

LSR  
Internet-Draft  
Intended status: Standards Track  
Expires: 6 May 2026

J. Dong  
L. Zhang  
Huawei Technologies  
2 November 2025

IGP Flexible Algorithm with Time Constraint  
draft-dong-lsr-flex-algo-time-constraint-01

## Abstract

IGP Flexible Algorithm allows IGPs to compute constraint-based paths over the network using different types of metrics or constraints.

This document defines a new type of constraint called "Time Constraint", which can be used to control the time period during which a Flex-Algorithm takes effect for constraint-based path calculation. Such mechanism may be useful in network scenarios where either the traffic demand or the characteristics of the network varies as a function of time. The procedures of using Time Constraint in Flexible Algorithm is also described.

## 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 6 May 2026.

## Copyright Notice

Copyright (c) 2025 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 Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Requirements Language . . . . .	3
2. Time Constraints of Flexible Algorithm . . . . .	3
2.1. Time Constraint Sub-TLV of IS-IS FAD TLV . . . . .	3
2.2. Time Constraint Sub-TLV of OSPF FAD TLV . . . . .	5
3. Procedures . . . . .	6
3.1. Time-Constrained Flex-Algorithm Definition Advertisement . . . . .	7
3.2. Advertisement of Node Participation in Time-Constrained Flex-Algorithm . . . . .	7
3.3. Time-Constrained Flex-Algorithm Path Calculation . . . .	7
3.4. Time-Constrained Flex-Algorithm and Packet Forwarding . .	8
3.5. Time-Constrained Flex-Algorithm Switching . . . . .	8
4. Security Considerations . . . . .	9
5. IANA Considerations . . . . .	9
6. Acknowledgments . . . . .	9
7. References . . . . .	9
7.1. Normative References . . . . .	9
7.2. Informative References . . . . .	10
Authors' Addresses . . . . .	11

## 1. Introduction

IGP Flexible Algorithm [RFC9350] allows IGPs to compute constraint-based paths over the network using different types of metrics or constraints.

In some network scenarios, the traffic demand varies as a function of time, and in different time period, the network path for such traffic can be computed using different set of constraints.

In some other network scenarios, the characteristics of the network itself, such as the connectivity between nodes, the link attributes or the node attributes may change as a function of time, the network path meeting specific requirements may need to be recalculated at specific time, and the calculation result can be used for a specific time duration.

[I-D.ietf-tvr-requirements] specifies that the schedule should be considered for making forwarding decisions and calculating paths in the distributed routing scenario. However, there are no detail descriptions on how to calculate paths in a distributed manner and forward packets based on schedule.

This document defines a new type of constraint called "Time Constraint", which can be used to control the time period during which a Flex-Algorithm takes effect for constraint-based path calculation. The procedure of using Time Constraints in Flexible Algorithm is also described.

### 1.1. 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.

## 2. Time Constraints of Flexible Algorithm

The Time Constraints of Flexible Algorithm are used to specify the time when the Flexible Algorithm is in effect, and only the network nodes and links that are operational and meet other constraints of the Flexible Algorithm during the active period can be used for path computation. It also indicates the time at which the constraint-based path computation of this Flexible Algorithm should be performed. Similar to other constraints of Flexible Algorithm, Time Constraint is defined as one sub-TLV of the Flexible Algorithm Definition (FAD). The Time Constraints sub-TLV specifies the flags, initial time, end time, recurrence, slot number and the duration of each time slot when the Flexible Algorithm takes effect (in this document it is called the Flexible Algorithm is active). The encoding of Time Constraint in IS-IS and OSPF are specified in below subsections.

### 2.1. Time Constraint Sub-TLV of IS-IS FAD TLV

IS-IS Flex- Algo Time Constraint Sub-TLV is a sub-TLV of the IS-IS FAD Sub-TLV. It has the following format:

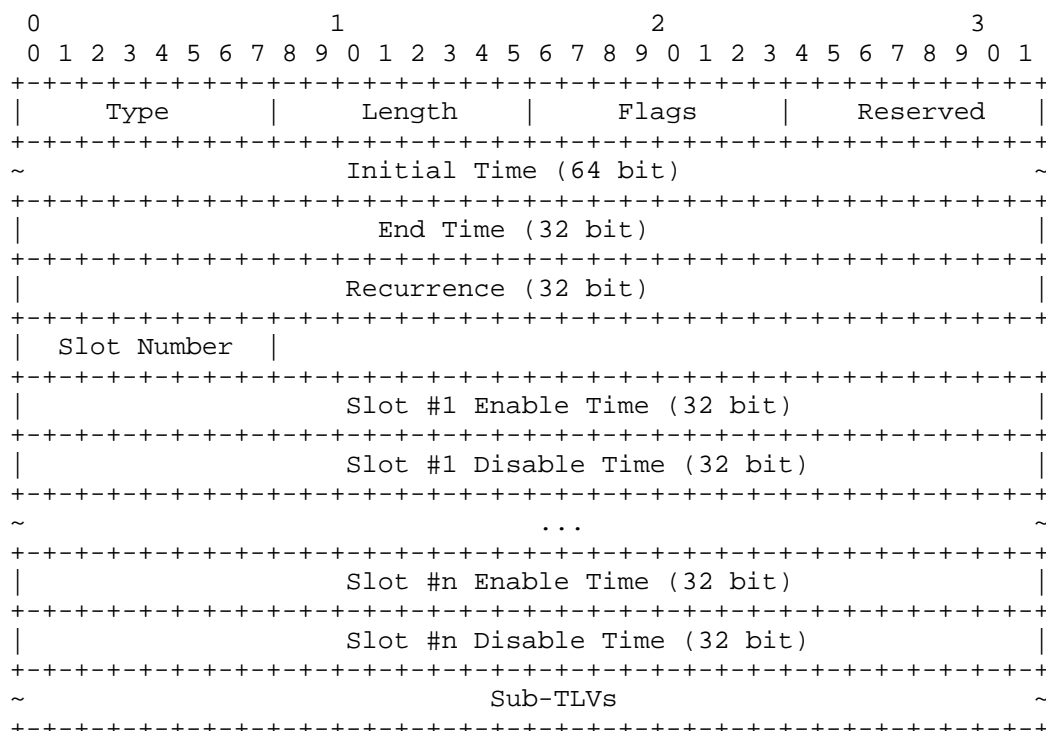


Figure 1: Encoding of Time Constraint Sub-TLV of IS-IS FAD TLV

Type (8 bits): TBD1, used to identify the type of Time Constraint Sub-TLV.

Length (8 bits): The size of the value field in octets.

Flags (8 bits): All flags are reserved for future use.

Reserved (8 bits): Reserved for future use. MUST be set to 0 on transmission and ignored at reception.

Initial Time (64 bits): The number of seconds since the epoch. It is the base timeline for the Flex-Algorithm's time constraint.

End Time (32 bits): The number of seconds since the initial time. It indicates the time when the Flex-Algorithm expires. When the end time of a Flex-Algorithm arrives, the routing table of this Flex-Algorithm should be deleted. When the End Time is set to 0, it indicates the Flex-Algorithm never expires.

Recurrence (32 bits): This is used to indicate the recurrence period of all the time slots of the Flexible Algorithm in seconds. When the recurrence is set to 0, it indicates the slots in the Flex-Algo are not cyclic.

Slot Number (8 bits): It is used to indicate the number of time slot in the following fields. Each time slot consists of enable time and disable time. The time slots indicate the duration when the Flex-Algorithm is considered active.

Slot# Enable Time (32 bits): The number of seconds since the initial time. It is used to indicate when the Flex-Algorithm start to take effect for a specific time slot.

Slot# Disable Time (32 bits): The number of seconds since the initial time. It is used to indicate when the Flex-Algorithm becomes ineffective.

Sub-TLVs: Optional sub-TLVs.

## 2.2. Time Constraint Sub-TLV of OSPF FAD TLV

OSPF Flex-Algo Time Constraint Sub-TLV is a sub-TLV of the OSPF FAD TLV. It has the following format:

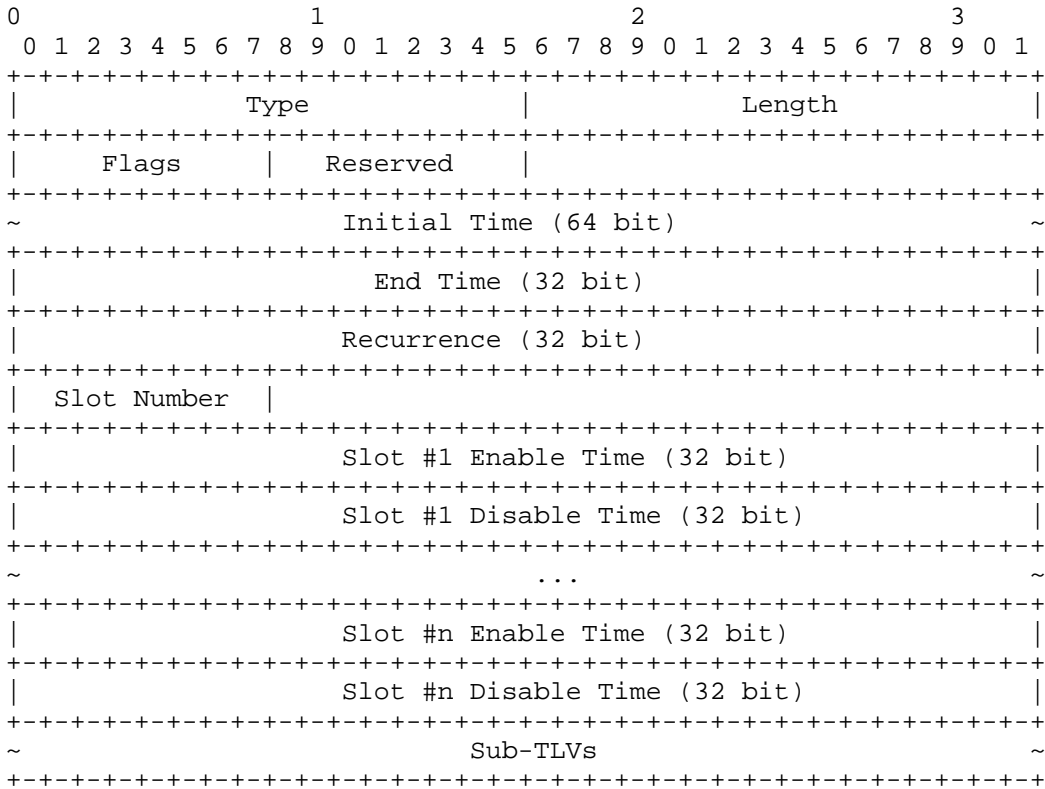


Figure 2: Encoding of Time Constraint Sub-TLV of OSPF FAD TLV

Type(16 bits): TBD2, used to identify the type of Time Constraint Sub-TLV.

Length (16 bits): The size of the value field in octets.

The description, format, and meaning of the flags, Initial Time, End time, Recurrence, Slot Number, Slot# Enable Time, Slot# Disable Time fields remain the same as in the Time Constraint Sub-TLV of IS-IS FAD TLV.

### 3. Procedures

The time constraint extension brings some updates to the Flex-Algorithm mechanism. This section describes the updates to Flex-Algorithm Definition advertisement, Flex-Algorithm path calculation, Flex-Algorithm-based packet forwarding and traffic switching between Flex-Algorithms.

It should be noted that the process of how a node receive time variable link-state information is out of scope of this document. It is assumed that each node can obtain the time variable information of the whole network by some means.

### 3.1. Time-Constrained Flex-Algorithm Definition Advertisement

The valid time slots and the recurrence of the Flex-Algorithm can be determined according to the predicted change of the traffic or the network. Multiple time-constrained Flex-Algorithms MAY be defined and advertised for path calculation in different time ranges. In this case there is usually no overlapping between different time-constrained Flex-Algorithms. A time-constrained Flex-Algorithm also includes with other constraints as defined in [RFC9350]. For example, the Metric Type, the Admin Groups, etc. Only a subset of the routers participating in the particular Flex-Algorithm needs to advertise the definition of the time-constrained Flex-Algorithm, in which the time constraints are encoded as one Sub-TLV of the FAD TLV (OSPF) or FAD sub-TLV (IS-IS).

### 3.2. Advertisement of Node Participation in Time-Constrained Flex-Algorithm

When a node is configured to participate in a time-constrained Flex-Algorithm, the node MUST advertise the Flex-Algorithm participation according to the data plane mechanism used with Flex-Algorithm. For SR-MPLS data plane, the mechanisms defined in [RFC8665], [RFC8666], [RFC8667] MUST be used, and for SRv6 data plane, the mechanisms defined in [RFC9352], [RFC9513] MUST be used.

Based on the End Time of the Flex-Algorithm, nodes participating in the Flex-Algorithm MAY withdraw the data plane identifiers it advertised for the Flex-Algorithm (e.g. the Algorithm-specific SR SIDs and SRv6 Locators).

### 3.3. Time-Constrained Flex-Algorithm Path Calculation

Constraint-based path computation of a time-constrained Flex-Algorithm SHOULD be performed before the enable time of each valid time slot of the Flex-Algorithm. A local timer MAY be used to determine the time at which the calculation is performed, so that the calculation is finished at the start time of the time slot. The mechanism for calculating Flex-Algorithm paths as defined in section 13 of [RFC9350] still applies here. The path computation SHOULD be based on the network topology at the moment of the start time, possibly with predicted changes of the network topology and attributes which may happen during the time slot. Only network nodes and links that are fully operational during the period of the Flex-

Algorithm time slot can be used in Flex-Algorithm path computation of that time slot. Topology changes which are received outside any time slot of a Flex-Algorithm are stored only and will not trigger path calculation of the Flex-Algorithm. This way, unnecessary constraint-based path calculation can be avoided.

#### 3.4. Time-Constrained Flex-Algorithm and Packet Forwarding

For a time-constrained Flex-Algorithm, the computed paths MUST be installed in the forwarding plane of network nodes which participate in the Flex-Algorithm before the enable time of each time slot of the Flex-Algorithm. For SR-MPLS and SRv6 data plane, the mechanisms of installing Flex-Algorithm-specific forwarding entries as described in section 14 of [RFC9350] applies here.

When a packet arrives at the edge of the network, according to the arrival time and local policy, the ingress edge node SHOULD determine which in effect time-constrained Flex-Algorithm will be used for forwarding the packet in the network, then the packet is encapsulated with the Flex-Algorithm-specific SR-MPLS Prefix SID or SRv6 SID corresponding to the selected Flex-Algorithm. The intermediate nodes SHOULD forward the packet according to forwarding entries of the Flex-Algorithm.

#### 3.5. Time-Constrained Flex-Algorithm Switching

In order to adapt to the changes in network traffic or network connectivity, multiple time-constrained Flex-Algorithms MAY be provisioned, the effective time slots of which are complementary. This can provide network paths to meet specific requirement at different time period. At the moments when a Flex-Algorithm become inactive (not in any of its valid time slot), traffic SHOULD be switched over to paths calculated with another active time-constrained Flex-Algorithm. Such switchover can be configured via local policy.

When a time-constrained Flex-Algorithm become inactive, network edge nodes MUST NOT use Flex-Algorithm-specific data plane encapsulation for any packets received after the disable time of the Flex-Algorithm. While the Flex-Algorithm-specific forwarding entries installed in the forwarding plane SHOULD NOT be deleted, this is to make sure that packets in-flight with Flex-Algorithm-specific data plane encapsulation can be forwarded correctly if the path is still available. Traffic SHOULD be switched to paths of other Flex-Algorithms which can meet the requirement while have complementary time slots.



When a Flex-Algorithm expires (the end time of the Flex-Algorithm arrives), the Flex-Algorithm-specific forwarding entries installed in the forwarding plane SHOULD be deleted after a local configured time period, so that the packets in-flight in the network can be forwarded correctly if the path is still available.

By the above mechanisms, there is no packet loss during the switchover of time-constrained Flex-Algorithms.

#### 4. Security Considerations

This document introduces no additional security vulnerabilities in addition to the ones as described in [RFC9350].

#### 5. IANA Considerations

IANA is requested to assign a new code point for "Flexible Algorithm Time Constraint Sub-TLV" under the "IS-IS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV" Registry.

Type	Description	Reference
TBD1	Flexible Algorithm Time Constraint	This document

Table 1

IANA is requested to assign a new code point for "Flexible Algorithm Time Constraint Sub-TLV" under the "OSPF Flexible Algorithm Definition TLV Sub-TLV" Registry.

Type	Description	Reference
TBD2	Flexible Algorithm Time Constraint	This document

Table 2

#### 6. Acknowledgments

The authors would like to thank XXX for the review and discussion of this document.

#### 7. References

##### 7.1. Normative References

- [RFC9350] Psenak, P., Ed., Hegde, S., Filsfils, C., Talaulikar, K., and A. Gulko, "IGP Flexible Algorithm", RFC 9350, DOI 10.17487/RFC9350, February 2023, <<https://www.rfc-editor.org/info/rfc9350>>.
- [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>>.
- [RFC8665] Psenak, P., Ed., Previdi, S., Ed., Filsfils, C., Gredler, H., Shakir, R., Henderickx, W., and J. Tantsura, "OSPF Extensions for Segment Routing", RFC 8665, DOI 10.17487/RFC8665, December 2019, <<https://www.rfc-editor.org/info/rfc8665>>.
- [RFC8666] Psenak, P., Ed. and S. Previdi, Ed., "OSPFv3 Extensions for Segment Routing", RFC 8666, DOI 10.17487/RFC8666, December 2019, <<https://www.rfc-editor.org/info/rfc8666>>.
- [RFC8667] Previdi, S., Ed., Ginsberg, L., Ed., Filsfils, C., Bashandy, A., Gredler, H., and B. Decraene, "IS-IS Extensions for Segment Routing", RFC 8667, DOI 10.17487/RFC8667, December 2019, <<https://www.rfc-editor.org/info/rfc8667>>.
- [RFC9352] Psenak, P., Ed., Filsfils, C., Bashandy, A., Decraene, B., and Z. Hu, "IS-IS Extensions to Support Segment Routing over the IPv6 Data Plane", RFC 9352, DOI 10.17487/RFC9352, February 2023, <<https://www.rfc-editor.org/info/rfc9352>>.
- [RFC9513] Li, Z., Hu, Z., Talaulikar, K., Ed., and P. Psenak, "OSPFv3 Extensions for Segment Routing over IPv6 (SRv6)", RFC 9513, DOI 10.17487/RFC9513, December 2023, <<https://www.rfc-editor.org/info/rfc9513>>.

## 7.2. Informative References

- [I-D.ietf-tvr-requirements]  
King, D., Contreras, L. M., and B. Sipos, "TVR (Time-Variant Routing) Requirements", Work in Progress, Internet-Draft, draft-ietf-tvr-requirements-01, 23 October 2023, <<https://datatracker.ietf.org/doc/html/draft-ietf-tvr-requirements-01>>.

Authors' Addresses

Jie Dong  
Huawei Technologies  
No. 156 Beiqing Road  
Beijing  
China  
Email: jie.dong@huawei.com

Li Zhang  
Huawei Technologies  
No. 156 Beiqing Road  
Beijing  
China  
Email: zhangli344@huawei.com