

LSR Working Group
Internet-Draft
Intended status: Standards Track
Expires: October 24, 2021

Yao. Liu
Shaofu. Peng
ZTE Corp.
April 22, 2021

**Flexible Algorithm with Bandwidth Constrains
draft-lp-lsr-fa-bandwidth-00**

Abstract

This document proposes extensions for IGP to allow the computation based on bandwidth constraints together with the existing metric in Flexible Algorithm.

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 <https://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 October 24, 2021.

Copyright Notice

Copyright (c) 2021 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 (<https://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](#)
- [2. Conventions used in this document](#) [2](#)
 - [2.1. Requirements Language](#) [2](#)
- [3. ISIS bandwidth constraint Sub-TLV of ISIS FAD Sub-TLV . . .](#) [3](#)
- [4. OSPF bandwidth constraint Sub-TLV of ISIS FAD Sub-TLV . . .](#) [3](#)
- [5. Calculation of Flex Algorithm Paths](#) [4](#)
 - [5.1. Calculation](#) [5](#)
 - [5.2. Illustration](#) [6](#)
- [6. Security Considerations](#) [6](#)
- [7. IANA Considerations](#) [7](#)
 - [7.1. ISIS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV](#) [7](#)
 - [7.2. OSPF Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV](#) [7](#)
- [8. References](#) [7](#)
 - [8.1. Normative References](#) [7](#)
 - [8.2. Informative References](#) [7](#)
- [Authors' Addresses](#) [8](#)

1. Introduction

[I-D.ietf-lsr-flex-algo] defines the Flex-Algorithm as a set of parameters consisting of calculation-type, metric-type and a set of constraints. The metric includes igp-metric, te-metric, delay-metric, and etc.

While the calculation of a path based on metrics is accumulative, e.g, the path calculated based on the delay-metric is that with the smallest sum of the metrics of each link. The bandwidth of a path depends on the link with the smallest bandwidth among the links that comprise the path.

It's more appropriate that the bandwidth acts as a constraint instead of a metric in the path computation.

This document proposes extensions for IGP to allow the computation based on bandwidth constraints together with the existing metric in FLEX-ALGO.

2. Conventions used in this document

2.1. Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "NOT RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in BCP

14 [RFC2119] [RFC8174] when, and only when, they appear in all capitals, as shown here.

3. ISIS bandwidth constraint Sub-TLV of ISIS FAD Sub-TLV

This document defines the ISIS bandwidth constraint Sub-TLV. It is a Sub-TLV of the ISIS FAD Sub-TLV [I-D.ietf-lsr-flex-algo] and has the following format:

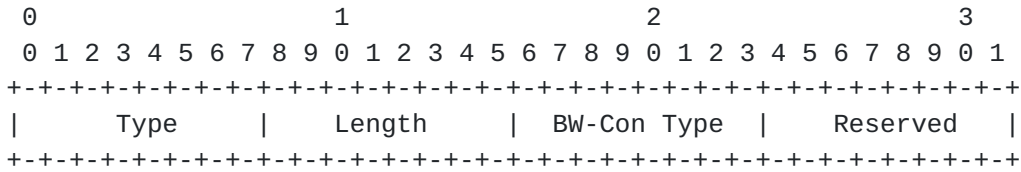


Figure 1: ISIS bandwidth constraint Sub-TLV

where:

- o Type: TBA
- o Length: 1 octets
- o BW-Con Type: 1 octets of bandwidth constraint information.

The BW-Con Type has the following values:

- 1: Choose the link with Maximum Link Bandwidth (MLB defined in [RFC5305]).
- 2: Choose the link with maximum Unidirectional Available Bandwidth (UAB defined in [RFC8570]).
- 3: Choose the link with the lowest bandwidth utilization rate.

The ISIS bandwidth constraint Sub-TLV MUST NOT appear more than once in an ISIS FAD Sub-TLV. If it appears more than once, the ISIS FAD Sub-TLV MUST be ignored by the receiver.

4. OSPF bandwidth constraint Sub-TLV of ISIS FAD Sub-TLV

This document defines the OSPF bandwidth constraint Sub-TLV. It is a Sub-TLV of the OSPF FAD Sub-TLV [I-D.ietf-lsr-flex-algo] and has the following format:

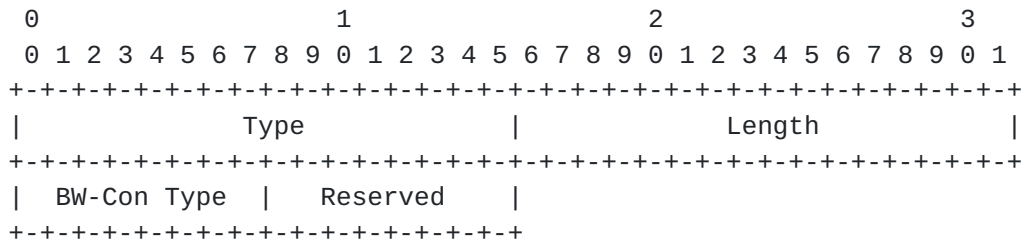


Figure 2: OSPF bandwidth constraint Sub-TLV

where:

- o Type: TBA
- o Length: 1 octets
- o BW-Con Type: 1 octets of bandwidth constraint information.

The BW-Con Type has the following values:

- 1: Choose the link with Maximum Bandwidth (MB defined in [[RFC3630](#)]).
- 2: Choose the link with maximum Unidirectional Available Bandwidth (UAB defined in [[RFC7471](#)]).
- 3: Choose the link with the lowest bandwidth utilization rate.

The OSPF bandwidth constraint Sub-TLV MUST NOT appear more than once in a OSPF FAD Sub-TLV. If it appears more than once, the OSPF FAD Sub-TLV MUST be ignored by the receiver.

5. Calculation of Flex Algorithm Paths

Routes that don't support the bandwidth constraint Sub-TLV MUST NOT participate in the corresponding Flex-Algo plane.

Existing rules for calculation of Flex-Algorithm paths specified in [[I-D.ietf-lsr-flex-algo](#)] are still applicable.

As for the new bandwidth constraint, first, select the optimal path(s) only according to the bandwidth constraint. If there are multiple candidate paths satisfying the bandwidth constraint, then select the best path based on the metric computation from these candidate paths.

5.1. Calculation

The result of IGP flex algo shortest path calculation based on bandwidth constraints is that each participating node in flex algo plane constructs a SPT (with itself as the root) per algorithm, which contains all other participating nodes as destination nodes.

Step1: Assume S is the root, and other nodes are D1, D2, D3,... Dn.

Step2: Find all paths from source node S to destination node D1. Suppose there is one or more paths, and it is recorded as path-1, path-2,..., path-m, where $m \geq 1$.

Step3: Initializes the current root node (cur-root) to S. Take two paths (such as path-1 and path-2) to compare to see who is more qualified for the constraint. Note that as mentioned above, the constraint conditions here refer to the combination of bandwidth constraint and metric, and compare based on bandwidth constraints firstly and then metric secondly. This can be compared in segments, that is, from the current root node to the destination node, finding a first crossed node which appears in two paths at the same time and is recorded as the current branch (cur-branch) node.

Step4: Compare the path segment from the current root node to the current branch node in the two paths to see who is more qualified for the path. If one of the two path segments is optimal and the other is suboptimal, it means that one of the two paths is optimal than the other. Add the optimal path to the "winning group". Note that if the path of the suboptimal is originally in the "winning group", the "winning group" should be cleared firstly and then new optimal path is added. The optimal path continues to compare with other paths, jump to Step3.

Otherwise, if the two path fragments are both optimal, the current root node (cur root) is updated as the current branch node. If the current root node reaches the destination node, the comparison is finished, and the two paths join the "winning group" to form ECMP (equal cost multi path equivalent multipath); otherwise, continue to find the next crossed node in the two paths as the new current branch node, and jump to Step4.

Step5: The above result will get the shortest paths from S to D1. Similarly, repeating Step2 for destination D2, D3,..., Dn respectively will get the shortest path from S to D2, D3,..., Dn respectively. Thus the shortest path tree with S as the root node in the flex algo plane is constructed.

5.2. Illustration

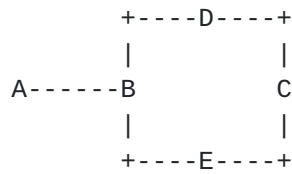


Figure 3: Reference Topology

As shown in Figure 3, there're 5 nodes in the given FA plane, whose metric type is delay metric and the bandwidth constrain is to choose the link with the maximum total bandwidth.

The bandwidth and metric for each link is shown in Table 1.

Link	Bandwidth	Metric
A-B	100	10
B-D	5	2
D-C	10	2
B-E	10	1
E-C	5	1

Table 1: Bandwidth and Metric

It is required to calculate the optimal path from A to C in this FA plane.

First, select the path who has the maximum bandwidth . Since the bandwidth of a path depends on the link with the smallest bandwidth, the maximum bandwidth s of path A-B-D-C and A-B-E-C are both 5. These two paths are chosen as candidate paths.

Second, compare the metric values of the candidate paths. The metric of path A-B-E-C is smaller, so it is chosen as the best path.

6. Security Considerations

TBD

7. IANA Considerations

7.1. ISIS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

Type: TBA

Description: bandwidth constraint Sub-TLV

Reference: This document.

7.2. OSPF Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV

Type: TBA

Description: bandwidth constraint Sub-TLV

Reference: This document.

8. References

8.1. Normative References

[I-D.ietf-lsr-flex-algo]

Psenak, P., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", [draft-ietf-lsr-flex-algo-13](#) (work in progress), October 2020.

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), DOI 10.17487/RFC2119, March 1997, <<https://www.rfc-editor.org/info/rfc2119>>.

[RFC8174] Leiba, B., "Ambiguity of Uppercase vs Lowercase in [RFC 2119](#) Key Words", [BCP 14](#), [RFC 8174](#), DOI 10.17487/RFC8174, May 2017, <<https://www.rfc-editor.org/info/rfc8174>>.

8.2. Informative References

[RFC3630] Katz, D., Kompella, K., and D. Yeung, "Traffic Engineering (TE) Extensions to OSPF Version 2", [RFC 3630](#), DOI 10.17487/RFC3630, September 2003, <<https://www.rfc-editor.org/info/rfc3630>>.

[RFC5305] Li, T. and H. Smit, "IS-IS Extensions for Traffic Engineering", [RFC 5305](#), DOI 10.17487/RFC5305, October 2008, <<https://www.rfc-editor.org/info/rfc5305>>.

[RFC7471] Giacalone, S., Ward, D., Drake, J., Atlas, A., and S. Previdi, "OSPF Traffic Engineering (TE) Metric Extensions", [RFC 7471](#), DOI 10.17487/RFC7471, March 2015, <<https://www.rfc-editor.org/info/rfc7471>>.

[RFC8570] Ginsberg, L., Ed., Previdi, S., Ed., Giacalone, S., Ward, D., Drake, J., and Q. Wu, "IS-IS Traffic Engineering (TE) Metric Extensions", [RFC 8570](#), DOI 10.17487/RFC8570, March 2019, <<https://www.rfc-editor.org/info/rfc8570>>.

Authors' Addresses

Liu Yao
ZTE Corp.

Email: liu.yao71@zte.com.cn

Peng Shaofu
ZTE Corp.

Email: peng.shaofu@zte.com.cn

