

Extended Administrative Groups in MPLS-TE
draft-osborne-mpls-extended-admin-groups-02

Abstract

This document provides additional administrative groups (sometimes referred to as "link colors") to the IGP extensions for MPLS-TE.

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](#)].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on January 13, 2014.

Copyright Notice

Copyright (c) 2013 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must

include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	2
1.1.	Do we need more than 32 bits?	2
2.	Extended Administrative Groups sub-TLV	4
2.1.	Packet Format	4
2.2.	Admin group numbering	5
2.3.	Backward compatability	5
2.3.1.	AG and EAG coexistence	5
2.3.2.	Desire for unadvertised EAG bits	5
3.	Attribute filters in RSVP	6
3.1.	EXTENDED_SESSION_ATTRIBUTE_RA	6
3.2.	Populating the attribute filter fields	7
3.3.	Formatting a Path message	7
3.4.	Interpreting the attribute filter fields	9
4.	Security Considerations	9
5.	IANA Considerations	9
6.	Acknowledgements	10
7.	Normative References	10
	Author's Address	10

[1.](#) Introduction

MPLS-TE advertises 32 administrative groups (commonly referred to as "colors" or "link colors") using the Administrative Group sub-TLV of the Link TLV. This is defined for OSPFv2 [[RFC3630](#)], OSPFv3 [[RFC5329](#)] and ISIS [[RFC5305](#)].

This document adds a sub-TLV to the IGP TE extensions, "Extended Administrative Group". This sub-TLV provides for additional administrative groups (link colors) beyond the current limit of 32.

[1.1.](#) Do we need more than 32 bits?

The IGP extensions to support MPLS-TE (RFCs 3630 and 5305) define a link TLV known as Administrative Group (AG) with a limit of 32 AGs per link. This property comes from [section 6.2 of RFC 2702](#) [[RFC2702](#)]. RFCs 3630 and 5305 describe the mechanics of the TLV; the actual definition of the field comes from [RFC 2702](#):

"[Administrative Groups] can be used to implement many policies with regard to both traffic and resource oriented performance optimization. Specifically, ...[AGs] can be used to:

1. Apply uniform policies to a set of resources that do not need to be in the same topological region.
2. Specify the relative preference of sets of resources for path placement of traffic trunks.
3. Explicitly restrict the placement of traffic trunks to specific subsets of resources.
4. Implement generalized inclusion / exclusion policies.
5. Enforce traffic locality containment policies. That is, policies that seek to contain local traffic within specific topological regions of the network.

Additionally, resource class attributes can be used for identification purposes."

The use of 'Specifically' in [RFC2702](#) is not read as normative; that is, the purpose of the quoted text is not to limit the use of AGs to the six listed policies, they are given as examples. However, the listed policies make good grounds to justify increasing the limit from 32.

Networks have grown over time, and MPLS-TE has grown right along with them. Implementing all six policies with only 32 bits gives the operator only five bits per policy with two bits left over. This can be quite constraining; AGs are a bit mask, so five bits does not mean 32 possible values, it means 5. Running a country-wide or world-wide MPLS-TE network with only five possible values for each case is clearly too constraining.

Even if an operator wishes to use AGs to implement only a single policy it is possible to run out of bit values. One such use case is #5, using AGs to constrain traffic within specific topological regions of the network. A large network may well have far more than 32 geographic regions. One particular operator uses AGs to flag network regions down to the metro scale, e.g. Seattle, San Francisco, Dallas, Chicago, St. Louis, etc. MPLS-TE tunnels are then specified with affinities to include or exclude specific metro regions in their path calculation. It is clear that 32 may not be enough even for a US-based network, nevermind a worldwide network.

[illegible]

The Type of the sub-TLV for OSPF and ISIS is TBD. The Length is the size of the Extended Admin Group (EAG) value in bytes. The EAG may be of any length, but MUST be a multiple of 4 bytes. The only limits on EAG size are those which are imposed by protocol-specific or media-specific constraints (e.g. max packet length).

2.2. Admin group numbering

By convention, the existing Administrative Group TLVs are numbered 0 (LSB) to 31 (MSB). The EAG values are a superset of AG. That is, bits 0-31 in the EAG have the same meaning and MUST have the same values as an AG flooded for the same link.

2.3. Backward compatability

There are two things to consider for backward compatibility with existing AG implementations - how do AG and EAG coexist, and what happens if a node has matching criteria for unadvertised EAG bits?

2.3.1. AG and EAG coexistence

If a node advertises EAG it MAY also advertise AG. If a node advertises both AG and EAG then the first 32 bits of the EAG MUST be identical to the advertised AG. If the AG and EAG advertised for a link differ, the EAG MUST take priority. This allows nodes which do not support EAG to obtain some link color information from the network, but also allow for an eventual migration away from AG. If a node advertises EAG without AG then any receiving node SHOULD alert the network operator to this violation via the appropriate mechanism, e.g. syslog.

2.3.2. Desire for unadvertised EAG bits

The existing AG sub-TLV is optional; thus a node may be configured with a preference to include red or exclude blue, and be faced with a link that is not advertising a value for either blue or red. What does an implementation do in this case? It shouldn't assume that red is set, but it is also arguably incorrect to assume that red is NOT set, as a bit must first exist before it can be set to 0.

Practically speaking this has not been an issue for deployments, as many implementations always advertise the AG bits, often with a default value of 0x00000000. However, this issue may be of more concern once EAGs are added to the network. EAGs may exist on some nodes but not others, and the EAG length may be longer for some links than for others.

Each implementation is free to choose its own method for handling this question. However, to encourage maximum interoperability an implementation SHOULD treat specified but unadvertised EAG bits as if they are set to 0. A node MAY provide other (configurable) strategies for handling this case.

3. Attribute filters in RSVP

In addition to updating the IGP sub-TLV, RSVP needs to be extended to provide the ability to signal desired resource affinities. This section provides that update.

3.1. EXTENDED_SESSION_ATTRIBUTE_RA

This section provides the EXTENDED_SESSION_ATTRIBUTE_RA.

[NOTE: This section reads like EXTENDED_SESSION_ATTRIBUTE_RA is another C-Type of the SESSION_ATTRIBUTE Class. Whether it is implemented like this or whether it ultimately gets specified as a new Class is up for discussion and needs to be resolved prior to publication as an RFC.]

[RFC 3209](#) defines two types of SESSION_ATTRIBUTE, one with resource affinities and one without. The former is C-Type 1 and is referred to in this document as SESSION_ATTRIBUTE_RA. The latter is referred to as SESSION_ATTRIBUTE_NO_RA and is C-Type 7.

The Class and C-Type for EXTENDED_SESSION_ATTRIBUTE_RA are 207 and TBD, respectively. The format of the EXTENDED_SESSION_ATTRIBUTE_RA is:

```

      0               1               2               3
      0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0 1
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                     Attribute filter length                                     |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                                                                                                 |
//                                     Exclude-any                                     //
|                                                                                                                                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                                                                                                 |
//                                     Include-any                                    //
|                                                                                                                                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|                                                                                                                                 |
//                                     Include-all                                    //
|                                                                                                                                 |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```



```

|  Setup Prio  | Holding Prio  |      Flags      | Name Length  |
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+
|
//          Session Name          (NULL padded display string)      //
|
+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+---+

```

The Exclude-any, Include-any and Include-all fields are collectively referred to as the "attribute filter fields". All three attribute filter fields MUST be the same length. All fields in the EXTENDED_SESSION_ATTRIBUTE_RA MUST be interpreted exactly as they are in the SESSION_ATTRIBUTE_RA.

The attribute filter length is the sum of the lengths of the three attribute filter fields, in bytes. If the user wishes to convey 128 bits of information in each of the fields, the total length of the attribute filter fields is $3 \times 128 = 384$ bits. The attribute filter length is thus $384/8 = 48$ bytes. The next 4 bytes of the EXTENDED_SESSION_ATTRIBUTE_RA are fixed - setup priority, holding priority, flags and name length - and the remainder of the object is the Session Name. If the user wishes to convey 128 bits of information each of the three attribute filter fields and provides a 64-byte Session Name then the total length of this object in bytes is $4 + 48 + 4 + 64 = 120$ bytes.

[3.2.](#) Populating the attribute filter fields

Each attribute filter field MUST be the same length. As with the EAG sub-TLV, each attribute filter field is a multiple of four bytes in length. The length of each field MUST be at least the minimum length necessary to fully convey the headend's matching criteria, and SHOULD be no longer than that. For example, if the headend wishes to Include-any bits 1 and 17 then all three fields MUST be at least 4 bytes in length and SHOULD be no more than 4 bytes in length. If the headend wishes to Include-any bits 1, 17 and 150 then all three fields MUST be at least 20 bytes (160 bits) in length and SHOULD be no longer than 20 bytes.

[3.3.](#) Formatting a Path message

[NOTE: Actual bits and bytes to be sorted out later. For now, this section describes the desired behavior without prescribing specific packet formats. Open questions include - do we need to specify a new Class to hold EXTENDED_SESSION_ATTRIBUTE_RA, or can we reuse C-Type? What's legal? What's least likely to break existing implementations? Once that's decided, we also need a section on how to handle errors such as an invalid combination of resource affinities, etc.)]

In order to provide for backward compatibility, a node MAY signal both SESSION_ATTRIBUTE_RA and EXTENDED_SESSION_ATTRIBUTE_RA. This allows nodes which understand only SESSION_ATTRIBUTE_RA to use it, and nodes which understand EXTENDED_SESSION_ATTRIBUTE_RA (and thus also understand SESSION_ATTRIBUTE_RA) to use it. If a node signals both SESSION_ATTRIBUTE_RA and EXTENDED_SESSION_ATTRIBUTE_RA, the first 32 bits of the EXTENDED_SESSION_ATTRIBUTE_RA MUST match the SESSION_ATTRIBUTE_RA. If they do not match, a node SHOULD alert the operator as to this mismatch, and MUST ignore the SESSION_ATTRIBUTE_RA in favor of the EXTENDED_SESSION_ATTRIBUTE_RA. This is essentially the same behavior as in [section 2.3.1](#) of this document.

A node MUST NOT signal the combination of (SESSION_ATTRIBUTE_NO_RA and EXTENDED_SESSION_ATTRIBUTE_RA).

A node MAY signal just EXTENDED_SESSION_ATTRIBUTE_RA.

A node MAY signal just SESSION_ATTRIBUTE_RA.

A node MUST NOT signal both SESSION_ATTRIBUTE_RA and SESSION_ATTRIBUTE_NO_RA.

There are eight combinations of [SESSION_ATTRIBUTE_NO_RA, SESSION_ATTRIBUTE_RA, and EXTENDED_SESSION_ATTRIBUTE_RA] , including the combination where none of the three is advertised. Their legality is summarized in the following table, using SA_NO_RA, SA_RA and ESA_RA as abbreviated column headers:

Valid?	SA_NO_RA	SA_RA	ESA_RA
Y	x		
Y		x	
Y			x
Y		x	x
N	x	x	
N	x		x
N	x	x	x
N			

3.4. Interpreting the attribute filter fields

Since the attribute filter fields are of variable length, it is possible that an RSVP message may indicate more bits than a given node has advertised for a link. It is equally possible that an RSVP message may indicate fewer bits than a given node has advertised for a link. In all cases, the shorter of the two fields (the attribute filter field or the locally configured link admin group) MUST be padded with zeros so that both fields are of equal length.

Specifically, length mismatches are to be handled as follows:

The length of any single attribute filter field is A.

The length of the configured link attribute for a given link is C.

If $C > A$, a node MUST pad the received attribute filter field values with zeros so that $C == A$. A node MUST NOT alter the length of the signalled attribute filter field; the zero padding is only local to a given node.

If $A > C$, a node MUST pad the locally configured link attributes with zeros so that $A == C$. A node SHOULD NOT use this information to alter the length of the EAG sub-TLV that it floods.

[NOTE: [rfc3209](#) is unclear about how the attribute filter fields are to be used. The intent appears to be that any bits set to 1 in any of the three attribute filter fields must be considered a match for filtering purposes, and that any bits set to 0 are not used to match. In other words, there is no way to say "the following bits MUST be zero" for any of the attribute filter fields. A 0 in an attribute filter field says "I do not care what the value of this bit is". I am making this inference largely from the text in Include-any and Include-all which says "A null set...automatically passes". If existing implementations treat these fields differently (e.g. a 0 MUST be matched as a zero) then I'd like to know that so I can get the text in this section right.]

4. Security Considerations

This extension adds no new security considerations.

5. IANA Considerations

This document requests a sub-TLV allocation in both OSPF and ISIS, as well as an RSVP C-Type from Class 207.

6. Acknowledgements

Thanks to Santiago Alvarez, Rohit Gupta, Liem Nguyen, Tarek Saad, and Robert Sawaya for their review and comments.

7. Normative References

- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2702] Awduche, D., Malcolm, J., Agogbua, J., O'Dell, M., and J. McManus, "Requirements for Traffic Engineering Over MPLS", [RFC 2702](#), September 1999.
- [RFC3209] Awduche, D., Berger, L., Gan, D., Li, T., Srinivasan, V., and G. Swallow, "RSVP-TE: Extensions to RSVP for LSP Tunnels", [RFC 3209](#), December 2001.
- [RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", [RFC 3630](#), September 2003.
- [RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), October 2008.
- [RFC5329] Ishiguro, K., Manral, V., Davey, A., and A. Lindem, "Traffic Engineering Extensions to OSPF Version 3", [RFC 5329](#), September 2008.

Author's Address

Eric Osborne
Cisco Systems

Email: eosborne@cisco.com

