A Hash-Based Naming Strategy for the Fog-to-Cloud Computing Paradigm

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

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

Abstract. The growth of the Internet connected devices population has fuelled the emergence of new distributed computer paradigms; one of these paradigms is the so-called Fog-to-Cloud (F2C) computing, where resources (compute, storage, data) are distributed in a hierarchical fashion between the edge and the core of the network. This new paradigm has brought new research challenges, such as the need for a novel framework intended to controlling and, more in general, facilitating the interaction among the heterogeneous devices conforming the environment at the edge of the network and the available resources at cloud. A key feature that this framework should meet is the capability of uniquely and unequivocally identify the connected devices. In this paper a hash-based naming strategy suitable to be used in the F2C environment is presented. The proposed naming method is based on three main components: certification, hashing and identification. This research is an ongoing work, thus, the steps to follow since a device connects to the F2C network until it receives a name are described and the major challenges that must be solved are analyzed.

Keywords: Naming, Identification, Fog-to-Cloud, Internet of Things.

1 Introduction

In simple words, the Internet of Things (IoT) is a communication paradigm where all kind of everyday objects are capable to connect to the Internet network with different purposes. This paradigm allows the creation of a range of new services and applications in diverse areas like smart homes, buildings and cities, eHealth, vehicular networks, wearables, monitoring and surveillance, etcetera. It is estimated that by 2020 the worldwide population of Internet connected objects will reach 50 billions [1].

Taking advantage of the large number of devices with network connectivity and in consideration of the expected growth, new computer paradigms have emerged, one of them is the Fog-to-Cloud (F2C) computing.

F2C is a collaborative and distributed compute model where resources (like storage, compute or data) are located in a hierarchical fashion not only at the core of the network but also at the edge [2]. In many cases, the resources conforming the F2C at the edge of the network are supplied by the end users, thus, users can not only access to the service provider or third parties resources but also share their own resources.

Being a hierarchical model, the resources are deployed in a bottom-up fashion, usually with the most constrained devices in the lower layer (very basic sensors and actuators) and in the top of the hierarchy a virtually unlimited resource data center: the cloud.

Many research efforts are focused in the design of a suitable F2C architecture [3] for managing the distributed storage, compute, data, control and networking functions.

A key functionality that any F2C architecture must meet is the capability to identify uniquely and unequivocally every device connected to the F2C network, thus, the adoption of a naming strategy is required.

The list of available naming schemes is not short [4] [5] and ranges from the use of existing services like the Domain Name Service (DNS) to the redesign of the computer networks as are known nowadays to a not host-based-centric network. The problem with those naming schemes is that most of them doesn't meet the inherent F2C requirements (such as interoperability, mobility, uniqueness and scalability) or the effort to implement them is far beyond the scheme itself.

Regardless the application specific requirements, according with [6] a good naming service should meet the three characteristics described in the "Zooko's Triangle": decentralization, human-meaningful names and secure mapping of names. These three design goals are represented as a side of the triangle and each side represents a design tradeoff, so according with the original author, it isn't possible to have the three characteristics at the same time.

In this paper a new distributed hash-based naming strategy that meets the aforementioned F2C requirements is presented. The proposed strategy is based in three support modules which are: certification, hashing and identification.

The remainder of this paper is organized as follow: In section 2 the hashing technique is discussed and similar works are reviewed. In section 3 the proposed hash-based naming strategy and its support modules are explained in detail. In section 4 the key advantages of the proposal are studied. Finally, in section 5 the research conclusions are exposed.

2 State of the Art

In this section the hash functions and its properties are briefly described, also an example of a hash string value is shown and three distinct works where authors have used a hash-based method for naming entities (virtual or physical) are analyzed.

2.1 Background

The hashing is a cryptographic technique widely used to map a data block of variable size to a fixed-length output. It means that it does not matter whether the input is 1 byte or 1 terabyte, the output will be a string with a predefined size length. In [7] the hash technique is described in function of its main properties, which are:

- Variable input size. In the hash function h(x) where x is the input, the size of x does not matter at all.
- Fixed length output. As said before, the output size isn’t in function of the input size.
- Compute facility. It is relatively easy to compute h(x) for any given x.
- One-way. For any given y, it is computationally infeasible to find x such that h(x) = y, what means that it cannot be “unhashed”.
- Collision resistance. It is computationally infeasible to find y != x such that h(x) = h(y). It means that two different inputs always produce two different outputs and vice versa, two different outputs always belong to two different inputs.

There are many algorithms designed to implement the hash function, the most popular are briefly reviewed in [8]. In the United States as well as much of the world, the MD5 and SHA algorithms are the most widely used [7], nevertheless any other algorithm that fulfill the listed properties is suitable for generating hash values.

In order to illustrate better the properties of the hash function three examples are presented below (table 1). In the examples, the SHA-1 algorithm is used to hash three different strings.

Table 1. Hash transformation examples using the SHA-1 algorithm.

First Example	
Input	Hello
Hash value	f7ff9e8b7bb2e09b70935a5d785e0cc5d9d0abf0
Second Example	
Input	hello
Hash value	aaf4c61ddcc5e8a2dabede0f3b482cd9aea9434d
Third Example	
Input	Hello World! This is a test
Hash value	cf3491c6524b19f1965b112c37e5360e6920a136

The input in every example is different. While in the first two examples the inputs difference is very subtle (only changes the first letter, from capital “H” to “h”), in the third example the input is not a word but a phrase. In the three cases the hash value output always is a totally different 160-bits string.

The hashing is a destructive process, what means that there is not a way to return to the original input starting from the hash output value.

2.2 Related Work

Some researchers have found in the properties of the hash function an opportunity for the creation of new naming schemes. In [9] the authors review extensively the use of the hash function for naming objects. They claim that in order to avoid collisions the SHA-256 algorithm must be implemented. However, in constrained environments or in scenarios where a higher collision probability can be tolerated, the system administrator can opt for using a truncated version of the hash function output. In no case they recommend the use of names with less than 100-bits; they assert that in those scenarios the collision resistance property cannot be guaranteed.

In the previously cited publication the authors use the SHA-256 algorithm to include a hash string as a segment in Universal Resource Locators (URL). With the purpose of standardize the uses of hash outputs in URLs, they specify a new URI scheme and a way to map these to URL's, however, their proposed method lacks of a clear hash input proposal. Although they mention that public keys are a good hash function input candidate, they let the users to choose the input value, so in scenarios where the user not only select an inappropriate input but also decide to use the truncated hash function output, the collision probability could be very high.

In [10] a hybrid naming scheme for vehicular content centric networks is presented. In the proposal the authors divide the content name (CN) into three parts: the scheme, the prefix and the hash. The first part is the naming scheme identifier that is used to represent CN. This field can take two different values in function of the used protocol.

The prefix is the hierarchical part of the name scheme and is used to identify the content originating node that is a vehicle and the content itself in a human-readable format. The distinct parts of the prefix are separated by a slash ("/") and the firsts four fields are reserved for the publisher vehicle's information. The rest of the hierarchical section signifies the information about the digital content (e.g. text, video, image, or any other digital content).

Finally, the hash section of the CN corresponds to the full or truncated hash value generated using the digital content, the content attributes or the public key of the information related to it.

According with the authors, the last field is used to uniquely identify the content item. Nevertheless, in a scenario in which two or more vehicles are sharing distinct contents but with the same attributes, if the hash function input are those attributes, the probability of having duplicated records making reference to different contents will be high. The solution to this problem could be to increase the number of attributes in the prefix section, but this decision will impact the lookup throughput, what in content centric networks is critical.

Another similar work is the presented in [11]. In their work the authors does not propose a naming strategy but a name resolution scheme consisting of two parts: name mapping and name resolution.

Basically, what they do is to use a hash function to translate the heterogeneous device name to a fixed-length string, hiding like this the original name from the outside Internet for security and privacy reasons.

In the name mapping the object name is received and translated to a 160-bits string using the SHA-1 hash function algorithm. A notable drawback that this strategy presents is that to be recognized by the system the user has to hash the object name and register the resulted string to the resolution system in advance. This could be a tedious task for users owning multiple devices.

Another weak point of this strategy is that two devices with the same name will have exactly the same hash output value and as result, duplicate register may exist in the resolution adopted scheme (DNS or DHT).

3 Proposed Naming Strategy

In this section the naming strategy is presented. The proposed scheme consists of three main components: a certification, a hash function and the identification module. The technique described in the next lines aims to be a part of the resource management functions (figure 1) in a F2C environment.

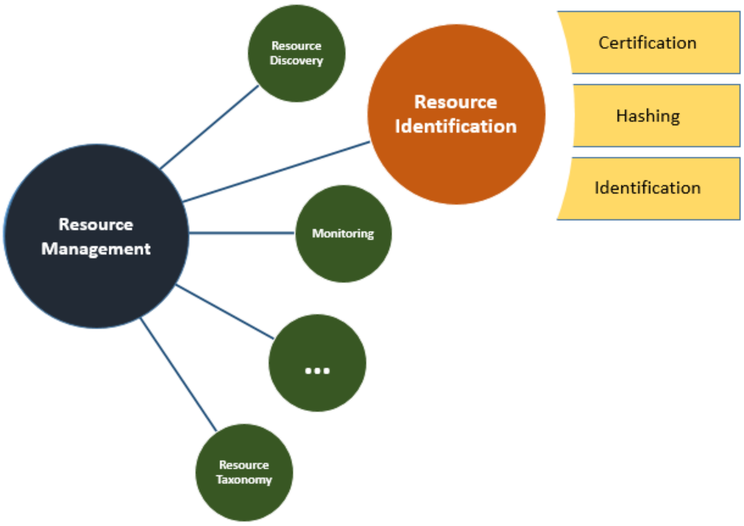

Fig. 1. Resource Identification strategy for the F2C Architecture.

3.1 Certification

The certification is the very first step that users must complete to have their device(s) connected to the F2C network. In this phase the users register his personal information in the system to get a secret key.

This registration process must to be done once per entity (person, institution or company) regardless the number of devices the entity wants to use in the F2C ecosystem. For example, if a government department needs to deploy thousands of devices through a specific area in the city, the institution only have to register once to get the secret key. This process is shown in the figure 2.

The implementation of this first phase will bring new challenges that must to be solved. The major challenges are related with the system security. There is a lot of attacks that the system must not only to resist but also to detect.

In the certification phase as well as in the other two components of the identification strategy to provide a secure communication channel that discard the risk of interception is a crucial requirement.

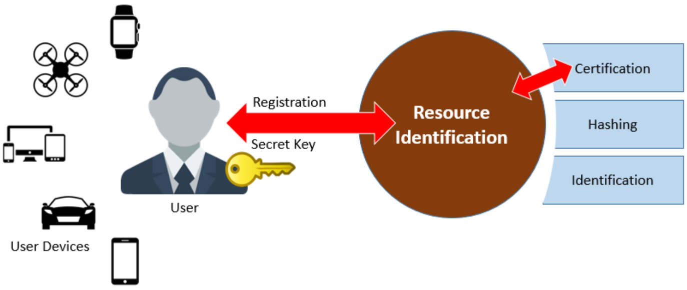

Fig. 2. Certification phase in the Resource Identification strategy

Apart from security, other considerations must to be taken into account in the certification, for example, due the key assignation will be a distributed process, a mechanism to disallow secret key overlapping will be necessary. Also, the system must to be able to suspend, revoke and update the user secret key.

3.2 Hash Function

Being the module that stores the naming scheme, the hash function is the core of the identification strategy. This function is the responsible of transforming the device identification input into a hash string.

The device identification input is composed by two concatenated string. The first string is the user secret key obtained during the registration phase while the other is an "optional" user string (figure 3).

Due the purpose of the second part of the string is to differentiate among the user devices, it will be optional only in those cases where the user owns or wants to use only one of his devices in the F2C network, otherwise it will be a mandatory field.

In [12] the author explains that it is nearly impossible to have one global naming convention mainly because industries have been using their own proprietary naming conventions for long time and migrating to a different naming convention will impact their infrastructure considerably. Nevertheless, this method does not force the users to abandon their own internal naming convention. In the second part of the string the user can use whatever value they want regardless the length.

Continuing with the previous government department example, let's assume that for internal reasons the institution uses a hierarchical naming convention that includes the city, a code area where the device is located and at the end a consecutive number. The internal records will look something like this: LA347-01, LA347-02, LA347-XX. The adoption of this or any other naming scheme won't affect the string conversion process.

Once the user identification string has been transformed into the final hash value it is stored in key nodes across the F2C network using Distributed Hash Tables (DHT). A full backup of the records always is kept in a cloud data server.

As well as in the previous step, the hash function module also have some challenges that must to be addressed. One of the biggest challenges is the need of an incentive that encourage users to keep using exactly the same string in both sides of the hash input. This is particularly important for the implementation of long term identification mechanisms and other historical functions because as was shown in Table 1, the minimum change in the input string will change dramatically the device identifier / name.

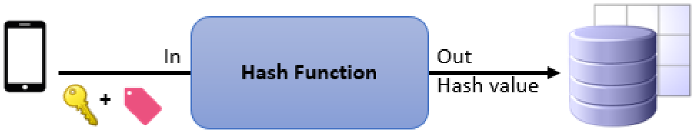

Fig. 3. Hash function process.

3.3 Identification

The last step in the proposed strategy is to look up for the hash value of the device in the DHT, it is the identification.

In this point there could be three distinct scenarios:

- The device is new in the system. When a device connects for the very first time to the F2C network the look up in the DHT won't find any coincidence. In that case, the system should register the device and perform other assistant tasks (e.g. device characterization) in order to recognize it in future interactions. The device will be registered in the DHT closer to the device physical location and after x seconds, this and other new records will be propagated to the upper F2C nodes in a hierarchical fashion, until the record(s) reach the cloud where will be stored for a long term.
- The device is connected to the F2C network in a known location. The DHT will store for a predefined period of time a cache with all the devices that have been connected in the last x days, so when a device reconnects to the same F2C node, it will recognize the device without going to an upper level to look up for the device information. In this process the network hierarchy will be leveraged. When a device is connected to a different but still close node from the habitual one, it won't have to go to cloud to have the device information; going to one layer higher will be enough (figure 4).
- The device is connected in a distinct location than the habitual. Let's assume a three layered F2C network; if the device is connected in a distant location and there is not information available in the same layer or even in the next upper layer, there is still the cloud database, what in terms of costs will be cheaper that characterize again the device.

In the figure 4 the "mobile device" uses to connect to the F2C network through the nodes "A" and "B", so every time it connects using one of those nodes the system automatically detects and retrieves all the device information. In the case that the node connects for the first time or after a long time to the F2C using the node "C" where there is not information available about this device, the system will search in the next upper layer for information about it. In the node "J" a copy of all the device information is kept and updated for the nodes "A" and "B".

Now, let's consider that the "mobile device" moved to the area of the aggregator node "H" and connects to the system using that node. Being the first time in this zone there is not information available about the device, neither at the node "H" nor at the next layer (node "K"), however, the node "J", aggregator of the nodes "A", "B", "C" and "D" registered the device in the cloud database so there is information about it that the node "K" can access anytime and thus, the node "H".

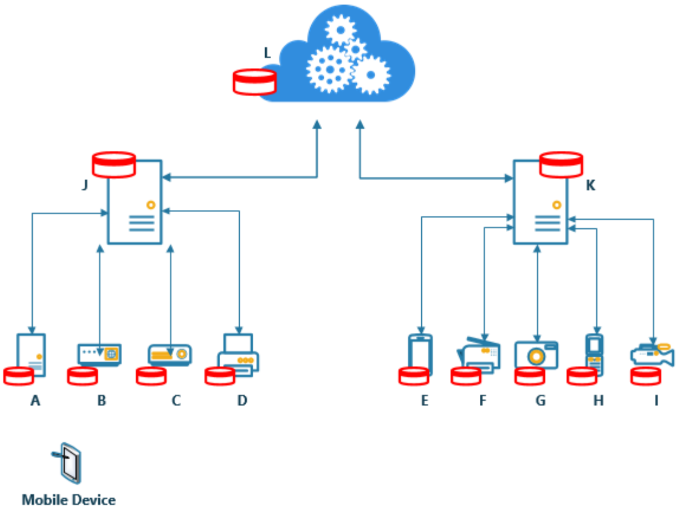

Fig. 4. Three layered Fog-to-Cloud network topology.

In this third step the main two problems that may arise are a poor throughput in the lookup process and a high network overhead caused by the mobile devices. Nevertheless, if suitable policies are applied these two problems can be overcome.

4 Proposal Advantages

The key advantages of the proposed naming strategy are the capability of assign worldwide unique names to the devices connected to the F2C network. In the case of two or more F2C providers sharing the certification and hashing modules, the device not only will use the same secret key to be identified but also it will keep the name regardless the system provider.

The use of the Distributed Hash Tables will facilitate the implementation of new functionalities in the system, such as a trust system, where the devices can get a classification in function of its availability, uptime, and other parameters. Other function that the historical information stored in the DHTs will allow to implement in the platform is a predictive resource utilization / available system without expose the device specs or location.

Finally, if the hash function is implemented correctly, the possibility of a duplicated name will be minimal, what means that the proposed strategy is secure.

5 Conclusions

In this research work an integral hash-based naming strategy suitable for the Fog-to-Cloud environment was proposed. The strategy is conformed for three main modules: certification, hashing and identification. The proposal meets the F2C requirements, such as mobility, scalability, security, privacy and uniqueness.

Even when the proposed naming strategy presents important advantages in comparison with other naming strategies and schemes there are still open challenges that must to be addressed. Those challenges include the need of provide a secure channel for the communications among edge devices and F2C nodes, a mechanism that disallow the secure key overlapping, the DHT lookup throughput, etcetera.

In order to solve the mentioned issues more research effort in every component of the proposed method is needed, so the future work will be focused in overcome the existing challenges.

Acknowledgement

This work is supported by the H2020 mF2C project (730929), as well as by the Spanish Ministry of Economy and Competitiveness and by the European Regional Development Fund under contract TEC2015-66220-R (MINECO/FEDER).

References

1. S. K. Datta, R. P. F. Da Costa, C. Bonnet (2015) Resource discovery in Internet of Things: Current trends and future standardization aspects. In: 2015 IEEE 2nd World Forum on Internet of Things (WF-IoT). pp 542–547
2. X. Masip-Bruin, E. Marín-Tordera, G. Tashakor, et al (2016) Foggy clouds and cloudy fogs: a real need for coordinated management of fog-to-cloud computing systems. IEEE Wireless Communications 23:120–128. doi: 10.1109/MWC.2016.7721750
3. OpenFog Consortium (2017) OpenFog Reference Architecture for Fog Computing.
4. European Research Cluster on the Internet of Things (2014) EU-China Joint White Paper on Internet-of-Things Identification.
5. Yijian Li, Raj Jain (2013) Naming in the Internet of Things.
6. Craig Webster (2011) WebNS: Model for a Peer-to-peer Name Service. Monash University
7. Easttom C (2015) Modern Cryptography: Applied Mathematics for Encryption and Information Security. McGraw-Hill Education
8. Kamlesh Kumar Raghuvanshi, Purnima Khurana, Purnima Bindal (2014) Study and Comparative Analysis of Different Hash Algorithm. JECAS 3:
9. Farrell S, Dannewitz C, Ohlman B, et al (2013) Naming Things with Hashes. doi: 10.17487/rfc6920
10. Bouk SH, Ahmed SH, Kim D (2015) Hierarchical and hash based naming with Compact Trie name management scheme for Vehicular Content Centric Networks. Computer Communications 71:73–83. doi: https://doi.org/10.1016/j.comcom.2015.09.014

11. Z. Yan, N. Kong, Y. Tian, Y. J. Park (2013) A Universal Object Name Resolution Scheme for IoT. In: 2013 IEEE International Conference on Green Computing and Communications and IEEE Internet of Things and IEEE Cyber, Physical and Social Computing. pp 1120–1124
12. Sandoche Balakrichenan (2016) Why DNS should be the naming service for Internet of Things?