Network Working Group
Internet Draft

Ted Lemon
Nominum, Inc.
Stuart Cheshire
Apple Computer, Inc.

Obsoletes: draft-ietf-dhc-concat-02.txt

April, 2002
Expires October, 2002

Encoding Long DHCP Options
<draft-ietf-dhc-concat-03.txt>

Status of this Memo

Abstract

This document specifies the processing rules for DHCP options that appear multiple times in the same message. Multiple instances of the same option are generated when an option exceeds 255 octets in size (the maximum size of a single option) or when an option needs to be split apart in order to take advantage of DHCP option overloading (Dynamic Host Configuration Protocol [1], Section 4.1. When multiple instances of the same option appear in the options, file and/or sname fields in a DHCP packet, the contents of these options are concatenated together to form a single option prior to processing.

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 the ordering that MUST be used by DHCP protocol agents when sending options with more than 255 bytes. This method also MUST be used for splitting options that are shorter than 255 bytes, if for some reason the encoding agent needs to do so. This method also MUST be used whenever a decoding agent receives a DHCP packet containing more than one of a certain type of option.

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 rendezvous with DHCP servers, and so on.

Option overload
: The DHCP packet format is based on the BOOTP packet format defined in RFC951 [4]. When used by DHCP protocol agents, BOOTP packets have three fields that can contain options. These are the optional parameters field, the sname field, and the filename field. The DHCP options specification [2] defines the DHCP Overload option, which specifies which of these three fields 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 RFC2119 [3].

Applicability

This specification applies when a DHCP agent is encoding a packet containing options, and some of those options must be broken into parts. This need can occur for two reasons. First, it can occur because the value of an option that needs to be sent is longer than 255 bytes. In this case, the encoding agent MUST follow the algorithm specified here. It can also occur 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. In this case, the encoding agent MUST either use this algorithm or not send the option at all.

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. In this case, the agent MUST concatenate these seperate instances of the same option in the way that we specify here.

The aggregate option buffer

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 RFC2131 [1]. This complicates the description of the option splitting mechanism because there are three seperate fields into which split options may be placed.

To further complicate matters, an option that doesn't fit into one field can't overlap the boundary into another field - 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 RFC2131 [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.

The encoding agent is free to choose to use either or both of the sname field and file field. If the encoding agent does not choose to use either or both of these two fields, then they MUST NOT be considered part of the aggregate option buffer in that case.

Encoding agent behaviour

Encoding agents decide to split options based on the reasons we have described in the preceding section entitled "applicability."

Options can 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. The encoding agent MUST NOT attempt to specify any semantic information based on how the option is split.

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 RFC2131 [1].

Each split portion of an option MUST be stored in the aggregate option buffer as if it were a normal variable-length option as described in RFC2132 [2]. The length fields of each split portion of the option MUST add up to the total length of the option data. For any given option being split, the option code field in each split portion 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 MUST consider them to be split portions of an option as described in the preceding section.

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 reassembled in the order that was described above under encoding agent behaviour.

The decoding agent should 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.

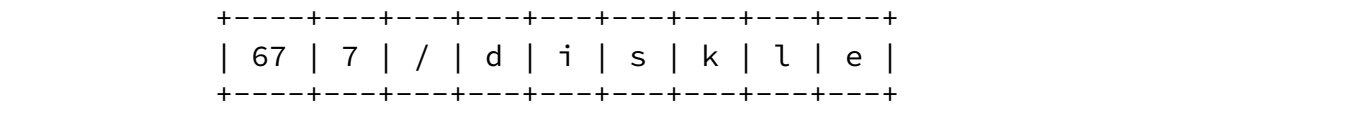
There is no semantic meaning to where an option is split - the encoding agent is free to split the option at any point, and the decoding agent MUST reassemble the split option parts into a single object, and MUST NOT treat each split portion of the option as a seperate object.

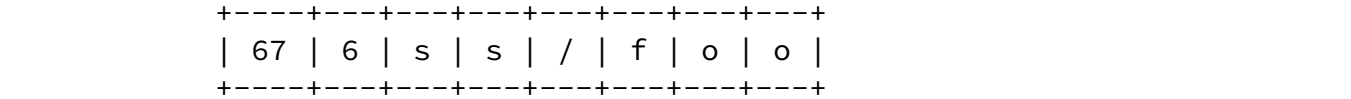
Example

Consider an option, Bootfile name (option code 67), with a value of "/diskless/foo". 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

This document raises no new security issues. Potential exposures to attack in the DHCP protocol are discussed in section 7 of the DHCP protocol specification RFC2131 [1].

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] Croft, W., Gilmore, J., "BOOTSTRAP PROTOCOL (BOOTP)", RFC951, Stanford University, Sun Microsystems, September 1985.

Author Information

Ted Lemon
Nominum, Inc.
2385 Bay Road
Redwood City, CA 94043
USA
email: mellon@nominum.com

Stuart Cheshire
Apple Computer, Inc.
1 Infinite Loop
Cupertino
California 95014
USA
Phone: +1 408 974 3207
EMail: rfc@stuartcheshire.org

Expiration

This document will expire on October 31, 2002.

Full Copyright Statement