A Novel and Scalable Naming Strategy for IoT Scenarios

Alejandro Gómez-Cárdenas, Xavi Masip-Bruin, Eva Marín-Tordera and Sarang Kahvazadeh

Advanced Network Architectures Lab (CRAAX)
Universitat Politècnica de Catalunya (UPC), Barcelona, Spain
{alejandg, xmasip, eva, skahvaza}@ac.upc.edu

Abstract. Fog-to-Cloud (F2C) is a novel paradigm aimed at increasing the benefits brought by the growing Internet-of-Things (IoT) devices population at the edge of the network. F2C is intended to manage the available resources from the core to the edge of the network, allowing services to choose and use either a specific cloud or fog offer or a combination of both. Recognized the key benefits brought by F2C systems, such as low-latency for real-time services, location awareness services, mobility support and the possibility to process data close to where they are generated, research efforts are being made toward the creation of a widely accepted F2C architecture. However, in order to achieve the desired F2C control framework, many open challenges must be solved. In this paper, we address the identity management challenges and propose an Identity Management System (IDMS) that is based on the fragmentation of the network resource IDs. In our approach, we divide the IDs into smaller fragments and then, when two nodes connect, they use a portion of their full ID (n fragments) for mutual identification. The conducted experiments have shown that an important reduction in both, the query execution times and the space required to store IDs, can be achieved when our IDMS is applied.

Keywords: IDMS, identity management, fog-to-cloud, resource identity.

1 Introduction

The Internet of Things (IoT) is a communication paradigm that allows all kind of objects to connect to the Internet network. According to [1] on 2020 the number of connected devices will reach the 50 billion, that is, 6.58 times more than estimated world population for the same year. Aligned to the constant growth of the IoT devices population, the amount of data they generate at the edge of the network is growing as well. Every day, large volumes of data in all formats (video, pictures, audio, plain text, among others) are generated and then moved to cloud datacenters to be processed. In fact, it is estimated that in the near future only an autonomous car will produce up to 4 TB data on a daily basis [2].

It is widely accepted that useful information can be extracted from data, using cloud-based data mining techniques. Nevertheless, moving large amounts of data from the edge to the datacenters located at the core of the network may incur signifi-

cant overhead in terms of time, network throughput, energy consumption and cost [3]. To overcome these issues, novel computing paradigms such as fog computing have emerged at the edge of the network.

Fog computing is a paradigm intended to extend cloud computing capacities to the edge of the network, allowing data to be processed and aggregated close to where it is generated [4]. The fact that Fog computing is deployed close to the end users devices facilitates some key characteristics for IoT services and applications, such as for example, low-latency, mobility, and location-awareness [5]. Indeed, Fog computing emerged to collaborate with cloud computing, thus not competing each other.

Nowadays, the new combined fog-to-cloud [6] proposed to ease service execution in hierarchical fashion fog, cloud, or combination of both. There are two ongoing projects to deploy the hierarchical and combined F2C system. One of them is called OpenFog consortium [7] and another one mF2C [8]. The mF2C project, at early stage proposed a hierarchical and layered architecture that the whole set of resources can be executed in cloud, fog, or combination of both. In mF2C, distributed fog nodes can be utilized for delay-sensitive and demanded low-latency services and processing at the edge of the network, and in parallel, cloud can be used for massive and long-term processing and storage.

In a realistic scenario, F2C is shown as a hierarchical three tiers architecture [9] where the most constrained devices are located at the lower tier. The middle tier is integrated by nodes that act as aggregator of the available resources for the lower layer (see Figure 1) and finally, at the top of the hierarchy, the cloud datacenter is located.

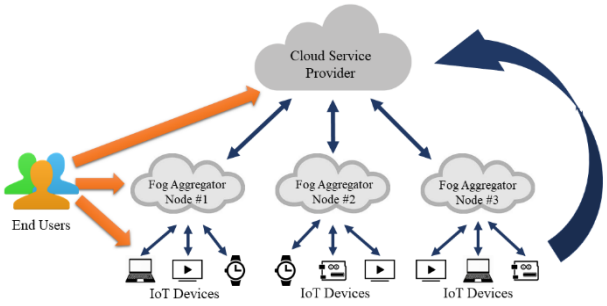

Fig. 1. Fog-to-Cloud general topology.

Certainly, the F2C resources continuum must be managed by a control strategy (sort of control plane), but because there are still many challenges to be solved, the control concept as a whole, is yet an open issue for Fog and surely F2C systems.

One of the challenges to be addressed in F2C systems is the lack of an Identity Management System that meets the specific paradigm requirements. In F2C, the Identity Management System (IDMS) is the set of functions aiming to provide a mechanism to assign and in general to manage, the resource identities of both, physical and virtual devices. According to [10], the management of the resource identifiers at the edge is very important for programming, addressing, things identification, data com-

munication, authentication, and security. Thus, the IDMS is a key component of the F2C control plane framework.

In short, some of the features an IDMS should provide in F2C system are: i) the capability to scale smoothly in parallel with the network; ii) supporting devices mobility without losing their identities; iii) security and privacy protection; iv) interoperability among different service providers and; v) supporting highly dynamic network topologies.

In this paper, we focus on the IDMS challenge and propose a solution that address the aforementioned system requirements. The key contributions of our work when compared with other available solutions include the mobility support, it is, the capability of the edge devices to keep their identifiers, even when they are on the move. Such ID persistence eases the mutual identification and authentication processes between a node and an aggregator node in future interactions. Likewise, the IDMS strategy that we propose allows to adjust the identifiers size that the resources use in the network without losing the identity uniqueness property. Finally, unlike other solutions, our proposal is focused in reducing the compute load required to identify the resources in the network. This undoubtedly benefits the entire network, especially the lowest layer in the hierarchy where the resources are very constrained and therefore a more efficient management of them is required.

The remainder of this paper is organized as follows. In section 2 other IDMS solutions are reviewed. In section 3 our IDMS proposal is described. The evaluation and results are presented in section 4 and finally, in section 5 the conclusions and future work are discussed.

2 Related Work

In computer networks, the name and the address of a device stand for two different things. The general distinction between a name and an address is that a name can remain with an entity even when that entity is mobile and moves among different locations (i.e. addresses) [11]. From the IDMS perspective, the mobility support offered by F2C means that the identifiers assigned to the network resources are persistent, i.e., they remain even if the attributes, such as the location of the devices change. Therefore, the usage of addressing techniques to manage the resources identity in F2C is not the proper solution. Rather, an IDMS that gives support to both, static and mobile nodes in the network, must be considered.

Under this premise, in this section we pay special attention to IDMS solutions whose target include IoT-devices. The rationale of this decision is that generally speaking, IoT puts together static and mobile devices, thus, providing support to all of them is mandatory in any solution to be deployed in the IoT arena.

In [12], authors present a smart home operating system for the IoT named Edge-OS_H. In EdgeOS_H, the architecture component in charge of managing the devices identities is the naming module. Such module allocates unique human friendly names describing the location (where), role (who) and data description (what) of the devices,

for example, LivingRoom.CellingLight.Bulb2. These names are used by the operating system to manage services, data and devices.

Nevertheless, the way in which $EdgeOS_H$ manages the devices identities presents several drawbacks that prevent it from being used in F2C environments. For example, human-meaningful names ease to disclose sensitive information and to access unauthorized network resources through masquerade attacks. Another issue refers to the fact that it is not prepared to support the tremendously large number of devices expected in F2C, i.e., therefore, it is not scalable. As a consequence, the authors concluded that an efficient IDMS for the IoT is still an open problem and further investigation is required.

Motivated by the need of an identity information service where the provider of the service is unable to access the information that passes through their servers, authors in [13] proposed BlindIdM, an Identity Management as a Service (IDaaS) model with a focus on data privacy protection. In such model three main type of actors are defined: users, service providers and identity providers. The user is a node in the network with the identity information of a set of entities and its goal is to transfer such information to the service provider in a secure fashion. The authors claim that through encryption techniques, BlindIdM permits to send the identity information from the user to the service provider without the identity provider being able to read it. To achieve this, the information is initially encrypted by the user, then re-encrypted by the identity provider and finally decrypted by the service provider. The results obtained during the evaluation of the proposal show assumable times for the three cryptographic operations, however, it is important to note that these operations were performed by powerful cloud data centers. Given the decentralized nature of the F2C paradigm, it is likely that some of the key functions of the control plane will be executed in the edge of the network, including the identity management service. In this sense, the three cryptographic operations proposed by the authors may cause an important bottleneck, degrading the overall system quality of service (QoS) in terms of response times.

In [14] authors introduce a user-centric identity management framework for IoT. They propose the creation of a global identity provider (gIdP), responsible for maintaining global identity. The gIdP is used by the service providers (SP) to generate local identity. However, this proposal has two major drawbacks: i) the global identity provider represents a single point of failure in the system –such centralization contradicts the F2C paradigm; ii) the proposed framework is intended to provide identities to the user rather than the devices. In F2C, regardless of whether several devices belong to the same person, every node in the network must have its own unique identifier, thus, an object-centric approach should be applied.

The work in [15] present a machine-to-machine IDMS that allows network devices to generate multiple pseudonyms to be used as identifier in different applications. They use anonymous attestation to perform verification of the pseudonym, i.e., an interactive method for one party to prove to another that the pseudonym is valid and should be accepted but without revealing anything else than the validity of the pseudonym. The problem of implementing this identity management strategy in a F2C systems is that the anonymous attestation is a set of complex mathematical expressions that the nodes have to solve in order to validate the identity of other nodes.

Thus, the calculations destined to validate the identity of other devices will add a significant delay in the connection establishments between nodes, mainly motivated by the low-computational power devices at the lowest F2C layer have.

3 IDMS Proposal

The IDMS proposal is partitioning globally unique IDs into the set of smaller fragments (fg). The fragments partitioning eases network resources to be identified by a fraction of their ID instead of the full identifier according to their position in the hierarchical F2C network as shown in Figure 2.

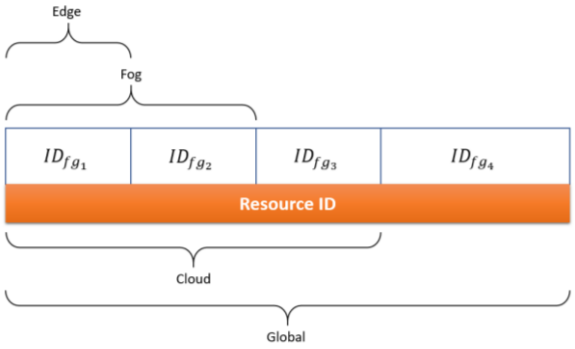

Fig. 2. Identifier fragmentation.

First of all, we define the hierarchical F2C network connection between two nodes. In F2C, the connection will be given by the node at the higher hierarchical level. According to [9], three layers are identified at early stage for the F2C system Although, the proposed three layer F2C system is not considering inter-service-provider interaction, therefore, we assume fourth layer such as follow:

- **Edge:** This F2C connection provides all occurred connection among resources (physical devices or virtual entities) under the same fog node. The resources that form an area at the edge layer are located geographically closer to each other. For example, an area at the edge can be considered as a hospital building or a school.
- **Fog:** The fog layer connection includes the connections among the fog nodes and the resources that they aggregate. An example of this connection layer can be a connection between a sensor and another device grouped under different fog nodes.
- **Cloud:** This Connection layer includes all resource connections established by the same service provider. The main difference with the fog layer connection is that resources may be located geographically far from each other. For example, resources in different cities connected by the same Internet Service Provider.
- **Global:** This connection layer is that all connections among resources stand in global concept. In this context, the resources may or may not be located close to

each other and thus inter service provider connectivity plays a key role. For example, connation between two smart cities provided by two service providers.

Figure 3 presents the four described F2C connection layer and its borders specifications. Since the number of layers in the F2C architecture may be changed, the set of F2C layer connection and the ID fragmentation policy may be changed as well to be aligned properly with the number of F2C layers. Therefore, it is worth highlighting that, this is a simple approach.

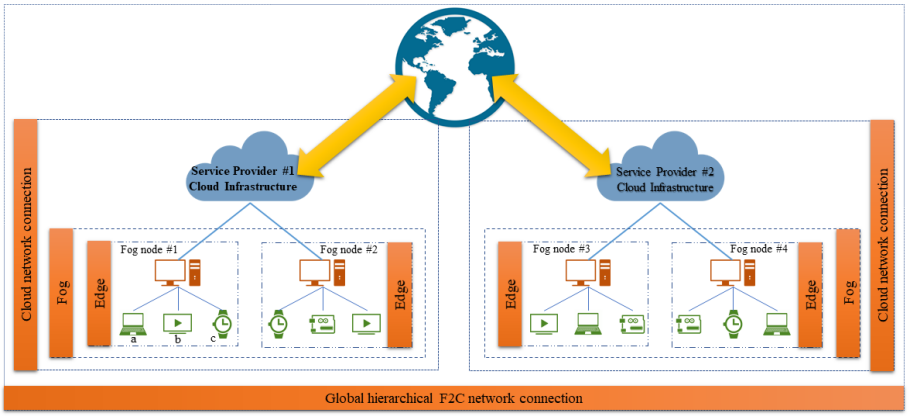

Fig. 3. Hierarchical F2C network connections.

Once the all F2C connection has been defined for F2C network topology, we divide the resource identifiers into *n* parts, where *n* is the number of F2C connections layer defined in the F2C system. Now, every time a connection between two nodes is established in the network, the nodes use a fraction of the identifier rather than using the full identifier for a mutual identification. The number of fragments to be used in each connection depends on the node at the higher hierarchical level. For example, the F2C network topology illustrated in Figure 3, the device (*b)* connects to the *Fog node #2*, such F2C connection will be set as Fog connection. Then, only two fragments of the global identifier will be utilized during the identification process.

In fact, from F2C connection and topological perspective, nodes which are located at higher layers need to use more ID fragments, and consequently, the utilized ID during the connections with other nodes will be larger. The reason for this is that nodes in higher layers have more devices as child. Therefore, to be able to identify each of these devices, longer fragments in identifiers will be required.

Regarding the fragments of identities division, we mention that according to the different use-cases and implementation needs, length of the fragments may be varied. The lower layer in a F2C system is the IoT layer. In the IoT layer, the length of the first fragment would depend on the maximum number of resource IDs that a fog node can store in cache during a given period of time, that is, the identifiers cache size. Large identifiers cache sizes in the fog nodes also entail larger identifier fragments. IoT devices might has limited resource characteristics, therefore, small cache sizes might be expected in this layer. Fog nodes can play a key role for adjusting the ID

fragment length to collision problems do not arise. Collision problem in the naming are addressed in [16], [17], and [18]. In the proposed identity management, a collision problem occurs when two or more resources in a F2C connection use the same identifier. Thus, since the purpose of IDs is identifying unambiguously a resource, the collision probability must to be reduced.

In order to enhance the IDMS security and privacy, the full resource identifier is not propagated nor stored through the network but it is only known by: i) the resource to which the ID belongs; ii) the fog node as long as the resource is connected to the F2C network through it, and; iii) other resources in a global connection that require the full resource ID for a proper identification. In short, preventing collisions during the identification process is the reason that drives nodes in a global connection to use their full ID instead of a fraction of it.

In our proposal, fog nodes play a key role because they perform IDs fragmentation and share the required resource ID fragments with other nodes according to the F2C connection layers in F2C systems.

4 Evaluation and Results

In this section we present the description of the experiment we used to validate our proposal and the results obtained. For the results, we have compared the storage required to store the resource identifiers and the queries execution times when the resources use their full identifier in the network and when they use a fraction of it, hence, two parameters have been considered during the evaluation.

In F2C, the resources grouped in the lowest layer of the network hierarchy will be the most challenging to identify. Such complexity is caused by the tremendous number of devices concentrated in the bottom of the network topology (user's devices, sensor networks and other IoT artifacts), the lack of control that the service provider will have over those devices and the highly dynamic network topology caused by the inherent mobility of many devices. Thus, recognized the aforementioned as a fact, in this section we focus in the IoT layer, hence evaluating the performance of our proposal when using the first ID fragment.

4.1 Experiment description

In the conducted experiment, we have used a Raspberry Pi 3 model B. Such device integrates an ARM 1.2Ghz quad-core processor and 1G of RAM memory. The reason for using that device is that we consider its specifications as the minimum hardware requirements that a device should meet in order to be considered for the fog node role in the F2C system.

The software we have preinstalled in the Raspberry Pi are Ubuntu 16.04 as Operating System and a SQL Database Management System (DBMS). Subsequently, we created five databases and filled them with a million of synthetic resource identifiers. The length of the resource identifier in the first database was set to 128 bytes (according with the length used in [19]). This first database was the one with the full identifi-

ers. In the next four databases a truncated version of the identifiers in the first database was stored. The IDs were truncated at 32, 16, 8 and 4 bytes respectively. In all the cases, the identifiers were generated using only the hexadecimal charset.

4.2 Used storage

In F2C, the IoT layer is the one with the most limited resources. In fact, many of the devices that operates in the lowest layer do not even have the necessary hardware resources to process the data they generate, therefore, an effective resource management is a must.

In this sense, the storage is one of the most constrained aspect of the devices in the IoT layer. A F2C framework that requires excessive storage capacity to store the data generated on runtime may disallow a large number of devices to be used as fog nodes, causing with this, in the worst scenario, that the existing fog nodes reject new connections because they are overloaded.

The storage required to store the resource IDs in the fog nodes is the first parameter we have evaluated. The results obtained during the validation (Table 1) show that truncating the identifier that the resources use in the IoT layer reduce the space in disk required to store them.

Table 1. Database sizes.

Database	Size (MB)	%
128 Bytes	162.17	100.00
32 Bytes	67.09	41.37
16 Bytes	51.08	31.50
8 Bytes	42.08	25.95
4 Bytes	37.06	22.85

Table 1 shows the size in megabytes of the databases previously described. The column in the right presents the percentage of the space required by the truncated databases with respect to the database that stores the full resource identifiers, it is, the 128 bytes identifiers.

It can be highlighted from the table that the difference in megabytes between the databases with the identifiers of 8 and 4 bytes is minimal, even when the identifiers stored in the first one are larger. This is rooted on the fact that the indexes that the DBMS uses are not in function on the length of the fields in the tables.

In all the cases, the space in disk required to store the identifiers fragments is between 58.63% and 77.15% less than the space needed to store the full identifiers.

4.3 Queries times

One of the main advantages that the F2C paradigm offers is the possibility to execute applications and services with a reduced delay than cloud computing. This opens the

door to the development and deployment of all kind of novel services that require real time responses, such as e-health services, online videogames, earthquake alarm triggers, etcetera. To achieve such goal, it is imperative that the individual components that integrate the F2C framework are highly efficient and avoid adding delays in the internal processes.

In the F2C framework, The IDMS component should be able to identify the resources in a time that allows to devices on the move to switch among different fog nodes without interrupt its activities. Such identification process includes the database lookup task. In this sense, our proposal aims at reducing the database lookup times, this by reducing the amount of information that the fog nodes store.

In the validation phase, we have used the databases described under the section 4.1 to measure the lookup times. We have measured ten times the time required to fetch among 200, 400, 600, 800 and 1,000 (thousands) records for each database and then we calculated the averages of the obtained results (Table 2).

Table 2. Queries execution times.

ID Length	IDs in the Fog Node (thousands)				
	200	**400**	**600**	**800**	**1,000**
128 Bytes	2.97 100.00%	6.29 100.00%	9.62 100.00%	12.93 100.00%	16.68 100.00%
32 Bytes	1.51 50.87%	2.49 39.65%	3.97 41.27%	4.95 38.28%	7.55 45.24%
16 Bytes	1.29 43.30%	2.34 37.24%	3.43 35.67%	4.92 38.01%	6.34 38.01%
8 Bytes	1.26 42.51%	2.20 34.90%	2.93 30.43%	4.42 34.16%	5.52 33.10%
4 Bytes	1.16 38.99%	1.91 30.31%	3.14 32.62%	3.98 30.80%	5.02 30.11%

Table 2 and Figure 4 summarize the results obtained. For the sake of comparison, percentage related to the first database are also included in Table 2. As it can be observed, using a fraction of the full resource identifiers reduces significantly the time required to search an item in the database. By using a quarter of the name of the devices, our proposal has shown a reduction of up to 49.13% in the search time. In fact, a 32 bytes ID is still a large identifier for the lower F2C layer, which means that the ID length can be reduced even more and with it, also the search time.

It's worth noting that in general, the times obtained when using 8 and 4 bytes identifiers are very similar. This means that the time behaves exponentially, what is justified by the management of indexes and primary keys used by the DBMS to improve the data retrieval process.

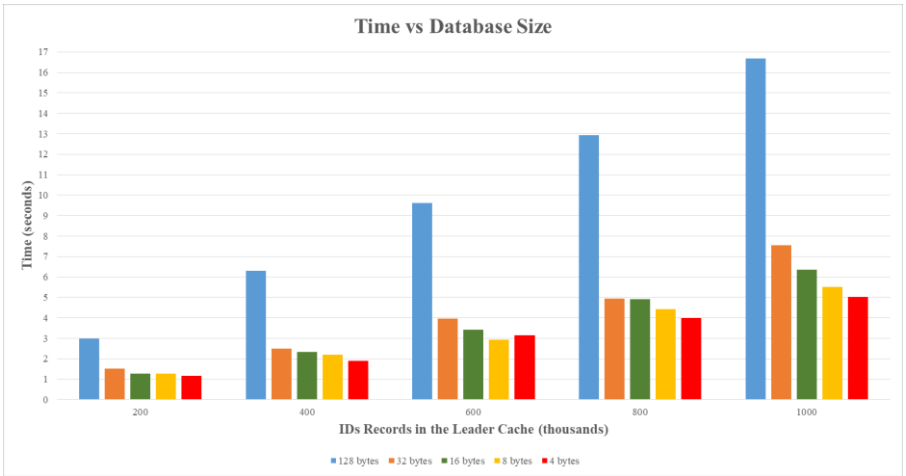

Fig. 4. Queries execution times.

In Figure 4, the queries execution times are presented graphically. The blue bars represent the lookup times in the database that stores the full resource identifiers. It can easily be observed that in all the cases the time required to search in such database are considerably longer than the queries execution times when the resources use a fraction of their full ID. In this figure, the exponential behavior of query execution times can be observed more clearly. This trend becomes more evident as the volume of data to be handled increases.

From the results shown in the Table 2 and Figure 4, we can conclude that when the edge devices use a fraction of their full identifier instead of the full version of it, the lookup time decreases significantly (between 54.76 and 69.89% for large volumes of data), all of this, without affecting the ID uniqueness property, it is, keeping a very low collision probability.

5 Conclusions and Future Work

The F2C compute paradigm have arose as a novel solution that intends both, to manage the resource continuum from the edge of the network to the cloud datacenter and to solve some of the cloud inherent limitations, such as the possibility of offering remote resources at the edge with a reduced latency to be used by delay sensitive services that require real time responses. However, there is still a list of open challenges that must be addressed before we can have a F2C framework that can be deployed. One of those challenges is the management of the resources identities in the network, especially, in the lower hierarchical layer, where most of those resources will be concentrated.

In this paper we propose a strategy to manage the identity of the resources that consists of fragmenting the unique global resource ID into smaller fragments. Each time a connection to a resource is established, the fog node that aggregates the re-

source to the network will determine the connection scope and thereafter, the number of fragments required for a mutual unambiguous identification.

The results obtained during the proposal validation phase show that the implementation of our proposal allows to reduce both, the space in disk required to store the resource identifiers in the fog nodes and the query execution times, achieving with this, a more efficient use of resources in the IoT layer and streamline the resource identification process.

Future work in this topic includes to implement this proposal in a real scenario to validate its effectiveness in the whole F2C environment and to propose an algorithm that allows to determine the optimal fragment lengths for each level in the network hierarchy.

Acknowledgment

This work is supported by the H2020 mF2C project (730929), by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund both under contract TEC2015-66220-R (MINECO/FEDER), and for Alejandro Gómez-Cárdenas by the Consejo Nacional de Ciencia y Tecnología de los Estados Unidos Mexicanos (CONACyT), under grant No. 411640.

References

1. Dave Evans: The Internet of Things: How the Next Evolution of the Internet is Changing Everything, (2011).
2. Andreas Burkert: Modern Cars' Insatiable Appetite for Data, (2017).
3. Mehdipour, F., Javadi, B., Mahanti, A.: FOG-Engine: Towards Big Data Analytics in the Fog. In: 2016 IEEE 14th Intl Conf on Dependable, Autonomic and Secure Computing, 14th Intl Conf on Pervasive Intelligence and Computing, 2nd Intl Conf on Big Data Intelligence and Computing and Cyber Science and Technology Congress(DASC/PiCom/DataCom/CyberSciTech). pp. 640–646 (2016).
4. Ferrer-Roca, O., Roca, D., Nemirovsky, M., Milito, R.: The Health Fog. Small Data on Health Cloud. Presented at the The International eHealth, Telemedicine and Health ICT Forum for Educational, Networking and Business, Luxembourg April 23 (2015).
5. Mohamed Firdhous, Osman Ghazali, Suhaidi Hassan: Fog Computing: Will it be the Future of Cloud Computing? Presented at the Proceedings of the Third International Conference on Informatics & Applications, Kuala Terengganu, Malaysia (2014).
6. Xavi Masip-Bruin, Eva Marín-Tordera, Admela Jukan, Guang-Jie Ren, Ghazal Tashakor: Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud (F2C) computing systems, (2016).
7. OpenFog Consortium: OpenFog Reference Architecture for Fog Computing, USA (2017).
8. mF2C Consortium: mF2C Project, http://www.mf2c-project.eu/.
9. Sarkar, S., Misra, S.: Theoretical modelling of fog computing: a green computing paradigm to support IoT applications. IET Networks. 5, 23–29 (2016).

10. Shi, W., Cao, J., Zhang, Q., Li, Y., Xu, L.: Edge Computing: Vision and Challenges. IEEE Internet of Things Journal. 3, 637–646 (2016).
11. European Telecommunications Standards Institute: Corporate telecommunication Networks (CN); User identification in a SIP/QSIG environment, (2004).
12. Cao, J., Xu, L., Abdallah, R., Shi, W.: EdgeOS_H: A Home Operating System for Internet of Everything. In: 2017 IEEE 37th International Conference on Distributed Computing Systems (ICDCS). pp. 1756–1764 (2017).
13. Nuñez, D., Agudo, I.: BlindIdM: A privacy-preserving approach for identity management as a service. International Journal of Information Security. 13, 199–215 (2014).
14. Chen, J., Liu, Y., Chai, Y.: An Identity Management Framework for Internet of Things. In: 2015 IEEE 12th International Conference on e-Business Engineering. pp. 360–364 (2015).
15. Fu, Z., Jing, X., Sun, S.: Application-based identity management in M2M system. In: 2011 International Conference on Advanced Intelligence and Awareness Internet (AIAI 2011). pp. 211–215 (2011).
16. S. Farrell, D. Kutscher, C. Dannewitz, B. Ohlman, A. Keranen, P. Hallam-Baker: Naming Things with Hashes, (2013).
17. Bouk, S.H., Ahmed, S.H., Kim, D.: Hierarchical and hash based naming with Compact Trie name management scheme for Vehicular Content Centric Networks. Computer Communications. 71, 73–83 (2015).
18. Savolainen, T., Soininen, J., Silverajan, B.: IPv6 Addressing Strategies for IoT. IEEE Sensors Journal. 13, 3511–3519 (2013).
19. Gómez-Cárdenas, A., Masip-Bruin, X., Marín-Tordera, E., Kahvazadeh, S., Garcia, J.: A Hash-Based Naming Strategy for the Fog-to-Cloud Computing Paradigm. In: Heras, D.B., Bougé, L., Mencagli, G., Jeannot, E., Sakellariou, R., Badia, R.M., Barbosa, J.G., Ricci, L., Scott, S.L., Lankes, S., and Weidendorfer, J. (eds.) Euro-Par 2017: Parallel Processing Workshops. pp. 316–324. Springer International Publishing, Cham (2018).