

MALLOC working group
Internet Draft
February 1999
Expires: August 1999

Baiju V. Patel, Intel Corp.
Munil Shah, Microsoft Corp.
Stephen R. Hanna, Sun Microsystems, Inc.
[draft-ietf-malloc-madcap-04.txt](#)

Multicast Address Dynamic Client Allocation Protocol (MADCAP)

Status of this Memo

This document is an Internet-Draft and is in full conformance with all provisions of [Section 10 of RFC 2026](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF), its areas, and its working groups. Note that other groups may also distribute working documents as Internet-Drafts.

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."

The list of current Internet-Drafts can be accessed at <http://www.ietf.org/ietf/lid-abstracts.txt>.

The list of Internet-Draft Shadow Directories can be accessed at <http://www.ietf.org/shadow.html>.

Abstract

This document defines a protocol, Multicast Address Dynamic Client Allocation Protocol (MADCAP), that allows hosts to request multicast addresses from multicast address allocation servers.

1. Introduction

Multicast Address Dynamic Client Allocation Protocol (MADCAP) is a protocol that allows hosts to request multicast address allocation services from multicast address allocation servers. This protocol is part of the Multicast Address Allocation Architecture being defined by the IETF Multicast Address Allocation Working Group. However, it may be used separately from the rest of that architecture as appropriate.

Internet Draft [draft-ietf-malloc-madcap-04.txt](#)

February 1999

[1.1.](#) Protocol Overview

MADCAP is built on a client-server model, where hosts request address allocation services from address allocation servers. See [Appendix A](#) for examples of typical protocol exchanges.

[1.2.](#) Terminology

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](#).

Throughout the rest of this document, the words "server" or "MADCAP server" refer to a host providing multicast address allocation services via MADCAP. The words "client" or "MADCAP client" refer to a host requesting multicast address allocation services via MADCAP.

[1.3.](#) Motivation and Protocol Requirements

For multicast applications to be deployed everywhere, there is a need to define a protocol that any host may use to allocate multicast addresses. Here are the requirements for such a protocol.

Quick response: The host should be able to allocate a multicast address and begin to use it promptly.

Low network load: Hosts that are not allocating or deallocating multicast addresses at the present time should not need to send or receive any network traffic.

Support for intermittently connected or power managed systems: Hosts should be able to be disconnected from the network, powered off, or otherwise inaccessible except during the brief period during which they are allocating a multicast address.

Multicast address scopes: The protocol must be able to allocate both the administratively scoped and globally scoped multicast addresses.

Efficient use of address space: The multicast address space is fairly small. The protocol should make efficient use of this scarce resource.

Authentication: Because multicast addresses are scarce, it is

important to protect against hoarding of these addresses. One way to do this is by authenticating clients.

Policy neutral: Allocation policies (such as who can allocate addresses) should not be dictated by the protocol.

[1.4.](#) Relationship with DHCP

MADCAP was originally based on DHCP. There are still some similarities and it may be possible to share some code between a DHCP implementation and a MADCAP implementation. However, MADCAP is completely separate from DHCP, with no dependencies between the two and many significant differences.

[2.](#) Protocol Description

The MADCAP protocol is a client-server protocol. In general, the client unicasts or multicasts a message to one or more servers, which optionally respond with messages unicast to the client.

Messages are always sent via UDP. A reserved port number dedicated for MADCAP is used on the server (port number 2535, as assigned by IANA). Any port number may be used on client machines. When a MADCAP server sends a message to a MADCAP client, it MUST use a destination port number that matches the source port number provided by the client in the message that caused the server to send its message.

MADCAP is a mechanism rather than a policy. MADCAP allows local system administrators to exercise control over configuration parameters where desired. For example, MADCAP servers may be configured to limit the number of multicast addresses allocated to a single client. Properly enforcing such a limit requires cryptographic security, as described in the Security Consideration section.

All MADCAP messages have the same format. Each message includes a version, a message type, a transaction id, and a type-length-value encoded options field.

The next few sections describe the MADCAP message format and message types. A full list of MADCAP options is provided in [section 3](#).

[2.1.](#) Message Format

Figure 1 gives the format of a MADCAP message and Table 1 describes each of the fields in the MADCAP message. The numbers in parentheses indicate the size of each field in octets. The names for the fields given in the figure will be used throughout this document to refer to the fields in MADCAP messages.

All multi-octet quantities are in network byte-order.

Any message whose UDP data is too short to hold this format (at least 12 bytes) MUST be ignored.

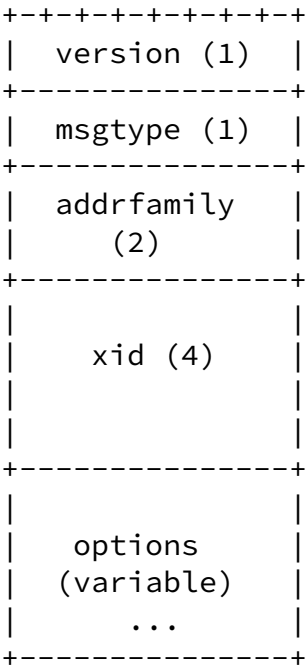


Figure 1: Format of a MADCAP message

FIELD	OCTETS	DESCRIPTION
version	1	Protocol version number (zero for this specification)
msgtype	1	Message type (DISCOVER, INFORM, etc.)
addrfamily	2	Address family (IPv4, IPv6, etc.)
xid	4	Transaction ID

options var Options field

Table 1: Description of fields in a MADCAP message

[2.1.1.](#) The version field

The version field must always be zero for this version of the protocol. Any messages that include other values in this field MUST be ignored.

[2.1.2.](#) The msgtype field

The msgtype field defines the "type" of the MADCAP message.

For more information about this field, see [section 2.2](#).

[2.1.3.](#) The addrfamily field

The addrfamily field defines the default address family (such as IPv4

or IPv6) for this MADCAP message. Unless otherwise specified, all addresses included in the message will be from this family. The set of address families is defined by the IANA, with IPv4 being 1 and IPv6 being 2.

[2.1.4.](#) The xid field

The xid field is a transaction id. This is a number chosen by the client so that the combination of xid, msgtype, and Client Identifier option value is unique over some short period of time.

The xid field is used by the client and server to associate messages and responses between a client and a server. Before a client sends a message, it chooses a number to use as an xid. When the server responds to a client message, it MUST use the same xid value in the response that the client used in the request. This allows the client to associate responses with the message that they are responding to.

When retransmitting messages (as described in [section 2.3](#)), the client MUST retransmit them without changing them, thereby using the same xid, msgtype, and Client Identifier option. The client SHOULD NOT use the same xid, msgtype, and Client Identifier together for two

different messages.

If a server receives a message with the same xid value, msgtype value, and Client Identifier option value as one received within [RESPONSE_CACHE_INTERVAL], it SHOULD treat this message as a retransmission of the previously received one and retransmit the response, if any. After [RESPONSE_CACHE_INTERVAL], the server may forget about the previously received message and treat any retransmissions of this message as if they were new messages.

This avoids retransmissions causing multiple allocations. An appropriate value for [RESPONSE_CACHE_INTERVAL] would be sixty seconds, but it may have any value (including zero seconds) and may be adjusted dynamically according to resource constraints on the server.

2.1.5. The options field

The options field consists of a list of tagged parameters that are called "options". All options consist of a two octet option code and a two octet option length, followed by the number of octets specified by the option length. In the case of some options, the length field is a constant but must still be specified.

The option field MUST contain any number of options that are not an end option followed by an end option (code 0). Any message whose

options field does not conform to this syntax MUST be ignored.

Any MADCAP client or server sending a MADCAP message MAY include any of the options listed in [section 3](#), subject to the restrictions in Table 3 and elsewhere in this document. They MAY also include other MADCAP options that are defined in the future. A MADCAP client or server MUST NOT include more than one option with the same option type in one MADCAP message.

All MADCAP clients and servers MUST recognize all options listed in this document and behave in accordance with this document when receiving and processing any of these options. Any unrecognized options MUST be ignored. If a message is received that does not conform to the requirements of this document (for instance, not including all required options) it MUST be ignored. The order of

options within a message has no significance and any order MUST be supported in an equivalent manner, with the exception that the End option must occur once per message, as the last option in the option field.

New MADCAP options may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs approved by the IESG.

[2.2.](#) Message Types

The msgtype field defines the "type" of a MADCAP message. Legal values for this field are:

Value	Message Type
-----	-----
1	DISCOVER
2	OFFER
3	REQUEST
4	RENEW
5	ACK
6	NAK
7	RELEASE
8	INFORM

Table 2: MADCAP message types

Throughout this document, MADCAP messages will be referred to by the type of the message; e.g., a MADCAP message with a message type of 0 will be referred to as an INFORM message.

Here are descriptions of the MADCAP message types. Table 3, which appears at the end of this section, summarizes which options are

allowed with each message type.

MADCAP clients and servers MUST handle all MADCAP message types defined in this document in a manner consistent with this document. If a client or server receives a message whose message type it does not recognize, it MUST ignore this message. Note, however, that under some circumstances this document requires or suggests that clients or servers ignore certain messages. For instance, clients that do not

send DISCOVER messages SHOULD ignore OFFER messages. Also, secure servers SHOULD ignore DISCOVER messages and all servers SHOULD ignore DISCOVER or INFORM messages that they cannot satisfy.

New MADCAP message types may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs approved by the IESG.

[2.2.1.](#) INFORM

The INFORM message is used by a MADCAP client that wants to acquire configuration parameters, especially a multicast scope list. This message also allows a client to determine which servers are likely to be able to handle future requests.

The MADCAP client sends out an INFORM message. The message may be unicast to a particular MADCAP server or multicast to a MADCAP Server Multicast Address. For more details about the MADCAP Server Multicast Address, see [section 2.9](#).

If a server receives an INFORM message and it can process the request successfully, it SHOULD unicast an ACK message to the client. If the INFORM message includes an Option Request List option, the server SHOULD try to include the specified options in its response, but is not required to do so. If the Option Request List option is not included, the server SHOULD include the Multicast Scope List option.

If a server receives an INFORM message and it can not process the request successfully, it SHOULD ignore the INFORM message.

If a client sends an INFORM message and does not receive any ACK messages in response after an appropriate delay, the client MAY resend its INFORM message, as described in [section 2.3](#).

When a MADCAP client sends an INFORM message, it MAY include the Requested Language option, which specifies which language the client would prefer for the zone names in the Multicast Scope List. The proper way to handle this tag with respect to zone names is discussed in the definition of the Multicast Scope List option.

[2.2.2.](#) DISCOVER

The DISCOVER message is a multicast message sent by a MADCAP client that wants to discover MADCAP servers that can probably satisfy a REQUEST.

MADCAP clients are not required to use the DISCOVER message. They MAY employ other methods to find MADCAP servers, such as sending a multicast INFORM message, caching an IP address that worked in the past or obtaining a DNS name or IP address from DHCP or prior configuration. Using the DISCOVER message has the particular advantage that it allows clients to automatically move to another server if one fails.

The MADCAP client begins by sending a multicast DISCOVER message to a MADCAP Server Multicast Address. Any servers that wish to assist the client respond by sending a unicast OFFER message to the client. If a server can process the request with a shorter lease time or later start time than the client requested, it SHOULD send an OFFER message with the lease time or start time that it can offer. However, it MUST NOT offer a lease time shorter than the minimum lease time specified by the client or a start time later than the maximum start time specified by the client.

For more details about the MADCAP Server Multicast Address, see [section 2.9](#).

After a suitable delay [DISCOVER-DELAY], the client MAY select the server it wants to use (if any) and send a multicast REQUEST message identifying that server. See [section 2.2.4](#) for more information about the REQUEST message.

The value of [DISCOVER-DELAY] and the mechanism used by the client in selecting the server it wants to use are implementation dependent.

When a MADCAP client sends a DISCOVER message, it MAY include the Lease Time, Minimum Lease Time, Start Time, Maximum Start Time, Number of Addresses Requested, and List of Address Ranges options, describing the addresses it wants to receive. However, it need not include any of these options. If one of these options is not included, the server will provide the appropriate default (maximum available for Lease Time, no minimum for Minimum Lease Time, as soon as possible for Start Time, no maximum for Maximum Start Time, one for Number of Addresses Requested, and any addresses available for List of Address Ranges). The Multicast Scope option MUST be included in the DISCOVER message so that the server knows what scope should be used. The Current Time option MUST be included if the Start Time or Maximum Start Time options are included.

If a client sends a DISCOVER message and does not receive any OFFER messages after an appropriate delay, the client MAY retransmit its DISCOVER message, as described in [section 2.3](#).

[2.2.3](#). OFFER

The OFFER message is a unicast message sent by a MADCAP server in response to a DISCOVER message that it can probably satisfy.

A MADCAP server is never required to send an OFFER message in response to a DISCOVER message. For instance, it may not be able to satisfy the client's request or it may have been configured to respond only to certain types of DISCOVER messages or not to respond to DISCOVER messages at all.

If a MADCAP server decides to send an OFFER message, it MUST include the Lease Time and Multicast Scope options, describing the addresses it is willing to provide. However, it need not include the List of Address Ranges option. If the List of Address Ranges Allocated option is not included, it is assumed that the server is willing to provide the number of addresses that the client requested. If the Start Time option is not included, it is assumed that the server is willing to provide the start time requested by the client (if any). The Current Time option MUST be included if the Start Time option is included.

If a server can process the request with a shorter lease time or later start time than the client requested, it SHOULD send an OFFER message with the lease time or start time that it can offer. However, it MUST NOT offer a lease time shorter than the minimum lease time specified by the client or a start time later than the maximum start time specified by the client.

If the server sends an OFFER message, it SHOULD attempt to hold enough addresses to complete the transaction. If it receives a multicast REQUEST message with the same xid and Client Identifier option as the DISCOVER message for which it is holding these addresses and a Server Identifier option that does not match its own, it SHOULD stop holding the addresses. The server SHOULD also stop holding the addresses after an appropriate delay [OFFER-HOLD] if the transaction is not completed. The value of this delay is implementation-specific.

As with all messages sent by the server, the xid field MUST match the xid field included in the client request to which this message is responding. The Client Identifier option MUST be included, with the value matching the one included in the client request. The Server

Identifier option MUST be included, with the value being the server's IP address. And the packet MUST NOT be retransmitted.

[2.2.4](#). REQUEST

The REQUEST message is used by a MADCAP client that wants to allocate one or more multicast addresses. It is not used for renewing an existing lease. The RENEW message is used for that.

If a REQUEST message is completing a transaction initiated by a DISCOVER message, the following procedure MUST be followed so that all MADCAP servers know which server was selected. The same xid value used in the DISCOVER message MUST be used in the REQUEST message. Also, the Server Identifier option MUST be included, using the Server Identifier of the server selected. Also, the REQUEST message MUST be multicast to the MADCAP Server Multicast Address.

If a REQUEST message is not completing a transaction initiated by a DISCOVER message, the REQUEST message MUST be unicast to the MADCAP server that the client wants to use. In this case, the Server Identifier option MAY be included, but need not be.

If the selected server can process the request successfully, it SHOULD unicast an ACK message to the client. Otherwise, it SHOULD unicast an NAK to the client. If a server can process the request with a shorter lease time or later start time than the client requested, it SHOULD send an ACK message with the lease time or start time that it can offer. However, it MUST NOT offer a lease time shorter than the minimum lease time specified by the client or a start time later than the maximum start time specified by the client.

When a MADCAP client sends a REQUEST message, it MAY include the Lease Time, Minimum Lease Time, Start Time, Maximum Start Time, Number of Addresses Requested, and List of Address Ranges options, describing the addresses it wants to receive. However, it need not include any of these options. If one of these options is not included, the server will provide the appropriate default (maximum available for Lease Time, no minimum for Minimum Lease Time, as soon as possible for Start Time, no maximum for Maximum Start Time, one for Number of Addresses Requested, and any addresses available for List of Address Ranges). The Multicast Scope option MUST be included in the REQUEST message so that the server knows what scope should be

used. The Current Time option MUST be included if the Start Time or Maximum Start Time options are included.

If a client sends a REQUEST message and does not receive any ACK or NAK messages in response after an appropriate delay, the client MAY resend its REQUEST message, as described in [section 2.3](#).

If the server responds with a NAK or fails to respond within a reasonable (implementation-dependent) delay [NO-RESPONSE-DELAY], the

client MAY try to find another server by sending a DISCOVER message with another xid or sending a REQUEST message with another xid to another server.

[2.2.5](#). ACK

The ACK message is used by a MADCAP server to respond affirmatively to an INFORM, REQUEST, or RELEASE message. The server unicasts the ACK message to the client from which it received the message to which it is responding.

The set of options included with an ACK message differs, depending on what sort of message it is responding to.

If the ACK message is responding to an INFORM message, it SHOULD include any options requested by the client using the Option Request List option in the INFORM message. If the Option Request List option was not included in the INFORM message, the server SHOULD include the Multicast Scope List option.

If the ACK message is responding to a REQUEST message, it MUST include Lease Time, Multicast Scope, and List of Address Ranges options. It MAY include a Start Time option. If a Start Time option is included, a Current Time option MUST also be included. If no Start Time option is included, the lease is assumed to start immediately.

If the ACK message is responding to a RENEW message, it MUST include Lease Time, Multicast Scope, and List of Address Ranges options. It MAY include a Start Time option. If a Start Time option is included, a Current Time option MUST also be included. If no Start Time option is included, the lease is assumed to start immediately.

If the ACK message is responding to a RELEASE message, it MUST only include Server Identifier and Client Identifier options.

As with all messages sent by the server, the xid field MUST match the xid field included in the client request to which this message is responding. The Client Identifier option MUST be included, with the value matching the one included in the client request. The Server Identifier option MUST be included, with the value being the server's IP address. And the packet MUST NOT be retransmitted.

[2.2.6.](#) NAK

The NAK message is used by a MADCAP server to respond negatively to a REQUEST, RELEASE, or RENEW message. The server unicasts the NAK message to the client from which it received the message to which it is responding.

As with all messages sent by the server, the xid field MUST match the xid field included in the client request to which this message is responding. The Client Identifier option MUST be included, with the value matching the one included in the client request. The Server Identifier option MUST be included, with the value being the server's IP address. And the packet MUST NOT be retransmitted.

[2.2.7.](#) RELEASE

The RELEASE message is used by a MADCAP client that wants to deallocate one or more multicast addresses before their lease expires.

The client unicasts the RELEASE message to the MADCAP server from which it allocated the addresses. If the selected server can process the request successfully, it SHOULD unicast an ACK message to the client. Otherwise, it SHOULD unicast a NAK to the client.

The lease to be released is whichever one was allocated with a Client Identifier option matching the one provided in the RELEASE message. It is not possible to release only part of the addresses in a single lease.

If a client sends a RELEASE message and does not receive any ACK or NAK messages in response after an appropriate delay, the client MAY

resend its RELEASE message, as described in [section 2.3](#).

If the server responds with a NAK or fails to respond within a reasonable (implementation-dependent) delay [NO-RESPONSE-DELAY], the client MAY send a RELEASE message with another xid to another server, provided that the Server Mobility feature was used in the original REQUEST message and that this feature is required for the subsequent RELEASE message sent to another server. For more information about the Server Mobility feature, see [section 2.12.1](#).

[2.2.8](#). RENEW

The RENEW message is used by a MADCAP client that wants to renew a multicast address lease, changing the lease time or start time.

The client unicasts the RENEW message to a MADCAP server. If the server can process the request successfully, it SHOULD unicast an ACK message to the client. Otherwise, it SHOULD unicast a NAK to the client.

The lease to be renewed is whichever one was allocated with a Client Identifier option matching the one provided in the RENEW message. The Lease Time, Start Time, and Current Time options MAY be included,

depending on whether the client wishes to change the lease time, the start time, or both.

If a client sends a RENEW message and does not receive any ACK or NAK messages in response after an appropriate delay, the client MAY resend its RENEW message, as described in [section 2.3](#).

If the server responds with a NAK or fails to respond within a reasonable (implementation-dependent) delay [NO-RESPONSE-DELAY], the client MAY send a RENEW message with another xid to another server, provided that the Server Mobility feature was used in the original REQUEST message and that this feature is required for the subsequent RENEW message sent to another server. For more information about the Server Mobility feature, see [section 2.12.1](#).

[2.2.9](#). Options Allowed

Table 3 summarizes which options are allowed with each message type.

Option -----	INFORM -----	ACK (in response to INFORM) -----
Lease Time	MUST NOT	MUST NOT
Server Identifier	MUST NOT	MUST
Client Identifier	MUST	MUST
Multicast Scope	MUST NOT	MUST NOT
Option Request List	MAY	MUST NOT
Start Time	MUST NOT	MUST NOT
Number of Addresses		
Requested	MUST NOT	MUST NOT
Requested Language	MAY	MUST NOT
Multicast Scope List	MUST NOT	MAY
List of Address Ranges	MUST NOT	MUST NOT
Current Time	MUST NOT	MAY
Feature List	MAY	MAY
Retry Time	MUST NOT	MUST NOT
Minimum Lease Time	MUST NOT	MUST NOT
Maximum Start Time	MUST NOT	MUST NOT
Option -----	DISCOVER -----	OFFER -----
Lease Time	MAY	MUST
Server Identifier	MUST NOT	MUST
Client Identifier	MUST	MUST
Multicast Scope	MUST	MUST
Option Request List	MUST NOT	MUST NOT
Start Time	MAY	MAY
Number of Addresses		
Requested	MAY	MUST NOT

Requested Language	MUST NOT	MUST NOT
Multicast Scope List	MUST NOT	MUST NOT
List of Address Ranges	MAY	MAY
Current Time	MAY	MAY
Feature List	MAY	MAY
Retry Time	MUST NOT	MUST NOT
Minimum Lease Time	MAY	MUST NOT
Maximum Start Time	MAY	MUST NOT
Option -----	REQUEST -----	ACK (in response to REQUEST) -----

Lease Time	MAY	MUST
Server Identifier	MUST (if multicast)	MUST
Client Identifier	MUST	MUST
Multicast Scope	MUST	MUST
Option Request List	MUST NOT	MUST NOT
Start Time	MAY	MAY
Number of Addresses		
Requested	MAY	MUST NOT
Requested Language	MUST NOT	MUST NOT
Multicast Scope List	MUST NOT	MUST NOT
List of Address Ranges	MAY	MUST
Current Time	MAY	MAY
Feature List	MAY	MAY
Retry Time	MUST NOT	MAY
Minimum Lease Time	MAY	MUST NOT
Maximum Start Time	MAY	MUST NOT
Option	RENEW	ACK (in response to RENEW)
-----	-----	-----
Lease Time	MAY	MUST
Server Identifier	MUST NOT	MUST
Client Identifier	MUST	MUST
Multicast Scope	MUST NOT	MUST
Option Request List	MUST NOT	MUST NOT
Start Time	MAY	MAY
Number of Addresses		
Requested	MUST NOT	MUST NOT
Requested Language	MUST NOT	MUST NOT
Multicast Scope List	MUST NOT	MUST NOT
List of Address Ranges	MUST NOT	MUST
Current Time	MAY	MAY
Feature List	MAY	MAY
Retry Time	MUST NOT	MAY
Minimum Lease Time	MAY	MUST NOT
Maximum Start Time	MAY	MUST NOT

Option	RELEASE	ACK (in response to RELEASE)
-----	-----	-----
Lease Time	MUST NOT	MUST NOT
Server Identifier	MUST NOT	MUST

Client Identifier	MUST	MUST
Multicast Scope	MUST NOT	MUST NOT
Option Request List	MUST NOT	MUST NOT
Start Time	MUST NOT	MUST NOT
Number of Addresses		
Requested	MUST NOT	MUST NOT
Requested Language	MUST NOT	MUST NOT
Multicast Scope List	MUST NOT	MUST NOT
List of Address Ranges	MUST NOT	MUST NOT
Current Time	MUST NOT	MAY
Feature List	MAY	MAY
Retry Time	MUST NOT	MUST NOT
Minimum Lease Time	MUST NOT	MUST NOT
Maximum Start Time	MUST NOT	MUST NOT
Option	NAK	
-----	---	
Lease Time	MUST NOT	
Server Identifier	MUST	
Client Identifier	MUST	
Multicast Scope	MUST NOT	
Option Request List	MUST NOT	
Start Time	MUST NOT	
Number of Addresses		
Requested	MUST NOT	
Requested Language	MUST NOT	
Multicast Scope List	MUST NOT	
List of Address Ranges	MUST NOT	
Current Time	MUST NOT	
Feature List	MAY	
Retry Time	MUST NOT	
Minimum Lease Time	MUST NOT	
Maximum Start Time	MUST NOT	

Table 3: Options allowed in MADCAP messages

[2.3.](#) Retransmission

MADCAP clients are responsible for all message retransmission. The client **MUST** adopt a retransmission strategy that incorporates an exponential backoff algorithm to determine the delay between retransmissions. The delay between retransmissions **SHOULD** be chosen to allow sufficient time for replies from the server to be delivered based on the characteristics of the internetwork between the client

and the server.

For example, in a 10Mb/sec Ethernet internetwork, the delay before the first retransmission SHOULD be at least 4 seconds. The delay before the next retransmission SHOULD be at least 8 seconds. The retransmission delay SHOULD be doubled with subsequent retransmissions up to a maximum of 64 seconds. The client MAY provide an indication of retransmission attempts to the user as an indication of the progress of the process. The client MAY halt retransmission at any point.

[2.4.](#) The Client Identifier

The Client Identifier option is included in each message sent by an MADCAP client. Its value is used to identify a particular client and MUST be unique across all clients in a multicast address allocation domain. In addition, for each lease requested by a client, the client must use a different client identifier. This allows the lease to be identified using just the client identifier.

The first octet of the client identifier is the client identifier type. If the client identifier type is 0, the remainder of the client identifier is a long random number (such as 128 bits from a decent random number generator). If the client identifier type is 1, the remainder of the client identifier is a two octet IANA-defined address family number, an address from the specified address family, and a client-specific identifier (such as a sequence number or the current time).

New MADCAP client identifier types may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs approved by the IESG.

The MADCAP server does not need to parse the client identifier. It SHOULD use the client identifier only as an opaque identifier, which must be unique for each lease. The purpose of defining different client identifier types is to allow MADCAP clients that already have a globally unique address to avoid the possibility of client ID collisions by using this address together with a client-specific identifier. MADCAP clients that do not have a globally unique address SHOULD use client identifier type 0.

In addition to associating client and server messages (along with the xid field, as described in the next section), the Client Identifier is used to control who can renew or release a lease. MADCAP servers MUST require that the client identifier used in a RENEW or RELEASE message match the client identifier used in the initial REQUEST

message. If not, the server MUST ignore the message. To provide true

security with this technique, the confidentiality of the client identifier must be maintained by encrypting all messages that contain it.

[2.5](#). Associating Client and Server Messages

Messages between clients and servers are associated with one another using the Client Identifier option and the xid field. For each transaction initiated by a client, the client MUST generate an xid value that is unique for that client identifier and likely to be unique across all client identifiers. For instance, a client might start with a random xid and increment from there. The client identifier option and xid field MUST be included in each message sent by the client or the server.

The client MUST check the client identifier option and xid field in each incoming message to ensure that they match the client identifier and xid for an outstanding transaction. If not, the message MUST be discarded. The server MUST check the client identifier option and xid field in each incoming message to establish the proper context for the message. If the message is inappropriate for its context, it SHOULD be discarded.

A transaction can be an attempt to allocate a multicast address (consisting of DISCOVER, OFFER, REQUEST, ACK, and NAK messages), an attempt to renew a lease (consisting of RENEW, ACK, and NAK messages), an attempt to release a previously allocated multicast address (consisting of RELEASE, ACK, and NAK messages), or an attempt to acquire configuration parameters (consisting of INFORM and ACK messages).

[2.6](#). Multicast Scopes

[RFC 2365](#) [3] provides for dividing the multicast address space into a number of administratively scopes. Routers should be configured so that each scope corresponds to a particular partition of the network into disjoint regions. Messages sent to a multicast address that falls within a certain administrative scope should only be delivered to hosts that have joined that multicast group *and* fall within the same region as the sender. For instance, packets sent to an address

in the organization-local scope should only be delivered to hosts that have joined that group and fall within the same organization as the sender.

Different sets of scopes may be in effect at different places in the network and at different times. Before attempting to allocate an address from an administrative scope (other than global or link-level scope, which are always in effect), a MADCAP client SHOULD

determine that the scope is in effect at its location at this time. Several techniques that a MADCAP client may use to determine the set of administrative scopes in effect (the scope list) are: manual configuration or configuration via MADCAP (using the Multicast Scope List option).

If a MADCAP client is unable to determine its scope list using one of these techniques, it MAY temporarily assume a scope list consisting of a single scope. If it is using IPv4, it SHOULD use IPv4 Local Scope (239.255.0.0/16), with a maximum TTL of 16. If it is using IPv6, it SHOULD use SCOP 3, with a maximum TTL of 16. Using this temporary scope list, it MAY attempt to contact a MADCAP server that can provide a scope list for it.

When a MADCAP client requests an address with a DISCOVER or REQUEST message, it MUST specify the administrative scope from which the address should be allocated. This scope is indicated with the Multicast Scope option. Likewise, the server MUST include the Multicast Scope option in all OFFER messages and all ACK messages sent in response to REQUEST messages.

[2.7](#). Multicast TTL

Another way to limit propagation of multicast messages is by setting the TTL field before sending them. This technique has several disadvantages in comparison to administratively scoped multicast addresses, but it is currently in widespread usage.

In an environment where administrative scoping is not in use and TTL scoping is, a MADCAP server MAY return a scope list that includes artificial scopes with TTLs less than 255. The MADCAP client MAY then allocate addresses from these scopes, but MUST NOT set the TTL field of any packet sent to such an address to a value greater than the

maximum TTL indicated in the scope list. In particular, it is recommended that the MADCAP Server Multicast Addresses associated with the IPv4 Local Scope (or SCOP 3 for IPv6) be blocked so that packets sent to them with TTL of 16 are not sent outside an appropriate boundary. This will allow MADCAP clients to use their default behavior for finding MADCAP servers.

In an environment where administrative scoping is in use, the maximum TTLs in the scope list SHOULD be 255. The admin scope zone boundary routers will prevent leakage of MADCAP packets beyond appropriate limits.

[2.8.](#) Locating MADCAP Servers

There are several ways for a MADCAP client to locate a MADCAP server.

For instance, the client may obtain a DNS name or IP address from DHCP or manual configuration.

One particularly convenient technique (and the recommended one) is for the client to send an INFORM message to a MADCAP Server Multicast Address and wait for ACK responses. This technique is described in more detail in the next section.

[2.9.](#) MADCAP Server Multicast Address

Each multicast scope has an associated MADCAP Server Multicast Address. This address has been reserved by the IANA as the address with a relative offset of -1 from the last address of a multicast scope.

A MADCAP client looking for servers that can provide multicast allocation services MAY send an INFORM message to a MADCAP Server Multicast Address. Any MADCAP servers listening to this address SHOULD respond with a unicast ACK message to the client if they wish to offer a response.

The MADCAP Server Multicast Address used by a client MAY be established by configuration (manually or via DHCP). If a client has no such configuration, it SHOULD use the MADCAP Server Multicast Address associated with IPv4 Local Scope (or SCOP 3 for IPv6) with maximum TTL of 16, unless otherwise configured.

[2.10.](#) Going Beyond the Local Scope

If a client receives no response to a message sent to a MADCAP Server Multicast Address (after retransmission), it MAY send the message to a larger scope and repeat this process as necessary. However, the client MUST NOT send a MADCAP message to the MADCAP Server Multicast Address associated with the global scope.

This technique allows MADCAP servers to provide services for scopes in which they do not reside. However, this is a dangerous and complicated technique and is **not** recommended at this time. Therefore, MADCAP clients SHOULD only send multicast messages to the MADCAP Server Multicast Address corresponding to the IPv4 Local Scope (or SCOP 3, if using IPv6), unless configured otherwise.

MADCAP servers that wish to provide services for scopes in which they do not reside MUST make special efforts to ensure that their services meet clients' needs for largely conflict-free allocation and accurate scope list information. In particular, coordinating with other servers that provide services for this scope may be difficult. Also, establishing which scope the client is in may be difficult. If a

MADCAP server is not prepared to provide services for scopes in which it does not reside, it SHOULD ignore REQUEST, RENEW, and RELEASE messages whose scope does not match (or is enclosed by) the scope of the MADCAP Server Multicast Address on which the request was received. It SHOULD also ignore INFORM messages that are not received on the MADCAP Server Multicast Address for IPv4 Local Scope.

[2.11.](#) Clock Skew

The Current Time option is used to detect and handle clock skew between MADCAP clients and servers. This option MUST be included in any MADCAP message that includes an absolute time (such as the Start Time option). It MAY be included in any DISCOVER, OFFER, REQUEST, or ACK message.

Clock skew is a situation where two systems have clocks that are not synchronized. MADCAP servers SHOULD expect and tolerate a small amount of clock skew with their clients by ensuring that multicast addresses are allocated for an extra period of time

[CLOCK_SKEW_ALLOWANCE] on either side of the lease given to the client. However, large amounts of clock skew require special handling. The value of [CLOCK_SKEW_ALLOWANCE] is implementation dependent. A reasonable value might be one hour.

The Current Time option contains the sender's opinion of the current time in UTC at or about the time the message was assembled. Because of delays in transmission and processing, this value will rarely match the receiver's opinion of the current time at the time the option is processed by the receiver.

If a MADCAP server detects clock skew greater than [CLOCK_SKEW_ALLOWANCE], it SHOULD ignore the client. The server MAY log a message. While it would be possible for the server to correct for the clock skew, this would probably cause other problems, such as the client advertising a session with an incorrect time. This is an error condition and should be handled as such.

[2.12](#). Optional Features

Each MADCAP client or server MAY implement one or more optional features. Optional features of MADCAP are identified with a two octet feature code.

A MADCAP client MAY request, require, or indicate support for an optional feature by including a Feature List option in a message. A MADCAP server MUST either provide all required features for this message or ignore that message. For more information about optional features, see the description of the Feature List option.

Table 4 lists the feature codes defined at this time and sections 2.12.1 and 2.12.2 describe how these features work.

New MADCAP feature codes may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs approved by the IESG.

Feature Code	Feature Name
-----	-----
0	Server Mobility
1	Retry After

Table 4: MADCAP message types

[2.12.1. Server Mobility](#)

The Server Mobility feature allows an address allocated on one MADCAP server to be renewed or released on a different MADCAP server. This requires communication and coordination among MADCAP servers. The primary benefits are immunity to the failure of a single MADCAP server and perhaps greater performance through load balancing.

In order to take advantage of the Server Mobility feature, a MADCAP client must ensure that the feature is implemented by both the server that is used for the original allocation and the server that is used for the renewal or release. The best way to ensure this is to include the Server Mobility feature in the required list of a Feature List option in the REQUEST message used to allocate the address (and the DISCOVER message, if one is used). When the time comes to renew or release the address, the client may, of course, send a unicast RENEW or RELEASE message to the server from which it allocated the address. However, if this server is unavailable, the client may send the RENEW or RELEASE message to any other server that includes the Server Mobility feature in its list of supported features. The client may find such a server by (for instance) sending an INFORM message with a required Server Mobility feature.

If the MADCAP client does not want to require this feature when allocating addresses, it may include the feature in the requested list of a Feature List option and see if the server includes the feature in the supported list of a Feature List option in the ACK message.

Even if the Server Mobility feature is used, there is no guarantee that a server will be available to perform the renewal or release or that the renewal or release will succeed. Server connectivity may have failed, for instance.

[2.12.2. Retry After](#)

The Retry After feature allows a MADCAP server to ask the MADCAP client to retry its request later, as may be required when allocating large numbers of addresses or allocating addresses for a long period

of time.

If a MADCAP client includes the Retry After feature in the supported list of a Feature List option in a REQUEST message, a MADCAP server that also implements the Retry After feature MAY decide to perform a future allocation. In this case, the MADCAP server will include an empty List of Address Ranges option in its ACK message, a Feature List option that includes the Retry After feature in the supported list, and a Retry Time option with a time after which the client should retry the REQUEST.

The client MUST NOT include the Retry After feature in the requested or required list of a Feature List option, since the decision about whether Retry After is desirable should be left to the MADCAP server.

At some later time (preferably after the time indicated in the Retry Time option), the client MAY send a REQUEST message with all the same options as the original REQUEST message (especially the Client Identifier option), but with a new xid value. The server MAY return a normal ACK or NAK message at this point or it MAY continue the transaction to a later time by including an empty List of Address Ranges option in its ACK message, a Feature List option that includes the Retry After feature in the supported list, and a Retry Time option with a later time after which the client should retry the REQUEST.

At any point after receiving the initial ACK message with the Retry Time option, the client MAY terminate the allocation process and any accompanying lease by sending to the server performing the allocation (or another server if the Server Mobility feature is also in effect) a RELEASE message with the Client Identifier included in the original REQUEST message.

The Retry After feature may also be used when renewing a lease. In this case, the description above applies except that the client sends a RENEW message instead of a REQUEST message.

If a client sends a RENEW message with a Client Identifier that matches a lease which is currently undergoing allocation with the Retry After feature in response to a REQUEST message, the server SHOULD ignore the RENEW message. Also, if a client sends a RENEW message with a Client Identifier that matches a lease which is currently undergoing allocation with the Retry After feature in

response to a RENEW message, but the options supplied with the two RENEW messages do not match, the server SHOULD ignore the second RENEW message.

[3. MADCAP Options](#)

The following options are defined for use in MADCAP messages. The options are listed in numerical order.

[3.1. End](#)

The End option marks the end of valid information in the options field. This option MUST be included at the end of the options field in each MADCAP message.

The code for this option is 0, and its length is 0.

Code	Len
0	0

[3.2. Lease Time](#)

This option is used in a client request (DISCOVER, REQUEST, or RENEW) to allow the client to request a lease time for the multicast address. In a server reply (OFFER or ACK), a MADCAP server uses this option to specify the lease time it is willing to offer.

The time is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 1, and its length is 4.

Code	Len	Lease Time			
1	4	t1	t2	t3	t4

[3.3. Server Identifier](#)

This option contains the IP address of a MADCAP server. A two octet address family (as defined by IANA) is stored first, followed by the address. The address family for this address is not determined by the `addrfamily` field so that addresses from one family may be allocated while communicating with a server via addresses of another family.

Internet Draft [draft-ietf-malloc-madcap-04.txt](#)

February 1999

All messages sent by a MADCAP server MUST include a Server Identifier option with the IP address of the server sending the message.

MADCAP clients MUST include a Server Identifier option in multicast REQUEST messages in order to indicate which OFFER message has been accepted.

The code for this option is 2, and its minimum length is 3.

Code	Len	Address Family	Address
2	n	family	a1 ...

[3.4. Client Identifier](#)

This option is used by MADCAP clients to specify their unique identifier. For more information about this option and how it is used, see [section 2.4](#).

The code for this option is 3, and its minimum length is 1.

Code	Len	Client Identifier
3	n	i1 i2 ...

[3.5. Multicast Scope](#)

The multicast scope option is used by the client to indicate the requested multicast scope in a DISCOVER or REQUEST message. It is also used by the MADCAP server to indicate the scope of an assigned address.

The client may obtain the scope list through the Multicast Scope List option or using some other means. The scope id is the first multicast address in the scope. The address family of the Scope ID is determined by the `addrfamily` field.

The code for this option is 4, and its minimum length is 1.

Code	Len	Scope ID
------	-----	----------

4	n	i1	...
---	---	----	-----

[3.6.](#) Option Request List

This option is used by a MADCAP client in an INFORM message to request that certain options be included in the server's ACK response. The server SHOULD try to include the specified options in its response, but is not required to do so.

The format of this option is a list of option codes.

The code for this option is 5 and the minimum length is 2.

Code	Len	Requested Options
5	n	Option1

[3.7.](#) Start Time

The Start Time option specifies the starting time for a multicast address lease.

A client may include this option in a DISCOVER, RENEW, or REQUEST message to request a multicast address for use at a future time. A server may include this option in an OFFER message or in an ACK in response to REQUEST or RENEW message to indicate that a lease has been granted which starts at a future time.

If the Start Time option is present, the IP Address Lease Time option specifies the duration of the lease beginning at the Start Time option value.

If the Start Time option is present, the Current Time option MUST also be present, as described in [section 2.11](#).

The time value is an unsigned 32 bit integer in network byte order

giving the number of seconds since 00:00 UTC, 1st January 1970. This can be converted to an NTP timestamp by adding decimal 2208988800. This time format will not wrap until the year 2106.

The code for this option is 6 and the length is 4.

Code	Len	Time			
6	4	t1	t2	t3	t4

[3.8.](#) Number of Addresses Requested

This option specifies the minimum and desired number of addresses requested by the client. It is only used in DISCOVER and REQUEST messages and is only sent by the client.

The minimum and desired number of addresses requested are unsigned 16 bit integers in network byte order. The minimum MUST be less than or equal to the desired number. If a message is received where this is not the case, the message MUST be ignored.

The client MAY obtain more than one address either by repeating the protocol for every address or by requesting several addresses at the same time via this option. When the client is requesting only one address, this option SHOULD not be included. A MADCAP server receiving a DISCOVER or REQUEST packet including this option MUST include between minimum and desired number of addresses in any OFFER or ACK response.

The code for this option is 7 and the length is 4.

Code	Len	Minimum	Desired
7	4	min	desired

[3.9.](#) Requested Language

This option specifies the language in which the MADCAP client would like strings such as zone names to be returned. It is only included in an INFORM message sent by the client. It is an [RFC 1766](#) [6] language tag. The proper way to handle this tag with respect to zone names is discussed further in the definition of the Multicast Scope List option.

The code for this option is 8 and the minimum length is 0.

Code	Len	Language Tag	
+-----+-----+-----+-----+-----+...+-----+			
8	n	L1	Ln
+-----+-----+-----+-----+-----+...+-----+			

[3.10](#). Multicast Scope List

This option is sent by the server in an ACK message in response to an INFORM message sent by the client.

If the client did not include a Requested Language option in its

INFORM message, the MADCAP server SHOULD return all zone names for each zone. If the client included a Requested Language option in its INFORM message, the MADCAP server MUST return no more than one zone name for each zone. For each zone, the MADCAP server SHOULD first look for a zone name that matches the requested language tag (using a case-insensitive ASCII comparison). If any names match, one of them should be returned. Otherwise, the MADCAP server SHOULD choose another zone name to return (if any are defined). It SHOULD give preference to zone names that are marked to be used if no name is available in a desired language.

The code for this option is 9 and the minimum length is 0.

The format of the multicast scope list option is:

Code	Len	Count	Scope List	
+-----+-----+-----+-----+-----+...+-----+				
9	p	m	L1	Lm
+-----+-----+-----+-----+-----+...+-----+				

The scope list is a list of m tuples, where each tuple is of the

server: Inside abcd.com with addresses 239.192.0.0-239.195.255.255, and world with addresses 224.0.1.0-238.255.255.255. Then this option will be given as:

```

      Code      Len      Count
+-----+-----+-----+-----+-----+...
|      9      |      51      |  2  |
+-----+-----+-----+-----+-----+...

      Scope ID      Last Address      TTL Name  Name  Lang  Language
                                Count Flags Length Tag
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...-...-+...
|239|192| 0 | 0 |239|195|255|255|10 | 1  | 128 |  2  | en  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...-...-+...

      Name
      Length Name
+-----+---+---+...-+---+---+...
|  15  | Inside abcd.com |
+-----+---+---+...-+---+---+...

      Scope ID      Last Address      TTL Name  Name  Lang  Language
                                Count Flags Length Tag
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...-...-+...
|224| 0 | 1 | 0 |238|255|255|255|16 | 1  | 128 |  2  | en  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+...-...-+...

      Name
      Length Name
+-----+---+...-+
|  5  | world |
+-----+---+...-+

```

[3.11.](#) List of Address Ranges

This option is used by the server to provide the list of all the address ranges allocated to the client.

This option is also used by the client when requesting a lease for a specific set of addresses.

The address family of the addresses is determined by the `addrfamily` field.

The code for this option is 10 and the minimum length is 0.

Code	Len	Address Range List			
+-----+-----+-----+-----+-----+-----+...+-----+					
10	n	L1	L2		Ln
+-----+-----+-----+-----+-----+-----+...+-----+					

where the Address Range List is of the following format.

StartAddress1	BlockSize1	StartAddress2	BlockSize2	...
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+				
... S1 ...	B11 B12	... S2 ...	B21 B22	
+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+-----+...+-----+				

[3.12.](#) Current Time

This option is used to express what the sender thinks the current time is. This is useful for detecting clock skew. This option **MUST** be included if the Start Time or Maximum Start Time options are used, as described in [section 2.11](#).

The time value is an unsigned 32 bit integer in network byte order giving the number of seconds since 00:00 UTC, 1st January 1970. This can be converted to an NTP [\[4\]](#) timestamp by adding decimal 2208988800. This time format will not wrap until the year 2106.

The code for this option is 11 and the length is 4.

Code	Len	Time			
+-----+-----+-----+-----+-----+-----+-----+-----+					
11	4	t1	t2	t3	t4
+-----+-----+-----+-----+-----+-----+-----+-----+					

[3.13.](#) Feature List

This option lists optional MADCAP features supported, requested, or required by the sender. This option **MAY** be included in any message

sent by a MADCAP server or client.

Optional features of MADCAP are identified with a two octet feature code. New MADCAP feature codes may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs approved by the IESG.

The Feature List option consists of three separate lists: supported features, requested features, and required features. Each list consists of an unordered list of feature codes. Features in the supported list are features that the sender supports. Features in the requested list are features that the sender would like the receiver to support and use. However, it is OK if the receiver does not support or use those features. Features in the required list are features that the sender requires. If the receiver does not support those features, it MUST ignore the message.

If a MADCAP client includes the Feature List option in a message, it MAY include features in any of the lists: supported, requested, and required. A MADCAP server that receives a message containing the Feature List option from a client MUST ignore the entire message if it does not support all of the features in the required list. If it supports all of the features in the required list, it MUST implement them as appropriate for this message. It SHOULD try to implement the features in the requested list and it MAY implement any of the features in the supported list. If an optional feature (such as Retry After) is not included in any part of the Feature List option included in the client's message (or if the client does not include a Feature List option in its message), the server MUST NOT use that feature in its response. If the server does respond to a client's message that includes a Feature List option, it MUST include a Feature List option listing the features that it supports and with an empty requested and required features list.

The code for this option is 12 and the minimum length is 6.

Code	Len	Supported	Requested	Required
12	n	FL1	FL2	FL3

where each of the Feature Lists is of the following format:

Feature Count	Feature Code 1	...	Feature Code m
m	FC1		FCm

[3.14.](#) Retry Time

The Retry Time option specifies the time at which a client should retry a REQUEST or RENEW message when using the Retry After feature.

This option should only be sent by a MADCAP server in an ACK when responding to a REQUEST or RENEW message that includes the Retry After feature in the supported, requested, or required list. For more discussion of Retry After, see [section 2.12.2](#).

If the Retry Time option is present, the Current Time option **MUST** also be present.

The time value is an unsigned 32 bit integer in network byte order giving the number of seconds since 00:00 UTC, 1st January 1970. This can be converted to an NTP timestamp by adding decimal 2208988800. This time format will not wrap until the year 2106.

The code for this option is 13 and the length is 4.

Code	Len	Time			
13	4	t1	t2	t3	t4

[3.15.](#) Minimum Lease Time

This option is used in a client request (DISCOVER, REQUEST, or RENEW) to allow the client to specify a minimum lease time for the multicast address. If a server cannot meet this minimum lease time, it **SHOULD** ignore a DISCOVER or send a NAK in response to a RENEW or REQUEST (unless it's a multicast REQUEST with a Server Identifier that doesn't match the server's, which should be ignored altogether).

The time is in units of seconds, and is specified as a 32-bit unsigned integer.

The code for this option is 14, and its length is 4.

Code	Len	Lease Time			
14	4	t1	t2	t3	t4

[3.16.](#) Maximum Start Time

The Maximum Start Time option specifies the latest starting time that the client is willing to accept for a multicast address lease.

A client may include this option in a DISCOVER, RENEW, or REQUEST message to specify that it does not want to receive a lease with a starting time later than the specified value. If a server cannot meet this maximum start time, it SHOULD ignore a DISCOVER or send a NAK in response to a RENEW or REQUEST (unless it's a multicast REQUEST with a Server Identifier that doesn't match the server's, which should be ignored altogether).

If the Maximum Start Time option is present, the Current Time option MUST also be present, as described in [section 2.11](#).

The time value is an unsigned 32 bit integer in network byte order giving the number of seconds since 00:00 UTC, 1st January 1970. This can be converted to an NTP timestamp by adding decimal 2208988800. This time format will not wrap until the year 2106.

The code for this option is 15 and the length is 4.

Code	Len	Time
15	4	t1 t2 t3 t4

[4.](#) Security Considerations

MADCAP has relatively basic security requirements. At present there is no way of enforcing authorized use of multicast addresses in the multicast routing/management protocols. Therefore, it is not possible to identify unauthorized use of multicast address by an adversary. Moreover, a multicast address allocated to a user/system can be used by other systems without violating terms of the multicast address allocation. For example, a system may reserve an address to be used for a work group session where each and every member of the work group is allowed to transmit packets using the allocated group address. In other words, the multicast address allocation protocol does not dictate how the address should be used, it only dictates the

time period for which it can be used and who gets to release it or renew it. When an address is allocated to a system/user, it basically means that no other user/system (most likely) will be allocated that address for the time period, without any restrictions on its use.

To protect against rogue MADCAP servers (mis-configured servers and intentional), clients in certain situations would like to authenticate the server. Similarly, for auditing or book-keeping purposes, the server may want to authenticate clients. Moreover, in some cases, the server may have certain policies in place to restrict number of addresses that are allocated to a system or a user. This feature is of much value when a well behaved but naive user or client

requests a large number of addresses, and therefore, inadvertently impacts other users or systems. Therefore, an administrator may want to exert a limited amount of control based on the client identification. The client identification could be based on the system or user identity. In most practical situations, system identification will suffice, however, particularly in case of multi-user systems, at times, user identification will play an important role. Therefore, authentication capabilities based on user identification may be desirable. As usual, data integrity is a strong requirement and if not protected, can lead to many problems including denial of service attacks.

In the case of MADCAP, confidentiality is not a strong requirement. In most of the cases, at least when a multicast address is in use, an adversary will be able to determine information that was contained in the MADCAP messages. In some cases, the users/systems may want to protect information in the MADCAP messages so that an adversary is not able to determine relevant information in advance and thus, plan an attack in advance. For example, if an adversary knows in advance (based on MADCAP messages) that a particular user has requested large number of address for certain time period and scope, he may be able to guess the purpose behind such request and target an attack. If MADCAP servers are configured to allow renewal or release purely on the basis of knowledge of the Client Identifier option (without consideration of authentication), preserving the confidentiality of MADCAP messages becomes more important. Also, there may be features added to the protocol in future that may have stronger confidentiality requirements.

The IPSEC protocol [8] meets client/server identification and integrity protection requirements stated above, requires no modification to the MADCAP protocol, and leverages extensive work in IETF and industry. Therefore, when security is a strong requirement, IPSEC SHOULD be used for protecting all the unicast messages of MADCAP protocol. When IPSEC based security is in use, all the multicast packets except INFORM MUST be dropped by the MADCAP server. The prevalent implementations of IPSEC support client identification in form of system identification and do not support user identification. However, when desired, IPSEC with appropriate API's may be required to support user identification.

5. IANA Considerations

This document defines several number spaces (MADCAP options, MADCAP message types, MADCAP client identifier types, and MADCAP features). For all of these number spaces, certain values are defined in this specification. New values may only be defined by IETF Consensus, as described in [7]. Basically, this means that they are defined by RFCs

approved by the IESG.

6. Acknowledgments

The authors would like to thank Rajeev Byrisetty, Steve Deering, Peter Ford, Mark Handley, Van Jacobson, David Oran, Thomas Pfenning, Dave Thaler, Ramesh Vyaghrapuri and the participants of the IETF for their assistance with this protocol.

Much of this document is based on [1] and [2]. The authors of this document would like to express their gratitude to the authors of these previous works. Any errors in this document are solely the fault of the authors of this document.

7. References

- [1] Droms, R., "Dynamic Host Configuration Protocol", [RFC 2131](#), March 1997.
- [2] Alexander, S., and R. Droms, "DHCP Options and BOOTP Vendor Extensions", [RFC 2132](#), March 1997.

- [3] Meyer, D., "Administratively Scoped IP Multicast", [RFC 2365](#), July 1998.
- [4] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation and Analysis", [RFC 1305](#), March 1992.
- [5] Yergeau, F., "UTF-8, a transformation format of ISO 10646", [RFC 2279](#), January 1998.
- [6] Alvestrand, H., "Tags for the Identification of Languages", [RFC 1766](#), March 1995.
- [7] Alvestrand, H. and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", [RFC 2434](#), October 1998.
- [8] R. Atkinson, S. Kent, "Security Architecture for the Internet Protocol", [RFC 2401](#), November 1998.

8. Authors' Addresses

Baiju V. Patel
Intel Corp.
2111 NE 25th Ave.
Hillsboro, OR 97124

Phone: 503 264 2422

Patel, Shah, and Hanna

[Page 34]

Internet Draft [draft-ietf-malloc-madcap-04.txt](#)

February 1999

Email: baiju.v.patel@intel.com

Munil Shah
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: 425 703 3924
Email: munils@microsoft.com

Stephen R. Hanna
Sun Microsystems, Inc.
One Network Drive
Burlington, MA 01803

APPENDIX A: Examples

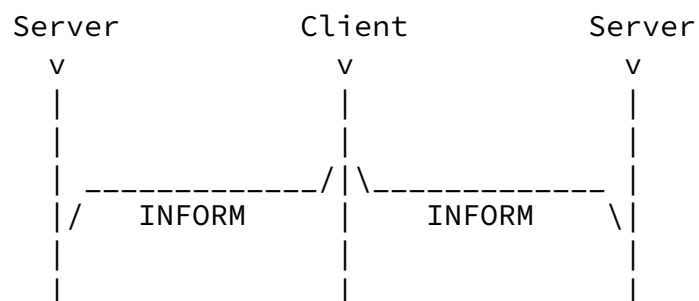
This appendix includes several examples of typical MADCAP protocol exchanges.

1. Multicast Scope List Discovery

In this example, a MADCAP client wants to determine the scope list in effect. The client is using IPv4, so it starts with by multicasting an INFORM packet to the MADCAP Server Multicast Address corresponding to IPv4 Local Scope. This packet includes the Client Identifier option, an Option Request List including the Multicast Scope List option code, and a Requested Language option containing the string "en", since the client is configured to prefer the English language.

Two MADCAP servers respond by sending ACK messages. These ACK messages include the Client Identifier option and xid supplied by the client, the server's Server Identifier, and the Multicast Scope List with one name per scope (the one that most closely matches the language tag "en").

The following figure illustrates this exchange.



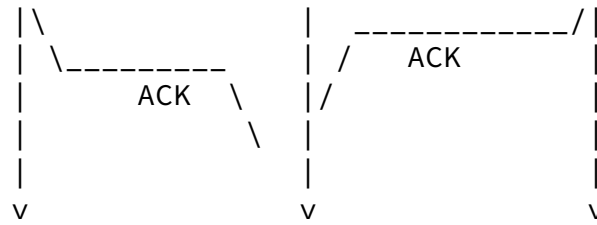


Figure 2: Timeline diagram of messages exchanged
in Multicast Scope List Discovery example

2. Multicast Discovery and Address Allocation

In this example, the MADCAP client wants to allocate a multicast address from the global scope for use during the next two hours.

The client begins by multicasting a DISCOVER packet to the MADCAP Server Multicast Address associated with IPv4 Local Scope. This packet includes the Lease Time, Client Identifier, and Multicast Scope options.

Any servers that receive the DISCOVER packet and can satisfy this request temporarily reserve an address for the client and unicast an OFFER packet to the client. These packets contain the Lease Time, Server Identifier, Client Identifier, and Multicast Scope options.

After a suitable delay, the client multicasts a REQUEST packet to the MADCAP Server Multicast Address. This packet contains all of the options included in the DISCOVER packet, but also includes the Server Identifier option, indicating which server it has selected for the request.

The server whose Server Identifier matches the one specified by the client responds with an ACK packet containing the options included in the OFFER packet, as well as a List of Address Ranges option listing the address allocated. All the other servers that had sent OFFER packets stop reserving an address for the client and forget about the whole exchange.

The client now has a two hour "lease" on the multicast address.

If the client had not received an ACK from the server, it would have

retransmitted its REQUEST packet for a while. If it still received no response, it would start over with a new DISCOVER message.

The following figure illustrates this exchange.

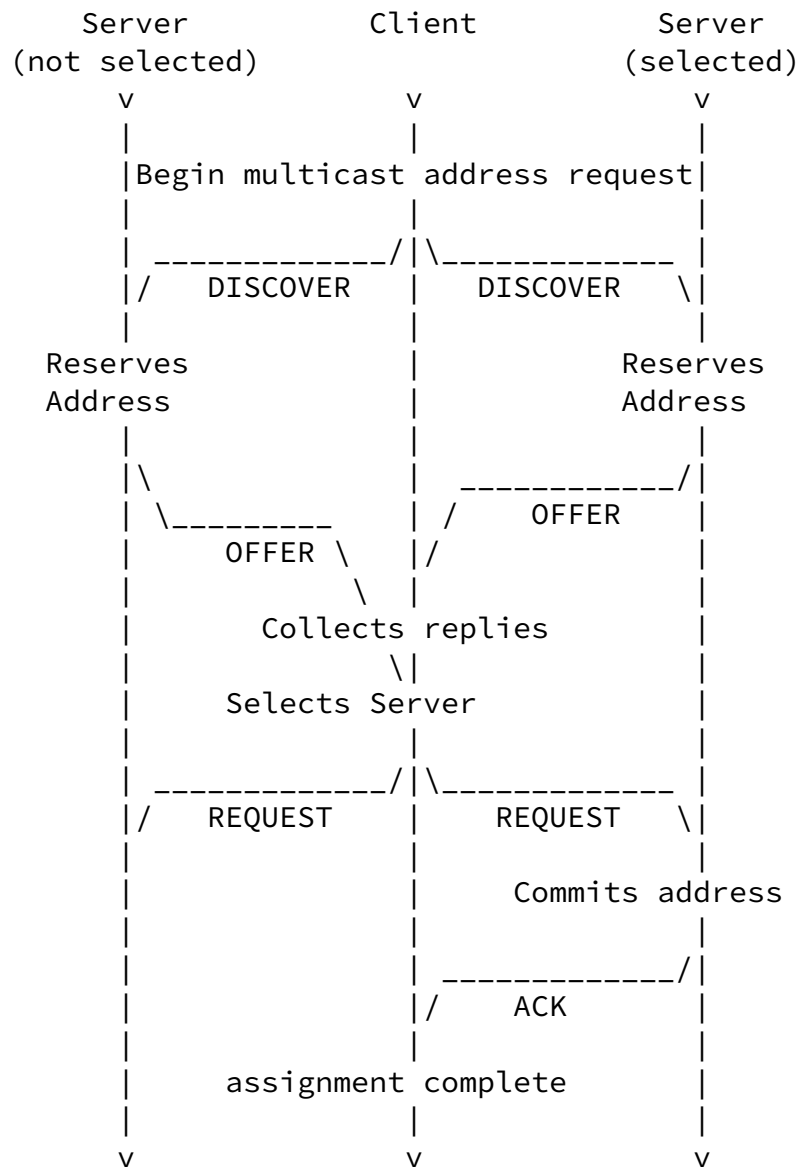


Figure 3: Timeline diagram of messages exchanged in Multicast Address Allocation example

3. Lease Extension

This is a continuation of the previous example. The client has already allocated a multicast address from the global scope for use during the next two hours. Half way through this two hour period, it decides that it wants to extend its lease for another hour.

The client unicasts a RENEW packet to the server from which it allocated the address. This packet includes the Lease Time and Client Identifier options. The Client Identifier matches the one used for the original allocation. The time included in the Lease Time is two hours, since the client wants the lease to expire two hours from the current time.

The server responds with an ACK packet indicating that the lease extension has been granted. This packet includes the Lease Time, Server Identifier, Client Identifier, Multicast Scope, and List of Address Ranges options.

If the server did not want to grant the requested lease extension, it would have responded with a NAK packet with the Client Identifier option.

The following figure illustrates this exchange.

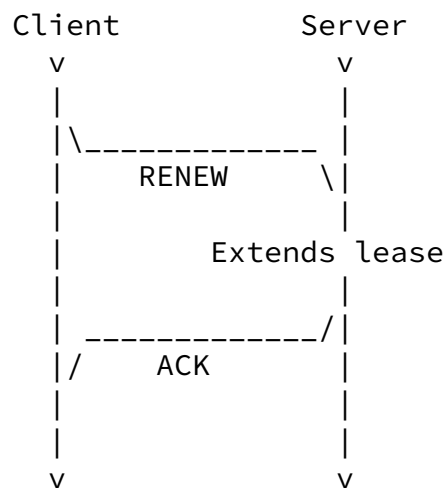


Figure 4: Timeline diagram of messages exchanged in Lease Extension example

[4.](#) Address Release

This is a continuation of the previous example. The client has already allocated a multicast address and extended its lease for another two hours. Half an hour later, the client finishes its use of the multicast address and wants to release it so it can be reused.

The client unicasts a RELEASE packet to the server from which it allocated the address. This packet includes the Client Identifier option. The Client Identifier matches the one used for the original allocation. When the server receives this packet, it cancels the

client's lease on the address and sends an ACK packet to the client indicating that the lease has been released. This packet includes the

Server Identifier and Client Identifier options.

The following figure illustrates this exchange.

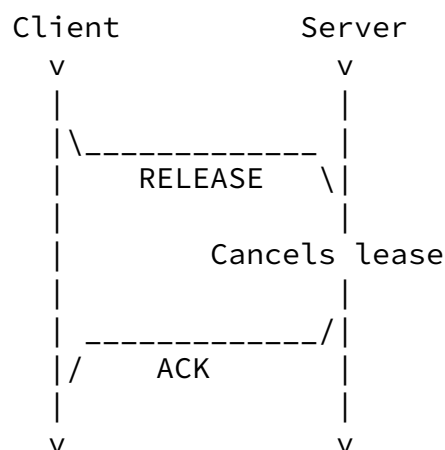


Figure 5: Timeline diagram of messages exchanged in Address Release example

5. Unicast Address Allocation

This is a continuation of the previous example. At some later time, the client decides to allocate another multicast address. Since it has recently worked with a server, it decides to try sending a unicast REQUEST to that server. If this doesn't work, it can always try a multicast DISCOVER, as illustrated in example 2.

The client unicasts a REQUEST packet to the server from which it wants to allocate the address. This packet includes the Lease Time, Client Identifier, and Multicast Scope options.

The server responds with an ACK packet containing the Lease Time, Client Identifier, and Multicast Scope options from the REQUEST packet, as well as the Server Identifier option and a List of Address Ranges option listing the address allocated.

The client now has a lease on the multicast address.

If the client had not received an ACK from the server, it would have retransmitted its REQUEST packet for a while. If it still received no response, it would start over with a multicast DISCOVER message.

The following figure illustrates this exchange.

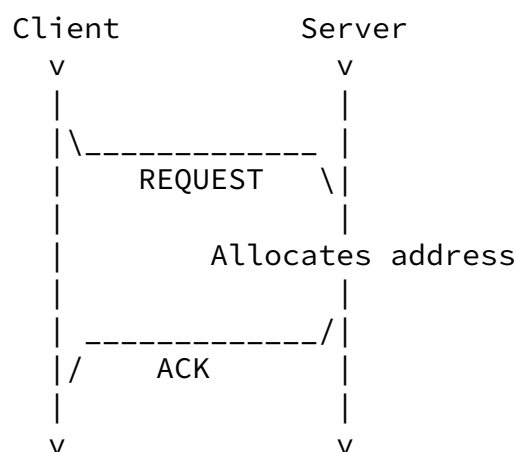


Figure 6: Timeline diagram of messages exchanged in Unicast Address Allocation example

