Mboned Working Group   Roger Kermode
Internet Engineering Task Force   Motorola
INTERNET-DRAFT   Dave Thaler
5 September 2000   Microsoft
Expires 5 March 2001

Scoped Address Discovery Protocol (SADP)
<**draft-ietf-mboned-sadp-02.txt**>

Status of this Memo

Abstract

   This document defines an application-layer protocol, the Scoped
   Address Discovery Protocol (SADP), for discovering the scoped
   multicast address(es) associated with a session at particular scopes
   within a hierarchically nested set of multicast scopes. SADP is
   designed to work within the context of Multicast Address Allocation
   Architecture [MAAA]. It is intended that SADP will provide the
   necessary general services for reliable multicast and searching
   applications to use expanding-scope searches in lieu of the well
   known, but less efficient expanding-ring search.

Copyright Notice

Contents

1. **Introduction**

   Administrative scoping [RFC2365] provides a useful means for limiting
   the spread of IP multicast traffic across the Internet. Unlike Time-
   To-Live (TTL) scoping, administrative scoping provides the means to
   ensure that, for a given scope and ignoring packet loss, the same set
   of nodes will receive a message, regardless of which node sent the
   message. Thus, the use of administrative scoping greatly simplifies
   the design of multicast protocols that require localization, since
   the non-reception of sent packets is solely due to loss and not
   design.

   The Multicast Address Dynamic Client Allocation Protocol (MADCAP)
   [RFC2730, NESTOPT] will provide applications with the means for
   discovering the various scopes that are locally visible at each point
   in the Internet. The determination of which scopes nest inside each
   other will be performed by the Multicast-Scope Zone Announcement
   Protocol (MZAP) [RFC2776]. MZAP's ability to provide this service
   will allow scopes to be arranged into hierarchies so that
   applications can then use expanding scope searches instead of the
   less efficient and more problematic expanding-ring (TTL) searches.
   One example of how expanding-scope searches provide increased
   localization can be found in the Scoped Hybrid Automatic Repeat
   reQuest with Forward Error Correction (SHARQFEC) reliable multicast
   protocol [SHARQFEC].

   While expanding-ring searches use one multicast address and
   increasing TTLs, expanding-scope searches involve changing the
   multicast addresses for each attempt at a different scope. For well-
   known services, these addresses can be obtained by applying an IANA-
   assigned offset from the top of the scope's address range.
   Applications, on the other hand, generally require the use of
   dynamically allocated addresses with offsets that will most likely
   vary from scope to scope.

   SADP builds upon the Multicast Address Allocation Architecture [MAAA]
   by adding a new application-layer service that allows applications to
   discover the relevant multicast address(es) associated with a session
   at each level in a hierarchy of scopes.

   SADP does not provide the means to allocate an address should one not
   be present for a session in a particular zone. In this case the
   application should take steps to obtain an address for that scope and
   then announce it to other application instances that join that scope
   at a later time. One proposed mechanism for allocating addresses is
   the Multicast Address Dynamic Allocation Protocol (MADCAP) [RFC2730].

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", SHALL NOT",
"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this
document are to be interpreted as described in RFC 2119 [RFC2119].

2. Overview

Administrative scoping affords the ability to create network
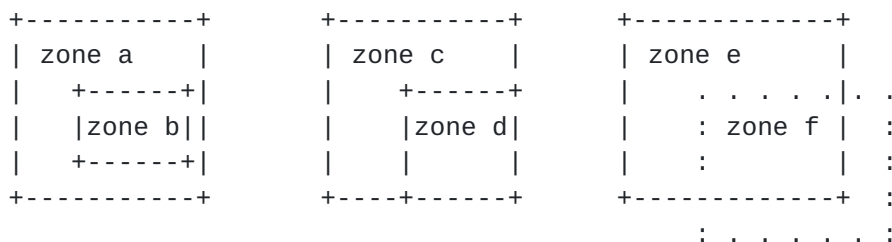partitions or zones in which multicast traffic addressed to one of a
block of addresses assigned to that zone will be limited to that
zone. The boundary of the zone is enforced by Zone Border Routers
(ZBRs) that reside at the edges of the zone. ZBRs must be carefully
configured so that traffic addressed within the zone does not pass
outside the zone. Ensuring consistency among boundary routers can be
a non-trivial task, and hence the Multicast Zone Announcement
Protocol (MZAP) [RFC2776], which is used to announce the existence of
zones, also provides the mechanisms to detect ZBR misconfigurations.

```
            .  .  .  .  .  .  .  .    +B+------>
             .                    . /
            .                      *
            .             <---+A*--------+C+--->
             .                 + .
              .               /  .
               .   Zone X <---   .
                .  .  .  .  .  ...
```
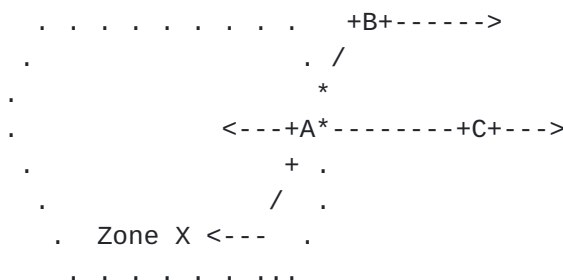
  A, B, C - Routers * - border interface   + - interface . - border

           Figure 1: Admin scope zone border example

Zones may be of different sizes and can also overlap. In addition to
the services of zone announcement and fault detection, MZAP also
provides mechanisms for determining and announcing the existence of
zones that nest inside others as shown in Figure 2.

```
   +-----------+        +-----------+        +-------------+
   | zone a    |        | zone c    |        | zone e      |
   |    +------+|        |    +------+        |    . . . . .|. .
   |    |zone b||        |    |zone d|        |    : zone f |   :
   |    +------+|        |    |      |        |    :        |   :
   +-----------+        +----+------+        +-------------+   :
                                                  : . . . . . :
```

    (a) "Contained"   (b) "Common Border" (c) "Overlap"
         zone b nests   zone d nests   zones e and f
         inside zone a   inside zone c   do not nest

             Figure 2: Zone nesting examples

This feature allows admin scope zones to be arranged in a hierarchy
as shown in Figure 3. The ability to nest admin scope zones in
hierarchies like that shown in Figure 3 is useful since it affords
localization through expanding-scope searches. For example, consider
a distributed application with session members distributed evenly
through out zone a. A session member in scope e, would perform a
search by multicasting a query within scope e, and if unsuccessful,
expand the scope to search in scope b, and eventually scope a if so
needed.

```
     . . . . .  . . . . . . . . . . . .
    .              scope a            .        Scope Boundaries
   .                                   .         . = scope  a
  .  _____      _____ .      - = scopes b,c
  . /     scope b     \    /  scope c       \ .    # = scopes d,e,f, & g
 .|                   |  |                 |.
 .|   #####     ##### |  |  #####    ##### |.
 .| #scope#  #scope#| | #scope#  #scope# |.
 .\ # d  #  # e  # |  | # f  #   #  g # /.
  .\ ####     #####/     ####     #### /.
   ._____/       _____/.
    . . . . . . . . . . . . . . . . .
```
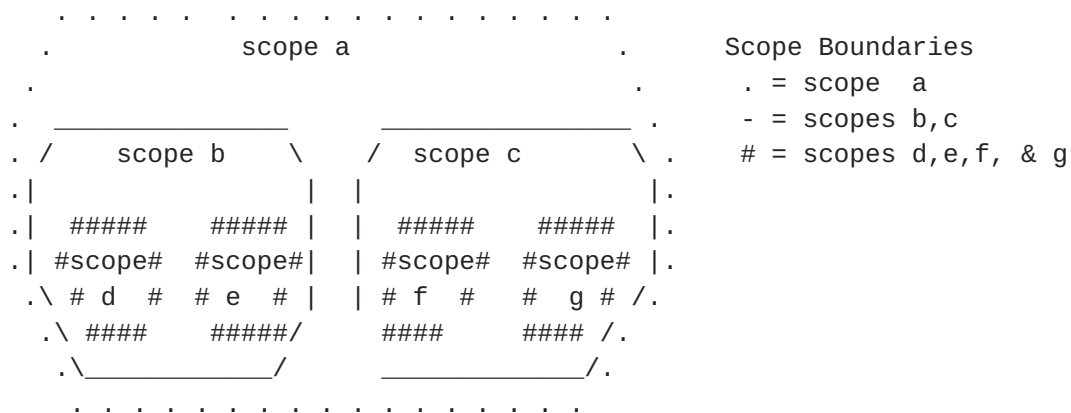
        Figure 3 : Admin Scope Zone Nesting Hierarchy example

In order for expanding-scope searches to be feasible, session members
must be able to determine two things:

o which scopes are involved in the hierarchy for a particular
  session.

o what address(es) are to be used for communicating with other
  session members within these scopes.

SADP affords the ability to discover this information by using a
single multicast group inside each scope [SADP-RELATIVE-GROUP] for
communication between SADP servers (see section 3.2) and the members
of various sessions. New members to a session use the channels
provided by the addresses to query existing SADP servers and session
members as to which specific scopes are valid and which scopes to
use. Since there is only one multicast address used per admin scope
zone for this purpose, members of a particular session will ignore
traffic intended for members of another session.

3. **Usage**

   In this section we summarize how session members can use SADP to
   determine which admin zones are used by the session's hierarchy and
   also the address(es) within these zones that are used by the current
   session members should such addresses exist.

3.1 **Session Identifiers**

   Each session that uses administrative scoping will be identified by a
   Session Identifier (SID) that corresponds to the address of the group
   used in the largest scope zone. Applications that require multiple
   addresses shall be decomposed into multiple individual sessions which
   will then be treated independently.

3.2 **Session Member Operation**

   Several predefined administrative scopes already exist [RFC2365]:

   o Link Local: Traffic is only carried across one physical link.

   o Local: Traffic is restricted to a specific network region.

   o Global: The entire multicast enabled network.

   By definition Link Local scopes nest inside Local scopes which in
   turn nest inside the Global Scope. Other scopes may exist between the
   local and global scopes. These scopes are constructed by the union of
   the admin scope zones that correspond to two or more topologically
   adjacent local scopes and are announced to routers within their
   confines using MZAP [RFC2776].

   The general algorithm that new members to a session should use to
   determine which scopes and addresses are involved in the hierarchy
   for a particular session is as follows:

   1) Determine largest scope, and address for the largest scope for
      the session. (this task is beyond the scope of this document,
      but can be assumed to involve some kind of out-of-band
      communication.)

   2) Starting with the SADP group [SADP-RELATIVE-GROUP] for the
      local scope, issue a SADP Request (SADP_REQ) message
      containing the SID address.

   3) Wait for a response on the SADP [SADP-RELATIVE-GROUP] address
      for at least [SADP-REQ-TIMEOUT] seconds. If no response is

       heard, wait for some small amount of time, and then
       repeat the request at the same scope.

    4) If after a total of 2 attempts at a given scope, no response
       has been received, increase the scope to the next largest scope
       and repeat steps 2 and 3. In cases where there are two
       non-nesting scopes larger than the current, try one scope and
       then the other, should the first scope not result in a reply.

    5) Continue steps 2), 3), and 4) until the largest scope has been
       queried or a response has been heard.

    In cases where the scope must be increased in order to find a session
    member that can reply, the new session member MAY decide to add
    levels to the hierarchy in order to increase localization for future
    session members. New session members that decide to take this step
    will use the existing addresses as discovered using SADP and request
    new ones. (e.g., via MADCAP [RFC2730]). Upon the successful
    allocation of a new address for use in the hierarchy, the new session
    member shall announce the new address via a SADP_NEW_ADDR message to
    the [SADP-RELATIVE-GROUP] address for the scope in which the address
    was allocated. This will cause the address to be cached by any SADP
    servers within the new address's scope.

    SADP servers and existing session members, upon hearing an SADP_REQ
    message from a new session member from [SADP-RELATIVE-GROUP] at a
    particular scope will issue an SADP Response (SADP_RESP) to the
    [SADP-RELATIVE-GROUP] at the same scope after waiting for a random
    amount of time (T) that is calculated as follows [HAND98]:

    Choose a random value X from a uniform random interval [0:1]   Let
    C = 256   Set
    T = [SADP-SUPPRESSION-INTERVAL] log(C*X + 1) / log(C)

    Should a member receive a SADP_RESP before its timer it expires it
    SHALL suppress its own response. This method ensures that close to
    one session member will respond.

3.3 SADP Server Operation

    Were SADP to be deployed in a wide scale session with the members of
    various sessions to use SADP between each other it would quickly
    cause catastrophic congestion. The reason for this is that whenever a
    new node joined a sparsely populated session with a large maximum
    scope, it would inevitably end up sending SADP_REQs to every scope up
    until the largest scope. Thus the highly likely occurrence of having

a global and continental scopes combined with numerous sparse
sessions (probably on the order of 10,000 to 100,000) would quickly
cause SADP_REQ flooding at the continental scope.

To address this shortcoming SADP allows, and in fact encourages, the
deployment of SADP servers. These servers subscribe to the [SADP-
RELATIVE-GROUP] for each scope they are in and cache the SADP_RESP
messages they receive at each scope. Having cached and merged the
responses for sessions at various scopes, they can then respond to
SADP_REQs heard at lower scopes using the information heard at the
larger scope(s). Should a SADP server hear a SADP_REQ at some
intermediate scope it MUST NOT announce address information for
scopes smaller than one on which the SADP_REQ was received.

The effect of allowing larger-scoped information to be announced at
lower scopes by SADP servers significantly reduces the number of
scopes a new session will have to query. New session members now need
only expand the scope until a SADP server is found. This is a marked
improvement over the case where no SADP servers exist and the search
must continue until an existing session member is found. An analysis
of how the pressence of SADP Servers improve SADP protcol performance
can be found in [KT99].

```
              Scope b Boundary
    Scope a         :  Scope a and Scope b
  _____        :      _____                _____
 /          \       :     /            \              /            \
 |Source at| _____:___\ |SADP Server | /_____ | New Session |
 |Scope a  | SADP_RESP/ | Scopes a,b | \ SADP_REQ    | Member      |
 _____/       :     _____/ _____\ | Scopes a,b  |
                   :                      SADP_RESP/ _____/
                   :
      Figure 4 : SADP Server acting as proxy session member
```
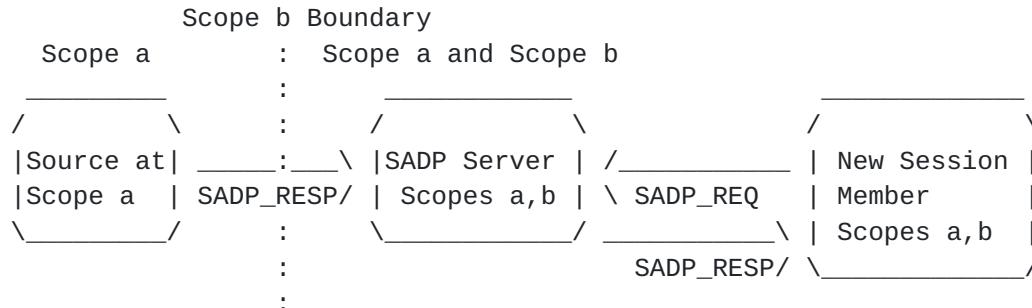
4. Packet Formats

All SADP messages are sent over UDP, with a destination port of
[SADP-PORT]. The common SADP message header (which follows the UDP
header), is shown below,

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    Version    |     PTYPE     |  Num Scopes  |  Addr Family   |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                      Session ID Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  +                                                             +
  |                    Authentication Block                     |
  +                                                             +
  |                                                             |
  +                                                             +
  |                                                             |
  +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Version: 8 bits
   The version defined in this document is version 0.

Packet Type (PTYPE): 8 bits
   The packet types defined in this document are:
       0: SADP Request
       1: SADP Response
       2: SADP New Address

Number of Scope Entries (Num Scopes) : 4 bits
   The number of scope entries present within a SADP_RESP
   message. This field should be set to zero for SADP_REQ messages.

Address Family (AddrFam): 4 bits
   This indicates the IANA-assigned address family number to be
   used for address contained in this message. Currently assigned
   values are listed in [RFC1700]. The values for IP
   addresses are:
       IPv4: 1
       IPv6: 2

Session ID Address: 32 bits (IPv4) or 128 bits (IPv6)
   The group address corresponding to the largest scope for this
   hierarchy of addresses.

Authentication Block:
   The Authentication Block provides information which can be used
   to confirm that the sender of the SADP message is a valid member
   of the session. Session Members that cannot confirm that the
   sender of a SADP Request Message is a session member or a
   known SADP Server MAY ignore it, while new session members that
   receive a SADP Response Message MUST ignore it.

   The authentication block consists of an MD5 digest that is
   constructed by applying the MD5 algorihtm [RFC1321] to these
   items in the following order:

     1) Session Identifer Address
     2) Address of the Session Member making the announcement
     3) The contents of any subsequent SADP Response or SADP New
        Address message data.

4.1 SADP Request

SADP Request (SADP_REQ) Messages have PTYPE=0, and are sent by new
session members that wish to learn which administrative scopes and
multicast addresses to use within a particular session. SADP_REQ
Messages are sent according to the algorithm described in 3.2.

4.2 SADP Response

The SADP Response (SADP_RESP) Message has PTYPE=1, and is sent in
response to a SADP_REQ Message to the same scope from which the
SADP_REQ was received. It contains the address that is to be used by
an application instance within a session for each scope that nests
within the scope to which the SADP_REQ was sent. (N.B. This includes
the scope to which the SADP_REQ was sent) Session members that
transmit SADP Response Messages MUST NOT include scope and address
information for scopes that are known to overlap or be larger than
that of the scope upon which the triggering SADP_REQ Message was

The format for a SADP Response Message is shown below:

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                         SADP Header
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                   Scope Start Address 1                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Scope 1 Session Address                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
                         . . . . . . .
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                   Scope Start Address N                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                   Scope N Session Address                     |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

Scope X Start Address : 32 bits (IPv4) or 128 bits (IPv6)
    The smallest address for the block of multicast addresses
    associated with a scope. If a scope X is valid for the range
    239.128.0.0 to 239.128.255.255, this field will be set to
    239.128.0.0.

    Scope X Session Address : 32 bits (IPv4) or 128 bits (IPv6)
        Address to be used for the named scope.

4.3 SADP New Address

    The SADP New Address (SADP_NEW_ADDR) Message has PTYPE=2. It is
    transmitted by session members that have attempted to find an address
    for a particular scope, failed, and have then subsequently allocated
    a new address for use in the session at that scope. Its purpose is
    to inform other members of the session of the existence of this newly
    allocated address and its availability for subsequent use.

    Should two members attempt to announce a new address to the same
    scope at the same time, their SADP_NEW_ADDR messages will result in a
    collision. SADP_NEW_ADDR collisions are resolved by the session
    members picking the lower of the two addresses.

```
 0                   1                   2                   3
 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
                          SADP Header
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
|                     Scope Start Address                       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|                  New Scope Session Address                    |
+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+=+
```

    Scope Start Address : 32 bits (IPv4) or 128 bits (IPv6)
        The smallest address for the block of multicast addresses
        associated with a scope. If a scope X is valid for the range
        239.128.0.0 to 239.128.255.255, this field will be set to
        239.128.0.0.

    New Scope Session Address : 32 bits (IPv4) or 128 bits (IPv6)
        Address of the newly allocated group to be used for the
        specified scope.

5. Constants

    [SADP-RELATIVE-GROUP]: The relative group with each scope, to which
    session members send SADP Requests and Responses. All application
    instances that use SADP for constructing hierarchies of scopes MUST
    subscribe to this address for each scope which nests within the
    session scope, in order to ensure that each application instance uses
    the hierarchy in the most efficient manner.

[SADP-REQ-TIMEOUT]: The time after which a session member that sends
SADP Request should wait before concluding that no session members
are present at the current scope. Default value is 3 seconds.

[SADP-SUPPRESSION-INTERVAL]: The interval over which a session member
chooses a random delay before responding to SADP Request. Default
value 2 seconds.

6. Security Considerations

SADP employs distributed mechanisms to allow new session members to
learn of the existence of session-specific admin scoped multicast
address. This fact lays SADP open to attack by malicious hosts that
could potentially mis-inform new session members of incorrect
addresses, thereby affecting a man-in-the-middle attack.

To prevent attacks of this nature by non-session members from
occurring all SADP messages are signed by the sender. However, this
measure does not prevent malicious hosts from joining a session and
then performing the same attack. Hence, SADP's security depends upon
a suitable gating process for new-member admittance combined with (as
yet to be determined) mechanisms that allow spoofed SADP messages to
be identified for removal before processing.

7. Acknowledgments

The Authors would like to acknowledge Mark Handley for the helpful
discussions and feedback which helped shape and refine this document.

8. References

[MAAA]   Handley, M., Thaler, D., and D. Estrin, "The Internet
            Multicast Address Allocation Architecture", Internet
            Draft, draft-ietf-malloc-arch-**.txt, June 2000.

[HAND98]   Handley, M., "Session directories and scalable internet
            multicast address allocation for private internets",
            In Proceedings of ACM SIGCOMM, pp 105-116, 1998.

[SHARQFEC] Kermode, R., "Scoped Hybrid Automatic Repeat reQuest with
            Forward Error Correction (SHARQFEC)", ACM SIGCOMM98,
            Vancouver Canada, September 1998.

[NESTOPT]   Kermode, R., "MADCAP Multicast Scope Nesting State
            Option", draft-ietf-malloc-madcap-nest-opt-05.txt,
            April, 2000.

   [KT99]   Kermode, R., & Thaler, D., "Support for reliable Sessions
              with a Large Number of Members", First International
              Workshop on Networked Group Communication, Pisa Italy,
              November 1999

   [RFC1321] Rivest, R. and S. Dusse, "The MD5 Message-Digest
              Algorithm", RFC 1321, April 1992.

   [RFC1700] Reynolds, J., Postel, J., "Assigned Numbers", RFC 1700,
              October 1994.

   [RFC1884] Hinden, R., Deering, S., "IP Version 6 Addressing
              Architecture", RFC 1884, December 1995.

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

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

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

   [RFC2776] Handley, M., Thaler, D., "Multicast-Scope Zone
              Announcement Protocol (MZAP)", RFC 2776, February 2000

9. Authors' Addresses

   Roger Kermode
   Motorola Australia Research Centre
   12 Lord St.
   Botany, NSW 2019
   Australia
   Email: Roger.Kermode@motorola.com

   David Thaler
   Microsoft
   One Microsoft Way
   Redmond, WA 98052
   USA
   Email: dthaler@microsoft.com

10. Full Copyright Statement