dhc
Internet-Draft
Updates: 3396 (if approved)
Intended status: Informational
Expires: 4 September 2025

T. Jensen
M. Justel
Microsoft

3 March 2025

DHCP Option Concatenation Considerations
draft-tojens-dhcp-option-concat-considerations-01

Abstract

DHCP has a length limit of 255 on individual options because of its one-byte length field for options. To accommodate longer options, splitting option data across multiple instances of the same Option Type is defined by RFC 3396. However, this mechanism was defined to require support for all option types. This has led to real-world implementations in the years since the RFC was published to deviate from these requirements to avoid breaking basic functionality. This document updates RFC 3396 to be more flexible regarding when DHCP agents are required to concatenate options to reflect deployement experiences.

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 4 September 2025.

Copyright Notice

Table of Contents

1. Introduction

[RFC3396] defines how DHCP agents are to split and concatenate option data within an DHCP message. This has proven to be valuable as more DHCP options are defined that require support for concatenation as their data can exceed the 255 octet limit for options. Examples include [RFC4702], [RFC6731] (as a MAY), [RFC7291], [RFC8572], [RFC8973], and [RFC9463].

However, the way that [RFC3396] defined concatenation is not the way it is supported by major DHCP agent implementations today. Additionally, [RFC3396] allows for non-sensical deployments, such as handling multiple Lease Times, because it requires concatenations whenever any option is duplicated. As a result, new or existing implementors of DHCP will find real-world behavior differs from the documented standard.

This document updates [RFC3396] to clarify how option concatenation works in practice along with why it needs to differ from the previous standard. It is not intended to invent new DHCP mechanisms; rather, it clarifies with the benefit of hindsight how DHCP behaves in practice and what implementors need to account for.

2. Conventions and Definitions

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP 14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. Previous Definition of Concatenation

[RFC3396] defines the terms "concatenation-requiring" and "non-concatenation-requiring" to describe DHCP Options. While at multiple points in the document it requires implementors to handle concatenation of Options defined as concatenation-requiring, it also contains the following text which effectively requires implementors to handle concatenation of either Option type:

"However, an implementation which supports any concatenation-requiring option MUST be capable of concatenating received options for both concatenation-requiring and non-concatenation-requiring options."

4. Implementation Challenges

In combination with the only permitted use of duplicate instances of an option type in the same DHCP message being option data splitting that then requires concatenation, this means that there is no way to recover from real-world DHCP deployment mistakes that can otherwise be handled under Jon Postel’s Robustness Principle [RFC791].

For example, if a DHCP server sends two instances of an option type with fixed length, such as Option 51 (IP Address Lease Time) [RFC2132], concatenating these into an eight-octet payload will result in a protocol violation: Option 51 has to be four octets long exactly. In common deployments of DHCP agents today, we observe that this situation is handled by choosing one of the option instances that has the correct length to accept as the Option 51 value and ignoring the other instances. If implementations decided instead to strictly adhere to always concatenating multiple instances of the same option type, this would entirely block IPv4 network connectivity for the network stack.

5. What Concatenation Intends to Achieve

DHCP option concatenation is intended to allow option data to exceed the 255 octet limit imposed by its single-octet length field. This can also be used for splitting options across DHCP message section boundaries when the Overload Option indicates that the sname field, file field, or both also contain option data.

6. Changes to How Options are Split

To support the intention of option concatenation without causing the challenges described in (Section 4), this document updates [RFC3396] to limit concatenation to concatenation-requiring options. DHCP agents SHOULD NOT provide multiple instances of an option type unless that option type is defined as concatenation-requiring. To split non-concatenation-requiring options is out-of-spec behavior that leads to implementation-specific message processing.

7. Changes to How Options are Concatenated

When DHCP agents receive messages with split options that are concatenation- required options, they MUST concatenate the duplicate concatenation-required options as described in [RFC3396].

If DHCP agents sending messages never split non-concatenation-requiring options, no further guidance would be needed. However, real-world deployments have seen out-of-spec behavior that clients may wish to be defensive against and liberal in parsing. Therefore, when DHCP agents receive messages with split options that are not concatenation-requiring options, they MAY make best-effort attempts to interpret the message or fail processing entirely as a protocol error. This is implementation-specific, though some reasonable suggestions are broken down in this section based on four types of situations involving non-concatenation- requiring options still being split (even though they should not be).

7.1. Duplicates of Options with Defined Concatenation Behavior

Some non-concatenation-requiring options may still define how they are to be processed when DHCP agents receive multiple instances of the option type. In this case, DHCP agents SHOULD follow the guidance defined by the option’s standard.

7.2. Duplicates of Fixed-length Options

Fixed-length options are options that only allow a single length value, such as Option 51 Lease Time which can only be four octets long. DHCP agents receiving messages with more than one instance of fixed-length non-concatenation-requiring options MAY choose to attempt processing one of the instances if it has the correct length.

7.3. Duplicates of Multiple-of-fixed-length Options

Multiple-of-fixed-length options are options that are lists of fixed-length elements, such as Option 6 DNS Name servers which MUST be a multiple of four octets long. DHCP agents receiving messages with more than one instance of multiple-of-fixed-length options MAY choose to attempt processing one of the instances if it has a correct length.

It MAY also choose to concatenate the options if the length of the concatenated option data is a correct length (which may indicate a need to split a long list because the total length is longer than 255 octets). DHCP agents MAY choose to only do this if the length of the concatenated option data is greater than 255 octets if it wants to reduce how permissive it is.

7.4. Duplicates of Arbitrary-length Options

Arbitrary-length options are options that may have any non-zero octet length, such as Option 114 Captive-Portal [RFC8910]. DHCP agents receiving messages with more than one instance of arbitrary-length options MAY concatenate or choose one instance to process. It MAY choose to do some validation of the content that would result from each approach, meaning if the data only makes sense in the context of the option’s definition, it could use that to decide which approach to take.

8. Security Considerations

This document changes the conditions under which a DHCP agent accepts or rejects option data. The only way this might reduce a DHCP agent’s security posture would be if it would have previously refused to process data from the network that it will now process. However, this was always possible for an attacker crafting DHCP messages. Any attacker capable of creating a malformed message could instead craft a well-formed message (which would be processed in the same way before and after this document). Therefore, this document does not introduce any additional security considerations beyond the previous definitions of DHCP and option concatenation.

9. IANA Considerations

This document has no IANA actions.

10. References

10.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, March 1997, <https://www.rfc-editor.org/rfc/rfc2119>.

[RFC3396] Lemon, T. and S. Cheshire, "Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4)", RFC 3396, DOI 10.17487/RFC3396, November 2002, <https://www.rfc-editor.org/rfc/rfc3396>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in RFC 2119 Key Words", BCP 14, RFC 8174, DOI 10.17487/RFC8174, May 2017, <https://www.rfc-editor.org/rfc/rfc8174>.

10.2. Informative References

[RFC2132] Alexander, S. and R. Droms, "DHCP Options and BOOTP Vendor Extensions", RFC 2132, DOI 10.17487/RFC2132, March 1997, <https://www.rfc-editor.org/rfc/rfc2132>.

[RFC4702] Stapp, M., Volz, B., and Y. Rekhter, "The Dynamic Host Configuration Protocol (DHCP) Client Fully Qualified Domain Name (FQDN) Option", RFC 4702, DOI 10.17487/RFC4702, October 2006, <https://www.rfc-editor.org/rfc/rfc4702>.

[RFC6731] Savolainen, T., Kato, J., and T. Lemon, "Improved Recursive DNS Server Selection for Multi-Interfaced Nodes", RFC 6731, DOI 10.17487/RFC6731, December 2012, <https://www.rfc-editor.org/rfc/rfc6731>.

[RFC7291] Boucadair, M., Penno, R., and D. Wing, "DHCP Options for the Port Control Protocol (PCP)", RFC 7291, DOI 10.17487/RFC7291, July 2014, <https://www.rfc-editor.org/rfc/rfc7291>.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/rfc/rfc791>.

[RFC8572] Watsen, K., Farrer, I., and M. Abrahamsson, "Secure Zero Touch Provisioning (SZTP)", RFC 8572, DOI 10.17487/RFC8572, April 2019, <https://www.rfc-editor.org/rfc/rfc8572>.

[RFC8910] Kumari, W. and E. Kline, "Captive-Portal Identification in DHCP and Router Advertisements (RAs)", RFC 8910, DOI 10.17487/RFC8910, September 2020, <https://www.rfc-editor.org/rfc/rfc8910>.

[RFC8973] Boucadair, M. and T. Reddy.K, "DDoS Open Threat Signaling (DOTS) Agent Discovery", RFC 8973, DOI 10.17487/RFC8973, January 2021, <https://www.rfc-editor.org/rfc/rfc8973>.

[RFC9463] Boucadair, M., Ed., Reddy.K, T., Ed., Wing, D., Cook, N., and T. Jensen, "DHCP and Router Advertisement Options for the Discovery of Network-designated Resolvers (DNR)", RFC 9463, DOI 10.17487/RFC9463, November 2023, <https://www.rfc-editor.org/rfc/rfc9463>.

Acknowledgments

Thank you to Dieter Siegmund, Stuart Cheshire, and Ted Lemon for their comments and suggestions.

Authors' Addresses

Tommy Jensen
Microsoft
Email: tojens.ietf@gmail.com

Milan Justel
Microsoft
Email: milanjustel@microsoft.com