consideration context and QoS in addition to provider trustworthiness. Service consumer submits an encrypted version for his query to a centralized broker, this broker manages a trust value to the connected objects besides to monitoring devices' QoS. Based on the services' queried QoS, context, and trust records, the broker matches consumer query to the most appropriate provider. Finally the broker distributes session key to the communicating entities to secure their further communications during service delivery.

Index Terms—IoT, Secure service discovery, QoS, trust, Fog Computing.

I. Introduction

IoT is a dynamic global network infrastructure with self-configuring capabilities based on standard and interoperable communication protocols. IoT definition is fuzzy due to the concepts and technologies it includes [1]. All the IoT definitions converge to the view that simple embedded sensor networking is now evolving to the much-needed standards and Internet enabled communication infrastructure between objects [2]–[8]. Designing an effective and efficient discovery mechanism must fulfill specific requirements. In this paper, we focus on a set of major requirements that aim to address the challenges standing out for service discovery in the IoT. The main requirements are security, lightweight, and trust. Additional requirements such as accuracy, performance, and scalability are also important and may be a subject for future work.

A. Secured Service Discovery

Security must be preserved in the communications occurring in all phases among devices and users or among devices. For example user request may be forged or even discarded, a response may be produced from a malicious device and don't actually reflects devices' real capabilities, accordingly, communications among entities must be secured. Attacks during IoT service discovery can be either passive or active. During the passive attacks an attacker may listen to the multicast traffic. In active attacks an attacker may ask for all presence service instances and identify the vulnerable versions and attack the corresponding hosts. He can also offer fake services to make someone connect.

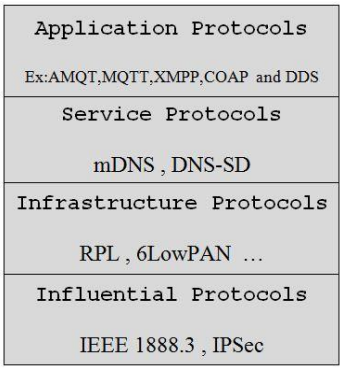

Fig. 1. Service discovery protocols inside the big picture.

B. Lightweight Service Discovery y

The enormous number of devices in IoT applications requires an efficient, dynamic and Zero Configurations technique for services discovery. Figure 1 shows the position of service discovery protocols in IoT protocol stack.

Multicast DNS (mDNS) and DNS Service Discovery (DNSSD) protocols have been designed for service discovery in resource- rich devices, there are research studies that produce light versions of them for IoT environments [11], [12]. The goal of the Zero Configuration Networking (Zeroconf) is to enable networking in the absence of administration. Zero configuration networking is required for environments where the administration is impractical. Essentially, to reduce network configuration to zero (or near zero) in Internet Protocol (IP) networks, it is necessary to perform specific operations without the existance of servers such as IP addresses distribution, name translation, and listing services without DHCP server, DNS server, and directory service respectively.

To fulfill lightweight service discovery in IoT, smart devices should join the network or leave it without affecting the behavior of the whole system. However mDNS and DNS-SD can smooth this way of development, the main drawback is the need for caching DNS entries especially when it comes to resource-constrained devices.

Manuscript received May 20, 2018; revised December 25, 2018.
doi:10.12720/jcm.14.1.40-46

C. Trust Based Service Discovery

A mechanism for matching consumer to provider must consider honest providers, so it must continuously manage trust values to IoT devices which provide services. Trust management still a hot topic of research. Recent work on trust focuses on either developing trust models [10] or selection and encoding trust metrics which accurately match user to honest providers. Research work on trust management is based on two main branches: direct observation or indirect recommendation for either social or QoS trust metrics.

In this paper a secured broker based, and trustworthy service discovery technique is introduced. The technique uses a broker software running on devices gateway to gain advantages of edge computing in off-loading IoT devices from implementing heavy weight security algorithms. The main tasks of a broker are:network initialization, accurately matching users to providers via implementing trust management model, and securing further communications between service provider and consumer via generating and distributing session keys. Service providers and consumers submit an encrypted Service Profile (SP) and Service Query (SQ) respectively. SP includes device IP, initial pr-configured trust value, QoS, service instance name, and domain ID. SQ contains requested service name, QoS and domain name. SP and SQ are represented in XML descriptors called TAGs. The TAGs are either encrypted or signed according to the security requirement for a service in specific context. Secured TAGs are transmitted during the discovery process. The power of encryption is utilized in securing SP , SQ TAGs, and service delivery. Finally a full diagram for the proposed technique is illustrated and performance evaluation experiment is held to outline proposed technique usability.

This paper is organized as follows: Section II provides an overview on secure service discovery protocols and trust management models in IoT also it examines related work and describes its main problems. Section III presents the proposed protocols' building blocks and their interactions. Section IV shows the experimental results for testing the proposed technique performance on Cooja Contiki simulator. Section V concludes this work.

II. Review

A. Service Discovery in IoT

Most of IoT architectures [13]–[18] include a service management or middleware layer that pairs a service with its requester based on addresses and names. This layer enables the IoT application programmers to work with heterogeneous objects without taking into consideration specific hardware platforms. Also, this layer processes received data, makes decisions, and delivers the required services over the network protocols [14], [16]. IoT services can be categorized into four types [18]:

1) Identity-related: Basic services that are used in other types of services.

2) Information Aggregation services that collect and summarize raw sensor data for further processing and reporting to the IoT application.

3) Collaborative-Aware: Services act on top of Information Aggregation Services and use the obtained data to make decisions.

4) Ubiquitous: Aims to make Collaborative-Aware services available anytime to anyone anywhere.

B. Service Discovery Protocols

The multicast DNS (mDNS) [19] and DNS Service Discovery (DNS-SD) [20] are two main protocols used in discovering services in IoT. Multicast DNS (mDNS) provides the ability to perform DNS-like operations on the local link without a Unicast DNS server. Three main benefits made mDNS best suited to IoT devices : (i) Zero configuration i.e. require little or no configuration during devices' setting up , (ii) Work when no infrastructure is present, and (iii) Work during failures.

DNS-SD has the zero configuration property and utilizes mDNS to send DNS packets to specific multicast addresses through UDP. DNS-SD allows clients to discover a list of named instances to the desired service, using standard DNS queries. Two steps are held during service discovery: finding host names of required services and pairing IP addresses with their host names using mDNS. The Pairing function multicasts network details like IP, and port number to each related host. The main disadvantages of these two protocols are the process of caching DNS entries especially in case of resource constrained devices. However, limiting the cache durtion for a specific interval and removing it after expiry can solve this issue but it may involve additional overhead and complexity.

A peer-to-peer (P2P) service discovery protocol that adopts the distributed hash table (DHT) techniques is introduced in [21]. This technique supports multi-attribute and range queries also it guarantee of scalability, robustness, and easy maintenance of the overall system.

In the discovery technique introduced in [22], IoT objects use P2P as a communication protocol. To identify services, Constrained Application Protocol (CoAP) based URIs are generated as they contains the resource paths and the names of the necessary service providers. The authors use a MAC address hashing technique to generate unique names for the endpoints. The main problem in this technique is that the MAC address can be spoofed which may lead to duplicate endpoints names.

A scalable and self-configurable P2P architecture for service discovery is reported in [23]. The architecture utilizes an IoT gateway which acts as a backbone for the SD architecture. The gateway also keeps track of objects joining or leaving the network and updates the list maintained at a CoAP server. Several gateways are interlinked through two P2P overlays namely distributed

local service (DLS) and distributed geographic table (DGT) to facilitate global service discovery.

The authors in [24] proposed an IoT framework that allows thing discovery in the smart home domain. The framework consists of three main layers. The proxy layer interfaces with physical objects to be discovered. The discovery layer allows objects registration through the configuration registry API. Only registered objects can be discovered by the consumer devices. Finally, the service enablement layer produces the functionalities of the discovery framework to the consumers using RESTful web services. The main drawback of that work is the lack of threat model for the utilized security techniques to sustain the authentication and access control in service discovery.

In [25] DNS Name Auto configuration (DNSNA) for global and IoT DNS names was introduced. The DNS names of IoT devices can be auto configured with the device's category and model in IPv6-based IoT environments. The DNS name lets user easily identify each IoT device for monitoring and remote-controlling in IoT environments. The work is considered an improvement to mDNS as it produced less traffic. DNSNA resolves an IoT device's DNS name into an IPv6 address in unicast through an authoritative DNS server. However this technique tunes mDNS for better QoS through reducing traffic, the literature didn't provide any security enhancements over mDNS.

Other research work in service discovery [26]–[28] merely focus on accurate matching between SP and SR in different application contexts such as smart cities. Specific techniques were utilized such as ontologies for context modeling and either centralized or decentralized where used interchangeably, however security issues were omitted.

Although decentralized service discovery fulfills the requirements of scalability, high performance, and zero-config, it has many drawbacks related to security and resources consumption.

C. Trust Management Models

The trustworthiness of services' providers must be known and validated to avoid malicious, poor quality, and time consuming providers. Trust and reputation mechanisms are needed to help service consumers to distinguish good services from bad ones. A varity of trust models already proposed for IoT such as social trust. The process of calculating the trust value as a result of direct observations, previous observations, and indirect recommendation is called trust assessment. Reputation, Experience, and Knowledge (REK) is a Trust Evaluation Model [30]. REK consists of the three trust indicators (TI) namely Reputation, Experience, and Knowledge. The Reputation and Experience TIs are obtained by incorporating previous interactions between entities over time. The Experience TI is personal perception of a trustor to a trustee. The Experience TI covers more about the trustors propensity portion of trust. The Knowledge TI represents direct observation of a trustor toward a Trustee by breaking down characteristics of the trustee respecting to a specific environment. Trust is affected by trustor's propensity and environment. The final trust value is obtained based on trustor observations and interactions between the two communication parties. Propensity includes both requirements for the trust goal and the trustors preferences about the trustees trustworthiness whereas the environment conditions are the considerations for some factors such as vulnerabilities, threats and risks. Fig. 2 illustrates the general model for Trust in IoT.

The following equation obtains the final trust value between two IoT objects say service provider and consumer, B, and A:

$$T(A,B) = \alpha\, Rep(B) + \beta\, Exp(A,B) + \gamma\, Kn(A,B) \quad (1)$$

where T(A,B) is the trust value of B which is perceived by A and α , β , and γ are three adjustable parameters and picked up from certain intervals. Digging through trust details outside the scope of this paper but they can found in [30].

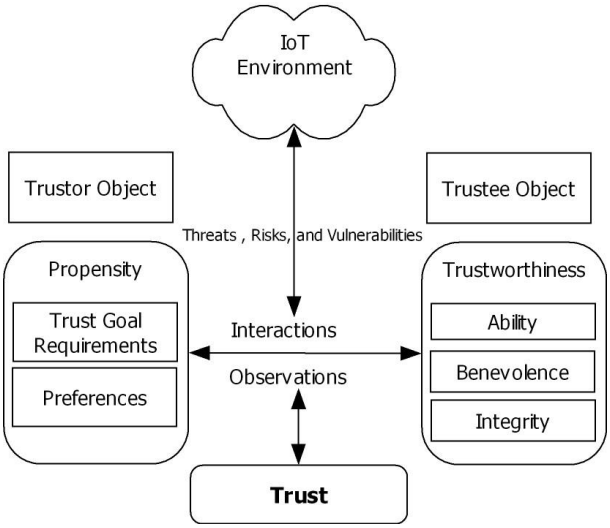

Fig. 2. A General Trust Model in IoT.

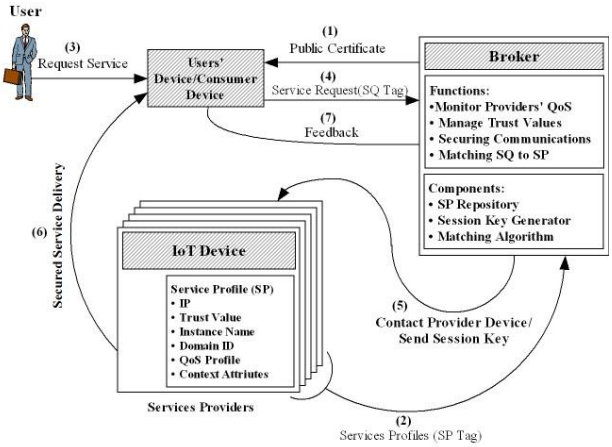

Fig. 3. The Proposed IoT service discovery block interaction

III. The Proposed SD Technique

The proposed discovery technique mainly focuses on securing service discovery while preserving the lightweight requirement and contacting only honest providers. Fig. 3 depicts the overall technique operation. The following subsections describes the operation of each block in the proposed technique.

A. Consumer/Provider IoT Devices

Service consumer interacts with provider devices to obtain specific service. The main users' concern is to obtain secured and as related response as could to his query. Provider devices are selected according to user query by the broker. Genetic algorithms and PSO are commonly used techniques during matching SQ to appropriate SP. SP and SQ TAGS contains QoS indicators and modeled service context. QoS have many aspects, for this application packet loss, throughput, transmission delay, and availability will be considered. User can specify either actual values or a scale to these aspects. Null value in SQ TAGS means that any level of such aspect is accepted. The service context refers to the queried services' characteristics. Representing a context in a computer readable format is referred to modeling. Several modeling techniques exist such as ontology, key-value, object, graphical, markup, and logic based. During this technique both TAGs are modeled as key-value pairs. Fig. 4 shows examples of SP and SQ TAGS in XML format. Provider devices are queried to obtain specific service. If providers and consumers are in the same domain, then they have one broker logic running in their gateway, otherwise they can communicate through several brokers.

Service Query

```
<?xml version="1.0"?>
<CameraSensor>
   <ip>172.31.51.86</ip>
   <context>
      <color>yes</color>
      <resolution>"400x300"</resolution>
      <rate>"20fps"</rate>
      <compression>NULL</compression>
      <audio>yes</audio>
  </context>
  <QoS>
      <maxLatency >"150ms"</maxLatency>
      <maxJitter >"50ms"</maxJitter >
  </QoS>
</CameraSensor>
```

Service Profile

```
<?xml version="1.0"?>
<CameraSensor>
   <ip>172.31.51.26</ip>
   <domain>aun.edu.eg</domain>
   <instance>camera1</instance>
   <trust>0.8</trust>
   <context>
     <color>yes</color>
     <resolutions>
      <resolution>"400x300"</resolution>
      <resolution>"800x600"</resolution>
      <resolution>"1024x768"</resolution>
     </resolutions>
     <rates>
      <rate>"20fps"</rate>
      <rate>"25fps"</rate>
      <rate>"30fps"</rate>
     </rates>
      <compression>"avi-asf"</compression>
      <audio>yes</audio>
  </context>
  <QoS>
      <maxLatency >"200ms"</maxLatency>
      <maxJitter >"40ms"</maxJitter >
  </QoS>
</CameraSensor>
```

Fig. 4. Service Query and profile examples.

B. Service Broker

Service broker is the main component in the proposed technique. A broker is a software installed on devices' gateway that have the following main functions:

1) Certificate distribution on IoT devices during the initialization phase.
2) Keeping a record of services profiles (SP TAGs).
3) Receive and decrypt SQ TAGS from service consumers.
4) Matching SQ TAG to suitable SP TAG.
5) Contacting providers and consumers to begin secure service delivery.
6) Managing and updating SP TAGs' trust/QoS values according to consumers' feedback.

During setup the broker distributes a certificate that contains his public key through multicast or broadcast message to devices attached on the gateway. IoT devices after receiving the certificate, encrypt their SP TAGs and send it back to the broker. The received SP TAGs are decrypted and stored in brokers' database. If a user requests specific service, the request is converted to SQ TAG via his application logic. This SQ TAG is encrypted and sent to the broker who in turn finds the best related SP to it. The matching delay may be time consuming according to the used algorithm power. The broker sends packet contains session key to both parties (users' device and the matched service provider) from that moment they can start secure service delivery. The sequence diagram in Fig. 5 summarizes the communications between a broker, consumer, and provider.

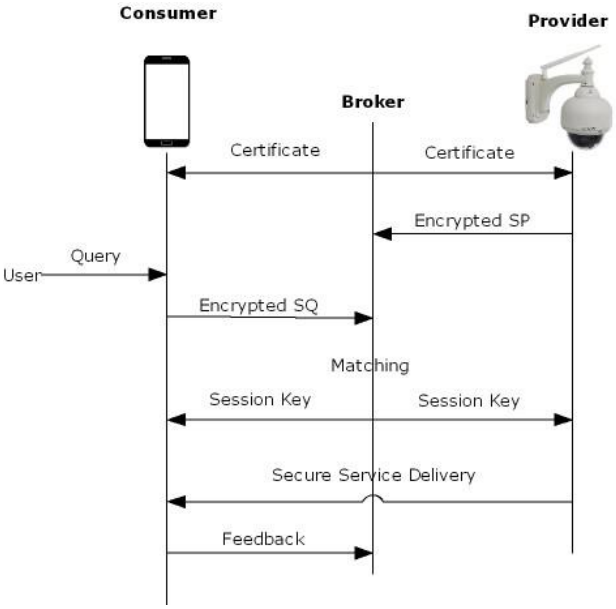

Fig. 5. Secure service delivery sequence diagram.

Security in the proposed technique is fulfilled as follows: The distribution of certificates and encrypting SQ TAGS keeps out intruders. The use of session keys in either message encryption or signing during service delivery ensures the integrity and confidentiality of information. Authentication is also preserved between the communicating parties through including the device ID in the session keys. Service consumer should submit a feedback to the broker about the provider, this is used to update providers' SP TAG.

IV. Results

The proposed techniques' goal is to secure the discovery of processes and taking into consideration the requirements of lightweight and trust. Encrypting SQ hinders request disclosure and interception attacks as

here is no usual technique for compromising public key cryptography, the same procedure goes regarding to further communications between consumer and provider. The main ingredients of a certificates and session keys are the IDs of the communicating entities and a timestamps when which an expiry interval is computed. The main benefit from including the IDs and timestamps is to ban the masquerading and replay attacks respectively. Finally the DoS/DDoS attacks have lower frequency especially if they are targeting to a resources rich devices such as brokers on devices gateway. Table I lists common types of attacks and their ability to compromise the system.

TABLE I: LIST OF COMMON ATTACKS AND THEIR ABILITY TO COMPROMISE THE PROPOSED TECHNIQUE

Attack	Ability to Compromise System	Justification
Client request disclosure	No	Client SQ is encrypted by Brokers' Certificate
Interception of request	No	Client SQ is encrypted by Brokers' Certificate
Message modification	No	Easily detected by receiver as all messages are encrypted
Disclosure/Service listing	No	Encrypted SP, SQ, and service delivery with session keys
Replay	No	Session Keys can't used twice
Masquerading	No	Devices' Identity are included and encrypted in the session Keys
DoS/DDoS	No	Brokers' logic runs on the devices gateway with rich resources

A. Proposed Technique Performance Evaluation

The proposed technique is tested on Cooja simulator with Contiki OS. The Collect-View tool is used to monitor various parameters from IoT devices. Table II lists the experiment parameters.

The proposed security technique is written in C and loaded into the sky motes in the experiment.

TABLE II: SIMULATION PARAMETERS

Parameter	Value
Radio Type	UDGM: Distance Loss
Number of Motes	8 + Broker
Mote Placement	Random
Mote Type	Sky
Collect-view Parameters	
Report Interval	60 Seconds
Report Randomness	60 Seconds
Hop-by-Hop Retransmission	31
Number of Reports	0 - Means Report forever

Fig. 6, shows the network topology which have device with ID 1 acts as a broker, other devices are service providers and consumers. The sensor map in the right side begins to build in the collect-view after 60 seconds.

The average delay of the mDNS and the proposed technique is depicted in Fig. 7.

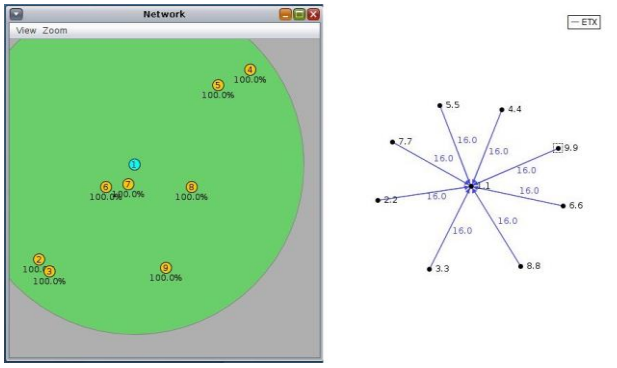

Fig. 6. Experiments' network topology and their corresponding sensor communication map

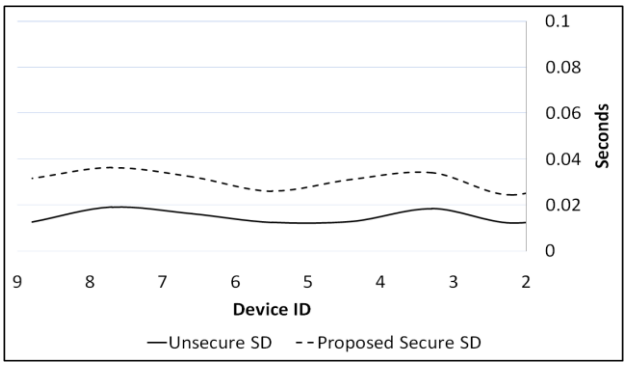

Fig. 7. Average delay for secured and unsecured SD.

The proposed technique incurs more delay during encrypting and decrypting packets. The net increase in delay is omitted if the security gained from proposed technique is taken into consideration.

TABLE III: AVERAGE POWER CONSUMPTION FOR UNSECURED SD AND PROPOSED SECURE SD

Power Consuming Item	Secured SD	Unsecured SD
LPM	0.15 mW	0.15 mW
CPU	0.365 mW	0.34375 mW
Radio Listen	0.38375 mW	0.38375 mW
Radio Transmit	0.01125 mW	0.01125 mW

For the data in Table III it is noticed that the power consumed Low Power Mode (LPM), radio listen, radio transmit is same for both secured and unsecured SD. However the proposed technique incurs more CPU cycles during encryption/decryption and hence more power is consumed.

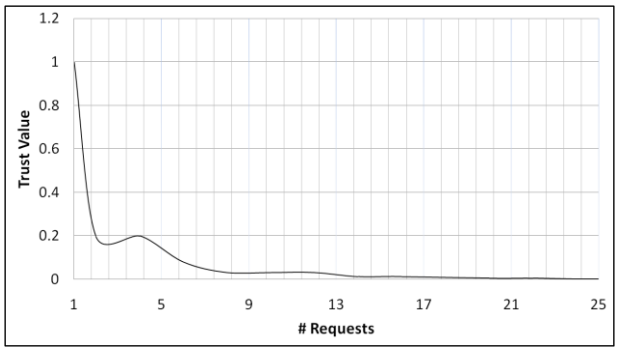

Fig. 8. Routing metric over time.

This last experiment in Fig. 8 is held to show the change in a malicious devices' trust values versus the number of request by different users directed to its service. A malicious node submits to the broker inaccurate SP tag with either fake QoS or service context.

From Fig. 8, the number of requests to the malicious providers' announced service increases and the providers' average-trust value decreases. This issue limits future assignments of queries to this malicious provider. Also a threshold may be configured at the broker to exclude providers below certain trust value from the matching process.

The lightweight requirement is achieved in the proposed technique through utilizing the service broker in managing security, communications, and QoS management in behalf of the IoT devices. This broker mainly resides on the devices' gateway which is assumed to be resource rich. During the normal operation of the IoT devices, the broker doing almost everything regarding security, trust management, and QoS monitoring, this is agreed in edge/fog computing concepts to offload resources' poor IoT devices.

V. Conclusions

IoT technology is rapidly invading our modern life to improve its quality. Secure and powerful service discovery is must to enable IoT in many application areas. The proposed techniques used broker to offload devices and manage further devices' communication. The broker continuously monitors and re-evaluates trust and QoS through feedbacks. Public key cryptography is utilized during SP and SQ submission to broker, whether symmetric key cryptography is utilized during service delivery. This makes the technique less vulnerable to common service discovery attacks. Simulation results prove that the proposed technique doesn't waste devices' resources and have negligible delay and power increase.

References

[1] M. R. Palattella, N. Accettura, and X. Vilajosana, "Standardized protocol stack for the internet of (important) things," *IEEE Communications Surveys & Tutorials*, vol. 15, no. 3, pp. 1389-1406, 2013.

[2] M. Zorzi, A. Gluhak, S. Lange, and A. Bassi, "From todayss INTRAnet of things to a future INTERnet of things: A wireless- and mobility related view," *IEEE Wireless Commun*., vol. 17, no. 6, pp. 44-51, December 2010.

[3] L. Coetzee and J. Eksteen, "The internet of things - promise for the future?" in *Proc. An Introduction, in IST-Africa Conference Proceedings*, May 2011.

[4] E. Fleisch, What is the Internet of Things? - An Economic Perspective, Auto-ID Labs, Tech. Rep., 2010.

[5] European Research Cluster on Internet of Things (IERC), Internet of Things - Pan European Research and Innovation Vision, IERC. [Online]. Available: http://www.internet-of-thingsresearch.eu/documents.htm, October 2011.

[6] L. Mainetti, L. Patrono, and A. Vilei, "Evolution of wireless sensor networks towards the internet of things: A survey," in *Prco. 19th International Conference on Software, Telecommunications and Computer Networks*, September 2011.

[7] J. P. Vasseur and A. Dunkels, "Interconnecting smart objects with IP: The next internet," *Morgan Kaufmann*, 2010.

[8] O. Hersent, D. Boswarthick, and O. Elloumi, *The Internet of Things: Key Applications and Protocols*, Wiley, 2012.

[9] Z. Sheng, S. Yang, Y. Yu, A. Vasilakos, J. McCann, and K. Leung, "A survey on the IETF protocol suite for the internet of things: Standards, challenges, and opportunities," *IEEE Wireless Commun*., vol. 20, no. 6, pp. 9198, Dec. 2013.

[10] U. Jayasinghe, A. Otebolaku, T. W. Um, and G. M. Lee, "Data centric trust evaluation and prediction framework for IOT," in *Proc. ITU Kaleidoscope: Challenges for a Data-Driven Society (ITU K)*, Nanjing, 2017, pp. 1-7.

[11] A. J. Jara, P. Martinez-Julia, and A. Skarmeta, "Light-weight multicast DNS and DNS-SD (lmDNS-SD): IPv6-based resource and service discovery for the web of things," in *Proc. 6th Int. Conf. IMIS Ubiquitous Comput*., 2012, pp. 731738.

[12] R. Klauck and M. Kirsche, "Chatty things Making the Internet of Things readily usable for the masses with XMPP," in *Proc. 8th Int. Conf. Collaborate Com*, 2012, pp. 6069.

[13] L. Atzori, A. Iera, and G. Morabito, "The Internet of Things: A survey," *Comput. Netw*., vol. 54, no. 15, pp. 2787-2805, Oct. 2010.

[14] R. Khan, S. U. Khan, R. Zaheer, and S. Khan, "Future Internet: The Internet of Things architecture, possible applications and key challenges," in *Proc. 10th Int. Conf. FIT*, 2012, pp. 257-260.

[15] Z. Yang, *et al*., "Study and application on the architecture and key technologies for IOT," in *Proc. ICMT*, 2011, pp. 747-751.

[16] M. Wu, T. J. Lu, F. Y. Ling, J. Sun, and H. Y. Du, "Research on the architecture of internet of things," in *Proc. 3rd ICACTE*, 2010.

[17] L. Tan and N. Wang, "Future internet: The internet of things," in *Proc. 3rd ICACTE*, 2010

[18] M. A. Chaqfeh and N. Mohamed, "Challenges in middleware solutions for the internet of things," in *Proc. Int. Conf. CTS*, 2012, pp. 21-26.

[19] S. Cheshire and M. Krochmal, Multicast DNS, Internet Eng. Task Force (IETF), Fremont, CA, USA, Request for Comments: 6762, 2013.

[20] M. Krochmal and S. Cheshire, DNS-based Service Discovery, Internet Eng. Task Force (IETF), Fremont, CA, USA, Request for Comments: 6763, 2013.

[21] F. Paganelli and D. Parlanti, "A DHT-based discovery service for the Internet of Things," *Journal of Computer Networks and Communications*, 2012.

[22] M. Liu, T. Leppanen, E. Harjula, Z. Ou, M. Ylianttila, and T. Ojala, "Distributed resource discovery in the machine-to-machine applications," in *Proc. IEEE 10th International Conference on Mobile Ad-Hoc and Sensor Systems (MASS)*, Oct. 2013, pp. 411-412.

[23] S. Cirani, L. Davoli, G. Ferrari, R. Leone, P. Medagliani, M. Picone, and L. Veltri, "A scalable and self-configuring architecture for service discovery in the internet of things," *Internet of Things Journal*, vol. 1, no. 5, pp. 508-521, Oct. 2014.

[24] S. K. Datta, "Towards securing discovery services in Internet of Things," in *Proc. IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, 2016, pp. 506-507.

[25] S. Lee, J. P. Jeong and J. S. Park, "DNSNA: DNS name auto configuration for Internet of Things devices," in *Proc. 18th International Conference on Advanced communication Technology* , Pyeongchang, 2016, p. 1.

[26] B. Jia, W. Li, and T. Zhou, "A centralized service discovery algorithm via multi-stage semantic service matching in internet of things," in *Proc. IEEE International Conference on Computational Science and Engineering (CSE) and IEEE International Conference on Embedded and Ubiquitous Computing*, Guangzhou, 2017, pp. 422-427.

[27] E. Wang and R. Chow, "What can i do here? IoT service discovery in smart cities," in *Proc. IEEE International Conference on Pervasive Computing and Communication Workshops (PerCom Workshops)*, Sydney, NSW, 2016, pp. 1-6.

[28] J. Bao and J. Xia, "A hybrid algorithm for service matchmaking based on ontology approach," in *Proc. IEEE 2nd Advanced Information Technology, Electronic and Automation Control Conference*, Chongqing, 2017, pp. 2420-2424.

[29] V. Suryani, Selo, and Widyawan, "A survey on trust in Internet of Things," in *Proc. 8th International Conference on Information Technology and Electrical Engineering*, Yogyakarta, 2016, pp. 1-6.

[30] N. B. Truong, T. W. Um, B. Zhou, and G. M. Lee, "From personal experience to global reputation for trust evaluation in the social internet of things," in *Proc. IEEE Global Communication (GLOBECOM) at Singapore*, December 2017.

