TSVWG
Internet-Draft
Intended status: Standards Track
Expires: September 9, 2010

F. Le Faucheur
J. Polk
Cisco
K. Carlberg
G11
March 8, 2010

Resource ReSerVation Protocol (RSVP) Extensions for Admission Priority
draft-ietf-tsvwg-emergency-rsvp-15.txt

Abstract

Some applications require the ability to provide an elevated probability of session establishment to specific sessions in times of network congestion. When supported over the Internet Protocol suite, this may be facilitated through a network layer admission control solution that supports prioritized access to resources (e.g., bandwidth). These resources may be explicitly set aside for prioritized sessions, or may be shared with other sessions. This document specifies extensions to the Resource reSerVation Protocol (RSVP) that can be used to support such an admission priority capability at the network layer.

Based on current security concerns, these extensions are intended for use in a single administrative domain.

This Internet-Draft will expire on September 9, 2010.

Copyright Notice

Table of Contents

1. Introduction

Some applications require the ability to provide an elevated probability of session establishment to specific sessions in times of network congestion.

Solutions to meet this requirement for elevated session establishment probability may involve session layer capabilities prioritizing access to resources controlled by the session control function. As an example, entities involved in session control (such as SIP user agents, when the Session Initiation Protocol (SIP) [RFC3261], is the session control protocol in use) can influence their treatment of session establishment requests (such as SIP requests). This may include the ability to "queue" session establishment requests when those can not be immediately honored (in some cases with the notion of "bumping", or "displacement", of less important session establishment requests from that queue). It may include additional mechanisms such as exemption from certain network management controls, and alternate routing.

Solutions to meet the requirement for elevated session establishment probability may also take advantage of network layer admission control mechanisms supporting admission priority. Networks usually have engineered capacity limits that characterize the maximum load that can be handled (say, on any given link) for a class of traffic while satisfying the quality of service requirements of that traffic class. Admission priority may involve setting aside some network resources (e.g. Bandwidth) out of the engineered capacity limits for the prioritized sessions only. Or alternatively, it may involve allowing the prioritized sessions to seize additional resources beyond the engineered capacity limits applied to normal sessions. This document specifies the necessary extensions to support such admission priority when network layer admission control is performed using the Resource reSerVation Protocol (RSVP) ([RFC2205]).

[RFC3181] specifies the Signaled Preemption Priority Policy Element that can be signaled in RSVP so that network node may take into account this policy element in order to preempt some previously admitted low priority sessions in order to make room for a newer, higher priority session. In contrast, this document specifies new RSVP extensions to increase the probability of session establishment without preemption of existing sessions. This is achieved by engineered capacity techniques in the form of bandwidth allocation models. In particular this document specifies two new RSVP Policy Elements allowing the admission priority to be conveyed inside RSVP signaling messages so that RSVP nodes can enforce selective bandwidth admission control decision based on the session admission priority. Appendix A of this document also provides examples of bandwidth

allocation models which can be used by RSVP-routers to enforce such admission priority on every link. A given reservation may be signaled with the admission priority extensions specified in the present document, with the preemption priority specified in [RFC3181] or with both.

1.1. Terminology

This document assumes the terminology defined in [RFC2753]. For convenience, the definition of a few key terms is repeated here:

- Policy Decision Point (PDP): The point where policy decisions are made.
- Local Policy Decision Point (LPDP): PDP local to the network element.
- Policy Enforcement Point (PEP): The point where the policy decisions are actually enforced.
- Policy Ignorant Node (PIN): A network element that does not explicitly support policy control using the mechanisms defined in [RFC2753].

2. Applicability Statement

A subset of RSVP messages are signaled with the Router Alert Option (RAO)([RFC2113],[RFC2711]). The security aspects and common practices around the use of the current IP Router Alert option and consequences on the use of IP Router Alert by applications such as RSVP are discussed in [I-D.rahman-rtg-router-alert-considerations]. Based on those, the extensions defined in this document are intended for use within a single administrative domain. Thus, in particular, the extensions defined in this document are not intended for use end to end on the Internet.

3. Requirements Language

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

4. Overview of RSVP extensions and Operations

Let us consider the case where a session requires elevated

probability of establishment, and more specifically that the preference to be granted to this session is in terms of network layer "admission priority" (as opposed to preference granted through preemption of existing sessions). By "admission priority" we mean allowing the priority session to seize network layer resources from the engineered capacity that has been set-aside for priority sessions (and not made available to normal sessions), or alternatively allowing the priority session to seize additional resources beyond the engineered capacity limits applied to normal sessions.

Session establishment can be made conditional on resource-based and policy-based network layer admission control achieved via RSVP signaling. In the case where the session control protocol is SIP, the use of RSVP-based admission control in conjunction with SIP is specified in [RFC3312].

Devices involved in the session establishment are expected to be aware of the application-level priority requirements of prioritized sessions. For example, considering the case where the session control protocol is SIP, the SIP user agents may be made aware of the resource priority requirements of a given session using the "Resource-Priority" header mechanism specified in [RFC4412]. The end-devices involved in the upper-layer session establishment simply need to copy the application-level resource priority requirements (e.g. As communicated in SIP "Resource-Priority" header) inside the new RSVP Application-Level Resource Priority Policy Element defined in this document.

Conveying the application-level resource priority requirements inside the RSVP message allows this application level requirement to be mapped/remapped into a different RSVP "admission priority" at a policy boundary based on the policy applicable in that policy area. In a typical model (see [RFC2753]) where PDPs control PEPs at the periphery of the policy area (e.g. On the first hop router), PDPs would interpret the RSVP Application-Level Resource Priority Policy Element and map the requirement of the prioritized session into an RSVP "admission priority" level. Then, PDPs would convey this information inside the new Admission Priority Policy Element defined in this document. This way, the RSVP admission priority can be communicated to downstream PEPs (i.e. RSVP Routers) of the same policy domain, which have LPDPs but no controlling PDP. In turn, this means the necessary RSVP Admission priority can be enforced at every RSVP hop, including all the (possibly many) hops which do not have any understanding of Application-Level Resource Priority semantics. It is not expected that the RSVP Application-Level Resource Priority Header Policy Element would be taken into account at RSVP-hops within a given policy area. It is expected to be used at policy area boundaries only in order to set/reset the RSVP

Admission Priority Policy Element.

Remapping by PDPs of the Admission Priority Policy Element from the Application-Level Resource Priority Policy Element may also be used at boundaries with other signaling protocols, such as the NSIS Signaling Layer Protocol (NSLP) for QoS Signaling ([I-D.ietf-nsis-qos-nslp]).

As can be observed, the framework described above for mapping/ remapping application level resource priority requirements into an RSVP admission priority can also be used together with [RFC3181] for mapping/remapping application level resource priority requirements into an RSVP preemption priority (when preemption is indeed deemed necessary by the prioritized session handling policy). In that case, when processing the RSVP Application-Level Resource Priority Policy Element, the PDPs at policy boundaries (or between various QoS signaling protocols) can map it into an RSVP "preemption priority" information. This Preemption priority information comprises a setup preemption level and a defending preemption priority level that can then be encoded inside the Preemption Priority Policy Element of [RFC3181].

Appendix B provides examples of various hypothetical policies for prioritized session handling, some of them involving admission priority, some of them involving both admission priority and preemption priority. Appendix B also identifies how the Application-Level Resource Priority needs to be mapped into RSVP policy elements by the PDPs to realize these policies.

4.1. Operations of Admission Priority

The RSVP Admission Priority policy element defined in this document allows admission bandwidth to be allocated preferentially to prioritized sessions. Multiple models of bandwidth allocation MAY be used to that end.

A number of bandwidth allocation models have been defined in the IETF for allocation of bandwidth across different classes of traffic trunks in the context of Diffserv-aware MPLS Traffic Engineering. Those include the Maximum Allocation Model (MAM) defined in [RFC4125], the Russian Dolls Model (RDM) specified in [RFC4127] and the Maximum Allocation model with Reservation (MAR) defined in [RFC4126]. These same models MAY however be applied for allocation of bandwidth across different levels of admission priority as defined in this document. Appendix A provides an illustration of how these bandwidth allocation models can be applied for such purposes and also introduces an additional bandwidth allocation model that we term the Priority Bypass Model (PrBM). It is important to note that the

models described and illustrated in Appendix A are only informative and do not represent a recommended course of action.

We can see in these examples how the RSVP Admission Priority can be used by RSVP routers to influence their admission control decision (for example by determining which bandwidth pool is to be used by RSVP for performing its bandwidth allocation) and therefore to increase the probability of reservation establishment. In turn, this increases the probability of application level session establishment for the corresponding session.

5. New Policy Elements

The Framework document for policy-based admission control [RFC2753] describes the various components that participate in policy decision making (i.e., PDP, PEP and LPDP).

As described in Section 4 of the present document, the Application-Level Resource Priority Policy Element and the Admission Priority Policy Element serve different roles in this framework:

- the Application-Level Resource Priority Policy Element conveys application level information and is processed by PDPs
- the emphasis of Admission Priority Policy Element is to be simple, stateless, and light-weight such that it can be processed internally within a node's LPDP. It can then be enforced internally within a node's PEP. It is set by PDPs based on processing of the Application-Level Resource Priority Policy Element.

[RFC2750] defines extensions for supporting generic policy based admission control in RSVP. These extensions include the standard format of POLICY_DATA objects and a description of RSVP handling of policy events.

The POLICY_DATA object contains one or more of Policy Elements, each representing a different (and perhaps orthogonal) policy. As an example, [RFC3181] specifies the Preemption Priority Policy Element. This document defines two new Policy Elements called:

- the Admission Priority Policy Element
- the Application-Level Resource Priority Policy Element

5.1. Admission Priority Policy Element

The format of the Admission Priority policy element is as shown in Figure 1:

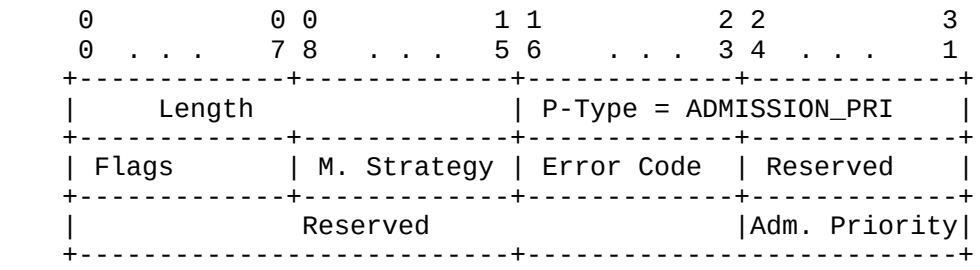


Figure 1: Admission Priority Policy Element

where:

- Length: 16 bits
  - Always 12. The overall length of the policy element, in bytes.
- P-Type: 16 bits
  - ADMISSION_PRI = To be allocated by IANA (see "IANA Considerations" section)
- Flags: Reserved
  - SHALL be set to zero on transmit and SHALL be ignored on reception
- Merge Strategy: 8 bits (applicable to multicast flows)
  - values are defined by corresponding registry maintained by IANA (see "IANA Considerations" section)
- Error code: 8 bits (applicable to multicast flows)
  - values are defined by corresponding registry maintained by IANA (see "IANA Considerations" section)
- Reserved: 8 bits
  - SHALL be set to zero on transmit and SHALL be ignored on reception

- Reserved: 24 bits
  - SHALL be set to zero on transmit and SHALL be ignored on reception
- Adm. Priority (Admission Priority): 8 bits (unsigned)
  - The admission control priority of the flow, in terms of access to network bandwidth in order to provide higher probability of session completion to selected flows. Higher values represent higher priority. Bandwidth allocation models such as those described in Appendix A are to be used by the RSVP router to achieve increased probability of session establishment. The admission priority value effectively indicates which bandwidth constraint(s) of the bandwidth constraint model in use is(are) applicable to admission of this RSVP reservation.

Note that the Admission Priority Policy Element does NOT indicate that this RSVP reservation is to preempt any other RSVP reservation. If a priority session justifies both admission priority and preemption priority, the corresponding RSVP reservation needs to carry both an Admission Priority Policy Element and a Preemption Priority Policy Element. The Admission Priority and Preemption Priority are handled by LPDPs and PEPs as separate mechanisms. They can be used one without the other, or they can be used both in combination.

5.1.1. Admission Priority Merging Rules

This section discusses alternatives for dealing with RSVP admission priority in case of merging of reservations. As merging applies to multicast, this section also applies to multicast sessions.

The rules for merging Admission Priority Policy Elements are defined by the value encoded inside the Merge Strategy field in accordance with the corresponding IANA registry. This registry applies both to the Merge Strategy field of the Admission Priority Policy Element defined in the present document and to the Merge Strategy field of the Preemption Priority Policy Elements defined in [RFC3181]. The registry initially contains the values already defined in [RFC3181] (see "IANA Considerations" section).

The only difference from [RFC3181] is that this document does not recommend a given merge strategy over the others for Admission Priority, while [RFC3181] recommends the first of these merge strategies for Preemption Priority. Note that with the Admission Priority (as is the case with the Preemption Priority), "Take highest priority" translates into "take the highest numerical value".

5.2. Application-Level Resource Priority Policy Element

The format of the Application-Level Resource Priority policy element is as shown in Figure 2:

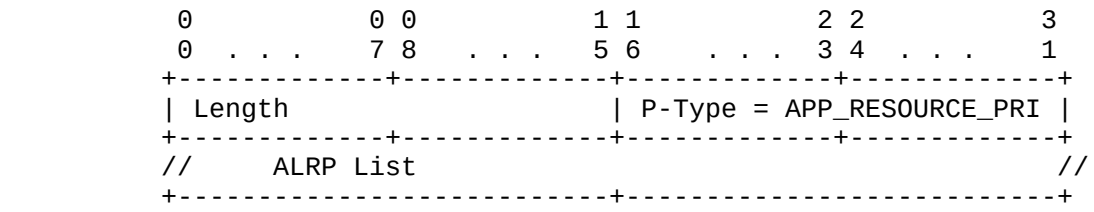


Figure 2: Application-Level Resource Priority Policy Element

where:

- Length:
  - The length of the policy element (including the Length and P-Type) is in number of octets (MUST be a multiple of 4) and indicates the end of the ALRP list.
- P-Type: 16 bits
  - APP_RESOURCE_PRI = To be allocated by IANA (see "IANA Considerations" section)
- ALRP List:
  - List of ALRP where each ALRP is encoded as shown in Figure 3.

ALRP:

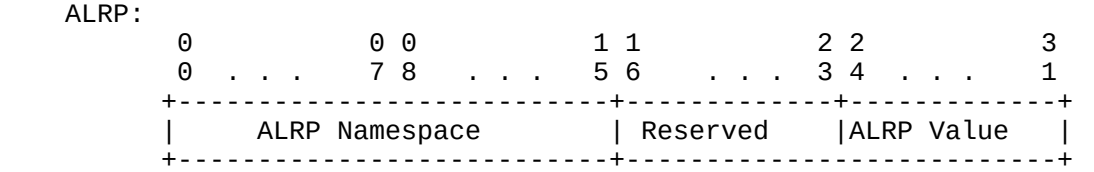


Figure 3: Application-Level Resource Priority

where:

- ALRP Namespace (Application-Level Resource Priority Namespace): 16 bits (unsigned)
  - Contains a numerical value identifying the namespace of the application-level resource priority. This value is encoded as

per the "Resource Priority Namespaces" IANA registry. (See IANA Considerations section for the request to IANA to extend the registry to include this numerical value).

- o Reserved: 8 bits
  - * SHALL be set to zero on transmit and SHALL be ignored on reception.
- o ALRP Value: (Application-Level Resource Priority Value): 8 bits (unsigned)
  - * Contains the priority value within the namespace of the application-level resource priority. This value is encoded as per the "Resource Priority Priority-Value" IANA registry. (See IANA Considerations section for the request to IANA to extend the registry to include this numerical value).

5.2.1. Application-Level Resource Priority Modifying and Merging Rules

When POLICY_DATA objects are protected by integrity, LPDPs should not attempt to modify them. They MUST be forwarded as-is to ensure their security envelope is not invalidated.

In case of multicast, when POLICY_DATA objects are not protected by integrity, LPDPs MAY merge incoming Application-Level Resource Priority elements to reduce their size and number. When they do merge those, LPDPs MUST do so according to the following rule:

- o The ALRP List in the outgoing APP_RESOURCE_PRI element MUST contain all the ALRPs appearing in the ALRP List of an incoming APP_RESOURCE_PRI element. A given ALRP MUST NOT appear more than once. In other words, the outgoing ALRP List is the union of the incoming ALRP Lists that are merged.

As merging applies to Multicast, this rule also applies to Multicast sessions.

5.3. Default Handling

As specified in section 4.2 of [RFC2750], Policy Ignorant Nodes (PINs) implement a default handling of POLICY_DATA objects ensuring that those objects can traverse PIN nodes in transit from one PEP to another. This applies to the situations where POLICY_DATA objects contain the Admission Priority Policy Element and the ALRP Policy Element specified in this document, so that those can traverse PIN nodes.

Section 4.2 of [RFC2750] also defines a similar default behavior for policy-capable nodes that do not recognized a particular Policy Element. This applies to the Admission Priority Policy Element and the ALRP Policy Element specified in this document, so that those can traverse policy-capable nodes that do not support these extensions defined in the present document.

6. Security Considerations

As this document defines extensions to RSVP, the security considerations of RSVP apply. Those are discussed in [RFC2205], [RFC4230] and [I-D.ietf-tsvwg-rsvp-security-groupkeying]. Approaches for addressing those concerns are discussed further below.

A subset of RSVP messages are signaled with the Router Alert Option (RAO)(,[RFC2711]). The security aspects and common practices around the use of the current IP Router Alert option and consequences on the use of IP Router Alert by applications such as RSVP are discussed in [I-D.rahman-rtg-router-alert-considerations]. As discussed in Section 2, the extensions defined in this document are intended for use within a single administrative domain.

[I-D.rahman-rtg-router-alert-considerations] discusses router alert protection approaches for Service Providers. These approaches can be used to protect a given network against the potential risks associated with the leaking of router alert packets resulting from the use of the present extensions in another domain. Also, where RSVP is not used, by simply not enabling RSVP on the routers of a given network, that network can generally isolate itself from any RSVP signaling that may leak from another network that uses the present extensions (since the routers will then typically ignore RSVP messages). Where RSVP is to be used internally within a given network, the network operator can activate, on the edge of his network, mechanisms that either tunnel or drop incoming RSVP messages in order to protect the given network from RSVP signaling that may leak from another network that uses the present extensions.

The ADMISSION_PRI and APP_RESOURCE_PRI Policy Elements defined in this document are signaled by RSVP through encapsulation in a Policy Data object as defined in [RFC2750]. Therefore, like any other Policy Elements, their integrity can be protected as discussed in section 6 of [RFC2750] by two optional security mechanisms. The first mechanism relies on RSVP Authentication as specified in [RFC2747] and [RFC3097] to provide a chain of trust when all RSVP nodes are policy capable. With this mechanism, the INTEGRITY object is carried inside RSVP messages. The second mechanism relies on the INTEGRITY object within the POLICY_DATA object to guarantee integrity

between RSVP Policy Enforcement Points (PEPs) that are not RSVP neighbors.

6.1. Use of RSVP Authentication between RSVP neighbors

RSVP authentication can be used between RSVP neighbors that are policy capable. RSVP Authentication (defined in [RFC2747] and [RFC3097]) SHOULD be supported by an implementation of the present document.

With RSVP authentication, the RSVP neighbors use shared keys to compute the cryptographic signature of the RSVP message. [I-D.ietf-tsvwg-rsvp-security-groupkeying] discusses key types, key provisioning methods as well as their respective applicability.

6.2. Use of INTEGRITY object within the POLICY_DATA object

The INTEGRITY object within the POLICY_DATA object can be used to guarantee integrity between non-neighboring RSVP PEPs. This is useful only when some RSVP nodes are Policy Ignorant Nodes (PINs). The INTEGRITY object within the POLICY_DATA object MAY be supported by an implementation of the present document.

Details for computation of the content of the INTEGRITY object can be found in Appendix B of [RFC2750]. This states that the Policy Decision Point (PDP), at its discretion, and based on destination PEP/PDP or other criteria, selects an Authentication Key and the hash algorithm to be used. Keys to be used between PDPs can be distributed manually or via standard key management protocol for secure key distribution.

Note that where non-RSVP hops may exist in between RSVP hops, as well as where RSVP capable Policy Ignorant Nodes (PINs) may exist in between PEPs, it may be difficult for the PDP to determine what is the destination PDP for a POLICY_DATA object contained in some RSVP messages (such as a Path message). This is because in those cases the next PEP is not known at the time of forwarding the message. In this situation, key shared across multiple PDPs may be used. This is conceptually similar to the use of key shared across multiple RSVP neighbors discussed in [I-D.ietf-tsvwg-rsvp-security-groupkeying]. We observe also that this issue may not exist in some deployment scenarios where a single (or low number of) PDP is used to control all the PEPs of a region (such as an administrative domain). In such scenarios, it may be easy for a PDP to determine what is the next hop PDP, even when the next hop PEP is not known, simply by determining what is the next region that will be traversed (say based on the destination address).

7. IANA Considerations

As specified in [RFC2750], Standard RSVP Policy Elements (P-type values) are to be assigned by IANA as per "IETF Consensus" policy following the policies outlined in [RFC2434] (this policy is now called "IETF Review" as per [RFC5226]) .

IANA needs to allocate two P-Types from the Standard RSVP Policy Element range:

- one P-Type to the Admission Priority Policy Element
- one P-Type to the Application-Level Resource Priority Policy Element.

In Section 5.1, the present document defines a Merge Strategy field inside the Admission Priority policy element. This registry is to be specified as also applicable to the Merge Strategy field of the Preemption Priority Policy Elements defined in [RFC3181]. Since it is conceivable that, in the future, values are added to the registry that only apply to the Admission Priority Policy Element or to the Preemption Priority Policy Element (but not to both), IANA needs to list the applicable documents for each value. IANA needs to allocate the following values::

- 0: Reserved
- 1: Take priority of highest QoS [RFC3181] [RFC-XXX]
- 2: Take highest priority [RFC3181] [RFC-XXX]
- 3: Force Error on heterogeneous merge [RFC3181] [RFC-XXX]

Following the policies outlined in [RFC5226], numbers in the range 4-127 are allocated according to the "IETF Review" policy, numbers in the range 128-240 as "First Come First Served" and numbers between 241-255 are reserved for "Private Use".

In Section 5.1, the present document defines an Error Code field inside the Admission Priority policy element. IANA needs to create a registry for this field and allocate the following values:

- 0: NO_ERROR Value used for regular ADMISSION_PRI elements
- 2: HETEROGENEOUS This element encountered heterogeneous merge

Following the policies outlined in [RFC5226], numbers in the range 3-127 are allocated according to the "IETF Review" policy, numbers in

the range 128-240 as "First Come First Served" and numbers between 241-255 are reserved for "Private Use". Value 1 is Reserved (for consistency with [RFC3181] Error Code values).

The present document defines an ALRP Namespace field in Section 5.2 that contains a numerical value identifying the namespace of the application-level resource priority. The IANA already maintains the Resource-Priority Namespaces registry (under the SIP Parameters) listing all such namespace. However, that registry does not currently allocate a numerical value to each namespace. Hence, this document requests the IANA to extend the Resource-Priority Namespaces registry in the following ways:

- a new column should be added to the registry
- the title of the new column should be "Namespace Numerical Value *"
- in the Legend, add a line saying "Namespace Numerical Value = the unique numerical value identifying the namespace"
- add a line at the bottom of the registry stating the following "* : [RFCXXX] " where XXX is the RFC number of the present document
- allocate an actual numerical value to each namespace in the registry and state that value in the new "Namespace numerical Value *" column.

A numerical value should be allocated immediately by IANA to all existing namespaces. Then, in the future, IANA should automatically allocate a numerical value to any new namespace added to the registry.

The present document defines an ALRP Priority field in Section 5.2 that contains a numerical value identifying the actual application-level resource priority within the application-level resource priority namespace. The IANA already maintains the Resource-Priority Priority-values registry (under the SIP Parameters) listing all such priorities. However, that registry does not currently allocate a numerical value to each priority-value. Hence, this document requests the IANA to extend the Resource-Priority Priority-Values registry in the following ways:

- for each namespace, the registry should be structured with two columns
- the title of the first column should read "Priority Values (least to greatest)"

- the first column should list all the values currently defined in the registry (e.g. For the drsn namespace: "routine", "priority", "immediate", "flash", "flash-override", "flash-override-override" for the drsn namespace)

- the title of the second column should read "Priority Numerical Value *"

- At the bottom of the registry, add a "Legend" with a line saying "Priority Numerical Value = the unique numerical value identifying the priority within a namespace"

- add a line at the bottom of the registry stating the following "* : [RFCXXX] " where XXX is the RFC number of the present document

- allocate an actual numerical value to each and state that value in the new "Priority Numerical Value *" column.

A numerical value should be allocated immediately by IANA to all existing priorities. Then, in the future, IANA should automatically allocate a numerical value to any new namespace added to the registry. The numerical value must be unique within each namespace. For the initial allocation, within each namespace, values should be allocated in decreasing order ending with 0 (so that the greatest priority is always allocated value 0). For example, in the drsn namespace, "routine" would be allocated numerical value 5 and "flash-override-override" would be allocated numerical value 0.

8. Acknowledgments

We would like to thank An Nguyen for his encouragement to address this topic and ongoing comments. Also, this document borrows heavily from some of the work of S. Herzog on Preemption Priority Policy Element [RFC3181]. Dave Oran and Janet Gunn provided useful input into this document. Ron Bonica, Magnus Westerlund, Cullen Jennings, Ross Callon and Tim Polk provided specific guidance for the applicability statement of the mechanisms defined in this document.

9. References

9.1. Normative References

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

[RFC2205] Braden, B., Zhang, L., Berson, S., Herzog, S., and S.

Jamin, "Resource ReSerVation Protocol (RSVP) -- Version 1 Functional Specification", RFC 2205, September 1997.

[RFC2434] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 2434, October 1998.

[RFC2747] Baker, F., Lindell, B., and M. Talwar, "RSVP Cryptographic Authentication", RFC 2747, January 2000.

[RFC2750] Herzog, S., "RSVP Extensions for Policy Control", RFC 2750, January 2000.

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

[RFC3181] Herzog, S., "Signaled Preemption Priority Policy Element", RFC 3181, October 2001.

[RFC3261] Rosenberg, J., Schulzrinne, H., Camarillo, G., Johnston, A., Peterson, J., Sparks, R., Handley, M., and E. Schooler, "SIP: Session Initiation Protocol", RFC 3261, June 2002.

[RFC3312] Camarillo, G., Marshall, W., and J. Rosenberg, "Integration of Resource Management and Session Initiation Protocol (SIP)", RFC 3312, October 2002.

[RFC4412] Schulzrinne, H. and J. Polk, "Communications Resource Priority for the Session Initiation Protocol (SIP)", RFC 4412, February 2006.

[RFC5226] Narten, T. and H. Alvestrand, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 5226, May 2008.

9.2. Informative References

[I-D.ietf-nsis-qos-nslp]
Manner, J., Karagiannis, G., and A. McDonald, "NSLP for Quality-of-Service Signaling", draft-ietf-nsis-qos-nslp-18 (work in progress), January 2010.

[I-D.ietf-tsvwg-rsvp-security-groupkeying]
Behringer, M. and F. Faucheur, "Applicability of Keying Methods for RSVP Security", draft-ietf-tsvwg-rsvp-security-groupkeying-05 (work in

progress), June 2009.

[I-D.rahman-rtg-router-alert-considerations]
Faucheur, F., "IP Router Alert Considerations and Usage", draft-rahman-rtg-router-alert-considerations-03 (work in progress), October 2009.

[RFC2113] Katz, D., "IP Router Alert Option", RFC 2113, February 1997.

[RFC2711] Partridge, C. and A. Jackson, "IPv6 Router Alert Option", RFC 2711, October 1999.

[RFC2753] Yavatkar, R., Pendarakis, D., and R. Guerin, "A Framework for Policy-based Admission Control", RFC 2753, January 2000.

[RFC4125] Le Faucheur, F. and W. Lai, "Maximum Allocation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4125, June 2005.

[RFC4126] Ash, J., "Max Allocation with Reservation Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering & Performance Comparisons", RFC 4126, June 2005.

[RFC4127] Le Faucheur, F., "Russian Dolls Bandwidth Constraints Model for Diffserv-aware MPLS Traffic Engineering", RFC 4127, June 2005.

[RFC4230] Tschofenig, H. and R. Graveman, "RSVP Security Properties", RFC 4230, December 2005.

Appendix A. Examples of Bandwidth Allocation Model for Admission Priority

Sections A.1 and A.2 respectively illustrate how the Maximum Allocation Model (MAM) ([RFC4125]) and the Russian Dolls Model (RDM) ([RFC4127]) can be used for support of admission priority. The Maximum Allocation model with Reservation (MAR) ([RFC4126]) could also be used in a similar manner for support of admission priority. Section A.3 illustrates how a simple "Priority Bypass Model" can also be used for support of admission priority.

For simplicity, operations with only a single "priority" level (beyond non-priority) are illustrated here; However, the reader will appreciate that operations with multiple priority levels can easily

be supported with these models.

In all the figures below:

x represents a non-priority session

o represents a priority session

A.1. Admission Priority with Maximum Allocation Model (MAM)

This section illustrates operations of admission priority when a Maximum Allocation Model (MAM) is used for bandwidth allocation across non-priority traffic and priority traffic. A property of the Maximum Allocation Model is that priority traffic can not use more than the bandwidth made available to priority traffic (even if the non-priority traffic is not using all of the bandwidth available for it).

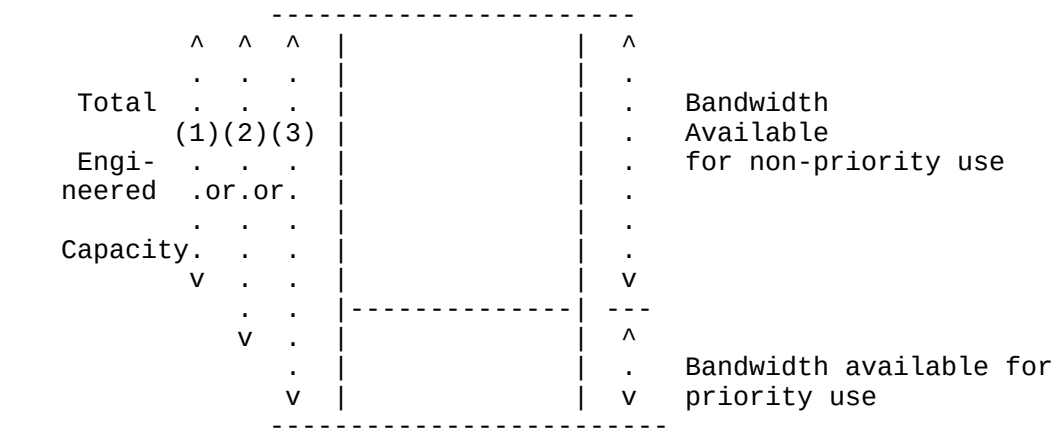


Figure 4: MAM Bandwidth Allocation

Figure 4 shows a link within a routed network conforming to this document. On this link are two amounts of bandwidth available to two types of traffic: non-priority and priority.

If the non-priority traffic load reaches the maximum bandwidth available for non-priority, no additional non-priority sessions can be accepted even if the bandwidth reserved for priority traffic is not currently fully utilized.

With the Maximum Allocation Model, in the case where the priority load reaches the maximum bandwidth reserved for priority sessions, no

additional priority sessions can be accepted.

As illustrated in Figure 4, an operator may map the MAM to the Engineered Capacity limits according to different policies. At one extreme, where the proportion of priority traffic is reliably known to be fairly small at all times and where there may be some safety margin factored in the engineered capacity limits, the operator may decide to configure the bandwidth available for non-priority use to the full engineered capacity limits; effectively allowing the priority traffic to ride within the safety margin of this engineered capacity. This policy can be seen as an economically attractive approach as all of the engineered capacity is made available to non-priority sessions. This policy is illustrated as (1) in Figure 4. As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth available to non-priority traffic to X, and the bandwidth available to priority traffic to 5% of X. At the other extreme, where the proportion of priority traffic may be significant at times and the engineered capacity limits are very tight, the operator may decide to configure the bandwidth available to non-priority traffic and the bandwidth available to priority traffic such that their sum is equal to the engineered capacity limits. This guarantees that the total load across non-priority and priority traffic is always below the engineered capacity and, in turn, guarantees there will never be any QoS degradation. However, this policy is less attractive economically as it prevents non-priority sessions from using the full engineered capacity, even when there is no or little priority load, which is the majority of time. This policy is illustrated as (3) in Figure 4. As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth available to non-priority traffic to 95% of X, and the bandwidth available to priority traffic to 5% of X. Of course, an operator may also strike a balance anywhere in between these two approaches. This policy is illustrated as (2) in Figure 4.

Figure 5 shows some of the non-priority capacity of this link being used.

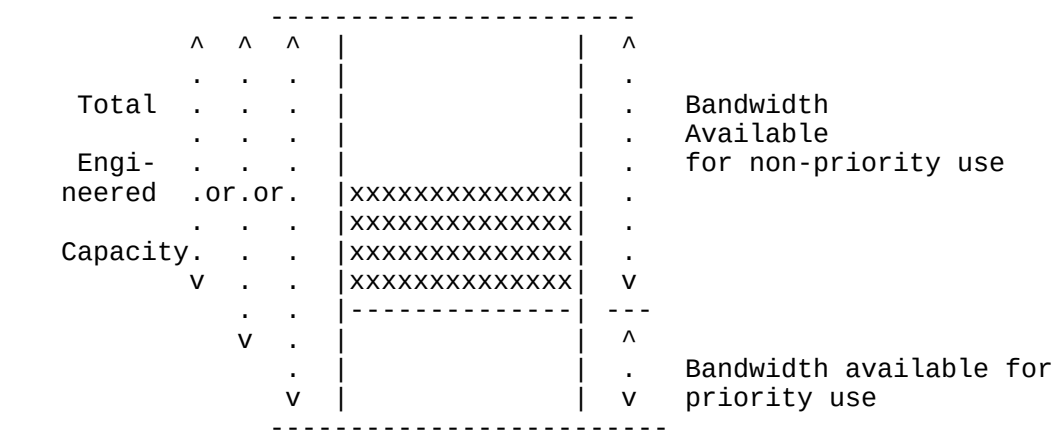


Figure 5: Partial load of non-priority calls

Figure 6 shows the same amount of non-priority load being used at this link, and a small amount of priority bandwidth being used.

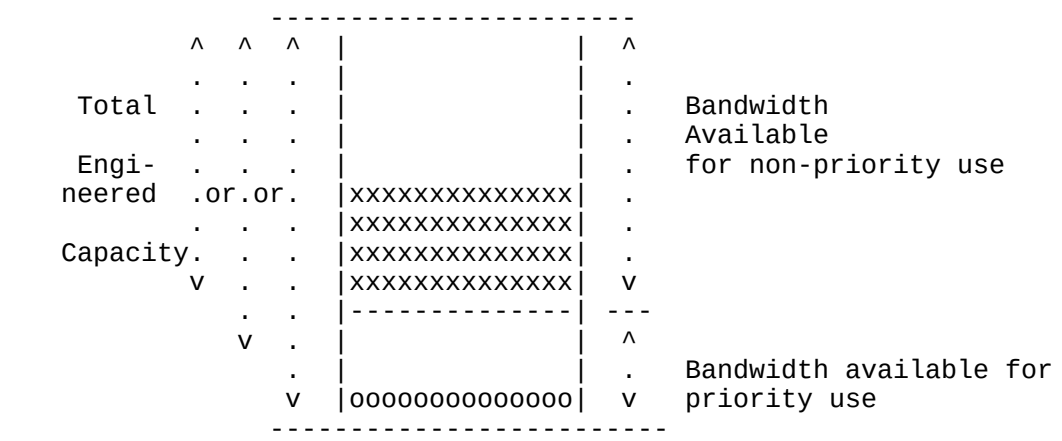


Figure 6: Partial load of non-priority calls & partial load of priority calls Calls

Figure 7 shows the case where non-priority load equates or exceeds the maximum bandwidth available to non-priority traffic. Note that additional non-priority sessions would be rejected even if the bandwidth reserved for priority sessions is not fully utilized.

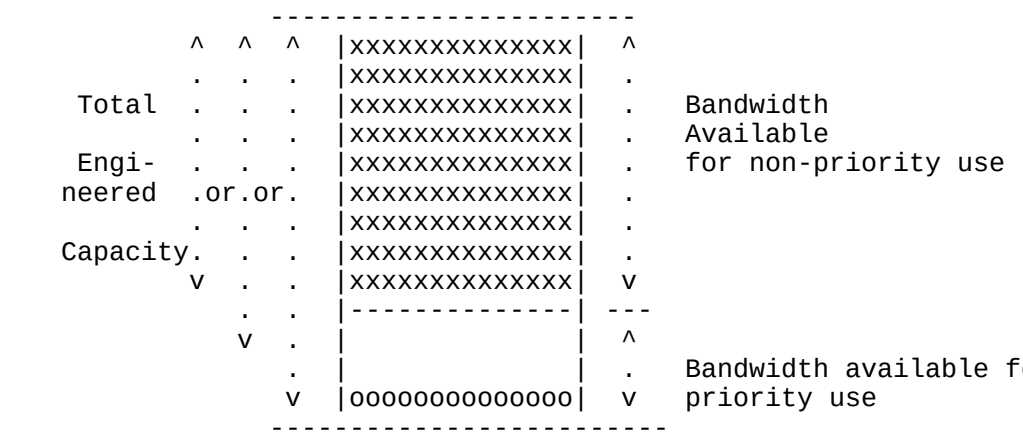


Figure 7: Full non-priority load & partial load of priority calls

Figure 8 shows the case where the priority traffic equates or exceeds the bandwidth reserved for such priority traffic.

In that case additional priority sessions could not be accepted. Note that this does not mean that such sessions are dropped altogether: they may be handled by mechanisms, which are beyond the scope of this particular document (such as establishment through preemption of existing non-priority sessions, or such as queuing of new priority session requests until capacity becomes available again for priority traffic).

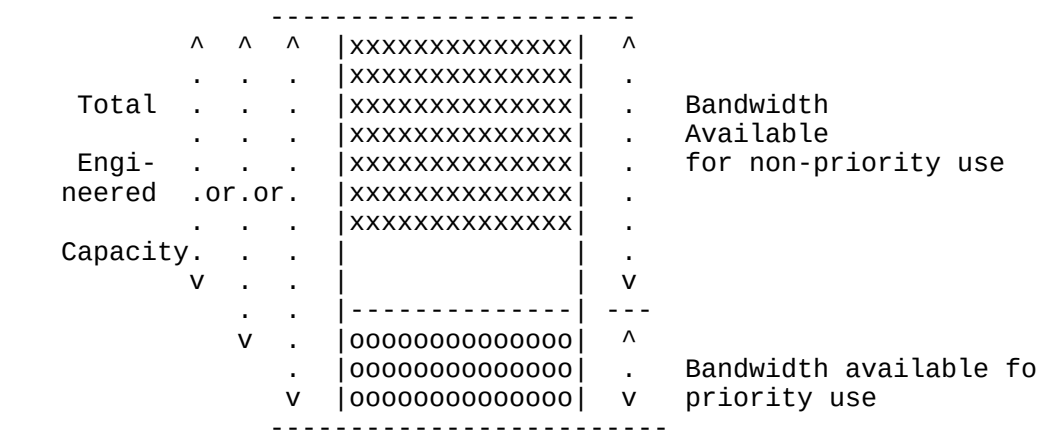


Figure 8: Partial non-priority load & Full priority load

A.2. Admission Priority with Russian Dolls Model (RDM)

This section illustrates operations of admission priority when a Russian Dolls Model (RDM) is used for bandwidth allocation across non-priority traffic and priority traffic. A property of the Russian Dolls Model is that priority traffic can use the bandwidth which is not currently used by non-priority traffic.

As with the MAM, an operator may map the RDM onto the Engineered Capacity limits according to different policies. The operator may decide to configure the bandwidth available for non-priority use to the full engineered capacity limits; As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth available to non-priority traffic to X, and the bandwidth available to non-priority and priority traffic to 105% of X.

Alternatively, the operator may decide to configure the bandwidth available to non-priority and priority traffic to the engineered capacity limits; As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth available to non-priority traffic to 95% of X, and the bandwidth available to non-priority and priority traffic to X.

Finally, the operator may decide to strike a balance in between. The considerations presented for these policies in the previous section in the MAM context are equally applicable to RDM.

Figure 9 shows the case where only some of the bandwidth available to non-priority traffic is being used and a small amount of priority traffic is in place. In that situation both new non-priority

sessions and new priority sessions would be accepted.

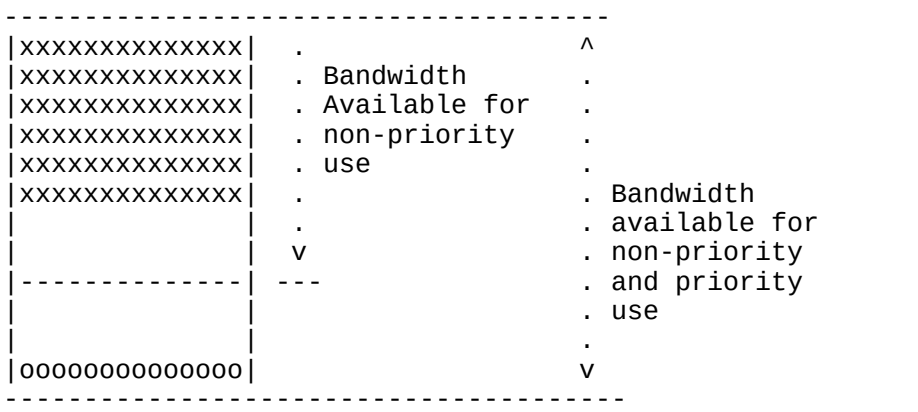


Figure 9: Partial non-priority load & Partial Aggregate load

Figure 10 shows the case where all of the bandwidth available to non-priority traffic is being used and a small amount of priority traffic is in place. In that situation new priority sessions would be accepted but new non-priority sessions would be rejected.

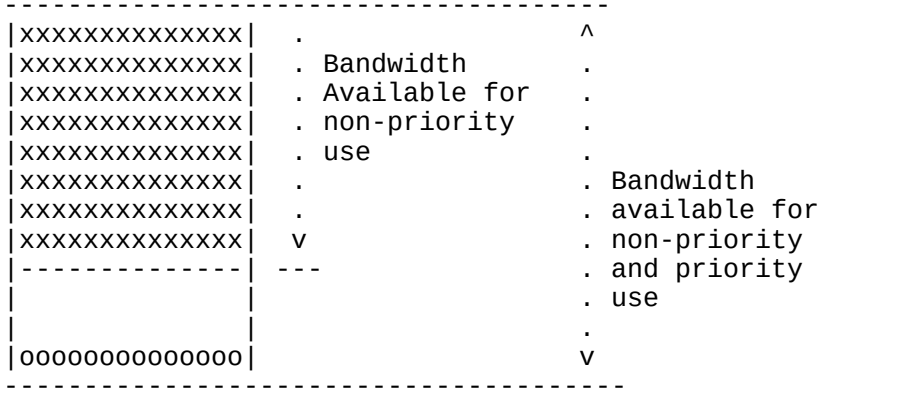


Figure 10: Full non-priority load & Partial Aggregate load

Figure 11 shows the case where only some of the bandwidth available to non-priority traffic is being used and a heavy load of priority traffic is in place. In that situation both new non-priority sessions and new priority sessions would be accepted. Note that, as illustrated in Figure 10, priority sessions use some of the bandwidth currently not used by non-priority traffic.

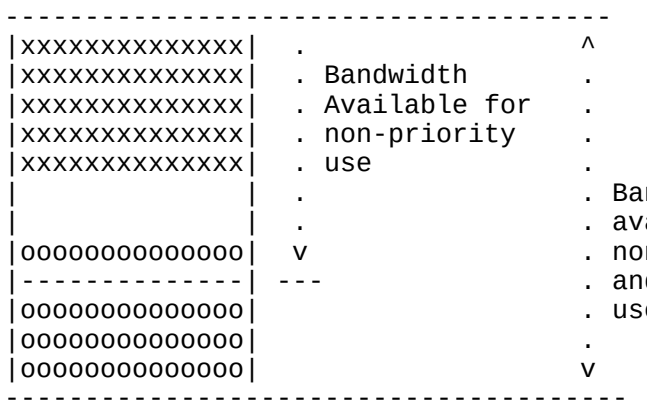


Figure 11: Partial non-priority load & Heavy Aggregate load

Figure 12 shows the case where all of the bandwidth available to non-priority traffic is being used and all of the remaining available bandwidth is used by priority traffic. In that situation new non-priority sessions would be rejected. In that situation new priority sessions could not be accepted right away. Those priority sessions may be handled by mechanisms, which are beyond the scope of this particular document (such as established through preemption of existing non-priority sessions, or such as queuing of new priority session requests until capacity becomes available again for priority traffic).

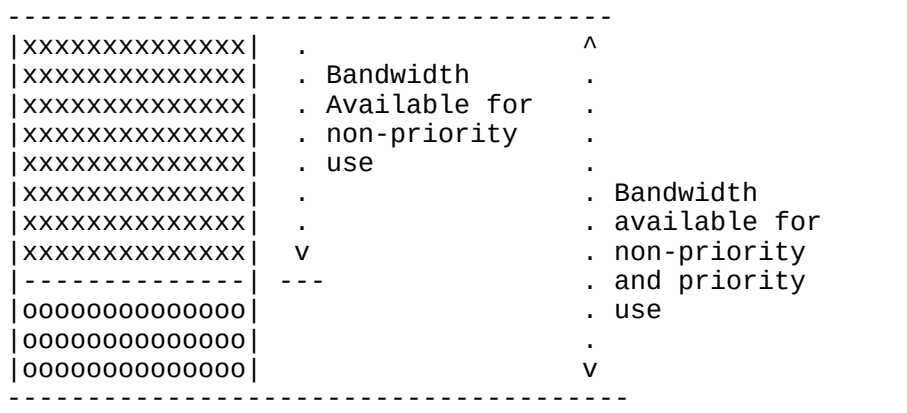


Figure 12: Full non-priority load & Full Aggregate load

A.3. Admission Priority with Priority Bypass Model (PrBM)

This section illustrates operations of admission priority when a simple Priority Bypass Model (PrBM) is used for bandwidth allocation across non-priority traffic and priority traffic. With the Priority Bypass Model, non-priority traffic is subject to resource based admission control while priority traffic simply bypasses the resource based admission control. In other words:

- when a non-priority session arrives, this session is subject to bandwidth admission control and is accepted if the current total load (aggregate over non-priority and priority traffic) is below the engineered/allocated bandwidth.

- when a priority session arrives, this session is admitted regardless of the current load.

A property of this model is that a priority session is never rejected.

The rationale for this simple scheme is that, in practice in some networks:

- the volume of priority sessions is very low for the vast majority of time, so it may not be economical to completely set aside bandwidth for priority sessions and preclude the utilization of this bandwidth by normal sessions in normal situations

- even in congestion periods where priority sessions may be more heavily used, those always still represent a fairly small proportion of the overall load which can be absorbed within the

safety margin of the engineered capacity limits. Thus, even if they are admitted beyond the engineered bandwidth threshold, they are unlikely to result in noticeable QoS degradation.

As with the MAM and RDM, an operator may map the Priority Bypass model onto the Engineered Capacity limits according to different policies. The operator may decide to configure the bandwidth limit for admission of non-priority traffic to the full engineered capacity limits; As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth limit for non-priority traffic to X. Alternatively, the operator may decide to configure the bandwidth limit for non-priority traffic to below the engineered capacity limits (so that the sum of the non-priority and priority traffic stays below the engineered capacity); As an example, if the engineered capacity limit on a given link is X, the operator may configure the bandwidth limit for non-priority traffic to 95% of X. Finally, the operator may decide to strike a balance in between. The considerations presented for these policies in the previous sections in the MAM and RDM contexts are equally applicable to the Priority Bypass Model.

Figure 13 illustrates the bandwidth allocation with the Priority Bypass Model.

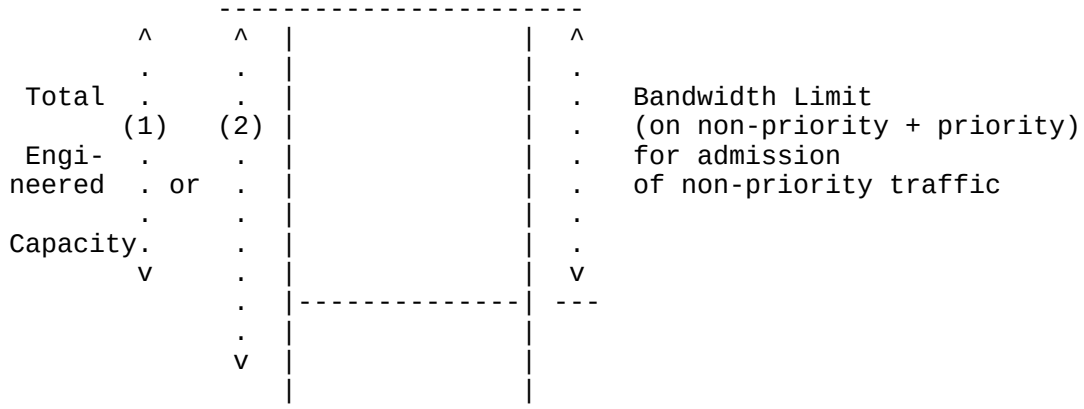


Figure 13: Priority Bypass Model Bandwidth Allocation

Figure 14 shows some of the non-priority capacity of this link being used. In this situation, both new non-priority and new priority sessions would be accepted.

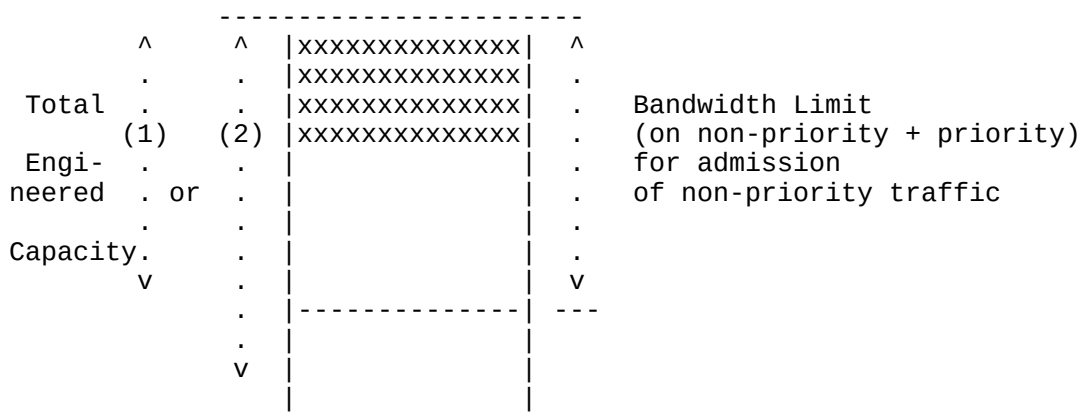


Figure 14: Partial load of non-priority calls

Figure 15 shows the same amount of non-priority load being used at this link, and a small amount of priority bandwidth being used. In this situation, both new non-priority and new priority sessions would be accepted.

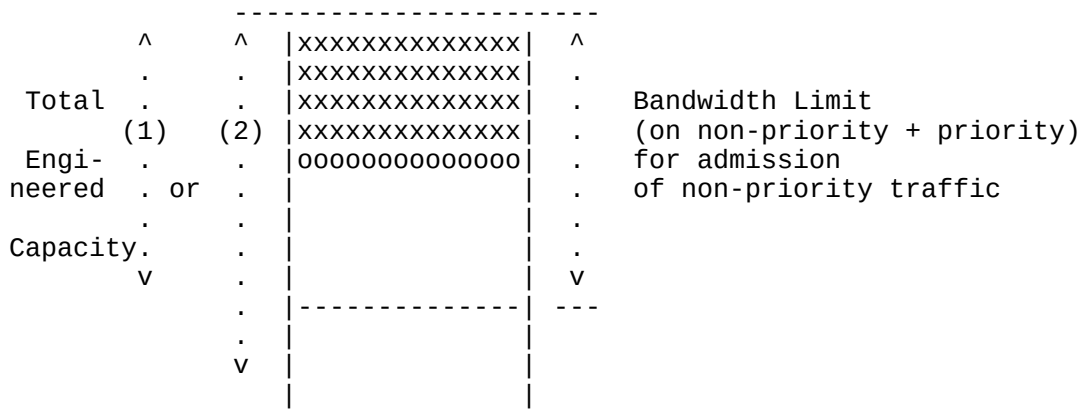


Figure 15: Partial load of non-priority calls & partial load of priority calls

Figure 16 shows the case where aggregate non-priority and priority load exceeds the bandwidth limit for admission of non-priority traffic. In this situation, any new non-priority session is rejected while any new priority session is admitted.

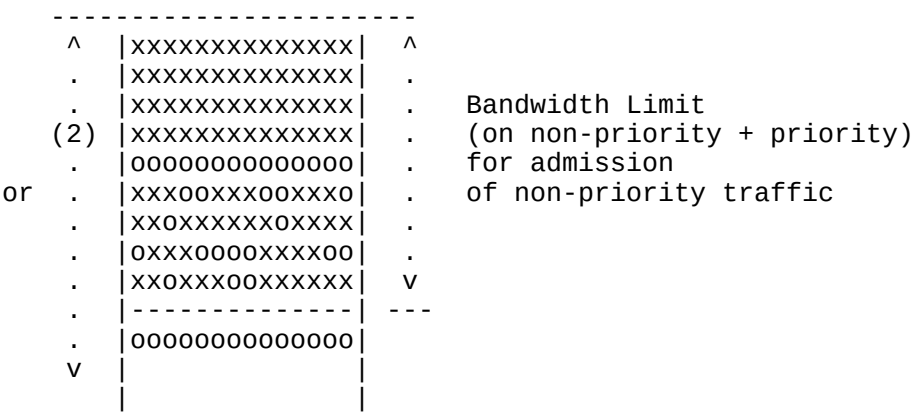


Figure 16: Full non-priority load

Appendix B. Example Usages of RSVP Extensions

This section provides examples of how RSVP extensions defined in this document can be used (in conjunctions with other RSVP functionality and SIP functionality) to enforce different hypothetical policies for handling prioritized sessions in a given administrative domain. This Appendix does not provide additional specification. It is only included in this document for illustration purposes.

We assume an environment where SIP is used for session control and RSVP is used for resource reservation.

We refer here to "Session Queueing" as the set of "session" layer capabilities that may be implemented by SIP user agents to influence their treatment of SIP requests. This may include the ability to "queue" session requests when those can not be immediately honored (in some cases with the notion of "bumping", or "displacement", of less important session requests from that queue). It may include additional mechanisms such as exemption from certain network management controls, and alternate routing.

We only mention below the RSVP policy elements that are to be enforced by PEPs. It is assumed that these policy elements are set at a policy area boundary by PDPs. The Admission Priority and Preemption Priority RSVP policy elements are set by PDPs as a result of processing the Application Level Resource Priority Policy Element (which is carried in RSVP messages).

If one wants to implement a prioritized service purely based on Session Queueing, one can achieve this by signaling prioritized

sessions:

- using "Resource-Priority" header in SIP
- not using Admission-Priority Policy Element in RSVP
- not using Preemption Policy Element in RSVP

If one wants to implement a prioritized service based on Session Queueing and on "prioritized access to network layer resources", one can achieve this by signaling prioritized sessions:

- using "Resource-Priority" header in SIP
- using Admission-Priority Policy Element in RSVP
- not using Preemption Policy Element in RSVP

Establishment of prioritized sessions will not result in preemption of any session. Different bandwidth allocation models can be used to offer different "prioritized access to network resources". Just as examples, this includes strict setting aside of capacity for prioritized sessions as well as simple bypass of admission limits for prioritized sessions.

If one wants to implement a prioritized service based on Session Queueing, on "prioritized access to network layer resources", and ensures that (say) "Prioritized-1" sessions can preempt "Prioritized-2" sessions, but non-prioritized sessions are not affected by preemption, one can do that by signaling prioritized sessions:

- using "Resource-Priority" header in SIP
- using Admission-Priority Policy Element in RSVP
- using Preemption Policy Element in RSVP with:
  - setup (Prioritized-1) > defending (Prioritized-2)
  - setup (Prioritized-2) <= defending (Prioritized-1)
  - setup (Prioritized-1) <= defending (Non-Prioritized)
  - setup (Prioritized-2) <= defending (Non-Prioritized)

If one wants to implement a prioritized service based on Session Queueing, on "prioritized access to network layer resources", and

ensure that prioritized sessions can preempt regular sessions, one could do that by signaling Prioritized sessions:

- using "Resource-Priority" header in SIP
- using Admission-Priority Policy Element in RSVP
- using Preemption Policy Element in RSVP with:
  - setup (Prioritized) > defending (Non-Prioritized)
  - setup (Non-Prioritized) <= defending (Prioritized)

If one wants to implement a prioritized service based on Session Queueing, on "prioritized access to network layer resources", and ensure that prioritized sessions can partially preempt regular sessions (i.e. Reduce their reservation size), one could do that by signaling prioritized sessions:

- using "Resource-Priority" header in SIP
- using Admission-Priority Policy Element in RSVP
- using Preemption in Policy Element RSVP with:
  - setup (Prioritized) > defending (Non-Prioritized)
  - setup (Non-Prioritized) <= defending (Prioritized)
- activate RFC4495 RSVP Bandwidth Reduction mechanisms

Authors' Addresses

Francois Le Faucheur
Cisco Systems
Greenside, 400 Avenue de Roumanille
Sophia Antipolis 06410
France

Phone: +33 4 97 23 26 19
Email: flefauch@cisco.com

James Polk
Cisco Systems
2200 East President George Bush Highway
Richardson, TX 75082-3550
United States

Phone: +1 972 813 5208
Email: jmpolk@cisco.com

Ken Carlberg
G11
123a Versailles Circle
Towson, MD 21204
United States

Email: carlberg@g11.org.uk

TSVWG WG
Internet-Draft
Expires: September 14, 2011
Intended Status: Standards Track (PS)
Updates: RFC 2205, 2210, & 4495 (if published as an RFC)

James Polk
Subha Dhesikan
Cisco Systems
March 14, 2011

Integrated Services (IntServ) Extension to Allow Signaling of Multiple Traffic Specifications and Multiple Flow Specifications in RSVPv1
draft-polk-tsvwg-intserv-multiple-tspec-06

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.

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.

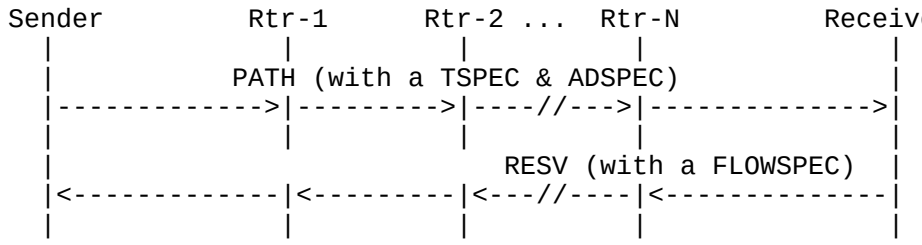


Figure 1. Concept of RSVP in a Single Direction

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.

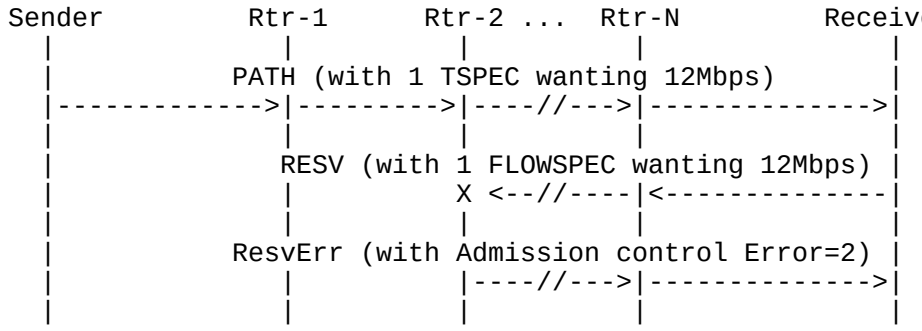


Figure 2. Concept of RSVP Rejection due to Limited Bandwidth

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:

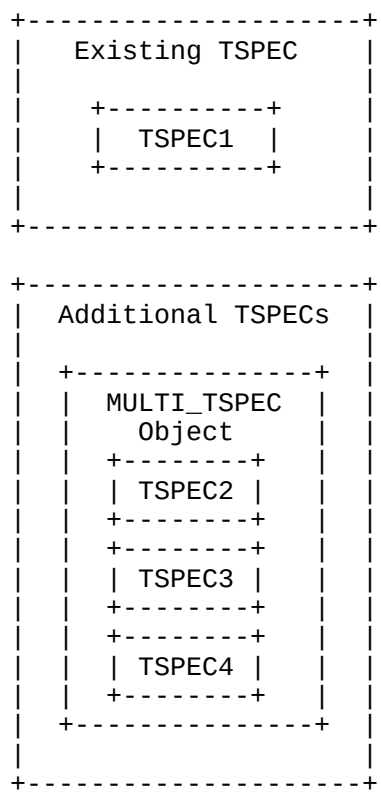


Figure 3. Encoding of Multiple Traffic Specifications in the TSPEC and MULTI_TSPEC objects

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:

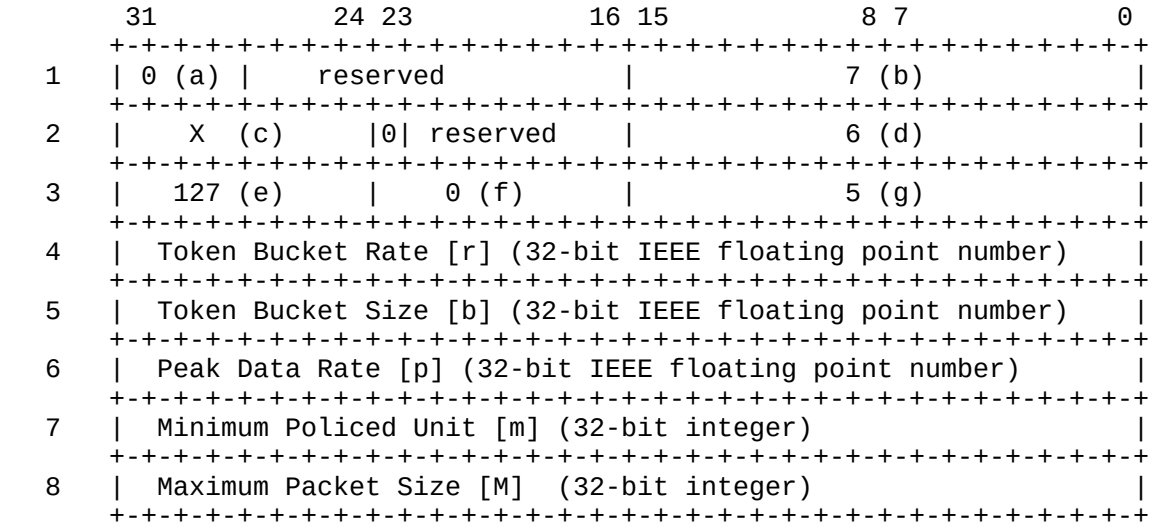
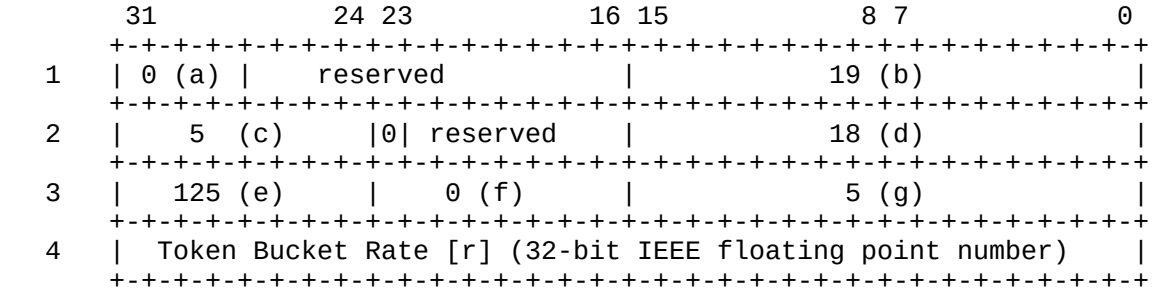


Figure 4. SENDER_TSPEC in PATH

(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.



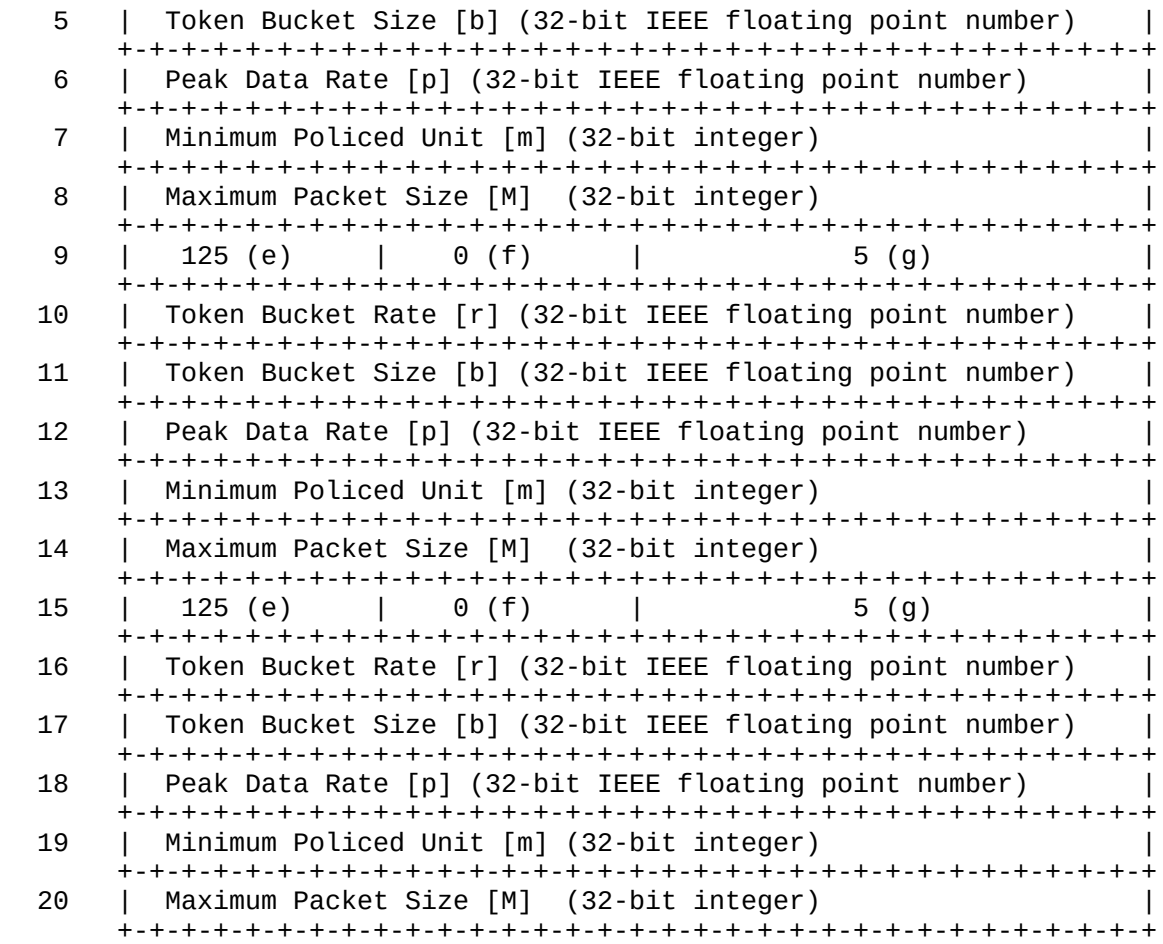


Figure 5. MULTI_TSPEC Object

(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.

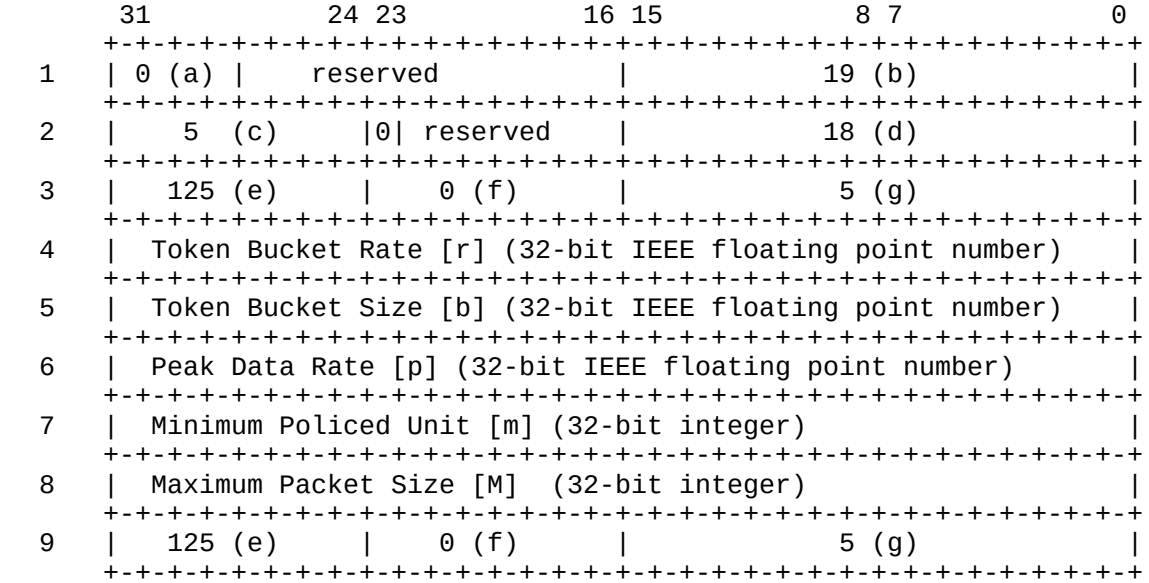
Each TSPEC is six 32-bit Words long (the per-service header plus the 5 values that are 1 Word each in length), therefore the length is in 6 Word increments for each additional TSPEC. Case in point, from the above Figure 5, Words 3-8 are the first TSPEC (2nd preferred), Words 9-14 are the next TSPEC (3rd 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:



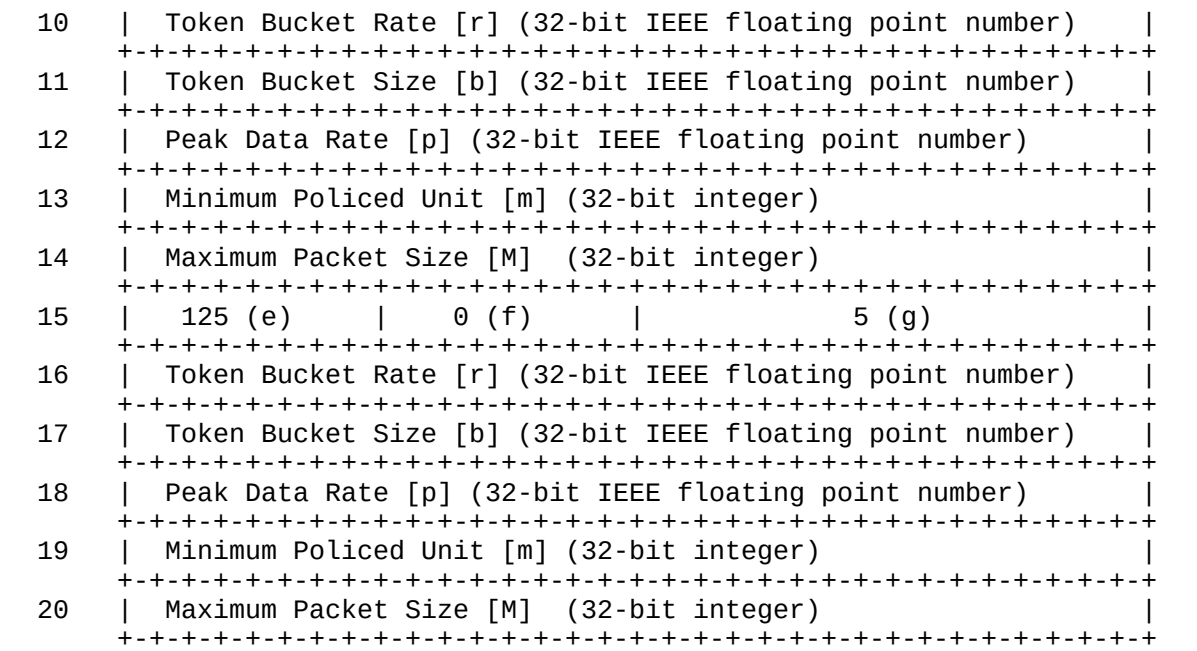


Figure 5. Multiple FLOWSPEC for Controlled-Load service

(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:

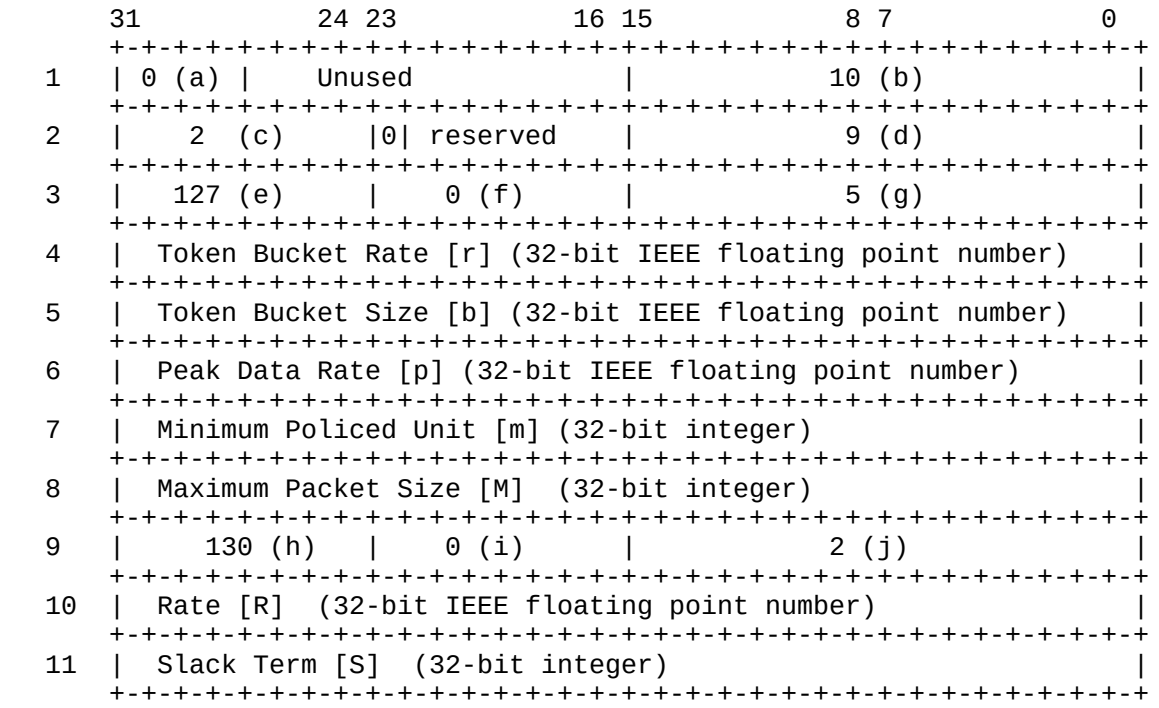
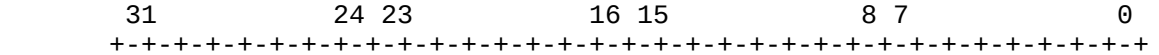


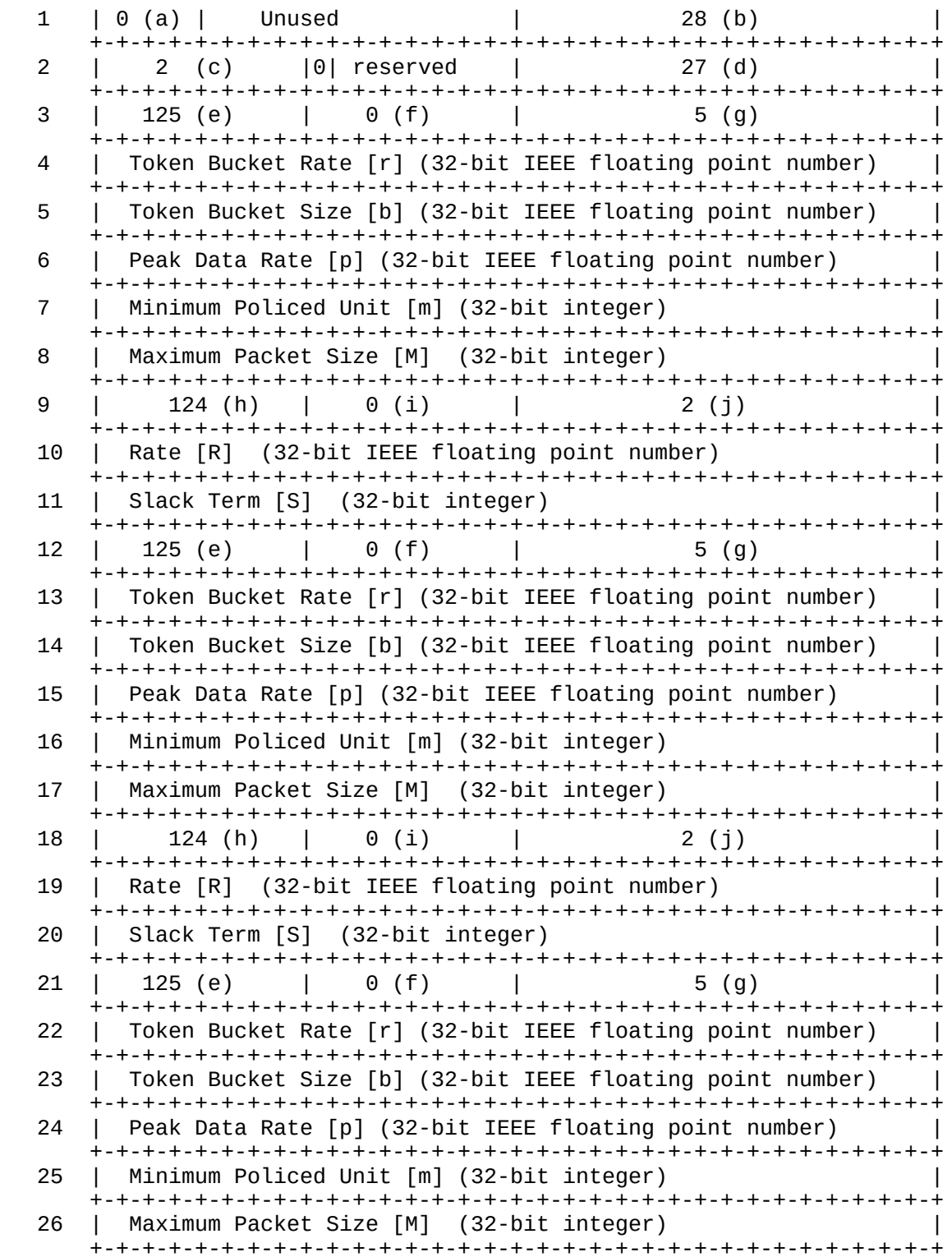
Figure 6. FLOWSPEC for Guaranteed service

(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.




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) |

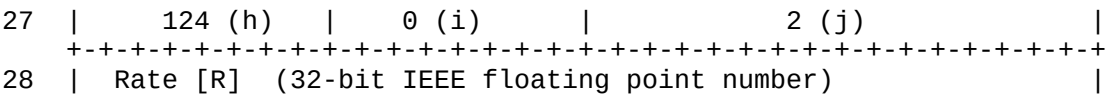


Figure 7. Multiple FLOWSPECs for Guaranteed service

(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] has the ability to partially

preempt existing reservations. However, it does not address the need that this draft 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 with 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 will extend the RFC 4495 defined error to work for previous hops while a reservation is being established.

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:

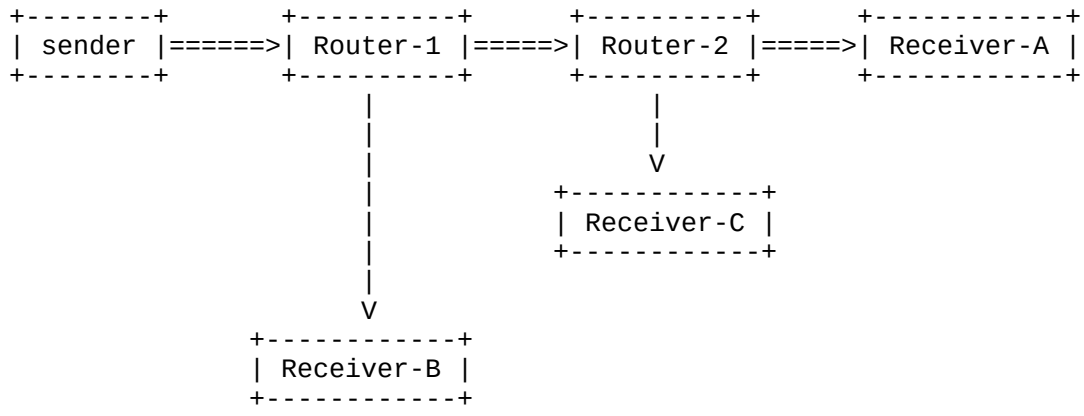


Figure 8. MULTI_TPSEC and Multicast

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.

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

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:

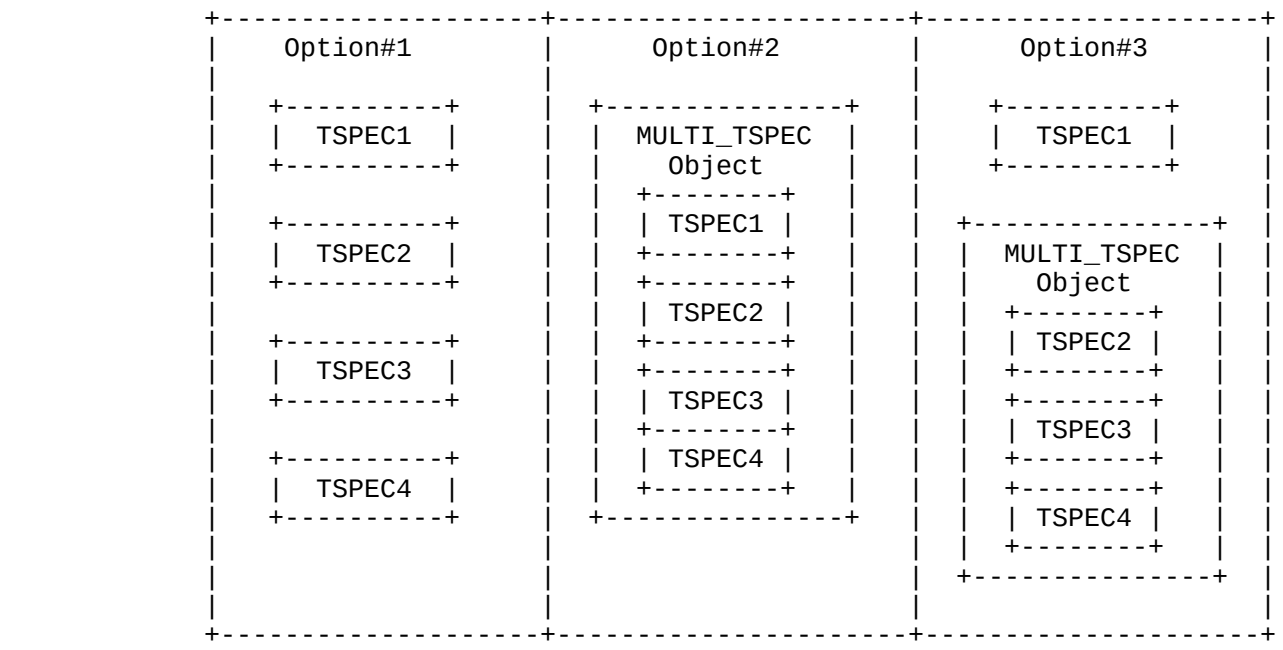


Figure 3. Concept of Option Choice

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.