Network Working Group
Internet-Draft
Intended status: Standards Track
Expires: January 5, 2021

Z. Li
S. Peng
Huawei Technologies
C. Li
C. Xie
China Telecom
D. Voyer
Bell Canada
X. Li
Tsinghua University
P. Liu
China Mobile
C. Cao
China Unicom
K. Ebisawa
Toyota Motor Corporation

July 4, 2020

Application-aware IPv6 Networking (APN6) Encapsulation draft-li-6man-app-aware-ipv6-network-02

Abstract

Application-aware IPv6 Networking (APN6) framework makes use of IPv6 encapsulation in order to convey the application-aware information along with the data packet to the network so to facilitate the service deployment and SLA guarantee.

This document defines the encodings of the application characteristic information, for the APN6 framework, that can be exchanged between an application and the network infrastructure through the use of IPv6 extension headers.

Requirements Language

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC 2119 [RFC2119].

Status of This Memo

This Internet-Draft is submitted in full conformance with the provisions of BCP 78 and BCP 79.

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute

working documents as Internet-Drafts. The list of current Internet-Drafts is at 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 January 5, 2021.

Table of Contents

1. Introduction

A multitude of applications are carried over the network, which have varying needs for network bandwidth, latency, jitter, and packet loss, etc. Some applications such as online gaming and live video streaming have very demanding network requirements thereof require special treatments in the network. However, in current networks, the network and applications are decoupled, that is, the network is not aware of the applications' requirements in a finer granularity. Therefore, it is difficult to provide truly fine-granular traffic operations for the applications and guarantee their SLA requirements. Such guarantee would also be provided by adopting the hierarchical orchestration and the interactions between the orchestrator and multiple controllers, but it would take a very long time by going through the control and management elements. Moreover, the interfaces between those elements require standardizations.

This document proposes encapsulations for the Application-aware IPv6 Networking (APN6) framework, which makes use of IPv6 encapsulations (i.e. Hop-by-Hop Options Header (HBH), Destination Options Header (DOH), Segment Routing Header(SRH)) to convey the application-aware information including the application identifiers and their requirements along with the packet to the network to facilitate the service deployment and SLA guarantee. The application-aware options (i.e. Application-aware ID Option and Service-Para Option) are defined, which can be used in the IPv6 encapsulations, including the above listed different IPv6 extension headers, for this purpose.

2. Terminologies

APN: Application-aware Networking

APN6: Application-aware IPv6 Networking, i.e. the data plane of APN

is IPv6

DPI: Deep Packet Inspection

3. Demanding Applications

This section shows the various demanding requirements of some applications. The traffic of these applications needs to be differentiated from other types of traffic and applied with special treatments in the network, that is, the network needs to be able to provide fine-granular traffic operations and acceleration to these demanding applications.

3.1. Online Gaming

Good network performance is normally a prerequisite for satisfactory game play, especially for the online gaming. Shooting or racing online gaming is normally based on quick action and needs to update the game status in real time by continuously sending and receiving updates to/from the game server and/or other players. The online gaming is very sensitive to the network latency.

[I-D.zhang-apn-acceleration-usecase] describes the game acceleration scenarios using APN. In these scenarios, APN can identify the specific requirements of particular gaming applications, steer the flows to the game processors close to the users, and provide SLA guaranteed network services such as low latency and high reliability.

3.2. Video streaming

The network latency, jitter, bandwidth, and packet loss are the key factors for the video streaming. Live video streaming has even more strict requirements. High quality video source require more bandwidth in order to stream properly. Real time streaming services also require real time content delivery from the web server to the end user ideally via carefully planned explicit TE paths. The online gaming often involves live video streaming.

[I-D.liu-apn-edge-usecase] describes the various application scenarios in edge computing to which the APN can be beneficial, including augmented reality, cloud gaming and remote control, which empowers the video business, users interaction business and user-

device interaction business. In those scenarios, APN can identify the specific requirements of edge computing applications on the network, process close to the users, provide SLA guaranteed network services such as low latency and high reliability.

4. Problem Statement

A number of IETF activities that have been or are being conducted, e.g. DiffServ, primarily intend to evolve the IP architecture to support new service definitions which allow preferential or differentiated treatment to be applied to certain types of traffic. The challenge when using traditional ways to guarantee SLA is that the packets are not able to carry enough information to express differentiated service requirements of various applications. The network devices mainly rely on the 5-tuple of the packets which cannot accurately identify applications and provide fine-grained service treatments accordingly. If more information is needed, it has to refer to DPI which will introduce more cost in the network and impose security challenges.

In the era of SDN the orchestrator is introduced for the orchestration of applications and the network. The SDN controller can be aware of the service requirements of the applications on the network through the interface interworking with the orchestrator. The service requirements is used by the controller for traffic management. However, the method raises the following problems: 1) The whole loop is long and time-consuming which is not suitable for the real-time adjustment for applications; 2) Too many interfaces are involved in the loop which proposes more challenges of standardization and inter-operability, and it is difficult to be standardized for easy interworking.

5. APN6 Framework and Key Components

Application-aware Networking (APN) Framework is introduced in [I-D.li-apn-framework] in more details. When the data plane of APN is IPv6, it is APN6. Both frameworks share the same diagram, as shown in Figure 1.

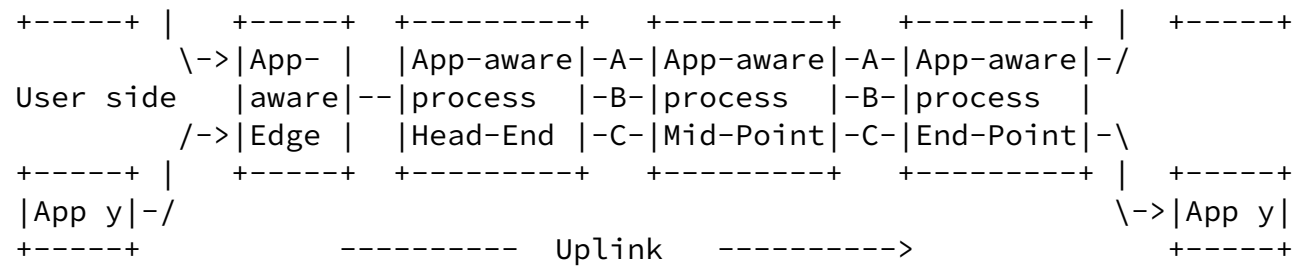

Figure 1 APN6 Framework and Key Components

The key components of the APN6 framework include Service-aware App, App-aware Edge Device, App-aware-process Head-End, App-aware-process Mid-Point, and App-aware-process End-Point, which are introduced as follows.

1. Service-aware App: The IPv6 enabled application that runs in the host obtains application characteristic information and encapsulate the packet with an IPv6 header that can, optionally, include an extension header with the application characteristic information. The application characteristic information (i.e. application-aware information) includes the following information:

- Application-aware identification information: identifying application, the user of application, i.e. the IPv6 packets as part of the traffic flow belonging to a specific SLA level/Application/User;

- Service requirements information: specifying at least one of the following parameters: bandwidth, delay, delay variation, packet loss ratio, security, etc.

If the application characteristic information is carried in the IPv6 packet, this information is used by the App-aware-process Head-End to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packet to its destination, that is, to steer the packet into a given policy which satisfies the application's requirements. If it is an SRv6 network and the SRv6 head-end receives the IPv6 packets carrying the application characteristic information, the SRv6 head-end will steer the traffic into an SRv6 policy/path that can satisfy its SLA requirements [I-D.ietf-spring-srv6-network-programming]. If the path cannot be found, the setup of a new path will be triggered.

In APN6, if the application characteristic information is directly added by the application and carried in the IPv6 packet sent by the host, it is called "Application-side Solution".

2. App-aware Edge Device: This network device receives packets from IPv6 enabled applications and obtains the application characteristic information. If the application is not Service-aware App, the application characteristic information can be obtained by packet inspection, derived from service information such as double VLAN tagging QinQ (C-VLAN and S-VLAN), or added according to the local policies, which is out of the scope of this document. The App-aware Edge Device adds the application characteristic information into the packet on behalf of the application. The packets carrying the application characteristic information will be sent to the App-aware-process Head-End, and the application characteristic information will be used to determine the path between the App-aware-process Head-End and the App-aware-process End-Point for forwarding the packets.

In APN6, if the application characteristic information is not directly added by the IPv6 enabled application but inferred at the App-aware Edge Device, it is called "Network-side Solution".

3. App-aware-process Head-End: This network device receives packets and obtains the application characteristic information. A set of paths, tunnels or SR/SRv6 policy, exist between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End maintains a mapping between the application characteristic information and the paths between the App-aware-process Head-End and the App-aware-process End-Point. The App-aware-process Head-End determines the path between the App-aware-process Head-End and the App-aware-process End-Point according to the application characteristic information carried in the packet and the

corresponding mapping, which satisfies the service requirements of the application. If there is no such mapping path found, the App-aware-process Head-End can set up a path towards the App-aware-process End-Point, and the mapping will be stored. The App-aware-process Head-End forwards the packets along the path. The application information conveyed by the IPv6 packet can also be copied into the outer IPv6 packet for further application-aware process.

4. App-aware-process Mid-Point: The Mid-Point provides the application-aware path service according to the path set up by the App-aware-process Head-End which satisfies the service requirements conveyed by the IPv6 packet. The Mid-Point may also adjust the resource locally in order to guarantee the service requirements depending on a specific policy and the application-aware information conveyed by the IPv6/SRv6 packet. Policy definitions and mechanisms are out of the scope of this document.

5. App-aware-process End-Point: The process of the specific service path will end at the End-Point. The service requirements information can be removed at the End-Point together with the outer IPv6 encapsulation or go on to be conveyed with the IPv6 packets if the Application-side Solution is used.

In this way, the network is aware of the applications and their requirements. According to the application characteristic information carried in the IPv6 packets the network is able to adjust its resources fast in order to satisfy the service requirement of applications. The flow-driven method also reduces the challenges of inter-operability and long control loop.

6. Application-aware Options

In order to support the Application-aware IPv6 networking, two application-aware options are defined:

- Application-aware ID Option
- Service-Para Option

6.1. Application-aware ID Option

The Application-aware ID option indicates the information of application, the user of application, and the application's SLA and service requirements, as illustrated in the following figure:

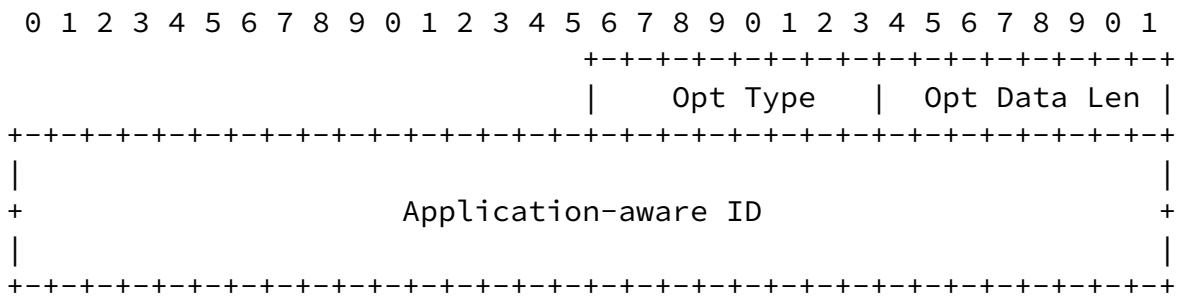

Figure 3. Application-aware ID Option

Opt Type: Type value is TBD1. 8-bit unsigned integer. Identifier of the type of this Application-aware ID Option.

Opt Data Len: 8-bit unsigned integer. Length of the Option Data field of this option, that is, length of the Application-aware ID, recommended to be 16 octets.

Option Data: Option-Type-specific data. It carries Application-aware ID.

The Application-aware ID has one of the following suggested structures:

-- Structure I: Any combination of SLA level (e.g. Gold, Silver, Bronze), APP ID, and/or user ID, and/or flow ID. The length of each field is variable, as shown in the following diagram:

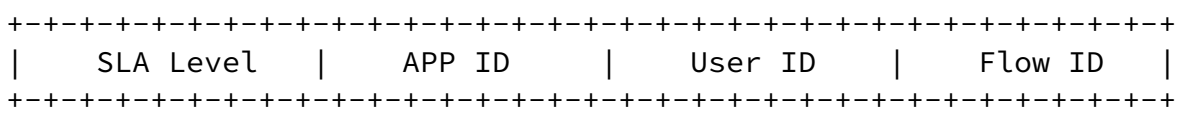

Figure 4. Application-aware ID Structure I

SLA Level: The level of SLA requirement of the application

APP ID: The identifier of the application

User ID: The user of the application

Flow ID: The particular flow of the application

-- Structure II: Any combination of SLA level (e.g. Gold, Silver, Bronze), APP ID, and/or user ID, and/or flow ID plus the arguments which indicating the service requirements (e.g. upper boundary of the latency: 10ms) of the identified application, as shown in the following diagram:

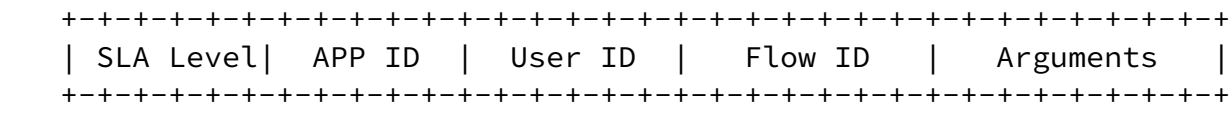

Figure 5. Application-aware ID Structure II

-- Structure III: An SRv6 SID, with its arguments as the information specified in Structure I or II, as shown in the following diagram:

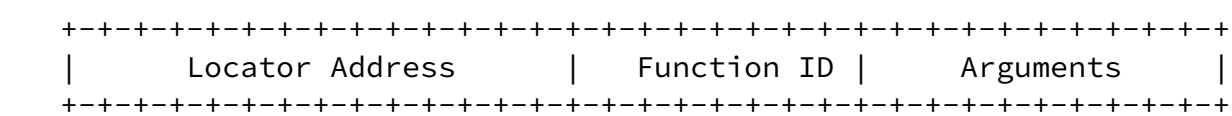

Figure 6. Application-aware ID Structure III

Section 7 introduces several locations that the Application-aware ID option can be encapsulated in the IPv6 packet, e.g., Hop-by-Hop Options Header and Destination Options Header, depending upon the scenarios and implementation requirements.

6.2. Service-Para Option

The Service-Para Option is a variable-length option carrying multiple service requirement parameters for a specific application. Each service requirement parameter is put into the corresponding Service-Para Sub-TLV, as shown in Figure 7. This Option can be put into the IPv6 Hop-by-Hop Options Header, Destination Options Header, and SRH TLV.

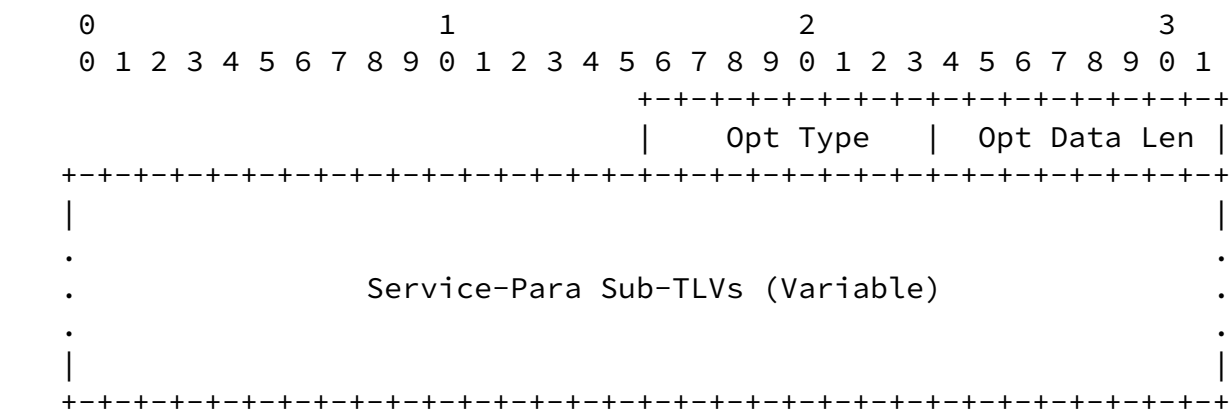

Figure 7. Service-Para Option

Opt Type: Type value is TBD2. 8-bit unsigned integer. Identifier of

the type of this Service-Para Option.

Opt Data Len: 8-bit unsigned integer. Length of the Option Data field of this option, that is, length of the Service-Para Sub-TLVs.

Option Data: Option-Type-specific data. It carries Service-Para Sub-TLVs. Variable-length field.

The corresponding Service-Para Sub-TLVs are shown in the following figures, respectively.

1. Bandwidth Sub-TLV

This Bandwidth sub-TLV indicates the bandwidth requirement of applications. The format of this sub-TLV is shown in the following diagram:

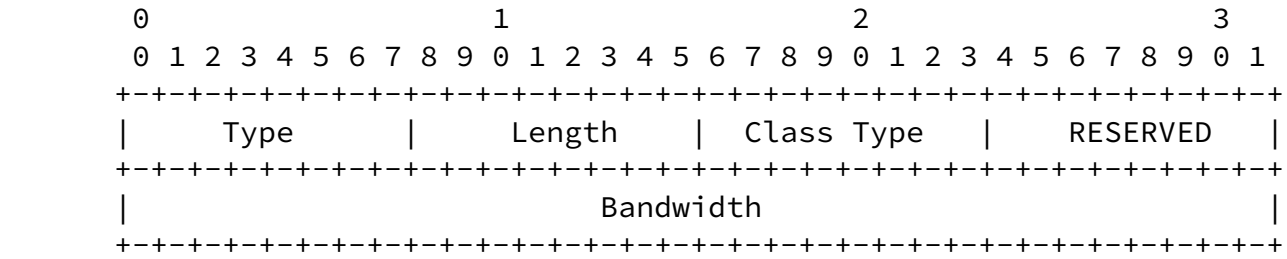

Figure 8. Bandwidth Sub-TLV

where:

Type: TBD3, the type of the Bandwidth Sub-TLV.

Length: 6 octets, the length of the data field of the Bandwidth Sub-TLV.

Class Type: The Bandwidth Type.

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

Bandwidth: This field carries the bandwidth requirement in Mbps along

the path.

2. Delay Sub-TLV

This Delay Sub-TLV indicates the delay requirement of applications. The format of this sub-TLV is shown in the following diagram:

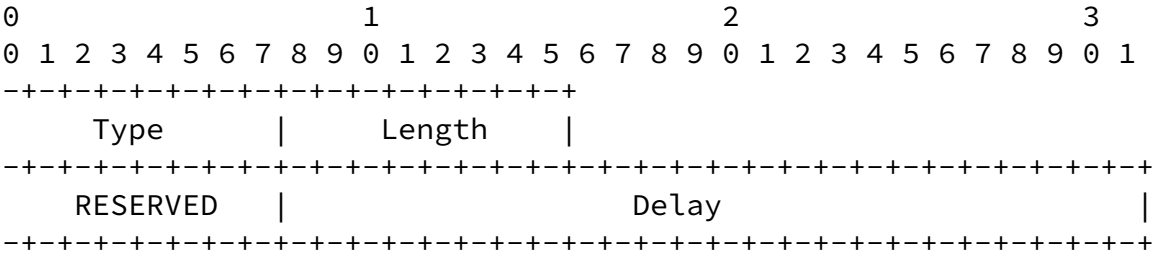

Figure 9. Delay Sub-TLV

where:

Type: TBD4, the type of the Delay Sub-TLV.

Length: 4 octets, the length of the data field of the Delay Sub-TLV.

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

Delay: This 24-bit field carries the delay requirements in microseconds, encoded as an integer value. When set to the maximum value 16,777,215 (16.777215 sec), then the delay is at least that value and may be larger. This value is the highest delay that can be tolerated.

3. Delay Variation Sub-TLV

This Delay Variation Sub-TLV indicates the delay variation requirement of applications. The format of this sub-TLV is shown in the following diagram:

0 1 2 3

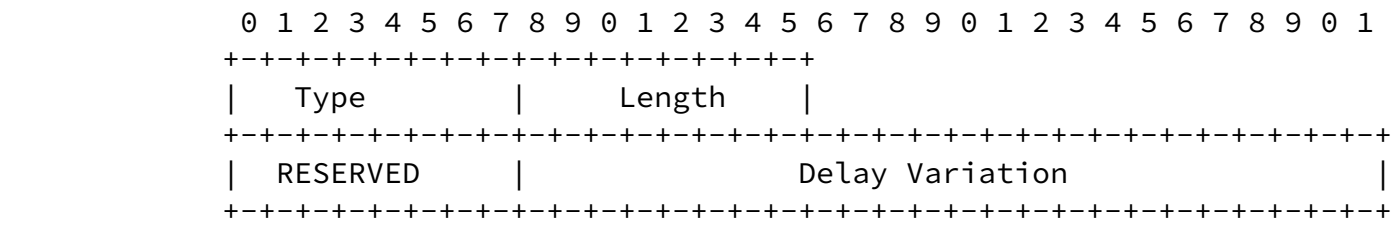

Figure 10. Delay Variation Sub-TLV

where:

Type: TBD5, the type of the Delay Variation Sub-TLV.

Length: 4 octets, the length of the data field of the Delay Variation Sub-TLV.

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

Delay Variation: This 24-bit field carries the delay variation requirements in microseconds, encoded as an integer value.

4. Packet Loss Ratio Sub-TLV

This Packet Loss Ratio Sub-TLV indicates the packet loss ratio requirement of applications. The format of this sub-TLV is shown in the following diagram:

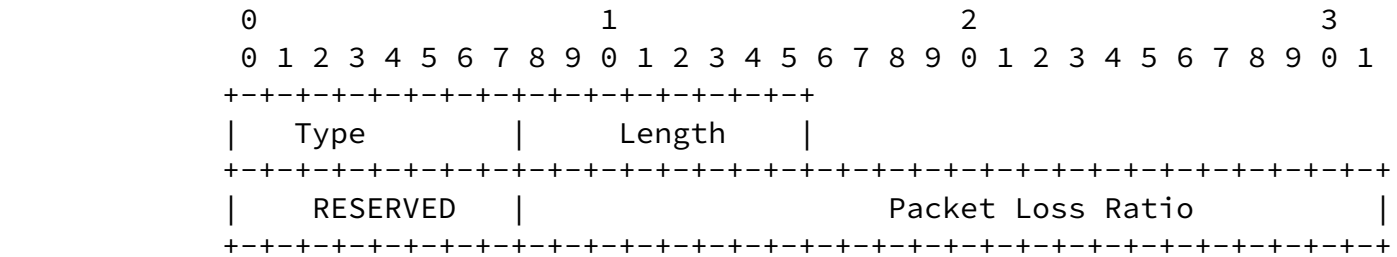

Figure 11. Packet Loss Ratio Sub-TLV

where:

Type: TBD6, the type of the Packet Loss Ratio Sub-TLV.

Length: 4 octets, the length of the data field of the Packet Loss Ratio Sub-TLV.

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

Packet Loss Ratio: This 24-bit field carries packet loss ratio requirement in packets per second. This value is the highest packet-loss ratio that can be tolerated.

7. Locations for placing the Application-aware Options

The Application-aware options can be placed in several locations in the IPv6 packet header depend upon the scenarios and implementation requirements.

7.1. Hop-by-Hop Options Header (HBH)

The application-aware options can be carried in the Hop-by-Hop Options Header as new options. By using the HBH Options Header, the information carried can be read by every node along the path. However, the current processing of the HBH Options Header goes to the

slow path, which will decrease the forwarding performance. A new enhanced HBH Options Header is proposed in [I-D.li-6man-hbh-fwd-hdr] in order to address the current limitations.

7.2. Destination Options Header (DOH)

The application-aware options can be carried in the Destination Options Header as new options.

7.3. Segment Routing Header (SRH)

The Application-aware options can be placed in the segment routing header (SRH), e.g., in the SRH TLV, the SID Arguments field, or the TAG field.

7.3.1. SRH TLV

The Application-aware options can be placed in the SRH TLV.

7.3.2. SID Arguments Field

The Application-aware ID option can be put in the SID Arguments field, which can be read by each node indicated by the SID in the SID list.

7.3.3. SRH TAG field

The Application-aware ID option can be put in the TAG field, which can be read by each node that processes the SRH.

8. IANA Considerations

IANA maintains the registry for the Options and Sub-TLVs.

Service-Para Option will require one new type code per sub-TLV defined in this document:

Type | Description

TBD1 | Application-aware ID Option

TBD2 | Service-Para Option

TBD3 | Bandwidth Sub-TLV

TBD4 | Delay Sub-TLV

TBD5 | Delay Variation Sub-TLV

TBD6 | Packet Loss Ratio Sub-TLV

9. Security Considerations

The Security Considerations described in [I-D.li-apn-problem-statement-usecases] can be referred to.

10. References

10.1. Normative References

[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>.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/info/rfc8200>.

10.2. Informative References

[I-D.ietf-spring-srv6-network-programming] Filsfils, C., Camarillo, P., Leddy, J., Voyer, D., Matsushima, S., and Z. Li, "SRv6 Network Programming", draft-ietf-spring-srv6-network-programming-15 (work in progress), March 2020.

[I-D.li-6man-hbh-fwd-hdr] Li, Z. and S. Peng, "Hop-by-Hop Forwarding Options Header", draft-li-6man-hbh-fwd-hdr-00 (work in progress), July 2020.

[I-D.li-apn-framework] Li, Z., Peng, S., Voyer, D., Li, C., Geng, L., Cao, C., Ebisawa, K., Previdi, S., and J. Guichard, "Application-aware Networking (APN) Framework", draft-li-apn-framework-00 (work in progress), March 2020.

[I-D.li-apn-problem-statement-usecases] Li, Z., Peng, S., Voyer, D., Xie, C., Liu, P., Qin, Z., Ebisawa, K., Previdi, S., and J. Guichard, "Problem Statement and Use Cases of Application-aware Networking (APN)", draft-li-apn-problem-statement-usecases-00 (work in progress), March 2020.

[I-D.liu-apn-edge-usecase] Liu, P., Geng, L., Peng, S., and Z. Li, "Use cases of Application-aware Networking (APN) in Edge Computing", draft-liu-apn-edge-usecase-00 (work in progress), March

2020.

[I-D.zhang-apn-acceleration-usecase]
Zhang, S., Cao, C., Peng, S., and Z. Li, "Use cases of Application-aware Networking (APN) in Game Acceleration", draft-zhang-apn-acceleration-usecase-00 (work in progress), June 2020.

[RFC3272] Awduche, D., Chiu, A., Elwalid, A., Widjaja, I., and X. Xiao, "Overview and Principles of Internet Traffic Engineering", RFC 3272, DOI 10.17487/RFC3272, May 2002, <https://www.rfc-editor.org/info/rfc3272>.

Authors' Addresses

Zhenbin Li
Huawei Technologies
Beijing 100095
China

Email: lizhenbin@huawei.com

Shuping Peng
Huawei Technologies
Beijing 100095
China

Email: pengshuping@huawei.com

Cong Li
China Telecom
Beijing 102209
China

Phone: +86-10-50902556
Email: licong@chinatelecom.cn

Chongfeng Xie
China Telecom
Beijing 102209
China

Phone: +86-10-50902116
Email: xiechf@chinatelecom.cn

Daniel Voyer
Bell Canada
Canada

Email: daniel.voyer@bell.ca

Xing Li
Tsinghua University
China

Email: xing@cernet.edu.cn

Peng Liu
China Mobile
China

Email: liupengyjy@chinamobile.com

Chang Cao
China Unicom
China

Email: liuc131@chinaunicom.cn

Kentaro Ebisawa
Toyota Motor Corporation
Japan

Email: ebisawa@toyota-tokyo.tech