2015 29th International Conference on Advanced Information Networking and Applications Workshops

DNS Name Autoconfiguration for IoT Home Devices

Sejun Lee*, Jaehoon (Paul) Jeong†, and Jungsoo Park‡
* Department of Computer Science & Engineering, Sungkyunkwan University, Republic of Korea
† Department of Interaction Science, Sungkyunkwan University, Republic of Korea
‡ Electronics and Telecommunications Research Institute, Republic of Korea
Email: {sejunlee,pauljeong}@skku.edu, pjs@etri.re.kr

***Abstract*—This paper proposes an autoconfiguration scheme for DNS names of home network devices. By this scheme, the DNS name of a home network device can be autoconfigured with the devices category and model in a home network. This DNS name lets home residents easily identify each device for monitoring and remote-controlling it in a home network.**

I. Introduction

Internet of Things (IoT) is a promising concept suggesting that everything is connected to the Internet and interacts with each other across a network. IoT is a global infrastructure which combines situational awareness knowledge and provides intelligent service convergence for smart devices via cloud. IoT is expected to be the core technology for the second connection-oriented society which indicates openness and sharing [1]. Along with this trend, a small-scale area for IoT is a smart home in which IoT devices exist, having various capacity specifications from low-capacity to high-capacity. As high-capacity devices, usually big appliances (e.g., smart TV, refrigerator, air conditioner, and washing machine), they are equipped with WiFi module, high performance CPU, basic storage, and tailored OS. These enable them to perform communication, sensing, computing, and control in home network area and provide remote control and monitoring functions through the Internet. Also, even though low-capacity devices (e.g., light, meter, temperature controller, and sensors) have limited or no computation capability, they will be useful for the easy management of home environment.

For network devices, some parameters of Internet connection (e.g., IPv6 address, default router, and DNS server) can be automatically configured by Neighbor Discovery (ND) for IPv6, IPv6 Stateless Address Autoconfiguration and IPv6 Router Advertisement (RA) Options for DNS Configuration [2]–[4]. However, to some extent the DNS names still need to be configured manually, which would consume considerable time, especially in IoT environment that includes enormous smart devices. It will be beneficial if such DNS names can be automatically configured such that they can be more easily readable by people.

In this paper, we propose an autoconfiguration scheme of DNS names for home network devices. The proposed scheme provides an automatic generation of a DNS name that consists of device category and device model for user-friendliness. Based on this autoconfiguration scheme, a user will be able to easily monitor and remotely control home devices with mobile smart devices, such as smartphone and tablet. Note that this paper is the enhanced version of our early IETF Internet draft [5].

The remaining of the paper is constructed as follows. Section II summarizes related work. Section III describes the problem formulation for DNS name autoconfiguration. Section IV explains the DNS name autoconfiguration for home network devices. Finally, in Section V, we concludes the paper along with future work.

II. Related Work

IoT contains traditional technologies (e.g., wireless sensor networks and machine-to-machine communications) and may create various services based on the Internet and web technology. Recently, studies about IoT technology have been increasing [1]. The Internet Engineering Task Force (IETF) as an Internet standard organization, began to make IoT protocols that can accommodate various low-capacity nodes. As a reprentative work, Constrained Application Protocol (CoAP) protocol was developed, based on the legacy HTTP for the world-wide web. CoAP works over User Datagram Protocol (UDP). Whenever an event (e.g., temperature and humidity) happens, CoAP is used to deliver the event to a server in the Internet [6].

Bonjour is an application that supports the zeroconf of DNS name service and service discovery in home environments [7]. Bonjour allows devices to locate other devices (e.g., computers, printers, access point, file server, and web server) and various services. This device locating service is implemented through Multicast Domain Name System (mDNS) service [8]. mDNS can search for the IP addresses of DNS names through zero-configuration in local network area. mDNS uses Domain Name System (DNS) packet formats for DNS name resolution. The advantage of mDNS is that it can resolve host names into IP addresses without any dedicated DNS server in small network environments. The disadvantage of mDNS is that in multi-subnet home networks, mDNS generates lots of multicast packets for DNS name resolution, leading to big overhead. On the other hand, our scheme uses unicast rather than multicast, the traffic for DNS name resolution will be significantly reduced in comparision with mDNS. When mDNS needs to know the host name for an IP address, it will send an IP query message in multicast. All hosts will receive this message and an host corresponding to the quired host name will respond to this message to mDNS that generated the query message [8]. In this paper, we propose a DNS name autoconfiguration scheme for DNS name geneation and resolution of IoT devices in home networks. Since our scheme works in unicast with a dedicated DNS server in a home network, it can provide an efficient DNS name services for IoT home devices. Also, since autoconfigured DNS names contain device category and device

978-1-4799-1775-4/15 $31.00 © 2015 IEEE
DOI 10.1109/WAINA.2015.104

IEEE computer society

model, home residents can easily identify devices with the DNS names.

III. Terminology

In this section, we define terminology for our autoconfiguration scheme for IoT home network devices, using device configuration and DNS search list.

- Device configuration is a factory configuration that has device category (e.g., smart TV, smartphone, tablet, and refrigerator) and device model (i.e., manufacturer name and device model identifier), Fig. 1 shows the DNS name autoconfiguration for IoT devices at home.

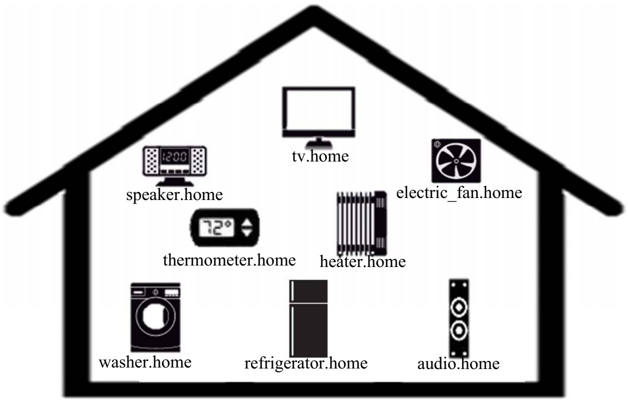

Fig. 1. IoT Home Device

- DNS Search List (DNSSL) is the list of DNS suffix domain names used by IPv6 hosts when they perform DNS query for short, unqualified domain names [3].
- DNSSL option is IPv6 RA option to deliver DNSSL information to IPv6 hosts [3].

IV. DNS Name Autoconfiguration

In this section, we describe our autoconfiguration scheme for DNS names of IoT home network devices. The DNS name autoconfiguration needs the acquisition of DNS search list through either RA [3] or DHCPv6 [9]. Once the DNS search list is obtained, the home network device autonomously constructs its DNS name with the DNS search list and its device information.

A. DNS Name Format

A DNS name for a home network device has the following format as in Fig. 2:

```
unique_id.device_model.device_category.domain_name
```

Fig. 2. Home Network Devices DNS Name Format

- **unique_id** is a unique identifier to guarantee the uniqueness of the DNS name in ASCII characters. The identifier may be a sequence number or alphanumeric with readability, such as product name.
- **device_model** is devices model name in ASCII characters. It is a product model name provided by the manufacturer.
- **device_category** is devices category name in ASCII characters, such as TV, refrigerator, air conditioner, smartphone, tablet, light, and meter.
- **domain_name** is DNS domain name that is encoded according to the specification of ”Representation and use of domain name” [10].

B. Procedure of DNS Name Autoconfiguration

This procedure consists of two phases: (i) DNS Name Generation Phase and (ii) DNS Name Registration Phase. Phase 1 for DNS name generation is as follows:

- **Step 1:** An IPv6 host as an IoT device receives a DNSSL option through either RA or DHCPv6.
- **Step 2:** An IPv6 checks the validity for the DNSSL option.
- **Step 3:** If the option is valid, it performs Duplicate Address Detection (DAD) for DNS Name.

Phase 2 for DNS Name Registration is as follows:

- **Step 1:** An IPv6 host as an IoT device receives an NI Query from a router in the same subnet.
- **Step 2:** It sends the router an NI Reply for its DNS name(s).
- **Step 3:** The router registers IoT DNS information into a DNS server through DNS dynamic update.
- **Step 4:** A control device (e.g., smartphone and smart TV) gets a DNS name list for IoT devices from the DNS server.
- **Step 5:** The control device can monitor and remote-control IoT devices with the DNS name list.

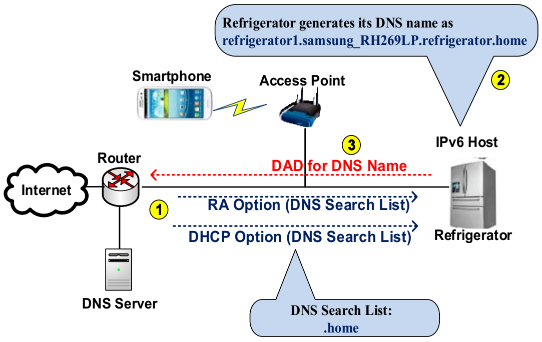
Refrigerator generates its DNS name as refrigerator1.samsung_RH269LP.refrigerator.home
2
Smartphone
Access Point
3
IPv6 Host
Router
DAD for DNS Name
Internet
1
RA Option (DNS Search List)
DHCP Option (DNS Search List)
Refrigerator
DNS Server
DNS Search List: .home

Fig. 3. IoT Device DNS Name Generation

1) Procedure of Device Name Generation: When an IPv6 host as an IoT device receives a DNSSL option through either RA or DHCPv6, it checks the validity for the DNSSL option. If the option is valid, the IPv6 host performs the DNS name

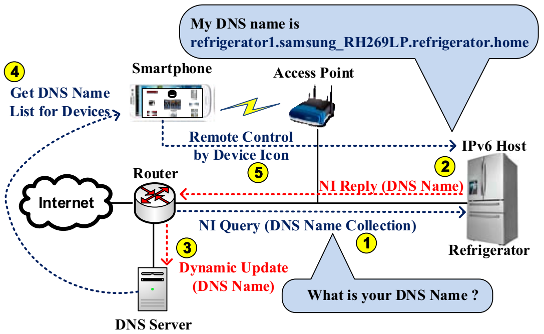

Fig. 4. IoT Device DNS Name Registration

autoconfiguration with each DNS suffix domain name in the DNSSL option as follows:

- **Step 1:** The host constructs its DNS name with the DNS suffix domain name along with device configuration and a selected identifier (as unique_id) that is considered unique.
- **Step 2:** The host performs the uniqueness test of the constructed DNS name. The uniqueness test is performed by DAD procedure in ND [2], [4].
- **Step 3:** If the DNS name is proven to be unique, it can be used as the devices DNS name, so the DNS autoconfiguration procedure is done. Otherwise, go to Step 1.

Now we explain DNS Name Uniqueness Test procedure, called DAD for DNS name. When the DNS search list has more than one DNS suffix domain name, the IPv6 host repeats the above procedure until all of the DNS suffixes are used for the DNS name autoconfiguration. As shown in Fig. 3, an IPv6 host generates an IPv6 address with 64-bit prefix from an RA option (or DHCPv6) and 64-bit hash value from the DNS name to be tested. It is assumed that an IPv6 host with an autoconfigured DNS name can respond to a DAD message for the multicast address corresponding to the DNS name. Before using such an IPv6 address associated with the DNS name, the IPv6 host performs the DAD to check whether the IPv6 address belongs to another IPv6 host or not. Note that the IPv6 host configures the IPv6 address corresponding to the DNS name as its address. If the address belongs to another IPv6 host, it is considered that the DNS name corresponding to the address is occupied by a different host. In this case, the IPv6 host selects another unique identifier (as unique_id) for a DNS name and repeats the uniqueness test of the new DNS name with the identifier.

- **Step 1:** The host computes the hash value of the DNS name to be tested for the uniqueness by using a hash function (i.e., MD5). It takes the first 64 bits of the hash value from most significant bit to construct a link-local multicast address corresponding to the DNS name [2], [4].
- **Step 2:** The host performs the uniqueness test of the constructed DNS name by the multicast query with the multicast address corresponding to the DNS name. The uniqueness test is performed through the DAD procedure in ND [2], [4].
- **Step 3:** If the DNS name is proven to be unique with no response for the DAD, the device configures the DNS name and the corresponding IPv6 address as its own DNS name and address, respectively, returning the success of the uniqueness test. Otherwise, return the failure of the uniqueness test.

2) DNS Name Registartion: Once IPv6 hosts as IoT devices have autoconfigured their DNS names, as a collector, any IPv6 node (i.e., router or host) in the same subnet can collect the device DNS names using IPv6 Node Information (NI) protocol [11].

For a collector to collect the device DNS names without any prior node information, a new NI query needs to be defined. That is, a new ICMPv6 Code should be defined for the collection of the IPv6 host DNS names. The Data field is not included in the ICMPv6 header since the NI query is for all the IPv6 hosts in the same subnet. The Qtype field for NI type is set to 2 for Node Name. The query should be transmitted by the collector to a link-local multicast address for this NI query. Assume that a link-local multicast address should be defined for device DNS name collection and that all the IPv6 hosts join this link-local multicast address for the device DNS name collection service.

When an IPv6 host receives this query sent by the collector in multicast, it transmits its Reply with a random interval between zero and Query Response Interval, as defined by Multicast Listener Discovery Version 2 [12]. This randomly delayed Reply allows the collector to collect the device DNS names with less frame collision probability by spreading out the Reply time instants. After the collector collects the device DNS names, it collects the IPv6 addresses corresponding to the DNS names by NI protocol [11]. For DNS name resolution service, the collector can register the pair of DNS name and IPv6 address for each IPv6 host into a recursive DNS server known to the collector using DNS dynamic update [13].

V. Conclusion

In this paper, we proposed our design of DNS Name Autoconfiguration for IoT home devices. As previous work, IPv6 stateless autoconfiguration has limitations for IoT devices. First, it provides prefix information, default gateway, and Maximum Transmission Unit (MTU). Second, it provides recursive DNS server addresses and DNS search list. Thus, the existing IPv6 stateless autoconfiguration can support basic network configurationr. However, for IoT environments with lots of devices, DNS name autoconfiguration is not provided for home users to use those devices easily. This paper proposed a new DNS name autoconfiguration scheme that is based on IPv6 stateless autoconfiguration. As future work, we will implement our DNS autoconfiguration scheme for IoT devices and compare it with the state-of-the-art scheme mDNS in a home network with multiple subnets.

VI. 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). This work was also partly supported by the ICT R&D program of MKE/KEIT [10041244, SmartTV 2.0 Software Platform] and ETRI. This research was supported in part by Global Research Laboratory Program (2013K1A1A2A02078326) through NRF, and the ICT R&D program of MSIP/IITP (14-824-09-013, Resilient Cyber-Physical Systems Research) and the DGIST Research and Development Program (CPS Global Center) funded by the Ministry of Science, ICT & Future Planning. Note that Jaehoon (Paul) Jeong is the corresponding author.

REFERENCES

[1] J. Gubbi, R. Buyya, S. Marusic, and M. Palaniswami, “Internet of things(iot): A vision, architectural elements, and future directions,” *Future Gener. Comput. Syst.*, vol. 29, no. 7, pp. 1645–1660, Sep 2013.

[2] T. Narten, E. Nordmark, W. Simpson, and H. Soliman, “Neighbor Discovery for IP version 6 (IPv6),” *IETF RFC 4861*, Sep. 2007.

[3] J. Jeong, S. Park, L. Beloeil, and S. Madanapalli, “IPv6 Router Advertisement Options for DNS Configuration,” *IETF RFC 6106*, Nov. 2010.

[4] S. Thomson, T. Narten, and T. Jinmei, “IPv6 Stateless Address Autoconfiguration,” *IETF RFC 4862*, Sep. 2007.

[5] J. Jeong and J. Park, “DNS Name Autoconfiguration for Home Network Devices,” *IETF draft-jeong-homenet-device-name-autoconf-01*, Sep. 2014.

[6] Z. Shelby, K. Hartke, and C. Bormann, “The Constrained Application Protocol (CoAP),” *IETF RFC 7252*, Jun. 2014.

[7] Apple, “Bonjour,” https://developer.apple.com/.

[8] S. Cheshire and M. Krochmal, “Multicast DNS,” *IETF RFC 6762*, Feb. 2013.

[9] R. Droms, “DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6),” *IETF RFC 3646*, Dec. 2003.

[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. Crawford and B. Haberman, “IPv6 Node Information Queries,” *IETF RFC 4620*, Aug. 2006.

[12] R. Vida and L. Costa, “Multicast Listener Discovery Version 2 (MLDv2) for IPv6,” *IETF RFC 3810*, Jun. 2004.

[13] P. Vixie, S. Thomson, Y. Rekhter, and J. Bound, “Dynamic Updates in the Domain Name System (DNS UPDATE),” *IETF RFC 2136*, Apr. 1997.