Multicast and Unicast MAC Address Assignment Protocol (MUMAAP)

Date: April 26th, 2019

Author(s):

Name	Affiliation	Address	Phone	email
Antonio de la Oliva	UC3M, InterDigital	Avda. de la Universidad 30, 28911, Leganes, Madrid, Spain	+34 91 6248803	aoliva@it.uc3m.es
Robert Gazda	InterDigital			Robert.Gazda@interdigital.com

1. Introduction/Background

In this document, we specify the protocol defined in IEEE 802.1CQ for the assignment of multicast and unicast addresses. The protocol is specified with two types of operation:

1. Self-assignment, where a claiming-based process is used, triggered from stations.
2. Server-based procedure where a station contacts an entity which will allocate addresses in a certain pool.

The protocol is designed in order to use a common set of messages for both the self-claiming and the server-based operation. This is done to keep the protocol concise and reduce the overall number of messages. The self-claiming protocol is referred to as MAC Address Self-Assignment Protocol (MASAP) and the server-based version is referred to as MAC Address Server based Assignment Protocol (MASBAP).

The MASAP protocol is based on the IEEE 1722 MAC Address Acquisition Protocol (MAAP). IEEE 1722 defines the MAC Address Acquisition Protocol (MAAP) which is used to self-claim a Multicast address of a pool of addresses allocated to IEEE 1722, to be used as flow identifiers in Audio/Video transmission. MAAP can only be used to self-claim (it does not support server-based assignment). Additionally, it does not include any support from infrastructure and a unicast address is assumed to be allocated to the station executing MAAP. As a result, MAAP "as-is" does not meet the needs of IEEE 802.1CQ.

MASAP operation as defined in this document uses the state machine, events, constants and timers as specified in IEEE 1722. However, MASAP includes support for authoritative responses from a Proxy serving the network, which is able to capture and book-keep all the PROBE messages in the network and directly inform the station of the result of the self-claiming process.

The MASBAP protocol is inspired in a simplified version of DHCP.

In this document, we will refer as station to an end node which runs the client side of the MASAP or MASBAP protocols, while we refer as Server to the infrastructure side running the server side of the MASAP or MASBAP protocols, regardless of being a Proxy or a Server. The Server could be located in operator network infrastructure components and could be located in Customer Premise Equipment, such as gateways, access points, routers, home network controllers, or set-top boxes.

Note: any TBD points in the document will be addresses as part of standardization, such as the assignment of EtherType, etc.

2. Protocol Summary

This document presents two differentiated protocols: i) MAC Address Self-Assignment Protocol (MASAP) and ii) MAC Address Server based Assignment Protocol (MASBAP). MASAP corresponds to the self-assignment operational mode, while MASBAP is used for server based assignment. Both protocols share the same message structure and options, although the behavior is defined in a different way for each.

MASAP uses a PROBE, DEFEND and ANNOUNCE mechanism, which relies on multicast support in the network. Clients select a unicast address (or range of addresses) by randomly selecting a local unicast address out of a pre-established range defined in IEEE 802.1CQ. Once the client has selected a local unicast MAC address, it will PROBE for the availability of that MAC address or a range of addresses (unicast or multicast), by sending a PROBE message to a pre-established multicast address, where all MASAP clients (or proxies) listen. After sending several PROBE messages without receiving any answer, the client understands the address can be allocated to it. Afterwards, the station starts DEFEND and ANNOUCE phases where it listens to PROBEs requesting it's allocated (i.e. self-assigned) range of addresses and announce its allocations periodically within the network.

This basic operation, which is consistent to IEEE 1722, although using another set of parameters and messages, is extended in this document to account for the fact that some IEEE 802 technologies (such as IEEE 802.11) will not support the reception of PROBE messages to stations that are not yet associated (i.e. stations doing the probing phase that do not yet have an assigned MAC address). Therefore, we extend the above client operation by enabling a Proxy in the network (for example the IEEE 802.11 AP) to keep track of the different allocations requested by the clients through MASAP and answer directly to stations issuing PROBEs. This will not only allow the operation of the protocol in technologies such as IEEE 802.11, but also improves the speed of the allocation of MAC addresses, since clients will receive confirmation immediately.

MASBAP uses a four-message exchange, similar to DHCP (discover, offer, request, acknowledge) to allocate addresses to stations. As in MASAP, the client will auto generate one address to be used as source address of its messages and start a DISCOVER, OFFER, REQUEST and ACK message exchange with the server. In response to the offer, the server (or proxy) will offer an address or range of addresses to the client. A client may receive several offers from separate servers. The server operation is stateless, so MAC addresses will be allocated only after reception of the REQUEST message by the server. The server issues an ACK to the client. After which, the client may use the allocated address or range of addresses in the network.

3. Protocol Details

This clause defines the protocol operation for both variants of the MUMAAP protocol.

3.1. MASAP Protocol Operation

MASAP is used to self-claim unicast and multicast addresses following IEEE 802c SLAP definition. The claiming of multicast addresses in the ranges defined by IEEE 1722 Tables B.9 and B10, are out of the scope of this specification and must use the rules defined in the IEEE 1722 MAAP specification.

3.1.1. Message Addressing

MASAP makes use of the following rules for addressing:

- Source MAC address for MASAP_PROBE messages will be chosen randomly from the range shown in Table 1.
- Source MAC address for MASAP_DEFEND and MASAP_ANNOUNCE messages will use the MAC Address previously assigned or the EUI-64/48 assigned to the station.
- Destination MAC address for MASAP_PROBE messages corresponds to the multicast address specified in Table 1.
- Destination MAC address for MASAP_DEFEND and MASAP_ANNOUNCE messages correspond to the source MAC address of the MASAP_PROBE message.

Table 1: Address allocation

Address Range	Function
TBD	Range of addresses to randomly select a source address for MASAP_PROBE messages
TBD	Multicast address used as destination in MASAP_PROBE messages

3.1.2. State Machine

The MASAP protocol is based on the IEEE 1722 MAC Address Acquisition Protocol (MAAP). MASAP operation as defined in this document uses the same state machine, events, constants and timers as specified in IEEE 1722. However, in MASAP, we extend MAAP to support authoritative answers (for example to PROBE requests) from a Proxy serving the network, which is able to capture and book-keep all the PROBE messages in the network and directly inform the station of the result of the self-claiming process.

This modification is shown in the next table, where we copy the state machine of the MAAP protocol and over it (gray cells), define the new state transitions. Note that the table denotes only the state machine transitions for the stations.

Table 2: MASAP State Machine

		STATE		
		INITIAL	**PROBE**	**DEFEND**
EVENT	**Begin!**	generate_address[a] ReserveAddress!	-X-	-X-
	Release![c]	-X-	Stop probe_timer INITIAL	Stop announce_timer INITIAL
	Restart!	generate_address ReserveAddress!	-X-	-X-
	ReserveAddress!	init_maap_probe_count Start probe_timer sProbe PROBE	-X-	-X-
	rProbe![b]	-X-	compare_MAC[d] Stop probe_timer INITIAL/Restart!	sDefend
	rDefend![b]	-X-	Stop probe_timer INITIAL/Restart!	compare_MAC[d] Stop announce_timer INITIAL/Restart!
	rAnnounce![b]	-X-	Stop probe_timer INITIAL/Restart!	compare_MAC[d] Stop announce_timer INITIAL/Restart!
	probeCount!	-X-	Stop probe_timer Start announce_timer sAnnounce DEFEND	-X-
	announcetimer!	-X-	-X-	Start announce_timer sAnnounce
	probetimer!	-X-	Start probe_timer sProbe dec_maap_probe_count	-X-
	PortOperational!	generate_address [a] ReserveAddress!	Stop probe_timer INITIAL/Restart!	Stop announce_timer INITIAL/Restart!
	rProxyAnswer (status == 1)	-X-	Stop probe_timer DEFEND	-X-
	rProxyAnswer (status == 2)	-X-	Stop probe_timer INITIAL/Restart!	Stop probe_timer INITIAL/Restart!
	rProxyAnswer (status == 3)	-X-	Stop probe_timer INITIAL/Restart!	Stop probe_timer INITIAL/Restart!
	rProxyAnswer (status == 3)	INITIAL/STOP	INITIAL/STOP	INITIAL/STOP

[a] A Begin! or PortOperational! event can be initiated with an assigned address range or MASAP can select an address range with the generate_address function. If an address range is supplied with the Begin! or PortOperational! event, generate_address will not be

called and the supplied address range will be used. If the application has previously obtained an address range and has access to persistent storage, the application may record the previous address range and attempt to reuse the saved address range.

[b] Only received MASAP_PROBE, MASAP_DEFEND, and MAAP_ANNOUNCE PDUs that conflict with the address range associated with this state machine generate rProbe!, rDefend!, and rAnnounce! events. All MASAP_PROBE, MASAP_DEFEND, and MASAP_ANNOUNCE PDUs that do not conflict with the address range associated with this state machine are ignored.

[c] After a Release! event is received and the state machine has returned to INITIAL state, the address range associated with this state machine is considered to be free, and the state machine can be destroyed.

[d] If the compare_MAC function returns TRUE then no further processing or protocol action is taken and the protocol state does not change.

Like MAAP, MASAP may operate in a P2P basis, where the stations agree among themselves the status of a certain MAC Address Range. However, if a Proxy is available in the network, MASAP brings the opportunity to reduce the Probing time with the Proxy answering the MASAP_PROBE messages with a MASAP_PROXY_ANSWER message. The use of the Proxy also allows the protocol to work on technologies where non associated clients will not listen to ANNOUNCE or DEFEND messages. In case a MASAP_PROBE message is answered by a MASAP_PROXY_ANSWER, the station goes into the DEFEND state but it omits the sending of MASAP_ANNOUNCE messages, since the Proxy will take care of book-keeping the status of the addresses.

3.1.3. Specific IEEE 802.11 Operation in MASAP

IEEE 802.11 has a set of different characteristics, such as the impossibility by the stations associated to an AP of listening to frames sent in pre-associated state. This will be the case of the MASAP_PROBE messages, which may be transported through ANQP to the AP. In this case, the MASAP protocol is runs with a Proxy able to answer MASAP_PROBE messages. This operational mode also omits the use of MASAP_ANNOUNCE messages, since probing nodes which are not associated, will not receive these messages, rendering them useless. It is possible also, that current behavior achieved with the MASAP_PROXY_ANSWER message (indication by AP of the successful allocation) can also be achieved through the use of an IEEE 802.11 Reassociation frame, containing a code such as DENIED_MAC_ADDRESS_POLICY_VIOLATION.

3.2. MASBAP Protocol Operation

MASBAP is used for assign unicast and multicast addresses following IEEE 802c SLAP definition with clients discovering and requested addresses from a MASBAP server(s) or proxy in the network.

3.2.1. Message Addressing

MASBAP makes use of the following rules for addressing:

- Source MAC address for MASBAP_DISCOVER messages will be chosen randomly from the range shown in Table 3.
- Source MAC address for MASBAP_REQUEST messages will use the MAC Address previously assigned or the EUI-64/48 assigned to the station.
- Destination MAC address for MASBAP_DISCOVER messages corresponds to the multicast address specified in Table 3.
- Destination MAC address for MASBAP_OFFER and MASBAP_ACK messages correspond to the source MAC address of the MASBAP_DISCOVER message.

Table 3: Address Allocation

Address Range	**Function**
TBD	Range of addresses to randomly select a source address for MASBAP_DISCOVER messages
TBD	Multicast address used as destination in MASBAP_DISCOVER messages

3.2.2. State Machine

The MASBAP protocol is defined through the following client state machine.

The client state machine make use of the following events (events are denoted with a ! at the end):

- Begin!: The state machine is initialized or reinitialized.
- Release!: A Release! event signals that the address range associated with this instance of the state machine is no longer in use.
- Restart!: A Restart! event signals that an error has been detected and that the state machine will be restarted.
- RequestAddress!: A RequestAddress! event signals the starting of the MASBAP process.
- rOffer!: A rOffer! event signals that a MASBAP_OFFER message has been received.
- rACK!: A rACK!! event signals that a MASBAP_ACK message has been received.
- eTIMER_expire!: A eTIMER_expire! event signals that the timer TIMER has expired.
- MESSAGE_count!: A MESSAGE_count! event signals that the number of messages MESSAGE sent has reached its maximum.
- PortOperational!: A PortOperational! event signals that the port has entered an operational state.

The client state machine makes use of the following actions:

- Select_address: select address decides if a previous known address for this network is known and if it must be included in the MASBAP_DISCOVER.
- Start_TIMER_timer: starts the timer for this specific lease
- Stop_TIMER_timer: stops the timer for this specific lease
- Reset_MESSAGE_count: set to 0 the MESSAGE count counter
- Increment_MESSAGE_count: increment by 1 the MESSAGE count counter
- sDISCOVER: Send a MASBAP_DISCOVER message
- sREQUEST: Send a MASBAP_REQUEST message
- Validate_requirements: This action checks that a partial fulfillment of the requested parameters of the MASBAP_DISCOVER message is enough for the operation of the client. It returns 1 if the state machine can continue or 0 if it does not.
- Select_offer: In case multiple MASBAP_OFFER messages are received; this action selects one of them for continuing the process. The mechanism to choose one might be related with the level of compliance with the requested parameters sent in the MASBAP_DISCOVER.

Moreover, the client state machine is based on 4 states in the following table:

Table 4: MASBAP client states

State	**Description**
Initial	Start of the state machine
Discover	Discover sent, waiting for response
Request	Offer received, Request sent, waiting for response
Bound	Address selected, ACK received

The following table presents the state machine of the MASBAP client:

Table 5: MASBAP Client State Machine

		STATE			
		INITIAL	**DISCOVER**	**REQUEST**	**BOUND**
EVENT	**Begin!**	Select_address RequestAddress!	-X-	-X-	-X-
	Release!	-X-	Stop_OfferRcv_timer INITIAL	Stop_ACKRcv_timer INITIAL	Stop_LifeTime _timer INITIAL
	Restart!	Select_address RequestAddress!	Stop_OfferRcv_timer INITIAL	Stop_ACKRcv_timer INITIAL	Stop_LifeTime _timer INITIAL
	RequestAddr ess!	Reset_DISCOVER count	-X-	-X-	-X-

	STATE			
	INITIAL	**DISCOVER**	**REQUEST**	**BOUND**
	Start_OfferRcv_timer sDISCOVER PROBE			
rOffer!	-X-	Select_Offer Validate_requirements[a] Stop_OfferRcv_timer sREQUEST Reset_REQUEST_count Start_ACKRcv_timer REQUEST	-X-	-X-
rACK! (status=3)	-X-	Stop_ACKRcv_timer INITIAL	Stop_ACKRcv_timer INITIAL	-X-
rACK! (status=4)	-X-	-X-	Stop_ACKRcv_timer Start_Lifetime_timer BOUND	-X-
rACK! (status=5-7)	-X-	Stop_ACKRcv_timer INITIAL	Stop_ACKRcv_timer REQUEST/rOffer!	-X-
rACK! (status=9)	-X-	Stop_ACKRcv_timer INITIAL	Stop_ACKRcv_timer INITIAL	-X-
rACK! (status=11)	-X-	Stop_ACKRcv_timer INITIAL	Stop_ACKRcv_timer REQUEST/rOffer!	-X-
eOfferRcv_expire!	-X-	Increment_DISCOVER_count sDISCOVER start_OfferRcv_timer	-X-	-X-
eACKRcv_expire!	-X-	-X-	Increment_REQUEST_count sREQUEST start_ACKRcv_timer	-X-
eLifeTime_expire![b]	-X-	-X-	-X-	INIT/Restart!
DISCOVER_count!	-X-	Stop_OfferRcv_timer INIT/Restart!	-X-	-X-
REQUEST_count!	-X-	-X-	Stop_ACKRcv_timer INIT/Restart!	-X-
PortOperational!	Select_address INITIAL/Request_Address!	INITIAL/Restart!	INITIAL/Restart!	

[a] In case Validate_requirements return 0, the OFFER_timer is not stopped and the client will keep waiting for other MASBAP_OFFER messages arriving.

[b] This event marks the expiration of the Lifetime timer, a client before the lifetime expires, might send a MASBAP_REQUEST message to the server containing the Station ID and MAC address range that wants to rebind.

Regarding the MASBAP server, its operation is completely stateless. All information provided to the client must be treated as informational (nothing allocated) until the MASBAP_ACK is sent. Only after transmission of the MASBAP_ACK, the server will block the address or range of addresses allocated to the specific station.

3.2.3. Security aspects of MASBAP

One of the weakness of these kind of protocols may be an address exhaustion attack, where an attacker tries to block all possible addresses available at the server. In the general case, with the current definition of the Count fields in the options, it will require of 2^34 interactions with the server.

In case a range of addresses available at the server does not cover all 48 bits of the address space, it is advisory to reduce the maximum number of addresses that can be requested in a single iteration with the server. In the case the client requests for more addresses than the allowed, a MASBAP_ACK message with status 11 (Parameter Problem) must be answered to the MASBAP_DISCOVER or MASBAP_REQUEST message.

4. Protocol Details

4.1. Message format

The EtherType of all MUMAAP frames shall be the EtherType given in Table 6:

Table 6: MUMAAP Ethertype

MUMAAP EtherType
TBD

The message format of the protocol uses a common control header, which will be included on all messages of the protocol. The control header is shown in Table 7:

Table 7: MUMAAP Base header

0 7	8 10	11 15	16 31	
subtype	ver	message_type	control_word	
Cookie			Status	length

The different fields of the control header are specified as follows:

Subtype (8 bits): The 1-octet **subtype** field is used to identify the format being carried by MUMAAP and distinguish between the MASAP and the MASBAP protocols, as defined in Table 8

Table 8: MUMAAP Subtype

MUMAAP Subtype	
MASAP	TBD
MASBAP	TBD

Version (3 bits): Three bits indicating the version of the protocol. As per this specification we will use all bits set to 0 denoting the first version of the protocol.

message_type (5 bits): The message_type field contains one of the defined MUMAAP message types as defined in Table 9. If a MUMAAP message is received with a reserved message_type, the MUMAAP frame shall be ignored.

Table 9: Message Type

Value	Function	Description
0	---	Reserved
1	MASAP_PROBE	Probe MAC address(es)
2	MASAP_DEFEND	Defend MAC address(es)
3	MASAP_ANNOUNCE	Announce MAC address(es)
4	MASAP_PROXY_ANSWER	Answer from proxy regarding Probe messages
5	MASBAP_DISCOVER	Request for a MAC address to a Server
6	MASBAP_OFFER	MAC allocation offer from the server
7	MASBAP_REQUEST	Confirmation of the addresses to be allocated

8	MASBAP_ACK	Confirmation of allocation from server to station or error reporting
8-1024	--	Reserved

control_word (16 bits): The control word contains the following flags:

Table 10: Control Word behavior

Bit	Name	Description
0	AAI	Bit set to 1: Address in the AAI space requested/provided
1	ELI	Bit set to 1: Address in the ELI space requested/provided
2	SAI	Bit set to 1: Address in the SAI space requested/provided
3	Reserved	Reserved for future use
4	64/48 bits	Bit set to 1: 64 bits address requested/provided Bit set to 0: 48 bits address requested/provided
5	Multicast/Unicast	Bit set to 1: Multicast address requested/provided Bit set to 0: Unicast address requested/provided
6	Infrastructure/Station	Bit set to 1: Message source is Server/Proxy Bit set to 0: Message source is an end-node
7	MAC Provided	Bit set to 1: MAC address is provided Bit set to 0: MAC address is not provided This bit is used by a station providing an already used MAC address as hint to a Server.
8	Station ID provided	Bit set to 1: Station ID is provided Bit set to 0: Station ID is not provided
9	Network ID provided	Bit set to 1: Network ID is provided Bit set to 0: Network ID is not provided
10	Code field provided	Bit set to 1: The message contains a code field Bit set to 0: The message does not contain a code field
8	Specific address type	Bit set to 1: Specific address type information is provided Bit set to 0: Specific address type information is not provided
12-15	Reserved	Reserved for future use

cookie (16 bits): The cookie field must be incremented from a local counter, for every new transaction in MUMAAP variants. Frames that are originated as consequence of receiving another frame, must copy the cookie of the originating frame, since they belong to the same dialog.

status (4 bits): Status code indicating the result of an action. The status can be chosen from Table 11
Table 11: Status Value

Value	Description
0	Field not used
1	MAC Range not in use
2	MAC Range in use
3	Re-generate addresses in the given prefix and use MASAP
4	ACK – Assignment accepted
5	Failure – Assignment cannot be completed
6	Failure – Requested quadrant not available
7	Failure – Requested range not available
8	Offer provided
9	Mandatory use of MASAP
10	Mandatory use of MASBAP
11	Parameter problem
12	Offer Provided - Partial fulfillment
13-15	Reserved

length (12 bits): Length of the message in octets.

4.2. MASAP message definition

MASAP messages are composed of different options which can be transported by the protocol depending on the actual message being carried. First, we define the options that can be transported and later for each message we will indicate which options can be added to the message and the values that need to carry.

4.2.1. Options:

For all options, type field is shown in Table 12. Length expresses the length of the option in Octets.

Table 12: Options code values

Type ID	Description
0	Station ID
1	48 bits MAC Address (Range)
2	64 bits MAC Address (Range)
3	Network ID
4	Specific MAC Range
5	48 bits MAC Range in Conflict
6	64 bits MAC Range in Conflict
7	MAC Address Count
8	Lifetime

Station ID:
This option provides up to 255 bytes to include the identifier of the Station.

Type	Length	Station Identifier
1 Octet	1 Octet	Up to 255 Octets

In all cases, a station receiving a message, different from MASAP_PROBE, MASAP_ANNOUNCE or MASAP_DEFEND with a Station ID that does not correspond to itself, the packet must be drop and stop processing.

48 bits MAC Address (Range):
This option provides the mechanism to transport a MAC address of 48 bits plus a 16 bit number indicating the amount of addresses requested. If a single address, not a range, wants to be specified, Count must be set to 1. When Count is higher than one then the count indicates that multiple MAC addresses are assigned or requested, starting with the MAC Address and including the next sequential addresses up to the count.

Type	Length	MAC Address	Count
1 Octet	1 Octet	48 bits	16 bits

64 bits MAC Address (Range):
This option provides the mechanism to transport a MAC address of 64 bits plus a 16 bit number indicating the amount of addresses requested. If a single address, not a range, wants to be specified, Count must be set to 1.

Type	Length	MAC Address	Count
1 Octet	1 Octet	64 bits	16 bits

Network ID:
This option provides up to 255 bytes to include the identifier of the network.

Type	Length	Network Identifier
1 Octet	1 Octet	Up to 255 Octets

Specific MAC Range:

This option is included in order the Server to request the station to perform self-claiming in a specific address space.

Type	Length	Length of MAC Address Prefix Subfield (bytes)	Prefix Trim subfield (bits)	Reserved	MAC Address Prefix (Bytes)
1 Octet	1 Octet	3 bits	3 bits	2 bits	0 – 8 Octets

The Length of MAC Address Prefix Bytes subfield is a subfield of 3 bits. When the Length of MAC Address Prefix subfield is set to one of the values of 1–6, that value indicates the length (in octets) of the MAC Address Prefix Bytes field. The Length of MAC Address Prefix Bytes subfield is not set to 0 or 7; those values are reserved.

The Prefix Trim subfield is a subfield of 3 bits and takes one of the values of 0–7, that value indicating number of bits to be truncated from the end of the MAC Address Prefix subfield in order to obtain the MAC Address Prefix. In other words, the MAC Address Prefix is represented as the value of the MAC Address Prefix Bytes field after truncation of some of the most significant bits of the last octet, with the number of truncated bits equal to the value of the Prefix Trim subfield.

The MAC Address Prefix Bytes field is a field of 1 to 8 octets (with the length signaled in the Length of MAC Address Prefix subfield of the Policy Flags field) containing the full bytes (prior to truncation per the Prefix Trim subfield) of the MAC Address Prefix relevant to address self-assignment.

48 bits MAC Range in Conflict
This option provides the mechanism to transport a MAC address of 48 bits in conflict plus a 16 bit number indicating the amount of addresses that are in conflict. If a single address, not a range, wants to be specified, Count must be set to 1.

Type	Length	MAC Address	Count
1 Octet	1 Octet	48 bits	16 bits

64 bits MAC Range in Conflict:
This option provides the mechanism to transport a MAC address of 64 bits in conflict plus a 16 bit number indicating the amount of addresses that are in conflict. If a single address, not a range, wants to be specified, Count must be set to 1.

Type	Length	MAC Address	Count
1 Octet	1 Octet	64 bits	16 bits

MAC Address Count
This option provides a mechanism for the station to request a number of addresses without providing a MAC address range.

Type	Length	MAC Address Count
1 Octet	1 Octet	1-2 Octets

In case the MAC Address Count option is not present in a MASBAP_REQUEST, it is assumed that the request is for a single MAC Address.

Lifetime
This option enables the server to set a lifetime for the specific MAC Address leasing

Type	Length	Lifetime
1 Octet	1 Octet	2 Octets

4.2.2. MASAP Protocol Messages

The self-claiming protocol will use the following definition of messages:

MASAP_PROBE

This message is used to probe for a free MAC address range.
The MASAP_PROBE message includes the following options:

0 7	8 10	11 15	16 31
subtype	ver	message_type	control_word
Cookie			Status / length
(Optional) Station ID			
48 bits MAC Address (Range) OR 64 bits MAC Address (Range)			

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being self-claimed belongs to.
- Bit 4 indicates if the address being self-claimed is 64 or 48 bits.
- Bit 5 indicates if the address is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to1, indicating the message carries MAC addresses

status field must be set to 0 in this message.
Station ID optionally contains the ID of the station sending this message.

48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is the first address of a consecutive range of addresses being requested. The **count** field is the number of addresses being requested. If only a single address is being requested, this field is set to one (1).

MASAP_DEFEND

This message is used to defend an already acquired MAC address range.

0 7	8 10	11 15	16 31
subtype	ver	message_type	control_word
Cookie			Status / length
(Optional) Station ID			
48 bits MAC Address (Range) OR 64 bits MAC Address (Range)			
48 bits MAC Range in Conflict OR 64 bits MAC Range in Conflict			

control_word bits must be turn 1 or 0 copying the values from the MASAP_PROBE message that triggered this message:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being self-claimed belongs to.
- Bit 4 indicates if the address being self-claimed is 64 or 48 bits.
- Bit 5 indicates if the address is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to1, indicating the message carries MAC addresses

status field must be set to 0 in this message.
Station ID is copied from the originating MASAP_PROBE message. If the MASAP_PROBE message originating this message contains a Station ID, this option must copy the one in the MASAP_PROBE message.
The 48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is copied from the originating MASAP_PROBE message.

The **48 bits MAC Range in Conflict OR 64 bits MAC Range in Conflict** is set to the first address that conflicts with a requested address range from a MASAP_PROBE. Count in this case is set to the number of addresses in conflict from the range.

MASAP_ANNOUNCE

This message is used to announce an already allocated MAC address range.

0 7	8 10	11 15	16 31
subtype	ver	message_type	control_word

Cookie	Status	length
(Optional) Station ID		
48 bits MAC Address (Range) OR 64 bits MAC Address (Range)		

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being self-claimed belongs to.
- Bit 4 indicates if the address being self-claimed is 64 or 48 bits.
- Bit 5 indicates if the address is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to1, indicating the message carries MAC addresses

status field must be set to 0 in this message.
Station ID optionally contains the ID of the station sending this message.

48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is the first address of a consecutive range of addresses being allocated to the station. The **count** field is the number of addresses that have been allocated, starting with the one provided in the MAC Address field. If only a single address is being requested, this field is set to one (1).

MASAP_PROXY_ANSWER

This message is used by a Server/Proxy to shorten the time required to Probe a MAC Address Range. The idea behind this message is that in the case of a Proxy book-keeping the addresses used in the network, it can quickly state the status of the addresses being probed.
This message can also be used to request the station to repeat the probe in a different MAC address range.

0 7	8 10	11 15	16 31
subtype	ver	message type	control word
Cookie			Status / Length
(Optional) Network ID			
(Optional) Specific MAC Range			
(Optional) 48 bits MAC Address (Range) OR 64 bits MAC Address (Range)			
(Optional) 48 bits MAC Range in Conflict OR 64 bits MAC Range in Conflict			

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being self-claimed belongs to.
- Bit 4 indicates if the address being self-claimed is 64 or 48 bits.
- Bit 5 indicates if the address is unicast or multicast
- Bit 6 will be set to 1, indicating this message is originated at a server.
- Bit 7 must be set to 1 in case the Specific MAC Range is provided, indicating the message carries MAC addresses range

status field must be set to either 1, 2, 3 or 12 in this message. The meaning of each code as follows:

- 1: Indicates the MAC Range is not in use and the address can be directly allocated to the station.
- 2: Indicates the MAC Range is in use and the station must choose a different range inmediately. In case status takes the value 2, the **48 bits MAC Address (Range) OR 64 bits MAC Address (Range)** and the **48 bits MAC Range in Conflict OR 64 bits MAC Range in Conflict** must be present in the message, with the same meaning as in the MASAP_DEFEND message.

- 3: Indicates the station should generate a new address or range of addresses in the range provided in the Specific MAC Range option and try self-claiming again. In case **status** takes the value 3, then the option **Specific MAC Range** must be provided.
- 10: MASAP is not supported in the network and MASBAP must be used.

Network ID optionally provides an indication of the network this server is serving.

4.3. MASBAP Protocol Messages

The server based protocol defines the following messages:

MASBAP_DISCOVER

This message is used to start the dialog with any Proxy/Server in the network.

0 7	8 10	11 15	16 31
subtype	ver	message_type	control_word
Cookie			Status / length
(Optional) Station ID			
(Optional) Specific MAC Range			
(Optional) 48 bits MAC Address (Range) OR 64 bits MAC Address (Range)			
(Optional) MAC Address Count			

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being requested.
- Bit 4 indicates if the address requested is 64 or 48 bits.
- Bit 5 indicates if the address requested is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to 1 or 0, depending if the message carries the Specific MAC Range or 48/64 MAC Address (Ranges).

status field must be set to 0 in this message.
Station ID optionally contains the ID of the station sending this message. Station ID is mandatory in case the message contains a **48 bits MAC Address (Range) OR 64 bits MAC Address (Range).**
Specific MAC Range is used to provide to the server a range for the MAC address being requested. The server will try to assign a MAC address belonging to the range provided in this option.
48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is used to provide a MAC Address or a range of MAC addresses that have been previously assigned to the station. The server will try to assign the same MAC Address or range to the station. If only a single address is being requested, this field is set to one (1).
MAC Address Count is used to request for a range of addresses without defining the range where the MAC Addresses should belong to. This option should not be added if a **48 bits MAC Address (Range) OR 64 bits MAC Address (Range)** is provided. In case neither options are provided, the server will assume a single MAC address is requested.

MASBAP_OFFER

Upon reception of a MASBAP_DISCOVER message, the server will send an offer for a possible MAC Address allocation to the client. Note that this allocation will consider the different hints provided by the station.

0 7	8 10	11 15	16 31
subtype	ver	message_type	control_word
Cookie			Status / length
(Optional) Network ID			
(Optional) Station ID			
Lifetime			
48 bits MAC Address (Range) OR 64 bits MAC Address (Range)			

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being allocated.
- Bit 4 indicates if the address allocated is 64 or 48 bits.
- Bit 5 indicates if the address allocated is unicast or multicast
- Bit 6 will be set to 1, indicating this message is originated at a server.
- Bit 7 must be set to 1 indicating the message carries a 48/64 MAC Address (Ranges).

status field must be set to 8 indicating an ongoing process or to 12 indicating a partial fulfillment of the constraints sent in the MASBAP_DISCOVER.
Network ID optionally contains the ID of the network the proxy sending this message belongs to.
Station ID optionally copies the Station ID from the MASBAP_DISCOVER message.
Lifetime indicates the lifetime of the leasing. A Lifetime option must be provided.
48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is used to provide a MAC Address or a range of MAC addresses to the station. This option must be provided in this message.

MASBAP_REQUEST

Upon receiving a MASBAP_OFFER message, the station confirms the allocation of the address/es to the server through this message. The server will not consider the addresses allocated before this message is received.

0 7	8 10	11 15	16 31	
subtype	ver	message_type	control_word	
Cookie			Status	length
(Optional) Network ID				
(Optional) Station ID				
Lifetime				
48 bits MAC Address (Range) OR 64 bits MAC Address (Range)				

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being allocated.
- Bit 4 indicates if the address allocated is 64 or 48 bits.
- Bit 5 indicates if the address allocated is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to 1 indicating the message carries a 48/64 MAC Address (Ranges).

status field must be set to 0.
Network ID optionally copies the Network ID in the MASBAP_OFFER message.
Station ID optionally includes the Station ID.
Lifetime indicates the lifetime of the leasing copied from the MASBAP_OFFER message.
48 bits MAC Address (Range) OR 64 bits MAC Address (Range) is copied from the MASBAP_OFFER message.

MASBAP_ACK

This message is used for the station to acknowledge the assignment of the address. Typically is sent as answer to a MASBAP_REQUEST to close the assignment process, although it is also used by the server to notify the station of any error, answering with it to a MASBAP_DISCOVER message.

0 7	8 10	11 15	16 31	
subtype	ver	message_type	control_word	
Cookie			Status	length
(Optional) Network ID				
(Optional) Station ID				
(Optional) Specific MAC Range				

control_word bits must be turn 1 or 0 according to the following list:

- Bits 0 to 2 must be turn to 1 depending on the quadrant the address being allocated.
- Bit 4 indicates if the address allocated is 64 or 48 bits.
- Bit 5 indicates if the address allocated is unicast or multicast
- Bit 6 will be set to 0, indicating this message is originated at a station.
- Bit 7 must be set to 1, indicating the message carries a 48/64 MAC Address (Ranges).

status field can take multiple values:

- 3: Indicates MASBAP is not available, and the station should generate a MAC address or a range in the range defined in the Specific MAC Range option and use MASAP. In case **status** takes the value 3, then the option **Specific MAC Range** must be provided.
- 4: Allocation completed succesfully
- 5 to 7: Failure while assigning the address considering the hints provided by the station.
- 9: MASBAP is not supported, use MASAP
- 11: There is a mismatch of parameters between the packets sent by the server and the ones sent by the station.

Network ID optionally contains the ID of the network being serviced by the Proxy.
Station ID optionally contains the ID of the station. In case Station ID was provided in the MASBAP_REQUEST message, this option must be included.
Specific MAC Range includes a range to be used for MASAP. This option must be present in status code is 3.

MASBAP_ACK can be used as reply of MASBAP_DISCOVER and MASBAP_REQUEST. A station should not consider an allocation as completed until a MASBAP_ACK message is received.