Project Number:	**IST-2001-32603**
Project Title:	**6NET**
CEC Deliverable Number:	**32603/RENATER/DS/3.4.3/A1**
Contractual Date of Delivery to the CEC:	**March 31st, 2003**
Actual Date of Delivery to the CEC:	**April, 2003**
Title of Deliverable:	**IPv6 multicast address allocation study**
Work package contributing to Deliverable:	**WP3**
Type of Deliverable*:	**R**
Deliverable Security Class**:	**PU**
Editors:	**RENATER**
Contributors:	Ralph Droms (CISCO), Jerome Durand (RENATER), Piers O'Hanlon (UCL), Jean-Jacques Pansiot (ULP), Bernard Tuy (RENATER), Stig Venaas (UNINETT)

* Type: P - Prototype, R - Report, D - Demonstrator, O - Other

** Security Class: PU- Public, PP – Restricted to other programme participants (including the Commission), RE – Restricted to a group defined by the consortium (including the Commission), CO – Confidential, only for members of the consortium (including the Commission)

Abstract:
This Deliverable examines the issue of address allocation for multicast. The main need for such an allocation mechanism is collision avoidance. Two kinds of collision are considered:

- address collision, and
- address and port collision.

This document also details the announcement mechanisms that can be used to let people know the existence of IPv6 multicast sessions (since the session announcement problem is closely related to the address allocation issue).
The document gives addressing architecture details for IPv6 multicast, identifies the many documents (RFCs and Internet-drafts) already in existence on the IPv6 address allocation issue, and describes the existing approaches to the session advertisement problem.
Current implementations for IPv6 multicast address allocation and session advertisement are given.
Finally, deployable solutions are investigated, some directly provided by the state of art, whilst others are provided by the partners involved in this area.

Keywords:
IPv6, Multicast, Address allocation

1. Introduction

There are many applications using multicast technology. These applications require globally unique group addresses, which may be permanent or transient, to exchange data amongst their peers.

Whilst systems such as SDR or web based mechanisms provide mechanisms for domain specific allocation they cannot offer a globally reliable Internet wide service.

A generic, simple and scalable mechanism is required, providing globally unique multicast addresses for user groups within specified time periods. Moreover the notion of “scope” in the IPv6 multicast address introduces an additional requirement: to be able to allocate a globally unique multicast address within a certain scope.

Such mechanisms do not exist today for IPv4, let alone for IPv6. It is a good opportunity to start with the IPv6 multicast applications to try and provide such a mechanism / protocol. This document gives a complete taxonomy of IPv6 multicast addresses allocation problem. Many documents are written on this subject and there is a real need for a reference in this domain.

The main need today for an allocation mechanism is collision avoidance. In the SSM model, as the channel is defined by a group address and a source, a collision could only occur only if a host would like to be a sender for 2 different groups having the same address. Therefore, it is sufficient for a sender to choose different addresses for different channels, this is a purely local decision. This deliverable will only consider the ASM model. In the ASM model, we can consider two kinds of collision. There can be an address collision, and addresses and ports collision. In the former case, the receivers would get both flows but would discard the bad one. The risk is a loss of bandwidth in the receiver sites. Note that with IGMPv3/MLDv2, a host can exclude unwanted sources. In this case the loss of bandwidth will be beyond the first hop router. In the case of address and port collision, there would be a loss of resources in receivers, applications receiving both flows. We can also differentiate collisions due from errors and collisions due to malicious users, wanting to disturb multicast sessions.

The session announcement problem is closely related to the address allocation issue. Some implementations today combine IPv6 multicast address allocation and session announcement. This is why this document also details the mechanisms that can be used to let people know the existence of IPv6 multicast sessions.

First of all this document gives the addressing architecture details for IPv6 multicast, describing the different types of addresses defined. Then it focuses on the many documents (RFCs and Internet-drafts) already in existence on the IPv6 address allocation problem. Following this is a section covering the existing approaches to the session advertisement problem. The next section details the current implementations for IPv6 multicast addresses allocation and session advertisement. Finally, given the aforementioned analyses, deployable solutions are investigated, some directly provided by the state of art, whilst others are provided by the partners involved in this area.

2. Table of contents

3. Multicast addressing architecture

The complete multicast architecture defined today in different RFCs and Internet Drafts is described in this section. It is very important to know all the different address types defined because the allocation mechanisms will depend of the architecture used.

3.1 Generalities

RFC 3513 (IP Version 6 Addressing Architecture) defines the IPv6 multicast address:

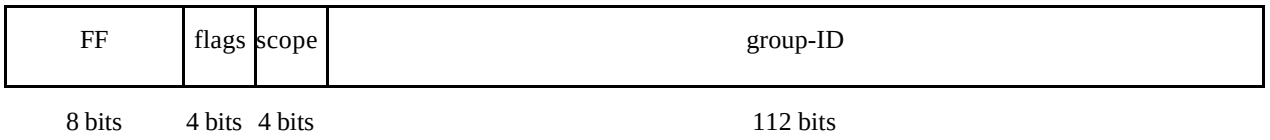

Figure 1: IPv6 multicast address structure

IPv6 multicast addresses are derived from FF00::/8 prefix. The flags field is a set of 4 flag bits.

Figure 2: The flags field

Only bit T is described in RFC 3513. Bits P and R are described in [RFC3306] and [EMBEDDEDRP]. The high order flag bit is not yet used. The use of the flags makes it possible to distinguish different address type that will be detailed in the following sections.

The scope field is described in a section below.

3.2 Well known / static addresses

RFC 3513 (IP Version 6 Addressing Architecture) includes the definition of the multicast addresses. Multicast addresses with bit T of the flag field set to 0 correspond to permanent multicast addresses, assigned by IANA (Internet Assigned Number Authority)

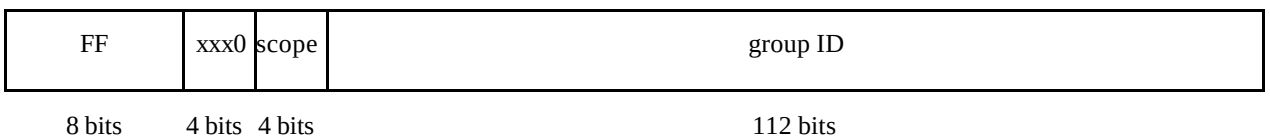

Figure 3: Permanent IPv6 multicast address structure

When IPv6 multicast is widely deployed, we can imagine that some organisms will have some permanent broadcasts. Some TV channels or radio channels could be given the right to be allocated a permanent IPv6 multicast address derived from FF00::/12 multicast prefix.

RFC 2375 defines the IPv6 multicast addresses already allocated. There are different kinds of "permanent" addresses. Some correspond to "low level" services (such as NTP, DHCP, cisco-rp-announce, SAP(announce, …). The second kind corresponds to permanent "commercial" services (such as TV channels if they don't use SSM). RFC 3307 defines the allocation guidelines for the permanent addresses. It is described later in this document.

3.3 Transient addresses

Transient addresses are IPv6 multicast addresses with bit T of the flag field set to 1.

3.3.1 General transient addresses

General transient addresses are addresses with all flags set to 0 but the bit T set to 1. It seems that there is no real recommendation for the use of these addresses at that time. Those addresses can be used for one shot sessions or test sessions.

3.3.2 Unicast prefix-based address

RFC 3306 defines a way to derive an IPv6 multicast address from an IPv6 unicast prefix.

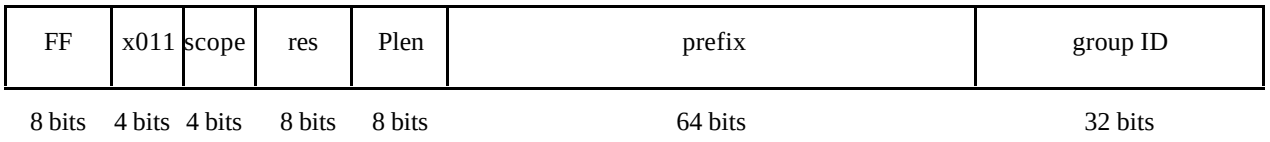

Figure 4: unicast prefix based IPv6 multicast address structure

If someone needs an IPv6 multicast address, he/she can derive it from its unicast prefix. There are potentially 2^{32} addresses per /64 prefix (per link). The question remains about the way to choose the last 32 bits of the prefix-based address. Nevertheless, the allocation problem is now on a smaller zone that is known and managed.

3.3.3 Embedded RP addresses

The Internet Draft draft-savola-mboned-mcast-rpaddr-02 [EMBEDDEDRP] defines a way to embed the RP (Rendezvous Point) address in the IPv6 multicast address. The following scheme shows the structure of such an address:

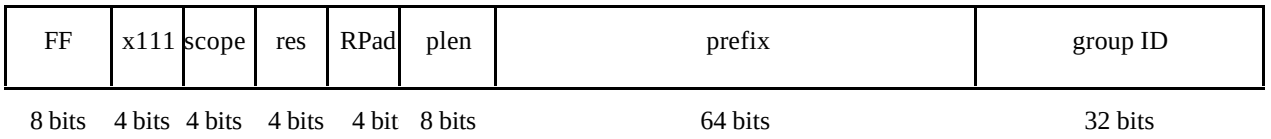

Figure 5: embedded RP IPv6 multicast address structure

The allocation problem is reduced with embedded RP addresses because there can be a collision only between addresses derived from the same RP. An entity that is well defined and managed.

3.3.4 SSM Addresses

SSM addresses are unicast prefix based address with prefix length field and prefix field set to 0. Therefore, SSM multicast addresses are derived from FF3x::/96 prefix.

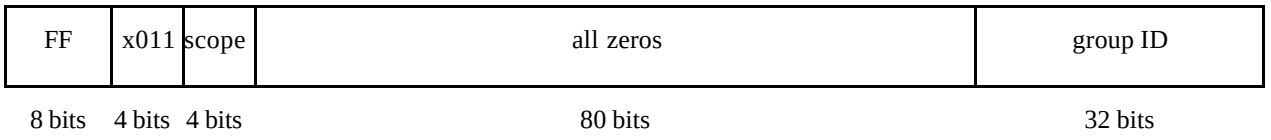

Figure 6: SSM IPv6 multicast address structure

As for SSM, there is no collision in IPv6 since the SSM range is well defined, and channels (S1,G) and (S2,G) are different (note that in IPv4, the SSM range is not entirely fixed, and a "Multicast Router Discovery SSM Range Option" has been defined, so an SSM address could possibly, by misconfiguration, collide with an ASM address). Therefore allocation of IPv6 SSM addresses is a purely local (host) problem. Obviously, there is a need for a per host mechanism (in the OS), to allocate automatically an SSM address (more or less similar to allocate TCP or UDP port numbers on the caller side). There is one constraint due to the mapping to multicast ethernet addresses: the allocation should try to have an equal probability to select the last 32 bits (see section 4.1 for more details). Random selection should be enough, since a collision at the ethernet multicast address level is not very serious. This allocation procedure could have a similar API to the RFC 2771 (see section 4.8). The OS has to remember which SSM addresses have been allocated, and probably for how long. This is because SSM may have to be advertised may be for a long time, and the application (source) (and the OS) may reboot many times during the life of the SSM channel.

3.4 Scoped multicast addresses

As described above, it is possible to specify a scope for every IPv6 multicast address. The following values are defined for the scope field:

1 node-local
2 link-local
4 admin-local
5 site-local
8 organisation-local
E global

The scope field makes it possible to controle the scope of the desired broadcast very easily. This was done in IPv4 with the TTL value. The scope field has a major impact on the allocation process as it has to be specified for every allocation. Using scoped addresses makes it easier to control address allocations than with global addresses.

Therefore the allocation procedures must take the scope into consideration.

3.5 GLOP

GLOP is defined for IPv4 in RFC 3180. GLOP makes it possible to embed the AS number in the IPv4 multicast address. Then it is possible in IPv4 to derive addresses from a specific AS (256 IPv4 multicast addresses per AS)

IPv6 would make it possible to have many more addresses per AS, given the length of the address. Nevertheless, there is no interest for GLOP when having the possibility to derive IPv6 multicast addresses from an IPv6 prefix (RFC 3306, see section below). Therefore we can consider that GLOP is useless in IPv6 world.

3.6 Summary

Prefix	Usage
FF0X::/16	Permanent IPv6 multicast addresses
FF1X::/16	General transient IPv6 multicast addresses
FF3X::/16	Unicast prefix based multicast addresses (transient)
FF3X::/96	SSM addresses (transient)
FF7X::/16	Embedded RP multicast addresses (transient)

Table 1 : Summary of IPv6 multicast ranges already defined (RFCs or I-D)

4. Multicast addresses allocation methods

This section gives a state of art for multicast addresses allocation methods. A number of documents are covered and our analysis has two main goals: to provide an overview of the documents and to recommend implementable and deployable solutions.

4.1 RFC 3307 - Allocation guidelines for IPv6 multicast addresses

RFC 3307 describes the guidelines for IPv6 multicast addresses allocation. The RFC 3307 is based on a **32 bits long group-ID**, as defined in RFC 2373, obseleted by RFC 3513. A problem is that RFC 3513 defines now a 112 bits long group-ID (See section 3.1)

The guidelines described in RFC3307 takes into consideration the mapping between IPv6 multicast addresses and link layer 2 addresses. An IPv6 multicast address is mapped into a MAC address following this rule: the last 32 bits of the IPv6 multicast address (defined as the group ID) are added to the MAC prefix 33-33.

For example, the address FF0E:30:2001:660:3001:4002:**AE45**:**2C56** will lead to the MAC address 33-33-**AE**-**45**-**2C**-**56**

One could criticise this consideration: the probability that two IPv6 multicast addresses on the same link are mapped in the same link-layer multicast address is very low and the consequences are minimal. Restricting the group-ID from 112 bits to 32 bits has a very important consequence on random based allocations as we will see later in this document.

This document also defines ranges of IPv6 multicast addresses and group-IDs that will be managed in a registry by IANA, or that will be reserved for dynamic address allocation.

4.1.1 Permanent IPv6 multicast addresses

These are addresses with flag bits T and P set to 0. They are allocated by IANA on an expert review basis. They must have **group-IDs** in the range 0x00000001 to 0x3FFFFFFF.

4.1.2 Permanent IPv6 multicast group identifiers

The aim for this is to be able to reach a given service in any network. These services are defined by group-IDs assigned by IANA on an expert review basis in the range 0x40000000 to 0x7FFFFFFF. These group-IDs should be used in prefix based IPv6 multicast addresses (RFC 3306). It is then possible to reach a service in a given network using the network prefix and the service suffix.

4.1.3 Dynamic IPv6 multicast addresses allocation

All dynamically allocated IPv6 multicast addresses, allocated by servers or end-nodes must have group-IDs in the range 0x80000000 to 0xFFFFFFFF according to the RFC. They must have T flag bit set to 1.

P=0 ; T=0	0x00000001 to 0x3FFFFFFF	Permanent multicast addresses
P=1 ; T=1	0x40000000 to 0x7FFFFFFF	Transient (T=1) addresses with permanent GIDs
P={0;1} ; T=1	0x80000000 to 0xFFFFFFFF	Transient dynamically allocated addresses

Table 2: Summary of Group-IDs ranges for IPv6 multicast addresses

4.2 Random choice

4.2.1 Introduction

From the wide address space available in IPv6, we can consider random method for IPv6 multicast addresses allocation. Moreover, the applications send and receive traffic with an address and a port number. If someone sends data to a group address that is already used, all the people belonging to this group will receive the new flow sent but the application will not take it into consideration because the ports are different (unless both flows are also sent to the same port but we can assume this is very rare)

The randomization algorithm is not discussed in this document.

4.2.2 Calculation

At this stage it is interesting to consider the probability of having a collision for a need of n IPv6 multicast addresses, when the number of bits to randomize is X

$$p(collision) = 1 - p(no\_collision)$$

$$= 1 - \frac{2^X!}{(2^X - n)!(2^X)^n}$$

To calculate this, we use Stirling approximation:

$$n! \approx \sqrt{2pn}\left(\frac{n}{e}\right)^n$$

4.2.3 Results

- If 32 bits available (X=32)

A classic temporary IPv6 multicast address is derived from the prefix FF1x::/16. RFC 3307 defines a 32 bits long group-ID.

The following table gives the probability of collision given the number of allocations (n) if we use only the last 32 bits.

Number of allocations	P(collision)
50	$<10^{-6}$
100	$8{,}90.10^{-6}$
500	$2{,}57.10^{-5}$
1 000	$1{,}21.10^{-4}$
2 000	$4{,}71.10^{-4}$
5 000	$2{,}88.10^{-3}$
10 000	$1{,}15.10^{-2}$
100 000	$6{,}18.10^{-1}$
1 000 000	1

Table 3 : Probability of collision with 32 bits

We see that if there are 5 000 groups created, then the probability that there is a collision is not acceptable.

- A random allocation could be used with /64 prefix based multicast addresses. The situation where more than 5000 multicast addresses need to be created on the link can be rare.
- This random allocation could also be used with embedded RP addresses if the RP is managing a few number of groups (less than 5000)
- Random allocations could be used for scoped addresses if the number of the groups needed in the zone remains low.
- This allocation cannot be used for general temporary addresses with global scope as there would not be the possibility to assign more than 5000 IPv6 multicast addresses in the Internet keeping a low probability of a collision.

- If 112 bits available (X=112)

The RFC 3513 defines a 112 bits long group-ID. This leads to more than 5.10^{33} possible group addresses, that can be considered as an infinite number.

In this case, we can have up to 10 billions addresses keeping a probability of collision that is almost null.

Nevertheless, a completely random multicast addresses will be difficult to manage, since it is impossible to tell who “owns” the address. It makes it difficult to enforce policies (for example to filter groups “owned” by some domain).

Moreover, using 112 bits does not match RFC3307 specifications. This is why we do not recommand this approach.

4.2.4 Conclusion for random allocations

Random allocation could be used by people having **no other mechanisms** to allocate addresses as it does not ensure unicity. To stay coherent with RFC 3307, and to ensure at least there will not be any collision with addresses allocate by other mechanisms such as MADCAP or DHCPv6, we suggest to reserve the group-ID range from 0xA0000000 to 0xAFFFFFFF for addresses randomly allocated.

Any host would specify if it wants to use random allocations. If the host does not specify an RP to use, then the host would derive an IPv6 multicast address from the unicast prefix of the link, the group-ID being randomly chosen in the range 0xA0000000 to 0xAFFFFFFF. If an RP is specified, then the host would build an IPv6 multicast address with the address of the RP embedded (see section 3.3.3), the group-ID would be chosen randomly in the range reserved for random allocations (we suggest range from 0xA000000 to 0xAFFFFFF)

4.3 SAP mechanism

RFC 2375 defines the reserved IPv6 multicast addresses. Some of them are used for SAP annoucements:

```
FF0X:0:0:0:0:0:2:7FFE    SAPv1 Announcements                        [SC3]
FF0X:0:0:0:0:0:2:7FFF    SAPv0 Announcements (deprecated)           [SC3]
FF0X:0:0:0:0:0:2:8000
    -FF0X:0:0:0:0:0:2:FFFF  SAP Dynamic Assignments                 [SC3]
```

SAP is defined by RFC 2974. IPv6 sessions are announced on the address `FF0X:0:0:0:0:0:2:7FFE` (where `X` is the scope desired for the annoucement). The addresses of the sessions can be chosen automatically in the range `FF0X:0:0:0:0:0:2:8000` - `FF0X:0:0:0:0:0:2:FFFF` by the session annoucement software (for example SDR). This leads to aproximately 500 000 possible addresses. The software chooses randomly an address that is not already advertised. This mechanism makes it possible to be given automatically an IPv6 multicast address that is not used but there are some issues:

- This solution might not scale if there is a need for a huge number of groups
- This solution is valid only for sessions advertised by SAP. Some people will like to have private groups not advertised. Therefore, this mechanism must not be used only for address allocation, but also for session advertisement.

This solution cannot be used to allocate global addresses. However this solution can be easily deployed for scoped addresses. In a domain, people could easily create local sessions and advertise it via SAP.

4.4 MADCAP

MADCAP stands for Multicast Address Dynamic Client Allocation Protocol. This is RFC 2730 from the Malloc IETF WG. It is a client server protocol similar to DHCP used to allocate to a client

one or more multicast addresses managed by a server. MADCAP supports several address families (including IPv4 and IPv6).

MADCAP clients may discover MADCAP servers either by configuration or by using a reserved multicast address. Addresses are allocated for a limited time (lease, similar to DHCP). A lease may be renewed. MADCAP manages both global and scoped multicast addresses, and a client may ask for configuration parameters, including a multicast scope list.

Note that MADCAP was initially part of a complete address allocation architecture including the MASC protocol (to allocate blocks of multicast addresses to domains, and AAP to allocate sub blocks to MADCAP servers).

4.4.1 Use with general temporary addresses

This option cannot be considered as it is too much complex, needing the complete allocation architecture described above. It failed for IPv4 and therefore cannot be a success for IPv6.

However, this solution would work for scoped addresses. If a MADCAP server is responsible for allocating addresses in a domain (for example a site), then it can allocate site-local addresses to any host in the site.

4.4.2 Use with unicast prefix derived addresses

With IPv6 and unicast prefix based multicast addresses, such a complex architecture is probably no more necessary, as servers could allocate addresses in the address range derived from their unicast prefix. This solution would be scalable, easy to implement and user-friendly. The only issue is that network administrators don't know MADCAP yet. This solution is fully conformant with RFC 3307, the MADCAP server must allocate prefix based multicast addresses with group-IDs in the range 0x8000000 to 0xFFFFFFFF.

4.4.3 Use with embedded RP addresses

Any MADCAP server could allocate RP based addresses, the only difference with the previous case is that there would be situations where there is more than one MADCAP server per /96 prefix.

4.5 DHCPv6

DHCPv6 [DHCPV6] was accepted as a Proposed Standard in December 2002, and is in the RFC Editor queue for publication since April 2003. Several DHCPv6 implementations have demonstrated interoperability at TAHI and Connectathon.

One of the primary differences between DHCPv4 [RFC2131] and DHCPv6 is that management of multiple addresses, which can be requested at any time by a client, is a fundamental part of the DHCPv6 architecture. Addresses are managed through Identity Associations (IA) in DHCPv6. To obtain addresses from a DHCPv6 server, a client creates an empty IA and sends it to the server. The server assigns addresses to the client and returns those addresses in the IA through a response message. An IA has an associated unique IA Identifier (IAID). Each address in the IA can have independent characteristics, such as a preferred lifetime and valid lifetime.

A client can, at any time, create a new IA, and send that to the server to ask for additional addresses. If the server policy allows the assignment of more addresses to the client, the server identifies those addresses and returns them to the client in the IA. There are three types of IAs defined for DHCPv6:

IA for non-temporary addresses (IA_NA), IA for temporary addresses [RFC3041] (IA_TA) and IA for prefix delegation (IA_PD). Each of these IAs have similar semantics and are treated similarly by clients and server.

DHCPv6 could be used for multicast address assignment through the definition and use of a new IA, the IA for multicast addresses (IA_MA). To obtain a multicast address, a client would create a new IA_MA and send it to the server. This message exchange can take place at any time and a client can send multiple IA_MAs to the server to obtain additional multicast addresses.

Multicast addresses would be managed in pools of available multicast addresses just as unicast addresses and prefixes are managed. The same kind of policy rule mechanisms would be used to control the assignment of multicast addresses. A multicast address would have a lifetime that would act as a lease on that address, so that a server could reassign multicast addresses to new clients over time.

DHCPv6 solution is almost the same as MADCAP. The only difference is that network administrators could be reluctant to deploy MADCAP as they don't know it as welle as DHCP. Moreover, it is not sure there will be many implementations of MADCAP if DHCPv6 can allocate multicast addresses. Anyway, the recommandations for the use of DHCPv6 are exactly the same as the ones for MADCAP described above.

4.6 ZMAAP - Zeroconf Multicast Address Allocation Protocol

This protocol is defined in [ZMAAP]. The mechanism is very simple: when a node needs a multicast address, it asks its local multicast address allocation server (called the mini-MAAS) an address, specifying the scope, the number of addresses needed and the desired lifetime. Then the mini-MAAS builds the IPv6 multicast addresses using a random function (at this stage the mini-MAAS uses addresses not assigned locally).

Then the mini-MAAS checks that the address is not reserved by any other mini-MAAS in the same scope domain. Therefore, the mini-MAAS sends an address claim message (ACLM), specifying the leases locally assigned. This message is send to the group of the mini-MAAS, using transport layer UDP. The scope of this mini-MAAS group address will be the same than the one needed for address allocation.

Every mini-MAAS must defend any allocation locally made. When a mini-MAAS receives an ACLM, it must check that the lease requested does not overlap with any local allocation. If there is an overlap, the mini-MAAS must reply with an address-in-use message (AIU). This message is also sent to the multicast group of mini-MAAS.

A mini-MAAS considers the lease assigned locally to be valid if it does not receive any AIU message within a certain delay.

ZMAAP allocation is not scalable. The problem is exactly the same as the use of SAP. It is not possible to imagine that all hosts exchange address usage on a given group. This solution could be used only for small scopes (admin-local or site-local). A group-ID range should be reserved for ZMAAP allocated addresses (for example 0xB0000000 to 0xBFFFFFFF).

4.7 RFC 2771: An Abstract API for Multicast Address Allocation

RFC 2771 describes the parameters that a client should provide to be given an IP multicast address and the data the allocation system should return. The allocation function is declared like this:

```
alloc_multicast_addr (in AddressFamily family,
                      in Scope scope,
                      in Integer minDesiredAddresses,
                      in Integer maxDesiredAddresses,
                      in Time minDesiredStartTime,
                      in Time maxDesiredStartTime,
                      in Time minDesiredLifetime,
                      in Time maxDesiredLifetime,
                      out Lease multicastAddressSetLease,
                      out Status status)
```

This API is typically the one used by application requiring MADCAP or DHCPv6 allocated addresses.

5. Group announcements

Allocating multicast addresses is closely related to the session announcement problem as some tools (for example SDR) are used for address allocation and session advertisement. This section gives the state of art for existing allocation methods.

5.1 SAP

SAP is described in RFC 2375. This protocol makes it possible to advertise the sessions on a given scope. The mechanism is based on a periodic flooding of group announcements. Even if this protocol corresponds to the needs we have today, it seems difficult to imagine having a large scale multicast deployment with many users using this broadcasting system. Nevertheless, we can easily imagine that this system can be deployed in a site or organisation scope.

SAP makes it possible to have a hierarchy of announcements. `FF0X:0:0:0:0:0:2:7FFE` is the address used for the "root" advertisement. It is then possible to have some directories, all using a different IPv6 multicast address.

We could imagine deploying this hierarchy that is not used at all today (in IPv4 and IPv6) to have some themed channels. RENATER should initiate this, broadcasting all its sessions in a dedicated directory. Including SAP addresses of the directories in a name server would make it possible to easily access some multicast content. For example, we could have a NS record for multicast-sessions.renater.fr that would be an address for an SAP announcement for all RENATER sessions.

5.2 Web, email...

One of the best solutions is to use the web or email for advertising the sessions. All the radio stations already provide on their web site a link for listening to the radio using unicast applications. Those links could refer to multicast addresses in the future when it is widely deployed. Web pages really match the need for SSM broadcasts because there is a similar client/server setup. (The site providing the information of the channel to use would also be the source)

Email can also be used for this purpose. In 6NET project, some IPv6 multicast videoconferences were advertised by emails to better control the dissemination of the events. Emails would certainly better fit to small ASM groups for video-conferences, network games.

5.3 DNS for well known / static group addresses...

This approach is good for allowing people to find out well known group addresses. As for unicast, people can easily remember a name. For example, if RENATER is given a static address for its periodic multicast videoconferences, it is then possible to add a record for this address with the name fmbone-talks.renater.fr

However, the DNS does not provide the corresponding UDP destination port for the multicast session. So DNS is only a partial solution. A UDP port could be reserved for these multicast well

known sessions. People willing to join the videoconferences would just have to remember this name.

According to RFC 3307, IANA is the registry for global addresses, it would be possible to have reverse mappings for these addresses. Those records could be interesting:

- to see the groups joined in the logs on a router (all permanent groups could appear with a name instead of an address)
- to control the usage of permanent groups. The applications could check that a permanent address is registered before using it. This behaviour could provide better control over the ff00::/12 address space – ensuring it is really used for permanent registered addresses.

Nevertheless, it does not seem possible to have reverse records for non-global IPv6 multicast addresses and also this feature may add too much complexity.

6. Existing session announcement implementations

The section covers two tools used within the community for session announcement, which include address allocation functionality. For more details on the applications refer to D5.1. A third tool is also presented in this section: an LDAP repository of SDP sessions.

6.1 Session Directory Tool (SDR)

6.1.1 Overview

SDR is a session directory tool designed to allow the advertisement and joining of multicast conferences on the Mbone. It was originally modelled on *sd* written by Van Jacobson at LBNL, but implements a later version of the session description protocol than sd. SDR was developed at UCL.

When SDR is running it lists all the announced sessions (including authenticated and encrypted sessions, after checking the signature and decrypting the sessions) that are currently scheduled on the Mbone. SDR listens on the standard SAP announcement multicast address for SAP packets and displays the SDP sessions in the main window.

SDR allows the user to join the sessions, where the relevant tools are automatically started up on the correct addresses/ports.

6.1.2 Address allocation

SDR allocates addresses based on the *admin zone*, which is based on the IPv4 case defined in Administratively Scoped IP Multicast [RFC2365]. However its use for SAP and the corresponding IPv6 operation is described in SAP [RFC2974].

SDR has a number of admin scopes predefined which may be selected by the user which will result is the resulting session using a predefined range of addresses, which will be advertised on a corresponding address.

SDR utilises a basic form of the Informed Partitioned Random Allocation [MH97] scheme which essentially use knowledge about existing addresses in use, coupled with a number of factors including scope and ttl resulting in address generation with a low probability of clashes. SDR creates a random address in the range selected and then checks to see if it is already in use, if is it selects another random address otherwise it uses that address.

6.2 Secure Conference Store (SCS)

6.2.1 Overview

The UCL Secure Conference Store is a web-based system for secured creation, storage and access to conference information. Currently, the system provides a store for session details for multicast conferences, so arranged that users can join the sessions easily via their web browser, using the SPAR Java applet to start the media tools in secure mode.

SCS has been developed within UCL.

6.2.2 Address allocation

Since the SCS is designed for advertisement of secure sessions it does not advertise these sessions on a SAP type address as this has been deemed a waste of resources. The SCS sessions are only accessible over the web.

SCS address allocation scheme uses “GLOP” addressing for IPv4 address allocation of private sessions. For IPv6 address allocation SCS currently prepends the global scope FF1E:: suffix to the addresses allocated by the IPv4 system internally. This is planned to be changed so it utilises the unicast-prefixed multicast addressing.

6.3 An LDAP repository of SDP sessions

This solution uses the model where content providers announce their sessions from one (or a few) well known location, so that a user can go to just one (or a few) locations to find large collections of available content. This might also be categorized in some way. This is a bit like TV program guides. By checking one place, you can know what is available from different providers, and there might be several categories, indexes etc. to find all content of a specific type, what is available at a specific time etc.

UNINETT is investigating this approach. Providers describe their sessions using SDP. SDP files are uploaded through web to a central repository. The content of the files is parsed and selected attributes are made searchable. There might also be some additional meta-data that one can search on. The files in the repository can be search for with a web interface, and the user can retrieve the correct SDP file.

LDAP is used for the repository and searchable attributes. This might be used to give access to non-HTTP based clients, and access to a standardized interface for querying for and retrieving the SDP content. A tool similar to sdr that uses SAP today, could instead access the repository using LDAP. It may also be possible to tie multiple LDAP based repositories together using LDAP referrals and possibly CIP for indexing and locating the right repository.

A prototype, not yet IPv6 enabled is available at the following url:

http://www.uninett.no/multimedia/streamingguide/index.en.html

The database is created with all sessions advertised by SAP.

7. Solutions for address allocation

This part summarizes all the possible solutions for IPv6 multicast address allocation.

7.1 SAP for scoped addresses

A tool like SDR fully complies the requirements for scoped announcements, if the choosen domain is not too big (if the need of addresses in the domain remains under 1 000 or 10 000)

This solution must be used only for sessions that need to be advertised as seen in a previous section of this document.

7.2 MADCAP server for a unicast based prefix

It seems possible to deploy now MADCAP servers for unicast based prefixes. Every MADCAP server would be responsible of allocating multicast addresses for hosts having addresses derived of the prefix it is responsible. Nevertheless, there does not seem to be any MADCAP implementation for IPv6. Microsoft is working on an implementation of MADCAP, information can be found at the following url:

http://msdn.microsoft.com/library/default.asp?url=/library/en-us/madcap/madcap/mcastrequestaddress.asp

A group-ID range (for example 0x80000000 to 0x8FFFFFFF) could be dedicated for MADCAP allocations so that there is no overlap with other allocation mechanisms.

7.3 MADCAP server per RP for embedded RP addresses

At the time of writing, we don't know how much the multicast model will be changed with RP embedded addresses, this is why we have to consider different possible deployment scenarios:

- If a site deploys an RP, then it can deploy a MADCAP server that will allocate addresses derived from the /96 prefix derived from the RP. As there is only one allocation server for the RP, then the allocation mechanism is useable and scalable.
- The second scenario is when a provider deploys an RP and shares it so that different sites can use it. The RP-provider could give an address space to any of its clients so that they can deploy MADCAP servers able to allocate addresses in this address space without overlapping. Let's take the following example: RENATER (prefix 2001:660::/32) has the RP 2001:660:3000:1::1. RP based addresses are derived from the prefix FF7E:140:2001:660:3000:1::/96. RENATER offers the RP service for its sites. RENATER can delegate prefixes to any of its sites. For example FF7E:140:2001:660:3000:1:8001::/112 could be allocated for site 1, FF7E:140:2001:660:3000:1:8002::/112 for site 2... In site 1, a MADCAP server could then allocate addresses derived from prefix FF7E:140:2001:660:3000:1:8001::/112...
- We can imagine deploying the MADCAP server on the RP itself. Knowing the RP address (with a static configuration or DHCPv6), the hosts would then request addresses from the

RP. The problem with this deployment scenario is that we can discuss having high level services (MADCAP) on the core network router that would act as RP.

7.4 DHCPv6 server for a unicast based prefix

This solution requires the definition of a new IA type (as said in the DHCPv6 section of this document). This can be easily done and this document recommends this approach. It is clear that DHCPv6 will be widely deployed and that it is a unique opportunity to definitely solve the address allocation problem.

This solution is very close to MADCAP servers for unicast based prefix described above.

To have a complete solution, we recommend that a group-ID range is defined and reserved for DHCPv6 allocations (for example range 0x90000000 to 0x9FFFFFFF). This way, DHCPv6 allocations would not overlap with any other allocations made by any other protocol (for example MADCAP or ZMAAP)

7.5 DHCPv6 server per RP for embedded RP addresses

The deployment scenarios should be exactly the same as those described for MADCAP.

7.6 Group-IDs range for random allocations

Group-ID range from 0xA0000000 to 0xAFFFFFFF should be reserved for random allocations. Therefore, randomly allocated addresses will not overlap with addresses allocated by mechanisms such as MADCAP or DHCPv6.

7.7 Multicast DAD (Duplicate Address Detection)

A “DAD” could work as follows :

- The host (H) who needs an IPv6 multicast address chooses randomly one (G)
- H sends a message RANDOM(H, G) on the group defined by address G
- If another host has already chosen G, it receives this message and sends a DUPLICATE message to H.

This solution is almost the same as ZMAAP, except that the ACLM message would be sent to the address randomly choosen, instead of a pre-defined group.

This solution is just a beginning of work from some members of the 6NET community. It should be then studied completely and integrated if this mechanism can be deployed and has advantages.

7.8 Group-ID ranges summary

We suggest to reserve the following group IDs for the following allocation mechanisms (this table comes in addition to table 2):

Group-ID ranges	Allocation mechanism
0x80000000 to 0x8FFFFFFF	MADCAP
0x90000000 to 0x9FFFFFFF	DHCPv6
0xA0000000 to 0xAFFFFFFF	Random
0xB0000000 to 0xBFFFFFFF	ZMAAP
0xC0000000 to 0xFFFFFFFF	*No usage at that time*

Table 4: group-IDs ranges allocated for allocation mechanisms

8. Conclusion

The IPv6 multicast address allocation problem is very complex. Many documents have been written on this subject, from address architecture to address allocation mechanisms or guidelines. There is a real need for a document integrating all already existing standards and new solutions, taking into account the experience gained with the IPv6 multicast deployment achieved in the frame of the so-called M6bone. This is what the 6NET community tries to do in this deliverable.

Reviewing the current standards and proposals under discussion at the IETF, we see there is a lack of harmony between documents. For example, RFC 3513 says nothing about bit P of the flag field described in RFC 3306. Moreover, RFC 3513 defines a 112 bits long group-ID although RFC 3307 defines guidelines for allocation of addresses with 32 bits long group-ID. This Deliverable is therefore a good opportunity to show the need for a better consistency among the RFCs and can act as a reference for future work.

This document selects all the solutions that can be deployed at the time of writing and makes some recommendations for future standardisation. A summary of all the solutions is described in section 7. Within the 6NET project, we will be testing and deploying the solutions discussed here. We plan to validate these allocation mechanisms on the emerging 6NET multicast test bed (M6Net).

9. References

[DHCPv6] Dynamic Host Configuration Protocol for IPv6 (DHCPv6), draft-ietf-dhc-dhcpv6-28.txt (expires April 2003)

[EMBEDDEDRP] Embedding the Address of RP in IPv6 Multicast Address, draft-savola-mboned-mcast-rpaddr-02.txt (expires September 2003)

[MH97] M. Handley “On scalable multimedia conferencing systems”, PHD thesis. UCL 1997.

[RFC2131] Dynamic Host Configuration Protocol. R. Droms. March 1997. (Obsoletes RFC1541) (Updated by RFC3396)(Status: DRAFT STANDARD)

[RFC 2365] 2365 Administratively Scoped IP Multicast. D. Meyer. July 1998.(Status: BEST CURRENT PRACTICE)

[RFC 2375] IPv6 Multicast Address Assignments. R. Hinden, S. Deering. July 1998. (Status: INFORMATIONAL)

[RFC 2771] An Abstract API for Multicast Address Allocation. R. Finlayson. February 2000. (Status: INFORMATIONAL)

[RFC 2974] 2974 Session Announcement Protocol. M. Handley, C. Perkins, E. Whelan. October 2000. (Status: EXPERIMENTAL)

[RFC 3306] Unicast-Prefix-based IPv6 Multicast Addresses. B. Haberman, D. Thaler. August 2002. (Status: PROPOSED STANDARD)

[RFC 3307] Allocation Guidelines for IPv6 Multicast Addresses. B. Haberman. August 2002. (Status: PROPOSED STANDARD)

[RFC 3396] Encoding Long Options in the Dynamic Host Configuration Protocol (DHCPv4). T. Lemon, S. Cheshire. November 2002. (Updates RFC2131) (Status: PROPOSED STANDARD)

[RFC 3513] IP Version 6 Addressing Architecture. R. Hinden, S. Deering. April 2003. (Obsoletes RFC2373) (Status: PROPOSED STANDARD)

[ZMAAP] Zeroconf Multicast Address Allocation Protocol (ZMAAP), draft-ietf-zeroconf-zmaap-02.txt (expired march 2003)