

Dynamic Host Configuration (DHC)  
Internet-Draft  
Intended status: Informational  
Expires: 23 May 2022

G.R. Ren  
L.H. He  
Y.L. Liu  
Tsinghua University  
19 November 2021

DHCPv6 Extension Practices and Considerations  
draft-ietf-dhc-problem-statement-of-mredhcpv6-08

Abstract

IP addresses assume an increasing number of attributes as communication identifiers to meet different requirements. Privacy protection, accountability, security, and manageability of networks can be supported by extending the DHCPv6 protocol as required. This document provides current extension practices and typical DHCPv6 server software in terms of extensions, defines a general model of DHCPv6, discusses some extension points, and presents extension cases.

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 23 May 2022.

Copyright Notice

Copyright (c) 2021 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components

extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
2. Terminology . . . . .	4
3. Current Extension Practices . . . . .	4
3.1. Standardized and Non-standardized DHCPv6 Extension Cases . . . . .	4
3.2. Current DHCPv6 Server Software Cases . . . . .	4
4. Extension Discussion . . . . .	5
4.1. DHCPv6 General Model . . . . .	5
4.2. Extension Points . . . . .	6
4.2.1. Messages . . . . .	6
4.2.2. Options . . . . .	6
4.2.3. Message Processing Functions . . . . .	7
4.2.4. Address Generation Mechanisms . . . . .	7
4.3. Extension Principles . . . . .	8
5. Extension Cases . . . . .	8
5.1. Software Configurations . . . . .	9
5.2. Option Definition and Server Modification . . . . .	9
5.3. Message Definition . . . . .	9
6. Security Considerations . . . . .	10
7. IANA Considerations . . . . .	10
8. Acknowledgements . . . . .	10
9. References . . . . .	10
9.1. Normative References . . . . .	10
9.2. Informative References . . . . .	10
Authors' Addresses . . . . .	14

## 1. Introduction

IP addresses play an essential role in communication over the Internet. Their generation and assignment are also closely linked to the privacy protection, accountability, security, and manageability of the network [I-D.gont-v6ops-ipv6-addressing-considerations]. The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) [RFC8415] is an important network protocol that can be used to dynamically provide IPv6 addresses and other network configuration parameters to IPv6 nodes. DHCPv6 can be continuously extended and improved through new options, protocols, and message processing mechanisms.

IP addresses assume an increasing number of properties as communication identifiers to meet different requirements. For example, APNA [APNA] and PAVI [PAVI] use addresses to enhance source responsibility and privacy protection. These requirements often need

to be reflected by IP address assignment protocols such as DHCPv6. Therefore, extensions to DHCPv6 are made to meet a wide variety of requirements, which is referred to as multi-requirement extensions to DHCPv6. However, it is not easy to extend DHCPv6 to meet a variety of requirements. Although DHCPv6 offers increasingly comprehensive functionality and DHCPv6 server software provides extension interfaces that allow administrators to change and customize the way they process and respond to DHCPv6 messages, there is still a lack of comprehensive understanding of where and how to extend in DHCPv6 effectively. Therefore, a detailed analysis is needed to clarify the issues and design principles and extract and unify design specifications to help better address the multi-demand scaling problem.

In summary, with the large-scale deployment and application of IPv6, new scenarios such as Data Center Network, Internet of Things, Industrial Internet, and Integrated satellite-terrestrial networks put forward new requirements for IP address allocation, e.g., the scale of address allocation, the efficiency of address update and synchronization, the address generation algorithms (such as association with location, identifier, and other information), and the scope of dynamic address configuration service relay and collaboration. At the same time, it also puts forward new requirements in network security, accountability, manageability, and privacy protection. These are what we call "multiple requirements". Multi-requirement extensions for DHCPv6 is to meet new scenarios and new requirements through the expansion of new messages, options, message processing functions, or address generation mechanisms for DHCPv6. Based on careful design principles, interfaces can be defined to support more customized multi-requirement extensions without sacrificing the stability of DHCPv6.

Some people would suggest that administrators modify the open-source DHCPv6 server to solve their problems. However, it takes considerable time to understand the code of an open-source DHCPv6 server, not to mention the time-consuming task of debugging errors, failures, or system crashes caused by modifying complex modules. Another problem is that as open-source software evolves, the source code of the server software may change (new features or bug fixes). Once the latest version of the open-source server software comes out [kea\_dhcp\_hook\_developers\_guide], users may need to rewrite their code. Therefore, the multi-requirement extensions to DHCPv6 to address the specific issues of administrators are essential and significant.

This document provides a survey of current extension practices and typical DHCPv6 server softwares on extensions and gives DHCPv6 extension considerations by defining a DHCPv6 general model, discussing the extension problems, and presenting extension cases.

## 2. Terminology

Familiarity with DHCPv6 and its terminology, as defined in [RFC8415], is assumed.

**Multi-requirement extensions:** The multi-requirement extensions for DHCPv6 is to meet new scenarios and requirements by extending DHCPv6 with new messages, options, message processing features, or address generation mechanisms.

## 3. Current Extension Practices

### 3.1. Standardized and Non-standardized DHCPv6 Extension Cases

Many documents attempt to extend DHCPv6. They can be classified into three categories.

Extended options	Most extensions for DHCPv6 are implemented in this way. New-defined options carry specific parameters in DHCPv6 messages, which helps DHCPv6 clients or servers know the detailed situation with each other.
Extended messages	Some documents define new protocols that aim to achieve specific goals, e.g., active leasequery [RFC7653], General Address Generation and Management System [GAGMS].
Extended entities	Some documents introduce third-party entities into the communications of DHCPv6 to achieve specific goals and provide better services, e.g., authentication [RFC7037].

### 3.2. Current DHCPv6 Server Software Cases

A lot of commercial and open source DHCPv6 servers exist, including Cisco Prime Network Registrar (CPNR) DHCP [CPNR], DHCP Broadband [DHCP\_Broadband], FreeRADIUS DHCP [FreeRADIUS\_DHCP], ISC DHCP [ISC\_DHCP], Kea DHCP [Kea\_DHCP], Microsoft DHCP [Microsoft\_DHCP], Nominum DHCP [Nominum\_DHCP], VitalQIP [VitalQIP], and WIDE DHCPv6 [WIDE\_DHCPv6]. Commercial and open-source DHCPv6 software often considers the extensions of DHCPv6 servers because they cannot always meet the requirements that the administrators want. For example,

CPNR DHCP server provides extension APIs and allows administrators to write extensions and functions to alter and customize how it handles and responds to DHCP requests. A network operator usually decides what packet process to modify, how to modify, and which extension point to attach the extension. Then the network operator writes the extension and adds the well-written extension to the extension point of the DHCP server. Finally, the network operator reloads the DHCP server and debugs whether the server runs as it expects. Similarly, Kea DHCP provides hook mechanisms, a well-designed interface for third-party code, to solve the problem that the DHCP server does not quite do what a network operator require.

#### 4. Extension Discussion

This section elaborates multi-requirement extensions for DHCPv6. Section 4.1 describes the general model of DHCPv6, while Section 4.2 analyzes the extension points and requirements.

##### 4.1. DHCPv6 General Model

Figure 1 summarizes the DHCPv6 general model and its possible extensions: messages, options, message processing functions, and address generation mechanisms.

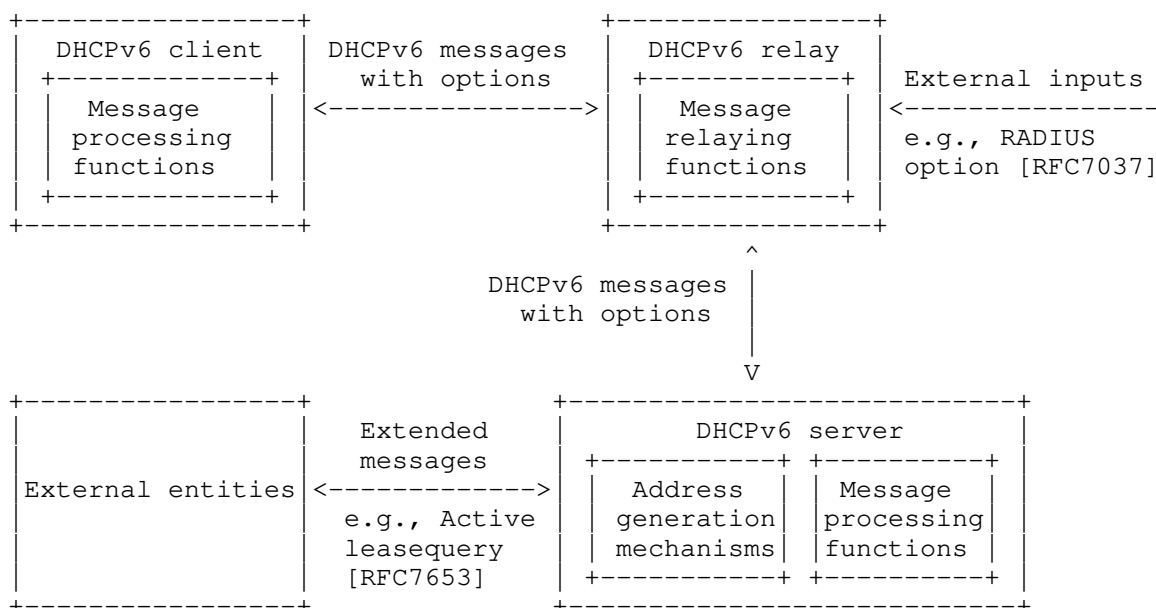


Figure 1: DHCPv6 general model and its possible extensions.

## 4.2. Extension Points

### 4.2.1. Messages

On the one hand, new messages can be designed and added to the DHCPv6 protocol to enrich its functionalities. For example, [RFC5007] defines new leasequery messages to allow a requestor to retrieve information on the bindings for a client from one or more servers. [RFC5460] expands on the Leasequery protocol by defines new messages and allowing for bulk transfer of DHCPv6 binding data via TCP. [RFC7653] defines active leasequery messages to keep the requestor up to date with DHCPv6 bindings. [RFC8156] defines failover messages to provide a mechanism for running two servers with the capability for either server to take over clients' leases in case of server failure or network partition.

On the other hand, people are concerned about the security and privacy issues of the DHCPv6 protocol. [RFC7824] describes the privacy issues associated with the use of DHCPv6, respectively. DHCPv6 does not provide privacy protection on messages and options. Other nodes can see the options transmitted in DHCPv6 messages between DHCPv6 clients and servers. Extended messages can be designed to secure exchanges between DHCPv6 entities.

### 4.2.2. Options

DHCPv6 allows defining options to transmit parameters between DHCPv6 entities for common requirements, e.g., DNS configurations [RFC3646], NIS configurations [RFC3898], SNTP configurations [RFC4075], relay agent subscriber-id [RFC4580], relay agent remote-id [RFC4649], FQDN configurations [RFC4704], relay agent echo request [RFC4994], network boot [RFC5970], Relay-Supplied Options [RFC6422], virtual subnet selection [RFC6607], client link-layer address [RFC6939], and software source binding prefix hint [RFC8539]. Also, these parameters may come from external entities. For example, [RFC7037] defines RADIUS option to exchange authorization and identification information between the DHCPv6 relay agent and DHCPv6 server.

In other cases, network operators may require DHCPv6 messages to transmit some self-defined options between clients and servers. Currently, the vendor-specific information option allows clients and servers to exchange vendor-specific information. Therefore, administrative domains can define and use the sub-options of the vendor-specific information option to serve their private purposes. The content of the self-defined options may come from two sources: devices and users. If the content of self-defined options comes from users, two methods can be used to solve the problem. The first one is that the clients provide related interfaces to receive such

information, which is currently merely supported. The second one is that DHCPv6 relays obtain such information and add it to the clients' requests. But this always depends on other protocols to allow DHCPv6 relays to get the information first.

#### 4.2.3. Message Processing Functions

Although current commercial or open-source DHCPv6 server softwares provide comprehensive functionalities, they still cannot meet all customers' requirements of processing DHCPv6 requests. Therefore, they will offer interfaces that customers can use to write their specific extensions to affect the way how DHCPv6 servers handle and respond to DHCP requests. For example, a network operator may want his DHCPv6 server to communicate with external servers. Thus, he may alter his DHCPv6 server through the given extensions to achieve such a goal. However, not all DHCPv6 software considers this extension.

#### 4.2.4. Address Generation Mechanisms

Currently, the DHCPv6 servers assign addresses, prefixes and other configuration options according to their configured policies. Generally, different networks may prefer different address generation mechanisms. Several address generation mechanisms for SLAAC [RFC4862] (e.g., IEEE 64-bit EUI-64 [RFC2464], Constant, semantically opaque [Microsoft], Temporary [RFC4941], and Stable, semantically opaque [RFC7217]) proposed for different requirements can be utilized in DHCPv6 protocol as well. Note that [RFC7943] is the DHCPv6 version of Stable, semantically opaque [RFC7217]. The many types of IPv6 address generation mechanisms available have brought about flexibility and diversity. Therefore, corresponding interfaces could be open and defined to allow other address generation mechanisms to be configured.

Moreover, several basic operations are defined to support the design of IPv6 addresses generation mechanisms. A new IPv6 address generation mechanism can be made up of the combination of the following basic operations. Also, new basic operations can be defined to support new functions.

Invert(x, n)	invert bit n of input x.
Insert(x, n, s)	insert s after bit n of input x.
Concatenate(x, y, ...)	concatenate input [x, y, ...] sequentially.
Replace(x, n, m, s)	change from bit n to bit m of input x into s.

Note that the length of  $s$  must be equal to  $m-n+1$ . When  $n=m$ , change only one bit of input  $x$ .

`Truncate(x, n, m)`      truncate from bit  $n$  to bit  $m$  of input  $x$  as the output

`Encrypt(x, k)`          use some specific encryption algorithm to encrypt input  $x$  with key  $k$ . Encryption algorithms can be IDEA, AES, RSA, etc.

`Hash(x)`                calculate the hash digest value of input  $x$ . Hash algorithms can be MD5, SHA1, SHA256, etc.

For example, temporary addresses in [RFC4941] can be expressed as `tempAddr(eui64, history) = Replace(Truncate(Hash(Concatenate(eui64, history)), 0, 63), 6, 6, 0)`, where `eui64` means the EUI-64 identifier defined in [RFC2464] and `history` means a history value defined in [RFC4941].

#### 4.3. Extension Principles

The principles used to conduct multi-requirement extensions for DHCPv6 are summarized as follows:

- 1) Do not change the basic design of DHCPv6.
- 2) Use simpler interfaces to define and support more extensions.

#### 5. Extension Cases

Administrative domains may enforce local policies according to their requirements, e.g., authentication, accountability. Several kinds of multi-requirement extensions are presented in this section, including configurations in current DHCPv6 software, option definition and server modification, and message definition between DHCPv6 entities and third-party entities. IPv6 addresses are related to manageability, security, traceability, and accountability of networks. As DHCPv6 assigns IPv6 addresses to IPv6 nodes, it is important that DHCPv6 provides interfaces to allow administrative domains to conduct extensions to meet their multi-requirements.



### 5.1. Software Configurations

Currently, many DHCPv6 servers provide administrative mechanisms, e.g., host reservation and client classification. Host reservation is often used to assign certain parameters (e.g., IP addresses) to specific devices. For example, a client with special access rights (e.g., a firewall rule that allows access based on the source's IP address) needs to keep its address allowed in the firewall configuration. Another use case is a device with a mission-critical network service that needs access by IP address in case a DNS lookup fails. Client classification is often used to differentiate between different types of clients and treat them accordingly in certain cases. This classification allows DHCP addresses or options to be assigned based on specific device characteristics or some network identifier. Grouping devices by client class makes it more convenient to perform bulk configuration settings. A typical example is the network access security policy. For example, a client class can be configured so that devices in that class are assigned IP addresses in subnets that are restricted to the public Internet due to security policies applied to the subnet/network on the router or firewall.

### 5.2. Option Definition and Server Modification

More complicated extensions of DHCPv6 are needed to meet specific requirements. For example, considering such a requirement that DHCPv6 servers assign IPv6 addresses generated by user identifiers to the clients in a network to hold users accountable, two extensions should be fulfilled to meet this requirement. The first one is that clients send their user identifiers to servers. This can be achieved by defining and using sub-options of vendor-specific information option. The second one is that servers use user identifiers to generate IP addresses. To achieve this goal, extension mechanisms provided by the server software such as extension points in CPNR [CPNR] and hook mechanisms in Kea DHCP [Kea\_DHCP] can be used.

### 5.3. Message Definition

Some extensions for DHCPv6 may need the support of third-party entities. For example, [RFC7037] introduces RADIUS entities into the message exchanges between DHCPv6 entities for better service provision. The authentication in [RFC7037] can also be used to meet the accountability requirement mentioned above because it is important to authenticate users first before assigning IP addresses generated from user identifiers. Usually, this kind of extension requires the definition of messages communicated between DHCPv6 entities and third-party entities, e.g., active leasequery [RFC7653].

## 6. Security Considerations

Security issues related with DHCPv6 are described in Section 22 of [RFC8415].

## 7. IANA Considerations

This document does not include an IANA request.

## 8. Acknowledgements

The authors would like to thank Bernie Volz, Tomek Mrugalski, Sheng Jiang, and Jinmei Tatuya for their comments and suggestions that improved the [I-D.ren-dhc-mredhcpv6]. Some ideas and thoughts of [I-D.ren-dhc-mredhcpv6] are contained in this document.

## 9. References

### 9.1. Normative References

- [RFC4862] Thomson, S., Narten, T., and T. Jinmei, "IPv6 Stateless Address Autoconfiguration", RFC 4862, DOI 10.17487/RFC4862, September 2007, <<https://www.rfc-editor.org/info/rfc4862>>.
- [RFC8415] Mrugalski, T., Siodelski, M., Volz, B., Yourtchenko, A., Richardson, M., Jiang, S., Lemon, T., and T. Winters, "Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 8415, DOI 10.17487/RFC8415, November 2018, <<https://www.rfc-editor.org/info/rfc8415>>.

### 9.2. Informative References

- [APNA] Lee, T.L., Pappas, C.P., Barrera, D.B., Szalachowski, P.S., and A.P. Perrig, "Source Accountability with Domain-brokered Privacy", December 2016.
- [CPNR] Cisco, "Cisco Prime Network Registrar", 2018, <<https://www.cisco.com/c/en/us/products/cloud-systems-management/prime-network-registrar/index.html>>.
- [DHCP\_Broadband] Weird Solutions, "DHCP Broadband", 2018, <<https://www.weird-solutions.com/carrier-solutions/dhcp-broadband>>.

- [FreeRADIUS\_DHCP]  
FreeRADIUS, "FreeRADIUS DHCP", 2017,  
<<https://wiki.freeradius.org/features/DHCP>>.
- [GAGMS] Liu, Y.L., He, L.H., and G.R. Ren, "GAGMS: A Requirement-Driven General Address Generation and Management System", November 2017.
- [I-D.gont-v6ops-ipv6-addressing-considerations]  
Gont, F.G. and G.G. Gont, "IPv6 Addressing Considerations", February 2021.
- [I-D.jia-intarea-scenarios-problems-addressing]  
Jia, Y., Trossen, D., Iannone, L., Shenoy, N., Mendes, P., and P. Liu, "Challenging Scenarios and Problems in Internet Addressing", Work in Progress, Internet-Draft, draft-jia-intarea-scenarios-problems-addressing-02, 23 October 2021, <<https://www.ietf.org/archive/id/draft-jia-intarea-scenarios-problems-addressing-02.txt>>.
- [I-D.lhan-problems-requirements-satellite-net]  
Han, L. and R. Li, "Problems and Requirements of Satellite Constellation for Internet", Work in Progress, Internet-Draft, draft-lhan-problems-requirements-satellite-net-01, 19 October 2021, <<https://www.ietf.org/archive/id/draft-lhan-problems-requirements-satellite-net-01.txt>>.
- [I-D.ren-dhc-mredhcpv6]  
Ren, G.R., He, L.H., and Y.L. Liu, "Multi-requirement Extensions for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", March 2017.
- [ISC\_DHCP] Internet System Consortium, "ISC DHCP", 2018,  
<<http://www.isc.org/downloads/dhcp/>>.
- [Kea\_DHCP] Internet System Consortium, "Kea DHCP", 2018,  
<<https://www.isc.org/kea/>>.
- [kea\_dhcp\_hook\_developers\_guide]  
Internet Systems Consortium, "Hook Developer's Guide", 2018, <[https://jenkins.isc.org/job/Kea\\_doc/doxygen/df/d46/hooksdgDevelopersGuide.html](https://jenkins.isc.org/job/Kea_doc/doxygen/df/d46/hooksdgDevelopersGuide.html)>.
- [Microsoft]  
Microsoft, "IPv6 interface identifiers", 2013, <[https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag\\_ip\\_v6\\_imp\\_addr7.msp?mfr=true](https://www.microsoft.com/resources/documentation/windows/xp/all/proddocs/en-us/sag_ip_v6_imp_addr7.msp?mfr=true)>.

- [Microsoft\_DHCP] Microsoft, "Microsoft DHCP", 2008, <[https://technet.microsoft.com/en-us/library/cc896553\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc896553(v=ws.10).aspx)>.
- [Nominum\_DHCP] Nominum, "Nominum DHCP", 2012, <[https://www.nominum.com/press\\_item/nominum-releases-new-version-of-carrier-grade-dhcp-software-for-telecom-providers/](https://www.nominum.com/press_item/nominum-releases-new-version-of-carrier-grade-dhcp-software-for-telecom-providers/)>.
- [PAVI] He, L.H., Ren, G.R., Liu, Y.L., and J.Y. Yang, "PAVI: Bootstrapping Accountability and Privacy to IPv6 Internet", April 2021.
- [RFC2464] Crawford, M., "Transmission of IPv6 Packets over Ethernet Networks", RFC 2464, DOI 10.17487/RFC2464, December 1998, <<https://www.rfc-editor.org/info/rfc2464>>.
- [RFC3646] Droms, R., Ed., "DNS Configuration options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3646, DOI 10.17487/RFC3646, December 2003, <<https://www.rfc-editor.org/info/rfc3646>>.
- [RFC3898] Kalusivalingam, V., "Network Information Service (NIS) Configuration Options for Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 3898, DOI 10.17487/RFC3898, October 2004, <<https://www.rfc-editor.org/info/rfc3898>>.
- [RFC4075] Kalusivalingam, V., "Simple Network Time Protocol (SNTP) Configuration Option for DHCPv6", RFC 4075, DOI 10.17487/RFC4075, May 2005, <<https://www.rfc-editor.org/info/rfc4075>>.
- [RFC4580] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Subscriber-ID Option", RFC 4580, DOI 10.17487/RFC4580, June 2006, <<https://www.rfc-editor.org/info/rfc4580>>.
- [RFC4649] Volz, B., "Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Relay Agent Remote-ID Option", RFC 4649, DOI 10.17487/RFC4649, August 2006, <<https://www.rfc-editor.org/info/rfc4649>>.

- [RFC4704] Volz, B., "The Dynamic Host Configuration Protocol for IPv6 (DHCPv6) Client Fully Qualified Domain Name (FQDN) Option", RFC 4704, DOI 10.17487/RFC4704, October 2006, <<https://www.rfc-editor.org/info/rfc4704>>.
- [RFC4941] Narten, T., Draves, R., and S. Krishnan, "Privacy Extensions for Stateless Address Autoconfiguration in IPv6", RFC 4941, DOI 10.17487/RFC4941, September 2007, <<https://www.rfc-editor.org/info/rfc4941>>.
- [RFC4994] Zeng, S., Volz, B., Kinnear, K., and J. Brzozowski, "DHCPv6 Relay Agent Echo Request Option", RFC 4994, DOI 10.17487/RFC4994, September 2007, <<https://www.rfc-editor.org/info/rfc4994>>.
- [RFC5007] Brzozowski, J., Kinnear, K., Volz, B., and S. Zeng, "DHCPv6 Leasequery", RFC 5007, DOI 10.17487/RFC5007, September 2007, <<https://www.rfc-editor.org/info/rfc5007>>.
- [RFC5460] Stapp, M., "DHCPv6 Bulk Leasequery", RFC 5460, DOI 10.17487/RFC5460, February 2009, <<https://www.rfc-editor.org/info/rfc5460>>.
- [RFC5970] Huth, T., Freimann, J., Zimmer, V., and D. Thaler, "DHCPv6 Options for Network Boot", RFC 5970, DOI 10.17487/RFC5970, September 2010, <<https://www.rfc-editor.org/info/rfc5970>>.
- [RFC6422] Lemon, T. and Q. Wu, "Relay-Supplied DHCP Options", RFC 6422, DOI 10.17487/RFC6422, December 2011, <<https://www.rfc-editor.org/info/rfc6422>>.
- [RFC6607] Kinnear, K., Johnson, R., and M. Stapp, "Virtual Subnet Selection Options for DHCPv4 and DHCPv6", RFC 6607, DOI 10.17487/RFC6607, April 2012, <<https://www.rfc-editor.org/info/rfc6607>>.
- [RFC6939] Halwasia, G., Bhandari, S., and W. Dec, "Client Link-Layer Address Option in DHCPv6", RFC 6939, DOI 10.17487/RFC6939, May 2013, <<https://www.rfc-editor.org/info/rfc6939>>.
- [RFC7037] Yeh, L. and M. Boucadair, "RADIUS Option for the DHCPv6 Relay Agent", RFC 7037, DOI 10.17487/RFC7037, October 2013, <<https://www.rfc-editor.org/info/rfc7037>>.

- [RFC7217] Gont, F., "A Method for Generating Semantically Opaque Interface Identifiers with IPv6 Stateless Address Autoconfiguration (SLAAC)", RFC 7217, DOI 10.17487/RFC7217, April 2014, <<https://www.rfc-editor.org/info/rfc7217>>.
- [RFC7653] Raghuvanshi, D., Kinnear, K., and D. Kukrety, "DHCPv6 Active Leasequery", RFC 7653, DOI 10.17487/RFC7653, October 2015, <<https://www.rfc-editor.org/info/rfc7653>>.
- [RFC7824] Krishnan, S., Mrugalski, T., and S. Jiang, "Privacy Considerations for DHCPv6", RFC 7824, DOI 10.17487/RFC7824, May 2016, <<https://www.rfc-editor.org/info/rfc7824>>.
- [RFC7943] Gont, F. and W. Liu, "A Method for Generating Semantically Opaque Interface Identifiers (IIDs) with the Dynamic Host Configuration Protocol for IPv6 (DHCPv6)", RFC 7943, DOI 10.17487/RFC7943, September 2016, <<https://www.rfc-editor.org/info/rfc7943>>.
- [RFC8156] Mrugalski, T. and K. Kinnear, "DHCPv6 Failover Protocol", RFC 8156, DOI 10.17487/RFC8156, June 2017, <<https://www.rfc-editor.org/info/rfc8156>>.
- [RFC8539] Farrer, I., Sun, Q., Cui, Y., and L. Sun, "Software Provisioning Using DHCPv4 over DHCPv6", RFC 8539, DOI 10.17487/RFC8539, March 2019, <<https://www.rfc-editor.org/info/rfc8539>>.
- [VitalQIP] Nokia, "Nokia VitalQIP", 2017, <<https://networks.nokia.com/products/vitalqip-ip-address-management>>.
- [WIDE\_DHCPv6] KAME project, "WIDE DHCPv6", 2008, <[http://ipv6int.net/software/wide\\_dhcpv6.html](http://ipv6int.net/software/wide_dhcpv6.html)>.

#### Authors' Addresses

Gang Ren  
Tsinghua University  
Beijing

Phone: +86-010 6260 3227  
Email: [rengang@cernet.edu.cn](mailto:rengang@cernet.edu.cn)

Lin He  
Tsinghua University  
Beijing

Email: he-lin@tsinghua.edu.cn

Ying Liu  
Tsinghua University  
Beijing

Email: liuying@cernet.edu.cn