2017 31st International Conference on Advanced Information Networking and Applications Workshops

DNSNAv4: DNS Name Autoconfiguration for Internet-of-Things Devices in IPv4 Networks

Keuntae Lee*, Seokhwa Kim*, and Jaehoon (Paul) Jeong†
* Department of Computer Science & Engineering, Sungkyunkwan University, Republic of Korea
† Department of Interaction Science, Sungkyunkwan University, Republic of Korea
Email: {keuntaelee, seokhwakim, pauljeong}@skku.edu

***Abstract*—This paper proposes a Domain Name System (DNS) Name Autoconfiguration called DNSNAv4 for Internet-of-Things (IoT) Devices in Internet Protocol (IP) version 4 (IPv4). The manual configuration of domain names of devices might be a cumbersome burden for users as the number of devices increases. A legacy scheme called DNS Name Autoconfiguration (DNSNA) for IPv6 networks can be used to register DNS names of IPv6 IoT devices automatically into a DNS system. However, the legacy scheme called DNSNA is not applicable in the current networks because most of networks in the Internet are mostly set up for IPv4. In this paper, we propose DNSNAv4 to support suitable functionality to let the legacy scheme of DNSNA provide DNS naming services to IoT devices in the IPv4 environment. In DNSNAv4, an IoT device can register devices by interacting with a dynamic host configuration protocol (DHCP) server so that the devices can register their DNS names and the corresponding IPv4 addresses into a DNS server through a DHCP server. Therefore, with the standard protocol of DHCP, DNSNAv4 can provide users with easy DNS name autoconfiguration of their IoT devices in the IPv4 environment.**

I. Introduction

The Internet of things (IoT) is the network of physical devices, vehicles, building systems, and various embedded devices with electronics, software, sensors, and actuators that enable these objects to collect and exchange data. Recently, the IoT has become one of the hottest research fields. According to Gartner, the number of IoT devices is expected to increase by almost 26 billion units by 2020 [1]. By a large number of devices, it is inefficient to manually configure their domain names in the Domain Name System (DNS), which allows DNS servers to translate between domain names and Internet Protocol (IP) addresses.

Lee et al. proposed a DNS Name Autoconfiguration (DNSNA) scheme for IoT devices [2]. Based on this, DNSNA has been designed to provide all kinds of IoT devices with an efficient DNS naming service. However, the current internet are mostly set up for IPv4 networks. To switch internet from IPv4 to IPv6, it takes a lot of cost such as money and time something like that. Clearly, we should use IPv6 networks lines in the future but it cannot right now. Sure, we can use IPv6 network in the IPv4 network environment through such as tunneling and address translation. But it is inefficient method because a technology to link IPv4 and IPv6 has more overhead better than the internet operates as pure IPv6 networks. That is why DNSNA is difficult to be applied to the currently network system. Therefore, we need a new scheme of DNSNA for IPv4 network environment. This paper proposes a DNSNA version 4 (DNSNAv4) for IoT devices in IPv4 network environment. For the applicability in the current network system, we use a standard protocol which is Dynamic Host Configuration Protocol (DHCP) version 4 (DHCPv4) [3]. The idea of DNSNAv4 is to use the existing protocols and techniques of DHCPv4 [3]. An IoT device requests an IP address to a DHCPv4 server. When the DHCPv4 server receives the the message of request, the DHCPv4 server replies to the request with an IPv4 address and DNS search list (DNSSL) [4]. If an IoT device receives the DNSSL, the IoT device generates its own DNS name with DNSSL and its device information. After the own DNS name generation, the IoT device requests the registration of its DNS name and corresponding IPv4 address to the DHCPv4 server. The DHCPv4 server updates the information of DHCPv4 client that is the IoT device when the DHCPv4 server receives the request message from the DHCP client. Before the update, the DHCPv4 server checks if the message is valid or not. If the message is valid, the DHCPv4 server updates the information of the IoT device that includes its the IPv4 address and the corresponding DNS name. The contributions of this paper are as follows:

- First, we suggest the requirements of the efficient DNS naming services for managing IoT devices efficiently in the IPv4 network environment.
- Second, we propose an architecture of DNSNAv4 for the efficient DNS naming services of IoT devices in the IPv4 network environment.
- Third, we augment the existing DHCP so that it may not require the significant changes of the DHCP in current network environment.
- Fourth, we provide a DNS name conflict resolution procedure for the DHCP as an extension.

The rest of this paper is organized as follows. Section II summarizes related work for IoT DNS naming. Section III describes the design and protocol of our DNSNAv4. Section IV explains an experiment in our testbed. In Section V, we conclude this paper along with future work.

II. Related Work

The IoT has been realized and will be popularly used for a variety of Internet services. The number of IoT devices in use has been increasing steadily. The naming system is an essential element to identify each IoT device in any network where several devices have the same manufacturer and device type. IoT devices have numerous connections between other

978-1-5090-6231-7/17 $31.00 © 2017 IEEE
DOI 10.1109/WAINA.2017.117

devices. In this environment, in order to create a variety of applications and services, security is an important factor.

The oneM2M technical specifications for a common M2M service layer along with an M2M device identification scheme standardized an object identifier (OID) [5] and suggested globally unique IDs for use in oneM2M. An M2M node ID consists of a higher arc (M2M node indication ID) and a sequence of four lower arcs (manufacturer ID, model ID, serial ID, and expanded ID), and the arcs are placeholders for the device's identification and description in the hierarchal tree of globally unique IDs. The M2M node indication ID (the higher arc) represents a globally unique identifier for the M2M node. The higher arc can be composed of several sub-arcs, which are variable. The higher arc is assigned and managed by International Telecommunication Union Telecommunication Standardization Sector (ITU-T) and International Organization for Standardization/International Electrotechnical Commission (ISO/IEC).

Bonjour [6] is Apple's implementation of zeroconfiguration networking to support DNS naming and service discovery. Bonjour can support the search and registration of apple devices in order to announce the services of an Apple device. Bonjour is based on mDNS [7] which is used by every device as both a DNS resolver and a DNS server. mDNS is used as a carrier protocol for DNS based service discovery and no authoritative DNS server is not used for name resolution. mDNS uses multicast for name resolution in a local network, however, it is not suitable to a multi-link network because of many traffic. mDNS is a privacy issue because it supports a naming without the security key. DNS based service discovery protocol (DNS-SD) [8] is used to perform service discovery by Bonjour software. DNS-SD can provide a list of service devices and port numbers for hosts.

Lee et al. proposed a DNS Name Autoconfiguration (DNSNA) scheme for IoT devices [2]. Based on this, DNSNA has been designed to provide all kinds of IoT devices with an efficient DNS naming service. DNSNA [2] was designed to operate in a variety of IoT environments having various sensors ranging from low capacity to high capacity. It consists of the following four steps. The first step is DNS name generation. The second is DNS name collection. The third is DNS name registration. The fourth is IoT device list retrieval. DNSNA uses either the router advertisement (RA) DNS option [4] over IPv6 neighbor discovery (ND) [9] or the DNS option over DHCP version 6 (DHCPv6) [10]. Once it obtains a DNSSL [4], the target network device autonomously constructs its DNS name using the DNSSL and its device information, and then registers its DNS name into a DNS server via a router in the same subnet. An IoT device generates a DNS name from the DNSSL in the DNS option(s). The IoT device checks the uniqueness of the the generated DNS name by performing duplicate address detection (DAD) for its IPv6 address corresponding to the DNS name in the subnet where it exists. If there is no reply for this DAD, the IoT device regards the DNS name as unique such that the other nodes in the subnet do not use the DNS name, so it configures the DNS name as its own. Otherwise, it repeats the generation of another DNS name and the execution of the DAD until the request for the DAD is not responded by any other node. A router sends node information (NI) query to collect the pair of the DNS name and the corresponding IPv6 address of each IoT device. The router checks if the collected DNS names of the IoT devices are registered in the DNS server. But, as mentioned in the previous section, DNSNA is not applicable the current network lines because of current many physical network lines are set up for IPv4 networks. For apply current network system, we need to a new scheme of DNSNA in the IPv4 network environment.

III. Proposed DNSNAv4

This paper is the another version of DNSNA in [2]. Fig. 1 shows a system configuration for DNSNAv4. To explain the process of DNSNAv4, a mobile device, an IoT device, a router, and a DNS server are organized as shown in Fig. 1.

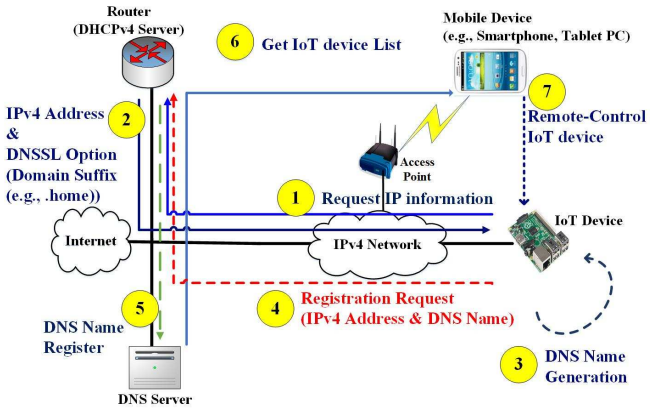

Fig. 1. A System Configuration for DNSNAv4

Since the latest routers also function as DHCPv4 servers, we assume that all routers and DHCPv4 servers used in this paper are the same equipment. First, an IoT device requests an IPv4 address to the DHCPv4 server in the subnet. The DHCPv4 server sends information with an IPv4 address and through a DNSSL option of DHCP which is DNS suffix information(e.g., .home) [3] [4]. The IoT device generates a DNS name for DNS service when the IoT device receives the reply message, if it is valid. After the IoT device generates the its DNS name, IoT device sends a request message to register the its DNS name. Then, the DHCPv4 server checks the DNS name if it is valid or not, when the DHCPv4 server receives a request message about to register. If the message is valid, DHCPv4 server updates the corresponding DNS name through DHCP [3]. Finally all configuration done, a user can get a DNS name list and remote-control IoT devices through own mobile device(e.g., smartphone, table PC). But, if the message is not valid or the DNS name is duplicate, DHCP does not specifically provide resolution about this problem [11] [12] [13]. Therefore, we need a new DHCP option because the goal of our scheme is to provide autoconfiguration with minimize user interaction. So about this issue, this paper is written about that in Section III-D

A. DNS Naming Service

The ultimate goal of DNSNA scheme is to support an IoT DNS naming service that is convenient for users, allowing them to easily distinguish the DNS names of their IoT devices for monitoring and remote-control in their local network or across the Internet [2]. To provide this service, either text-based or graphic-based display can be used. Text-based display shows the list of IoT devices registered into the DNS server, and the user can see the detailed device information by clicking a device listed on the display [2]. The device information can be registered into a service resource record in a DNS zone file for a service. The service name along with the device information rather than the service resource record itself is shown because the user does not need to see the detailed, intricate service resource. Graphic-based display is used for searching and identifying IoT devices in a user-friendly way. For example, if there are two air conditioners in a user's home, the name of the air conditioners may be identical or similar to each other. Instead of differentiating them with distinct IPv4 addresses, graphic-based display can differentiate the DNS names of the air conditioners them with different locations (e.g., bedroom and living room).

B. DNS Name Format

The proposed DNS name format uses a more hierarchical structure using OIDs as an IoT naming service, which is defined in DNSNA [2]. We can apply an oneM2M OID [5] as part of the DNS name format for an IoT device on the Internet. ITU-T and ISO/IEC developed OIDs to assign a globally unique ID to an M2M node. The information contained in the DNS name format in Fig. 2 is described below:

```
unique_id.object_identifier.location.domain_name
```

Fig. 2. DNS Name Format Using Object Identifier

- **unique_id** is an identifier that guarantees the uniqueness of the DNS name in ASCII characters. The identifier can be a sequence number or alphanumeric with readability, such as a product name. For example, a unique_id for a TV could be TV1.
- **object_identifier** is a composite ID that consists of an M2M node indication ID (such as the concatenation of the managing organization, administration, data country code, and M2M node), manufacturer ID, model ID, and serial number ID. For example, 0.2.481.1.100.3030.10011 is an OID [5] where 0 is the managing organization ITU-T, 2 is the administration, 481 is the data country code for Korea 1 is an M2M node, 100 is the node manufacturer, 3030 is the node model, and 10011 is the node serial number.
- **location** is a device's physical location, including the macro location (e.g., living room in an apartment) and micro location (e.g., the center, left-bottom wall, or right-upper corner of the living room), as well as a concatenation of micro and macro locations.
- **domain_name** is a DNS domain name (e.g., skku.edu or home) encoded according to the specification of the representation and domain name use [10].

If the DNS name of an IoT device includes the location information, users can easily identify the physical location of each device. We can consider the physical location in macro and micro terms. If macro location information (such as living room, kitchen, or bedroom in an apartment) is available to an IoT device, a keyword for that location can be used in the DNS name as a subdomain name, such as living_room. The physical location of each device is defined as a macro location for DNS naming. Using the micro location, an IoT device can be located in the center, wall, or corner of the room specified by the macro location. For example, suppose that a robot cleaner is in the right upper corner of a living room. If the DNS name for the cleaning robot contains the right-upper corner of the living room, a home resident can find it easily. In this paper, we have specified such a detailed location for an IoT device as a micro location subdomain name, such as "right_upper_corner.".

C. Procedure for DNSNAv4

This section explains the procedure for our DNS name autoconfiguration scheme, called DNSNAv4. DNSNAv4 is divided into four steps: IoT device DNS name generation, DNS name registration-1, DNS name registration-2, and DNS name retrieval. After these four steps, the DNS name of an IPv4 node (i.e., host) is generated according to the type of the IPv4 node (e.g., mobile device or static device) along with its location.

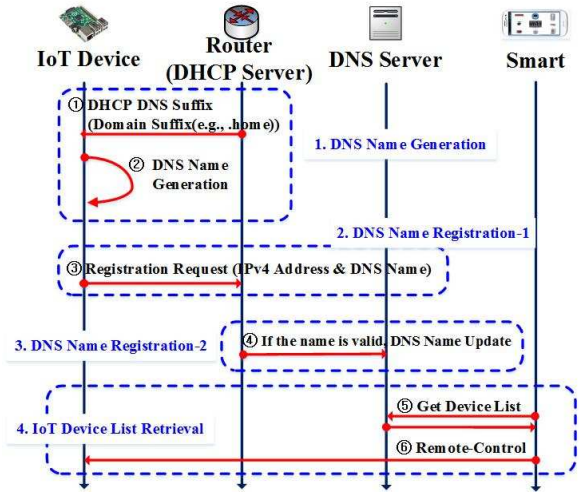

Fig. 3. Sequence Diagram for DNSNAv4

Fig. 3 shows a sequence diagram for DNSNAv4. The different point of previously DNSNA [2], the router does not require the reply. Because in the IPv4 network case, if an IPv4 node is connected to IPv4 network, an IPv4 node Although not shown in Fig. 3, it is started that an IoT device requests an IPv4 address to a DHCPv4 server. Then, the DHCPv4 server sends a message which are an IPv4 address and DHCP DNS suffix option. For the IoT device name generation, the IoT device gets information which are a DHCP DNS suffix information

from the DHCPv4 server and own device information such as previous mentioned OID, unique_id something like that. The IoT device generates own DNS name with these information.

Second, the DNS name is registered DNS names of IoT devices via a DHCPv4 server after the step of the DNS name generation. An IoT device with the generated the DNS name requests to register the IPv4 address and the corresponding DNS name to the DHCPv4 server in subnet.

Third, the DHCPv4 server checks this request if it is valid or not. If the request message is valid, the DHCPv4 server updates the DNS name with the corresponding IPv4 address to an authoritative DNS server that may hold the DNS name of each host. Otherwise, if the request message is not valid, the DHCPv4 server replies an error message to the IPv4 node. After the IoT device receives this error message, the IoT device regenerates own DNS name and requests the DHCPv4 server to register again. It performs for the new DNS name and repeats this process until it has verified the uniqueness of the generated DNS name.

Fourth, DNS name retrieval is the process to retrieve domain names to represent them on a user application. The authoritative DNS server maintains the DNS information about IoT devices in a zone file. After the DNS name registration, a user's smart device (e.g., smartphone, tablet PC) can retrieve the list of IoT devices from the authoritative DNS server and identify them with either text-based or graphic-based user interfaces. With these interfaces, users can easily monitor and remote-control IoT devices in the target network.

D. DNS Name Conflict

In provided our scenario, we can consider the problem about the DNS name conflict. The main contribution of this paper is the resolution of a DNS name confilct. In [3] [11] [12] [13], those do not provide the exactly resolution. Actually in [13], there mentioned this problem of a DNS name confilct and offered some resolution. However, it cannot be solved with a some degree of resolution. The ultimate goal of our DNSNA scheme is to provide that each IoT device can autoconfiguration with minimize human interaction. So, we need a clearly DNS name resolution for IoT devices in the IPv4 network because of the case of a large number of IoT device almost does not have input devices. This section covers the following two issues for our scheme in the IPv4 network environment:

First, in the case of invalid a DNS register message, it is happened when an IoT device requests to register own the DNS name and the corresponding DNS name to a DHCPv4 server. If the DHCPv4 server receives the request message but it is not valid, the DHCPv4 server is divided into two behaviors by the type of a message. In the type of a message is different, the DHCPv4 server ignores this message. In the type of a message is correct, the DHCPv4 server replies an error message with the corresponding error code to the IPv4 node. After the IPv4 node receives this error message, the IPv4 node regenerates the request message and transmits the DHCPv4 server again. This process can be done up to three times, if it is failed again. If the number is exceeded, the DHCPv4 server should be ignored the request message from the corresponding IPv4 address. The reason why has limitation three times is a service resource issue. If a malicious device sends a request message continually, the DHCPv4 server will be overloaded by the attacker. That is why has limitation the three times.

Second, in the case of a DNS name duplication, similarly as mentioned previously it is happened when an IoT device requests to register own the DNS name and the corresponding DNS name to a DHCPv4 server. If the DHCPv4 server receives the request message but the DNS name is duplicate, the DHCPv4 server replies the other error message to the IPv4 node. After the IPv4 node receives this error message, the IPv4 node regenerates own DNS name and requests the DHCPv4 server to register again. An IoT device, which is the IPv4 node, requests to register the IPv4 address and the corresponding regenerated DNS name to the DHCPv4 server. It performs for the new DNS name and repeats this process until it has verified the uniqueness of the generated DNS name.

IV. Experiment In Testbed

We implemented DNSNAv4 to prove the concept of DNSNAv4 for DNS name services in a smart-home IoT environment. We used a Raspberry Pi as an IoT device in our DNSNAv4 testbed, a desktop PC as an authoritative DNS server, and a desktop PC as an DHCP server.

```
pi@raspberrypi:~/Desktop/DNSNAv4 $ sudo ./DNSNA_Client
DNSNA Client Program Starting....
Timestamp : Thu Jan 26 07:20:50 2017

 - param 1 : Interface        = eth0
 - param 2 : IP Address       = 115.145.178.247
 - param 3 : DNSNA-Id Default = model.category.location
 - param 4 : DNSNA-Name File  = local_name.txt
 - param 5 : DHCP Server IP   = 115.145.178.183
DNS Search List : cpslab
Local machine name : led1.raspberryPi.led.livingRoom
Create Domain Name : led1.raspberryPi.led.livingRoom.cpslab
```

Fig. 4. Reception of an IPv4 Address and DNSSL at an IoT Device

```
pi@raspberrypi:~/Desktop $ sudo ./DNSNA_Server
  Timestamp : Thu Jan 26 07:20:45 2017
  DHCP Server is running.........
=========================================================
* Server Port           : 1234
* Dns server Ip address : 115.145.178.174
* Dns server rndc key   : lrZ3jSO7dy/3EQxNjd6aPQ==
=========================================================

  Timestamp : Thu Jan 26 07:20:50 2017
=========================================================
* Recv Message       LEN : 58
=========================================================
* Type       : 111,    DNS Update Request
* DNS Name   : led1.raspberryPi.led.livingRoom.cpslab
* IP Address : 115.145.178.247
* Length     : 58
=========================================================
```

Fig. 5. Reception of an IoT Device's DNS Name Registration at a DHCP Server

Fig. 4 shows the screen of the IoT device. The IoT device requests an IPv4 address and DNSSL to a DHCP server, then the DHCP server replies to the request with the corresponding information (i.e., IPv4 address and DNSSL). The IoT device generates own its DNS name when the IoT device receives this

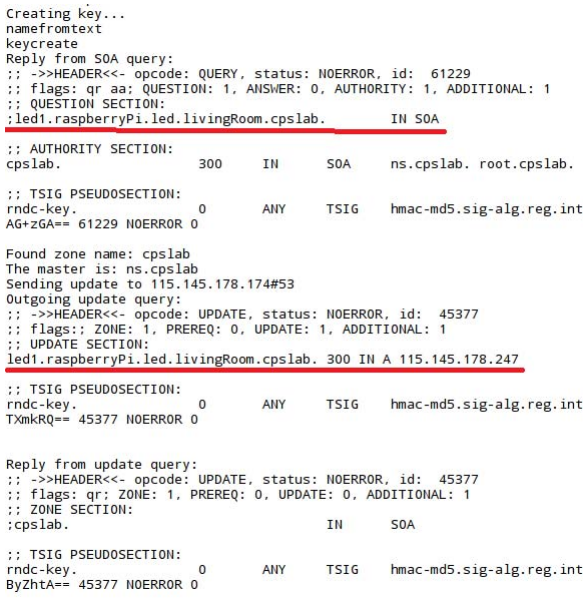

Fig. 6. DNS Dynamic Update

information. After generating its DNS name, the IoT device requests the registration of its own DNS name through a DHCP server. As shown in Fig. 4, the red marks are the IPv4 address and DNSSL which are received from the DHCP server.

Fig. 5 shows the screen of the DHCP server. The DHCP server receives a message to register a DNS name from an IoT device. The DHCP server checks the uniqueness of the DNS name with its local storage, that is, whether it is duplicate or not. If the DNS name is not duplicate, the DHCP server saves this DNS name in its local storage. This information, which is saved in the local storage, is updated periodically for the registration of DNS names. As shown in Fig. 5, the red mark is the message that the DHCP server receives from an IoT device to register the DNS name of the IoT device.

Fig. 6 shows the DNS dynamic update [14] for registering the DNS name and IPv4 address of the IoT device into an authoritative DNS server. We use the Berkeley Internet Name Domain (BIND) software package as the authoritative DNS server for DNS services. The DHCP server checks the existence of the global DNS name of the Raspberry Pi by sending a DNS name query to the DNS server. If no corresponding IPv4 address for the queried DNS name is returned from the DNS server, this means that no other IPv4 node is using that DNS name. In that case, the DHCP server registers the DNS information (i.e., DNS name and IPv4 address) of the Raspberry Pi into the DNS server. Otherwise, the DHCP server cannot register the DNS information, so it needs to notify the corresponding IoT device of the duplicated DNS name, letting the IoT device generate a new DNS name and try again to register it into the DNS server via the DHCP server.

Note that our DNSNAv4 allows for a partial deployment in that only IPv4 hosts understanding the DNSNAv4 protocol can join our DNSNAv4 service. Thus, DNSNAv4-aware IoT devices can coexist with DNSNAv4-unaware IoT devices in the Internet.

V. Conclusion

In this paper, we propose a DNS Name Autoconfiguration (DNSNAv4) for DNS naming services in Internet of Things (IoT) devices in IPv4 network environments. DNSNAv4 aims at providing an efficient DNS name autoconfiguration service for IoT devices with minimum human intervention. For the easy implementation of DNS name autoconfiguration, DNSNAv4 uses the Dynamic Host Configuration Protocol (DHCP) that is a standard protocol for the network parameter autoconfiguration, such as IP address allocation, recursive DNS server addresses, and DNS suffix domain names. Thus, with DNSNAv4, users can easily perform the DNS name registration of their IoT devices. We believe that our DNSNAv4 is a promising approach for the convenient IoT DNS naming services in the IPv4 network environments. As future work, we will research on the extension of the DHCP for service discovery in DNSNAv4. We will also consider the security issues of DNSNAv4.

Acknowledgment

This research was supported by Basic Science Research Program through the National Research Foundation of Korea (NRF) funded by the Ministry of Science, ICT & Future Planning (2014006438). Note that Jaehoon (Paul) Jeong is the corresponding author.

References

[1] Gartner, Inc., "Gartner Says the Internet of Things Installed Base Will Grow to 26 Billion Units By 2020," http://www.gartner.com/newsroom/id/2636073.
[2] Sejun Lee and Jaehoon (Paul) Jeong and Jung-Soo Park, "DNSNA: DNS Name Autoconfiguration for Internet of Things Devices," in *Proceedings of the 18th International Conference on Advanced Communication Technology (ICACT)*, Feb. 2016.
[3] R. Droms, "Dynamic Host Configuration Protocol," *IETF RFC 2132*, Mar. 1997.
[4] J. Jeong, S. Park, L. Beloeil, and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration," *IETF RFC 6106*, Nov. 2010.
[5] M2M, "Object Identifier based M2M Device Identification Scheme," http://www.onem2m.org.
[6] Apple, "Bonjour," https://developer.apple.com/.
[7] S. Cheshire and M. Krochmal, "Multicast DNS," *IETF RFC 6762*, Feb. 2013.
[8] S. Cheshire and M.Krochmal, "DNS-Based Service Discovery," *IETF RFC 6763*, Feb. 2013.
[9] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, "Neighbor Discovery for IP version 6 (IPv6)," *IETF RFC 4861*, Sep. 2007.
[10] R. Droms, J. Bound, B. Volz, C. Perkins, and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)," *IETF RFC 3315*, Jun. 2003.
[11] M. Stapp, T. Lemon, and A. Gustafsson, "A DNS Resource Record (RR) for Encoding Dynamic Host Configuration Protocol (DHCP) Information (DHCID RR)," *IETF RFC 4701*, Oct. 2006.
[12] M. Stapp, B. Volz, and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option," *IETF RFC 4702*, Oct. 2006.
[13] M. Stapp and B. Volz, "Resolution of Fully Qualified Domain Name (FQDN) Conflicts among Dynamic Host Configuration Protocol (DHCP) Clients," *IETF RFC 4703*, Oct. 2006.
[14] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)," *IETF RFC 2136*, Apr. 1997.