Dynamic Resource Discovery and Management for Edge Computing Based on SPF for HADR Operations

Manas Pradhan
Information Technology for Command and Control
Fraunhofer Institute for Communication,
Information Processing and Ergonomics
Wachtberg, Germany
Email: manas.pradhan@fkie.fraunhofer.de

Filippo Poltronieri, Mauro Tortonesi
Distributed Systems Research Group
University of Ferrara
Ferrara, Italy
Email: filippo.poltronieri@unife.it, mauro.tortonesi@unife.it

***Abstract*—The Smart City concept tries to inherit the advantages of Internet-of-Things (IoT) into its realm to function alongside the existing legacy systems. One of the most promising aspects of IoT is Edge Computing, which tries to move the computing, traditionally done via a centralized infrastructure like the cloud to the edge of the network. This allows remote deployment of IoT assets closer to the source and application area of information enabling faster response times of action. Smart Cities of future envision using Edge Computing to their advantage for remote and distributed computing. Sieve, Process and Forward (SPF) is an Edge Computing solution for dynamic IoT applications for Smart City scenarios. The military is looking forward to use, as well as develop the SPF platform for its Edge Computing requirements. But currently, the SPF platform does not have the mechanism for remote discovery of edge resources and their management to leverage its potential completely. This paper tries to propose a resource discovery and management architecture and methodology for SPF to support future Human Assistance and Disaster Recovery (HADR) operations in Smart City environments with the vision of enabling interoperability between civilian and military platforms.**

***Index Terms*—HADR, Interoperability, IoT, ICT, Smart City, Edge Computing**

I. Introduction

The concept of Internet-of-Things (IoT) adoption in everyday life has gained momentum in the past 5 years or so [1], [2]. Processors, controllers, sensors and actuators are getting embedded in everyday things impacting the interaction and perception of humans with their surroundings. Ranging from the mobile devices, smart home assistants, building sensors to industrial automation enablers, the scale and diversity of implementations of IoT-technologies has multiplied manifold. Apart from the hardware aspects, there has been development in IoT related communication technologies, data communication protocols, security and privacy mechanisms and interoperability aspects between IoT technologies as well as between IoT and legacy technologies [3]–[6].

Along with the spurt of IoT, there has been the development of the concept of Smart Cities. The populations around the world are converging towards the city-spheres exerting huge pressure on city administrations to come up with new ideas to tackle the needs of the populations [7]. One of the ways to deal is to develop the Information and Communications Technology (ICT) platforms in the cities which can assist the administrations get real-time data and analysis on the citizens' problems [8] along with the traditional administrative techniques [9]. The various techniques like crowdsourcing, crowdsensing and edge-computing allow the administrators to get on-ground and closest to the problem-area data and thus help them actuate specific and effective responses [10]. These techniques can in-turn provide various services to the citizens in terms of data about the city and its functioning.

The Smart Cities are adopting the concepts of IoT due to its inherent advantages for mobile and remote deployment. Edge computing based on IoT tries to gather data, analyze and compute, and actuate closest to the source [11]. Sieve, Process and Forward (SPF) is such a Edge Computing and Value-of-Information (VoI) based solution for dynamic IoT applications for Smart City Edge Computing scenarios [12].

The NATO IST-147 group for Military Application of IoT used SPF in a live demonstration at ICMCIS 2018 conference in Warsaw, Poland to demonstrate military using the civilian or public city assets for HADR operations alongside the military IoT and legacy assets [17]. The SPF platform was used to consume Warsaw city data as an edge platform from publicly available APIs for street camera access and send relevant data to the US Army's Android Tactical Assault Kit (ATAK) Command and Control (C2) application. During the demonstration some limiting aspects of SPF were discovered.

The SPF architecture is limited in its functionality in the sense that there is no way to dynamically locate or discover resources as and when required and manage them. This management aspect also limits the way SPF chooses the edge resources for use-case based executions on the fly. This paper tries to propose a generic methodology to the existing SPF platform to extend its functionality w.r.t how the assets on the edge can be discovered and managed to leverage their availability and capabilities to the fullest.

978-1-5386-9383-4/19/$31.00 ©2019 IEEE

The rest of the paper is divided as follows: Section III gives an insight into SPF as an Edge Computing Solution for IoT environments and its limitations. Section IV describes the asset discovery, monitoring, management and actuation of Edge Nodes. Section IV-B describes the interfaces involved for communication through the whole life-cycle of an interaction of SPF and other military ICT components. Section II points out the related work for existing edge computing platforms w.r.t asset discovery and management. Finally, section V presents the conclusions and future work.

II. Related Work

Resource discovery and management for IoT application is still an active research topic. Different protocols have been proposed and analyzed in order to locate resources and services in a dynamic and challenging environment. The work in [22] discusses and evaluates different discovery technologies for the IoT, by illustrating solutions such as multicast Domain Name System (mDNS), multicast CoAP, the Simple Service Discovery Protocol (SSDP), and so on. An interesting work that makes use of Name Data Networking solutions (NDN) is proposed in [18]. In this work, the authors present a NDN discovery mechanism based on a service-response model. In this model, a consumer asks for a desired service to the devices in the neighborhood using a broadcast message with a predefined Time to live (TTL). If a service provider is not found, the consumer sends another message with an increased TTL until a provider for the service is found or the maximum defined TTL is reached. On the other end, a provider will reply back to the consumer if it is eligible to satisfy its request. Finally, the authors propose a deferral scheme to avoid collisions over the same service request. Instead, a discovery approach based on an interoperability model to bridge a NDN and an Internet Protocol (IP) network is described in [20]. This approach makes use of mDNS inside the IP network and the Named Publish Subscribe Networking protocol on the NDN network to discover both consumers and devices. The authors propose the adoption of a gateway solution called Future Internet eXchange Point (FIXP) to bridge the different communication protocols. In [21], the authors describe a discovery approach, which makes use of MQTT to keep track of publishers/advertisers (IoT devices) in a IoT-Fog environment. In particular, the authors propose a protocol, namely Smart and Power Efficient Node Discovery Protocol (SPEND) as solution to create a reliable and energy efficiency discovery solution for IoT applications. Experimental results seems to evaluate the protocol power efficiency and effectiveness, and the authors conclude that MQTT is a reliable and efficient protocol for constrained devices.

III. Sieve, Process and Forward

SPF is a Edge Computing and Value-of-Information based solution [13] for the management of IoT applications at the edge. SPF exploits the proximity of sensors and devices to collect and process data directly at the edge of the network to provide responsive and more effective services to the end users. In addition, SPF supports also the coordinated execution of processing tasks among the edge and the Cloud depending on the status of the available IoT devices and the computational resource requirements.

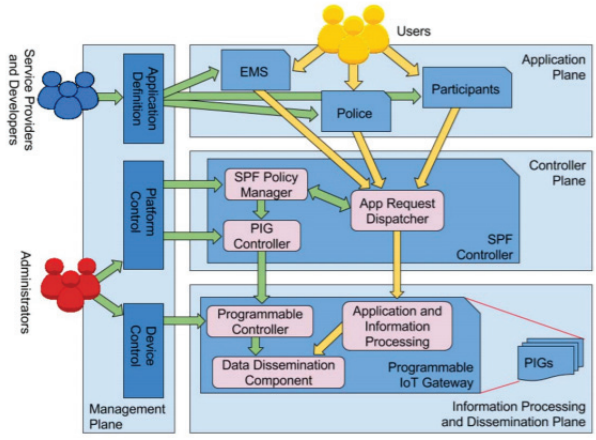

Fig. 1. Generic SPF Architecture— scale and include better resolution image here [12]

SPF adopts the VoI metric as information evaluation criterion to evaluate a single piece of information from its generation to its delivery to the end users of services. In fact, SPF filters information objects by using a minimum content difference threshold for new IoT data, for processing and dissemination in order to reduce the deluge of data to be analyzed, by filtering and processing only the most valuable pieces of information.

A. Existing Architecture

Fig. 1 illustrates the main components of SPF: the SPF Controller and the SPF Programmable IoTs Gateway (PIGs). Within the SPF architecture, the SPF Controller is responsible for the management functionalities and it is the interface between users and services. Instead, PIGs are responsible to collect raw data from IoT sensors in order to produce valuable information to be disseminated to the end users of services. PIGs can be deployed directly on the gateway nodes or dedicated hardware at the edge or on a Cloud Computing platform.

More in detail, the SPF architecture adopts the Adaptive, Information-centric, and Value-based (AIV) information maturity model, which divides the information processing stage in three different phases [15]. First, PIGs collect raw data from sensors at the edge. Then, PIGs filter and elaborate raw data to produce Information Objects (IOs) on pipelines. Finally, PIGs aggregate IOs on services to produce Consumer Ready Information Objects (CRIOs), which represent the information in its final stage, ready to be consumed by users. With this design, SPF aims at enabling the reusability of components (pipelines and services) and maximizing the utility of the single piece of information, e.g. an IO could be used to produce one or more CRIOs.

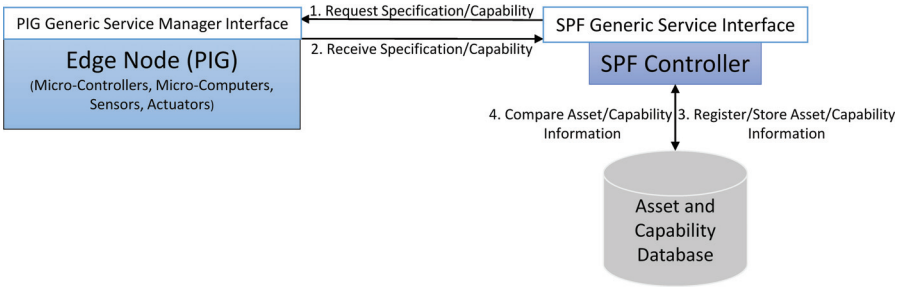

Fig. 2. SPF Resource Discovery and Registry

B. Architectural Limitations

The current version of SPF does not support the discovery and management of the edge resources. The SPF controller modules have to be hard-coded with device specific configurations that can neither be inserted, changed nor accessed at run-time [15]. If a SPF controller would want to discover PIG resources at the edge and keep a track of the edge resources' information or metadata, then there is no way to have a listing/registry of the edge resources available and their capabilities. Thus the model of SPF which enables users to trigger specific services at run-time based on requirements would not scale and its capabilities can not be leveraged to the maximum. In addition, in the future it would not be possible for remote configuration and updates of these edge devices without their metadata and availability information.

IV. Resource Discovery and Registry

For leveraging the full capability of SPF i.e. IoT at the edge, it is necessary that use-cases or stakeholders' requests can be taken of by the SPF controller, as and when required. The SPF controller should be able to find out the best possible match of a resource at any instant for a service request. The controller, based on the information/action requested by a stakeholder should be able to find a PIG with the adequate resources in terms of computation power, connectivity, bandwidth etc. and its associated resources such as sensors, actuators etc.

In order for the SPF to trigger a particular PIG gateway, it should know:

1) What is the location of the PIG gateway (edge node) in a city or area?
2) What are the capabilities of the particular edge node i.e. what resources are available on it?
3) What are the various assets associated with the PIG such as sensors, actuators, controllers etc. ?

Fig. 2 shows the process of resource discovery and storage and can be performed in the following way:

1) With the help of the Generic Service Interface (GSI) running on the SPF, the SPF can issue requests to return asset (PIG) specification and its associated capabilities. It means that the SPF requests for:
   - The id of the PIG controller which can be dynamically generated by the PIG based on a randomized algorithm.
   - The ids of the resources associated with the PIG.
   - The type of the resource i.e. the resource is a sensor, actuator, computer etc.
   - The capability of the resource. For example, what is the precision of a radiation sensor.
   - Availability and life of the resource.
2) The Generic Service Manager Interface (GSMI) on PIG in turn interprets the request for resource and capability specification and returns the IDs and capabilities of its associated resources.
3) The SPF controller receives the responses from the PIGs and stores them its Asset and Capability Database for further operations. If a request comes from a stakeholder application, like a Command and Control (C2) application from the city police to access a particular resource in a particular location, then the SPF controller can look up its database and trigger the PIG responsible for the corresponding asset and capability.

As described in [17], MQTT topics can be used for IoT-based applications for Human Assistance and Disaster Recovery (HADR) scenarios for achieving interoperability and light-weighted protocol-based data exchange. These MQTT topics can contain "Request Specification Topics" which can ask for and thus return asset and capability information as JSON payloads from the PIGs. These JSON payloads can then in turn be interpreted at the SPF end and stored in the database as relational data. The SQL-based databases like *SQLlite* can be used for the light-weight in-memory storage at the SPF controller.

A. Resource Monitoring and Management

The generic architecture for SPF can be extended to include monitoring and management of the edge nodes. Fig. 3 shows the basic mechanism for this purpose by reusing the SPF functionalities/modules and the generic architecture components. The idea is to use resource discovery and registry capabilities as described in sub-section IV to implement:

1) Node Condition Monitoring and Service Provisioning: The SPF Controller can issue requests to the PIG to instantiate a pipeline for health and status monitoring of the PIG and its associated resources. The GSI on PIG can trigger the pipelines to send keep-alive messages alongst with other requested health data at

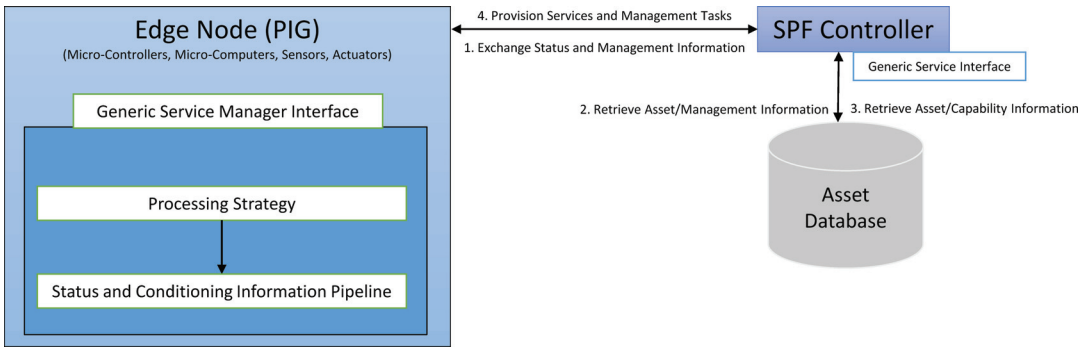

Fig. 3. Resource Monitoring and Management of Edge Nodes

constant intervals (a scheduler on pipeline decides the frequency of the messages sent out to SPF Controller). The Controller can then enter the data associated with the edge node in its Asset Database. The controller can use this data to trigger a service for a particular PIG for a use-case whenever required.

2) Device Access Control: With the access to the PIG, the Controller can control which devices are permitted to connect to the HADR infrastructure. Since the current state of HADR operations use MQTT messaging which in turn uses the MQTT broker which can be deployed alongside the SPF controller The Asset Database then can be used to find out the availability of the edge nodes and which nodes need still to be connected to the HADR Infrastructure and which ones to drop off. The MQTT broker can then block the users or PIGs from publishing or subscribing to the data on the HADR infrastructure. This decision can be made based on factors like:
   - Health Status of the PIG and its associated resources
   - Connectivity and bandwidth restrictions of the SPF Controller's network resources
   - Proximity and redundancy of the PIGs available
   - Asset Database resources available on the controller
   - Access permission changes to assets
3) Node Security and Legitimacy Monitoring: The health status and keep-alive message variables can be used to control the state of edge nodes and monitor their activities. These messages can indicate any unscrupulous activity of the nodes in sending data to the SPF Controller and thus any anomaly when detected can be used to remove a device from the HADR network and ensure legitimacy of the message senders.
   For example, the Asset Database can also store the historical telemetry data received from the MQTT Topics. Using data analytics on the Asset Database records, potential anomalous behavior for the PIGs can be identified and a preventative solution can be implemented. A corrective action for the anomalous PIG would be that, it can be blacklisted i.e. removed from the Asset Database list and the broker would not accept or forward packets from the the PIG. If the network is bridged, the other SPF controllers can be notified regarding the anomalous PIG blocking the PIG out of the HADR network.
4) Remote Service Deployment and Updates: The registry of the assets can be used locate to push new services to be deployed on the PIGs and also to update the existing services by using a ftp (File Transfer Protocol) service over a higher bandwidth network such as LTE.

B. Interfaces Involved and Process-flow

Fig. 4 shows the entire process flow involving the SPF components for a scenario involving a service invocation request from a C2 (Stakeholder) Application. A live demonstration by the NATO IST-147 group was carried out in Warsaw, Poland which showed the military using the SPF edge computing solution for using in HADR scenarios [16], [17]. Considering a similar use-case where the SPF controller can be deployed on a Mobile Tactical Operations Center (MTOC) i.e. on a mobile military vehicle. The MTOC can then interact with multiple deployed PIGs on the ground and with C2 application counterparts from a mobile operations center.

The following steps describe the interaction:

1) The C2 application or a user manually triggers a specific service to be delivered. In this case, the C2 might ask for a High Definition (HD) video stream to be delivered from a certain area defined by a bounding-box or perimeter to see the status of a building collapse during an earthquake, the affected humans and vicinity.
2) The MTOC application receives the request and accordingly sends the service invocation request for HD camera streams to be delivered to the GSI of SPF Controller. The GSI interprets the service request and looks up in its database as to which PIG and associated resource has to be invoked. The decision as to which PIG has be selected and invoked for service execution from the Asset Database in this case would be done based on these parameters:
   - The PIG stationed closest to the requested area.
   - If the PIG has a camera resource associated with it which can deliver HD video stream.
   - If the PIG satisfies the minimal Quality-of-Service (QoS) requirements based on network characteristics.

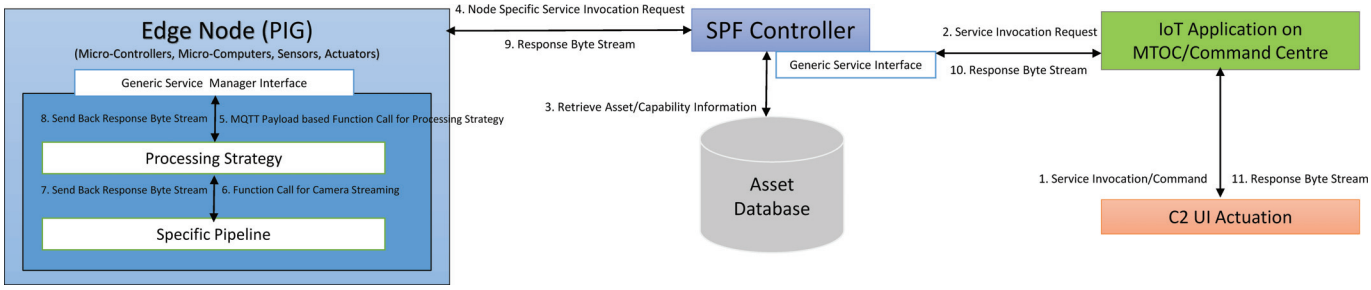

Fig. 4. Service Invocation and Process Flow for Tasks

After it selects and retrieves the required asset (PIG) information, it sends over the request to the location specific PIG asking to perform a specific operation and return the result.

3) The GSMI at the Edge Node (PIG) receives and interprets the request and triggers the service strategy responsible for video streaming operations. This in turn triggers and selects a pipeline which uses the HD camera asset to deliver the results back to the PIG interface.
4) Once the specific camera asset is triggered, it starts sending back video stream as the stream of bytes to the PIG interface. The processing strategy can here decide that the stream of bytes can be delivered directly as an UDP video stream instead of packing the data into a JSON payload based on a MQTT topic. The PIG controller thus starts streaming these sequence of bytes to the SPF controller which then eventually gets delivered to the C2 UI where the stream is decoded and recontructed to be viewed as a video stream.

V. Conclusion

The paper describes a generic architecture of an Edge Computing solution based on SPF for a military HADR scenario which can be extended for any 3rd party usage such as Smart City ICT systems. The idea is to make the service interfaces abstract and extend them for a Service Oriented Architecture (SOA) approach. This would enable better cohesion between the interacting applications and allow for future application/service integration and scalability in operations. Also, a way to store and retrieve asset information and capabilities is described so that a specific asset can be used for a specific use-case based request in a HADR scenario.

Currently, there is no dynamic coupling of the remote edge nodes' operations and their relative workloads being reported back to the SPF controllers. Future work involves resource manager configurations on the SPF Controllers getting updated monitoring data from the PIGs regarding their computational capability and resources available such as threads and processes running. The idea is to select at the run-time, which of the edge nodes are suitable for task offloading based on the monitoring data received. These dynamic run-time configuration managers would dictate the edge nodes' (single or multiple) selection for task offloading in addition to the asset databases which store the nodes' discovery and access information.

References

[1] Bandyopadhyay, Debasis, and Jaydip Sen. "Internet of things: Applications and challenges in technology and standardization." Wireless Personal Communications 58.1 (2011): 49-69.
[2] Saha, Himadri Nath, Abhilasha Mandal, and Abhirup Sinha. "Recent trends in the Internet of Things." Computing and Communication Workshop and Conference (CCWC), 2017 IEEE 7th Annual. IEEE, 2017.
[3] Lin, Jie, et al. "A survey on internet of things: Architecture, enabling technologies, security and privacy, and applications." IEEE Internet of Things Journal 4.5 (2017): 1125-1142.
[4] Hui, Terence KL, R. Simon Sherratt, and Daniel Díaz Sánchez. "Major requirements for building Smart Homes in Smart Cities based on Internet of Things technologies." Future Generation Computer Systems 76 (2017): 358-369.
[5] Wollschlaeger, Martin, Thilo Sauter, and Juergen Jasperneite. "The future of industrial communication: Automation networks in the era of the internet of things and industry 4.0." IEEE Industrial Electronics Magazine 11.1 (2017): 17-27.
[6] Atzori, Luigi, Antonio Iera, and Giacomo Morabito. "The internet of things: A survey." Computer networks 54.15 (2010): 2787-2805.
[7] Petrolo, Riccardo, Valeria Loscri, and Nathalie Mitton. "Towards a smart city based on cloud of things, a survey on the smart city vision and paradigms." Transactions on Emerging Telecommunications Technologies 28.1 (2017): e2931.
[8] Santana, Eduardo Felipe Zambom, et al. "Software platforms for smart cities: Concepts, requirements, challenges, and a unified reference architecture." ACM Computing Surveys (CSUR) 50.6 (2017): 78.
[9] Yaqoob, Ibrar, et al. "Enabling communication technologies for smart cities." IEEE Communications Magazine 55.1 (2017): 112-120.
[10] Guo, Bin, et al. "Mobile crowd sensing and computing: when participatory sensing meets participatory social media." IEEE Communications Magazine 54.2 (2016): 131-137.
[11] Shi, Weisong, and Schahram Dustdar. "The promise of edge computing." Computer 49.5 (2016): 78-81.
[12] Tortonesi, Mauro, et al. "SPF: an SDN-based middleware solution to mitigate the IoT information explosion." 2016 IEEE Symposium on Computers and Communication (ISCC).
[13] M. Tortonesi, M. Govoni, A. Morelli, G. Riberto, C. Stefanelli, N. Suri, Taming the IoT data deluge: An innovative information-centric service model for fog computing applications, Future Generation Computer Systems, 2018.
[14] Abdelwahab, S., Hamdaoui, B., Guizani, M., & Znati, T. (2015). "Cloud of things for sensing as a service: Sensing resource discovery and virtualization." 2015 IEEE Global Communications Conference, GLOBECOM 2015.
[15] Tortonesi, M., Michaelis, J., Morelli, A., Suri, N., & Baker, M. A. (2016, June). SPF: An SDN-based middleware solution to mitigate the IoT information explosion. In 2016 IEEE Symposium on Computers and Communication (ISCC) (pp. 435-442). IEEE.
[16] Johnsen, Frank T., et al. "Application of IoT in military operations in a smart city." 2018 International Conference on Military Communications and Information Systems (ICMCIS). IEEE, 2018.

[17] Pradhan, Manas, et al. “Toward an Architecture and Data Model to Enable Interoperability between Federated Mission Networks and IoT-Enabled Smart City Environments.” IEEE Communications Magazine, 16 Oct. 2018, pp. 163–169.
[18] Amadeo, M., Campolo, C., & Molinaro, A. (2016). "NDNe: Enhancing named data networking to support cloudification at the edge." IEEE Communications Letters, 20(11): 2264-2267.
[19] Aazam, M., & Huh, E. -. (2014). "Fog computing and smart gateway based communication for cloud of things." 2014 International Conference on Future Internet of Things and Cloud, FiCloud 2014: 464-470.
[20] Quevedo, José & Ferreira, Rui & Guimarães, Carlos & Aguiar, Rui & Corujo, Daniel. (2017). "Internet of Things Discovery in Interoperable Information Centric and IP Networks: IoT Discovery in Interoperable Information Centric and IP Networks." Internet Technology Letters.
[21] Venanzi, R., Kantarci, B., Foschini, L., & Bellavista, P. (2018). "MQTT-driven sustainable node discovery for internet of things-fog environments." IEEE International Conference on Communications, 2018-May.
[22] Bröring, Arne & Datta, Soumya Kanti & Bonnet, Christian. (2016). 6th "A Categorization of Discovery Technologies for the Internet of Things." International Conference on the Internet of Things: 131-139.