Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: July 17, 2022

Shaofu. Peng
Bin. Tan
Quan. Xiong
ZTE Corporation

January 13, 2022

IGP Flexible Algorithm with Deterministic Routing
draft-peng-lsr-flex-algo-deterministic-routing-00

Abstract

IGP Flex Algorithm proposes a solution that allows IGPs themselves to compute constraint based paths over the network, and it also specifies a way of using Segment Routing (SR) Prefix-SIDs and SRv6 locators, or pure IP prefix to steer packets along the constraint-based paths. This document describes how to compute deterministic paths within Flex-algo plane.

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 July 17, 2022.

Copyright Notice

Table of Contents

1. Introduction

IGP Flex Algorithm [I-D.ietf-lsr-flex-algo] proposes a solution that allows IGPs themselves to compute constraint based paths over the network, and it also specifies a way of using Segment Routing [RFC8402] Prefix-SIDs and SRv6 locators, or pure IP prefix [I-D.ietf-lsr-ip-flexalgo] to steer packets along the constraint-based paths. It specifies a set of extensions to ISIS, OSPFv2 and OSPFv3 that enable a router to send TLVs that identify (a) calculation-type, (b) specify a metric-type, and (c)describe a set of constraints on the topology, that are to be used to compute the best paths along the constrained topology. A given combination of calculation-type, metric-type, and constraints is known as an FAD (Flexible Algorithm Definition).

[RFC8655] describes the architecture of deterministic network and defines the QoS goals of deterministic forwarding: Minimum and maximum end-to-end latency from source to destination, timely delivery, and bounded jitter (packet delay variation); packet loss

ratio under various assumptions as to the operational states of the nodes and links; an upper bound on out-of-order packet delivery. In order to achieve these goals, deterministic networks use resource reservation, explicit routing, service protection and other means. A deterministic path is typically (but not necessarily) explicit routes so that it does not normally suffer temporary interruptions caused by the convergence of routing or bridging protocols.

IGP Flex-algo has the characteristic mentioned in [RFC8655]: under a single administrative control or within a closed group of administrative control. IGP Flex-algo supports Min Unidirectional Link Delay (defined in [RFC8570]) metric type to compute shortest paths with minimum delay, however, the cumulative delay is essentially the accumulation of transmission delay of all links, excluding node delay. In order to make up for this gap, it is necessary to enhance IGP flex-algo to compute the path with deterministic delay, i.e., including deterministic node delay and link transmission delay.

This document describes how to compute distributed shortest paths with deterministic delay metric within Flex-algo plane, as the basis of the whole distributed deterministic scheme. It should be noted that relying on this enhancement alone does not guarantee complete determinacy, it needs to be used in conjunction with other tools, such as creating additional backup explicit path with consistent delay metric for PREOF (Packet Replication, Elimination, and Ordering Functions), turning off local repair, smoothing the delay jitter during route convergence, providing deterministic forwarding mechanism, etc.

2. 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. Determinisitc Links

When a packet is forwarded to a link, the delay produced includes two parts: the first part is the dwell delay of the packet in the node, and the second part is the transmission delay of the packet on the link. In packet switching networks, priority based queuing scheme is generally used. It may give better average latency, but may have worst case latency. We call those links bound with a queue mechanism that can not guarantee node delay are non-determinisitc links.

On the contrary, those links bound with a queue mechanism that can provide deterministic node delay are called deterministic links. Typical queue mechanisms are:

- IEEE 802.1 WG has specified IEEE802.1Qch [CQF] which uses cyclic queuing and forwarding (CQF) mechanism and relies on time synchronization. According to CQF, the maximum delay experienced by a given packet is (H+1)*D, the minimum delay experienced by a given packet is (H-1)*D, and the delay jitter is 2*D, where H is the number of hops and D is cycle duration. Other variants based on CQF can avoid relying on time synchronization, but only the same cycle duration for all nodes. Basically, the packet received in the current sending window (i.e., cycle) will ensure that it can be sent in the next sending window, then the deterministic node delay, on average, is one cycle duration, or seveval cycle durations if the forwarding delay (from incoming port to outgoing port) inside the node can't be ignored.

- [I-D.peng-detnet-deadline-based-forwarding] introduced a deadline based forwarding mechanism that allow packet to control its expected dwell time in the node according to the planned deadline. There are two policies for deadline queue to schedule packets. For early sending policy, the end-to-end delay is H*(P~D), jitter is H*Q, where, H is the number of hops, P is the forwarding delay inside the node, D is the planned deadline; For punctual sending policy, the end-to-end delay is H*D, jitter is a single authorization time. That is, the packet received at any time will ensure that it can be sent in offset time P~D or D respectively for these two policies.

3.1. Deterministic Link Bound with CQF

A node may configure the CQF based packet scheduling parameter information for its local link, including CQF scheduling enable/disable, one or more cycle durations. Accordingly, for each cycle duration, the node delay/jitter attributes of the link will be obtained. The meanings of these parameters or attributes of the link are as follows:

- CQF scheduling enable/disable: the CQF scheduling algorithm can be enabled for a link, then the packets sent to that link will be scheduled by the CQF scheduling algorithm.

- Cycle duration: the duration of the cycle of CQF, which is also called cycle_size. One or more cycle_size with different lengths can be configured for a link, such as 10us, 20us, 30us, and so on.

- Node delay/jitter:

  * According to classical TSN CQF, for a given cycle_size, it can be deduced that the minimum delay in the node of the packet is 0, the maximum delay in the node is 2*cycle_size, the average delay in the node is cycle_size, and the delay jitter in the node is 2*cycle_size. The detailed reasons for these data are as follows: if a node receives a packet at the tail end of cycle i and sends that packet at the head end of cycle i+1, the resulting node delay, i.e., the minimum node delay, is 0; if a node receives a packet at the head end of cycle i and sends that packet at the tail end of cycle i+1, the resulting node delay, i.e., the maximum node delay, is 2*cycle_ size; the average node delay is one cycle_size, and the node delay jitter is 2*cycle_size. Each cycle_size corresponds to a different set of delay/jitter attributes.

  * However, for some variants based on TSN CQF, if the forwarding delay inside the node can't be ignored, e.g, wasting 2 cycle duration, then the minimum node delay, the maximum node delay, and the average node delay need to add 2 cycle_size respectively, but the node delay jitter is still 2*cycle_size.

3.1.1. ISIS Advertisement of Deterministic Link Bound with CQF

3.1.1.1. Advertisement of Forwarding Delay in Node

The forwarding delay is related to the chip implementation and is generally constant.

A new IS-IS sub-TLV is defined: the Forwarding Delay sub-TLV, which is advertised within TLV-22, 222, 23, 223, 141, 25. At most only one Forwarding Delay sub-TLV can be included.

The following format is defined for the Forwarding Delay sub-TLV:

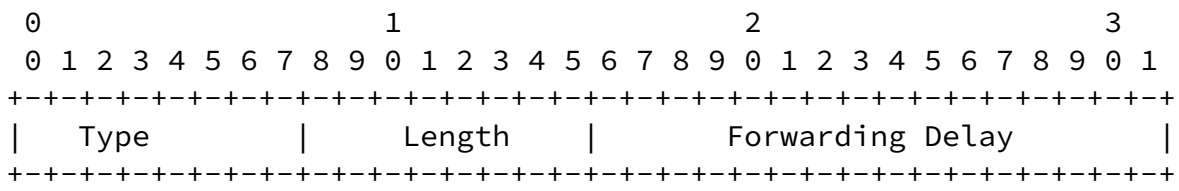

Figure 1

where:

Type: TBD

Length: 2

Forwarding Delay: The latency of packet from the incoming port (or generated from control plane) to the outgoing port, in units of microseconds. If the forwarding delay can be ignored, it is set to 0. If this sub-TLV is not advertised, the forwarding delay can be regarded as 0.

NOTE: for all links of a specific node, it may be possible that they have the same forwarding delay, therefore the forwarding delay can also be advertised by a unified node attribute. This would be considered in future versions.

3.1.1.2. Advertisement of CQF Parameters

A new IS-IS sub-TLV is defined: the Cycle Durations sub-TLV, which is advertised within TLV-22, 222, 23, 223, 141, 25. At most only one

Cycle Durations sub-TLV can be included.

The following format is defined for the Cycle Durations sub-TLV:

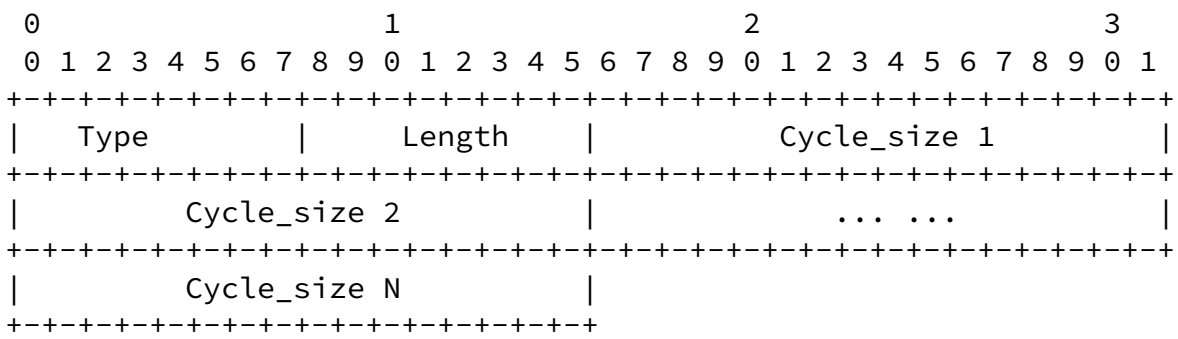

Figure 2

where:

Type: TBD

Length: 2*N, depending on the count of the cycle_size.

Cycle_size: The length of cycle duration, in units of microseconds. A link can support multiple cycle durations, for example, 10us, 20us, 30us, etc, each for a specific service requirement.

Only those links that enable CQF scheduling algorithm need to advertise the Cycle Durations sub-TLV, otherwise there is no need to advertise.

Note that the advertised cycle_size must be consistent with the CQF queue scheduling mechanism actually instantiated by the link in the

forwarding plane. If the forwarding plane does not instantiate a CQF queue scheduling supporting a certain cycle_size, which is however advertised in the Cycle Durations sub-TLV, the subsequent route computation may get wrong results.

For a given cycle_size, it can deduce the corresponding node delay and jitter attributes, so these attributes can no longer be explicitly included in the Cycle Durations sub-TLV. As mentioned

earlier, if the forwarding delay (assuming P) is not 0, the minimum node delay, the maximum node delay, and the average node delay need to take P into account respectively. P is replaced by ((P/cycle_size)+1)*cycle_size for deducing. That is:

- If P is 0, for a given cycle_size, the minimum node delay is 0, the maximum node delay is 2*cycle_size, the average node delay is cycle_size, and the node delay jitter is 2*cycle_size.

- If P is not 0, for a given cycle_size, the minimum node delay is ((P/cycle_size)+1)*cycle_size, the maximum node delay is ((P/cycle_size)+3)*cycle_size, the average node delay is ((P/cycle_size)+2)*cycle_size, and the node delay jitter is 2*cycle_size.

3.1.2. OSPF Advertisement of Deterministic Link Bound with CQF

To be defined in next version.

3.2. Deterministic Link Bound with Deadline

To be described in next version.

4. Deterministic Routes Computation

4.1. Bind CQF parameters with Flex-Algo

The binding relationship <algorithm, cycle_size> can be configured on one or more nodes participating in the same IGP Flex-algo plane, and then advertised in the IGP domain. If there are multiple binding relationship advertised for the same algorithm, it should choose to use the binding cycle_size contained in the FAD with the highest priority.

If a Flex-algo plane eventually uses a binding cycle_size, all links participated to the Flex-algo plane must be configured with CQF scheduling enabled and corresponding cycle_size, otherwise, links that do not meet the conditions must be excluded from the Flex-algo plane.

4.1.1. ISIS Advertisement of Flex-algo Binding CQF

The Flexible Algorithm definition can specify the binding cycle_size that are used to determine the deterministic delay metric for the computed path within the Flex-algo plane.

A new IS-IS sub-TLV is defined: the FAD Binding Cycle-size Sub-TLV, which is advertised within IS-IS Flexible Algorithm Definition Sub-TLV. At most only one FAD Binding Cycle-size Sub-TLV can be included.

The following format is defined for the FAD Binding Cycle-size Sub-TLV:

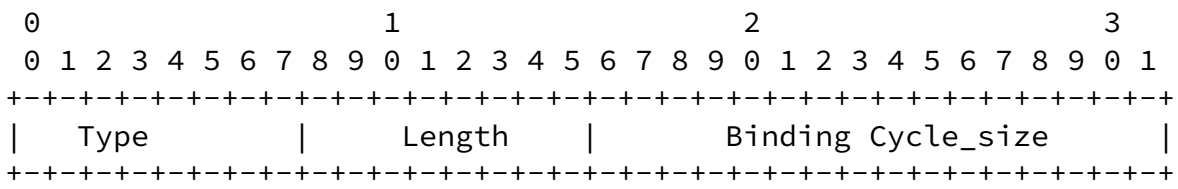

Figure 3

where:

Type: TBD

Length: 2

Binding Cycle_size: Cycle_size of CQF scheduling bound by Flex-algo, in units of microseconds.

The binding cycle_size contained in the FAD with the highest priority will take effect. If the FAD with the highest priority does not contain the FAD Binding Cycle-size Sub-TLV, assuming the Metric-Type is Min Unidirectional Link Delay, the traditional path considering only link transmission delay will be calculated, otherwise, the path will consider both node delay and link delay.

4.1.2. FAD Flags Extensions

A new flag (C-flag) is introduced to ISIS Flexible Algorithm Definition Flags Sub-TLV, to indicate to compute CQF based SPF path when Metric-Type is Min Unidirectional Link Delay. In other words, it will compute shortest path with minimum deterministic end-to-end delay, which contains accumulated node delay provided by CQF and accumulated link transmission delay.

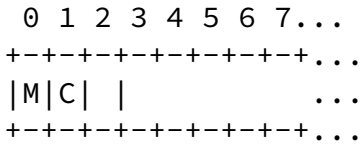

Figure 4

where:

C-flag: introduced by this document. When set, CQF based SPF path is computed.

4.1.3. OSPF Advertisement of Flex-algo Binding CQF

To be defined in next version.

4.1.4. CQF based Deterministic Routes Computation

This document reuse the existing Metric-Type, Min Unidirectional Link Delay, combined with the C-flag, to compute CQF based shortest path with minimum deterministic end-to-end delay, which contains accumulated node delay provided by CQF and accumulated link transmission delay.

NOTE: Whether new metric type need to be introduced needs to be discussed in the WG.

For a Flex-algo plane that bound to a specific cycle_size, the delay metric of a candidate path within the Flex-algo plane equals:

H * node delay, where H is the number of hops, and node delay can be deduced by the cycle_size and forwarding delay as above; plus

Accumulated link transmission delay;

From the source node to the destination node, the candidate path with minimum deterministic delay metric is the best one. This calculation result may be different from the traditional calculation result considering only link transmission delay, depending on the proportion of node delay.

The deterministic delay jitter of a candidate path within the Flex-algo plane equals:

node delay jitter, which is 2*cycle_size; plus

Accumulated link delay jitter, which is almost 0;

4.2. Bind Deadline parameters with Flex-Algo

To be described in next version.

5. IANA Considerations

TBD

6. Security Considerations

TBD.

7. Acknowledgements

TBD

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-18 (work in progress), October 2021.

[I-D.ietf-lsr-ip-flexalgo]
Britto, W., Hegde, S., Kaneriya, P., Shetty, R., Bonica, R., and P. Psenak, "IGP Flexible Algorithms (Flex-Algorithm) In IP Networks", draft-ietf-lsr-ip-flexalgo-04 (work in progress), December 2021.

[I-D.peng-detnet-deadline-based-forwarding]
Peng, S. and B. Tan, "Deadline Based Deterministic Forwarding", draft-peng-detnet-deadline-based-forwarding-00 (work in progress), January 2022.

[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>.

[RFC8402] Filsfils, C., Ed., Previdi, S., Ed., Ginsberg, L., Decraene, B., Litkowski, S., and R. Shakir, "Segment Routing Architecture", RFC 8402, DOI 10.17487/RFC8402, July 2018, <https://www.rfc-editor.org/info/rfc8402>.

[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>.

[RFC8655] Finn, N., Thubert, P., Varga, B., and J. Farkas, "Deterministic Networking Architecture", RFC 8655, DOI 10.17487/RFC8655, October 2019, <https://www.rfc-editor.org/info/rfc8655>.

8.2. Informative References

[CQF] "IEEE802.1Qch", 2017, <https://ieeexplore.ieee.org/document/7961303>.

Authors' Addresses

Shaofu Peng
ZTE Corporation
China

Email: peng.shaofu@zte.com.cn

Bin Tan
ZTE Corporation
China

Email: tan.bin@zte.com.cn

Quan Xiong
ZTE Corporation
China

Email: xiong.quan@zte.com.cn