Network Working Group
Request for Comments: 2908
Category: Informational

D. Thaler
Microsoft
M. Handley
ACIRI
D. Estrin
ISI

September 2000

The Internet Multicast Address Allocation Architecture

Status of this Memo

Copyright Notice

Abstract

This document proposes a multicast address allocation architecture (MALLOC) for the Internet. The architecture is modular with three layers, comprising a host-server mechanism, an intra-domain server-server coordination mechanism, and an inter-domain mechanism.

Table of Contents

1. Introduction

This document proposes a multicast address allocation architecture (MALLOC) for the Internet, and is intended to be generic enough to apply to both IPv4 and IPv6 environments.

As with unicast addresses, the usage of any given multicast address is limited in two dimensions:

Lifetime:
An address has a start time and a (possibly infinite) end time, between which it is valid.

Scope:
An address is valid over a specific area of the network. For example, it may be globally valid and unique, or it may be a private address which is valid only within a local area.

This architecture assumes that the primary scoping mechanism in use is administrative scoping, as described in RFC 2365 [1]. While solutions that work for TTL scoping are possible, they introduce significant additional complication for address allocation [2]. Moreover, TTL scoping is a poor solution for multicast scope control, and our assumption is that usage of TTL scoping will decline before this architecture is widely used.

2. Requirements

From a design point of view, the important properties of multicast allocation mechanisms are robustness, timeliness, low probability of clashing allocations, and good address space utilization in situations where space is scare. Where this interacts with multicast routing, it is desirable for multicast addresses to be allocated in a manner that aids aggregation of routing state.

- Robustness/Availability

  The robustness requirement is that an application requiring the allocation of an address should always be able to obtain one, even in the presence of other network failures.

- Timeliness

  From a timeliness point of view, a short delay of up to a few seconds is probably acceptable before the client is given an address with reasonable confidence in its uniqueness. If the session is defined in advance, the address should be allocated as soon as possible, and should not wait until just before the

session starts. It is in some cases acceptable to change the multicast addresses used by the session up until the time when the session actually starts, but this should only be done when it averts a significant problem such as an address clash that was discovered after initial session definition.

- Low Probability of Clashes

  A multicast address allocation scheme should always be able to allocate an address that can be guaranteed not to clash with that of another session. A top-down partitioning of the address space would be required to completely guarantee that no clashes would occur.

- Address Space Packing in Scarcity Situations

  In situations where address space is scarce, simply partitioning the address space would result in significant fragmentation of the address space. This is because one would need enough spare space in each address space partition to give a reasonable degree of assurance that addresses could still be allocated for a significant time in the event of a network partition. In addition, providing backup allocation servers in such a hierarchy, so that fail-over (including partitioning of a server and its backup from each other) does not cause collisions would add further to the address space fragmentation.

  Since guaranteeing no clashes in a robust manner requires partitioning the address space, providing a hard guarantee leads to inefficient address space usage. Hence, when address space is scarce, it is difficult to achieve constant availability and timeliness, guarantee no clashes, and achieve good address space usage. As a result, we must prioritize these properties. We believe that, when address space is scarce, achieving good address space packing and constant availability are more important than guaranteeing that address clashes never occur. What we aim for in these situations is a very high probability that an address clash does not occur, but we accept that there is a finite probability of this happening. Should a clash occur (or should an application start using an address it did not allocate, which may also lead to a clash), either the clash can be detected and addresses changed, or hosts receiving additional traffic can prune that traffic using source-specific prunes available in IGMP version 3, and so we do not believe that this is a disastrous situation.

  In summary, tolerating the possibility of clashes is likely to allow allocation of a very high proportion of the address space in the presence of network conditions such as those observed in [3].

We believe that we can get good packing and good availability with good collision avoidance, while we would have to compromise packing and availability significantly to avoid all collisions.

Finally, in situations where address space is not scarce, such as with IPv6, achieving good address space usage is less important, and hence partitioning may potentially be used to guarantee no collisions among hosts that use this architecture.

2.1. Address Dynamics

Multicast addresses may be allocated in any of three ways:

Static:
Statically allocated addresses are allocated by IANA for specific protocols that require well-known addresses to work. Examples of static addresses are 224.0.1.1 which is used for the Network Time Protocol [13] and 224.2.127.255 which is used for global scope multicast session announcements. Applications that use multicast for bootstrap purposes should not normally be given their own static multicast address, but should bootstrap themselves using a well-known service location address which can be used to announce the binding between local services and multicast addresses.

Static addresses typically have a permanent lifetime, and a scope defined by the scope range in which they reside. As such, a static address is valid everywhere (although the set of receivers may be different depending on location), and may be hard-coded into applications, devices, embedded systems, etc. Static addresses are also useful for devices which support sending but not receiving multicast IP datagrams (Level 1 conformance as specified in RFC 1112 [7]), or even are incapable of receiving any data at all, such as a wireless broadcasting device.

Scope-relative:
RFC 2365 [1] reserves the highest 256 addresses in every administrative scope range for relative assignments. Relative assignments are made by IANA and consist of an offset which is valid in every scope. Relative addresses are reserved for infrastructure protocols which require an address in every scope, and this offset may be hard-coded into applications, devices, embedded systems, etc. Such devices must have a way (e.g. via MZAP [9] or via MADCAP [4]) to obtain the list of scopes in which they reside.

The offsets assigned typically have a permanent lifetime, and are valid in every scope and location. Hence, the scope-relative address in a given scope range has a lifetime equal to that of the scope range in which it falls.

Dynamic:
For most purposes, the correct way to use multicast is to obtain a dynamic multicast address. These addresses are provided on demand and have a specific lifetime. An application should request an address only for as long as it expects to need the address. Under some circumstances, an address will be granted for a period of time that is less than the time that was requested. This will occur rarely if the request is for a reasonable amount of time. Applications should be prepared to cope with this when it occurs.

At any time during the lifetime of an existing address, applications may also request an extension of the lifetime, and such extensions will be granted when possible. When the address extension is not granted, the application is expected to request a new address to take over from the old address when it expires, and to be able to cope with this situation gracefully. As with unicast addresses, no guarantee of reachability of an address is provided by the network once the lifetime expires.

These restrictions on address lifetime are necessary to allow the address allocation architecture to be organized around address usage patterns in a manner that ensures addresses are aggregatable and multicast routing is reasonably close to optimal. In contrast, statically allocated addresses may be given sub-optimal routing.

3. Overview of the Architecture

The architecture is modular so that each layer may be used, upgraded, or replaced independently of the others. Layering also provides isolation, in that different mechanisms at the same layer can be used by different organizations without adversely impacting other layers.

There are three layers in this architecture (Figure 1). Note that these layer numbers are different from the layer numbers in the TCP/IP stack, which describe the path of data packets.

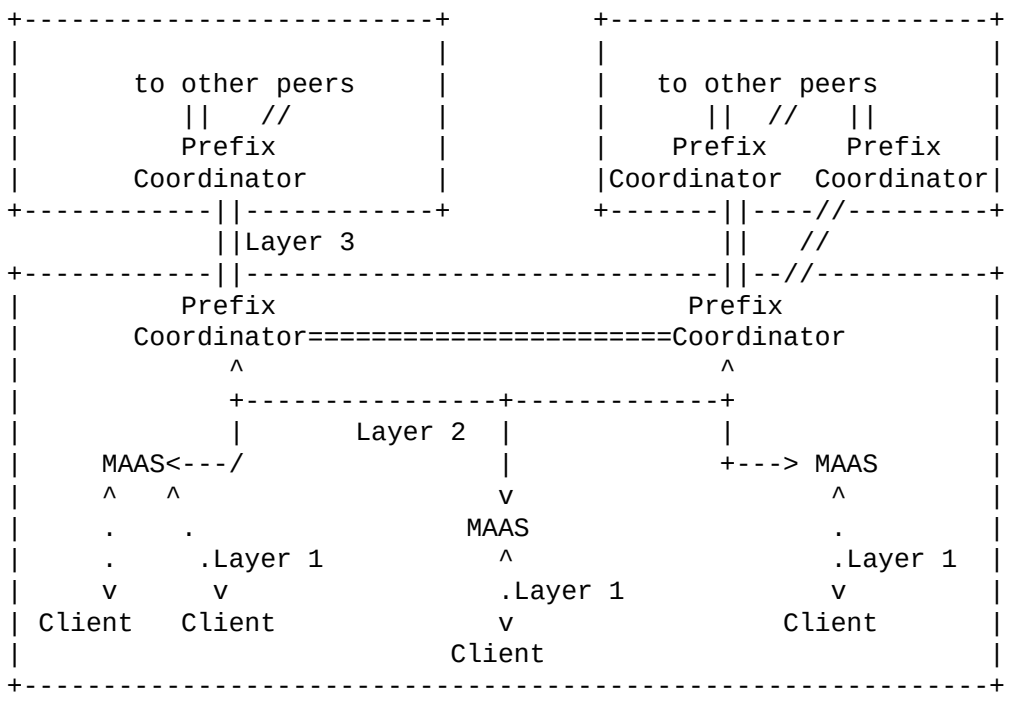

Figure 1: An Overview of the Multicast Address Allocation Architecture

Layer 1
A protocol or mechanism that a multicast client uses to request a multicast address from a multicast address allocation server (MAAS). When the server grants an address, it becomes the server's responsibility to ensure that this address is not then reused elsewhere within the address's scope during the lifetime granted.

Examples of possible protocols or mechanisms at this layer include MADCAP [4], HTTP to access a web page for allocation, and IANA static address assignments.

An abstract API for applications to use for dynamic allocation, independent of the Layer 1 protocol/mechanism in use, is given in [11].

Layer 2
An intra-domain protocol or mechanism that MAAS's use to coordinate allocations to ensure they do not allocate duplicate addresses. A MAAS must have stable storage, or some equivalent robustness mechanism, to ensure that uniqueness is preserved across MAAS failures and reboots.

MAASs also use the Layer 2 protocol/mechanism to acquire (from "Prefix Coordinators") the ranges of multicast addresses out of which they may allocate addresses.

In this document we use the term "allocation domain" to mean an administratively scoped multicast-capable region of the network, within which addresses in a specific range may be allocated by a Layer 2 protocol/mechanism.

Examples of protocols or mechanisms at this layer include AAP [5], and manual configuration of MAAS's.

Layer 3
An inter-domain protocol or mechanism that allocates multicast address ranges (with lifetimes) to Prefix Coordinators. Individual addresses may then be allocated out of these ranges by MAAS's inside allocation domains as described above.

Examples of protocols or mechanisms at this layer include MASC [6] (in which Prefix Coordinators are typically routers without any stable storage requirement), and static allocations by AS number as described in [10] (in which Prefix Coordinators are typically human administrators).

Each of the three layers serves slightly different purposes and as such, protocols or mechanisms at each layer may require different design tradeoffs.

4. Scoping

To allocate dynamic addresses within administrative scopes, a MAAS must be able to learn which scopes are in effect, what their address ranges and names are, and which addresses or subranges within each scope are valid for dynamic allocation by the MAAS.

The first two tasks, learning the scopes in effect and the address range and name(s) of each scope, may be provided by static configuration or dynamically learned. For example, a MAAS may simply passively listen to MZAP [9] messages to acquire this information.

To determine the subrange for dynamic allocation, there are two cases for each scope, corresponding to small "indivisible" scopes, and big "divisible" scopes. Note that MZAP identifies which scopes are divisible and which are not.

(1) For small scopes, the allocation domain corresponds to the entire topology within the administrative scope. Hence, all MAASs inside the scope may use the entire address range (minus the last

256 addresses reserved as scope-relative addresses), and use the Layer 2 mechanism/protocol to coordinate allocations. For small scopes, Prefix Coordinators are not involved.

Hence, for small scopes, the effective "allocation domain" area may be different for different scopes. Note that a small, indivisible scope could be larger or smaller than the Allocation Scope used for big scopes (see below).

(2) For big scopes (including the global scope), the area inside the scope may be large enough that simply using a Layer 2 mechanism/protocol may be inefficient or otherwise undesirable. In this case, the scope must span multiple allocation domains, and the Layer 3 mechanism/protocol must be used to divvy up the scoped address space among the allocation domains. Hence, a MAAS may learn of the scope via MZAP, but must acquire a subrange from which to allocate from a Prefix Coordinator.

For simplicity, the effective "allocation domain" area will be the same for all big scopes, being the granularity at which all big scopes are divided up. We define the administrative scope at this granularity to be the "Allocation Scope".

4.1. Allocation Scope

The Allocation Scope is a new administrative scope, defined in this document and to be reserved by IANA with values as noted below. This is the scope that is used by a Layer 2 protocol/mechanism to coordinate address allocation for addresses in larger, divisible scopes.

We expect that the Allocation Scope will often coincide with a unicast Autonomous System (AS) boundary.

If an AS is too large, or the network administrator wishes to run different intra-domain multicast routing in different parts of an AS, that AS can be split by manual setup of an allocation scope boundary that is not an AS boundary. This is done by setting up a multicast boundary dividing the unicast AS into two or more multicast allocation domains.

If an AS is too small, and address space is scarce, address space fragmentation may occur if the AS is its own allocation domain. Here, the AS can instead be treated as part of its provider's allocation domain, and use a Layer 2 protocol/mechanism to coordinate allocation between its MAAS's (if any) and those of its provider. An AS should probably take this course of action if:

- it is connected to a single provider,
- it does not provide transit for another AS, and
- it needs fewer than (say) 256 multicast addresses of larger than AS scope allocated on average.

4.1.1. The IPv4 Allocation Scope -- 239.251.0.0/16

The address space 239.251.0.0/16 is to be reserved for the Allocation Scope. The ranges 239.248.0.0/16, 239.249.0.0/16 and 239.250.0.0/16 are to be left unassigned and available for expansion of this space. These ranges should be left unassigned until the 239.251.0.0/16 space is no longer sufficient.

4.1.2. The IPv6 Allocation Scope -- SCOP 6

The IPv6 "scop" value 6 is to be used for the Allocation Scope.

5. Overview of the Allocation Process

Once Layer 3 allocation has been performed for large, divisible scopes, and each Prefix Coordinator has acquired one or more ranges, then those ranges are passed to all MAAS's within the Prefix Coordinator's domain via a Layer 2 mechanism/protocol.

MAAS's within the domain receive these ranges and store them as the currently allowable addresses for that domain. Each range is valid for a given lifetime (also acquired via the Layer 3 mechanism/protocol) and is not revoked before the lifetime has expired. MAAS's also learn of small scopes (e.g., via MZAP) and store the ranges associated with them.

Using the Layer 2 mechanism/protocol, each MAAS ensures that it will exclude any addresses which have been or will be allocated by other MAAS's within its domain.

When a client needs a multicast address, it first needs to decide what the scope of the intended session should be, and locate a MAAS capable of allocating addresses within that scope.

To pick a scope, the client will either simply choose a well-known scope, such as the global scope, or it will enumerate the available scopes (e.g., by sending a MADCAP query, or by listening to MZAP messages over time) and allow a user to select one.

Locating a MAAS can be done via a variety of methods, including manual configuration, using a service location protocol such as SLP [12], or via a mechanism provided by a Layer 1 protocol itself. MADCAP, for instance, includes such a facility.

Once the client has chosen a scope and located a MAAS, it then requests an address in that scope from the MAAS located. Along with the request it also passes the acceptable range for the lifetimes of the allocation it desires. For example, if the Layer 1 protocol in use is MADCAP, the client sends a MADCAP REQUEST message to the MAAS, and waits for a NAK message or an ACK message containing the allocated information.

Upon receiving a request from a client, the MAAS then chooses an unused address in a range for the specified scope, with a lifetime which both satisfies the acceptable range specified by the client, and is within the lifetime of the actual range.

The MAAS uses the Layer 2 mechanism/protocol to ensure that such an address does not clash with any addresses allocated by other MAASs. For example, if Layer 2 uses manual configuration of non-overlapping ranges, then this simply consists of adhering to the range configured in the local MAAS. If, on the other hand, AAP is used at Layer 2 to provide less address space fragmentation, the MAAS advertises the proposed allocation domain-wide using AAP. If no clashing AAP claim is received within a short time interval, then the address is returned to the client via the Layer 1 protocol/mechanism. If a clashing claim is received by the MAAS, then it chooses a different address and tries again. AAP also allows each MAAS to pre-reserve a small "pool" of addresses for which it need not wait to detect clashes.

If a domain ever begins to run out of available multicast addresses, a Prefix Coordinator in that domain uses the Layer 3 protocol/mechanism to acquire more space.

6. Security Considerations

The architecture described herein does not prevent an application from just sending to or joining a multicast address without allocating it (just as the same is true for unicast addresses today). However, there is no guarantee that data for unallocated addresses will be delivered by the network. That is, routers may drop data for unallocated addresses if they have some way of checking whether a destination address has been allocated. For example, if the border routers of a domain participate in the Layer 2 protocol/mechanism and cache the set of allocated addresses, then data for unallocated

addresses in a range allocated by that domain can be dropped by creating multicast forwarding state with an empty outgoing interface list and/or pruning back the tree branches for those groups.

A malicious application may attempt a denial-of-service attack by attempting to allocate a large number of addresses, thus attempting to exhaust the supply of available addresses. Other attacks include releasing or modifying the allocation of another party. These attacks can be combatted through the use of authentication with policy restrictions (such as a maximum number of addresses that can be allocated by a single party).

Hence, protocols/mechanisms that implement layers of this architecture should be deployable in a secure fashion. For example, one should support authentication with policy restrictions, and should not allow someone unauthorized to release or modify the allocation of another party.

7. Acknowledgments

Steve Hanna provided valuable feedback on this document. The members of the MALLOC WG and the MBone community provided the motivation for this work.

8. References

[1] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

[2] Mark Handley, "Multicast Session Directories and Address Allocation", Chapter 6 of PhD Thesis entitled "On Scalable Multimedia Conferencing Systems", University of London, 1997.

[3] Mark Handley, "An Analysis of Mbone Performance", Chapter 4 of PhD Thesis entitled "On Scalable Multimedia Conferencing Systems", University of London, 1997.

[4] Hanna, S., Patel, B. and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, December 1999.

[5] Handley, M. and S. Hanna, "Multicast Address Allocation Protocol (AAP)", Work in Progress.

[6] Estrin, D., Govindan, R., Handley, M., Kumar, S., Radoslavov, P. and D. Thaler, "The Multicast Address-Set Claim (MASC) Protocol", RFC 2909, September 2000.

[7] Deering, S., "Host Extensions for IP Multicasting", STD 5, RFC 1112, August 1989.

[8] Rekhter, Y. and T. Li, "A Border Gateway Protocol 4 (BGP-4)", RFC 1771, March 1995.

[9] Handley, M., Thaler, D. and R. Kermode, "Multicast-Scope Zone Announcement Protocol (MZAP)", RFC 2776, February 2000.

[10] Meyer, D. and P. Lothberg, "GLOP Addressing in 233/8", RFC 2770, February 2000.

[11] Finlayson, R., "Abstract API for Multicast Address Allocation", RFC 2771, February 2000.

[12] Guttman, E., Perkins, C., Veizades, J. and M. Day, "Service Location Protocol, Version 2", RFC 2608, June 1999.

[13] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation and Analysis", RFC 1305, March 1992.

9. Authors' Addresses

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052-6399

EMail: dthaler@microsoft.com

Mark Handley
AT&T Center for Internet Research at ICSI
1947 Center St, Suite 600
Berkeley, CA 94704

EMail: mjh@aciri.org

Deborah Estrin
Computer Science Dept/ISI
University of Southern California
Los Angeles, CA 90089

EMail: estrin@usc.edu

10. Full Copyright Statement

Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.