                        SBM (Subnet Bandwidth Manager):
A Protocol for RSVP-based Admission Control over IEEE 802-style networks

Status of this Memo

   This document specifies an Internet standards track protocol for the
   Internet community, and requests discussion and suggestions for
   improvements. Please refer to the current edition of the "Internet
   Official Protocol Standards" (STD 1) for the standardization state
   and status of this protocol. Distribution of this memo is unlimited.

Copyright Notice

Abstract

   This document describes a signaling method and protocol for RSVP-
   based admission control over IEEE 802-style LANs. The protocol is
   designed to work both with the current generation of IEEE 802 LANs as
   well as with the recent work completed by the IEEE 802.1 committee.

1. Introduction

   New extensions to the Internet architecture and service models have
   been defined for an integrated services Internet [RFC-1633, RFC-2205,
   RFC-2210] so that applications can request specific qualities or
   levels of service from an internetwork in addition to the current IP
   best-effort service. These extensions include RSVP, a resource
   reservation setup protocol, and definition of new service classes to
   be supported by Integrated Services routers. RSVP and service class
   definitions are largely independent of the underlying networking
   technologies and it is necessary to define the mapping of RSVP and
   Integrated Services specifications onto specific subnetwork
   technologies. For example, a definition of service mappings and

reservation setup protocols is needed for specific link-layer
technologies such as shared and switched IEEE-802-style LAN
technologies.

This document defines SBM, a signaling protocol for RSVP-based
admission control over IEEE 802-style networks. SBM provides a
method for mapping an internet-level setup protocol such as RSVP onto
IEEE 802 style networks. In particular, it describes the operation
of RSVP-enabled hosts/routers and link layer devices (switches,
bridges) to support reservation of LAN resources for RSVP-enabled
data flows. A framework for providing Integrated Services over
shared and switched IEEE-802-style LAN technologies and a definition
of service mappings have been described in separate documents [RFC-
FRAME, RFC-MAP].

2. Goals and Assumptions

   The SBM (Subnet Bandwidth Manager) protocol and its use for admission
   control and bandwidth management in IEEE 802 level-2 networks is
   based on the following architectural goals and assumptions:

      I. Even though the current trend is towards increased use of
      switched LAN topologies consisting of newer switches that support
      the priority queuing mechanisms specified by IEEE 802.1p, we
      assume that the LAN technologies will continue to be a mix of
      legacy shared/ switched LAN segments and newer switched segments
      based on IEEE 802.1p specification. Therefore, we specify a
      signaling protocol for managing bandwidth over both legacy and
      newer LAN topologies and that takes advantage of the additional
      functionality (such as an explicit support for different traffic
      classes or integrated service classes) as it becomes available in
      the new generation of switches, hubs, or bridges. As a result,
      the SBM protocol would allow for a range of LAN bandwidth
      management solutions that vary from one that exercises purely
      administrative control (over the amount of bandwidth consumed by
      RSVP-enabled traffic flows) to one that requires cooperation (and
      enforcement) from all the end-systems or switches in a IEEE 802
      LAN.

      II. This document specifies only a signaling method and protocol
      for LAN-based admission control over RSVP flows. We do not define
      here any traffic control mechanisms for the link layer; the
      protocol is designed to use any such mechanisms defined by IEEE
      802. In addition, we assume that the Layer 3 end-systems (e.g., a
      host or a router) will exercise traffic control by policing
      Integrated Services traffic flows to ensure that each flow stays
      within its traffic specifications stipulated in an earlier
      reservation request submitted for admission control. This then

allows a system using SBM admission control combined with per flow
shaping at end systems and IEEE-defined traffic control at link
layer to realize some approximation of Controlled Load (and even
Guaranteed) services over IEEE 802-style LANs.

III. In the absence of any link-layer traffic control or priority
queuing mechanisms in the underlying LAN (such as a shared LAN
segment), the SBM-based admission control mechanism only limits
the total amount of traffic load imposed by RSVP-enabled flows on
a shared LAN. In such an environment, no traffic flow separation
mechanism exists to protect the RSVP-enabled flows from the best-
effort traffic on the same shared media and that raises the
question of the utility of such a mechanism outside a topology
consisting only of 802.1p-compliant switches. However, we assume
that the SBM-based admission control mechanism will still serve a
useful purpose in a legacy, shared LAN topology for two reasons.
First, assuming that all the nodes that generate Integrated
Services traffic flows utilize the SBM-based admission control
procedure to request reservation of resources before sending any
traffic, the mechanism will restrict the total amount of traffic
generated by Integrated Services flows within the bounds desired
by a LAN administrator (see discussion of the NonResvSendLimit
parameter in Appendix C). Second, the best-effort traffic
generated by the TCP/IP-based traffic sources is generally rate
adaptive (using a TCP-style "slow start" congestion avoidance
mechanism or a feedback-based rate adaptation mechanism used by
audio/video streams based on RTP/RTCP protocols) and adapts to
stay within the available network bandwidth. Thus, the
combination of admission control and rate adaptation should avoid
persistent traffic congestion. This does not, however, guarantee
that non-Integrated-Services traffic will not interfere with the
Integrated Services traffic in the absence of traffic control
support in the underlying LAN infrastructure.

3. Organization of the rest of this document

   The rest of this document provides a detailed description of the
   SBM-based admission control procedure(s) for IEEE 802 LAN
   technologies. The document is organized as follows:

   * Section 4 first defines the various terms used in the document and
      then provides an overview of the admission control procedure with
      an example of its application to a sample network.

   * Section 5 describes the rules for processing and forwarding PATH
      (and PATH_TEAR) messages at DSBMs (Designated Subnet Bandwidth
      Managers), SBMs, and DSBM clients.

   * Section 6 addresses the inter-operability issues when a DSBM may
     operate in the absence of RSVP signaling at Layer 3 or when
     another signaling protocol (such as SNMP) is used to reserve
     resources on a LAN segment.

   * Appendix A describes the details of the DSBM election algorithm
     used for electing a designated SBM on a LAN segment when more than
     one SBM is present. It also describes how DSBM clients discover
     the presence of a DSBM on a managed segment.

   * Appendix B specifies the formats of SBM-specific messages used and
     the formats of new RSVP objects needed for the SBM operation.

   * Appendix C describes usage of the DSBM to distribute configuration
     information to senders on a managed segment.

4. Overview

4.1. Definitions

   - Link Layer or Layer 2 or L2: We refer to data-link layer
     technologies such as IEEE 802.3/Ethernet as L2 or layer 2.

   - Link Layer Domain or Layer 2 domain or L2 domain: a set of nodes
     and links interconnected without passing through a L3 forwarding
     function. One or more IP subnets can be overlaid on a L2 domain.

   - Layer 2 or L2 devices: We refer to devices that only implement
     Layer 2 functionality as Layer 2 or L2 devices. These include
     802.1D bridges or switches.

   - Internetwork Layer or Layer 3 or L3: Layer 3 of the ISO 7 layer
     model. This document is primarily concerned with networks that use
     the Internet Protocol (IP) at this layer.

   - Layer 3 Device or L3 Device or End-Station: these include hosts
     and routers that use L3 and higher layer protocols or application
     programs that need to make resource reservations.

   - Segment: A L2 physical segment that is shared by one or more
     senders. Examples of segments include (a) a shared Ethernet or
     Token-Ring wire resolving contention for media access using CSMA
     or token passing ("shared L2 segment"), (b) a half duplex link
     between two stations or switches, (c) one direction of a switched
     full-duplex link.

-   Managed segment: A managed segment is a segment with a DSBM
    present and responsible for exercising admission control over
    requests for resource reservation. A managed segment includes
    those interconnected parts of a shared LAN that are not separated
    by DSBMs.

-   Traffic Class: An aggregation of data flows which are given
    similar service within a switched network.

-   User_priority: User_priority is a value associated with the
    transmission and reception of all frames in the IEEE 802 service
    model: it is supplied by the sender that is using the MAC service.
    It is provided along with the data to a receiver using the MAC
    service. It may or may not be actually carried over the network:
    Token-Ring/802.5 carries this value (encoded in its FC octet),
    basic Ethernet/802.3 does not, 802.12 may or may not depending on
    the frame format in use. 802.1p defines a consistent way to carry
    this value over the bridged network on Ethernet, Token Ring,
    Demand-Priority, FDDI or other MAC-layer media using an extended
    frame format. The usage of user_priority is fully described in
    section 2.5 of 802.1D [IEEE8021D] and 802.1p [IEEE8021P] "Support
    of the Internal Layer Service by Specific MAC Procedures".

-   Subnet: used in this memo to indicate a group of L3 devices
    sharing a common L3 network address prefix along with the set of
    segments making up the L2 domain in which they are located.

-   Bridge/Switch: a layer 2 forwarding device as defined by IEEE
    802.1D. The terms bridge and switch are used synonymously in this
    document.

-   DSBM: Designated SBM (DSBM) is a protocol entity that resides in a
    L2 or L3 device and manages resources on a L2 segment. At most one
    DSBM exists for each L2 segment.

-   SBM: the SBM is a protocol entity that resides in a L2 or L3
    device and is capable of managing resources on a segment. However,
    only a DSBM manages the resources for a managed segment. When more
    than one SBM exists on a segment, one of the SBMs is elected to be
    the DSBM.

-   Extended segment: An extended segment includes those parts of a
    network which are members of the same IP subnet and therefore are
    not separated by any layer 3 devices. Several managed segments,
    interconnected by layer 2 devices, constitute an extended segment.

- Managed L2 domain: An L2 domain consisting of managed segments is
    referred to as a managed L2 domain to distinguish it from a L2
    domain with no DSBMs present for exercising admission control over
    resources at segments in the L2 domain.

- DSBM clients: These are entities that transmit traffic onto a
    managed segment and use the services of a DSBM for the managed
    segment for admission control over a LAN segment. Only the layer 3
    or higher layer entities on L3 devices such as hosts and routers
    are expected to send traffic that requires resource reservations,
    and, therefore, DSBM clients are L3 entities.

- SBM transparent devices: A "SBM transparent" device is unaware of
    SBMs or DSBMs (though it may or may not be RSVP aware) and,
    therefore, does not participate in the SBM-based admission control
    procedure over a managed segment. Such a device uses standard
    forwarding rules appropriate for the device and is transparent
    with respect to SBM. An example of such a L2 device is a legacy
    switch that does not participate in resource reservation.

- Layer 3 and layer 2 addresses: We refer to layer 3 addresses of
    L3/L2 devices as "L3 addresses" and layer 2 addresses as "L2
    addresses". This convention will be used in the rest of the
    document to distinguish between Layer 3 and layer 2 addresses used
    to refer to RSVP next hop (NHOP) and previous hop (PHOP) devices.
    For example, in conventional RSVP message processing, RSVP_HOP
    object in a PATH message carries the L3 address of the previous
    hop device. We will refer to the address contained in the RSVP_HOP
    object as the RSVP_HOP_L3 address and the corresponding MAC
    address of the previous hop device will be referred to as the
    RSVP_HOP_L2 address.

4.2. Overview of the SBM-based Admission Control Procedure

   A protocol entity called "Designated SBM" (DSBM) exists for each
   managed segment and is responsible for admission control over the
   resource reservation requests originating from the DSBM clients in
   that segment. Given a segment, one or more SBMs may exist on the
   segment. For example, many SBM-capable devices may be attached to a
   shared L2 segment whereas two SBM-capable switches may share a half-
   duplex switched segment. In that case, a single DSBM is elected for
   the segment. The procedure for dynamically electing the DSBM is
   described in Appendix A. The only other approved method for
   specifying a DSBM for a managed segment is static configuration at
   SBM-capable devices.

The presence of a DSBM makes the segment a "managed segment".
Sometimes, two or more L2 segments may be interconnected by SBM
transparent devices. In that case, a single DSBM will manage the
resources for those segments treating the collection of such segments
as a single managed segment for the purpose of admission control.

4.2.1. Basic Algorithm

Figure 1 - An Example of a Managed Segment.

```
     +-------+        +-----+        +------+     +-----+     +--------+
     |Router |        | Host|        | DSBM |     | Host|     | Router |
     | R2    |        | C   |        +------+     | B   |     | R3     |
     +-------+        +-----+        /            +-----+     +--------+
         |               |         /                 |            |
         |               |        /                  |            |
     ================================================================LAN
             |                                          |
             |                                          |
         +------+                                  +-------+
         | Host |                                  | Router|
         |  A   |                                  | R1    |
         +------+                                  +-------+
```
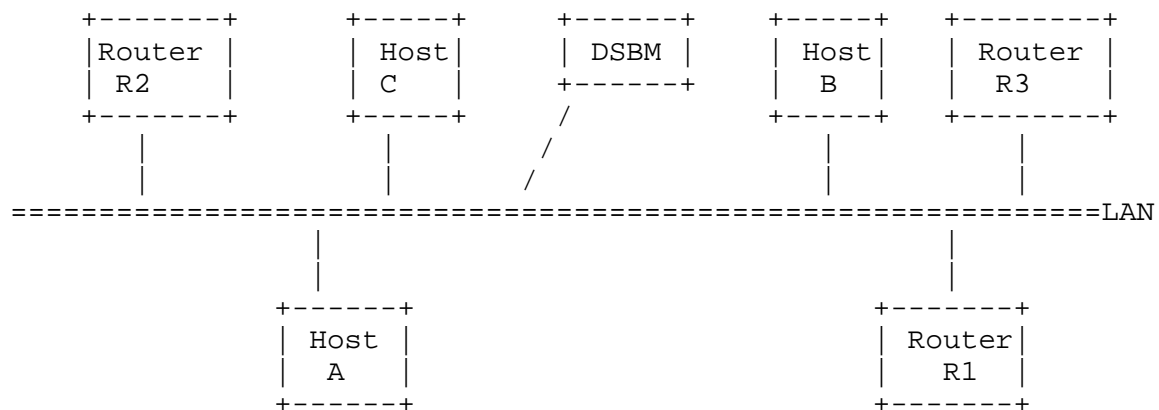
Figure 1 shows an example of a managed segment in a L2 domain that
interconnects a set of hosts and routers. For the purpose of this
discussion, we ignore the actual physical topology of the L2 domain
(assume it is a shared L2 segment and a single managed segment
represents the entire L2 domain). A single SBM device is designated
to be the DSBM for the managed segment. We will provide examples of
operation of the DSBM over switched and shared segments later in the
document.

The basic DSBM-based admission control procedure works as follows:

1. DSBM Initialization: As part of its initial configuration, DSBM
    obtains information such as the limits on fraction of available
    resources that can be reserved on each managed segment under its
    control. For instance, bandwidth is one such resource. Even
    though methods such as auto-negotiation of link speeds and
    knowledge of link topology allow discovery of link capacity, the
    configuration may be necessary to limit the fraction of link
    capacity that can be reserved on a link. Configuration is likely
    to be static with the current L2/L3 devices. Future work may
    allow for dynamic discovery of this information. This document
    does not specify the configuration mechanism.

2. DSBM Client Initialization: For each interface attached, a DSBM
   client determines whether a DSBM exists on the interface. The
   procedure for discovering and verifying the existence of the DSBM
   for an attached segment is described in Appendix A. If the client
   itself is capable of serving as the DSBM on the segment, it may
   choose to participate in the election to become the DSBM. At the
   start, a DSBM client first verifies that a DSBM exists in its L2
   domain so that it can communicate with the DSBM for admission
   control purposes.

   In the case of a full-duplex segment, an election may not be
   necessary as the SBM at each end will typically act as the DSBM
   for outgoing traffic in each direction.

3. DSBM-based Admission Control: To request reservation of resources
   (e.g., LAN bandwidth in a L2 domain), DSBM clients (RSVP-capable
   L3 devices such as hosts and routers) follow the following steps:

   a) When a DSBM client sends or forwards a RSVP PATH message over
      an interface attached to a managed segment, it sends the PATH
      message to the segment's DSBM instead of sending it to the RSVP
      session destination address (as is done in conventional RSVP
      processing). After processing (and possibly updating an
      ADSPEC), the DSBM will forward the PATH message toward its
      destination address. As part of its processing, the DSBM builds
      and maintains a PATH state for the session and notes the
      previous L2/L3 hop that sent it the PATH message.

      Let us consider the managed segment in Figure 1. Assume that a
      sender to a RSVP session (session address specifies the IP
      address of host A on the managed segment in Figure 1) resides
      outside the L2 domain of the managed segment and sends a PATH
      message that arrives at router R1 which is on the path towards
      host A.

      DSBM client on Router R1 forwards the PATH message from the
      sender to the DSBM. The DSBM processes the PATH message and
      forwards the PATH message towards the RSVP receiver (Detailed
      message processing and forwarding rules are described in
      Section 5). In the process, the DSBM builds the PATH state,
      remembers the router R1 (its L2 and l3 addresses) as the
      previous hop for the session, puts its own L2 and L3 addresses
      in the PHOP objects (see explanation later), and effectively
      inserts itself as an intermediate node between the sender (or
      R1 in Figure 1) and the receiver (host A) on the managed
      segment.

b) When an application on host A wishes to make a reservation for
   the RSVP session, host A follows the standard RSVP message
   processing rules and sends a RSVP RESV message to the previous
   hop L2/L3 address (the DSBMs address) obtained from the PHOP
   object(s) in the previously received PATH message.

c) The DSBM processes the RSVP RESV message based on the bandwidth
   available and returns an RESV_ERR message to the requester
   (host A) if the request cannot be granted. If sufficient
   resources are available and the reservation request is granted,
   the DSBM forwards the RESV message towards the PHOP(s) based on
   its local PATH state for the session. The DSBM merges
   reservation requests for the same session as and when possible
   using the rules similar to those used in the conventional RSVP
   processing (except for an additional criterion described in
   Section 5.8).

d) If the L2 domain contains more than one managed segment, the
   requester (host A) and the forwarder (router R1) may be
   separated by more than one managed segment. In that case, the
   original PATH message would propagate through many DSBMs (one
   for each managed segment on the path from R1 to A) setting up
   PATH state at each DSBM. Therefore, the RESV message would
   propagate hop-by-hop in reverse through the intermediate DSBMs
   and eventually reach the original forwarder (router R1) on the
   L2 domain if admission control at all DSBMs succeeds.

4.2.2. Enhancements to the conventional RSVP operation

   (D)SBMs and DSBM clients implement minor additions to the standard
   RSVP protocol. These are summarized in this section. A detailed
   description of the message processing and forwarding rules follows in
   section 5.

4.2.2.1 Sending PATH Messages to the DSBM on a Managed Segment

   Normal RSVP forwarding rules apply at a DSBM client when it is not
   forwarding an outgoing PATH message over a managed segment. However,
   outgoing PATH messages on a managed segment are sent to the DSBM for
   the corresponding managed segment (Section 5.2 describes how the PATH
   messages are sent to the DSBM on a managed segment).

4.2.2.2 The LAN_NHOP Objects

   In conventional RSVP processing over point-to-point links, RSVP nodes
   (hosts/routers) use RSVP_HOP object (NHOP and PHOP info) to keep
   track of the next hop (downstream node in the path of data packets in
   a traffic flow) and the previous hop (upstream nodes with respect to

the data flow) nodes on the path between a sender and a receiver.
Routers along the path of a PATH message forward the message towards
the destination address based on the L3 routing (packet forwarding)
tables.

For example, consider the L2 domain in Figure 1. Assume that both the
sender (some host X) and the receiver (some host Y) in a RSVP session
reside outside the L2 domain shown in the Figure, but PATH messages
from the sender to its receiver pass through the routers in the L2
domain using it as a transit subnet. Assume that the PATH message
from the sender X arrives at the router R1. R1 uses its local routing
information to decide which next hop router (either router R2 or
router R3) to use to forward the PATH message towards host Y.
However, when the path traverses a managed L2 domain, we require the
PATH and RESV messages to go through a DSBM for each managed segment.
Such a L2 domain may span many managed segments (and DSBMs) and,
typically, SBM protocol entities on L2 devices (such as a switch)
will serve as the DSBMs for the managed segments in a switched
topology. When R1 forwards the PATH message to the DSBM (an L2
device), the DSBM may not have the L3 routing information necessary
to select the egress router (between R2 and R3) before forwarding the
PATH message. To ensure correct operation and routing of RSVP
messages, we must provide additional forwarding information to DSBMs.

For this purpose, we introduce new RSVP objects called LAN_NHOP
address objects that keep track of the next L3 hop as the PATH
message traverses an L2 domain between two L3 entities (RSVP PHOP and
NHOP nodes).

4.2.2.3 Including Both Layer-2 and Layer-3 Addresses in the LAN_NHOP

When a DSBM client (a host or a router acting as the originator of a
PATH message) sends out a PATH message to the DSBM, it must include
LAN_NHOP information in the message. In the case of a unicast
destination, the LAN_NHOP address specifies the destination address
(if the destination is local to its L2 domain) or the address of the
next hop router towards the destination. In our example of an RSVP
session involving the sender X and receiver Y with L2 domain in
Figure 1 acting as the transit subnet, R1 is the ingress node that
receives the PATH message. R1 first determines that R2 is the next
hop router (or the egress node in the L2 domain for the session
address) and then inserts a LAN_NHOP object that specifies R2's IP
address. When a DSBM receives a PATH message, it can now look at the
address in the LAN_NHOP object and forward the PATH message towards
the egress node after processing the PATH message. However, we
expect the L2 devices (such as switches) to act as DSBMs on the path
within the L2 domain and it may not be reasonable to expect these
devices to have an ARP capability to determine the MAC address (we

call it L2ADDR for Layer 2 address) corresponding to the IP address
in the LAN_NHOP object.

Therefore, we require that the LAN_NHOP information (generated by the
L3 device) include both the IP address (LAN_NHOP_L3 address) and the
corresponding MAC address (LAN_NHOP_L2 address) for the next L3 hop
over the L2 domain. The LAN_NHOP_L3 address is used by SBM protocol
entities on L3 devices to forward the PATH message towards its
destination whereas the L2 address is used by the SBM protocol
entities on L2 devices to determine how to forward the PATH message
towards the L3 NHOP (egress point from the L2 domain). The exact
format of the LAN_NHOP information and relevant objects is described
later in Appendix B.

4.2.2.4 Similarities to Standard RSVP Message Processing

   - When a DSBM receives a RSVP PATH message, it processes the PATH
      message according to the PATH processing rules described in the
      RSVP specification. In particular, the DSBM retrieves the IP
      address of the previous hop from the RSVP_HOP object in the PATH
      message and stores the PHOP address in its PATH state. It then
      forwards the PATH message with the PHOP (RSVP_HOP) object modified
      to reflect its own IP address (RSVP_HOP_L3 address). Thus, the
      DSBM inserts itself as an intermediate hop in the chain of nodes
      in the path between two L3 nodes across the L2 domain.

   - The PATH state in a DSBM is used for forwarding subsequent RESV
      messages as per the standard RSVP message processing rules. When
      the DSBM receives a RESV message, it processes the message and
      forwards it to appropriate PHOP(s) based on its PATH state.

   - Because a DSBM inserts itself as a hop between two RSVP nodes in
      the path of a RSVP flow, all RSVP related messages (such as PATH,
      PATH_TEAR, RESV, RESV_CONF, RESV_TEAR, and RESV_ERR) now flow
      through the DSBM. In particular, a PATH_TEAR message is routed
      exactly through the intermediate DSBM(s) as its corresponding PATH
      message and the local PATH state is first cleaned up at each
      intermediate hop before the PATH_TEAR message gets forwarded.

   - So far, we have described how the PATH message propagates through
      the L2 domain establishing PATH state at each DSBM along the
      managed segments in the path. The layer 2 address (LAN_NHOP_L2
      address) in the LAN_NHOP object should be used by the L2 devices
      along the path to decide how to forward the PATH message toward
      the next L3 hop. Such devices will apply the standard IEEE 802.1D
      forwarding rules (e.g., send it on a single port based on its
      filtering database, or flood it on all ports active in the
      spanning tree if the L2 address does not appear in the filtering

database) to the LAN_NHOP_L2 address as are applied normally to
data packets destined to the address.

4.2.2.5 Including Both Layer-2 and Layer-3 Addresses in the RSVP_HOP
        Objects

   In the conventional RSVP message processing, the PATH state
   established along the nodes on a path is used to route the RESV
   message from a receiver to a sender in an RSVP session. As each
   intermediate node builds the path state, it remembers the previous
   hop (stores the PHOP IP address available in the RSVP_HOP object of
   an incoming message) that sent it the PATH message and, when the RESV
   message arrives, the intermediate node simply uses the stored PHOP
   address to forward the RESV after processing it successfully.

   In our case, we expect the SBM entities residing at L2 devices to act
   as DSBMs (and, therefore, intermediate RSVP hops in an L2 domain)
   along the path between a sender (PHOP) and receiver (NHOP). Thus,
   when a RESV message arrives at a DSBM, it must use the stored PHOP IP
   address to forward the RESV message to its previous hop. However, it
   may not be reasonable to expect the L2 devices to have an ARP cache
   or the ARP capability to map the PHOP IP address to its corresponding
   L2 address before forwarding the RESV message.

   To obviate the need for such address mapping at L2 devices, we use a
   RSVP_HOP_L2 object in the PATH message. The RSVP_HOP_L2 object
   includes the Layer 2 address (L2ADDR) of the previous hop and
   complements the L3 address information included in the RSVP_HOP
   object (RSVP_HOP_L3 address).

   When a L3 device constructs and forwards a PATH message over a
   managed segment, it includes its IP address (IP address of the
   interface over which PATH is sent) in the RSVP_HOP object and adds a
   RSVP_HOP_L2 object that includes the corresponding L2 address for the
   interface. When a device in the L2 domain receives such a PATH
   message, it remembers the addresses in the RSVP_HOP and RSVP_HOP_L2
   objects in its PATH state and then overwrites the RSVP_HOP and
   RSVP_HOP_L2 objects with its own addresses before forwarding the PATH
   message over a managed segment.

   The exact format of RSVP_HOP_L2 object is specified in Appendix B.

4.2.2.6 Loop Detection

   When an RSVP session address is a multicast address and a SBM, DSBM,
   and DSBM clients share the same L2 segment (a shared segment), it is
   possible for a SBM or a DSBM client to receive one or more copies of
   a PATH message that it forwarded earlier when a DSBM on the same wire

forwards it (See Section 5.7 for an example of such a case). To
facilitate detection of such loops, we use a new RSVP object called
the LAN_LOOPBACK object. DSBM clients or SBMs (but not the DSBMs
reflecting a PATH message onto the interface over which it arrived
earlier) must overwrite (or add if the PATH message does NOT already
include a LAN_LOOPBACK object) the LAN_LOOPBACK object in the PATH
message with their own unicast IP address.

Now, a SBM or a DSBM client can easily detect and discard the
duplicates by checking the contents of the LAN_LOOPBACK object (a
duplicate PATH message will list a device's own interface address in
the LAN_LOOPBACK object). Appendix B specifies the exact format of
the LAN_LOOPBACK object.

4.2.2.7 802.1p, User Priority and TCLASS

The model proposed by the Integrated Services working group requires
isolation of traffic flows from each other during their transit
across a network. The motivation for traffic flow separation is to
provide Integrated Services flows protection from misbehaving flows
and other best-effort traffic that share the same path. The basic
IEEE 802.3/Ethernet networks do not provide any notion of traffic
classes to discriminate among different flows that request different
services. However, IEEE 802.1p defines a way for switches to
differentiate among several "user_priority" values encoded in packets
representing different traffic classes (see [IEEE802Q, IEEE8021p] for
further details). The user_priority values can be encoded either in
native LAN packets (e.g., in IEEE 802.5's FC octet) or by using an
encapsulation above the MAC layer (e.g., in the case of Ethernet, the
user_priority value assigned to each packet will be carried in the
frame header using the new, extended frame format defined by IEEE
802.1Q [IEEE8021Q]. IEEE, however, makes no recommendations about how
a sender or network should use the user_priority values. An
accompanying document makes recommendations on the usage of the
user_priority values (see [RFC-MAP] for details).

Under the Integrated Services model, L3 (or higher) entities that
transmit traffic flows onto a L2 segment should perform per-flow
policing to ensure that the flows do not exceed their traffic
specification as specified during admission control. In addition, L3
devices may label the frames in such flows with a user_priority value
to identify their service class.

For the purpose of this discussion, we will refer to the
user_priority value carried in the extended frame header as the
"traffic class" of a packet. Under the ISSLL model, the L3 entities,
that send traffic and that use the SBM protocol, may select the
appropriate traffic class of outgoing packets [RFC-MAP]. This

selection may be overridden by DSBM devices, in the following manner.
once a sender sends a PATH message, downstream DSBMs will insert a
new traffic class object (TCLASS object) in the PATH message that
travels to the next L3 device (L3 NHOP for the PATH message). To some
extent, the TCLASS object contents are treated like the ADSPEC object
in the RSVP PATH messages. The L3 device that receives the PATH
message must remove and store the TCLASS object as part of its PATH
state for the session. Later, when the same L3 device needs to
forward a RSVP RESV message towards the original sender, it must
include the TCLASS object in the RESV message. When the RESV message
arrives at the original sender, the sender must use the user_priority
value from the TCLASS object to override its selection for the
traffic class marked in outgoing packets.

The format of the TCLASS object is specified in Appendix B. Note
that TCLASS and other SBM-specific objects are carried in a RSVP
message in addition to all the other, normal RSVP objects per RFC
2205.

4.2.2.8 Processing the TCLASS Object

In summary, use of TCLASS objects requires following additions to the
conventional RSVP message processing at DSBMs, SBMs, and DSBM
clients:

*   When a DSBM receives a PATH message over a managed segment and the
    PATH message does not include a TCLASS object, the DSBM MAY add a
    TCLASS object to the PATH message before forwarding it. The DSBM
    determines the appropriate user_priority value for the TCLASS
    object. A mechanism for selecting the appropriate user_priority
    value is described in an accompanying document [RFC-MAP].

*   When SBM or DSBM receives a PATH message with a TCLASS object over
    a managed segment in a L2 domain and needs to forward it over a
    managed segment in the same L2 domain, it will store it in its
    path state and typically forward the message without changing the
    contents of the TCLASS object. However, if the DSBM/SBM cannot
    support the service class represented by the user_priority value
    specified by the TCLASS object in the PATH message, it may change
    the priority value in the TCLASS to a semantically "lower" service
    value to reflect its capability and store the changed TCLASS value
    in its path state.

[NOTE: An accompanying document defines the int-serv mappings over
IEEE 802 networks [RFC-MAP] provides a precise definition of
user_priority values and describes how the user_priority values
are compared to determine "lower" of the two values or the
"lowest" among all the user_priority values.]

   *   When a DSBM receives a RESV message with a TCLASS object, it may
       use the traffic class information (in addition to the usual
       flowspec information in the RSVP message) for its own admission
       control for the managed segment.

       Note that this document does not specify the actual algorithm or
       policy used for admission control. At one extreme, a DSBM may use
       per-flow reservation request as specified by the flowspec for a
       fine grain admission control. At the other extreme, a DSBM may
       only consider the traffic class information for a very coarse-
       grain admission control based on some static allocation of link
       capacity for each traffic class. Any combination of the options
       represented by these two extremes may also be used.

   *   When a DSBM (at an L2 or L3) device receives a RESV message
       without a TCLASS object and it needs to forward the RESV message
       over a managed segment within the same L2 domain, it should first
       check its path state and check whether it has stored a TCLASS
       value. If so, it should include the TCLASS object in the outgoing
       RESV message after performing its own admission control. If no
       TCLASS value is stored, it must forward the RESV message without
       inserting a TCLASS object.

   *   When a DSBM client (residing at an L3 device such as a host or an
       edge router) receives the TCLASS object in a PATH message that it
       accepts over an interface, it should store the TCLASS object as
       part of its PATH state for the interface. Later, when the client
       forwards a RESV message for the same session on the interface, the
       client must include the TCLASS object (unchanged from what was
       received in the previous PATH message) in the RESV message it
       forwards over the interface.

   *   When a DSBM client receives a TCLASS object in an incoming RESV
       message over a managed segment and local admission control
       succeeds for the session for the outgoing interface over the
       managed segment, the client must pass the user_priority value in
       the TCLASS object to its local packet classifier. This will ensure
       that the data packets in the admitted RSVP flow that are
       subsequently forwarded over the outgoing interface will contain
       the appropriate value encoded in their frame header.

   *   When an L3 device receives a PATH or RESV message over a managed
       segment in one L2 domain and it needs to forward the PATH/RESV
       message over an interface outside that domain, the L3 device must
       remove the TCLASS object (along with LAN_NHOP, RSVP_HOP_L2, and
       LAN_LOOPBACK objects in the case of the PATH message) before
       forwarding the PATH/RESV message. If the outgoing interface is on
       a separate L2 domain, these objects may be regenerated according
       to the processing rules applicable to that interface.

5. Detailed Message Processing Rules

5.1. Additional Notes on Terminology

   *   An L2 device may have several interfaces with attached segments
       that are part of the same L2 domain. A switch in a L2 domain is an
       example of such a device. A device which has several interfaces
       may contain a SBM protocol entity that acts in different
       capacities on each interface. For example, a SBM protocol entity
       could act as a SBM on interface A, and act as a DSBM on interface
       B.

   *   A SBM protocol entity on a layer 3 device can be a DSBM client,
       and SBM, a DSBM, or none of the above (SBM transparent). Non-
       transparent L3 devices can implement any combination of these
       roles simultaneously. DSBM clients always reside at L3 devices.

   *   A SBM protocol entity residing at a layer 2 device can be a SBM, a
       DSBM or none of the above (SBM transparent). A layer 2 device will
       never host a DSBM client.

5.2. Use Of Reserved IP Multicast Addresses

   As stated earlier, we require that the DSBM clients forward the RSVP
   PATH messages to their DSBMs in a L2 domain before they reach the
   next L3 hop in the path. RSVP PATH messages are addressed, according
   to RFC-2205, to their destination address (which can be either an IP
   unicast or multicast address). When a L2 device hosts a DSBM, a
   simple-to-implement mechanism must be provided for the device to
   capture an incoming PATH message and hand it over to the local DSBM
   agent without requiring the L2 device to snoop for L3 RSVP messages.

   In addition, DSBM clients need to know how to address SBM messages to
   the DSBM. For the ease of operation and to allow dynamic DSBM-client
   binding, it should be possible to easily detect and address the
   existing DSBM on a managed segment.

To facilitate dynamic DSBM-client binding as well as to enable easy
detection and capture of PATH messages at L2 devices, we require that
a DSBM be addressed using a logical address rather than a physical
address. We make use of reserved IP multicast address(es) for the
purpose of communication with a DSBM. In particular, we require that
when a DSBM client or a SBM forwards a PATH message over a managed
segment, it is addressed to a reserved IP multicast address. Thus, a
DSBM on a L2 device needs to be configured in a way to make it easy
to intercept the PATH message and forward it to the local SBM
protocol entity. For example, this may involve simply adding a static
entry in the device's filtering database (FDB) for the corresponding
MAC multicast address to ensure the PATH messages get intercepted and
are not forwarded further without the DSBM intervention.

Similarly, a DSBM always sends the PATH messages over a managed
segment using a reserved IP multicast address and, thus, the SBMs or
DSBM clients on the managed segments must simply be configured to
intercept messages addressed to the reserved multicast address on the
appropriate interfaces to easily receive PATH messages.

RSVP RESV messages continue to be unicast to the previous hop address
stored as part of the PATH state at each intermediate hop.

We define use of two reserved IP multicast addresses. We call these
the "AllSBM Address" and the "DSBMLogicalAddress". These are chosen
from the range of local multicast addresses, such that:

* They are not passed through layer 3 devices.

* They are passed transparently through layer 2 devices which are
   SBM transparent.

* They are configured in the permanent database of layer 2 devices
   which host SBMs or DSBMs, such that they are directed to the SBM
   management entity in these devices. This obviates the need for
   these devices to explicitly snoop for SBM related control packets.

* The two reserved addresses are 224.0.0.16 (DSBMLogicalAddress) and
   224.0.0.17 (AllSBMAddress).

These addresses are used as described in the following table:

Type	DSBMLogicaladdress	AllSBMAddress
DSBM Client	* Sends PATH messages to this address	* Monitors this address to detect the presence of a DSBM

		* Monitors this address to receive PATH messages forwarded by the DSBM
SBM	* Sends PATH messages to this address	* Monitors and sends on this address to participate in election of the DSBM
		* Monitors this address to receive PATH messages forwarded by the DSBM
DSBM	* Monitors this address for PATH messages directed to it	* Monitors and sends on this to participate in election of the DSBM
		* Sends PATH messages to this address

The L2 or MAC addresses corresponding to IP multicast addresses are
computed algorithmically using a reserved L2 address block (the high
order 24-bits are 00:00:5e). The Assigned Numbers RFC [RFC-1700]
gives additional details.

5.3. Layer 3 to Layer 2 Address Mapping

As stated earlier, DSBMs or DSBM clients residing at a L3 device must
include a LAN_NHOP_L2 address in the LAN_NHOP information so that L2
devices along the path of a PATH message do not need to separately
determine the mapping between the LAN_NHOP_L3 address in the LAN_NHOP
object and its corresponding L2 address (for example, using ARP).

For the purpose of such mapping at L3 devices, we assume a mapping
function called "map_address" that performs the necessary mapping:

                L2ADDR object = map_addr(L3Addr)

We do not specify how the function is implemented; the implementation
may simply involve access to the local ARP cache entry or may require
performing an ARP function. The function returns a L2ADDR object
that need not be interpreted by an L3 device and can be treated as an
opaque object. The format of the L2ADDR object is specified in
Appendix B.

5.4. Raw vs. UDP Encapsulation

We assume that the DSBMs, DSBM clients, and SBMs use only raw IP for
encapsulating RSVP messages that are forwarded onto a L2 domain.
Thus, when a SBM protocol entity on a L3 device forwards a RSVP
message onto a L2 segment, it will only use RAW IP encapsulation.

5.5. The Forwarding Rules

   The message processing and forwarding rules will be described in the
   context of the sample network illustrated in Figure 2.

   Figure 2 - A sample network or L2 domain consisting of switched and
   shared L2 segments

```
  ..........
          .
+------+  .   +------+ seg A  +------+ seg C  +------+ seg D +------+
| H1   |_____|  R1  |_____|  S1  |_____|  S2  |_____|  H2  |
|      | .   |      | .      |      | .      |      | .     |      |
+------+ .   +------+        +------+        +------+        +------+
         .                      |              /
1.0.0.0  .                      |             /
         .                      |___         /
         .               seg B  |           / seg E
  ..........                     |          /
              2.0.0.0           |         /
                         +-----------+
                         |    S3     |
                         |           |
                         +-----------+
                               |
                               |
                               |
                               |
                seg F          |                ................
             ----------------------------                    .
              |          |           |           .
          +------+   +------+     +------+        .        +------+
          | H3   |   | H4   |     |  R2  |_____|  H5  |
          |      |   |      |     |      | .              |      |
          +------+   +------+     +------+ .              +------+
                                           .
                                           .        3.0.0.0
                                           ................
```
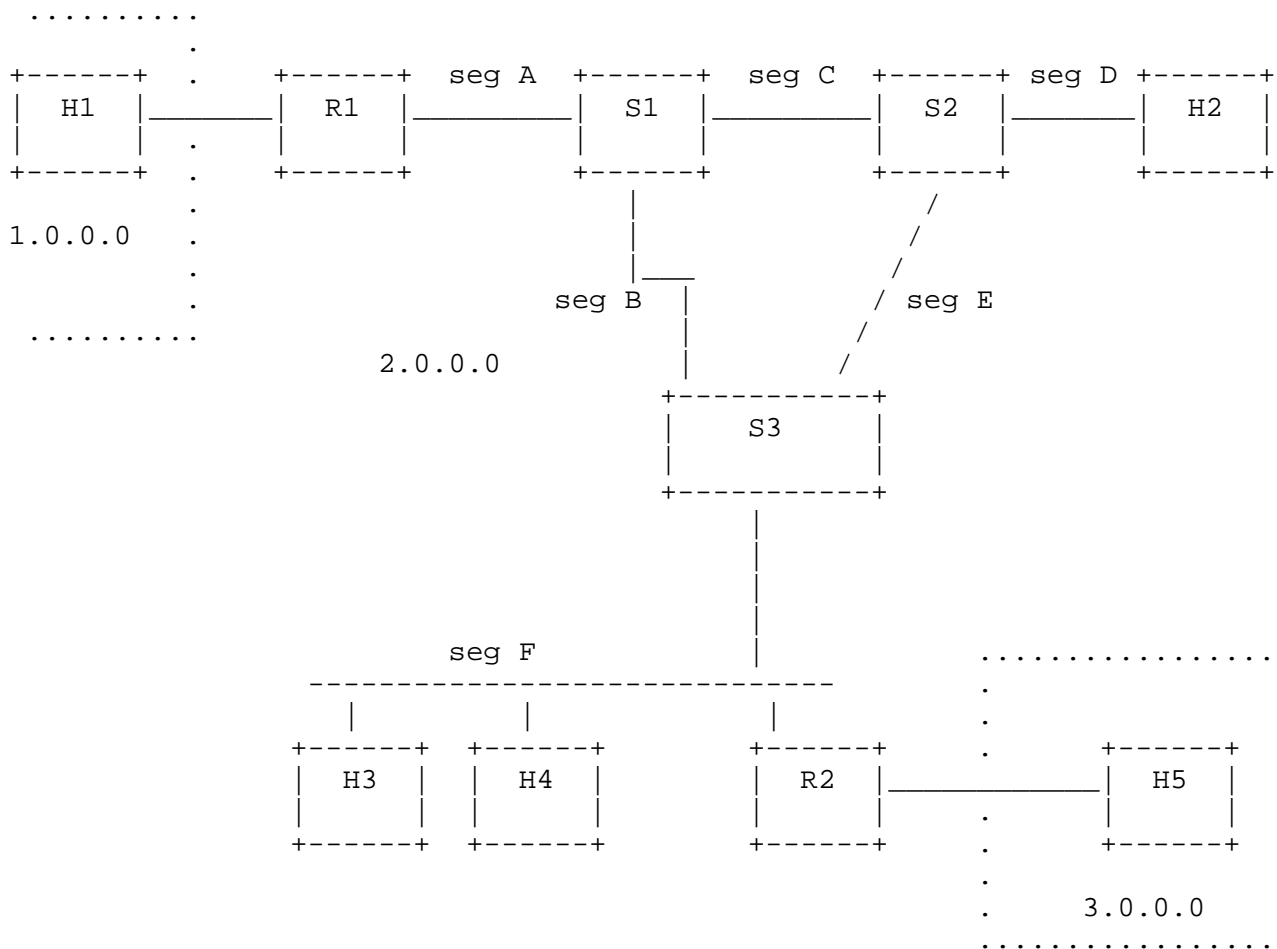
   Figure 2 illustrates a sample network topology consisting of three IP
   subnets (1.0.0.0, 2.0.0.0, and 3.0.0.0) interconnected using two
   routers. The subnet 2.0.0.0 is an example of a L2 domain consisting
   of switches, hosts, and routers interconnected using switched
   segments and a shared L2 segment. The sample network contains the
   following devices:

```
   Device           Type                     SBM Type

   H1, H5      Host (layer 3)           SBM Transparent
   H2-H4       Host (layer 3)           DSBM Client
   R1          Router (layer 3)         SBM
   R2          Router (layer 3)         DSBM for segment F
   S1          Switch (layer 2)         DSBM for segments A, B
   S2          Switch (layer 2)         DSBM for segments C, D, E
   S3          Switch (layer 2)         SBM
```

The following paragraphs describe the rules, which each of these
devices should use to forward PATH messages (rules apply to PATH_TEAR
messages as well). They are described in the context of the general
network illustrated above. While the examples do not address every
scenario, they do address most of the interesting scenarios.
Exceptions can be discussed separately.

The forwarding rules are applied to received PATH messages (routers
and switches) or originating PATH messages (hosts), as follows:

1. Determine the interface(s) on which to forward the PATH message
   using standard forwarding rules:

   * If there is a LAN_LOOPBACK object in the PATH message, and it
      carries the address of this device, silently discard the
      message. (See the section below on "Additional notes on
      forwarding the PATH message onto a managed segment).

   * Layer 3 devices use the RSVP session address and perform a
      routing lookup to determine the forwarding interface(s).

   * Layer 2 devices use the LAN_NHOP_L2 address in the LAN_NHOP
      information and MAC forwarding tables to determine the
      forwarding interface(s). (See the section below on "Additional
      notes on forwarding the PATH message onto a managed segment")

   2. For each forwarding interface:

   * If the device is a layer 3 device, determine whether the
      interface is on a managed segment managed by a DSBM, based on
      the presence or absence of I_AM_DSBM messages. If the interface
      is not on a managed segment, strip out RSVP_HOP_L2, LAN_NHOP,
      LAN_LOOPBACK, and TCLASS objects (if present), and forward to
      the unicast or multicast destination.

> (Note that the RSVP Class Numbers for these new objects are
> chosen so that if an RSVP message includes these objects, the
> nodes that are RSVP-aware, but do not participate in the SBM
> protocol, will ignore and silently discard such objects.)

  *   If the device is a layer 2 device or it is a layer 3 device
      and the interface is on a managed segment, proceed to rule
      #3.

   3. Forward the PATH message onto the managed segment:

  *   If the device is a layer 3 device, insert LAN_NHOP address
      objects, a LAN_LOOPBACK, and a RSVP_HOP_L2 object into the PATH
      message. The LAN_NHOP objects carry the LAN_NHOP_L3 and
      LAN_NHOP_L2 addresses of the next layer 3 hop. The RSVP_HOP_L2
      object carries the device's own L2 address, and the
      LAN_LOOPBACK object contains the IP address of the outgoing
      interface.

      An L3 device should use the map_addr() function described
      earlier to obtain an L2 address corresponding to an IP address.

  * If the device hosts the DSBM for the segment to which the
     forwarding interface is attached, do the following:

     - Retrieve the PHOP information from the standard RSVP HOP
       object in the PATH message, and store it. This will be used
       to route RESV messages back through the L2 network. If the
       PATH message arrived over a managed segment, it will also
       contain the RSVP_HOP_L2 object; then retrieve and store also
       the previous hop's L2 address in the PATH state.

     - Copy the IP address of the forwarding interface (layer 2
       devices must also have IP addresses) into the standard RSVP
       HOP object and the L2 address of the forwarding interface
       into the RSVP_HOP_L2 object.

     - If the PATH message received does not contain the TCLASS
       object, insert a TCLASS object. The user_priority value
       inserted in the TCLASS object is based on service mappings
       internal to the device that are configured according to the
       guidelines listed in [RFC-MAP]. If the message already
       contains the TCLASS object, the user_priority value may be
       changed based again on the service mappings internal to the
       device.

* If the device is a layer 3 device and hosts a SBM for the
  segment to which the forwarding interface is attached, it *is
  required* to retrieve and store the PHOP info.

  If the device is a layer 2 device and hosts a SBM for the
  segment to which the forwarding interface is attached, it is
  not required to retrieve and store the PHOP info. If it does
  not do so, the SBM must leave the standard RSVP HOP object and
  the RSVP_HOP_L2 objects in the PATH message intact and it will
  not receive RESV messages.

  If the SBM on a L2 device chooses to overwrite the RSVP HOP and
  RSVP_HOP_L2 objects with the IP and L2 addresses of its
  forwarding interface, it will receive RESV messages. In this
  case, it must store the PHOP address info received in the
  standard RSVP_HOP field and RSVP_HOP_L2 objects of the incident
  PATH message.

  In both the cases mentioned above (L2 or L3 devices), the SBM
  must forward the TCLASS object in the received PATH message
  unchanged.

* Copy the IP address of the forwarding interface into the
  LAN_LOOPBACK object, unless the SBM protocol entity is a DSBM
  reflecting a PATH message back onto the incident interface.
  (See the section below on "Additional notes on forwarding a
  PATH message onto a managed segment").

* If the SBM protocol entity is the DSBM for the segment to which
  the forwarding interface is attached, it must send the PATH
  message to the AllSBMAddress.

* If the SBM protocol entity is a SBM or a DSBM Client on the
  segment to which the forwarding interface is attached, it must
  send the PATH message to the DSBMLogicalAddress.

5.5.1. Additional notes on forwarding a PATH message onto a managed
       segment

   Rule #1 states that normal IEEE 802.1D forwarding rules should be
   used to determine the interfaces on which the PATH message should be
   forwarded. In the case of data packets, standard forwarding rules at
   a L2 device dictate that the packet should not be forwarded on the
   interface from which it was received. However, in the case of a DSBM
   that receives a PATH message over a managed segment, the following
   exception applies:

    E1. If the address in the LAN_NHOP object is a unicast address,
        consult the filtering database (FDB) to determine whether the
        destination address is listed on the same interface over which
        the message was received. If yes, follow the rule below on
        "reflecting a PATH message back onto an interface" described
        below; otherwise, proceed with the rest of the message
        processing as usual.

    E2. If there are members of the multicast group address (specified
        by the addresses in the LAN_NHOP object), on the segment from
        which the message was received, the message should be
        forwarded back onto the interface from which it was received
        and follow the rule on "reflecting a PATH message back onto an
        interface" described below.

  *** Reflecting a PATH message back onto an interface ***

   Under the circumstances described above, when a DSBM reflects the
   PATH message back onto an interface over which it was received, it
   must address it using the AllSBMAddress.

   Since it is possible for a DSBM to reflect a PATH message back
   onto the interface from which it was received, precautions must be
   taken to avoid looping these messages indefinitely. The
   LAN_LOOPBACK object addresses this issue. All SBM protocol
   entities (except DSBMs reflecting a PATH message) overwrite the
   LAN_LOOPBACK object in the PATH message with the IP address of the
   outgoing interface. DSBMs which are reflecting a PATH message,
   leave the LAN_LOOPBACK object unchanged. Thus, SBM protocol
   entities will always be able to recognize a reflected multicast
   message by the presence of their own address in the LAN_LOOPBACK
   object. These messages should be silently discarded.

5.6. Applying the Rules -- Unicast Session

   Let's see how the rules are applied in the general network
   illustrated previously (see Figure 2).

   Assume that H1 is sending a PATH for a unicast session for which H5
   is the receiver. The following PATH message is composed by H1:

                             RSVP Contents
   RSVP session IP address   IP address of H5 (3.0.0.35)
   Sender Template   IP address of H1 (1.0.0.11)
   PHOP   IP address of H1 (1.0.0.11)
   RSVP_HOP_L2   n/a (H1 is not sending onto a managed
                                  segment)
   LAN_NHOP   n/a (H1 is not sending onto a managed

```
                                  segment)
      LAN_LOOPBACK                n/a  (H1 is not sending onto a managed
                                  segment)


                                  IP Header
      Source address             IP address of H1 (1.0.0.11)
      Destn address              IP addr of H5 (3.0.0.35, assuming raw mode
                                   & router alert)


                                  MAC Header
      Destn address              The L2 addr corresponding to R1 (determined
                                   by map_addr() and routing tables at H1)
```

Since H1 is not sending onto a managed segment, the PATH message is
composed and forwarded according to standard RSVP processing rules.

Upon receipt of the PATH message, R1 composes and forwards a PATH
message as follows:

```
                                  RSVP Contents
      RSVP session IP address    IP address of H5
      Sender Template            IP address of H1
      PHOP                       IP address of R1 (2.0.0.1)
                                  (seed the return path for RESV messages)
      RSVP_HOP_L2                L2 address of R1
      LAN_NHOP                   LAN_NHOP_L3 (2.0.0.2) and
                                 LAN_NHOP_L2 address of R2 (L2ADDR)
                                  (this is the next layer 3 hop)
      LAN_LOOPBACK               IP address of R1 (2.0.0.1)


                                  IP Header
      Source address             IP address of H1
      Destn address              DSBMLogical IP address (224.0.0.16)


                                  MAC Header
      Destn address              DSBMLogical MAC address
```

  * R1 does a routing lookup on the RSVP session address, to
    determine the IP address of the next layer 3 hop, R2.

  * It determines that R2 is accessible via seg A and that seg A
    is managed by a DSBM, S1.

  * Therefore, it concludes that it is sending onto a managed
    segment, and composes LAN_NHOP objects to carry the layer 3
    and layer 2 next hop addresses. To compose the LAN_NHOP
    L2ADDR object, it invokes the L3 to L2 address mapping function

("map_address") to find out the MAC address for the next hop
L3 device, and then inserts a LAN_NHOP_L2ADDR object (that
carries the MAC address) in the message.

* Since R1 is not the DSBM for seg A, it sends the PATH message
  to the DSBMLogicalAddress.

Upon receipt of the PATH message, S1 composes and forwards a PATH
message as follows:

                              RSVP Contents
RSVP session IP address   IP address of H5
Sender Template   IP address of H1
PHOP   IP addr of S1 (seed the return path for RESV
                          messages)
RSVP_HOP_L2   L2 address of S1
LAN_NHOP   LAN_NHOP_L3 (IP) and LAN_NHOP_L2
                               address of R2
                          (layer 2 devices do not modify the LAN_NHOP)
LAN_LOOPBACK   IP addr of S1

                              IP Header
Source address   IP address of H1
Destn address   AllSBMIPaddr (224.0.0.17, since S1 is the
                          DSBM for seg B).

                              MAC Header
Destn address   All SBM MAC address (since S1 is the DSBM
                          for seg B).

* S1 looks at the LAN_NHOP address information to determine the
  L2 address towards which it should forward the PATH message.

* From the bridge forwarding tables, it determines that the L2
  address is reachable via seg B.

* S1 inserts the RSVP_HOP_L2 object and overwrites the RSVP HOP
  object (PHOP) with its own addresses.

* Since S1 is the DSBM for seg B, it addresses the PATH message
  to the AllSBMAddress.

Upon receipt of the PATH message, S3 composes and forwards a PATH
message as follows:

```
                              RSVP Contents
     RSVP session IP addr       IP address of H5
     Sender Template            IP address of H1
     PHOP                       IP addr of S3 (seed the return
                                    path for RESV messages)
     RSVP_HOP_L2                L2 address of S3
     LAN_NHOP                   LAN_NHOP_L3 (IP) and
                                LAN_NHOP_L2 (MAC) address of R2
                                (L2 devices don't modify  LAN_NHOP)
     LAN_LOOPBACK               IP address of S3

                               IP Header
     Source address              IP address of H1
     Destn address               DSBMLogical IP addr (since S3 is
                                     not the DSBM for seg F)

                              MAC Header
     Destn address               DSBMLogical MAC address
```

   * S3 looks at the LAN_NHOP address information to determine the
      L2 address towards which it should forward the PATH message.

   * From the bridge forwarding tables, it determines that the L2
      address is reachable via segment F.

   * It has discovered that R2 is the DSBM for segment F. It
      therefore sends the PATH message to the DSBMLogicalAddress.

   * Note that S3 may or may not choose to overwrite the PHOP
      objects with its own IP and L2 addresses. If it does so, it
      will receive RESV messages. In this case, it must also store
      the PHOP info received in the incident PATH message so that
      it is able to forward the RESV messages on the correct path.

   Upon receipt of the PATH message, R2 composes and forwards a PATH
   message as follows:

```
                         RSVP Contents
   RSVP session IP addr  IP address of H5
   Sender Template       IP address of H1
   PHOP                  IP addr of R2 (seed the return path for RESV
                         messages)
   RSVP_HOP_L2           Removed by R2  (R2 is not sending onto a
                             managed segment)
   LAN_NHOP              Removed by R2  (R2 is not sending onto a
                         managed segment)
```

```
                        IP Header
Source address          IP address of H1
Destn address           IP address of H5, the RSVP session address

                        MAC Header
Destn address           L2 addr corresponding to H5, the next
                        layer 3 hop
```

* R2 does a routing lookup on the RSVP session address, to
  determine the IP address of the next layer 3 hop, H5.

* It determines that H5 is accessible via a segment for which
  there is no DSBM (not a managed segment).

* Therefore, it removes the LAN_NHOP and RSVP_HOP_L2 objects
  and places the RSVP session address in the destination
  address of the IP header. It places the L2 address of the
  next layer 3 hop, into the destination address of the MAC
  header and forwards the PATH message to H5.

5.7. Applying the Rules - Multicast Session

   The rules described above also apply to multicast (m/c) sessions.
   For the purpose of this discussion, it is assumed that layer 2
   devices track multicast group membership on each port individually.
   Layer 2 devices which do not do so, will merely generate extra
   multicast traffic. This is the case for L2 devices which do not
   implement multicast filtering or GARP/GMRP capability.

   Assume that H1 is sending a PATH for an m/c session for which H3 and
   H5 are the receivers. The rules are applied as they are in the
   unicast case described previously, until the PATH message reaches R2,
   with the following exception. The RSVP session address and the
   LAN_NHOP carry the destination m/c addresses rather than the unicast
   addresses carried in the unicast example.

   Now let's look at the processing applied by R2 upon receipt of the
   PATH message. Recall that R2 is the DSBM for segment F. Therefore, S3
   will have forwarded its PATH message to the DSBMLogicalAddress, to be
   picked up by R2. The PATH message will not have been seen by H3 (one
   of the m/c receivers), since it monitors only the AllSBMAddress, not
   the DSBMLogicalAddress for incoming PATH messages. We rely on R2 to
   reflect the PATH message back onto seg f, and to forward it to H5. R2
   forwards the following PATH message onto seg f:

```
                        RSVP Contents
RSVP session addr   m/c session address
Sender Template     IP address of H1
```

```
PHOP                    IP addr of R2 (seed the return path for
                        RESV messages)
RSVP_HOP_L2             L2 addr of R2
LAN_NHOP                m/c session address and corresponding L2 address
LAN_LOOPBACK            IP addr of S3 (DSBMs reflecting a PATH
                        message don't modify this object)


                          IP Header
Source address          IP address of H1


Destn address           AllSBMIP address (since R2 is the DSBM for seg F)


                          MAC Header
Destn address           AllSBMMAC address (since R2 is the
                          DSBM for seg F)
```

Since H3 is monitoring the All SBM Address, it will receive the PATH
message reflected by R2. Note that R2 violated the standard
forwarding rules here by sending an incoming message back onto the
interface from which it was received. It protected against loops by
leaving S3's address in the LAN_LOOPBACK object unchanged.

R2 forwards the following PATH message on to H5:

```
                          RSVP Contents
RSVP session addr       m/c session address
Sender Template         IP address of H1
PHOP                    IP addr of R2 (seed the return path for RESV
                        messages)
RSVP_HOP_L2             Removed by R2 (R2 is not sending onto a
                        managed segment)
LAN_NHOP                Removed by R2 (R2 is not sending onto a
                        managed segment)
LAN_LOOPBACK            Removed by R2 (R2 is not sending onto a
                        managed segment)


                          IP Header
Source address          IP address of H1
Destn address           m/c session address


                          MAC Header
Destn address           MAC addr corresponding to the m/c
                        session address
```

* R2 determines that there is an m/c receiver accessible via a
   segment for which there is no DSBM. Therefore, it removes the
   LAN_NHOP and RSVP_HOP_L2 objects and places the RSVP session
   address in the destination address of the IP header. It

        places the corresponding L2 address into the destination
        address of the MAC header and multicasts the message towards
        H5.

5.8. Merging Traffic Class objects

   When a DSBM client receives TCLASS objects from different senders
   (different PATH messages) in the same RSVP session and needs to
   combine them for sending back a single RESV message (as in a wild-
   card style reservation), the DSBM client must choose an appropriate
   value that corresponds to the desired-delay traffic class. An
   accompanying document discusses the guidelines for traffic class
   selection based on desired service and the TSpec information [RFC-
   MAP].

   In addition, when a SBM or DSBM needs to merge RESVs from different
   next hops at a merge point, it must decide how to handle the TCLASS
   values in the incoming RESVs if they do not match. Consider the case
   when a reservation is in place for a flow at a DSBM (or SBM) with a
   successful admission control done for the TCLASS requested in the
   first RESV for the flow. If another RESV (not the refresh of the
   previously admitted RESV) for the same flow arrives at the DSBM, the
   DSBM must first check the TCLASS value in the new RESV against the
   TCLASS value in the already installed RESV. If the two values are
   same, the RESV requests are merged and the new, merged RESV installed
   and forwarded using the normal rules of message processing. However,
   if the two values are not identical, the DSBM must generate and send
   a RESV_ERR message towards the sender (NHOP) of the newer, RESV
   message. The RESV_ERR must specify the error code corresponding to
   the RSVP "traffic control error" (RESV_ERR code 21) that indicates
   failure to merge two incompatible service requests (sub-code 01 for
   the RSVP traffic control error) [RFC-2205]. The RESV_ERR message may
   include additional objects to assist downstream nodes in recovering
   from this condition. The definition and usage of such objects is
   beyond the scope of this memo.

5.9. Operation of SBM Transparent Devices

   SBM transparent devices are unaware of the entire SBM/DSBM protocol.
   They do not intercept messages addressed to either of the SBM related
   local group addresses (the DSBMLogicalAddrss and the ALLSBMAddress),
   but instead, pass them through. As a result, they do not divide the
   DSBM election scope, they do not explicitly participate in routing of
   PATH or RESV messages, and they do not participate in admission
   control. They are entirely transparent with respect to SBM operation.

According to the definitions provided, physical segments
interconnected by SBM transparent devices are considered a single
managed segment. Therefore, DSBMs must perform admission control on
such managed segments, with limited knowledge of the segment's
topology. In this case, the network administrator should configure
the DSBM for each managed segment, with some reasonable approximation
of the segment's capacity. A conservative policy would configure the
DSBM for the lowest capacity route through the managed segment. A
liberal policy would configure the DSBM for the highest capacity
route through the managed segment. A network administrator will
likely choose some value between the two, based on the level of
guarantee required and some knowledge of likely traffic patterns.

This document does not specify the configuration mechanism or the
choice of a policy.

5.10. Operation of SBMs Which are NOT DSBMs

In the example illustrated, S3 hosts a SBM, but the SBM on S3 did not
win the election to act as DSBM on any segment. One might ask what
purpose such a SBM protocol entity serves. Such SBMs actually provide
two useful functions. First, the additional SBMs remain passive in
the background for fault tolerance. They listen to the periodic
announcements from the current DSBM for the managed segment (Appendix
A describes this in more detail) and step in to elect a new DSBM when
the current DSBM fails or ceases to be operational for some reason.
Second, such SBMs also provide the important service of dividing the
election scope and reducing the size and complexity of managed
segments. For example, consider the sample topology in Figure 3
again. the device S3 contains an SBM that is not a DSBM for any f the
segments, B, E, or F, attached to it. However, if the SBM protocol
entity on S3 was not present, segments B and F would not be separate
segments from the point of view of the SBM protocol. Instead, they
would constitute a single managed segment, managed by a single DSBM.
Because the SBM entity on S3 divides the election scope, seg B and
seg F are each managed by separate DSBMs. Each of these segments have
a trivial topology and a well defined capacity. As a result, the
DSBMs for these segments do not need to perform admission control
based on approximations (as would be the case if S3 were SBM
transparent).

Note that, SBM protocol entities which are not DSBMs, are not
required to overwrite the PHOP in incident PATH messages with their
own address. This is because it is not necessary for RESV messages to
be routed through these devices. RESV messages are only required to
be routed through the correct sequence of DSBMs. SBMs may not
process RESV messages that do pass through them, other than to
forward them towards their destination address, using standard

forwarding rules.

SBM protocol entities which are not DSBMs are required to overwrite
the address in the LAN_LOOPBACK object with their own address, in
order to avoid looping multicast messages. However, no state need be
stored.

6. Inter-Operability Considerations

There are a few interesting inter-operability issues related to the
deployment of a DSBM-based admission control method in an environment
consisting of network nodes with and without RSVP capability. In the
following, we list some of these scenarios and explain how SBM-aware
clients and nodes can operate in those scenarios:

6.1. An L2 domain with no RSVP capability.

It is possible to envisage L2 domains that do not use RSVP signaling
for requesting resource reservations, but, instead, use some other
(e.g., SNMP or static configuration) mechanism to reserve bandwidth
at a particular network device such as a router. In that case, the
question is how does a DSBM-based admission control method work and
interoperate with the non-RSVP mechanism. The SBM-based method does
not attempt to provide an admission control solution for such an
environment. The SBM-based approach is part of an end to end
signaling approach to establish resource reservations and does not
attempt to provide a solution for SNMP-based configuration scenario.

As stated earlier, the SBM-based approach can, however, co-exist with
any other, non-RSVP bandwidth allocation mechanism as long as
resources being reserved are either partitioned statically between
the different mechanisms or are resolved dynamically through a common
bandwidth allocator so that there is no over-commitment of the same
resource.

6.2. An L2 domain with SBM-transparent L2 Devices.

This scenario has been addressed earlier in the document. The SBM-
based method is designed to operate in such an environment. When
SBM-transparent L2 devices interconnect SBM-aware devices, the
resulting managed segment is a combination of one or more physical
segments and the DSBM for the managed segment may not be as efficient
in allocating resources as it would if all L2 devices were SBM-aware.

6.3. An L2 domain on which some RSVP-based senders are not DSBM clients.

   All senders that are sourcing RSVP-based traffic flows onto a managed
   segment MUST be SBM-aware and participate in the SBM protocol. Use
   of the standard, non-SBM version of RSVP may result in over-
   allocation of resources, as such use bypasses the resource management
   function of the DSBM. All other senders (i.e., senders that are not
   sending streams subject to RSVP admission control) should be elastic
   applications that send traffic of lower priority than the RSVP
   traffic, and use TCP-like congestion avoidance mechanisms.

   All DSBMs, SBMs, or DSBM clients on a managed segment (a segment with
   a currently active DSBM) must not accept PATH messages from senders
   that are not SBM-aware. PATH messages from such devices can be easily
   detected by SBMs and DSBM clients as they would not be multicast to
   the ALLSBMAddress (in case of SBMs and DSBM clients) or the
   DSBMLogicalAddress (in case of DSBMs).

6.4. A non-SBM router that interconnects two DSBM-managed L2 domains.

   Multicast SBM messages (e.g., election and PATH messages) have local
   scope and are not intended to pass between the two domains. A
   correctly configured non-SBM router will not pass such messages
   between the domains. A broken router implementation that does so may
   cause incorrect operation of the SBM protocol and consequent over- or
   under-allocation of resources.

6.5. Interoperability with RSVP clients that use UDP encapsulation and
     are not capable of receiving/sending RSVP messages using RAW_IP

   This document stipulates that DSBMs, DSBM clients, and SBMs use only
   raw IP for encapsulating RSVP messages that are forwarded onto a L2
   domain. RFC-2205 (the RSVP Proposed Standard) includes support for
   both raw IP and UDP encapsulation. Thus, a RSVP node using only the
   UDP encapsulation will not be able to interoperate with the DSBM
   unless DSBM accepts and supports UDP encapsulated RSVP messages.

7. Guidelines for Implementers

   In the following, we provide guidelines for implementers on different
   aspects of the implementation of the SBM-based admission control
   procedure including suggestions for DSBM initialization, etc.

7.1. DSBM Initialization

   As stated earlier, DSBM initialization includes configuration of
   maximum bandwidth that can be reserved on a managed segment under its
   control. We suggest the following guideline.

In the case of a managed segment consisting of L2 devices
interconnected by a single shared segment, DSBM entities on such
devices should assume the bandwidth of the interface as the total
link bandwidth. In the case of a DSBM located in a L2 switch, it
might additionally need to be configured with an estimate of the
device's switching capacity if that is less than the link bandwidth,
and possibly with some estimate of the buffering resources of the
switch (see [RFC-FRAME] for the architectural model assumed for L2
switches). Given the total link bandwidth, the DSBM may be further
configured to limit the maximum amount of bandwidth for RSVP-enabled
flows to ensure spare capacity for best-effort traffic.

7.2. Operation of DSBMs in Different L2 Topologies

Depending on a L2 topology, a DSBM may be called upon to manage
resources for one or more segments and the implementers must bear in
mind efficiency implications of the use of DSBM in different L2
topologies. Trivial L2 topologies consist of a single "physical
segment". In this case, the 'managed segment' is equivalent to a
single segment. Complex L2 topologies may consist of a number of
Admission control on such an L2 extended segment can be performed
from a single pool of resources, similar to a single shared segment,
from the point of view of a single DSBM.

This configuration compromises the efficiency with which the DSBM can
allocate resources. This is because the single DSBM is required to
make admission control decisions for all reservation requests within
the L2 topology, with no knowledge of the actual physical segments
affected by the reservation.

We can realize improvements in the efficiency of resource allocation
by subdividing the complex segment into a number of managed segments,
each managed by their own DSBM. In this case, each DSBM manages a
managed segment having a relatively simple topology. Since managed
segments are simpler, the DSBM can be configured with a more accurate
estimate of the resources available for all reservations in the
managed segment. In the ultimate configuration, each physical segment
is a managed segment and is managed by its own DSBM. We make no
assumption about the number of managed segments but state, simply,
that in complex L2 topologies, the efficiency of resource allocation
improves as the granularity of managed segments increases.

8. Security Considerations

The message formatting and usage rules described in this note raise
security issues, identical to those raised by the use of RSVP and
Integrated Services. It is necessary to control and authenticate

access to enhanced qualities of service enabled by the technology described in this RFC. This requirement is discussed further in [RFC-2205], [RFC-2211], and [RFC-2212].

[RFC-RSVPMD5] describes the mechanism used to protect the integrity of RSVP messages carrying the information described here. A SBM implementation should satisfy the requirements of that RFC and provide the suggested mechanisms just as though it were a conventional RSVP implementation. It should further use the same mechanisms to protect the additional, SBM-specific objects in a message.

Finally, it is also necessary to authenticate DSBM candidates during the election process, and a mechanism based on a shared secret among the DSBM candidates may be used. The mechanism defined in [RFC-RSVPMD5] should be used.

9. References

[RFC 2205]   Braden, R., Zhang, L., Berson, S., Herzog, S. and S.
                Jamin, "Resource ReSerVation Protocol (RSVP) -- Version
                1 Functional Specification", RFC 2205, September 1997.

[RFC-RSVPMD5] Baker, F., Lindell, B. and M. Talwar, "RSVP
                Cryptographic Authentication", RFC 2747, January 2000.

[RFC 2206]   Baker, F. and J. Krawczyk, "RSVP Management Information
                Base", RFC 2206, September 1997.

[RFC 2211]   Wroclawski, J., "Specification of the Controlled-Load
                Network Element Service", RFC 2211, September 1997.

[RFC 2212]   Shenker, S., Partridge, C. and R. Guerin,
                "Specification of Guaranteed Quality of Service", RFC
                2212, September 1997.

[RFC 2215]   Shenker, S. and J. Wroclawski, "General
                Characterization Parameters for Integrated Service
                Network Elements", RFC 2215, September 1997.

[RFC 2210]   Wroclawski, J., "The Use of RSVP with IETF Integrated
                Services", RFC 2210, September 1997.

[RFC 2213]   Baker, F. and J. Krawczyk, "Integrated Services
                Management Information Base", RFC 2213, September 1997.

   [RFC-FRAME]   Ghanwani, A., Pace, W., Srinivasan, V., Smith, A. and
                  M.Seaman, "A Framework for Providing Integrated
                  Services Over Shared and Switched LAN Technologies",
                  RFC 2816, May 2000.

   [RFC-MAP]   Seaman, M., Smith, A. and E. Crawley, "Integrated
                  Service Mappings on IEEE 802 Networks", RFC 2815, May
                  2000.

   [IEEE802Q]   "IEEE Standards for Local and Metropolitan Area
                  Networks: Virtual Bridged Local Area Networks", Draft
                  Standard P802.1Q/D9, February 20, 1998.

   [IEEEP8021p]   "Information technology - Telecommunications and
                  information exchange between systems - Local and
                  metropolitan area networks - Common specifications -
                  Part 3: Media Access Control (MAC) Bridges: Revision
                  (Incorporating IEEE P802.1p: Traffic Class Expediting
                  and Dynamic Multicast Filtering)", ISO/IEC Final CD
                  15802-3 IEEE P802.1D/D15, November 24, 1997.

   [IEEE8021D]   "MAC Bridges", ISO/IEC 10038, ANSI/IEEE Std 802.1D-
                  1993.

A.1. Introduction

   To simplify the rest of this discussion, we will assume that there is
   a single DSBM for the entire L2 domain (i.e., assume a shared L2
   segment for the entire L2 domain). Later, we will discuss how a DSBM
   is elected for a half-duplex or full-duplex switched segment.

   To allow for quick recovery from the failure of a DSBM, we assume
   that additional SBMs may be active in a L2 domain for fault
   tolerance. When more than one SBM is active in a L2 domain, the SBMs
   use an election algorithm to elect a DSBM for the L2 domain. After
   the DSBM is elected and is operational, other SBMs remain passive in
   the background to step in to elect a new DSBM when necessary. The
   protocol for electing and discovering DSBM is called the "DSBM
   election protocol" and is described in the rest of this Appendix.

A.1.1. How a DSBM Client Detects a Managed Segment

   Once elected, a DSBM periodically multicasts an I_AM_DSBM message on
   the AllSBMAddress to indicate its presence. The message is sent every
   period (e.g., every 5 seconds) according to the RefreshInterval timer
   value (a configuration parameter). Absence of such a message over a
   certain time interval (called "DSBMDeadInterval"; another
   configuration parameter typically set to a multiple of
   RefreshInterval) indicates that the DSBM has failed or terminated and
   triggers another round of the DSBM election. The DSBM clients always
   listen for periodic DSBM advertisements. The advertisement includes
   the unicast IP address of the DSBM (DSBMAddress) and DSBM clients
   send their PATH/RESV (or other) messages to the DSBM. When a DSBM
   client detects the failure of a DSBM, it waits for a subsequent
   I_AM_DSBM advertisement before resuming any communication with the
   DSBM. During the period when a DSBM is not present, a DSBM client may
   forward outgoing PATH messages using the standard RSVP forwarding
   rules.

   The exact message formats and addresses used for communication with
   (and among) SBM(s) are described in Appendix B.

A.2. Overview of the DSBM Election Procedure

   When a SBM first starts up, it listens for incoming DSBM
   advertisements for some period to check whether a DSBM already exists
   in its L2 domain. If one already exists (and no new election is in
   progress), the new SBM stays quiet in the background until an
   election of DSBM is necessary. All messages related to the DSBM
   election and DSBM advertisements are always sent to the
   AllSBMAddress.

   If no DSBM exists, the SBM initiates the election of a DSBM by
   sending out a DSBM_WILLING message that lists its IP address as a
   candidate DSBM and its "SBM priority". Each SBM is assigned a
   priority to determine its relative precedence. When more than one
   SBM candidate exists, the SBM priority determines who gets to be the
   DSBM based on the relative priority of candidates. If there is a tie
   based on the priority value, the tie is broken using the IP
   addresses of tied candidates (one with the higher IP address in the
   lexicographic order wins). The details of the election protocol start
   in Section A.4.

A.2.1 Summary of the Election Algorithm

   For the purpose of the algorithm, a SBM is in one of the four states
   (Idle, DetectDSBM, ElectDSBM, IAMDSBM).

   A SBM (call it X) starts up in the DetectDSBM state and waits for a
   ListenInterval for incoming I_AM_DSBM (DSBM advertisement) or
   DSBM_WILLING messages. If an I_AM_DSBM advertisement is received
   during this state, the SBM notes the current DSBM (its IP address and
   priority) and enters the Idle state. If a DSBM_WILLING message is
   received from another SBM (call it Y) during this state, then X
   enters the ElectDSBM state. Before entering the new state, X first
   checks to see whether it itself is a better candidate than Y and, if
   so, sends out a DSBM_WILLING message and then enters the ElectDSBM
   state.

   When a SBM (call it X) enters the ElectDSBM state, it sets a timer
   (called ElectionIntervalTimer, and typically set to a value at least
   equal to the DSBMDeadInterval value) to wait for the election to
   finish and to discover who is the best candidate. In this state, X
   keeps track of the best (or better) candidate seen so far (including
   itself). Whenever it receives another DSBM_WILLING message it updates
   its notion of the best (or better) candidate based on the priority
   (and tie-breaking) criterion. During the ElectionInterval, X sends
   out a DSBM_WILLING message every RefreshInterval to (re)assert its
   candidacy.

   At the end of the ElectionInterval, X checks whether it is the best
   candidate so far. If so, it declares itself to be the DSBM (by
   sending out the I_AM_DSBM advertisement) and enters the IAMDSBM
   state; otherwise, it decides to wait for the best candidate to
   declare itself the winner. To wait, X re-initializes its ElectDSBM
   state and continues to wait for another round of election (each round
   lasts for an ElectionTimerInterval duration).

A SBM is in Idle state when no election is in progress and the DSBM
is already elected (and happens to be someone else). In this state,
it listens for incoming I_AM_DSBM advertisements and uses a
DSBMDeadIntervalTimer to detect the failure of DSBM. Every time the
advertisement is received, the timer is restarted. If the timer
fires, the SBM goes into the DetectDSBM state to prepare to elect the
new DSBM. If a SBM receives a DSBM_WILLING message from the current
DSBM in this state, the SBM enters the ElectDSBM state after sending
out a DSBM_WILLING message (to announce its own candidacy).

In the IAMDSBM state, the DSBM sends out I_AM_DSBM advertisements
every refresh interval. If the DSBM wishes to shut down (gracefully
terminate), it sends out a DSBM_WILLING message (with SBM priority
value set to zero) to initiate the election procedure. The priority
value zero effectively removes the outgoing DSBM from the election
procedure and makes way for the election of a different DSBM.

A.3. Recovering from DSBM Failure

When a DSBM fails (DSBMDeadIntervalTimer fires), all the SBMs enter
the ElectDSBM state and start the election process.

At the end of the ElectionInterval, the elected DSBM sends out an
I_AM_DSBM advertisement and the DSBM is then operational.

A.4. DSBM Advertisements

The I_AM_DSBM advertisement contains the following information:

1. DSBM address information -- contains the IP and L2 addresses of
    the DSBM and its SBM priority (a configuration parameter --
    priority specified by a network administrator). The priority
    value is used to choose among candidate SBMs during the election
    algorithm. Higher integer values indicate higher priority and the
    value is in the range 0..255. The value zero indicates that the
    SBM is not eligible to be the DSBM. The IP address is required
    and used for breaking ties. The L2 address is for the interface
    of the managed segment.

2. RegreshInterval -- contains the value of RefreshInterval in
    seconds. Value zero indicates the parameter has been omitted in
    the message. Receivers may substitute their own default value in
    this case.

3. DSBMDeadInterval -- contains the value of DSBMDeadInterval in
    seconds. If the value is omitted (or value zero is specified), a
    default value (from initial configuration) should be used.

4. Miscellaneous configuration information to be advertised to
    senders on the managed segment. See Appendix C for further
    details.

A.5. DSBM_WILLING Messages

When a SBM wishes to declare its candidacy to be the DSBM during an
election phase, it sends out a DSBM_WILLING message. The DSBM_WILLING
message contains the following information:

1. DSBM address information -- Contains the SBM's own addresses (IP
    and L2 address), if it wishes to be the DSBM. The IP address is
    required and used for breaking ties. The L2 address is the
    address of the interface for the managed segment in question.
    Also, the DSBM address information includes the corresponding
    priority of the SBM whose address is given above.

A.6. SBM State Variables

For each network interface, a SBM maintains the following state
variables related to the election of the DSBM for the L2 domain on
that interface:

    a) LocalDSBMAddrInfo -- current DSBM's IP address (initially,
    0.0.0.0) and priority. All IP addresses are assumed to be in
    network byte order. In addition, current DSBM's L2 address is
    also stored as part of this state information.

    b) OwnAddrInfo -- SBM's own IP address and L2 address for the
    interface and its own priority (a configuration parameter).

    c) RefreshInterval in seconds. When the DSBM is not yet elected,
    it is set to a default value specified as a configuration
    parameter.

    d) DSBMDeadInterval in seconds. When the DSBM is not yet elected,
    it is initially set to a default value specified as a
    configuration parameter.

    f) ListenInterval in seconds -- a configuration parameter that
    decides how long a SBM spends in the DetectDSBM state (see
    below).

    g) ElectionInterval in seconds -- a configuration parameter that
    decides how long a SBM spends in the ElectDSBM state when it has
    declared its candidacy.

   Figure 3 shows the state transition diagram for the election protocol
   and the various states are described below. A complete description of
   the state machine is provided in Section A.10.

A.7. DSBM Election States

      DOWN -- SBM is not operational.

      DetectDSBM -- typically, the initial state of a SBM when it
      starts up. In this state, it checks to see whether a DSBM already
      exists in its domain.

      Idle -- SBM is in this state when no election is in progress and
      it is not the DSBM. In this state, SBM passively monitors the
      state of the DSBM.

      ElectDSBM -- SBM is in this state when a DSBM election is in
      progress.

      IAMDSBM -- SBM is in this state when it is the DSBM for the L2
      domain.

A.8. Events that cause state changes

      StartUp -- SBM starts operation.

      ListenInterval Timeout -- The ListenInterval timer has fired.
      This means that the SBM has monitored its domain to check for an
      existing DSBM or to check whether there are candidates (other
      than itself) willing to be the DSBM.

      DSBM_WILLING message received -- This means that the SBM received
      a DSBM_WILLING message from some other SBM. Such a message is
      sent when a SBM wishes to declare its candidacy to be the DSBM.

      I_AM_DSBM message received -- SBM received a DSBM advertisement
      from the DSBM in its L2 domain.

      DSBMDeadInterval Timeout -- The DSBMDeadIntervalTimer has fired.
      This means that the SBM did not receive even one DSBM
      advertisement during this period and indicates possible failure
      of the DSBM.

      RefreshInterval Timeout -- The RefreshIntervalTimer has fired. In
      the IAMDSBM state, this means it is the time for sending out the
      next DSBM advertisement. In the ElectDSBM state, the event means
      that it is the time to send out another DSBM_WILLING message.

        ElectionInterval Timeout -- The ElectionIntervalTimer has fired.
        This means that the SBM has waited long enough after declaring
        its candidacy to determine whether or not it succeeded.

A.9. State Transition Diagram (Figure 3)

```
                            +-----------+
           +--<-------------<-|DetectDSBM |---->------+
           |                  +----------+            |
           |                                          |
           |                                          |
           |                                          |
           |    +------------+       +---------+      |
        +->---|    Idle     |--<>---|ElectDSBM|--<--+
           |    +------------+       +---------+      |
                      |                    |
                      |                    |
                      |                    |
                      |     +-----------+  |
           +<<- +---|  IAMDSBM  |-<-+
                |    +-----------+
                |
                |    +-----------+
           +>>-| SHUTDOWN  |
                +-----------+
```
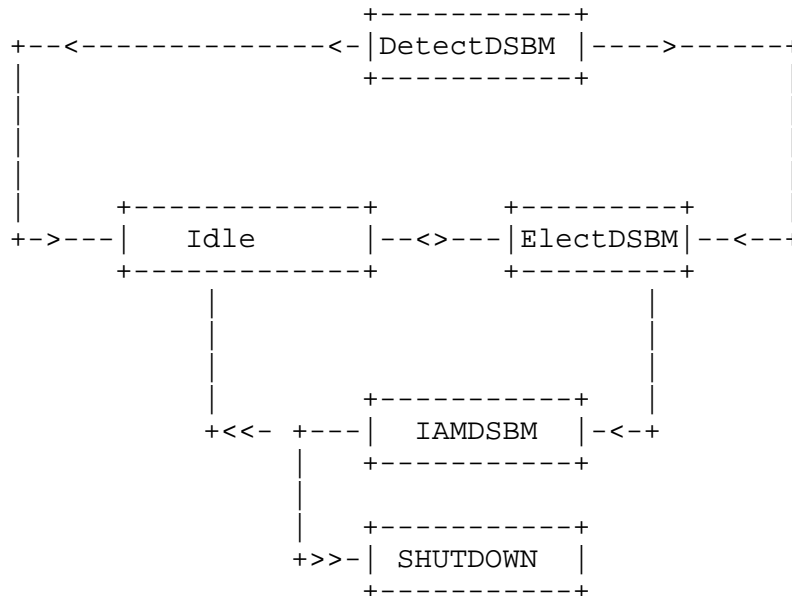
A.10. Election State Machine

   Based on the events and states described above, the state changes at
   a SBM are described below. Each state change is triggered by an event
   and is typically accompanied by a sequence of actions. The state
   machine is described assuming a single threaded implementation (to
   avoid race conditions between state changes and timer events) with no
   timer events occurring during the execution of the state machine.

   The following routines will be frequently used in the description of
   the state machine:

   ComparePrio(FirstAddrInfo, SecondAddrInfo)
     -- determines whether the entity represented by the first parameter
        is better than the second entity using the priority information
        and the IP address information in the two parameters. If any
        address is zero, that entity automatically loses; then first
        priorities are compared; higher priority candidate wins. If there
        is a tie based on the priority value, the tie is broken using the
        IP addresses of tied candidates (one with the higher IP address
        in the lexicographic order wins). Returns TRUE if first entity
        is a better choice. FALSE otherwise.

```
    SendDSBMWilling Message()
    Begin
        Send out DSBM_WILLING message listing myself as a candidate for
        DSBM (copy OwnAddr and priority into appropriate fields)
        start RefreshIntervalTimer
        goto ElectDSBM state
    End

    AmIBetterDSBM(OtherAddrInfo)
    Begin
        if (ComparePrio(OwnAddrInfo, OtherAddrInfo))
             return TRUE

        change LocalDSBMInfo = OtherDSBMAddrInfo
        return FALSE
    End

    UpdateDSBMInfo()
    /* invoked in an assignment such as LocalDSBMInfo = OtherAddrInfo */
    Begin
        update LocalDSBMInfo such as  IP addr, DSBM L2 address,
        DSBM priority, RefreshIntervalTimer, DSBMDeadIntervalTimer
    End
```

A.10.1 State Changes

   In the following, the action "continue" or "continue in current
   state" means an "exit" from the current action sequence without a
   state transition.

```
 State:      DOWN
 Event:      StartUp
 New State:  DetectDSBM
 Action:     Initialize the local state variables (LocalDSBMADDR and
             LocalDSBMAddrInfo set to 0). Start the ListenIntervalTimer.


 State:      DetectDSBM
 New State:  Idle
 Event:      I_AM_DSBM message received
 Action:     set LocalDSBMAddrInfo = IncomingDSBMAddrInfo
             start DeadDSBMInterval timer
             goto Idle State


 State:      DetectDSBM
 Event:      ListenIntervalTimer fired
 New State:  ElectDSBM
 Action:     Start ElectionIntervalTimer
             SendDSBMWillingMessage();
```

```
 State:       DetectDSBM
 Event:       DSBM_WILLING message received
 New State:   ElectDSBM
 Action:      Cancel any active timers

              Start ElectionIntervalTimer
              /* am I a better choice than this dude? */
              If (ComparePrio(OwnAddrInfo, IncomingDSBMInfo)) {
                  /* I am better */
                  SendDSBMWillingMessage()
              } else {
                  Change LocalDSBMAddrInfo = IncomingDSBMAddrInfo
                  goto ElectDSBM state
              }

 State:       Idle
 Event:       DSBMDeadIntervalTimer fired.
 New State:   ElectDSBM
 Action:      start ElectionIntervalTimer
              set LocalDSBMAddrInfo = OwnAddrInfo
              SendDSBMWiliingMessage()

 State:       Idle
 Event:       I_AM_DSBM message received.
 New State:   Idle
 Action:      /* first check whether anything has changed */
              if (!ComparePrio(LocalDSBMAddrInfo, IncomingDSBMAddrInfo))
                  change LocalDSBMAddrInfo to reflect new info
              endif
              restart DSBMDeadIntervalTimer;
              continue in current state;

 State:       Idle
 Event:       DSBM_WILLING Message is received
 New State:   Depends on action (ElectDSBM or Idle)
 Action:      /* check whether it is from the DSBM itself (shutdown) */
              if (IncomingDSBMAddr == LocalDSBMAddr) {
                  cancel active timers
                  Set LocalDSBMAddrInfo = OwnAddrInfo
                  Start ElectionIntervalTimer
                  SendDSBMWillingMessage() /* goto ElectDSBM state */
              }

              /* else, ignore it */
              continue in current state

 State:       ElectDSBM
 Event:       ElectionIntervalTimer Fired
```

```
 New State:  depends on action (IAMDSBM or Current State)
 Action:     If (LocalDSBMAddrInfo == OwnAddrInfo) {
                   /* I won */
                   send I_AM_DSBM message
                   start RefreshIntervalTimer
                   goto IAMDSBM state
             } else {   /* someone else won, so wait for it to declare
                           itself to be the DSBM */
                   set LocalDSBMAddressInfo = OwnAddrInfo
                   start ElectionIntervalTimer
                   SendDSBMWillingMessage()
                   continue in current state
             }


 State:      ElectDSBM
 Event:      I_AM_DSBM message received
 New State:  Idle
 Action:     set LocalDSBMAddrInfo = IncomingDSBMAddrInfo
             Cancel any active timers
             start DeadDSBMInterval timer
             goto Idle State


 State:      ElectDSBM
 Event:      DSBM_WILLING message received
 New State:  ElectDSBM
 Action:     Check whether it's a loopback and if so, discard, continue;
             if (!AmIBetterDSBM(IncomingDSBMAddrInfo)) {
                 Change LocalDSBMAddrInfo = IncomingDSBMAddrInfo
                 Cancel RefreshIntervalTimer
             } else if (LocalDSBMAddrInfo == OwnAddrInfo) {
                 SendDSBMWillingMessage()
             }
             continue in current state


 State:      ElectDSBM
 Event:      RefreshIntervalTimer fired
 New State:  ElectDSBM
 Action:     /* continue to send DSBMWilling messages until
                election interval ends */
             SendDSBMWillingMessage()


 State:      IAMDSBM
 Event:      DSBM_WILLING message received
 New State:  depends on action (IAMDSBM or SteadyState)
 Action:     /* check whether other guy is better */
             If (ComparePrio(OwnAddrInfo, IncomingAddrInfo)) {
             /* I am better */
                 send I_AM_DSBM message
```

```
                  restart RefreshIntervalTimer
                  continue in current state
              } else {
                  Set LocalDSBMAddrInfo = IncomingAddrInfo
                  cancel active timers
                  start DSBMDeadIntervalTimer
                  goto SteadyState
              }
```

```
 State:        IAMDSBM
 Event:        RefreshIntervalTimer fired
 New State:    IAMDSBM
 Action:       send I_AM_DSBM message
               restart RefreshIntervalTimer
```

```
 State:        IAMDSBM
 Event:        I_AM_DSBM message received
 New State:    depends on action (IAMDSBM or Idle)
 Action:       /* check whether other guy is better */
               If (ComparePrio(OwnAddrInfo, IncomingAddrInfo))  {
                   /* I am better */
                   send I_AM_DSBM message
                   restart RefreshIntervalTimer
                   continue in current state
               } else {
                   Set LocalDSBMAddrInfo = IncomingAddrInfo
                   cancel active timers
                   start DSBMDeadIntervalTimer
                   goto Idle State
               }
```

```
 State:        IAMDSBM
 Event:        Want to shut myself down
 New State:    DOWN
 Action:       send DSBM_WILLING message with My address filled in, but
               priority set to zero
               goto Down State
```

A.10.2 Suggested Values of Interval Timers

   To avoid DSBM outages for long period, to ensure quick recovery from
   DSBM failures, and to avoid timeout of PATH and RESV state at the
   edge devices, we suggest the following values for various timers.

   Assuming that the RSVP implementations use a 30 second timeout for
   PATH and RESV refreshes, we suggest that the RefreshIntervalTimer
   should be set to about 5 seconds with DSBMDeadIntervalTimer set to 15
   seconds (K=3, K*RefreshInterval). The DetectDSBMTimer should be set

to a random value between (DSBMDeadIntervalTimer,
2*DSBMDeadIntervalTimer). The ElectionIntervalTimer should be set at
least to the value of DSBMDeadIntervalTimer to ensure that each SBM
has a chance to have its DSBM_WILLING message (sent every
RefreshInterval in ElectDSBM state) delivered to others.

A.10.3. Guidelines for Choice of Values for SBM_PRIORITY

   Network administrators should configure SBM protocol entity at each
   SBM-capable device with the device's "SBM priority" for each of the
   interfaces attached to a managed segment. SBM_PRIORITY is an 8-bit,
   unsigned integer value (in the range 0-255) with higher integer
   values denoting higher priority. The value zero for an interface
   indicates that the SBM protocol entity on the device is not eligible
   to be a DSBM for the segment attached to the interface.

   A separate range of values is reserved for each type of SBM-capable
   device to reflect the relative priority among different classes of
   L2/L3 devices. L2 devices get higher priority followed by routers
   followed by hosts. The priority values in the range of 128..255 are
   reserved for L2 devices, the values in the range of 64..127 are
   reserved for routers, and values in the range of 1..63 are reserved
   for hosts.

A.11. DSBM Election over switched links

   The election algorithm works as described before in this case except
   each SBM-capable L2 device restricts the scope of the election to its
   local segment. As described in Section B.1 below, all messages
   related to the DSBM election are sent to a special multicast address
   (AllSBMAddress). AllSBMAddress (its corresponding MAC multicast
   address) is configured in the permanent database of SBM-capable,
   layer 2 devices so that all frames with AllSBMAddress as the
   destination address are not forwarded and instead directed to the SBM
   management entity in those devices. Thus, a DSBM can be elected
   separately on each point-to-point segment in a switched topology. For
   example, in Figure 2, DSBM for "segment A" will be elected using the
   election algorithm between R1 and S1 and none of the election-related
   messages on this segment will be forwarded by S1 beyond "segment A".
   Similarly, a separate election will take place on each segment in
   this topology.

   When a switched segment is a half-duplex segment, two senders (one
   sender at each end of the link) share the link. In this case, one of
   the two senders will win the DSBM election and will be responsible
   for managing the segment.

If a switched segment is full-duplex, exactly one sender sends on the
link in each direction. In this case, either one or two DSBMs can
exist on such a managed segment. If a sender at each end wishes to
serve as a DSBM for that end, it can declare itself to be the DSBM by
sending out an I_AM_DSBM advertisement and start managing the
resources for the outgoing traffic over the segment. If one of the
two senders does not wish itself to be the DSBM, then the other DSBM
will not receive any DSBM advertisement from its peer and assume
itself to be the DSBM for traffic traversing in both directions over
the managed segment.

Appendix B Message Encapsulation and Formats

   To minimize changes to the existing RSVP implementations and to
   ensure quick deployment of a SBM in conjunction with RSVP, all
   communication to and from a DSBM will be performed using messages
   constructed using the current rules for RSVP message formats and raw
   IP encapsulation. For more details on the RSVP message formats, refer
   to the RSVP specification (RFC 2205). No changes to the RSVP message
   formats are proposed, but new message types and new L2-specific
   objects are added to the RSVP message formats to accommodate DSBM-
   related messages. These additions are described below.

B.1 Message Addressing

   For the purpose of DSBM election and detection, AllSBMAddress is used
   as the destination address while sending out both DSBM_WILLING and
   I_AM_DSBM messages. A DSBM client first detects a managed segment by
   listening to I_AM_DSBM advertisements and records the DSBMAddress
   (unicast IP address of the DSBM).

B.2. Message Sizes

   Each message must occupy exactly one IP datagram. If it exceeds the
   MTU, such a datagram will be fragmented by IP and reassembled at the
   recipient node. This has a consequence that a single message may not
   exceed the maximum IP datagram size, approximately 64K bytes.

B.3. RSVP-related Message Formats

   All RSVP messages directed to and from a DSBM may contain various
   RSVP objects defined in the RSVP specification and messages continue
   to follow the formatting rules specified in the RSVP specification.
   In addition, an RSVP implementation must also recognize new object
   classes that are described below.

B.3.1. Object Formats

   All objects are defined using the format specified in the RSVP
   specification. Each object has a 32-bit header that contains length
   (of the object in bytes including the object header), the object
   class number, and a C-Type. All unused fields should be set to zero
   and ignored on receipt.

B.3.2. SBM Specific Objects

   Note that the Class-Num values for the SBM specific objects
   (LAN_NHOP, LAN_LOOPBACK, and RSVP_HOP_L2) are chosen from the
   codespace 10XXXXXX. This coding assures that non-SBM aware RSVP nodes
   will ignore the objects without forwarding them or generating an
   error message.

   Within the SBM specific codespace, note the following interpretation
   of the third most significant bit of the Class-Num:

          a) Objects of the form 100XXXXX are to be silently
             discarded by SBM nodes that do not recognize them.

          b) Objects of the form 101XXXXX are to be silently
             forwarded by SBM nodes that do not recognize them.

B.3.3. IEEE 802 Canonical Address Format

   The 48-bit MAC Addresses used by IEEE 802 were originally defined in
   terms of wire order transmission of bits in the source and
   destination MAC address fields. The same wire order applied to both
   Ethernet and Token Ring. Since the bit transmission order of Ethernet
   and Token Ring data differ - Ethernet octets are transmitted least
   significant bit first, Token Ring most significant first - the
   numeric values naturally associated with the same address on
   different 802 media differ. To facilitate the communication of
   address values in higher layer protocols which might span both token
   ring and Ethernet attached systems connected by bridges, it was
   necessary to define one reference format - the so called canonical
   format for these addresses. Formally the canonical format defines the
   value of the address, separate from the encoding rules used for
   transmission. It comprises a sequence of octets derived from the
   original wire order transmission bit order as follows. The least
   significant bit of the first octet is the first bit transmitted, the
   next least significant bit the second bit, and so on to the most
   significant bit of the first octet being the 8th bit transmitted; the
   least significant bit of the second octet is the 9th bit transmitted,
   and so on to the most significant bit of the sixth octet of the
   canonical format being the last bit of the address transmitted.

   This canonical format corresponds to the natural value of the address
   octets for Ethernet. The actual transmission order or formal encoding
   rules for addresses on media which do not transmit bit serially are
   derived from the canonical format octet values.

This document requires that all L2 addresses used in conjunction with
the SBM protocol be encoded in the canonical format as a sequence of
6 octets. In the following, we define the object formats for objects
that contain L2 addresses that are based on the canonical
representation.

B.3.4. RSVP_HOP_L2 object

RSVP_HOP_L2 object uses object class = 161; it contains the L2
address of the previous hop L3 device in the IEEE Canonical address
format discussed above.

RSVP_HOP_L2 object: class = 161, C-Type represents the addressing
format used. In our case, C-Type=1 represents the IEEE Canonical
Address format.

```
       0               1               2               3
+---------------+---------------+---------------+---------------+
|            Length             |      161      |C-Type(addrtype)|
+---------------+---------------+---------------+---------------+
|                  Variable length Opaque data                  |
+---------------+---------------+---------------+---------------+
```

C-Type = 1 (IEEE Canonical Address format)

When C-Type=1, the object format is:

```
       0               1               2               3
+---------------+---------------+---------------+---------------+
|            12                 |      161      |      1        |
+---------------+---------------+---------------+---------------+
|                  Octets 0-3 of the MAC address                |
+---------------+---------------+---------------+---------------+
|  Octets 4-5 of the MAC addr.  |      ///      |     ///       |
+---------------+---------------+---------------+---------------+
```

/// -- unused (set to zero)

B.3.5. LAN_NHOP object

LAN_NHOP object represents two objects, namely, LAN_NHOP_L3 address
object and LAN_NHOP_L2 address object.
      <LAN_NHOP object> ::= <LAN_NHOP_L2 object> <LAN_NHOP_L3 object>

LAN_NHOP_L2 address object uses object class = 162 and uses the same
format (but different class number) as the RSVP_HOP_L2 object. It
provides the L2 or MAC address of the next hop L3 device.

```
            0                 1                 2                 3
    +---------------+---------------+---------------+---------------+
    |          Length               |      162      |C-Type(addrtype)|
    +---------------+---------------+---------------+---------------+
    |                Variable length Opaque data                   |
    +---------------+---------------+---------------+---------------+
```

   C-Type = 1 (IEEE 802 Canonical Address Format as defined below) See
   the RSVP_HOP_L2 address object for more details.

   LAN_NHOP_L3 object uses object class = 163 and gives the L3 or IP
   address of the next hop L3 device.

   LAN_NHOP_L3 object: class = 163, C-Type specifies IPv4 or IPv6
   address family used.

   IPv4 LAN_NHOP_L3 object: class =163, C-Type = 1

```
    +---------------+---------------+---------------+---------------+
    |          Length = 8           |      163      |      1        |
    +---------------+---------------+---------------+---------------+
    |                     IPv4 NHOP address                        |
    +-------------------------------------------------------------+
```

   IPv6 LAN_NHOP_L3 object: class =163, C-Type = 2

```
    +---------------+---------------+---------------+---------------+
    |          Length = 20          |      163      |      2        |
    +---------------+---------------+---------------+---------------+
    |                  IPv6 NHOP address (16 bytes)                |
    +-------------------------------------------------------------+
```
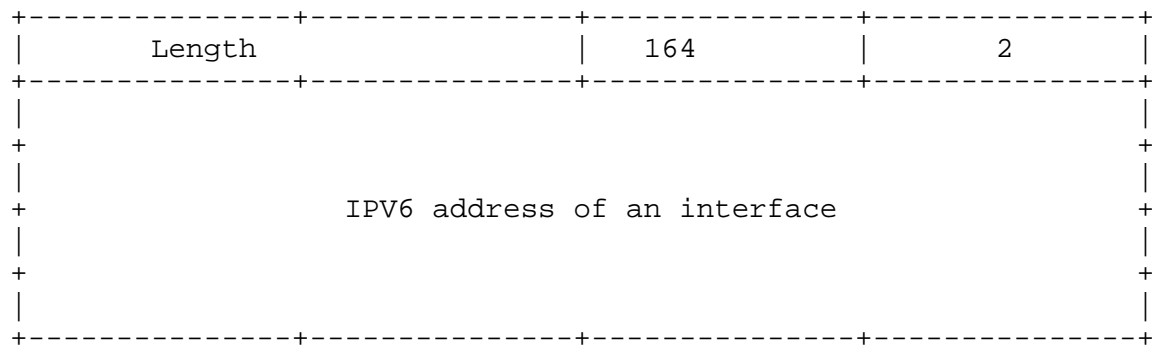
B.3.6. LAN_LOOPBACK Object

   The LAN_LOOPBACK object gives the IP address of the outgoing
   interface for a PATH message and uses object class=164; both IPv4 and
   IPv6 formats are specified.
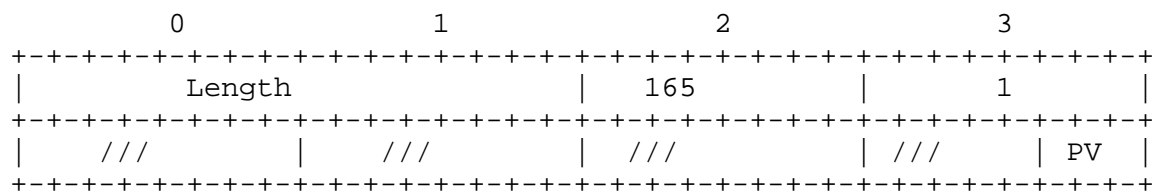
   IPv4 LAN_LOOPBACK object: class = 164, C-Type = 1

```
            0                 1                 2                 3
    +---------------+---------------+---------------+---------------+
    |          Length               |      164      |      1        |
    +---------------+---------------+---------------+---------------+
    |                IPV4 address of an interface                  |
    +---------------+---------------+---------------+---------------+
```

IPv6 LAN_LOOPBACK object: class = 164, C-Type = 2

```
+---------------+---------------+--------------+--------------+
|        Length                 |      164     |      2       |
+---------------+---------------+--------------+--------------+
|                                                             |
+                                                             +
|                                                             |
+             IPV6 address of an interface                    +
|                                                             |
+                                                             +
|                                                             |
+---------------+---------------+--------------+--------------+
```

B.3.7. TCLASS Object

TCLASS object (traffic class based on IEEE 802.1p) uses object
class = 165.

```
         0               1               2               3
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|          Length               |      165     |      1       |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
|    ///        |     ///       |     ///      |   ///   | PV  |
+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Only 3 bits in data contain the user_priority value (PV).

B.4. RSVP PATH and PATH_TEAR Message Formats

As specified in the RSVP specification, a PATH and PATH_TEAR messages
contain the RSVP Common Header and the relevant RSVP objects.

For the RSVP Common Header, refer to the RSVP specification (RFC
2205). Enhancements to an RSVP_PATH message include additional
objects as specified below.

```
<PATH Message> ::= <RSVP Common Header> [<INTEGRITY>]
                <RSVP_HOP_L2> <LAN_NHOP>
                <LAN_LOOPBACK> [<TCLASS>]  <SESSION><RSVP_HOP>
                <TIME_VALUES> [<POLICY DATA>] <sender descriptor>
```

```
<PATH_TEAR Message> ::= <RSVP Common Header> [<INTEGRITY>]
                <LAN_LOOPBACK> <LAN_NHOP> <SESSION> <RSVP_HOP>
                [<sender descriptor>]
```

If the INTEGRITY object is present, it must immediately follow the
RSVP common header. L2-specific objects must always precede the
SESSION object.

B.5. RSVP RESV Message Format

As specified in the RSVP specification, an RSVP_RESV message contains
the RSVP Common Header and relevant RSVP objects. In addition, it may
contain an optional TCLASS object as described earlier.

B.6. Additional RSVP message types to handle SBM interactions

New RSVP message types are introduced to allow interactions between a
DSBM and an RSVP node (host/router) for the purpose of discovering
and binding to a DSBM. New RSVP message types needed are as follows:

```
RSVP Msg Type (8 bits)       Value
DSBM_WILLING                 66
I_AM_DSBM                    67
```

All SBM-specific messages are formatted as RSVP messages with an RSVP
common header followed by SBM-specific objects.

<SBMP_MESSAGE> ::= <SBMP common header> <SBM-specific objects>

where <SBMP common header> ::= <RSVP common Header> [<INTEGRITY>]

For each SBM message type, there is a set of rules for the
permissible choice of object types. These rules are specified using

Backus-Naur Form (BNF) augmented with square brackets surrounding
optional sub-sequences. The BNF implies an order for the objects in a
message. However, in many (but not all) cases, object order makes no
logical difference. An implementation should create messages with the
objects in the order shown here, but accept the objects in any
permissible order. Any exceptions to this rule will be pointed out in
the specific message formats.

DSBM_WILLING Message

<DSBM_WILLING message> ::= <SBM Common Header> <DSBM IP ADDRESS>
                           <DSBM L2 address> <SBM PRIORITY>

I_AM_DSBM Message

<I_AM_DSBM> ::= <SBM Common Header> <DSBM IP ADDRESS> <DSBM L2 address>
                <SBM PRIORITY> <DSBM Timer Intervals>
                [<NON_RESV_SEND_LIMIT>]

For compatibility reasons, receivers of the I_AM_DSBM message must be
prepared to receive additional objects of the Unknown Class type
[RFC-2205].

All I_AM_DSBM messages are multicast to the well known AllSBMAddress.
The default priority of a SBM is 1 and higher priority values
represent higher precedence. The priority value zero indicates that
the SBM is not eligible to be the DSBM.

Relevant Objects

DSBM IP ADDRESS objects use object class = 42; IPv4 DSBM IP ADDRESS
object uses <Class=42, C-Type=1> and IPv6 DSBM IP ADDRESS object uses
<Class=42, C-Type=2>.

IPv4 DSBM IP ADDRESS object: class = 42, C-Type =1
```
          0               1               2               3
   +--------------+--------------+--------------+--------------+
   |                   IPv4 DSBM IP Address                    |
   +--------------+--------------+--------------+--------------+
```

IPv6 DSBM IP ADDRESS object: Class = 42, C-Type = 2

```
   +--------------+--------------+--------------+--------------+
   |                                                          |
   +                                                          +
   |                                                          |
   +                   IPv6 DSBM IP Address                   +
   |                                                          |
   +                                                          +
   |                                                          |
   +--------------+--------------+--------------+--------------+
```

<DSBM L2 address> Object is the same as <RSVP_HOP_L2> object with C-
Type = 1 for IEEE Canonical Address format.

<DSBM L2 address> ::= <RSVP_HOP_L2>

A SBM may omit this object by including a NULL L2 address object.
For C-Type=1 (IEEE Canonical address format), such a version of the
L2 address object contains value zero in the six octets corresponding
to the MAC address (see section B.3.4 for the exact format).

SBM_PRIORITY Object: class = 43, C-Type =1

```
            0               1               2               3
+---------------+---------------+---------------+---------------+
|    ///        |    ///        |    ///        | SBM priority  |
+---------------+---------------+---------------+---------------+
```

TIMER INTERVAL VALUES.

The two timer intervals, namely, DSBM Dead Interval and DSBM Refresh
Interval, are specified as integer values each in the range of 0..255
seconds. Both values are included in a single "DSBM Timer Intervals"
object described below.

DSBM Timer Intervals Object: class = 44, C-Type =1

```
+---------------+---------------+---------------+---------------+
|    ///        |    ///        | DeadInterval  | RefreshInterval|
+---------------+---------------+---------------+---------------+
```

NON_RESV_SEND_LIMIT Object: class = 45, C-Type = 1

```
     0       1       2       3
+---------------+---------------+---------------+---------------+
|  NonResvSendLimit(limit on traffic allowed to send without RESV)|
|                                                                 |
+---------------+---------------+---------------+---------------+
```

<NonResvSendLimit> ::= <Intserv Sender_TSPEC object>
(class=12, C-Type =2)

The NON_RESV_SEND_LIMIT object specifies a per-flow limit on the
profile of traffic which a sending host is allowed to send onto a
managed segment without a valid RSVP reservation (see Appendix C for
further details on the usage of this object). The object contains the
NonResvSendLimit parameter. This parameter is equivalent to the
Intserv SENDER_TSPEC (see RFC 2210 for contents and encoding rules).
The SENDER_TSPEC includes five parameters which describe a traffic
profile (r, b, p, m and M). Sending hosts compare the SENDER_TSPEC
describing a sender traffic flow to the SENDER_TSPEC advertised by
the DSBM. If the SENDER_TSPEC of the traffic flow in question is less
than or equal to the SENDER_TSPEC advertised by the DSBM, it is
allowable to send traffic on the corresponding flow without a valid
RSVP reservation in place. Otherwise it is not.

The network administrator may configure the DSBM to disallow any sent
traffic in the absence of an RSVP reservation by configuring a
NonResvSendLimit in which r = 0, b = 0, p = 0, m = infinity and M =

0. Similarly the network administrator may allow any traffic to be
sent in the absence of an RSVP reservation by configuring a
NonResvSendLimit in which r = infinity, b = infinity, p = infinity, m
= 0 and M = infinity. Of course, any of these parameters may be set
to values between zero and infinity to advertise finite per-flow
limits.

The NON_RESV_SEND_LIMIT object is optional. Senders on a managed
segment should interpret the absence of the NON_RESV_SEND_LIMIT
object as equivalent to an infinitely large SENDER_TSPEC (it is
permissible to send any traffic profile in the absence of an RSVP
reservation).

Appendix C The DSBM as a Source of Centralized Configuration Information

   There are certain configuration parameters which it may be useful to
   distribute to layer-3 senders on a managed segment. The DSBM may
   serve as a centralized management point from which such parameters
   can easily be distributed. In particular, it is possible for the
   network administrator configuring a DSBM to cause certain
   configuration parameters to be distributed as objects appended to the
   I_AM_DSBM messages. The following configuration object is defined at
   this time. Others may be defined in the future. See Appendix B for
   further details regarding the NON_RESV_SEND_LIMIT object.

C.1. NON_RESV_SEND_LIMIT

   As we QoS enable layer 2 segments, we expect an evolution from
   subnets comprised of traditional shared segments (with no means of
   traffic separation and no DSBM), to subnets comprised of dedicated
   segments switched by sophisticated switches (with both DSBM and
   802.1p traffic separation capability).

   A set of intermediate configurations consists of a group of QoS
   enabled hosts sending onto a traditional shared segment. A layer-3
   device (or a layer-2 device) acts as a DSBM for the shared segment,
   but cannot enforce traffic separation. In such a configuration, the
   DSBM can be configured to limit the number of reservations approved
   for senders on the segment, but cannot prevent them from sending. As
   a result, senders may congest the segment even though a network
   administrator has configured an appropriate limit for admission
   control in the DSBM.

   One solution to this problem which would give the network
   administrator control over the segment, is to require applications
   (or operating systems on behalf of applications) not to send until
   they have obtained a reservation. This is problematic as most
   applications are used to sending as soon as they wish to and expect
   to get whatever service quality the network is able to grant at that
   time. Furthermore, it may often be acceptable to allow certain
   applications to send before a reservation is received. For example,
   on a segment comprised of a single 10 Mbps ethernet and 10 hosts, it
   may be acceptable to allow a 16 Kbps telephony stream to be
   transmitted but not a 3 Mbps video stream.

   A more pragmatic solution then, is to allow the network administrator
   to set a per-flow limit on the amount of non-adaptive traffic which a
   sender is allowed to generate on a managed segment in the absence of
   a valid reservation. This limit is advertised by the DSBM and
   received by sending hosts. An API on the sending host can then
   approve or deny an application's QoS request based on the resources

   requested.

   The NON_RESV_SEND_LIMIT object can be used to advertise a Flowspec
   which describes the shape of traffic that a sender is allowed to
   generate on a managed segment when its RSVP reservation requests have
   either not yet completed or have been rejected.

ACKNOWLEDGEMENTS

   Authors are grateful to Eric Crawley (Argon), Russ Fenger (Intel),
   David Melman (Siemens), Ramesh Pabbati (Microsoft), Mick Seaman
   (3COM), Andrew Smith (Extreme Networks) for their constructive
   comments on the SBM design and the earlier versions of this document.

6. Authors' Addresses

   Raj Yavatkar
   Intel Corporation
   2111 N.E. 25th Avenue,
   Hillsboro, OR 97124
   USA

   Phone: +1 503-264-9077
   EMail: yavatkar@ibeam.intel.com

   Don Hoffman
   Teledesic Corporation
   2300 Carillon Point
   Kirkland, WA 98033
   USA

   Phone: +1 425-602-0000

   Yoram Bernet
   Microsoft
   1 Microsoft Way
   Redmond, WA 98052
   USA

   Phone: +1 206 936 9568
   EMail: yoramb@microsoft.com

Fred Baker
Cisco Systems
519 Lado Drive
Santa Barbara, California 93111
USA

Phone: +1 408 526 4257
EMail: fred@cisco.com

Michael Speer
Sun Microsystems, Inc
901 San Antonio Road UMPK15-215
Palo Alto, CA 94303

Phone: +1 650-786-6368
EMail: speer@Eng.Sun.COM

Full Copyright Statement

Acknowledgement