International Journal on Advanced Science Engineering Information Technology

Vol.13 (2023) No. 2
ISSN: 2088-5334

Hybrid Centralized Peer to Peer Architecture for Resource Discovery and Secure Communication in Internet of Things

Lokesh B. Bhajantri [a], Gangadharaiah S [b,*]

[a] Department of Information Science and Engineering, Basaveshwar Engineering College, Bagalkot, 587103, India
[b] Department of Computer Science, S.R.S.M.N. Government First Grade College, Barkur, Karnataka, 576210, India
Corresponding author: *gangadhar.s@gmail.com

***Abstract*— The Internet of Things (IoT) made communication between people and objects easy. It helps to build smart cities, homes, manufacturing systems, health monitoring systems, etc., for mankind. The increased adoption of IoT applications enabled many smart devices on the Internet platform. These devices deployed across the globe may have varying computational and communication capabilities. It is a great challenge to manage IoT resources efficiently. Some of well-known protocols are defined to identify and access IoT resources locally in a real-world environment. Many authors have adopted the Distributed Hash Table (D.H.T.) based Peer to Peer (P2P) model for global and massive resource management. However, D.H.T. based solutions have many shortcomings and are not perfectly suitable for the IoT domain. In this paper, it has been proposed a novel Hybrid Centralized Peer to Peer (HCP2P) architecture for efficient resource discovery and access mechanism. The proposed solution builds a secure communication channel among trusted peer devices with the aid of an HCP2P server. The trusted devices can discover and access the required resource efficiently and securely with reduced load on the central server. The proposed HCP2P solutions are evaluated on both hardware prototypes and simulations. The proposed model gives almost constant resource registration, discovery, and access time. This evaluation showed that HCP2P architecture performance is superior to traditional DHT-based P2P architecture. Finally, the performance parameters of the proposed scheme are evaluated in terms of resource registration time, discovery time, and hop-count.**

***Keywords*— Internet of Things; distributed hash table; device registration; resource discovery; resource access.**

Manuscript received 8 Jan. 2022; revised 12 Sep. 2022; accepted 27 Dec. 2022. Date of publication 30 Apr. 2023.

I. INTRODUCTION

The Internet of Things (IoT) consists of billions of heterogeneous devices connected to global infrastructure to provide advanced services to users in a real-world environment. Many applications are already deployed and provide invaluable services to mankind, for example, patient monitoring systems, environmental monitoring systems, smart energy metering systems, intelligent transport systems, etc. [1] [2]. These services are expected to run on devices with limited CPU processing power, transmission, and storage capacity. For uninterrupted services, the IoT ecosystem should incorporate a scalable, robust, and autonomous resource management system [3].

The resource management process in IoT applications like smart cities is tedious. Resource management includes resource modeling, resource discovery, resource allocation, and resource maintenance. Many existing solutions are developed for small-scale IoT resource deployment [4]. However, for large scale deployment of resources, most of the authors have adopted a D.H.T. based P2P system. In P2P architecture, network devices create an overlay network on top of a standard I.P. network and can handle both requests and responses, as shown in Fig.1.

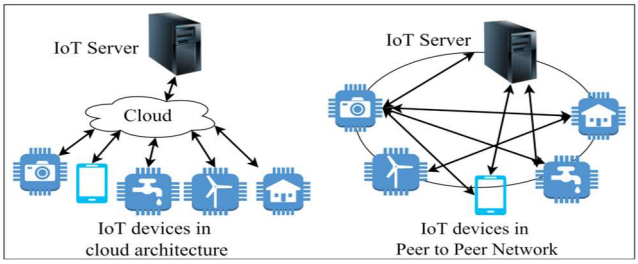

Fig. 1 IoT Device Cloud and Peer-to-Peer Architecture

This is more advantageous than the client and server model, where the client and server are differentiated, and only the server will respond to the client's request. The Distributed Hash Table (D.H.T.) based P2P network works on a similar

principle of the P2P network, but it uses the hashing technique to identify the resources in the network. IoT applications built on D.H.T. based P2P network generally run the D.H.T. peer node instance at the gateway. The D.H.T. instances from the peer-to-peer network for identifying and sharing the resources within its networks. There are many downsides while adopting D.H.T. based P2P system, mainly trust and cooperation of participating nodes and challenges in handling group and range queries which are outlined in the later part of this section.

Resource discovery focuses on identifying the resources, their capabilities, properties, and access control mechanism [5]. Based on the location of the device (i.e., local or remote), different naming services, registration, de-registration, and defined address assignment methods are developed. Many existing solutions for IoT systems are mostly adopted from the standard Internet Protocol (I.P.). The DNS service discovery protocol (DNS-SD) protocol is proposed for IoT that uses standard DNS interfaces, packet structure, and servers for resource discovery [6]. A distributed version of it called Multicast DNS (mDNS) is developed as a resource discovery protocol for the local network. In the mDNS protocol, the source multicasts the query in the local network. The device with a matching name will respond to the query by multicasting its I.P. address. Due to its multicast mechanism, it generates a lot of packets that make it unsuitable for IoT networks.

Service Location Protocol (S.L.P.) is a service discovery protocol used to find the services in the network. It has three agents for operation like User Agent (UA), Service Agent (SA), and Directory Agent (DA). The S.L.P. protocol extensively uses multicast U.D.P. packets for its operation, making it unsuitable for IoT applications. The Universal Plug and Play (UPnP) supports the zero-configuration concept based on standard Internet Protocol [7]. However, it has security issues like any device which supports UPnP assumes the surrounding network as trusted, which is risky in wireless sensor based IoT networks. CoAP and MQTT protocols are widely accepted in the IoT domain [8] [9]. A new MQTT-SN has been developed, which runs over U.D.P. protocol. This will provide resource discovery through wild cards. The CoAP has an interface similar to HTTP but runs over the U.D.P. protocol. It is based on the client-server model, where each sensor acts as a server and the application acts as a client. The resource discovery is provided at the gateway by standard link format. (/. well-known/core).

Compared to the client-server model [10] [11], P2P network architecture provides many advantages, as follows:

- High scalability.
- Avoids a single point of failure.
- Minimal administration.
- Effective utilization of network edge devices.
- Supports internetworking of heterogeneous systems.
- Fault-tolerant.
- Autonomous and self-organizing.
- Supports dynamic network (devices can join and leave at any time)
- High Reliability.

Many file-sharing systems (Napster, Kazaa, Gnutella), multiplayer games (Unreal Tournament, DOOM), and collaborative applications (ICQ) have adopted P2P architecture. The P2P based design is widely adapted in mobile ad-hoc networks, wireless mesh networks, wireless sensor networks, IoT, etc.

The work done by Goudarzi, Rahmani, and Mosleh [12] investigates the different resource discovery techniques, classification, and challenges in the IoT ecosystem. It also discusses the distributed architecture of resource discovery and important works carried out in D.H.T. based P2P architecture in the IoT ecosystem. The work given by Achir et al. [13] proposes a detailed survey on service discovery and selection in IoT. Also, it discusses the taxonomy of various approaches for service discovery and selection of resources in IoT. Finally, it is described the challenges and future research directions.

The global resource discovery based on a Pastry-like D.H.T. called X.M.H.T. (eXtendible Meta Hash Table) is proposed to solve the scalability issue [14]. The proposed work explains resource registration and inter and intra-domain resource discovery. But the proposed architecture has no scope for differentiating public and private resources. Also, the proposed design expects every local resource to be registered under global D.H.T. (X.M.H.T. peer), even if it is strictly utilized under the local domain. The work done in Li [15] proposes a new method for generating Node ID and Resource ID for D.H.T. based overlay networks. The Node ID number is generated based on the geo location of D.H.T. node, I.P. address, and U.D.P. Port number. This work fails to explain the steps involved in node registration and also does not clearly discuss how local and global resource discovery queries are handled.

The work given by Murturi et al. [16] depicts a new resource discovery mechanism using metadata. The resources are classified as private and public resources. The public resources from any local network are consolidated into a single file name. This file is copied to neighboring edge devices. The main disadvantages of the proposed design are overhead in maintaining a valid copy of metadata across the multiple edge nodes, frequent entry and exit of public resources/devices in the wireless network creating huge load on edge devices, global resource discovery is not possible as edges contain only information about neighbor nodes. The work done by Kamel, Crispo, and Ligeti [17] proposes DHT-based overlay network resource discovery. The network resources are classified as public and private resources. The public resources are shared across the network, whereas accessing private resources requires the key. But the proposed work is unclear about D.H.T. key generation for resources having similar attributes.

The agent based IoT service discovery [18] is proposed to improvise the system's energy efficiency. The gateway acts as an intermediate node between server and client devices and helps offloads the task. Each device in the network is expected to register itself to the central server, but this process is cumbersome for the local resources accessed in the same network. In work by Kamel et al. [19], the author proposes a secure resource discovery algorithm called Cipher Policy Attributed Based Encryption (CP-ABE). In this algorithm, the registration process requires resources to list the attributes of the clients for which resources should be hidden since the access verification is done only for non-cooperative clients and also uses the computational resources efficiently. The

work done by Kamet et al. [20] proposes a decentralized resource discovery and registration model which offload the computation work to multiple nodes. In this model, Region-based Distributed Hash Table (R.D.H.T.) is used for the physical location of peer nodes, and fine-grained attributes for clients are used for distributed resource discovery.

The work given by Mocanu et al. [21] presents a data fusion technique for Peer-to-Peer networks. The presented network considers the scenario of smart cities as a design of a spider overlay network environment. Also, it discusses the security aspects of the networks. Both chain and ring approaches are considered for data fusion in peer-to-peer overlay networks. It has also evaluated the efficiency of data over the peers in the networks.

The IoT resource discovery in Human Assistance and Disaster Recovery (H.A.D.R.) operations has been presented in previous studies [22] [23]. The first work proposes Programmable IoT Gateways (P.I.G.s), called SPF (Sieve, Process, and Forward) controllers for resource discovery. During H.A.D.R. operations, users forward the query to the SPF controller, which initiates the required process at P.I.G.s. The second work discusses the agent-based dynamic resource discovery, where separate agents are created for each protocol (HTTP, CoAP, and MQTT) at the gateway. Both works provide conceptual architecture. Many authors have proposed similar DHT-based P2P solutions in wireless sensor networks [24], MANETs [25], Wireless Adhoc Networks [26], and wireless mesh networks [27], [28].

Although DHT-based P2P solutions are highly scalable and fault-tolerant, there are many challenges [29] [30] associated as follows:

- There is no single designated authority that looks over DHT-based networks. Each participating node must cooperate and trust each other for smooth operation, which is a major challenge since trust in participating entities is unclear.
- Natively D.H.T. overlay networks do not support range and group queries. Few extensions supporting these features at every participating node make the system complex and less efficient.
- There is no absolute guarantee of data integrity and consistency, as there is no centralized authority for coordination.
- The D.H.T. network is based on a request-response model. It does not support events and triggers.
- Search time depends on the number of participating nodes and their location.
- As member nodes can join from any part of the world, handling confidential data like medical health records becomes a really challenging issue.
- It is difficult to build analytical data features in DHT-based systems as queries will be routed to different nodes in the P2P network [31]-[33].

This paper proposes a hybrid approach to overcome the shortcomings of DHT-based solutions. The proposed work describes the following contributions: a centralized approach is adopted for the resource registration and resource discovery phase. The secure peer-to-peer connection is established among nodes after obtaining metadata of required resources from the central server. In the resource discovery phase, the I.P. address of the gateway hosting IoT resource, the public key of the gateway, the device identification number, and the derived session key for a specific resource are obtained for peer-to-peer communication. In the proposed work, a centralized HCP2P server acts as a central authority and root of trust coordinating peer-to-peer communication. The proposed architecture adopts two levels of secure communication; at first, it adopts secure communication between gateways and later between end-to-end devices for information exchanges. The detailed working environment for the proposed model is explained in section 2. Results and discussions of the proposed work are discussed in section 3. Finally, it is concluded the proposed work in section 4.

II. Material and Method

This section discusses the need for the gateway-to-gateway communication, proposed system architecture, device registration, resource discovery, public resources access method with secure P2P communication, and algorithms for each.

A. Gateway to Gateway Communications

In the IoT network, sensor devices forward the information to the cloud server through its gateway, and interested clients can subscribe from the cloud server. Even though this data access sequence is common, it is less efficient because most of the cases and event producers and consumers are geographically nearby. The storing and accessing of information through the cloud increase the transmission delay and overloads the cloud server and network, specifically in smart city scenarios, which generate huge amounts of data around the clock. A peer can address this issue to peer network between gateways or devices for efficient resource access and reduced load on network elements and servers.

Further integration of peer devices may provide attractive services to users. For example, the city traffic signals can dynamically configure it signaling time based on vehicle arrival rate, emergency vehicle services (ambulance and fire extinguisher), and status of the nearest metro station or weather information from the meteorology department.

Another example is that a smart building infrastructure gateway can provide better services to its residents if it coordinates with city water management, sewage management, and emergency services. It can also communicate with the nearest residential apartment gateway to better utilize resources like a swimming pool, shared parking area, etc.

B. Proposed System Architecture

The proposed model assumes an IoT device and its operation as a resource because any operation on an IoT device consumes system resources. These resources are further modeled as local (private) and global (public) resources. The private and global resources are accessed in the local networks and outside the networks, respectively. For example, a fire alarm sensor's operation configuration is considered private and allowed to access within its own network whereas S.O.S. (Save Our Souls - a distress code to signal danger) events from it are considered public and shared across trusted peer devices. Establishing peer-to-peer communications between trusted gateways will help reduce network traffic and server load.

The illustration of the proposed architecture is shown in Fig.2, and the main stages are shown in Fig.3. This architecture comprises two IoT networks located at different geographical locations. Each network can have multiple smart devices registered to its respective home gateway (GWA or G.W.B.). The gateways GWA and G.W.B. are willing to be part of the P2P system that will register with HCP2P cloud server. The HCP2P cloud server acts as a coordinator or trusts anchor and stores each gateway's I.P. address, port number, and public key registered for the P2P network. The HCP2P server also stores the metadata of public resources that are accessed across the P2P network.

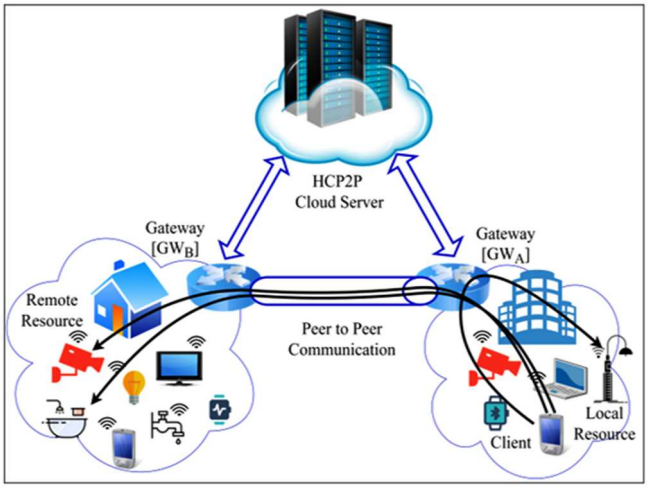

Fig. 2 The Proposed System Architecture

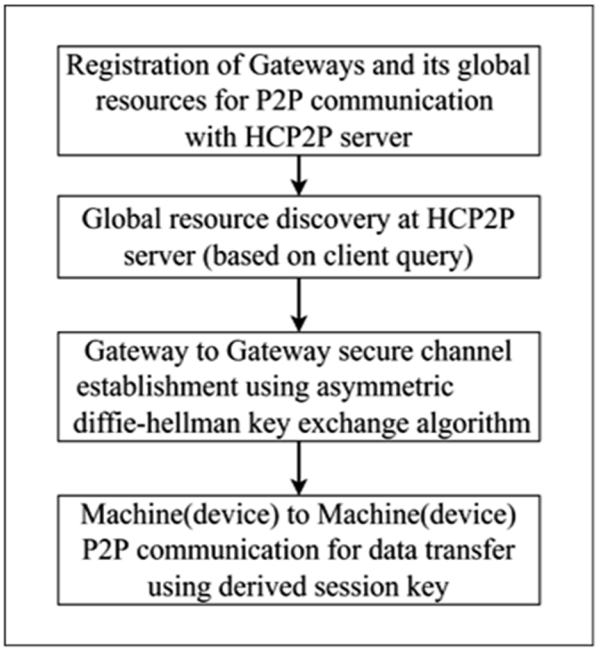

Fig. 3 Stages of Proposed System Architecture

The client sends requests to the HCP2P cloud server to access remote resources. The HCP2P server finds the appropriate global resource based on the client's request, profile, and different context awareness. The server forwards the remote device and gateway information to the client. For example, in Fig.2, a client requests data from a CCTV camera located in a foreign network through its gateway (GW_A). The communication between HCP2P server and gateway is implemented through a secure key derived from station-to-station protocol (variation of asymmetric Diffie-Hellman key exchange algorithm). Both gateways (GW_A and G.W.B.) establish secure channels for resource sharing with the help of HCP2P server.

After successful connection establishment, the client can access the remote resource (CCTV camera) using a session key derived from the pre-shared key of the device while registering with its gateway. This avoids sharing secret keys outside the network and helps achieve perfect forward secrecy. The detailed working of the resource registration, resource discovery, and resource access are explained in the following sections. Each IoT device will register with its gateway according to the device registration procedure explained in Fig.4. After successful registration. The client device can request local or global resources according to the resource discovery procedure, which is discussed in Fig.5. Once the resource is discovered, devices can exchange the information according to the procedure. A detailed description of the same is shown in Fig.8.

C. Resource Registration

Fig.4 explains the sequence diagram for the new IoT device registration process in the IoT environment. The pseudo-code for the same is described in an algorithm1. As discussed earlier, IoT devices can have resources that can be categorized as public and private. The private resources are registered only with the local gateway and can provide the services within its own local network, but public resource metadata is exported to the HCP2P server for remote access. The following steps explain the registration of a new IoT device in detail.

- To establish a P2P network, each gateway must register with a HCP2P server hosted on the cloud with its I.P. address, port number, public key, and geographical location.
- Gateway (G.W.) sends the registration request to the HCP2P server using a secret key (K), which is derived from the station-to-station protocol (variation of asymmetric Diffie-Hellman key exchange algorithm). This secret key (K) is shared only between a specific gateway and server.
- The HCP2P server updates the gateway information to its database and acknowledges with a gateway unique identification number (G.W.I.D.) encrypted using a shared secret key (K).
- Each IoT device registers with a nearby gateway (G.W.) using a Device Identification Number (D.E.V.I.D.), operations, and public and private resources. In this scenario, It is assumed that each IoT device shares the symmetric secret key (Ks) with the gateway, and communication with the local gateway is done through this pre-assigned secret key.
- Gateway acknowledges (A.C.K.) each IoT device after successful registration of an IoT device for P2P communication.
- An IoT device that is interested in sharing its public resources that will send the request to the HCP2P server through the gateway. The IoT device shares its public resource metadata and Access Control List (A.C.L.).

- After the successful registration of public resources, the HCP2P server acknowledges the IoT device.

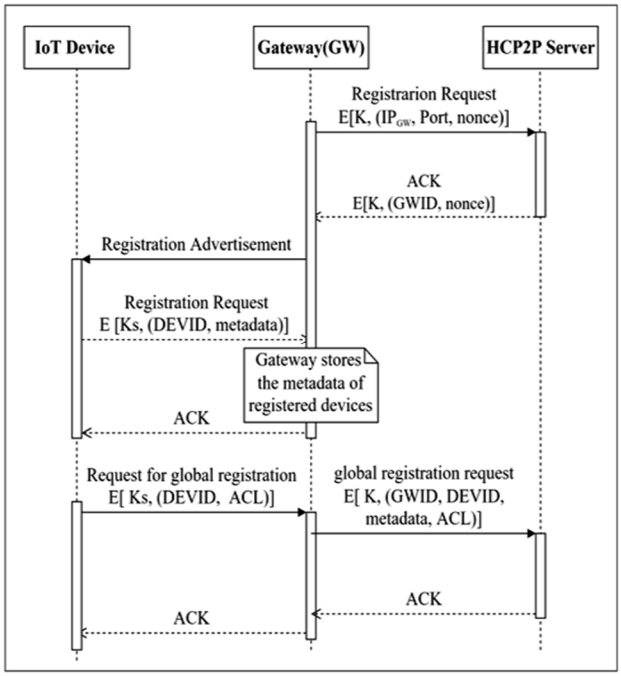

Fig. 4 Resource Registration

Algorithm 1: Resource registration

```
/* Every gateway register to HCP2P Server
securely using secret key K */

GWID=register_gateway_to_server(IPaddr, port)

/* IoT device can register with local gateway
*/
for (each_new_device)
{
    if(! Device_registered)
      {
        register_to_local_gateway(DEVID,
         preshared-key, list_of_operations);
      }
}

/*Each IoT device register its public resources
to server*/
    if(! registered)
       {
         register_to_server(DEVID,
         WID,metadata, ACL);
       }
```

D. Resource Discovery

Fig.5 outlines the procedure for the discovery of local and global resources, and the pseudo-code for the same is described in algorithm2. For further discussion, it is assumed that IoT devices and gateway (G.W.) are already registered according to the previous node registration procedure (Fig.4). The communication between IoT device (client) and gateway is through a pre-shared symmetric key (Ks).

The communication between the gateway and the HCP2P server is done through a secret key (K) derived using an asymmetric station-to-station algorithm. The following steps discuss local and global resource discovery sequences.

- The approach to accessing local resources within its gateway is straightforward. The IoT device (client) requests a list of local resources to its gateway.
- The gateway (G.W.) responds with the list of registered permitted local resources.

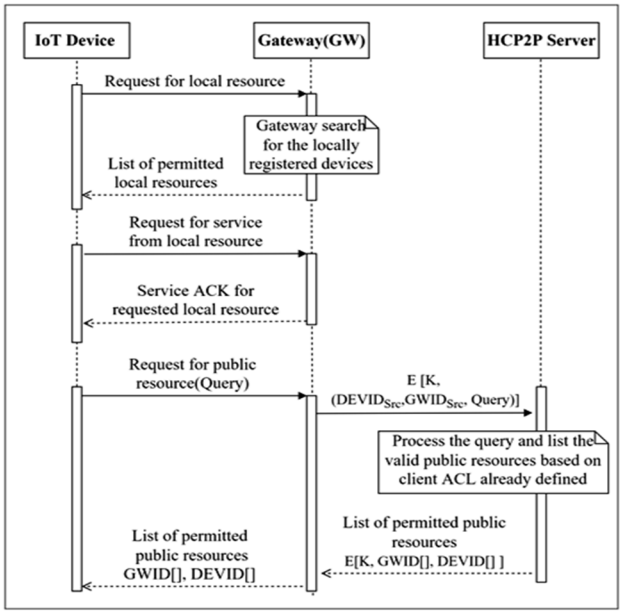

Fig. 5 Local and Global Resource Discovery

Algorithm 2: Resource Discovery

```
/* client request for local resources from
              its gateway */
List[] = request_resources_from_gateway();

 /* Select resource from local resource */
 for each ( List[i] = Required_Resource) {
   request_gateway_for_service(List[i]) ;
                    }

   /* Request for HCP2P server for global
resource located at remote network, return
      list of devices and gateway */
 {DEVID[],GWID[]}=requestGlobalRes(Query);

/* request for accessing public resource to
              remote gateway */
 request_public_resource(DEVID[i],GWID[i]);
```

- The client device can request service from any one of the devices in the list.
- Once the service request is handled, acknowledgment is sent to the client's device.
- If a client device requests a global resource, the HCP2P server processes the client request based on its profile, context awareness, location, and security policy and then forwards the matching device and their gateway information (D.E.V.I.D. [ ], G.W.I.D. [ ]) to the client.
- The client device can contact one or more remote IoT devices (D.E.V.I.D. [ ], G.W.I.D. [ ]) located in another network.

E. Accessing Public Resources with Secure P2P Communication

In this section, Fig.6 and Fig.7 outline the steps involved in the data exchange between client and remote IoT resources, whereas Fig.8 and algorithm 3 discuss the detailed steps.

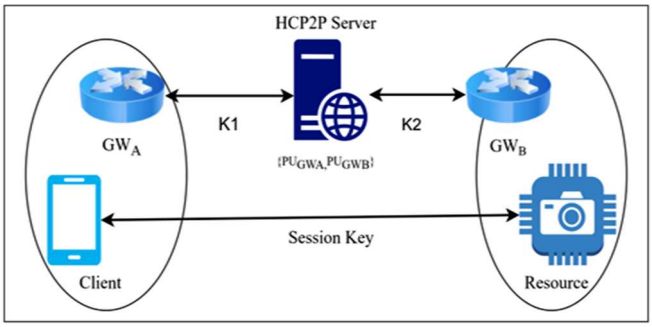

Fig. 6 Different Types of Keys used for Secure Channel

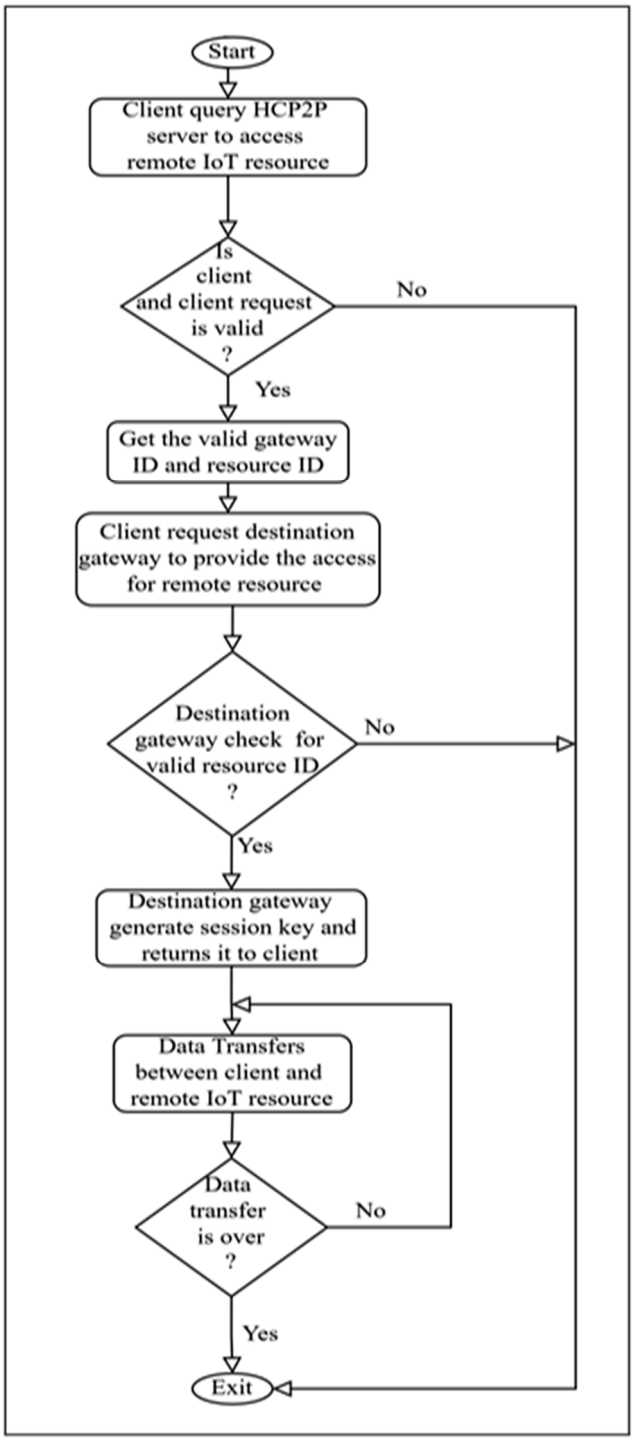

Fig. 7 Flowchart for Remote IoT Resource Access

To establish a secure channel between gateway GWA and HCP2P server, secret key K1 is used, while between gateway G.W.B. and HCP2P server K2 key is used. These K1 and K2 keys are generated by station-to-station protocol (variation of asymmetric Diffie-Hellman key exchange algorithm). Once the end-to-end secure channel is established between gateways, the destination gateway generates a session key, which will be used for data access from a remote client. This session key is used only for a specific session, which avoids sharing any device-specific key with remote devices.

Fig.8 shows the procedure followed by the client to access the information from the remote IoT resource server. The steps are as below.

- As a result of the resource discovery procedure discussed in Fig.5, the client has D.E.V.I.D. and G.W.I.D. of remote IoT devices.
- The client requests GW_A to establish a connection with $G.W._B$. The GW_A forwards the request to HCP2P server for I.P. address, port number, and public key (P.U.G.W.B.) of G.W.B. encrypted using K1 secret key which is shared only between the server and GWA.
- The server responds with I.P. address, port number and public key (P.U.G.W.B.) of G.W.B. encrypted using K1 secret key, which is shared only between the server and GWA.
- GWA generates a connection establishment request to G.W.B., which is encrypted using the public key of G.W.B. along with its identity and random number N.A.
- Once G.W.B. receives a connection request, it will contact HCP2P server for the public key of GWA which is encrypted using secret key K2, which is shared only between the server and G.W.B.
- The server responds with the public key of GWA (P.U.G.W.A.) which is encrypted using secret key K2, which is shared only between the server and GWB.
- GWB sends the acknowledgment for a connection request which includes the random number generated by G.W.B. (N.B.), the random number sent by GWA(N.A.), and its identity G.W.B.
- GWA confirms its identity to G.W.B. by sending back the random number of G.W.B. (N.B.) encrypted using the public key (P.U.G.W.B.) of G.W.B.
- Once the secure connection is established between two gateways, GWA requests for the access of remote IoT device (D.I.V.I.D.R.S., Resource Server).
- G.W.B. acknowledges GWA with a session key encrypted using P.U.G.W.A. to access the required IoT device. This session key will be derived from the pre-shared symmetric IoT device key. Every time a new session key is derived from the private key of specific IoT device. This helps perfect forward secrecy and avoids sharing the private key to unknown client.
- GWA communicates the session key to client. Using a session key, the client can access the IoT resource server.
- At the end of data transfer, client requests for connection termination and it is acknowledged by IoT device.

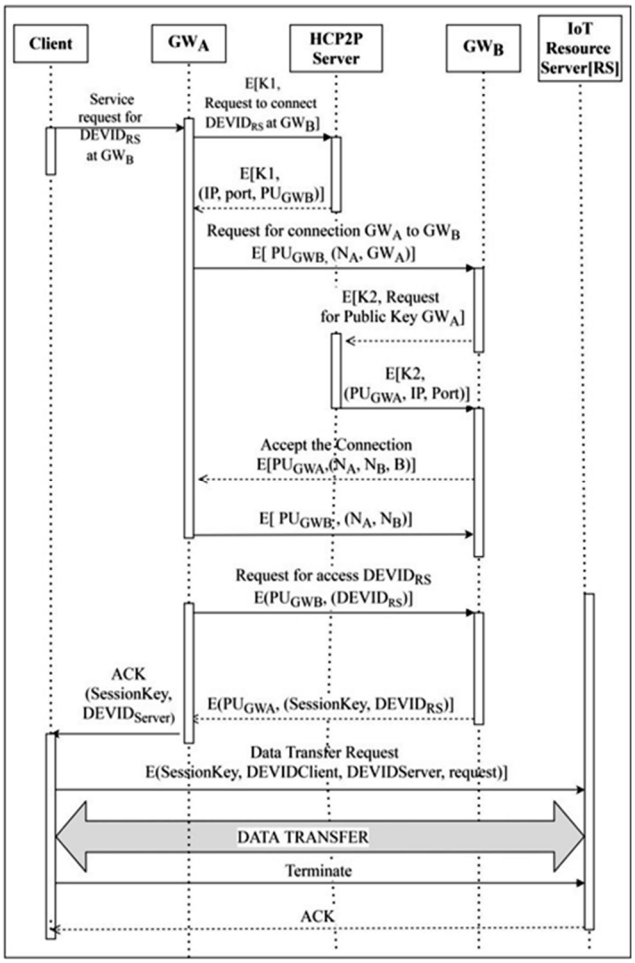

Fig. 8 Accessing Public Resource with Secure P2P Communication

Algorithm 3: Public Resource Access

```
Initialize: Client gateway has received DEVID
and GWID for remote IoT resource

/* Client request its gateway(GWA) to connect
remote IoT device(DEVIDRS)  */
request_connect(DEVIDRS, GWB);

/* Gateway GWA  forward the request to HCP2P
server encrypted using shared key K1, server
respond with  IP addr, port and Public key of
GWB */

{IPGWB,        PortGWB,        PUGWB}        =
request_gateway_info_server(DEVIDRS, GWB);

/* GWA request GWB for connection to using
PUGWB*/
Connect (GWA, GWB, NA);

/* GWB  request HCP2P server for public key of
GWA using  K2 to establish a connection with GWA
using PUGWA */
{PUGWA} = request_public_key_info_server(GWA) ;
connect_to_client_gateway (GWA, PUGWA);

/* client gateway GWA request to access IoT
resource DEVIDRS present at  GWB responds with
IoT resource session key which is derived from
pre shared key*/
{sessionKey} = request_access(DEVIDRS);

/* client connect to DEVIDRS (remote IoT device)
for accessing using sessionKey */
Connect(DEVIDRS);
{buffer[] } = read(DEVIDRS);
```

III. Results and Discussion

This section compares the proposed HCP2P-based architecture's effectiveness with the D.H.T.-based resource discovery method. Both hardware and simulation methods evaluate the proposed design. In the first stage, a minimal hardware setup (two Raspberry Pi 4 Model B devices with a cloud server) is used to experiment. The values derived from this experiment are fed to a simulation setup (standalone computer) to test the model's validity for many gateways.

The test setup includes two Raspberry Pi 4 Model B devices with 4 GB RAM as gateways/D.H.T. nodes of 400 km. Each Raspberry Pi is connected to a router with a 100 Mbps internet connection (port forwarding). Both Raspberry Pi boards are connected with temperature and humidity sensors to get the real-time sensor values for an experiment. The centralized Amazon EC2 (m1.large) with Relational Database Service (R.D.S.) is hosted at Mumbai (India) data center. The first Raspberry Pi device is hosted in Bangalore (India), around 700km. A second Raspberry Pi device from Mumbai is installed at Udupi city, which is 700k.m. Both Raspberry Pi devices are installed with Raspbian OS, Python 3.7, and Kademlia package.

To evaluate the proposed architecture on a larger scale, simulation was carried out on a Desktop computer with Intel i5 processor, 16 GB RAM, and 250 GB S.S.D., installed with Ubuntu 18.04 LTS, Python3.7, Kademlia, Apache, Mysql, and Wireshark. Multiple instances of D.H.T. nodes are created on loopback addresses with different port numbers to simulate D.H.T.-based experiments. For evaluation of the proposed HCP2P architecture, multiple client instances are created on different ports that are communicated to the server on the same machine.

A. Simulation Procedure

To set up the test scenario for HCP2P server within 1000km, both raspberry Pi devices are configured as a gateway (port forwarding to home router) which communicates with HCP2P server which is installed in Amazon data center in Mumbai, India. In the setup test scenario for HCP2P server above 1000 km, both raspberry Pi devices are configured as gateway (port forwarding to home router) which communicates with HCP2P server installed in Amazon data center in Singapore. To test D.H.T. network performance below 1000 km, both Raspberry Pi devices (Bangalore and Udupi) are configured as D.H.T. nodes along with D.H.T. nodes on Amazon Data Center in Mumbai, India. The D.H.T. node network performance is tested for a distance above 1000km, and multiple instances of D.H.T. nodes are created in Amazon web service at different Data centers like Mumbai, Singapore, Tokyo, London, and Bahrain. The node at Mumbai acts as a bootstrap node. To validate a proposed system on many nodes, values obtained by physical setup are fed to the simulation setup on a standalone computer.

B. Performance Parameters

The following are the performance metrics pursued in this work:

1) Resource Registration Time: The time a new resource takes to register itself in the IoT ecosystem. In the proposed HCP2P architecture, resource registration is updated at the

central server, whereas, in the D.H.T. network, registration must be done at multiple nodes.

2) Resource Discovery Time: The total time lapsed from request generation to identification of required resource.

3) Hop-Count: Hop-count refers to the number of intermediate devices (Routers) through which packets must pass between the source and destination device. Increased hop count adversely affects the network performance by increasing the network traffic and load on the routers.

Fig.9 and Fig.10 discuss the registration time for a new resource in the proposed HCP2P server and D.H.T.-based network. In Fig. 9, the registration time of resources in the HCP2P server (<=1000 km) is compared with two distinct D.H.T. networks, one with D.H.T. nodes (<=1000km) and another D.H.T. network with nodes running at least 1000 km apart. In this experiment, registration time for new resources with the HCP2P server (<=1000km) is below 70ms.

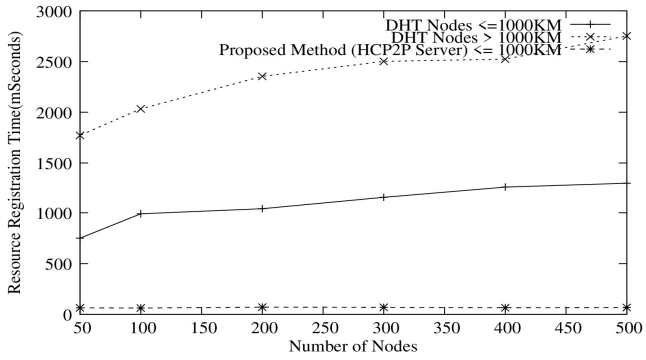

Fig. 9 HCP2P Server Registration Time Below 1000km

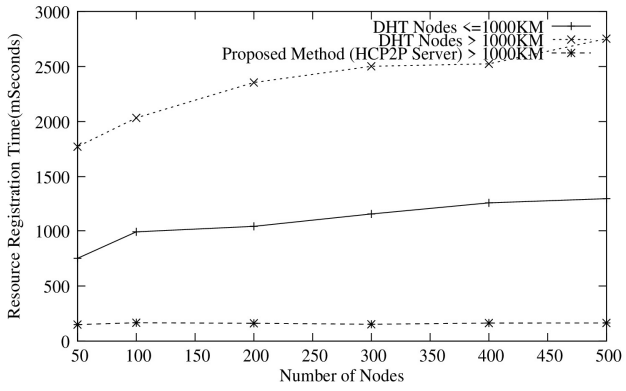

Fig. 10 HCP2P Server Registration Time Above 1000km

The new resources first register with its local gateway, and the global resources and gateway information are registered in a centralized HCP2P server. However, in a D.H.T.-based network, resources are registered with D.H.T. nodes (Raspberry Pi) which will search its neighbour nodes, and based on the key, resource information is inserted into multiple hash tables across the P2P network. This consumes comparatively larger registration time as requested D.H.T. nodes must wait for consensus among D.H.T. nodes before inserting new information. In this experiment, D.H.T. nodes less than 1000km apart have taken 750ms to 1200ms which is comparatively higher than HCP2P-based systems.

Fig.10 discusses a similar experiment to Fig.9, except that the HCP2P server is deployed above 1000 km. In this experiment, registration time with the HCP2P server is almost twice as in Fig.9. This is because of the transmission delay between the gateway and server, which are more than 1000 km apart. However, this time is comparatively lesser than traditional D.H.T.-based P2P networks.

Once the source gateway establishes a connection with the remote gateway, a resource is made available to the client through a session key. In this experiment, Fig.11, resource discovery time with HCP2P server (<=1000km) is around 100ms. In a D.H.T.-based network, clients request the specific resource to a gateway running D.H.T. Later, and the gateway broadcasts the specific resource request to multiple D.H.T. nodes in the network. Based on acknowledgment, the gateway at the client side is allowed to access remote resources. The time required to access the resource in D.H.T. nodes (<=1000km) require around 100ms. But D.H.T. nodes that are above 1000km are around 400ms to 650ms.

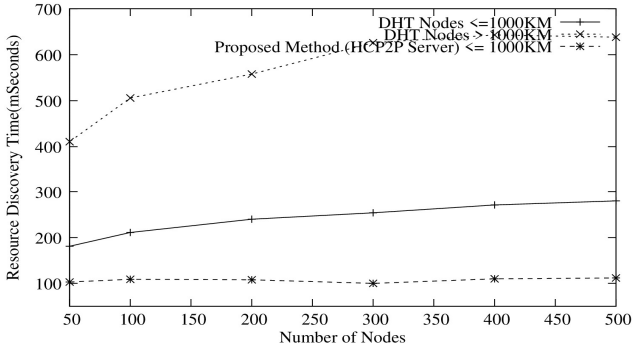

Fig. 11 Resource Discovery Time in HCP2P Server Below 1000km

Fig.12 discusses the discovery time with the HCP2P server deployed above 1000 km. In this experiment, the discovery time of the specified resource is around 250ms in the HCP2P server. Fig.12 shows resource discovery time is comparatively less than in traditional D.H.T.-based P2P networks. In traditional D.H.T.-based P2P systems, the client broadcasts the query and waits for a response from neighbour nodes. However, in the HCP2P model, the query is forwarded only to the central server, which provides the gateway address of the required resource.

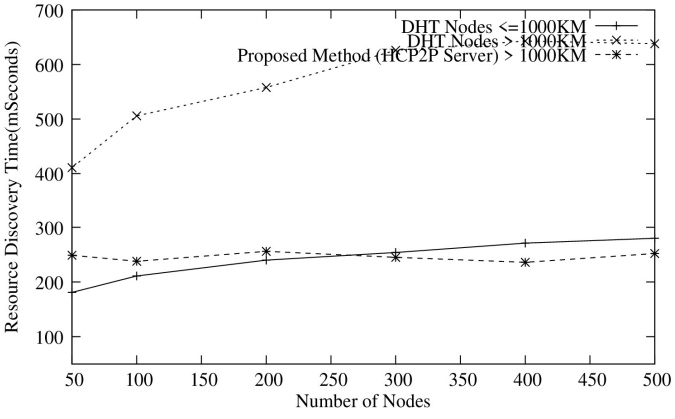

Fig. 12 Resource Discovery Time in HCP2P Server Above 1000km

Fig.13 and Fig.14 compare the hop count for the proposed HCP2P model and D.H.T. network. In Fig.13, the performance of HCP2P server deployed within 1000 km is

compared with two distinct D.H.T. networks, one with D.H.T. nodes deployed within 1000km and another with nodes running at least 1000km apart. In this experiment, the hop count between client and HCP2P server (<=1000km) is between 7 hops to 9 hops. In a D.H.T.-based network, nodes ping its neighboring nodes simultaneously for any query or update. The neighboring nodes may recursively forward this query to other nodes until the target node is identified. This will generate multiple U.D.P. packets for each transaction, which puts a load on intermediate nodes, and the total number of hop counts will go up to 70 hops. Similarly, for D.H.T. nodes (>1000km), the hop count may reach up to 140.

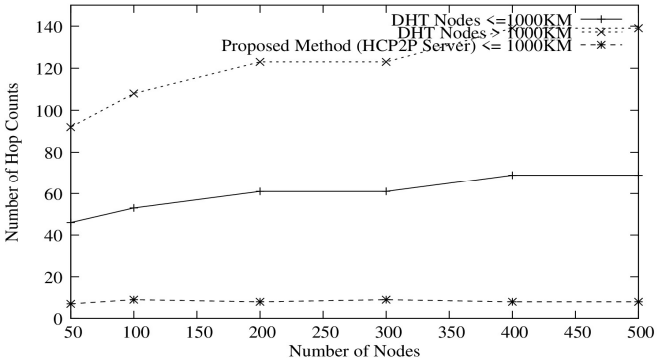

Fig. 13 Hop Count in HCP2P Server Below 1000km.

Fig.14 discusses the finding of hop count with HCP2P server deployed above 1000 km. The experiment is conducted for multiple D.H.T. nodes. The hop count obtained between the client and HCP2P server (>1000km) is 14 hops to 19 hops. But in the D.H.T.-based P2P network, multiple packets are broadcasted, and each packet crosses a similar number of hops.

However, taking the total number of hops crossed by multiple packets will be between 40 to 140 nodes, generating unnecessary packets across the network. Experiments discussed in Fig.13 and Fig.14 show that HCP2P servers perform much better than D.H.T. nodes (<=1000km and >1000km).

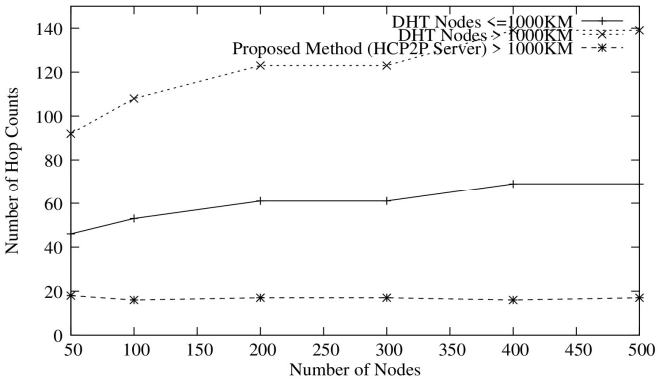

Fig. 14 Hop Count in HCP2P Server Above 1000km

IV. Conclusion

Many authors have already proposed the D.H.T.-based P2P network model to handle massive resources in IoT-based applications. Even though D.H.T.-based resource discovery is highly resilient and scalable, it incurs many maintenance costs. D.H.T.-based solutions lack peer trust, making them unsuitable for IoT applications. The proposed HCP2P architecture is more secure and efficient compared to existing D.H.T.-based P2P solutions for IoT. In the proposed design, the central authority server (HCP2P) is the root of trust and coordinates the communication between two registered gateways. It also adopts two security mechanisms: the secure P2P connection established between trusted gateways and the derived session key used for accessing the destination resource. The proposed architecture provides the peer-to-peer communication among gateways for resource discovery and resource access while reducing the load on the cloud server. Experimental results of the proposed HCP2P architecture show that resource registration, discovery, and access time are more efficient than traditional D.H.T.-based P2P systems.

References

[1] P. Franco, J. M. Martínez, Y.-C. Kim, and M. A. Ahmed, "A Cyber-Physical Approach for Residential Energy Management: Current State and Future Directions," Sustainability, vol. 14, no. 8, pp. 4639, Apr. 2022, DOI: 10.3390/su14084639.

[2] A. Gupta, and A. Al-Anbuky, "IoT-Based Patient Movement Monitoring: The Post-Operative Hip Fracture Rehabilitation Model," *Future Internet*, vol.13, no.8, pp.195-201, Jul.2021, DOI: 10.3390/fi13080195.

[3] Z. Hu, and H. Tang, "Design and Implementation of Intelligent Vehicle Control System Based on Internet of Things and Intelligent Transportation," Scientific Programming, vol. 2022, pp.1–11, Jan. 2022, DOI: 10.1155/2022/6201367.

[4] F. Hussain, S. A. Hassan, R. Hussain, and E. Hossain, "Machine Learning for Resource Management in Cellular and IoT Networks: Potentials, Current Solutions, and Open Challenges," *IEEE Communications Surveys & Tutorials,* pp.1–11, 2020, DOI: 10.1109/comst.2020.2964534.

[5] P. Gomes, E. Cavalcante, T. Batista, C. Taconet, D.Conan, S.Chabridon, F.C. Delicato, and P.F. Pires, "A Semantic-based Discovery Service for the Internet of Things," *Journal of Internet Services and Applications*, vol.10, no.1, May, 2019, DOI: 10.1186/s13174-019-0109-8.

[6] K. Elsayed, M. A. B. Ibrahim, and H. S. Hamza, "Service Discovery in Heterogeneous IoT Environments based on O.C.F./IoTivity", *In the proceedings of 15th International Wireless Communications & Mobile Computing Conference (I.W.C.M.C.)*, pp. 1160-1165,Jun.2019.

[7] G. Kayas, M. Hossain, J. Payton, and S. M. R. Islam, "SUPnP: Secure Access and Service Registration for UPnP-Enabled Internet of Things," *Journal of IEEE Internet of Things*, vol.8, no.14, pp.11561–11580, Jul.2021, DOI: 10.1109/jiot.2021.3058699.

[8] F. A. Alasdair, and E. J. Alqahtani, "Securing Communication between Fog Computing and IoT Using Constrained Application Protocol (CoAP): A Survey," *Journal of Communications*, vol.15, no.1, pp. 14–30, Jan. 2020, DOI: 10.12720/jcm.15.1.14-30.

[9] F. Montori, L. Gigli, L. Sciullo, and M. D. Felice, "LA-MQTT: Location-Aware Publish-Subscribe Communications for the Internet of Things," A.C.M. Transactions on Internet of Things, vol.3, no.3, pp.1-28, Aug. 2022, DOI: 10.1145/3529978.

[10] Z. Talib, and S. Al-Azez, "Optimised Green IoT Network Architectures," 2018. Accessed: Apr. 26, 2022. [Online]. Available: https://etheses.whiterose.ac.uk/22224/1/Al-Azez_ZTS_PhD_2018.pdf.

[11] E. Khatibi, and M. Sharifi, "Resource Discovery Mechanisms in Pure Unstructured Peer-to-Peer Systems: A Comprehensive Survey," *Journal of Peer-to-Peer Networking and Applications,* vol.14, no.3, pp.1-18, Mar.2021, DOI: 10.1007/s12083-020-01027-9.

[12] P. Goudarzi, A. M. Rahmani, and M. Mosleh, "Resource Discovery Approaches in CloudIoT: A Systematic Review," The Journal of Supercomputing, May 2022, DOI: 10.1007/s11227-022-04541-0.

[13] M.Achir, A. Abdelli, L.Mokdad, and J. Benothman, "Service Discovery and Selection in IoT: A Survey and a Taxonomy**",** *Journal of Network and Computer Applications*, vol.20, pp.1–40, May, 2022, DOI: https://doi.org/10.1016/j.jnca.2021.103331.

[14] G. Tanganelli, C. Vallati, and E. Mingozzi, "Edge-Centric Distributed Discovery and Access in the Internet of Things," *IEEE Internet of*

Things Journal, vol.5, no.1, pp.425–438, Feb.2018, DOI: 10.1109/jiot.2017.2767381.

[15] Y. Li, "Peer-to-Peer Based Web of Things Resource Management", *In the proceedings of Complex, Intelligent, and Software Intensive Systems*, pp. 402-416, Jan, 2020.

[16] L. Murturi, C. Avasalcai, C. Tsigkanos, and S. Dustdar, "Edge-to-Edge Resource Discovery using Metadata Replication", *In the proceedings of IEEE 3rd International Conference on Fog and Edge Computing (I.C.F.E.C.)*, pp.1-6, May, 2019.

[17] M. B. M. Kamel, B. Crispo, and P. Ligeti, "A Decentralized and Scalable Model for Resource Discovery in IoT Network", *In the proceedings of Wireless and Mobile Computing, Networking and Communications (WiMob)*, pp.1-4, Oct.2019.

[18] P. Krivic, P. Skocir, and M. Kusek, "Agent-Based Approach for Energy-Efficient IoT Services Discovery and Management", *In the proceedings of Agents and Multi-Agent Systems: Technologies and Applications*, vol.96, pp.57-66, Jan.2019.

[19] M. B. M. Kamel, Y. Yan, P. Ligeti, and C. Reich, "A Decentralized Resource Discovery Using Attribute Based Encryption for Internet of Things," In Proceedings of the 2020 4th Cyber Security in Networking Conference (CSNet), pp.1-3, Oct. 2020.

[20] M. B. M. Kamel, Y. Yan, P. Ligeti, and C. Reich, "Attred: Attribute Based Resource Discovery for IoT," Journal on Sensors, vol. 21, no. 14, p. 4721, Jul. 2021, DOI: 10.3390/s21144721.

[21] B. Mocanu, F. Pop, A. Mihaita, C. Dobre, and A. Castiglione, "Data Fusion technique in SPIDER Peer-to-Peer Networks in Smart Cities for Security Enhancement**",** *Journal of Information Science*, vol. 479, pp.607-621, Apr. 2019.

[22] L. Campioni, R. Lenzi,F. Poltronieri, M. Pradhan, M.Tortonesi, C.Stafenelli, and N. Suri, "M.A.R.G.O.T.: Dynamic IoT Resource Discovery for H.A.D.R. Environments", *In the proceedings of IEEE Military Communications Conference (M.I.L.C.O.M.)*, pp. 809-814, Nov. 2019.

[23] M. Pradhan, F. Poltronieri, and M. Tortonesi, "Dynamic Resource Discovery and Management for Edge Computing Based on SPF for H.A.D.R. Operations", *In the Proceedings of Military Communications and Information Systems (I.C.M.C.I.S.)*, pp. 1-6, May, 2019.

[24] L. Cheklat, M. Amad, M. Omar, and A. Boukerram, "C.H.E.A.R.P.: Chord-based hierarchical energy-aware routing protocol for wireless sensor networks," *Journal of Computer Science and Information Systems*, vol. 18, no.3, pp.813–834, 2021, DOI: 10.2298/csis200308043.

[25] S. Zahid, K. Ullah, A. Waheed, S. Basar, M. Zareei, and R. R. Biswal, "Fault Tolerant DHT-Based Routing in MANET", Journal of Sensors, vol. 22, no.4280, pp. 1-24, Jun. 2022, DOI: 10.3390/s22114280.

[26] R. Kousar, M. Alhaisoni, S. A. Akhtar, N. Shah, A. Qamar, and A. Karim, "A Secure Data Dissemination in a DHT-Based Routing Paradigm for Wireless Ad Hoc Network," International Journal of *Wireless Communications and Mobile Computing*, vol. 2020, pp. 1–32, Aug. 2020, DOI: 10.1155/2020/2740654.

[27] Y. Ohba, J. Y. Koh, N. Ng, and S. L. Keoh, "Performance Evaluation of a Blockchain-based Content Distribution over Wireless Mesh Networks," *IEEE 7th World Forum on Internet of Things (WF-IoT)*, pp.258-263, Jul.2021, DOI: 10.1109/WF-IoT51360.2021.9595503.

[28] N. Shukla, D. Datta, M. Pandey, and S. Srivastava, "Towards software defined low maintenance structured peer-to-peer overlays", Journal of Peer-*to-Peer Networking and Applications, v*ol.14, no.8, pp. 1242-1260, 2021, DOI:10.1002/s12083-021-01112-7.

[29] F. O. Ehiagwina, N. A. Iromini, I. S. Olatinwo, K. Raheem, and K. Mustapha, "A State-of-the-Art Survey of Peer-to-Peer Networks: Research Directions, Applications and Challenges," *Journal of Engineering Research and Sciences,* vol.1, no.1, pp. 19–38, Feb. 2022, DOI: 10.55708/js0101003.

[30] W. Hou, Y. Jiang, W. Lei, A. Xu, H. Wen, and S. Chen, "A P2P Network based Edge Computing Smart Grid Model for Efficient Resources Coordination", Journal of *Peer-to-Peer Networking and Applications,* vol.13, no.1, pp.1-12, Jan. 2020, DOI: 10.1007/s12083-019-00870-9.

[31] A. Alhussain, H. Kurdi, and L. Altoaimy, "Managing Trust and Detecting Malicious Groups in Peer-to-Peer IoT Networks," Journal of *Sensors*, vol. 21, no. 13, pp. 4484, Jun. 2021, DOI: 10.3390/s21134484.

[32] F. L.L.de-Mendonça, D.Cunha, B. J. G.Praciano, M. Zanatta, J. Costa, and R. Sousa, "P2PIoT: A Peer-To-Peer Communication Model for the Internet of Things." *2019 Workshop on Communication Networks and Power Systems (W.C.N.P.S.)*,pp. 1-5, Nov.2019.

[33] S. M. Muzammal, R. K. Murugesan, and N. Z. Jhanjhi, "A Comprehensive Review on Secure Routing in Internet of Things: Mitigation Methods and Trust-Based Approaches," *IEEE Internet of Things Journal*, vol.8, no.6, pp.4186–4210, Mar. 2021, DOI: 10.1109/jiot.2020.3031162.