Network WG James Polk
Internet-Draft Subha Dhesikan
Expires: January 4, 2012 Cisco Systems
Intended Status: Standards Track (PS) July 4, 2011
Updates: RFC 2205, 2210, & 4495 (if published as an RFC)

Integrated Services (IntServ) Extension to Allow Signaling of Multiple Traffic Specifications and Multiple Flow Specifications in RSVPv1

draft-ietf-tsvwg-intserv-multiple-tspec-00

Abstract

This document defines extensions to Integrated Services (IntServ) allowing multiple traffic specifications and multiple flow specifications to be conveyed in the same Resource Reservation Protocol (RSVPv1) reservation message exchange. This ability helps optimize an agreeable bandwidth through a network between endpoints in a single round trip.

This Internet-Draft will expire on January 4, 2012.

Copyright Notice

Table of Contents

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT",

"SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC 2119].

1. Introduction

This document defines how Integrated Services (IntServ) [RFC2210] includes multiple traffic specifications and multiple flow specifications in the same Resource Reservation Protocol (RSVPv1) [RFC2205] message. This ability helps optimize an agreeable bandwidth through a network between endpoints in a single round trip.

There is a separation of function between RSVP and IntServ, in which RSVP does not define the internal objects to establish controlled load or guarantee services. These are generally left to be opaque in RSVP. At the same time, IntServ does not require that RSVP be the only reservation protocol for transporting both the controlled load or guaranteed service objects - but RSVP does often carry the objects anyway. This makes the two independent - yet related in usage, but are also frequently talked about as if they are one and the same. They are not.

The 'traffic specification' contains the traffic characteristics of a sender's data flow and is a required object in a PATH message. The TSPEC object is defined in RFC 2210 to convey the traffic specification from the sender and is opaque to RSVP. The ADSPEC object - for 'advertising specification' - is used to gather information along the downstream data path to aid the receiver in the computation of QoS properties of this data path. The ADSPEC is also opaque to RSVP and is defined in RFC 2210. Both of these IntServ objects are part of the Sender Descriptor [RFC2205].

Once the Sender Descriptor is received at its destination node, after having traveled through the network of routers, the SENDER_TSPEC information is matched with the information gathered in the ADSPEC, if present, about the data path. Together, these two objects help the receiver build its flow specification (encoded in the FLOWSPEC object) for the RESV message. The RESV message establishes the reservation through the network of routers on the data path established by the PATH message. If the ADSPEC is not present in the Sender_Descriptor, it cannot aid the receiver in building the flow specification.

The SENDER_TSPEC is not changed in transit between endpoints (i.e., there are no bandwidth request adjustments along the way). However, the ADSPEC is changed, based on the conditions experienced through the network (i.e., bandwidth availability within each router) as the RSVP message travels hop-by-hop.

Today, real-time applications have evolved such that they are able to dynamically adapt to available bandwidth, not only by dropping and adding layers, but also by reducing frame rates and resolution. It is therefore limiting to have a single bandwidth request in Integrated Services, and by extension, RSVP.

With only one traffic specification in a PATH message and only one flow specification in a RESV message (with some styles of reservations a RESV message may actually contain multiple flow specifications, but then there is only one per sender), applications will either have to give up altogether on session establishment in case of failure of the reservation establishment for the highest "bandwidth or will have to resort to multiple successive RSVP signaling attempts in a trial-and-error manner until they finally establish the reservation a lower "bandwidth". These multiple signaling round-trip would affect the session establishment time and in turn would negatively impact the end user experience.

The objective of this document is to avoid such roundtrips as well as allow applications to successfully receive some level of bandwidth allotment that it can use for its sessions.

While the ADSPEC provides an indication of the bandwidth available along the path and can be used by the receiver in creating the FLOWSPEC, it does not prevent failures or multiple round-trips as described above. The intermediary routers provide a best attempt estimate of available bandwidth in the ADSPEC object. However, it does not take into account external policy considerations (RFC 2215). In addition, the available bandwidth at the time of creating the ADSPEC may not be available at the time of an actual request in an RESV message. These reasons may cause the RESV message to be rejected. Therefore, the ADSPEC object cannot, by itself, satisfy the requirements of the current generations of real-time applications.

It needs to be noted that the ADSPEC is unchanged by this new mechanism. If ADSPEC is included in the PATH message, it is suggested that the receiver use this object in determining the flow specification.

This document creates a means for conveying more than one "bandwidth" within the same RSVP reservation set-up (both PATH and RESV) messages to optimize the determination of an agreed upon bandwidth for this reservation. Allowing multiple traffic specifications within the same PATH message allows the sender to communicate to the receiver multiple "bandwidths" that match the different sending rates that the sender is capable of transmitting at. This allows the receiver to convey this multiple "bandwidths" in the RESV so those can be considered when RSVP makes the actual reservation admission into the network. This allows the applications to dynamically adapt their data stream to available network resources.

The concept of RSVP signaling is shown in a single direction below, in Figure 1. Although the TSPEC is opaque to RSVP, it is shown along with the RSVP messages for completeness. The RSVP messages themselves need not be the focus of the reader. Instead, the number of round trips it takes to establish a reservation is the focus here.

```
   Sender          Rtr-1      Rtr-2 ...  Rtr-N         Receiver
     |               |          |          |               |
     |           PATH (with a TSPEC & ADSPEC)              |
     |-------------->|--------->|----//--->|-------------->|
     |               |          |          |               |
     |                            RESV (with a FLOWSPEC)   |
     |<--------------|<---------|<---//----|<--------------|
     |               |          |          |               |
```

Figure 1. Concept of RSVP in a Single Direction

Figure 1 shows a successful one-way reservation using RSVP and IntServ.

Figure 2 shows a scenario where the RESV message, containing a FLOWSPEC, which is generated by the Receiver, after considering both the Sender TSPEC and the ADSPEC, is rejected by an intermediary router.

```
   Sender          Rtr-1      Rtr-2 ...  Rtr-N         Receiver
     |               |          |          |               |
     |           PATH (with 1 TSPEC wanting 12Mbps)        |
     |-------------->|--------->|----//--->|-------------->|
     |               |          |          |               |
     |              RESV (with 1 FLOWSPEC wanting 12Mbps)  |
     |               |          X <--//----|<--------------|
     |               |          |          |               |
     |           ResvErr (with Admission control Error=2)  |
     |               |          |----//--->|-------------->|
     |               |          |          |               |
```

Figure 2. Concept of RSVP Rejection due to Limited Bandwidth

The scenario above is where multiple TSPEC and multiple FLOWSPEC optimization helps. The Sender may support multiple bandwidths for a given application (i.e., more than one codec for voice or video) and therefore might want to establish a reservation with the highest (or best) bandwidth that the network can provide for a particular codec.

For example, bandwidths of:

    12Mbps,
     4Mbps, and
     1.5Mbps

for the three video codecs the Sender supports.

This document will discuss the overview of the proposal to include multiple TSPECs and FLOWSPECs RSVP in section 2. In section 3, the overview of the entire solution is provided. This section also contains the new parameters which are defined in this document. The multiple TSPECs in a PATH message and the multiple FLOWSPEC in a RESV message, both for controlled load and guaranteed service are described in this section. Section 4 will cover the rules of usage of this IntServ extension. This section contains how this document needs to extend the scenario of when a router in the middle of a reservation cannot accept a preferred bandwidth (i.e., FLOWSPEC), meaning previous routers that accepted that greater bandwidth now have too much bandwidth reserved. This requires an extension to RFC 4495 (RSVP Bandwidth Reduction) to cover reservations being established, as well as existing reservations. Section 4 also includes the merging rules.

2. Overview of Proposal for Including Multiple TSPECs and FLOWSPECS

Presently, this is the format of a PATH message [RFC2205]:

```
        <PATH Message> ::= <Common Header> [ <INTEGRITY> ]

                                  <SESSION> <RSVP_HOP>

                                  <TIME_VALUES>

                                 [ <POLICY_DATA> ... ]

                                 [ <sender descriptor> ]

        <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                                                   ^^^^^^^^^^^^
                                 [ <ADSPEC> ]
```

where the SENDER_TSPEC object contains a single traffic specification.

For the PATH message, the focus of this document is to modify the <sender_descriptor> in such a way to include more than one traffic specification. This solution does this by retaining the existing SENDER_TSPEC object above, highlighted by the '^^^^' characters, and complementing it with a new optional MULTI_TSPEC object to convey additional traffic specifications in this PATH message. No other object within the PATH message is affected by this IntServ extension.

This extension modifies the sender descriptor by specifically augmenting it to allow an optional <MULTI_TSPEC> object after the optional <ADSPEC>, as shown below.

```
           <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>

                                 [ <ADSPEC> ]  [ <MULTI_TSPEC> ]
                                                 ^^^^^^^^^^^
```

As can be seen above, the MULTI_TSPEC is in addition to the SENDER_TSPEC - and is only to be used, per this extension, when more than one TSPEC is to be included in the PATH message.

Here is another way of looking at the proposal choices:

```
            +---------------------+
            |   Existing TSPEC    |
            |                     |
            |    +-----------+    |
            |    |  TSPEC1   |    |
            |    +-----------+    |
            |                     |
            +---------------------+

            +---------------------+
            |  Additional TSPECs  |
            |                     |
            |  +---------------+  |
            |  |  MULTI_TSPEC  |  |
            |  |    Object     |  |
            |  |  +--------+   |  |
            |  |  | TSPEC2 |   |  |
            |  |  +--------+   |  |
            |  |  +--------+   |  |
            |  |  | TSPEC3 |   |  |
            |  |  +--------+   |  |
            |  |  +--------+   |  |
            |  |  | TSPEC4 |   |  |
            |  |  +--------+   |  |
            |  +---------------+  |
            |                     |
            +---------------------+
```

 Figure 3. Encoding of Multiple Traffic Specifications in
           the TSPEC and MULTI_TSPEC objects

This solution is backwards compatible with existing implementations of [RFC2205] and [RFC2210], as the multiple TSPECs and FLOWSPECs are inserted as optional objects and such objects do not need to be processed, especially if they are not understood.

This solution defines a similar approach for encoding multiple flow specifications in the RESV message. Flow specifications beyond the first one can be encoded in a new "MULTI_FLOWSPEC" object contained in the RESV message.

In this proposal, the original SENDER_TSPEC and the FLOWSPEC are left untouched, allowing routers not supporting this extension to process the PATH and the RESV message without issue. Two new additional objects are defined in this document. They are the MULTI_TSPEC and the MULTI_FLOWSPEC for the PATH and the RESV message, respectively. The additional TSPECs (in the new MULTI_TSPEC Object) are included in the PATH and the additional FLOWSPECS (in the new MULTI_FLOWSPEC Object) are included in the RESV message as new (optional) objects. These additional objects will have a class number of 11bbbbbb, allowing older routers to ignore the object(s) and forward each unexamined and unchanged, as defined in section 3.10 of [RFC 2205].

NOTE: it is important to emphasize here that including more than one FLOWSPEC in the RESV message does not cause more than one FLOWSPEC to be granted. This document requires that the receiver arrange these multiple FLOWSPECs in the order of preference according to the order remaining from the MULTI_TSPECs in the PATH message. The benefit of this arrangement is that RSVP does not have to process the rest of the FLOWSPEC if it can admit the first one.

3. Multi_TSPEC and MULTI_FLOWSPEC Solution

For the Sender Descriptor within the PATH message, the original TSPEC remains where it is, and is untouched by this IntServ extension. What is new is the use of a new <MULTI_TSPEC> object inside the sender descriptor as shown here:

```
<sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>

                         [ <ADSPEC> ]  [ <MULTI_TSPEC> ]
                                         ^^^^^^^^^^^
```

The preferred order of TSPECs sent by the sender is this:

- preferred TSPEC is in the original SENDER_TSPEC
- the next in line preferred TSPEC is the first TSPEC in the MULTI_TSPEC object
- the next in line preferred TSPEC is the second TSPEC in the MULTI_TSPEC object
- and so on...

The composition of the flow descriptor list in a Resv message depends upon the reservation style. Therefore, the following shows the inclusion of the MULTI_FLOWSPEC object with each of the styles:

```
WF Style:
          <flow descriptor list> ::=  <WF flow descriptor>

          <WF flow descriptor> ::= <FLOWSPEC> [MULTI_FLOWSPEC]

FF style:
          <flow descriptor list> ::=

                    <FLOWSPEC>  <FILTER_SPEC>  [MULTI_FLOWSPEC] |

                    <flow descriptor list> <FF flow descriptor>

          <FF flow descriptor> ::=

                    [ <FLOWSPEC> ] <FILTER_SPEC> [MULTI_FLOWSPEC]

SE style:
          <flow descriptor list> ::= <SE flow descriptor>

          <SE flow descriptor> ::=

                   <FLOWSPEC> <filter spec list> [MULTI_FLOWSPEC]

          <filter spec list> ::=  <FILTER_SPEC>

                            |  <filter spec list> <FILTER_SPEC>
```

3.1 New MULTI_TSPEC and MULTI-RSPEC Parameters

This extension to Integrated Services defines two new parameters They are:

1. <parameter name> Multiple_Token_Bucket_Tspec, with a parameter number of 125.

2. <parameter name> Multiple_Guaranteed_Service_RSpec with a parameter number of 124

These are IANA registered in this document.

The original SENDER_TSPEC and FLOWSPEC for Controlled Service maintain the <parameter name> of Token_Bucket_Tspec with a parameter number of 127. The original FLOWSPEC for Guaranteed Service maintains the <parameter name> of Guaranteed_Service_RSpec with a parameter number of 130.

3.2 Multiple TSPEC in a PATH Message

Here is the object from [RFC2210]. It is used as a SENDER_TSPEC in a PATH message:

```
      31           24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1  | 0 (a) |    reserved           |             7 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2  |    X  (c)     |0| reserved    |             6 (d)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  3  |   127 (e)     |    0 (f)      |             5 (g)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  4  |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  5  |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  6  |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  7  |  Minimum Policed Unit [m] (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  8  |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 4. SENDER_TSPEC in PATH

```
  (a) - Message format version number (0)
  (b) - Overall length (7 words not including header)
  (c) - Service header, service number
          - '1' (Generic information) if in a PATH message;
  (d) - Length of service data, 6 words not including
        per-service header
  (e) - Parameter ID, parameter 127 (Token Bucket TSpec)
  (f) - Parameter 127 flags (none set)
  (g) - Parameter 127 length, 5 words not including per-service
        header
```

For completeness, Figure 4 is included in its original form for backwards compatibility reasons, as if there were only 1 TSPEC in the PATH. What is new when there are more than one TSPEC in this reservation message is the new MULTI_TSPEC object in Figure 5 containing, for example, 3 (Multiple_Token_Bucket_Tspec) TSPECs in a PATH message.

```
      31           24 23           16 15            8 7             0
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1  | 0 (a) |    reserved           |            19 (b)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2  |    5  (c)     |0| reserved    |            18 (d)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  3  |   125 (e)     |    0 (f)      |             5 (g)             |
```

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  4  |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
  5   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  6   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  7   |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  8   |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  9   |   125 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 10   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 11   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 12   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 13   |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 14   |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 15   |   125 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 16   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 17   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 18   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 19   |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 20   |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5. MULTI_TSPEC Object

(a) - Message format version number (0)
(b) - Overall length (19 words not including header)
(c) - Service header, service number 5 (Controlled-Load)
(d) - Length of service data, 18 words not including per-service header
(e) - Parameter ID, parameter 125 (Multiple Token Bucket TSpec)
(f) - Parameter 125 flags (none set)
(g) - Parameter 125 length, 5 words not including per-service header

Figure 5 shows the 2nd through Nth TSPEC in the PATH in the preferred order. The message format (a) remains the same for a second TSPEC and for other additional TSPECs.

The Overall Length (b) includes all the TSPECs within this object, plus the 2nd Word (containing fields (c) and (d)), which MUST NOT be repeated. The service header fields (e),(f) and(g) are repeated for each TSPEC.

The Service header, here service number 5 (Controlled-Load) MUST remain the same.

Each TSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd preferred), and Words 15-20 are the final TSPEC (and 4th preferred) in this example of 3 TSPECs in this MULTI_TSPEC object. There is no limit placed on the number of TSPECs a MULTI_TSPEC object can have. However, it is RECOMMENDED to administratively limit the number of TSPECs in the MULTI_TSPEC object to 9 (making for a total of 10 in the PATH message).

The TSPECS are included in the order of preference by the message generator (PATH) and MUST be maintained in that order all the way to the Receiver. The order of TSPECs that are still grantable, in conjunction with the ADSPEC at the Receiver, MUST retain that order in the FLOWSPEC and MULTI_FLOWSPEC objects.

3.3 Multiple FLOWSPEC for Controlled-Load service

The format of an RSVP FLOWSPEC object requesting Controlled-Load service is the same as the one used for the SENDER_TSPEC given in Figure 4.

The format of the new MULTI_FLOWSPEC object is given below:

```
        31           24 23           16 15            8 7             0
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   1   | 0 (a) |    reserved           |           19 (b)              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   2   |    5  (c)     |0| reserved    |           18 (d)              |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   3   |   125 (e)     |    0 (f)      |             5 (g)             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   4   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   5   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   6   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   7   |  Minimum Policed Unit [m] (32-bit integer)                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
   8   |  Maximum Packet Size [M]  (32-bit integer)                    |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
   9   |   125 (e)     |    0 (f)      |             5 (g)             |
       +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
10   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
11   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
12   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
13   |  Minimum Policed Unit [m] (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
14   |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
15   |   125 (e)     |    0 (f)      |             5 (g)             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
16   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
17   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
18   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
19   |  Minimum Policed Unit [m] (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
20   |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 5. Multiple FLOWSPEC for Controlled-Load service

(a) - Message format version number (0)
(b) - Overall length (19 words not including header)
(c) - Service header, service number 5 (Controlled-Load)
(d) - Length of controlled-load data, 18 words not including per-service header
(e) - Parameter ID, parameter 125 (Multiple Token Bucket TSpec)
(f) - Parameter 125 flags (none set)
(g) - Parameter 125 length, 5 words not including per-service header

This is for the 2nd through Nth TSPEC in the RESV, in the preferred order.

The message format (a) remains the same for a second TSPEC and for additional TSPECs.

The Overall Length (b) includes the TSPECs, plus the 2nd Word (fields (c) and (d)), which MUST NOT be repeated. The service header fields (e),(f) and(g), which are repeated for each TSPEC.

The Service header, here service number 5 (Controlled-Load) MUST remain the same for the RESV message. The services, Controlled-Load and Guaranteed MUST NOT be mixed within the same RESV message. In other words, if one TSPEC is a Controlled Load service TSPEC, the remaining TSPECs MUST be Controlled Load service. This same rule also is true for Guaranteed Service - if one TSPEC is for Guaranteed

Service, the rest of the TSPECs in this PATH or RESV MUST be for Guaranteed Service.

The Length of controlled-load data (d) also increases to account for the additional TSPECs.

Each FLOWSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd preferred), and Words 15-20 are the final TSPEC (and 4th preferred) in this example of 3 TSPECs in this FLOWSPEC. There is no limit placed on the number of TSPECs a particular FLOWSPEC can have.

Within the MULTI_FLOWSPEC, any SENDER_TSPEC that cannot be reserved - based on the information gathered in the ADSPEC, is not placed in the RESV or based on other information available to the receiver. Otherwise, the order in which the TSPECs were in the PATH message MUST be in the same order they are in the FLOWSPEC in the RESV. This is the order of preference of the sender, and MUST be maintained throughout the reservation establishment, unless the ADSPEC indicates one or more TSPECs cannot be granted, or the receiver cannot include any TSPEC due to technical or administrative constraints or one or more routers along the RESV path cannot grant a particular TSPEC. At any router that a reservation cannot honor a TSPEC, this TSPEC MUST be removed from the RESV, or else another router along the RESV path might reserve that TSPEC. This rule ensures this cannot happen.

Once one TSPEC has been removed from the RESV, the next in line TSPEC becomes the preferred TSPEC for that reservation. That router MUST generate a ResvErr message, containing an ERROR_SPEC object with a Policy Control Failure with Error code = 2 (Policy Control Failure), and an Error Value sub-code 102 (ERR_PARTIAL_PREEMPT) to the previous routers, clearing the now over allocation of bandwidth for this reservation. The difference between the previously accepted TSPEC bandwidth and the currently accepted TSPEC bandwidth is the amount this error identifies as the amount of bandwidth that is no longer required to be reserved. The ResvErr and the RESV messages are independent, and not normally sent by the same router. This aspect of this document is the extension to RFC 2205 (RSVP).

If a RESV cannot grant the final TSPEC, normal RSVP rules apply with regard to the transmission of a particular ResvErr.

3.4 Multiple FLOWSPEC for Guaranteed service

The FLOWSPEC object, which is used to request guaranteed service contains a TSPEC and RSpec. Here is the FLOWSPEC object from [RFC2215] when requesting Guaranteed service:

```
      31           24 23           16 15            8 7             0
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1   | 0 (a) |    Unused             |            10 (b)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2   |    2  (c)     |0| reserved    |             9 (d)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  3   |   127 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  4   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  5   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  6   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  7   |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  8   |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  9   |     130 (h)   |    0 (i)      |            2 (j)              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  10  |  Rate [R]  (32-bit IEEE floating point number)                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  11  |  Slack Term [S]  (32-bit integer)                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 6. FLOWSPEC for Guaranteed service

```
(a) - Message format version number (0)
(b) - Overall length (9 words not including header)
(c) - Service header, service number 2 (Guaranteed)
(d) - Length of per-service data, 9 words not including
      per-service header
(e) - Parameter ID, parameter 127 (Token Bucket TSpec)
(f) - Parameter 127 flags (none set)
(g) - Parameter 127 length, 5 words not including parameter header
(h) - Parameter ID, parameter 130 (Guaranteed Service RSpec)
(i) - Parameter xxx flags (none set)
(j) - Parameter xxx length, 2 words not including parameter header
```

The difference in structure between the Controlled-Load FLOWSPEC and Guaranteed FLOWSPEC is the RSPEC, defined in [RFC2212].

For completeness, Figure 6 is included in its original form for backwards compatibility reasons, as if there were only 1 FLOWSPEC in the RESV. What is new when there is more than one TSPEC in the FLOWSPEC in a RESV message is the new MULTI_FLOWSPEC object in Figure 7 containing, for example, 3 FLOWSPECs requesting Guaranteed Service.

```
       31           24 23           16 15            8 7             0
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  1   | 0 (a) |    Unused             |            28 (b)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  2   |    2  (c)     |0| reserved    |            27 (d)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  3   |   125 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  4   |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  5   |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  6   |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  7   |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  8   |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  9   |     124 (h)   |    0 (i)      |            2 (j)              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  10  |  Rate [R]  (32-bit IEEE floating point number)                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  11  |  Slack Term [S]  (32-bit integer)                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  12  |   125 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  13  |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  14  |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  15  |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  16  |  Minimum Policed Unit [m] (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  17  |  Maximum Packet Size [M]  (32-bit integer)                    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  18  |     124 (h)   |    0 (i)      |            2 (j)              |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  19  |  Rate [R]  (32-bit IEEE floating point number)                |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  20  |  Slack Term [S]  (32-bit integer)                             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  21  |   125 (e)     |    0 (f)      |             5 (g)             |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  22  |  Token Bucket Rate [r] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  23  |  Token Bucket Size [b] (32-bit IEEE floating point number)    |
      +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  24  |  Peak Data Rate [p] (32-bit IEEE floating point number)       |
```

```
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
  25 |  Minimum Policed Unit [m] (32-bit integer)                     |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

```
 26  |  Maximum Packet Size [M]  (32-bit integer)                    |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 27  |     124 (h)    |    0 (i)      |           2 (j)              |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 28  |  Rate [R]  (32-bit IEEE floating point number)                |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
 29  |  Slack Term [S]  (32-bit integer)                             |
     +-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+-+
```

Figure 7. Multiple FLOWSPECs for Guaranteed service

(a) - Message format version number (0)
(b) - Overall length (9 words not including header)
(c) - Service header, service number 2 (Guaranteed)
(d) - Length of per-service data, 9 words not including
      per-service header
(e) - Parameter ID, parameter 125 (Token Bucket TSpec)
(f) - Parameter 125 flags (none set)
(g) - Parameter 125 length, 5 words not including parameter header
(h) - Parameter ID, parameter 124 (Guaranteed Service RSpec)
(i) - Parameter 124 flags (none set)
(j) - Parameter 124 length, 2 words not including parameter header

There MUST be 1 RSPEC per TSPEC for Guaranteed Service. Therefore, there are 5 words for Receiver TSPEC and 3 words for the RSPEC. Therefore, for Guaranteed Service, the TSPEC/RSPEC combination occurs in increments of 8 words.

4. Rules of Usage

The following rules apply to nodes adhering to this specification:

4.1 Backward Compatibility

If the recipient does not understand this extension, it ignores this MULTI_TSPEC object, and operates normally for a node receiving this RSVP message.

4.2 Applies to Only a Single Session

When there is more than one TSPEC object or more than one FLOWSPEC object, this MUST NOT be considered for more than one flow created. These are OR choices for the same flow of data. In order to attain three reservations between two endpoints, three different reservation requests are required, not one reservation request with 3 TSPECs.

4.3 No Special Error Handling for PATH Message

If a problem occurs with the PATH message - regardless of this extension, normal RSVP procedures apply (i.e., there is no new PathErr code created within this extension document) - resulting in a PathErr message being sent upstream towards the sender, as usual.

4.4 Preference Order to be Maintained

When more than one TSPEC is in a PATH message, the order of TSPECs is decided by the Sender and MUST be maintained within the SENDER_TSPEC. The same order MUST be carried to the FLOWSPECs by the receiver. No additional TSPECS can be introduced by the receiver or any router processing these new objects. The deletion of TSPECs from a PATH message is not permitted. The deletion of the TSPECs when forming the FLOWSPEC is allowed by the receiver in the following cases:

- If one or more preferred TSPECs cannot be granted by a router as discovered during processing of the ADSPEC by the receiver, then they can be omitted when creating the FLOWSPEC(s) from the TSPECs.

- If one or more TSPECs arriving from the sender is not preferred by the receiver, then the receiver MAY omit any while creating the FLOWSPEC. A good reason to omit a TSPEC is if, for example, it does not match a codec supported by the receiver's application(s).

The deletion of the TSPECs in the router during the processing of this MULTI_FLOWSPEC object is allowed in the following cases:

- If the original FLOWSPEC cannot be granted by a router then the router may discard that FLOWSPEC and replace it with the topmost FLOWSPEC from the MULTI_FLOWSPEC project. This will cause the topmost FLOWSPEC in the MULTI_FLOWSPEC object to be removed. The next FLOWSPECs becomes the topmost FLOWSPEC.

- If the router merges multiple RESV into a single RESV message, then the FLOWSPEC and the multiple FLOWSPEC may be affected

The preferred order of the remaining TSPECs or FLOWSPECs MUST be kept intact both at the receiver as well as the router processing these objects.

4.5 Bandwidth Reduction in Downstream Routers

If there are multiple FLOWSPECs in a single RESV message, it is quite possible that a higher bandwidth is reserved at a previous downstream device. Thus, any device that grants a reservation that is not the highest will have to inform the previous downstream routers to reduce the bandwidth reserved for this particular session.

The bandwidth reduction RFC [RFC4495] does not address the need that this document addresses. RFC 4495 defines an ability to preempt part of an existing reservation so as to admit a new incoming reservation with a higher priority, in lieu of tearing down the whole reservation having a lower priority. It does not specify the capability to reduce the bandwidth a RESV set up along the data path before the reservation is realized (from source to destination), when a subsequent router cannot support a more preferred FLOWSPEC contained in that RESV. This document extends the RFC 4495 defined partial teardown error to reduce bandwidth from previous downstream hops while a reservation is being established.

For example, if a 12Mbps TSPEC were granted for a reservation on previous hops, but could not be granted at the current hop, while the 4Mbps TSPEC could be granted (provided there is a MULTI_TSPEC with a 4Mbps TSPEC), this modification to the bandwidth reduction function would work by having the 4Mbps granting node send a reduction error to the downstream routers that installed 12Mbps for this reservation, thus clearing bandwidth that is now unnecessarily installed for a 4Mbps reservation.

4.6 Merging Rules

RFC 2205 defines the rules for merging as combining more than one FLOWSPEC into a single FLOWSPEC. In the case of MULTI_FLOWSPECs, merging of the two (or more) MULTI_FLOWSPEC MUST be done to arrive at a single MULTI_FLOWSPEC. The merged MULTI_FLOWSPEC will contain all the flow specification components of the individual MULTI_FLOWSPECs in descending orders of bandwidth. In other words, the merged FLOWSPEC MUST maintain the relative order of each of the individual FLOWSPECs. For example, if the individual FLOWSPEC order is 1,2,3 and another FLOWSPEC is a,b,c, then this relative ordering cannot be altered in the merged FLOWSPEC.

A byproduct of this is the ordering between the two individual FLOWSPECs cannot be signaled with this extension. If two (or more) FLOWSPECs have the same bandwidth, they are to be merged into one FLOWSPEC using the rules defined in RFC 2205. It is RECOMMENDED that the following rules are used for determining ordering (in TSPEC and FLOWSPEC):

For Controlled Load - in descending order of BW based on the Token Bucket Rate 'r' parameter value

For Guaranteed Service - in descending order of BW based on the RSPEC Rate 'R' parameter value

The resultant FLOWSPEC is added to the MULTI_FLOWSPEC based on its bandwidth in descending orders of bandwidth.

As a result of such merging, the number of FLOWSPECs in a MULTI_FLOWSPEC object should be the sum of the number of FLOWSPECs from individual MULTI_FLOWSPEC that have been merged *minus* the number of duplicates.

4.7 Applicability to Multicast

An RSVP message with a MULTI_TSPEC works just as well in a multicast scenario as it does in a unicast scenario. In a multicast scenario, the bandwidth allotted in each hop is the lowest bandwidth that can be admitted along the various path. For example:

```
+--------+       +----------+      +----------+      +------------+
| sender |======>| Router-1 |=====>| Router-2 |=====>| Receiver-A |
+--------+       +----------+      +----------+      +------------+
                      |                  |
                      |                  |
                      |                  V
                      |            +------------+
                      |            | Receiver-C |
                      |            +------------+
                      |
                      V
               +------------+
               | Receiver-B |
               +------------+
```

Figure 8. MULTI_TPSEC and Multicast

If the sender (in Figure 8) sends 3 TSPECs (i.e., 1 TSPEC Object, and 2 in the MULTI_TSPEC Object) of 12Mbps, 5Mbps and 1.5Mbps. Let us say the path from Receiver-B to Router-1 admitted 5Mbps, Receiver-C to Router-2 admitted 1.5Mbps and Receiver-A to Router-2 admitted 12Mbps.

When the Resv message is send upstream from Router-2, the combining of 1.5Mbps (to Receiver-C) and 12Mbps (to Receiver-A) will be resolved to 1.5Mbps (lowest that can be admitted). Only a Resv with 1.5Mbps will be sent upstream from Router-2. Likewise, at Router-1, the combining of 1.5Mbps (to Router-2) and 5Mbps (to Receiver-B) will be resolved to 1.5Mbps units.

This is to allow the sender to transmit the flow at a rate that can be accepted by all devices along the path. Without this, if Router-2 receives a flow of 12Mbps, it will not know how to create a flow of 1.5Mbps down to Receiver-B. A differentiated reservation for the various paths along a multicast path is only possible with a Media-aware network device (MANE). The discussion of MANE and how it relates to admission control is outside the scope of this draft.

4.8 MULTI_TSPEC Specific Error

Since this mechanism is backward compatible, it is possible that a router without support for this MULTI_TSPEC extension will reject a reservation because the bandwidth indicated in the primary FLOWSPECs is not available. This means that an attempt with a lower bandwidth might have been successful, if one were included in a MULTI_TSPEC Object. Therefore, one should be able to differentiate between an admission control error where there is insufficient bandwidth when all the FLOWSPECs are considered and insufficient bandwidth when only the primary FLOWSPEC is considered.

This requires the definition of an error code within the ERROR_SPEC Object. When a router does not have sufficient bandwidth even after considering all the FLOWSPEC provided, it issues a new "MULTI_TSPEC bandwidth unavailable " error. This will be an Admission Control Failure (error #1), with a subcode of 6. A router that does not support this MULTI_TSPEC extension will return the "requested bandwidth unavailable" error as defined in RFC 2205 as if there was no MULTI_TSPEC in the message.

4.9 Other Considerations

- RFC 4495 articulates why a ResvErr is more appropriate to use for reducing the bandwidth of an existing reservation vs. a ResvTear.

- Refreshes only include the TSPECs that were accepted. One SHOULD be sent immediately upon the Sender receiving the RESV, to ensure all routers in this flow are synchronized with which TSPEC is in place.

- Periodically, it might be appropriate to attempt to increase the bandwidth of an accepted reservation with one of the TSPECs that were not accepted by the network when the reservation was first installed. This SHOULD NOT occur too regularly. This document currently offers no guidance on the frequency of this bump request for a rejected TSPEC from the PATH.

4.10 Known Open Issues

Here are the know open issues within this document:

o Both the idea of MULTI_RSPEC and MULTI_FLOWSPEC need to be fleshed out, and IANA registered.

o Need to ensure the cap on the number of TSPECs and FLOWSPECs is viable, yet controlled.

5. Security considerations

The security considerations for this document do not exceed what is already in RFC 2205 (RESV) or RFC 2210 (IntServ), as nothing in either of those documents prevent a node from requesting a lot of bandwidth in a single TSPEC. This document merely reduces the signaling traffic load on the network by allowing many requests that fall under the same policy controls to be included in a single round-trip message exchange.

Further, this document does not increase the security risk(s) to that defined in RFC 4495, where this document creates additional meaning to the RFC 4495 created error code 102.

A misbehaving Sender can include too many TSPECs in the MULTI_TSPEC object, which can lead to an amplification attack. That said, a bad implementation can create a reservation for each TSPEC received from within the Resv message. The number of TSPECs in the new MULTI_TSPEC object is limited, and the spec clearly states that only a single reservation is to be set up per Resv message.

To ensure the integrity of RSVP, the RSVP Authentication mechanisms defined in [RFC2747] and [RFC3097] SHOULD be used. Those protect RSVP message integrity hop-by-hop and provide node authentication as well as replay protection, thereby protecting against corruption and spoofing of RSVP messages.

6. IANA considerations

This document IANA registers the following new parameter name in the Integ-serv assignments at [IANA]:

Registry Name: Parameter Names
Registry:

Value	Description	Reference
125	Multiple_Token_Bucket_Tspec	[RFCXXXX]
124	Multiple_Guaranteed_Service_RSpec	[RFCXXXX]

Where RFCXXXX is replaced with the RFC number assigned to this Document.

This document IANA registers the following new error subcode in the Error code section, under the Admission Control Failure (error=1), of the rsvp-parameters assignments at [IANA]:

Registry Name: Error Codes and Globally-Defined Error Value
                Sub-Codes
Registry:

"Admission Control
 Failure"

Error Subcode		meaning	Reference
6	=	MULTI_TSPEC bandwidth unavailable	[RFCXXXX]

7. Acknowledgments

The authors wish to thank Fred Baker, Joe Touch, Bruce Davie, Dave Oran, Ashok Narayanan, Lou Berger, Lars Eggert, Arun Kudur and Janet Gunn for their helpful comments and guidance in this effort.

And to Francois Le Faucheur, who provided text in this version.

8. References

8.1. Normative References

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

[RFC2205] R. Braden, Ed., L. Zhang, S. Berson, S. Herzog, S. Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997

[RFC2210] J. Wroclawski, "The Use of RSVP with IETF Integrated Services", RFC 2210, September 1997

[RFC2212] S. Shenker, C. Partridge, R. Guerin, "Specification of Guaranteed Quality of Service", RFC 2212, September 1997

[RFC2215] S. Shenker, J. Wroclawski, "General Characterization Parameters for Integrated Service Network Elements", RFC 2212, September 1997

[RFC4495] J. Polk, S. Dhesikan, "A Resource Reservation Protocol (RSVP) Extension for the Reduction of Bandwidth of a Reservation Flow", RFC 4495, May 2006

[RFC2747] F. Baker, B. Lindell, M. Talwar, " RSVP Cryptographic Authentication", RFC2747, January 2000

[RFC3097] R. Braden, L. Zhang, "RSVP Cryptographic Authentication -- Updated Message Type Value", RFC 3097, April 2001

8.2. Informative References

[IANA] http://www.iana.org/assignments/integ-serv

Authors' Addresses

James Polk
3913 Treemont Circle
Colleyville, Texas, USA
+1.817.271.3552

mailto: jmpolk@cisco.com

Subha Dhesikan
Cisco Systems
170 W. Tasman Drive
San Jose, CA 95134 USA

mailto: sdhesika@cisco.com

Appendix A: Alternatives for Sending Multiple TSPECs

This appendix describes the discussion within the TSVWG of which approach best fits the requirements of sending multiple TSPECs within a single PATH or RESV message. There were 3 different options proposed, of which - 2 were insufficient or caused more harm than other options.

Looking at the format of a PATH message [RFC2205] again:

```
        <PATH Message> ::= <Common Header> [ <INTEGRITY> ]

                                  <SESSION> <RSVP_HOP>

                                  <TIME_VALUES>

                                 [ <POLICY_DATA> ... ]

                                 [ <sender descriptor> ]

        <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>
                                                   ^^^^^^^^^^^^
                                 [ <ADSPEC> ]
```

For the PATH message, the focus of this document is with what to do with respect to the <SENDER_TSPEC> above, highlighted by the '^^^^' characters. No other object within the PATH message will be affected by this IntServ extension.

The ADSPEC is optional in IntServ; therefore it might or might not be in the RSVP PATH message. Presently, the SENDER_TSPEC is limited to one bandwidth associated with the session. This is changed in this extension to IntServ to multiple bandwidths for the same session. There are multiple options on how the additional bandwidths may be added:

Option #1 - creating the ability to add one or more additional
            (and complete) SENDER_TSPECs,

or

Option #2 - create the ability for the one already allowed
             SENDER_TSPEC to carry more than one bandwidth amount
             for the same reservation.

or

Option #3 - create the ability for the existing SENDER_TSPEC to
             remain unchanged, but add an optional <MULTI_TSPEC>
             object to the <sender descriptor> such as this:

```
        <sender descriptor> ::= <SENDER_TEMPLATE> <SENDER_TSPEC>

                                 [ <ADSPEC> ]  [ <MULTI_TSPEC> ]
                                                ^^^^^^^^^^^
```

Here is another way of looking at the option choices:

```
+--------------------+----------------------+---------------------+
|    Option#1        |      Option#2        |      Option#3       |
|                    |                      |                     |
|   +----------+     |  +---------------+   |    +----------+     |
|   |  TSPEC1  |     |  |  MULTI_TSPEC  |   |    |  TSPEC1  |     |
|   +----------+     |  |    Object     |   |    +----------+     |
|                    |  |  +--------+   |   |                     |
|   +----------+     |  |  | TSPEC1 |   |   |  +---------------+  |
|   |  TSPEC2  |     |  |  +--------+   |   |  |  MULTI_TSPEC  |  |
|   +----------+     |  |  +--------+   |   |  |    Object     |  |
|                    |  |  | TSPEC2 |   |   |  |  +--------+   |  |
|   +----------+     |  |  +--------+   |   |  |  | TSPEC2 |   |  |
|   |  TSPEC3  |     |  |  +--------+   |   |  |  +--------+   |  |
|   +----------+     |  |  | TSPEC3 |   |   |  |  +--------+   |  |
|                    |  |  +--------+   |   |  |  | TSPEC3 |   |  |
|   +----------+     |  |  | TSPEC4 |   |   |  |  +--------+   |  |
|   |  TSPEC4  |     |  |  +--------+   |   |  |  +--------+   |  |
|   +----------+     |  +---------------+   |  |  | TSPEC4 |   |  |
|                    |                      |  |  +--------+   |  |
|                    |                      |  +---------------+  |
|                    |                      |                     |
+--------------------+----------------------+---------------------+
```

 Figure 3. Concept of Option Choice

Option #1 and #2 do not allow for backward compatibility. If the currently used SENDER_TSPEC and FLOWSPEC objects are changed, then unless all the routers requiring RSVP processing are upgraded, this functionality cannot be realized. As it is unlikely that all routers along the path will have the necessary enhancements as per this extension at one given time, therefore, it is necessary this enhancement be made in a way that is backward compatible. Therefore, option #1 and option #2 has been discarded in favor of option #3, which had WG consensus in a recent IETF meeting.

Option #3: This option has the advantage of being backwards compatible with existing implementations of [RFC2205] and [RFC2210], as the multiple TSPECs and FLOWSPECs are inserted as optional objects and such objects do not need to be processed, especially if they are not understood.

Option#3 applies to the FLOWSPEC contained in the RESV message as well. In this option, the original SENDER_TSPEC and the FLOWSPEC are left untouched, allowing routers not supporting this extension to be able to process the PATH and the RESV message without issue. Two new additional objects are defined in this document. They are the MULTI_TSPEC and the MULTI_FLOWSPEC for the PATH and the RESV message, respectively. The additional TSPECs (in the new MULTI_TSPEC Object) are included in the PATH and the additional FLOWSPECS (in the new MULTI_FLOWSPEC Object) are included in the RESV message as new (optional) objects. These additional objects will have a class number of 11bbbbbb, allowing older routers to ignore the object(s) and forward each unexamined and unchanged, as defined in section 3.10 of [RFC 2205].

We state in the document body that the top most FLOWSPEC of the new MULTI_FLOWSPEC Object in the RESV message replaces the existing FLOWSPEC when it is determined by the receiver (perhaps along with the ADSPEC) that the original FLOWSPEC cannot be granted. Therefore, the ordering of preference issue is solved with Option#3 as well.

NOTE: it is important to emphasize here that including more than one FLOWSPEC in the RESV message does not cause more than one FLOWSPEC to be granted. This document requires that the receiver arrange these multiple FLOWSPECs in the order of preference according to the order remaining from the MULTI_TSPECs in the PATH message. The benefit of this arrangement is that RSVP does not have to process the rest of the FLOWSPEC if it can admit the first one.

Additional details of these options can be found in the draft-polk-tsvwg-...-01 version of this appendix (which includes the RSVP bit mapping of fields in the TSPECs, if the reader wishes to search for that doc.