Network Working Group
Internet Draft

Ted Lemon
Nominum, Inc.

New draft

July, 2001
Expires January, 2002

Encoding Long DHCP Options
<draft-ietf-dhc-concat-01.txt>

Status of this Memo

Abstract

This draft specifies how DHCP options in a DHCP packet can be aggregated so that DHCP protocol agents can send options that are more than 255 bytes in length.

Introduction

The DHCP protocol [1] specifies objects called "options" that are encoded in the DHCP packet to pass information between DHCP protocol agents. These options are encoded as a one-byte type code, a one-byte length, and a buffer consisting of the number of bytes specified in the length, from zero to 255.

In some cases it may be useful to send DHCP options that are longer than 255 bytes, however. RFC2131 [1] specifies that when more than one option with a given type code appears in the DHCP packet, all such options should be concatenated together. It does not, however, specify the order in which this concatenation should occur.

We specify here an ordering that can be used by DHCP protocol agents when sending and receiving options with more than 255 bytes. DHCP protocol agents MAY use the ordering described here to encode existing options if such options exceed 255 bytes in length. The main purpose of this specification, however, is to provide a specification that new DHCP option drafts can reference.

Terminology

DHCP protocol agents
: This refers to any device on the network that sends or receives DHCP packets - any DHCP client, server or relay agent. The nature of these devices is not important to this specification.

Encoding agent
: The DHCP protocol agent that is composing a DHCP packet to send.

Decoding agent
: The DHCP protocol agent that is processing a DHCP packet it has received.

Options
: DHCP options are collections of data with type codes that indicate how the options should be used. Options can specify information that is required for the DHCP protocol, IP stack configuration parameters for the client, information allowing the client to rendesvous with DHCP servers, and so on.

Option overload
: The DHCP packet format is based on the BOOTP packet format defined in [4]. When used by DHCP protocol agents, BOOTP packets have three fields that can contain options. These are the actual option buffer, the server name buffer, and the filename buffer. The DHCP options specification [2] defines the DHCP Overload option, which specifies which of these three buffers is actually being used in any given DHCP message to store DHCP options.

Requirements language

In this document, the key words "MAY", "MUST, "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [3].

Applicability

This specification applies in any case where a DHCP protocol agent is encoding a packet containing options, and some of those

options must be broken into parts. This can occur either because such an option is longer than 255 bytes, or because there is not sufficient space in the current output buffer to store the option, but there is space for part of the option, and there is space in another output buffer for the rest. DHCP protocol agents encoding an option in this case MAY use the algorithm specified here.

This specification also applies in any case where a DHCP protocol agent has received a DHCP packet that contains more than one instance of an option of a given type. DHCP protocol agents that are decoding such options MAY follow the algorithm specified here.

The rationale for making this optional is that existing implementations have not been subject to a requirement to encode or decode options in this way, and it is not our intention to potentially cause existing implementations to be in violation of this specification.

The aggregate option buffer

This behaviour only applies in cases where this specification is applicable, as previously defined in the Applicability section. DHCP options can be stored in the DHCP packet in three seperate portions of the packet. These are the optional parameters field, the sname field, and the file field, as described in [1]. This complicates the description of the option splitting mechanism because there are three seperate buffers into which split options may be stored.

To further complicate matters, an option that doesn't fit into one buffer can't overlap the boundary into another buffer - the encoding agent must instead break the option into two parts and store one part in each buffer.

To simplify this discussion, we will talk about an aggregate option buffer, which will be the aggregate of the three buffers. This is a logical aggregation - the buffers MUST appear in the locations in the DHCP packet described in [1].

The aggregate option buffer is made up of the optional parameters field, the file field, and the sname field, in that order. WARNING: This is not the physical ordering of these fields in the DHCP packet.

Options MUST NOT be stored in the aggregate option buffer in such in such a way that they cross either boundary between the three fields in the aggregate buffer.

Encoding agent behaviour

Encoding agents MAY split options as described in this specification. Encoding agents supporting options that require this MUST split options as described here, if it is necessary to do so in order to encode the entire contents of the option.

Options MAY be split on any octet boundary. No split portion of an option that has been split can contain more than 255 octets. The split portions of the option MUST be stored in the aggregate option buffer in sequential order - the first split portion MUST be stored first in the aggregate option buffer, then the second portion, and so on.

Note that because the aggregate option buffer does not represent the physical ordering of the DHCP packet, if an option were split into three parts and each part went into one of the possible option fields, the first part would go into the optional parameters field, the second part would go into the file field, and the third part would go into the sname field. This maintains consistency with section 4.1 of [1].

Each split portion of an option MUST be stored in the aggregate option buffer in the same way that a full option would be stored - each portion contains a single byte option type code, and then a single byte indicating the length of the portion of the option payload being stored in that portion, and then the payload itself. The length byte MUST NOT include itself or the option code. For any given option being split, the option code MUST be the same.

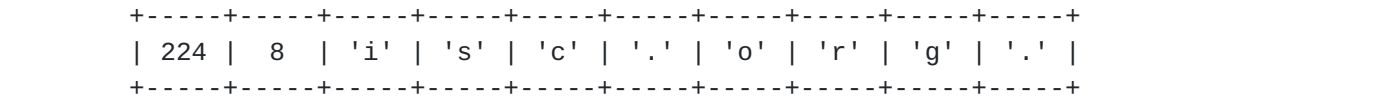
Decoding agent behaviour

When a decoding agent is scanning an incoming DHCP packet's option buffer and finds two or more options with the same option code, it MAY consider them to be a split option as described here. Decoding agents that support options that require the behaviour described in this draft MUST consider such options to be split.

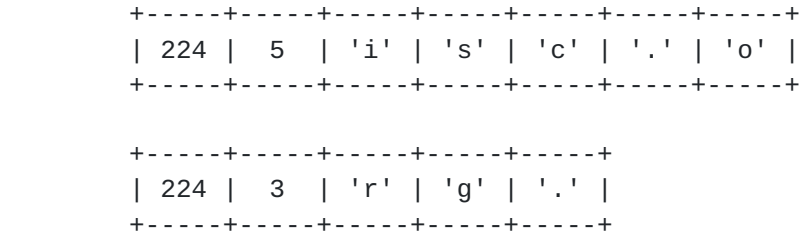
In the case that a decoding agent finds a split option, it must treat the contents of that option as a single option, and the contents must be ordered as described above under encoding agent behaviour. The decoding agent MUST ensure that when the option's value is used, any alignment issues that are particular to the machine architecture on which the decoding agent is running are accounted for - there is no requirement that the encoding agent align the options in any particular way.

Example

Consider an option, option 224, with a value of "isc.org.". Normally, this would be encoded as a single option, as follows:



If an encoding agent needed to split the option in order to fit it into the option buffer, it could encode it as two seperate options, as follows, and store it in the aggregate option buffer in the following sequence:



Security Considerations

DHCP currently provides no authentication or security mechanisms. Potential exposures to attack are discussed in section 7 of the DHCP protocol specification [1]. The Classless Static Routes option can be used to misdirect network traffic by providing incorrect IP addresses for routers.

References

[1] Droms, R., "Dynamic Host Configuration Protocol", RFC 2131, Bucknell University, March 1997.

[2] Alexander, S. and Droms, R., "DHCP Options and BOOTP Vendor Extensions", RFC 2132, Silicon Graphics, Inc., Bucknell University, March 1997.

[3] Bradner, S., "Key words for use in RFCs to indicate requirement levels", RFC 2119, Harvard University, March 1997.

[4] Mogul, J., Postel, J., "Internet Standard Subnetting Procedure", RFC950, Stanford University, USC/Information Sciences Institute, August 1985.

Author Information

Ted Lemon
Nominum, Inc.
950 Charter Street
Redwood City, CA 94043
email: mellon@nominum.com

Expiration

This document will expire on January 31, 2002.

Full Copyright Statement