NETWORK Working Group
INTERNET-DRAFT
Category: Standards Track
22 October 2002
Expires in six months

Octavian Catrina, Editor
International University
Dave Thaler
Bernard Aboba
Microsoft
Erik Guttman
Sun Microsystems

Zeroconf Multicast Address Allocation Protocol (ZMAAP)

<draft-ietf-zeroconf-zmaap-02.txt>

Status of this Memo

Copyright Notice

Abstract

Today, with the rapid rise of home networking, there is an increasing need for auto-configuration mechanisms. This document specifies a protocol to be used on small networks without a multicast address allocation server in order to allow peer to peer allocation of multicast addresses.

Table of Contents

1. Introduction

Servers and network administration staff are not available in all environments. Home networks and ad-hoc networks, for example, need to rely entirely on zero-configuration protocols [ZCREQTS]. This document defines the Zeroconf Multicast Address Allocation Protocol (ZMAAP), that allows hosts on small networks to allocate addresses without the need for multicast address allocation servers.

The Internet Multicast Address Allocation Architecture [RFC2908] provides a three-layer framework for allocating multicast addresses. The top layer is used to decide which address range to use for allocation. The middle layer is used to coordinate among peers allocating from the same range. The bottom layer is used to provide scalability in a managed environment whereby a small number of servers can allocate addresses to a large number of hosts. In a zero-configuration environment, less scalability is required, and hence the bottom layer will not be needed. ZMAAP thus fits into the Multicast Address Allocation Architecture as a middle-layer protocol which is used between end nodes.

ZMAAP allows applications to allocate unique addresses from certain address ranges, to defend those allocations and to detect conflicts

in those allocations.

1.1 Changes since the previous version of this document

1. Timers introduced before AIU responses to ACLM messages are sent

   If there are many members of a group, an ACLM for that group's allocation would cause a massive implosion of AIUs - like a distributed denial of service attack. While it is unlikely that ZMAAP will be used for large groups maintained by of thousands of mini-MAASs, we won't rule out the possibility.

   Currently mini-MAASs respond to ACLMs with an AIU immediately.

   Now, before sending an AIU in response to an ACLM, mini-MAASs will wait a random interval and listen for other AIUs defending the allocation. If such an AIU is received, the mini-MAAS cancels its waiting timer and does not send the AIU.

2. To prevent thrashing, rules slow down and stop allocation attempts

   As more and more addresses in an address space are allocated, claims for random address ranges in the address space have a higher chance of collision. To prevent thrashing (excessively repeated attempts for allocation), new rules are

3. Security: Text on WEP removed

4. Add a magic number to ZMAAP headers

   This will be used to detect collision in the use of the same multicast address by multiple applications.

5. Clarification of address range conflicts

   Non overlapping ranges conflict even if the lease id is the same.

2. Terminology

This document uses the following terms:

Multicast
  IP Multicast, as defined in [RFC1112] and [RFC2460].

Multicast Address
  An IP multicast address or group address, as defined in [RFC1112] and [RFC2373]. An identifier for a group of nodes.

Multicast Scope
A range of multicast addresses configured so that traffic

sent to these addresses is limited to some subset of the internetwork. See [RFC2365] and [RFC2373].

Multicast Address Allocation Server (MAAS)
A node providing multicast address allocation services to network clients.

Mini-MAAS
A service providing multicast address allocation services to applications running on the same host. Mini-MAASs cooperate to provide network-wide services in small networks without MAASs.

In this document, the key words "MAY", "MUST", "MUST NOT", "optional", "recommended", "SHOULD", and "SHOULD NOT", are to be interpreted as described in [RFC2119].

3. Requirements and Design Considerations

As described in [RFC2771], a multicast allocation API provides two main services to multicast applications. First, it allows enumeration of the set of available multicast scopes applications may attempt to allocate in. Second, applications can dynamically allocate multicast addresses in scopes they specify.

Hosts may also use MADCAP [RFC2730] for these features. MADCAP provides various functions, including allocation of addresses in scopes which are not available using ZMAAP.

In general, applications should be unaware of which protocol is being used to allocate multicast addresses (e.g., MADCAP, ZMAAP, or local allocation of SSM [SSM] addresses).

ZMAAP satisfies the general requirements for multicast address allocation mechanisms specified in [RFC2908]: robustness, availability and low probability of clashes in the presence of host and network failures, short allocation delay and efficient use of the address space. ZMAAP is expected to work in a unreliable environment (for example on laptops in an ad-hoc network, that can be switched off and back on at any moment).

Applications can obtain the following services from a mini-MAAS making use of ZMAAP. For further discussion, see Appendix A.

- Obtain an enumeration of supported multicast scopes.
- Allocate an address in a specified scope.
- Renew an existing address allocation, which an application is using.
- Get notified when an allocation has been cancelled by the mini-

MAAS due to an allocation conflict.

4. Zeroconf Multicast Address Configuration Protocol

4.1 Protocol Overview

ZMAAP is a peer-to-peer protocol that allows mini-MAASs to coordinate their multicast address allocations. Three messages are used for this purpose: Address In Use (AIU), Address Claim (ACLM).

To obtain a multicast address allocation, an application makes a request to a local mini-MAAS, specifying the scope and number of addresses. The mini-MAAS selects the addresses to be allocated and multicasts a ZMAAP ACLM request to the other mini-MAASs. The mini-MAAS issues this request repeatedly.

Two things may occur. If an AIU response is received, the mini-MAAS has requested a range of addresses which conflicts with an existing allocation. In this case, the mini-MAAS must select a new range of addresses and try again, or give up. If, on the other hand, the receives no AIU response when the allotted time expire, it assumes it has succeeded in allocating an address.

A mini-MAAS will defend an address allocation which it has made. An application can also ask a local mini-MAAS to defend an address allocation it uses, learned through another mechanism (for example, through the use of an API). A mini-MAAS which receives an ACLM message which it defends will issue an AIU response immediately, indicating that the allocation already exists and the ACLM message conflicts.

Mini-MAASs may cache information about allocations to aid in selecting addresses which do not conflict with others. A cache entry is maintained for the duration of the intended allocation lifetime indicated in ZMAAP messages. The lifetime can be extended using AIU messages. If an address is not in the cache, it is considered available for allocation.

4.2 Transmission of ZMAAP Messages

A mini-MAAS sends AIU messages for the addresses that it currently has allocated, before their allocation lifetime expires.

All ZMAAP messages are multicast using UDP. The reserved UDP port number is TBD. The address allocations communicated in any message MUST belong to the same multicast scope.

All messages are sent to a reserved IPv4 scope-relative multicast address, or IPv6 variable scope multicast address, called in the following the ZMAAP multicast address.

These address assignments are TBD. The destination address of a message MUST be in the same multicast scope as the address

allocations it contains. A mini-MAAS MUST listen to messages sent to the ZMAAP multicast address for all scopes in which it is has allocated addresses or is in the process of allocating addresses.

ZMAAP is used to allocate addresses in all ranges for which coordination must be done among multiple machines, but within an area smaller than an Admin scope. This way, the ranges used by MADCAP, ZMAAP, and SSM are all disjoint and clear ownership is preserved. MADCAP is used for ranges which require coordination across an Admin scope or larger, and SSM does not require coordination among multiple machines.

The ranges which are defined or under discussion today, which ZMAAP would be used for, include:

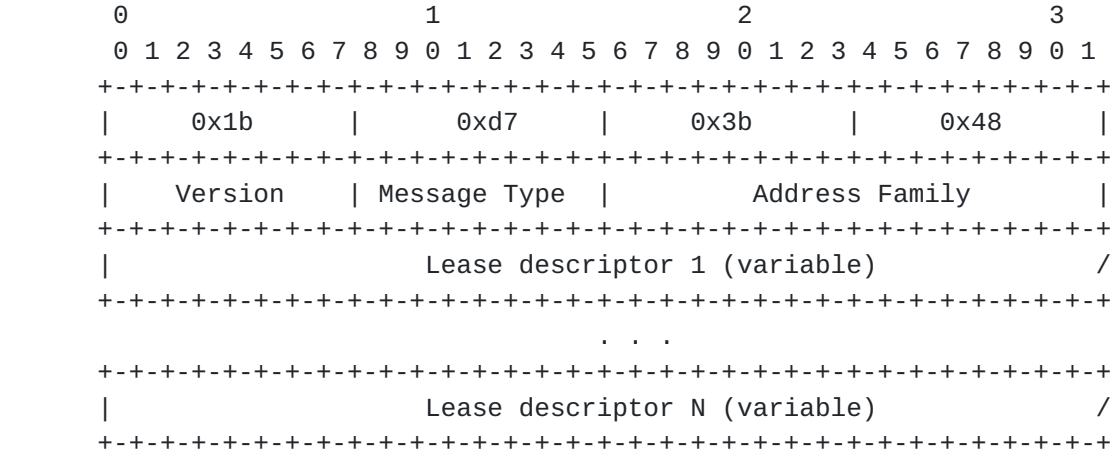
Allocation Scope
(1) IPv4 Dynamic Link-Local [TBD]
(2) IPv6 Dynamic Link-Local [RFC2373]
(3) IPv6 Dynamic Subnet-Local [RFC2373]
(4) IPv4 Unicast-Prefix-based [TBD]
(5) IPv6 Unicast-Prefix-based [UNIPREFIX]

To date, no range of addresses for (1) or (4) has been defined.

4.3 Protocol Message Format

ZMAAP uses two messages: Address In Use (AIU), used to announce an existing address allocation, and Address Claim (ACLM), used to announce a desired address allocation. ZMAAP implementations MUST support both these messages.

The ZMAAP messages have the following common format:



The initial 4 bytes in the message constitute a 'magic number' which is used to detect protocol collisions on the port and multicast group

where ZMAAP is operating. Any message which does not begin with the magic number MUST be silently discarded.

The Version field indicates the ZMAAP version. It MUST be 1 for the version described in this document.

The Message Type field defines the type of ZMAAP message. The following values are defined:

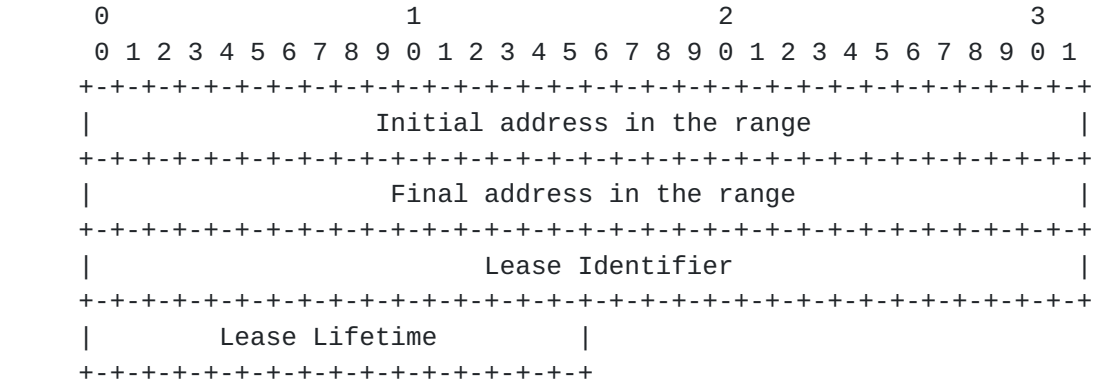
Value	Message type
0	Address Claim (ACLM)
1	Address In Use (AIU)

The Address Family field indicates the address family for all the addresses in the ZMAAP message, using the values defined by IANA [IANA]. This version of ZMAAP supports the IPv4 and IPv6 address families:

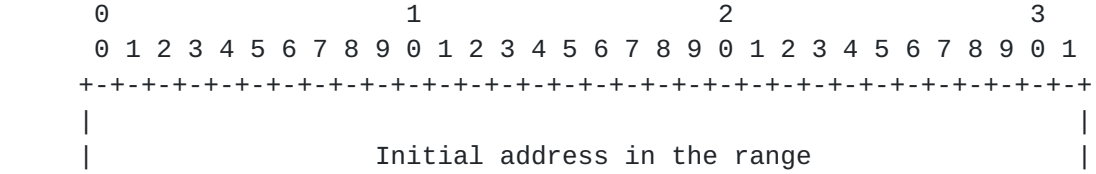
Value	Address Family
1	IPv4
2	IPv6

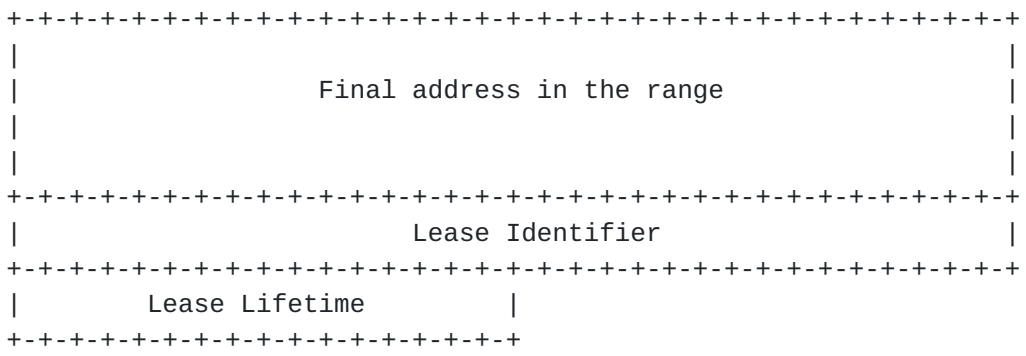
Lease descriptors describe address allocations in ZMAAP messages.

An IPv4 address is represented by 4 bytes in network byte order. The lease descriptor for IPv4 addresses has the following format:



An IPv6 address is represented by 16 bytes in network byte order. The lease descriptor for IPv6 addresses has the following format:





The initial and final addresses define the range of addresses claimed or allocated. When individual addresses are allocated rather than ranges, the Initial and Final addresses are identical.

Addresses in the lease descriptor belong to the address family indicated by the Address Family field in the message header.

The Lease Identifier distinguishes different allocations of the same address range. It is assigned by the allocator mini-MAAS, using an implementation dependent method. For example, it can be computed as a hash of the allocator's address and the allocation time (and UDP transmission port in case more than one mini-MAAS resides on the same host).

The Lease Lifetime is the number of seconds which a lease may be cached after it has been received.

The number of lease descriptors in a ZMAAP message is limited by the condition that the message fits into a payload of maximum 576 bytes for IPv4 packets and 1280 bytes for IPv6 packets. If the number of lease descriptors is too large to fit into the maximum payload, they are sent in separate ZMAAP messages.

4.4 ZMAAP mini-MAAS behavior

A ZMAAP mini-MAAS performs four functions: Claiming, defending, verifying and detecting conflicts in allocations.

A mini-MAAS MUST maintain state information for the allocations it makes or it maintains as a result of requests from its local applications. This will be referred to as the 'allocation record.'

It MAY also cache state information for other allocations, learned from received ZMAAP messages. This could be useful, for example, to assist in selecting multicast addresses that will be unlikely to conflict with preexisting allocations. The term 'allocation record' as used below will NOT include this state information.

4.4.1 Claiming an address

To allocate multicast addresses, an application makes a request from the local mini-MAAS, indicating the scope, the number of addresses desired and the allocation lifetime.

The mini-MAAS selects free addresses by consulting its allocation record and creates a lease descriptor. To reduce the likelihood of collisions, a random selection of the free addresses is strongly recommended (see Section 4.3). A unique identifier ("lease identifier") is associated with each allocation to distinguish allocations of the same addresses.

The mini-MAAS starts the claiming by sending an ACLM message containing the lease descriptor. After sending the ACLM message it MUST start a Claim Timer for [ANNOUNCE-WAIT] seconds. Also, it SHOULD resend the ACLM message, first after [RESEND-WAIT] seconds, and later doubling after each send, until either the Claim Timer expires, or the claim is aborted.

If the mini-MAAS receives an AIU message or an ACLM message listing addresses being claimed, it MUST abort the claiming, stop the Claim Timer, and give up on the addresses indicated in the AIU or ACLM message. It MAY select new addresses and restart the claiming procedure, after waiting. Each successive attempt T, the mini-MAAS waits for a uniformly distributed random interval from 0 to ([RESTART-WAIT] * T) milliseconds.

If the Claim Timer expires, the mini-MAAS commits the allocation and communicates the lease to the application. To complete a new address allocation, a mini-MAAS MUST send an AIU message containing its lease descriptor.

4.4.2 Defending an Address

A mini-MAAS MUST defend all allocations in its allocation record.

If there is any overlapping between address ranges in the received ACLM and address ranges in its allocation record, the mini-MAAS MUST respond with an AIU. This AIU contains the Lease Descriptors of the mini-MAASs own allocations whose address ranges overlap with those in the ACLM.

Note that the AIU is sent regardless of whether the Lease Identifiers for an overlapping range match (same allocation) or not (conflicting allocations). The other mini-MAAS will be able to distinguish these cases using the Lease Identifier and take an appropriate action.

To send an AIU in response to an ACLM message, the mini-MAAS starts a

random Defense Timer, uniformly distributed from 0 to [DEFEND-WAIT] milliseconds. During this waiting interval, the mini-MAAS listens

for AIU response to the ACLM. If the mini-MAAS detects the same AIU which it would have sent, the mini-MAAS cancels the Defense Timer. Otherwise, when the Defense Timer expires, the mini-MAAS sends the AIU.

ZMAAP allows address allocations to persist without the initial allocator. Other session participants can share the defense of the address allocation by registering with their local mini-MAASs and indicating the lease identifier (learned from the session initiator via some session announcement mechanism, see Appendix B.) A mini-MAAS MAY add an allocation to its record, even if it was not the mini-MAAS which allocated the address. Before it does this, it MUST verify the lease identifier is correct (see Section 4.4.3).

4.4.3 Verifying a Lease Descriptor

A valid lease identifier matches an existing (defended) allocation, and does not conflict with any other allocations.

A mini-MAAS MUST verify the validity of a lease identifier before adding it to its allocation record for 'shared defense' of an address (see Section 4.4.2 and Appendix A).

To verify a lease identifier is correct, the mini-MAAS claims it using ACLM messages, as described in section 4.4.1.

4.4.4. Detecting an Allocation Conflict

A mini-MAAS that receives an AIU message MUST check its allocation record to determine the status of the indicated allocations.

If the mini-MAAS is currently trying to allocate any of the addresses in the AIU message, the mini-MAAS MUST try a different address or give up trying to allocate addresses (see Section 4.4.1).

If any ranges in the AIU message overlap without exactly matching recorded allocations then an allocation conflict exists. Also, if an address range in an AIU matches a recorded address allocation with a different lease identifiery, this also indicates a conflict. The mini-MAAS MUST remove the allocation from its allocation record. The mini-MAAS will inform any local applications registered for the canceled allocation, if it has implemented this functionality (see Appendix A).

4.4.5. Deallocation and Lease Lifetime

A mini-MAAS maintains an address allocation in its record as long as a local application uses it.

When an allocation is not needed anymore by any local application, the mini-MAAS removes it from the allocation record, so it stops

defending it. However, the allocation can still be defended by other mini-MAASs interested in preserving it. When an allocation is no longer defended by any mini-MAAS, the addresses can be reallocated.

The Lease Lifetime indicated in ZMAAP messages limits the lifetime of cache entries used to assist in address selection for new allocations. A mini-MAAS MAY send AIUs to extend an expiring lifetime for any of its allocations. It SHOULD NOT send AIUs to reduce the lifetime (in particular set it to 0), since other mini-MAASs may intend to preserve it.

5. Timer Default Values

ANNOUNCE-WAIT	3 seconds
RESEND-WAIT	200 milliseconds
RESTART-WAIT	100 milliseconds
DEFEND-WAIT	100 milliseconds

6. Security Considerations

In the interest of simplicity, this draft does not prescribe a means of securing the multicast auto-configuration mechanism. Thus it is possible that hosts will allocate conflicting multicast addresses for a period of time, or that non-conforming hosts will attempt to deny service to other hosts by allocating the same multicast addresses.

A 'greedy' mini-MAAS which simply ignored others' advertisements and allocated any address it wished could steal addresses from others. If there were more than one such 'greedy' mini-MAAS on the network, address allocation conflicts would never be detected or corrected.

These threats are most serious in wireless networks since attackers on a wired network will require physical access to the home network, while wireless attackers may reside outside the home.

In order to counter these threats, IP or link layer security could be applied to authenticate messages and thereby prevent the attacks listed above. For example, if all authorized hosts in the network shared a preconfigured key, this could be used with the IP Authentication Header [RFC2402] to discard unauthenticated datagrams. Admittedly, preconfiguration of keys runs counter to the goals of zero configuration networking. There's no free lunch.

7. IANA Considerations

This document requires an allocation for a UDP port number, and a range of IPv4 multicast addresses for link-local dynamic multicast address allocation.

Appendix A Application Programmer Interface (API) Definition

The ZMAAP API will be presented as a set of abstract functions followed by language specific mappings. These functions and their names are derived from the Abstract API for Multicast Address Allocation [RFC2771]. This API is specified in a separate document [ZMAAPAPI].

What distinguishes the ZMAAP API from the general multicast address allocation API is the need for two additional functions: Shared ownership, for renewal and defense of allocations, and conflict notification for applications to determine if and when an allocation can no longer be used.

Normally the mini-MAAS allocating an address maintains an allocation record entry for it. This implies the mini-MAAS will defend the address from conflicting claims and will send AIU messages before the lease lifetime expires for all active allocations. In the case of 'shared ownership', (initiated via the ZMAAP API), a mini-MAAS first verifies that the lease is still valid (section 4.4.3), then it adds the record to its own allocation record.

Appendix B Session Management Implications

Multicast address allocation alone is not useful. A mechanism is needed in order to discover sessions using multicast allocations. This serves applications attempting to join an existing session, those initiating new sessions and also for rendezvous at a new session address if there has been a conflict with an address currently in use.

Sessions are announced using the Session Announcement Protocol (SAP) [RFC2974]. They are described using the Session Description Protocol (SDP) [RFC2327].

SAP describes how to announce sessions for IPv4 using global and adminstrative scoped multicast.

This technique can be used to announce sessions which are allocated in other scopes as well. For IPv4 link-local scope session announcement, an IP time-to-live of 1 is used. This limits propogation of the announcement to the same scope where it is meaningful.

An additional SDP session attribute SHOULD be included for use in announcing sessions for addresses allocated with ZMAAP. This attribute allows coordination with ZMAAP.

```
a=zmaap-lease-id:<lease-id>
```

<lease-id> is set to the lease identifier associated with the

announced session.

An application which receives a session announcement with this attribute may use it to form a Lease Descriptor and request the ZMAAP API to either defend the allocation or for notification if there is an address allocation conflict.

References

[IANA] Address Family Numbers. http://www.isi.edu/in-notes/iana/assignments/address-family-numbers

[MDNS] Ebisov, L., Aboba, B., Thaler, D., "Multicast DNS", draft-ietf-dnsext-mdns-06.txt, October 2001. Work in progress.

[RFC1112] Deering, S., "Host Extensions for IP Multicasting", RFC 1112, August 1989.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, March 1997.

[RFC2327] Handley, M., Jacobson, V., "SDP: Session Description Protocol", RFC 2327, April 1998.

[RFC2365] Meyer, D., "Administratively Scoped IP Multicast", BCP 23, RFC 2365, July 1998.

[RFC2373] Hinden, R. and Deering, S., "IP Version 6 Addressing Architecture", RFC 2373, July 1998.

[RFC2402] Kent, S., Atkinson, R., "IP Authentication Header", RFC 2402, November 1998.

[RFC2460] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", RFC 2460, December 1998.

[RFC2730] Hanna, S., Patel, B., and M. Shah, "Multicast Address Dynamic Client Allocation Protocol (MADCAP)", RFC 2730, December 1999.

[RFC2771] Finlayson, R., "An Abstract API for Multicast Address Allocation", RFC 2771, February 2000.

[RFC2908] Thaler, D., Handley, M., and D. Estrin, "The Internet Multicast Address Allocation Architecture", RFC 2908, September 2000.

[RFC2974] Handley, M., Perkins, C., Whelan, E., "Session Announcement Protocol", RFC 2974, October 2000.

[SSM] IANA, "Single-source IP Multicast Address Range",

http://www.isi.edu/in-notes/iana/assignments/single-source-multicast, October 1998.

[UNIPREFIX] Haberman, B., Thaler, D., "Unicast-Prefix-based IPv6 Multicast Addresses", Internet Draft, draft-ietf-ipngwg-uni-based-mcast-03.txt, October 2001. Work in progress.

[V4LL] Cheshire, S., Aboba, B., "Dynamic Configuration of IPv4 link-local addresses", draft-ietf-zeroconf-ipv4-linklocal-04.txt, July 2001. Work in progress.

[ZCREQTS] Hattig, M., "Zeroconf Requirements", draft-ietf-zeroconf-reqts-06.txt, November 2000. Work in progress.

[ZMAAPAPI] Guttman, E., "An API for the Zeroconf Multicast Address Allocation Protocol", draft-ietf-zeroconf-zmaap-api-00.txt. Work in Progress.

Acknowledgments

This draft has been benefited from work by Mark Handley and Steve Hanna on the Multicast Address Allocation Protocol (AAP). Prashant Agarwal's master thesis work at International University provided helpful insights.

Authors' Addresses

Octavian Catrina, Editor
International University in Germany
International University Campus 2
D-76646 Bruchsal, Germany

Phone: +49 7251 700 221
EMail: Octavian.Catrina@i-u.de

Dave Thaler
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 (425) 703-8835
EMail: dthaler@microsoft.com

Bernard Aboba
Microsoft Corporation
One Microsoft Way
Redmond, WA 98052

Phone: +1 (425) 936-6605

EMail: bernarda@microsoft.com

Erik Guttman
Sun Microsystems
Eichhoelzelstr. 7
74915 Waibstadt Germany

Phone: +49 172 865 5497
Email: erik.guttman@sun.com

Full Copyright Statement

Acknowledgement

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