Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: August 31, 2022

Shaofu. Peng
ZTE Corporation
Tony. Li
Juniper Networks

February 27, 2022

IGP Flexible Algorithm with Deterministic Routing
draft-peng-lsr-flex-algo-deterministic-routing-02

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 delay 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 August 31, 2022.

Table of Contents

1. Introduction 3
2. Requirements Language 4
3. Determinisitc Links 4
3.1. Deterministic Link Bound with CQF 5
3.2. Deterministic Link Bound with Deadline 6
4. Deterministic Delay Metric Extension to ISIS 7
4.1. CQF Scheduding Delay Intra Node Sub-Sub-TLV 8
4.2. Deadline Scheduding Delay Intra Node Sub-Sub-TLV 10
4.2.1. Another Simplified Extension 11
5. Deterministic Delay Metric Extension to OSPF 12
6. Announcement Suppression 12
7. Deterministic Routes Computation 12
7.1. Bind CQF Scheduling parameters with Flex-Algo 13
7.1.1. ISIS Advertisement of Flex-algo Binding CQF 13
7.1.2. OSPF Advertisement of Flex-algo Binding CQF 14
7.2. Bind Deadline Scheduling parameters with Flex-Algo . . . 14
7.2.1. ISIS Advertisement of Flex-algo Binding Deadline . . 14
7.2.2. OSPF Advertisement of Flex-algo Binding Deadline . . 15
7.3. CQF based Deterministic Routes Computation 16
7.4. Deadline based Deterministic Routes Computation 16
8. Routing Convergence and Redundance Considerations 18
9. Examples of Deterministic delay SPF 20
9.1. CQF Based Deterministic Delay SPF Path Example 20
9.2. Deadline Based Deterministic Delay SPF Path Example . . . 21
10. Use Cases 23
11. IANA Considerations 24
11.1. ISIS Deterministic Delay Metric Sub-TLV 24
11.2. Sub-Sub-TLVs for ISIS Deterministic Delay Metric Sub-TLV 24
11.3. IGP Metric-Type Registry 24
11.4. ISIS Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV 25
11.5. OSPF IANA considerations 25
12. Security Considerations 25
13. Acknowledgements 25
14. References 25
14.1. Normative References 25
14.2. Informative References 26
Authors' Addresses 27

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 redundant deterministic delay path with consistent delay metric for PREOF (Packet Replication, Elimination, and Ordering Functions), smoothing the delay jitter during route convergence, providing deterministic forwarding mechanism, admission control, 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. [SP-LATENCY] analyzes the guaranteed latency with the traditional strict priority scheme, and shows that low bounded latency is achievable when high priority traffic is constrained in low utilization, but deteriorates quickly with increasing utilization of high priority traffics. DiffServ [RFC2475] with strict priority has been widely deployed in the network, the existing non-deterministic service flow may set the highest priority, so it is difficult to support deterministic services based on it without any modification. 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. Therefore, other new scheduling mechanisms need to be introduced, and their scheduling priority is higher than that of the traditional strict priority queue. The typical queue mechanisms are as follows:

- IEEE 802.1 WG has specified IEEE802.1Qav [CBS] which uses credit-based shaper mechanism to assign packets to different queues by giving a credit value which is related with reserved bandwidth. The credit values of different transmission queues will automatically change with the packet transmission process, which will ensure that the packet with lower priority will also get transmission. CBS shaper is similar to Weighted Fair Queuing (WFQ), and they all control the sending of packets based on reserved bandwidth. The worst-case delay calculation of class A of CBS is relatively simple, but other classes are complex. For class A traffic, the queuing delay equals to the maximum size of the interference frame (such as 2000 octes) divided by the port bandwidth.

- 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 intra node (from incoming port to outgoing port) 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 in-time policy, the end-to-end delay is H*(F~D), jitter is H*Q, where, H is the number of hops, F is the forwarding delay intra node, D is the planned deadline, and Q is the scheduling delay; For on-time policy, the end-to-end delay is H*D, jitter is 0 (however there may be one authorization time due to the granularity of queue scheduling). That is, the packet received at any time will ensure that it can be sent in offset time F~D or D respectively for these two policies.

This document mainly describes the deterministic link based on CQF or Deadline algorithm. Other algorithms will be described in the future.

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 one 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 intra 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.

3.2. Deterministic Link Bound with Deadline

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

- Deadline scheduling enable/disable: the deadline scheduling algorithm can be enabled for a link, then the packet forwarded to the link will be scheduled by the deadline based packet scheduling algorithm. The dwell time of the packet in the node does not exceed the maximum allowable dwell time D, where, D = forwarding delay intra node (F) + specific deadline scheduling delay (Q).

- Supported deadline scheduling delay set: the set composed of one or more deadline scheduling delays <Q1, Q2, ..., Qn>, assuming that Q1 is the minimum and Qn is the maximum in the set. Generally, the difference between two adjacent elements in the set is fixed, for example, a fixed interval (I).

- Scheduling policy: for each scheduling delay Q, there are two scheduling policies: in-time policy and on-time policy. In case

of in-time policy, the scheduling delay of the packet may be sent to the outgoing port when it does not reach Q; In the on-time policy, the packet is sent to the outgoing port only when the scheduling delay of the packet is equal to Q. Therefore, for in-time policy, the actual dwell time of the packet in the node is within the range [F, F+Q], i.e., the minimum node delay is F, the maximum node delay is F+Q, and the node delay jitter is Q; For on-time policy, the actual dwell time of the packet in the node is equal to F+Q, i.e., the minimum node delay is F+Q, the maximum node delay is also F+Q, and the node delay jitter is 0 (however, there may be one authorization time due to the granularity of queue scheduling).

4. Deterministic Delay Metric Extension to ISIS

This document registers a new sub-TLV, Deterministic Delay Metric, in the "Sub-TLVs for IS-IS Sub-TLVs for TLVs Advertising Neighbor Information" registry, to distribute deterministic delay information. The deterministic delay advertised by this sub-TLV MUST be the delay from the local neighbor to the remote neighbor (i.e., the forward-path latency). The format of this sub-TLV is shown in the following diagram:

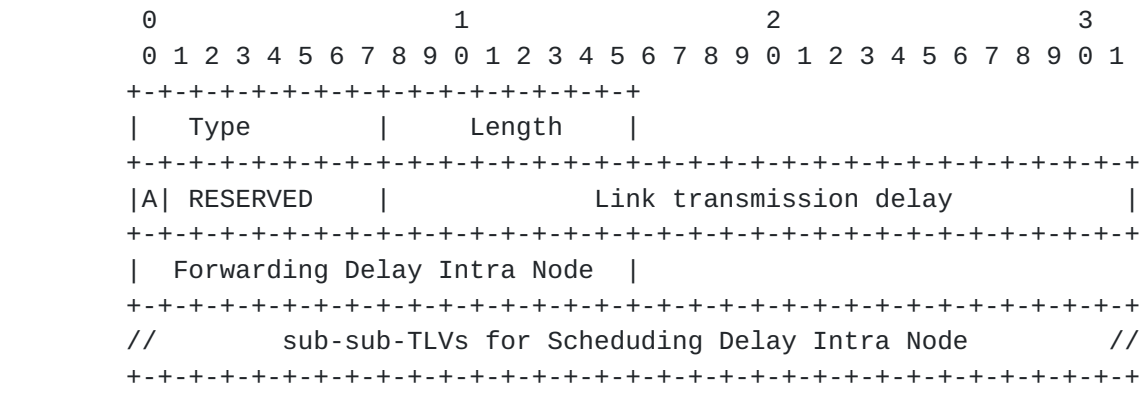

Figure 1

where:

Type: TBD

Length: Variable.

A bit: This field represents the Anomalous (A) bit. The A bit is set when one or more measured values of link transmission delay exceed a configured maximum threshold. The A bit is cleared when the measured value falls below its configured reuse threshold. If

the A bit is cleared, the sub-TLV represents steady-state link transmission delay.

RESERVED: This field is reserved for future use. It MUST be set to 0 when sent and MUST be ignored when received.

Link transmission delay: This 24-bit field carries the average measured link transmission delay value (in microseconds) over a configurable interval, encoded as an integer value. Implementations MAY also permit the configuration of an offset value (in microseconds) to be added to the measured delay value, to facilitate the communication of operator-specific delay constraints. When the delay value is set to the maximum value 16,777,215 (16.777215 seconds), then the delay is at least that value and may be larger.

Forwarding Delay Intra Node: This 16-bit field carries the forwarding delay value (in microseconds) intra node. It represents the latency of packet from the incoming port (or generated from control plane) to the outgoing port. If the forwarding delay can be ignored, it is set to 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.

sub-sub-TLVs for Scheduding Delay Intra Node: Optional sub-sub-TLVs are contained to indicate the scheduling delay that is related to the specific scheduling algorithm such as CQF, deadline, etc. If this field is absent, the scheduling delay is unknown. Typically, a link may enable a single scheduling algorithm to get deterministic scheduling delay, so that a single sub-sub-TLV is included. However, it is possible for a link to enable multiple different scheduling algorithms, as long as these algorithms can coordinate the forwarding resources, in this case, multiple sub-sub-TLVs are included. Supported Sub-sub-TLVs are specified in the following sub-sections.

4.1. CQF Scheduding Delay Intra Node Sub-Sub-TLV

CQF Scheduding Delay Intra Node Sub-Sub-TLV is an optional Sub-Sub-TLV of Deterministic Delay Metric Sub-TLV. At most only one CQF Scheduding Delay Intra Node Sub-Sub-TLV can be included.

The following format is defined for the CQF Scheduding Delay Intra Node Sub-Sub-TLV:

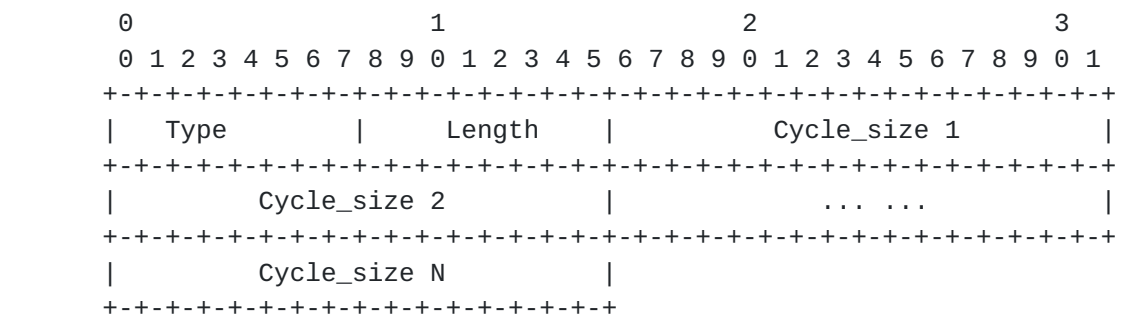

Figure 2

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 CQF Scheduding Delay Intra Node Sub-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 CQF Scheduding Delay Intra Node Sub-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 CQF Scheduding Delay Intra Node Sub-Sub-TLV.

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

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

4.2. Deadline Scheduding Delay Intra Node Sub-Sub-TLV

Deadline Scheduding Delay Intra Node Sub-Sub-TLV is an optional Sub-Sub-TLV of Deterministic Delay Metric Sub-TLV. At most only one Deadline Scheduding Delay Intra Node Sub-Sub-TLV can be included.

The following format is defined for the Deadline Scheduding Delay Intra Node Sub-Sub-TLV:

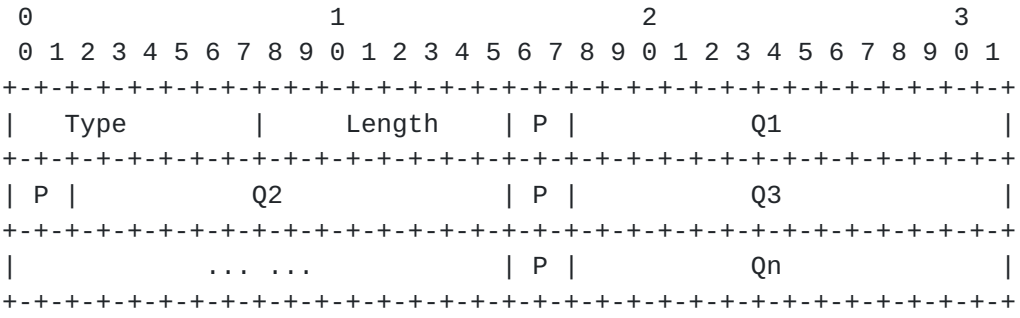

Figure 3

where:

Type: TBD

Length: 2*N, depending on the count of the supported deadline scheduling delay.

Q: Indicates the scheduling delay set, <Q1, Q2, ..., Qn>, supported by the link, in units of microseconds. For each supported scheduling delay, the highest two bits represent the scheduling policy P. The value of scheduling policy P can be:

0, not defined yet;

1, indicates that it supports the in-time policy;

2, indicates that it supports the on-time policy;

3, indicates that it supports both in-time policy and on-time policy.

As mentioned earlier, given the scheduling delay Q and its scheduling policy, combined with the forwarding delay F intra node, the corresponding delay and jitter attributes in the node can be derived. Therefore, these attributes can no longer be explicitly included in the Deadline Scheduding Delay Intra Node Sub-Sub-TLV.

Note that the scheduling delay Q advertised in the Deadline Scheduding Delay Intra Node Sub-Sub-TLV must be consistent with the deadline queue scheduling mechanism actually instantiated by the link in the forwarding plane. If the forwarding plane does not instantiate the deadline queue scheduling supporting a certain scheduling delay Q, which is however advertised in the Deadline Scheduding Delay Intra Node Sub-Sub-TLV, the subsequent route computation may get wrong results.

4.2.1. Another Simplified Extension

If the set <Q1, Q2, ..., Qn> to be advertised contains too many elements, and the difference between two adjacent elements in the set is a fixed interval (I), and the scheduling policy supported for all elements are same, another more simplified extension, the Simplified Deadline Scheduding Delay Intra Node Sub-Sub-TLV, can be defined as below.

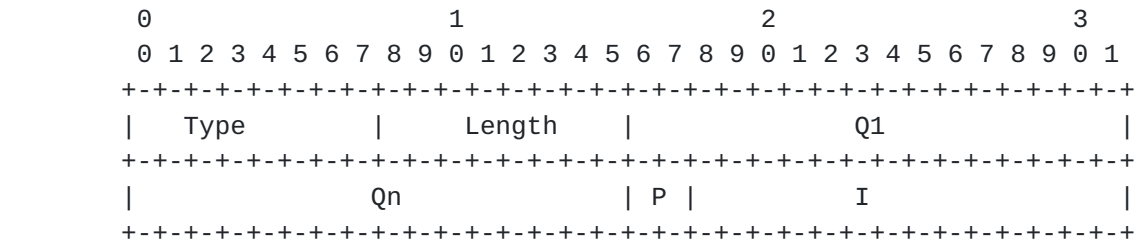

Figure 4

where:

Type: TBD

Length: 6.

Q1: the minimum scheduling delay, in units of microseconds.

Qn: the maximum scheduling delay, in units of microseconds.

I: the fixed interval between any two adjacent elements in the set, in units of microseconds. The highest two bits represent the scheduling policy P. The value of scheduling policy P can be:

0, not defined yet;

1, indicates that it supports the in-time policy;

2, indicates that it supports the on-time policy;

3, indicates that it supports both in-time policy and on-time policy.

5. Deterministic Delay Metric Extension to OSPF

To be defined in next version.

6. Announcement Suppression

The value of Deterministic Delay Metric defined in this document contains node delay provided by instantiated scheduling algorithm and link transmission delay provided by some measure mechanisms. For the announcement of the node delay part, it is constant and depend on the capability of instantiating the scheduling algorithm. However, for the announcement of the link transmission delay part, a measure mechanism may frequently produce different measurements. Please refer to [RFC8570] section 6 for the same principle of announcement suppression.

7. Deterministic Routes Computation

In order to use the deterministic link resources in the network to compute a deterministic delay SPF path, corresponding Flex-algo plane need to be created. To dinstinguish between traditional low latency SPF path (based on metric type "Min Unidirectional Link Delay") and deterministic low latency SPF path introduced in this document, new metric type, i.e., Deterministic Delay Metric, will be defined and used in Flexible Algorithm Definition (FAD).

- Metric-Type: TBD, to be used during the calculation of deterministic low latency SPF path.

Additional FAD constraints are also necessary, to bind individual item from the scheduling delay set.

It is possible to create multiple flex-algo instances each binding to different scheduling delay for different service requirements.

Note that sometimes from the perspective of the end-to-end delay requirements of the service flow, the node delay of the ingress PE node can be ignored and regarded as 0. However, this has no implication for the rules of deterministic low latency SPF path

calculation. During the deterministic low latency SPF path calculation, we must always take the node delay into account, because any node may be the intermediate node of an SPF path and its node delay is taken into account. But if the node takes itself as the root and does not consider its node delay when calculating the deterministic low latency SPF path, the calculation results may be inconsistent, resulting in the loop.

7.1. Bind CQF Scheduling 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.

7.1.1. 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 CQF Sub-Sub-TLV, which is advertised within IS-IS Flexible Algorithm Definition Sub-TLV. At most only one FAD Binding CQF Sub-Sub-TLV can be included.

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

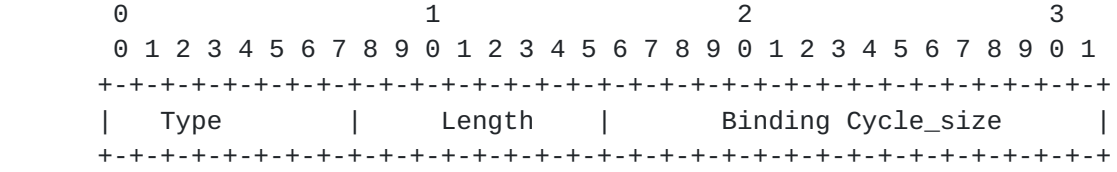

Figure 5

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 CQF Sub-Sub-TLV, the traditional path considering only link transmission delay will be calculated (i.e., degenerating into the calculation result similar as based on Min Unidirectional Link Delay metric type), otherwise, the path will consider both node delay and link delay.

7.1.2. OSPF Advertisement of Flex-algo Binding CQF

To be defined in next version.

7.2. Bind Deadline Scheduling parameters with Flex-Algo

The binding relationship <algorithm, scheduling delay, scheduling policy> 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 scheduling delay and scheduling policy contained in the FAD with the highest priority.

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

7.2.1. ISIS Advertisement of Flex-algo Binding Deadline

The Flexible Algorithm definition can specify the binding deadline scheduling delay and scheduling policy 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 Deadline Sub-Sub-TLV, which is advertised within IS-IS Flexible Algorithm Definition Sub-TLV. At most only one FAD Binding Deadline Sub-Sub-TLV can be included.

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

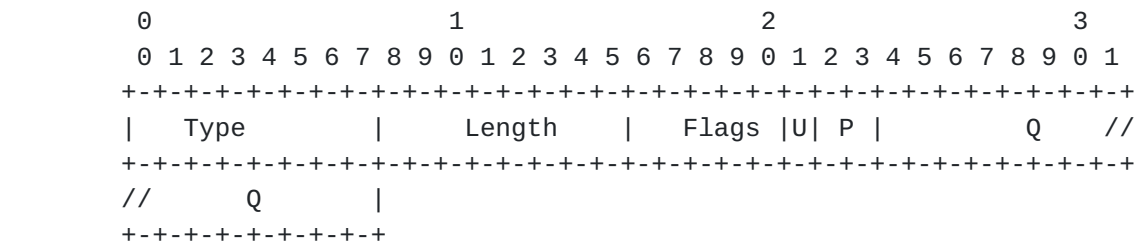

Figure 6

where:

Type: TBD

Length: 3

Flags: Two flags are currently defined.

U-flag: 1 bit, indicating whether the value of scheduling delay Q is known or unknown. 0 indicates known and 1 indicates unknown.

P-flag: 2 bits, indicating scheduling policy. The value can be: 0, not defined yet; 1, indicates the in-time policy; 2, indicates the on-time policy; 3, not defined yet.

Q: Indicates the deadline scheduling delay Q bound by flex algorithm, in units of microseconds. Note that if the U-flag is 1, the value of Q must be ignored and set to 0.

The binding deadline parameter contained in the FAD with the highest priority will take effect. If the FAD with the highest priority does not contain the FAD Binding Deadline Sub-Sub-TLV, the traditional path considering only link transmission delay will be calculated (i.e., degenerating into the calculation result similar as based on Min Unidirectional Link Delay metric type), otherwise, the path will consider both node delay and link delay.

Note that the FAD Binding CQF Sub-Sub-TLV and the FAD Binding Deadline Sub-Sub-TLV MUST not appear in FAD at the same time, otherwise, the first one is selected.

7.2.2. OSPF Advertisement of Flex-algo Binding Deadline

To be defined in next version.

7.3. CQF based Deterministic Routes Computation

This document use the new Metric-Type, Deterministic Delay, combined with the FAD Binding CQF Sub-Sub-TLV, 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.

For a Flex-algo plane that bound to a specific CQF 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 intra node as described in Section 4.1; 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. If the number of intermediate nodes included in the two candidate paths is different, the node delay may be different. For example, a traditional optimal low latency path only considering the link transmission delay may contain more hops, resulting in not being recognized as the optimal deterministic latency path.

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;

7.4. Deadline based Deterministic Routes Computation

This document use the new Metric-Type, Deterministic Delay, combined with the FAD Binding Deadline Sub-Sub-TLV, to compute deadline based shortest path with minimum deterministic end-to-end delay, which contains accumulated node delay provided by deadline and accumulated link transmission delay.

For a Flex-algo plane that bound to a specific deadline scheduling parameter, 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 scheduling delay, scheduling policy and forwarding delay intra node as described in Section 4.2; plus

Accumulated link transmission delay;

Assuming that the bound scheduling delay Q and scheduling policy P are obtained from the FAD Binding Deadline Sub-Sub-TLV (note that if the bound scheduling delay Q is an unknown value, the scheduling delay Q is temporarily replaced by 0 during path compuation), the node delay contributed by any intermediate node i in the candidate path is:

- For in-time policy, the node delay is in the range of [F(i), F(i)+Q], where F(i) represents the forwarding delay intra node i. Because the node delay value in this case is a range, and we need to get a specific value for SPF computation, thus there are several options to select a specific value as node delay, i.e., select F(i), or F(i)+Q, or the average of F(i) and F(i)+Q. This document take F(i)+Q as the default option.

- For on-time policy, the node delay is equal to F(i)+Q.

It should be noted that the above calculation process is used to select the optimal deterministic delay path from multiple candidate paths. However, once the deterministic delay SPF path is obtained, the deterministic delay metric of the deterministic delay SPF path should reflect the actual delay. Especially:

- When the bound scheduling delay Q is an unknown value, the deterministic delay metric of the deterministic delay SPF path is an formula containing variable quantity Q. In this case, the value of scheduling delay Q needs to be given through other methods, such as carried in the forwarded data packet. This means that the same path can provide different delays for different services.

- For in-time policy, the min delay of the SPF path is H*F, which is different with the max delay of SPF path is H*(F+Q), so that delay jitter is H*Q.

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

- Accumulated node delay jitter, which is H*Q for in-time policy and 0 for on-time policy; plus

- Accumulated link delay jitter, which is almost 0;

8. Routing Convergence and Redundance Considerations

As described in [I-D.ietf-lsr-flex-algo], Loop Free Alternate (LFA) paths for a given Flex-Algorithm MUST be computed using the same constraints as the calculation of the primary paths for that Flex-Algorithm. Within the Segment Routing framework, [I-D.ietf-rtgwg-segment-routing-ti-lfa] can provide TI-LFA path, as the expected post-convergence paths from the point of local repair, in any two connected network using a link-state IGP. However, ordinary IGP convergence and FRR protection may not meet the needs of deterministic services. The main reasons include:

- IGP convergence may cause considerable packet loss rate, even if FRR switching is implemented on the basis of rapid fault detection.

- The cumulative deterministic delay of the LFA path may be very different from that of the primary path, which does not meet the strict requirements for delay jitter.

Thus, according to Service Protection function defined in [RFC8655], packets can be spreaded over multiple disjoint forwarding paths to mitigate or eliminate the packet loss rate. In the context of Flex-algo, an additional redundant deterministic delay path different from FRR path need to be created, when if PLR enable Packet Replication Function (PRF) and the destination enable Packet Elimination Function (PEF). In this case, the data packets are sent along the primary deterministic delay SPF path and the redundant deterministic delay path at the same time, with almost the same cumulative delay.

The additional redundant deterministic delay path within the Flex-algo plane is often a traffic engineering path that is calculated by PLR based on the constraints contained in FAD and the following constraints:

- The number of nodes intersecting the primary and redundant deterministic delay paths shall be minimized;

- The difference between the number of hops of the primary and redundant deterministic delay paths shall be minimized;

- The difference between the cumulative link transmission delay of the primary and redundant deterministic delay paths shall be minimized.

Unlike LFA FRR path, more scheduling parameters read from link-state database can be attempted to used in the redundant deterministic delay path, to obtain the delay equal or close to the primary path.

Take deadline based path as an example, suppose that the number of hops in the primary deterministic delay SPF path is m, and the intermediate nodes passing through are A1, A2, ..., Am, the forwarding delay intra node for each hop is Fa, the scheduling delay intra node for each hop is Qa, and the cumulative link transmission delay is La, then the cumulative deterministic delay of the primary deterministic delay SPF path is the following formula:

o Delay(primary) = m*Fa + m*Qa + La

Similarly, suppose that the number of hops in the redundant deterministic delay path is n, and the intermediate nodes passing through are B1, B2, ..., Bn, the forwarding delay intra node for each hop is Fb, the scheduling delay intra node for each hop is Qb, and the cumulative link transmission delay is Lb, then the cumulative deterministic delay of the redundant deterministic delay path is the following formula:

o Delay(redundant) = n*Fb + n*Qb + Lb

The value of Delay(primary) can be calculated based on the known value of Qa that is bound to the flex-algo. Then, an appropriate Qb is slelected to make Delay(redundant) equal to Delay(primary). Qb, that is likely to be different from the bound value Qa, SHOULD be carried in the packets sent along the redundant deterministic delay path to get the expected latency.

If the value of Qa bound to the Flex-algo is unknown, states per service should be maintained at the ingress node, to determine the specific value of Qa according to SLA of the service sent along the primary deterministic delay SPF path. On this basis, Qb is then calculated. In this case, both Qa and Qb SHOULD be carried in the packets to get the expected latency.

If the further packet replication function continues to be implemented on an intermediate node of the network, the intermediate node only needs to regard itself as the head node of the new protection sub-domain, and can still adopt the above scheme. The intermediate node can also get the value of Delay(primary) based on the bound known Qa (or get from packets), On this basis, Qb is then calculated.

It should be noted that both packets sent along primary deterministic delay SPF path and redundant deterministic delay path within a flex-algo plane MUST use SIDs or prefix related with that algortihm.

The FIB entries within the flex-algo plane, such as SID entries, contain specific deterministic scheduling parameters to enable the

packet to excecute corresponding scheduling function. However, if the packet also carries scheduling parameters, the one in the packet must be preferred.

9. Examples of Deterministic delay SPF

As shown in Figure 7, the IGP flex-algo 128 plane contains five nodes, of which each link is a bidirectional link. The figure shows the transmission delay parameters of each lin, e.g, the transmission delay of the link between node R1 and node R2 is 10us.

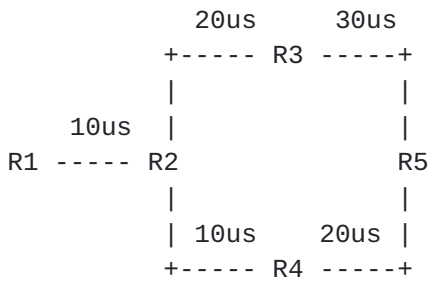

Figure 7

9.1. CQF Based Deterministic Delay SPF Path Example

It is assumed that the links of all nodes in the network are configured with consistent CQF scheduling parameters and have consistent node delay and delay jitter attributes, as follows:

Forwarding delay intra node = 0us

CQF enable/disable = ON

Supported cycle_size set = <10 us, 20 us>

Configure FAD of IGP flex-algo 128, set metric-type to Deterministic Delay, and set bound CQF scheduling parameters (cycle_size = 10us). Suppose that FAD is optimal after negotiation.

Taking node R1 as an example, it takes itself as the root to calculate the deterministic delay SPT as shown in Figure 8. In the figure, the sum of the node delay and the link transmssion delay is marked on each link. For example, the delay of link from node R2 to R3 is 10+20, where 10 is node delay and 20 is link transmission delay.

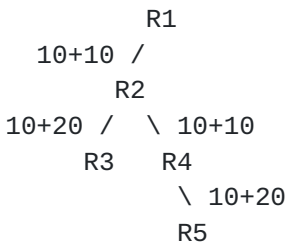

Figure 8

Therefore, with R1 as the source node and R5 as the destination node, the cumulative deterministic delay of CQF based SPF path (R1-R2-R4-R5) is 70us, The cumulative deterministic delay jitter is 20us.

Assuming that node R5 advertised SID-R5 that belongs to the flex-algo 128 plane, the following deterministic SPF FIB entry will be created on node R1.

KEY: SID-R5

Forwarding information:

next_hop = R2

interface = link(R1-R2)

metric_type = Deterministic Delay

scheduling algorithm = CQF with cycle_size 10 us

total_metric = 70 us

total_metric_variation = 20 us

9.2. Deadline Based Deterministic Delay SPF Path Example

It is assumed that the links of all nodes in the network are configured with consistent deadline scheduling parameters, as follows:

Forwarding delay intra node = 5us

Deadline enable/disable = ON

Supported scheduling delay set = <10us, 20us, 30us, 40us, 50us, 60us>, each item in the set support both in-time and on-time policy

Configure FAD of IGP flex-algo 128, set metric-type to Deterministic Delay, and set bound Deadline scheduling parameters (Q = 10us, with in-time policy). Suppose that FAD is optimal after negotiation.

Taking R1 node as an example, it takes itself as the root to calculate the deterministic delay SPT as shown in Figure 9. In the figure, the sum of the node delay and the link transmssion delay is marked on each link. Note that this document suggest to take F+Q as node delay during calculation even for in-time policy. For example, the delay of link from node R2 to R3 is 15+20, where 15 is node delay and 20 is link transmission delay.

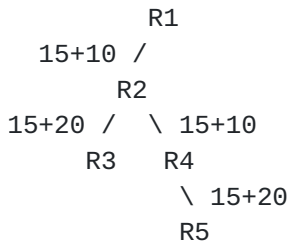

Figure 9

Therefore, with R1 as the source node and R5 as the destination node, the cumulative deterministic delay of Deadline based SPF path (R1-R2-R4-R5) is 85us, and the cumulative deterministic delay jitter is 30us.

Assuming that node R5 advertised SID-R5 that belongs to the fle-algo 128 plane, the following deterministic SPF FIB entry will be created on node R1.

KEY: SID-R5

Forwarding information:

next_hop = R2

interface = link(R1-R2)

metric_type = Deterministic Delay

scheduling algorithm = Deadline with Q=10 us with in-time policy

total_metric = 85 us

total_metric_variation = 30 us

Similarly, if on-time policy is bound to the flex-algo, the cumulative deterministic delay of Deadline based SPF path (R1-R2-R4-R5) is 85us, but the cumulative deterministic delay jitter is 0. The deterministic SPF FIB entry created on node R1 is changed to:

KEY: SID-R5

Forwarding information:

next_hop = R2

interface = link(R1-R2)

metric_type = Deterministic Delay

scheduling algorithm = Deadline with Q=10 us with on-time policy

total_metric = 85 us

total_metric_variation = 0

10. Use Cases

[RFC8578] described various deterministic routing use cases from multiple industries, including: Pro Audio and Video, Electrical Utilities, Building Automation Systems, Wireless for Industrial Applications, Cellular Radio, Industrial Machine to Machine (M2M), Mining Industry, Private Blockchain, Network Slicing, etc. Among them, some industries are now transitioning to packet based infrastructure, and some industries have already linked their different subsystems through networks (intra-domain or inter-domain). These industries have put forward the requirements of delay and delay jitter with different indicators, such as BAS requires low delay (10ms ~ 100ms) and low jitter (1ms); M2M requires that the underlying network infrastructure must ensure that the maximum end-to-end message delivery time is between 100 us and 50 ms; Mining industry requires predictable time delay to realize real-time monitoring. The deterministic paths can be centralized centralized computing, or distributed computing when there is a lack of controller.

The mechanism introduced in this document can get a SPF path with determinstic delay metric, but more importantly, with deterministic

dealy jitter. The determinsitic delay metric of the path actually depends on the network scale. It can be large or small, but it can be guaranteed to be the smallest of all candidate paths. The determinsitic delay jitter is also bounded and may be a cumulative value related to the number of hops or a value independent of the number of hops. SPF Paths with such characteristics will benefit multiple applications as mentioned above.

11. IANA Considerations

11.1. ISIS Deterministic Delay Metric Sub-TLV

This document registers the following Sub-TLV in the "Sub-TLVs for IS-IS Sub-TLVs for TLVs Advertising Neighbor Information" registry:

Type	Description	22	23	25	141	222	223
TBA1	Deterministic Delay Metric	y	y	y	y	y	y

11.2. Sub-Sub-TLVs for ISIS Deterministic Delay Metric Sub-TLV

This document registers the following Sub-TLV in the "Sub-TLVs for IS-IS Sub-TLVs for TLVs Advertising Neighbor Information" registry:

Type	Description	Reference
TBA2	CQF Scheduding Delay Intra Node	This document Section 4.1
TBA3	Deadline Scheduding Delay Intra Node	This document Section 4.2

11.3. IGP Metric-Type Registry

This document registers the following values in the "IGP Metric-Type Registry" for FAD:

Type: TBA4 (suggested 4)

Description: Deterministic Delay Metric as defined in this document

Reference: This document (Section 4)

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

This document defines the following Sub-Sub-TLVs in the "Sub-Sub-TLVs for Flexible Algorithm Definition Sub-TLV" registry:

Type	Description	Reference
TBA5	FAD Binding CQF	This document Section 7.1.1
TBA6	FAD Binding Deadline	This document Section 7.2.1

11.5. OSPF IANA considerations

TBD.

12. Security Considerations

TBD.

13. Acknowledgements

The authors would like to acknowledge the review and inputs from Peter Psenak, Bin Tan, Quan Xiong.

14. References

14.1. 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.ietf-rtgwg-segment-routing-ti-lfa]
Litkowski, S., Bashandy, A., Filsfils, C., Francois, P., Decraene, B., and D. Voyer, "Topology Independent Fast Reroute using Segment Routing", draft-ietf-rtgwg-segment-routing-ti-lfa-08 (work in progress), January 2022.

[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>.

[RFC2475] Blake, S., Black, D., Carlson, M., Davies, E., Wang, Z., and W. Weiss, "An Architecture for Differentiated Services", RFC 2475, DOI 10.17487/RFC2475, December 1998, <https://www.rfc-editor.org/info/rfc2475>.

[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>.

[RFC8578] Grossman, E., Ed., "Deterministic Networking Use Cases", RFC 8578, DOI 10.17487/RFC8578, May 2019, <https://www.rfc-editor.org/info/rfc8578>.

[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>.

14.2. Informative References

[CBS] "IEEE802.1Qav", 2009, <https://ieeexplore.ieee.org/document/8684664>.

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

[SP-LATENCY]
"Guaranteed Latency with SP", 2020,
<https://www.ieee802.org/1/files/public/docs2020/dd-grigorjew-strict-priority-latency-0320-v02.pdf>.

Authors' Addresses

Shaofu Peng
ZTE Corporation
China

Email: peng.shaofu@zte.com.cn

Tony Li
Juniper Networks
United States of America

Email: tony.li@tony.li