Workgroup: IPWAVE Working Group
Internet-Draft:
draft-jeong-ipwave-iot-dns-autoconf-13
Published: 21 August 2022
Intended Status: Standards Track
Expires: 22 February 2023
Authors: J. Jeong, Ed.
Sungkyunkwan University
J. Ahn
Sungkyunkwan University
S. Lee
Ericsson-LG
J. Park
ETRI

DNS Name Autoconfiguration for Internet-of-Things Devices in IP-Based Vehicular Networks

Abstract

This document specifies an autoconfiguration scheme for device discovery and service discovery in IP-based vehicular networks. Through the device discovery, this document supports the global (or local) DNS naming of Internet-of-Things (IoT) devices, such as sensors, actuators, and in-vehicle units. By this scheme, the DNS name of an IoT device can be autoconfigured with the device's model information in wired and wireless target networks (e.g., vehicle, road network, home, office, shopping mall, and smart grid). Through the service discovery, IoT users (e.g., drivers, passengers, home residents, and customers) in the Internet (or local network) can easily identify each device for monitoring and remote-controlling it in a target network.

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at https://datatracker.ietf.org/drafts/current/.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on 22 February 2023.

Copyright Notice

Table of Contents

1. Introduction

Many Internet-of-Things (IoT) devices (e.g., sensors, actuators, and in-vehicle units) have begun to have wireless communication capability (e.g., WiFi, Bluetooth, and ZigBee) for monitoring and remote-controlling in a local network or the Internet. According to the capacity, such IoT devices can be categorized into high-capacity devices and low-capacity devices. High-capacity devices have a high-power processor and a large storage, such as vehicles, road infrastructure devices (e.g., road-side unit, traffic light, and

loop-detector), appliances (e.g., television, refrigerator, air conditioner, and washing machine), and smart devices (smartphone and tablet). They are placed in environments (e.g., vehicle, road network, home, office, shopping mall, and smart grid) for the direct use for human users, and they require the interaction with human users. Low-capacity devices have a low-power processor and a small storage, such as sensors (e.g., in-vehicle units, light sensor, meter, and fire detector) and actuators (e.g., vehicle engine, signal light, street light, and room temperature controller). They are installed for the easy management of environments (e.g., vehicle, road network, home, office, store, and factory), and they do not require the interaction with human users.

For the Internet connectivity of IoT devices, a variety of parameters (e.g., address prefixes, default routers, and DNS servers) can be automatically configured by Neighbor Discovery (ND) for IP Version 6, IPv6 Stateless Address Autoconfiguration, and IPv6 Router Advertisement (RA) Options for DNS Configuration [RFC4861] [RFC4862] [RFC8106].

For these IoT devices, the manual configuration of DNS names will be cumbersome and time-consuming as the number of them increases rapidly in a network. It will be good for such DNS names to be automatically configured such that they are readable to human users.

Multicast DNS (mDNS) in [RFC6762] can provide DNS service for networked devices on a local link (e.g., home network and office network) without any conventional recursive DNS server. mDNS also supports the autoconfiguration of a device's DNS name without the intervention of the user. mDNS aims at the DNS naming service for the local DNS names of the networked devices on the local link rather than the DNS naming service for the global DNS names of such devices in the Internet. However, for IoT devices accessible from the Internet, mDNS cannot be used. Thus, a new autoconfiguration scheme becomes required for the global DNS names of IoT devices.

This document proposes a DNS Name Autoconfiguration (DNSNA) for the global (or local) DNS names of IoT devices in IP-based vehicular networks. Since an autoconfigured DNS name contains the model identifier (ID) of a device, IoT users in the Internet (or local network) can easily identify such a device. The autoconfigured DNS names and the corresponding IP addresses of the IoT devices are registered with local or remote authoritative DNS servers that manage the DNS suffixes of the DNS domain names. With these DNS names, they will be able to monitor and remote-control their IoT devices with their smart devices (e.g., smartphone and tablet PC) by resolving their DNS names into the corresponding IP addresses.

For cloud-based DNS naming services of IoT devices, a cloud server can collect DNS zone files having the global DNS names and IP addresses of the IoT devices from multiple DNS servers and provide IoT users with such global DNS names of IoT devices relevant to the IoT users. These IoT users can monitor and remote-control their IoT devices in the Internet with the global DNS names and IP addresses, using their smart devices.

1.1. Applicability Statements

It is assumed that IoT devices have networking capability through wired or wireless communication media, such as Ethernet [IEEE-802.3], WiFi [IEEE-802.11][IEEE-802.11a][IEEE-802.11b] [IEEE-802.11g] [IEEE-802.11n], Dedicated Short-Range Communications (DSRC) [DSRC-WAVE][IEEE-802.11p], Bluetooth [IEEE-802.15.1], and ZigBee [IEEE-802.15.4] in a local area network (LAN) or personal area network (PAN). Note that IEEE 802.11p was renamed IEEE 802.11 Outside the Context of a Basic Service Set [IEEE-802.11-OCB] in 2012. IPv6 packet delivery over an IEEE 802.11-OCB link is defined in [RFC8691].

Also, it is assumed that each IoT device has a factory configuration (called device configuration) having device model information by manufacturer ID and model ID (e.g., vehicle, road-side unit, smart TV, smartphone, tablet, and refrigerator). This device configuration can be read by the device for DNS name autoconfiguration.

2. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

3. Terminology

This document uses the terminology described in [RFC4861] and [RFC4862]. In addition, four new terms are defined below:

* Device Configuration: A factory configuration that has device model information by manufacturer ID and model ID (e.g., vehicle, road-side unit, smart TV, smartphone, tablet, and refrigerator).

* DNS Search List (DNSSL): The list of DNS suffix domain names used by IPv6 hosts when they perform DNS query searches for short, unqualified domain names [RFC8106].

* DNSSL Option: IPv6 RA option to deliver the DNSSL information to IPv6 hosts [RFC8106].

4. Overview

This document specifies an autoconfiguration scheme for an IoT device using device configuration and DNS search list. Device configuration has device model information (e.g., device's manufacturer and model). DNS search list has DNS suffix domain names that represent the DNS domains of a network having the IoT device [RFC8106].

As an IPv6 host, the IoT device can obtain DNS search list through IPv6 Router Advertisement (RA) with DNS Search List (DNSSL) Option [RFC4861][RFC8106] or DHCPv6 with Domain Search List Option [RFC3315][RFC3736][RFC3646].

The IoT device can construct its DNS name with the concatenation of manufacturer ID, model ID, and domain name. Since there exist more than one device with the same model, the DNS name should have a unique identification (e.g., unique ID or serial ID) to differentiate multiple devices with the same model.

Since both RA and DHCPv6 can be simultaneously used for the parameter configuration for IPv6 hosts, this document considers the DNS name autoconfiguration in the coexistence of RA and DHCP.

5. DNS Name Autoconfiguration

The DNS name autoconfiguration for an IoT device needs the acquisition of DNS search list through either RA [RFC8106] or DHCPv6 [RFC3646]. Once the DNS search list is obtained, the IoT device autonomously constructs its DNS name(s) with the DNS search list and its device information.

5.1. DNS Name Format with Object Identifier

A DNS name for an IoT device can have the following format with object identifier (OID), which is defined in [oneM2M-OID], as in Figure 1:

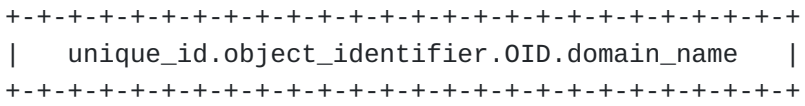

Figure 1: IoT Device DNS Name Format with OID

Fields:

unique_id	unique identifier to guarantee the uniqueness of the DNS name in ASCII characters. The identifier MAY be alphanumeric with readability, e.g., product name plus a sequence number.
object_identifier	device's object identifier that consists of a higher arc, that is, M2M node indication ID (i.e., the concatenation of the managing organization, administration, data country code, and M2M node) and a sequence of four arcs (i.e., manufacturer ID, model ID, serial ID, and expanded ID) as defined in [oneM2M-OID]. The fields are separated by an underscore '_'.
OID	subdomain for the keyword of OID to indicate that object_identifier is used.
domain_name	domain name that represents a DNS domain for the network having the IoT devices.

Note each subdomain (i.e., unique_id, object_identifier, OID, and domain_name) in the domain name format in Figure 1 is expressed using the name syntax described in [RFC1035].

5.2. Procedure of DNS Name Autoconfiguration

The procedure of DNS name autoconfiguration is performed through a DNSSL option delivered by either RA [RFC8106] or DHCPv6 [RFC3646].

5.2.1. DNS Name Generation

When as an IPv6 host a device receives a DNSSL option through either RA or DHCPv6, it checks the validity of the DNSSL option. If the option is valid, the IPv6 host performs the DNS name autoconfiguration with each DNS suffix domain name in the DNSSL option as follows:

1. The host constructs its DNS name with the DNS suffix domain name along with device configuration (i.e., manufacturer ID, model ID, and serial ID) and a selected identifier (as unique_id) that is considered unique, which is human-friendly, as shown in Figure 1.

2. The host constructs an IPv6 unicast address as a tentative address with a 64-bit network prefix and the last 64 bits of the MD5 hashed value of the above DNS name.

3. The host constructs the solicited-node multicast address in [RFC4861] corresponding to the tentative IPv6 address.

4. The host performs Duplicate Address Detection (DAD) for the IPv6 address with the solicited-node multicast address [RFC4861] [RFC4862].

5. If there is no response from the DAD, the host sets the IPv6 tentative address as its IPv6 unicast address and regards the constructed DNS name as unique on the local link. Otherwise, since the DAD fails because of DNS name conflict, go to Step 1 for a new DNS name generation with another identifier for unique_id.

6. Since the DNS name is proven to be unique, it is used as the device's DNS name and the DNS autoconfiguration is done for the given DNS suffix domain name. Also, the host joins the solicited-node multicast address for the verified DNS name in order to prevent other hosts from using this 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 along with the IPv6 unicast autoconfiguration corresponding to the DNS name.

5.2.2. DNS Name Registration

Once as IPv6 hosts the 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 [RFC4620].

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 (e.g., 3) 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 scope multicast address (e.g., all-nodes multicast address, FF02::1) SHOULD be defined for device DNS name collection such 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 its DNS name with a random interval between zero and Query Response Interval, as defined by Multicast Listener Discovery Version 2 [RFC3810]. 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 resolves the DNS names into the corresponding IPv6 addresses by NI protocol [RFC4620] with the ICMPv6 Code 1 of NI Query. This code indicates that the Data field of the NI Query has the DNS name of an IoT device. The IoT device that receives this NI query sends the collector an NI Reply with its IPv6 address in the Data field.

For DNS name resolution service, the collector can register the pair(s) of DNS name and IPv6 address for each IPv6 host with an appropriate designated DNS server for the DNS domain suffix of the DNS name. It is assumed that the collector is configured to register DNS names with the designated DNS server in a secure way based on DNSSEC [RFC4033][RFC6840]. This registration of the DNS name and IPv6 address can be performed by DNS dynamic update [RFC2136]. Before registering the DNS name with the designated DNS server, the collector SHOULD verify the uniqueness of the DNS name in the intended DNS domain by sending a DNS query for the resolution of the DNS name. If there is no corresponding IPv6 address for the queried DNS name, the collector registers the DNS name and the corresponding IPv6 address for each IPv6 host with the designated DNS server. On the other hand, if there is such a corresponding IPv6 address, the DNS name is regarded as duplicate (i.e., not unique), and so the collector notifies the corresponding IoT device with the duplicate DNS name of an error message of DNS name duplication using NI protocol. When an IoT device receives such a DNS name duplication error, it needs to construct a new DNS name and repeats the procedure of device DNS name generation along with the uniqueness test of the device DNS name in its subnet.

The two separate procedures of the DNS name collection and IPv6 address resolution in the above NI protocol can be consolidated into a single collection for the pairs of DNS names and the corresponding IPv6 addresses. For such an optimization, a new ICMPv6 Code (e.g., 4) is defined for the NI Query to query the pair of a DNS name and the corresponding IPv6 address. With this code, the collector can collect the pairs of each IoT device's DNS name and IPv6 address in one NI query message rather than two NI query messages.

For DNS name registration of IoT devices as IPv6 hosts, DHCPv6 [RFC3315] can be used instead of the NI protocol. For this purpose, a new DHCP option (called DNSNA option) needs to be defined to collect the pair of a DNS name and the corresponding IPv6 address of an IoT device. As a DNS information collector, a DHCPv6 server (or a router running a DHCPv6 server) sends a request message for the DHCP DNSNA option to IoT devices as its DHCPv6 clients under its address pool. The clients respond to this request message by sending the DHCPv6 server a reply message with their DNS information. Thus, the DHCPv6 server can collect the pairs of DNS names and the corresponding IPv6 addresses of the IoT devices. Then, as a

collector, the DHCPv6 server can register the DNS names and the corresponding IPv6 addresses of IoT devices with the designated DNS server.

To allow only a legitimate IoT device to register its DNS name and IPv6 address with the designated DNS server via a router (or DHCPv6 server), the IoT device can sign its registration message with its private key through a digital signature, and the router (or DHCPv6 server) can verify the message with the IoT device's public key. For the detailed authentication based on a digital signature, refer to [DNSNA-FGCS].

5.2.3. DNS Name Retrieval

For device discovery, a smart device (e.g., smartphone) can retrieve the DNS names of IoT devices by contacting a global (or local) DNS server having the IoT device DNS names. If the smart device can retrieve the zone file with the DNS names, it can display the information of IoT devices in a target network, such as a vehicle's internal network, home network, and office network. With this information, the user can monitor and control the IoT devices in the Internet (or local network). To monitor or remote-control IoT devices, Constrained Application Protocol (CoAP) can be used [RFC7252].

6. Location-Aware DNS Name Configuration

If the DNS name of an IoT device includes location information, it allows users to easily identify the physical location of each device. This document proposes the representation of a location in a DNS name. In this document, the location in a DNS name consists of two levels for a detailed location specification, such as macro-location for a large area and micro-location for a small area.

To denote both macro-location (i.e., mac_loc) and micro-location (i.e., mic_loc) into a DNS name, the following format is described as in Figure 2:

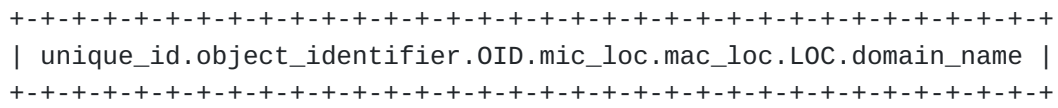

Figure 2: Location-Aware Device DNS Name Format

Fields:

unique_id	unique identifier to guarantee the uniqueness of the DNS name in ASCII characters. The identifier MAY be alphanumeric with readability, such as product name plus a sequence number.
object_identifier	device's object identifier that consists of a higher arc, that is, M2M node indication ID (i.e., the concatenation of the managing organization, administration, data country code, and M2M node) and a sequence of four arcs (i.e., manufacturer ID, model ID, serial ID, and expanded ID) as defined in [oneM2M-OID]. The fields are seperated by an underscore '_'.
OID	subdomain for the keyword of OID to indicate that object_identifier is used.
mic_loc	device's micro-location, such as an offset in a road segment and coordinates in 2-dimensional (or 3-dimensional) space.
mac_loc	device's macro-location, such as a road segment and 2-dimensional (or 3-dimensional) space.
LOC	subdomain for the keyword of LOC to indicate that mac_loc and mic_loc are used.
domain_name	domain name that represents a DNS domain for the network having the IoT devices.

Note each subdomain (e.g., mic_loc and mac_loc) in the domain name format in Figure 2 is expressed using the name syntax described in [RFC1035].

7. Macro-Location-Aware DNS Name

If location information (such as cross area, intersection, and road segment in a road network) is available to an IoT device, a keyword, coordinate, or location ID for the location information can be used to construct a DNS name as subdomain name. This location information lets users track the position of mobile devices (such as vehicle, smartphone, and tablet). The physical location of the device is defined as macro-location for DNS naming.

A subdomain name for macro-location (denoted as mac_loc) MAY be placed between micro-location (denoted as mic_loc) and the keyword LOC of the DNS name format in Figure 2. For the localization of

macro-location, a localization scheme for indoor or outdoor can be used [SALA].

8. Micro-Location-Aware DNS Name

An IoT device can be located in the center or edge in a place that is specified by macro-location. For example, assume that a loop-detector is located in the start or end position of a road segment. If the DNS name for the loop-detector contains the start or end position of the road segment, a road network administrator can find it easily. In this document, for this DNS naming, the detailed location for an IoT device can be specified as a micro-location subdomain name.

A subdomain name for micro-location (denoted as mic_loc) MAY be placed between the keyword OID and macro-location (denoted as mac_loc) of the DNS name format in Figure 2. For the localization of micro-location, a localization scheme for indoor or outdoor can be used [SALA].

9. DNS Name Management for Mobile IoT Devices

Some IoT devices can have mobility, such as vehicle, smartphone, tablet, laptop computer, and cleaning robot. This mobility allows the IoT devices to move from a subnet to another subnet where subnets can have different domain suffixes, such as coordinate.road_segment.road, coordinate.intersection.road, living_room.home and garage.home. The DNS name change (or addition) due to the mobility should be considered.

To deal with DNS name management in mobile environments, whenever an IoT device enters a new subnet and receives DNS suffix domain names, it generates its new DNS names and registers them with a designated DNS server, specified by RDNSS option.

When the IoT device recognizes the movement to another subnet, it can delete its previous DNS name(s) from the DNS server having the DNS name(s), using DNS dynamic update [RFC2136]. For at least one DNS name to remain in a DNS server for the location management in Mobile IPv6 [RFC6275], the IoT device does not delete its default DNS name in its home network in Mobile IPv6.

10. Device Discovery for IoT Devices

DNSNA can facilitate the device discovery of a user for IoT devices using a global (or local) DNS server having the IoT device DNS information, as discussed in Section 5.2.3. This device discovery based on unicast outperforms mDNS [RFC6762] using multicast in terms of the discovery speed and the network bandwidth usage for discovery.

For example, a vehicle can have its own internal network having in-vehicle devices (e.g., Electronic Control Units (ECUs) such as engine control module, powertrain control module, transmission control module, and brake control module). When the vehicle's internal network is constructed by the Ethernet, those ECUs can autoconfigure their DNS names with the DNSNA and register them with the vehicle's local DNS server [ID-IPWAVE-PS]. The local DNS server can register them with a global DNS server accessible by the automotive service center to monitor and make on-line diagnosis on them.

11. Service Discovery for IoT Devices

DNS SRV resource record (RR) can be used to support the service discovery of the services provided by IoT devices [RFC2782]. This SRV RR specifies a service name, a transport layer protocol, the corresponding port number, and an IP address of a process running in an IP host as a server to provide a service. An instance for a service can be specified in this SRV RR in DNS-based service discovery [RFC6763]. After the DNS name registration in Section 5.2, IoT devices can register their services with the DNS server via a router with DNS SRV RRs for their services.

After the service registration, an IoT user can retrieve services available in his/her target network through service discovery, which can fetch the SRV RRs from the DNS server in the target network. Once (s)he retrieves the list of the SRV RRs, (s)he can monitor or remote-control the devices or their services by using the known protocols and domain information of the devices or their services. For this monitoring or remote-controlling of IoT devices, Constrained Application Protocol (CoAP) can be used [RFC7252].

12. IoT Cloud for IoT Device Management

IoT Cloud is a cloud system to monitor and remote-control IoT devices when a user exists either indoors or outdoors. If IoT devices are installed into smart spaces (e.g., smart home and smart building) connected to the IoT Cloud, the indoor location and movement of each IoT device can be tracked and displayed by the user's smartphone App. To enable location-based services for IoT devices, the IoT Cloud can be facilitated by various IoT services including DNS Name Autoconfiguration (DNSNA) in this document.

DNSNA is used to generate the DNS name and IPv6 address of each IoT devices. It can construct the DNS name of an IoT device with the physical location (as shown in Section 6) with Indoor Positioning Systems (IPS) such as Smartphone-Assisted Localization Algorithm (SALA) [SALA] and Particle Filtering-Based IPS [PF-IPS].

As explained in [SALA], DNSNA can register the IoT device's DNS naming information with a DNS server via its local router. The IoT Cloud can interact with the DNS server to get the information of IoT devices. A smartphone user can contact the IoT Cloud to retrieve the list of IoT devices belonging to his physical location(s) and display the IoT devices on his smartphone App with the layout of the location(s). Since the IPv6 addresses of the IoT devices are known to him, the user can remote-control them through Constrained Application Protocol (CoAP) [RFC7252]

13. Security Considerations

This document shares all the security issues of the NI protocol that are specified in the "Security Considerations" section of [RFC4620].

To prevent the disclosure of location information for privacy concern, the subdomains related to location can be encrypted by a shared key or public-and-private keys. For example, a DNS name of vehicle1.oid1.OID.coordinate1.road_segment_id1.LOC.road can be represented as vehicle1.oid1.OID.xxx.yyy.LOC.road where vehicle1 is unique ID, oid1 is object ID, xxx is a string of the encrypted representation of the coordinate (denoted as coordinate1) in a road segment, and yyy is a string of the encrypted representation of the road segment ID (denoted as road_segment_id1). Thus, the location of the vehicle1 can be protected from unwanted users by encryption.

14. Acknowledgments

This work was supported by the National Research Foundation of Korea (NRF) grant funded by the Korea government, Ministry of Science and ICT (MSIT) (No. 2020R1F1A1048263).

This work was supported by the MSIT (Ministry of Science and ICT), Korea, under the ITRC (Information Technology Research Center) support program (IITP-2021-2017-0-01633) supervised by the IITP (Institute for Information & communications Technology Promotion).

15. Contributors

This document is the group work of IPWAVE working group. This document has the following contributing authors considered co-authors:

*Keuntae Lee (Sungkyunkwan University)

*Seokhwa Kim (Sungkyunkwan University)

16. References

16.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC4861] Narten, T., Nordmark, E., Simpson, W., and H. Soliman, "Neighbor Discovery for IP Version 6 (IPv6)", RFC 4861, September 2007, <https://www.rfc-editor.org/rfc/rfc4861>.

[RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, September 2007, <https://www.rfc-editor.org/rfc/rfc4862>.

[RFC8106] Jeong, J., Park, S., Beloeil, L., and S. Madanapalli, "IPv6 Router Advertisement Options for DNS Configuration", RFC 8106, March 2017, <https://www.rfc-editor.org/rfc/rfc8106>.

[RFC3315] Droms, R., Ed., Bound, J., Volz, B., Lemon, T., Perkins, C., and M. Carney, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3315, July 2003, <https://www.rfc-editor.org/rfc/rfc3315>.

[RFC3736] Droms, R., "Stateless Dynamic Host Configuration Protocol (DHCP) Service for IPv6", RFC 3736, April 2004, <https://www.rfc-editor.org/rfc/rfc3736>.

[RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, December 2003, <https://www.rfc-editor.org/rfc/rfc3646>.

[RFC1035] Mockapetris, P., "Domain Names - Implementation and Specification", RFC 1035, November 1987, <https://www.rfc-editor.org/rfc/rfc1035>.

[RFC4033] Arends, R., Ed., Austein, R., Larson, M., Massey, D., and S. Rose, "DNS Security Introduction and Requirements", RFC 4033, March 2005, <https://www.rfc-editor.org/rfc/rfc4033>.

[RFC6840] Weiler, S., Ed. and D. Blacka, Ed., "Clarifications and Implementation Notes for DNS Security (DNSSEC)", RFC 6840, February 2013, <https://www.rfc-editor.org/rfc/rfc6840>.

[RFC8691] Benamar, N., Haerri, J., Lee, J., and T. Ernst, "Basic Support for IPv6 Networks Operating Outside the Context of a Basic Service Set over IEEE Std 802.11", RFC 8691, December 2019, <https://www.rfc-editor.org/rfc/rfc8691>.

[ID-IPWAVE-PS]
Jeong, J., Ed., "IPv6 Wireless Access in Vehicular Environments (IPWAVE): Problem Statement and Use Cases", Work in Progress, Internet-Draft, draft-ietf-ipwave-vehicular-networking-29, May 2022, <https://datatracker.ietf.org/doc/html/draft-ietf-ipwave-vehicular-networking-29>.

16.2. Informative References

[RFC6762] Cheshire, S. and M. Krochmal, "Multicast DNS", RFC 6762, February 2013, <https://www.rfc-editor.org/rfc/rfc6762>.

[RFC4620] Crawford, M. and B. Haberman, Ed., "IPv6 Node Information Queries", RFC 4620, August 2006, <https://www.rfc-editor.org/rfc/rfc4620>.

[RFC3810] Vida, R. and L. Costa, "Multicast Listener Discovery Version 2 (MLDv2) for IPv6", RFC 3810, June 2004, <https://www.rfc-editor.org/rfc/rfc3810>.

[RFC2136] Vixie, P., Ed., Thomson, S., Rekhter, Y., and J. Bound, "Dynamic Updates in the Domain Name System (DNS UPDATE)", RFC 2136, April 1997, <https://www.rfc-editor.org/rfc/rfc2136>.

[RFC6275] Perkins, C., Ed., Johnson, D., and J. Arkko, "Mobility Support in IPv6", RFC 6275, July 2011, <https://www.rfc-editor.org/rfc/rfc6275>.

[IEEE-802.3] "IEEE Standard for Ethernet", December 2012.

[IEEE-802.11] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", March 2012.

[IEEE-802.11a] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - High-speed Physical Layer in the 5 GHZ Band", September 1999.

[IEEE-802.11b] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Higher-Speed

Physical Layer Extension in the 2.4 GHz Band", September 1999.

[IEEE-802.11g] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Further Higher Data Rate Extension in the 2.4 GHz Band", April 2003.

[IEEE-802.11n] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications - Amendment 5: Enhancements for Higher Throughput", March 2009.

[DSRC-WAVE] Morgan, Y., "Notes on DSRC & WAVE Standards Suite: Its Architecture, Design, and Characteristics", IEEE Communications Surveys & Tutorials, 12(4), 2012.

[IEEE-802.11p] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications - Amendment 6: Wireless Access in Vehicular Environments", July 2010.

[IEEE-802.11-OCB] "Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications", IEEE Std 802.11-2016, December 2016.

[IEEE-802.15.1] "Part 15.1: Wireless Medium Access Control (MAC) and Physical Layer (PHY) specifications for Wireless Personal Area Networks (WPANs)", June 2005.

[IEEE-802.15.4] "Part 15.4: Low-Rate Wireless Personal Area Networks (LR-WPANs)", September 2011.

[oneM2M-OID] "Object Identifier based M2M Device Identification Scheme", February 2014.

[DNSNA-FGCS] Lee, K., Kim, S., Jeong, J., Lee, S., Kim, H., and J. Park, "A Framework for DNS Naming Services for Internet-of-Things Devices", Elsevier Future Generation Computer Systems, Vol. 92, March 2019.

[SALA] Jeong, J., Yeon, S., Kim, T., Lee, H., Kim, S., and S. Kim, "SALA: Smartphone-Assisted Localization Algorithm for Positioning Indoor IoT Devices", Springer Wireless Networks, Vol. 24, No. 1, January 2018.

[PF-IPS] Shen, Y., Hwang, B., and J. Jeong, "Particle Filtering-Based Indoor Positioning System for Beacon Tag Tracking", IEEE Access, Vol. 8, December 2020.

[RFC2782] Gulbrandsen, A., Vixie, P., and L. Esibov, "A DNS RR for specifying the location of services (DNS SRV)", RFC 2782, February 2000, <https://www.rfc-editor.org/rfc/rfc2782>.

[RFC6763] Cheshire, S. and M. Krochmal, "DNS-Based Service Discovery", RFC 6763, February 2013, <https://www.rfc-editor.org/rfc/rfc6763>.

[RFC7252] Shelby, Z., Hartke, K., and C. Bormann, "The Constrained Application Protocol (CoAP)", RFC 7252, June 2014, <https://www.rfc-editor.org/rfc/rfc7252>.

Appendix A. Changes from draft-jeong-ipwave-iot-dns-autoconf-12

The following changes are made from draft-jeong-ipwave-iot-dns-autoconf-12:

* This version updates the version and date of [ID-IPWAVE-PS] as a reference.

Authors' Addresses

Jaehoon Paul Jeong (editor)
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

Phone: +82 31 299 4957
Email: pauljeong@skku.edu
URI: http://iotlab.skku.edu/people-jaehoon-jeong.php

Yoseop Ahn
Department of Computer Science and Engineering
Sungkyunkwan University
2066 Seobu-Ro, Jangan-Gu
Suwon
Gyeonggi-Do
16419
Republic of Korea

Phone: +82 31 299 4106
Email: ahnjs124@skku.edu
URI: http://iotlab.skku.edu/people-Ahn-Yoseop.php

Sejun Lee
Ericsson-LG
77, Heungan-Daero 81 Beon-Gil, Dongan-Gu
Anyang-Si
Gyeonggi-Do

14117
Republic of Korea

Phone: +82 31 450 4099
Email: prosejun14@gmail.com

Jung-Soo Park
Electronics and Telecommunications Research Institute
218 Gajeong-Ro, Yuseong-Gu
Daejeon
34129
Republic of Korea

Phone: +82 42 860 6514
Email: pjs@etri.re.kr