SPRING
Internet-Draft
Intended status: Standards Track
Expires: 1 September 2026

Z. Li
Z. Du
China Mobile
W. Cheng
J. Wang
G. Zhang
Centec Networks
28 February 2026

Fine-grained QoE Enhancement using Semantic Tables
draft-li-spring-fine-grained-qoe-enhancement-00

Abstract

This document describes a fine-grained Quality of Experience (QoE) enhancement mechanism using semantic tables deployed at network forwarding nodes. The mechanism enables application-level SLA (Service Level Agreement) guarantees by carrying address indices and high-frequency-changing information in packets while maintaining low-frequency-changing semantic information at network nodes. This approach overcomes the limitations of traditional Application-aware Networking (APN) solutions, including excessive packet header overhead. The mechanism supports collaborative optimization across network, computing, and energy dimensions, and can be deployed over MPLS, IPv4/v6, SRv6, and other protocol data planes.

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 1 September 2026.

Copyright Notice

Table of Contents

1. Introduction

To provide better Quality of Experience (QoE) for users, networks need to offer fine-grained or even application-level Service Level Agreement (SLA) guarantees.

Current approaches have the following limitations:

SDN-based Centralized Approach:
: Uses orchestrators to perceive application requirements and arrange paths. This approach has long decision paths, making it unsuitable for latency-sensitive applications, and faces difficulties in interfacing between multiple systems.

Traditional Network Packets:
: Traditional network packets cannot carry sufficient information to indicate the diverse applications or services and their SLA requirements.

To address these issues, the industry proposed the Application-aware Networking (APN) mechanism [I-D.ietf-apn-framework]. APN supports inserting APN information (such as ID information and SLA information) into IPv6 packet extension headers. Network nodes, such as headend nodes, can parse this APN information and provide services on demand.

While APN provides a valuable framework, certain deployment scenarios may benefit from alternative approaches that address the following considerations:

1. Large Packet Header Modifications: Requires defining entirely new packet header formats.

2. High Overhead: Application/service, user, and network requirements must be carried per-packet.

3. Lack of Computing-Network Collaboration: Does not consider computing-related information and cannot perform computing-network collaborative optimization.

This document proposes a solution using semantic tables to enable fine-grained QoE enhancement while addressing these limitations.

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. Terminology

This document uses the following terminology:

Semantic Table:
A data structure deployed at network forwarding nodes containing user/service/application information and their SLA guarantee requirements.

Address Index (ADDR):
An identifier used to retrieve semantic tables at transit nodes, identifying specific application/user groups.

High-Frequency-Changing Information:
Dynamic information such as network, computing, and energy parameters that change frequently.

Low-Frequency-Changing Information:
Static configuration information that changes infrequently, such as user/application identifiers and bandwidth requirements.

QoE (Quality of Experience):
The degree of delight or annoyance of the user of an application or service, resulting from the fulfillment of expectations with respect to the utility and/or enjoyment of the application or service.

SLA (Service Level Agreement):
A commitment between a service provider and a client regarding aspects of the service such as quality, availability, and responsibilities.

3. Problem Statement

Current approaches for providing fine-grained QoE guarantees face the following core issues:

1. Packet Overhead: Each packet carries complete application/user information and SLA requirements, leading to excessive packet header overhead.

2. Security Issues: Sensitive user/application information is transmitted in plaintext across the network, posing privacy leak risks.

3. Lack of Flexibility: Cannot distinguish between high-frequency-changing and low-frequency-changing information; all information must be carried per-packet.

4. Computing-Network Separation: Existing solutions mainly focus on network resources and lack awareness and collaborative optimization capabilities for computing resources.

The design goals of this mechanism include:

* Reduce packet header overhead by carrying only necessary indices and high-frequency-changing information in packets

* Improve security by keeping sensitive information within network nodes

* Support collaborative optimization across network, computing, and energy dimensions

* Maintain compatibility with existing protocol data planes (MPLS, IPv4/v6, SRv6, etc.)

4. Solution Overview

This mechanism proposes a fine-grained QoE enhancement method based on semantic tables:

* Packets carry address indices and high-frequency-changing information (or only address indices)

* Specific semantics of low-frequency-changing information are maintained at forwarding nodes

* Packets passing through transit nodes trigger semantic table lookups using the address index to execute corresponding policies

4.1. Information Classification

This mechanism classifies QoE-related information into two categories:

Low-Frequency-Changing Information (deployed in semantic tables):

* User/service/application identification information

* Bandwidth requirements

* Delay tolerance

* Jitter tolerance

* Computing capacity requirements

* Other relatively stable SLA parameters

High-Frequency-Changing Information (carried in packets):

* DSCP value adjustments

* Queue priority

* Queue buffer depth

* Process priority

* Other dynamically changing parameters

4.2. Semantic Table Distribution Methods

This mechanism supports the following semantic table distribution methods:

Centralized:
Semantic actions for fine-grained SLA guarantees at transit nodes are distributed via the southbound interface of a centralized controller (such as an SDN controller).

Distributed:
Semantic actions for fine-grained SLA guarantees at transit nodes are advertised via distributed routing protocols (such as OSPF, BGP, etc.).

Hybrid:
Centralized distribution within domains and distributed advertisement between domains.

Manual Configuration:
Administrators directly configure semantic tables on each node (not recommended for large-scale deployments).

4.3. Semantic Table Content Acquisition

This mechanism does not restrict how semantic table content is acquired. Methods include:

* Active notification by users/applications/services through the northbound interface of controllers/orchestrators

* Active advertisement to network nodes via distributed routing protocols

* Passive detection by network edge nodes through DPI (Deep Packet Inspection) and subsequent advertisement to network nodes

5. Protocol Specification

5.1. Semantic Table Structure

Each node’s semantic table MUST contain the following mandatory fields:

Field Name	Length	Description
ADDR	32 bits	Address index carried in packets for identifying application/user groups
APP-Group-ID	Variable	Application group identification
USER-Group-ID	Variable	User group identification

Table 1: Mandatory Fields in Semantic Table

The semantic table MAY contain the following optional fields:

Field Name	Length	Description
Bandwidth	32 bits	Required bandwidth guarantee (in Kbps)
Delay	32 bits	Maximum tolerable delay for user/service (in microseconds)
Jitter	32 bits	Maximum tolerable jitter for user/service (in microseconds)
Computing-Capacity	32 bits	Minimum computing resources required by user/service
Priority	8 bits	Service priority level (0-255, higher is more important)

Table 2: Optional Fields in Semantic Table

5.2. Packet Format

The packet format defined in this mechanism uses a TLV (Type-Length-Value) structure for flexible extension:

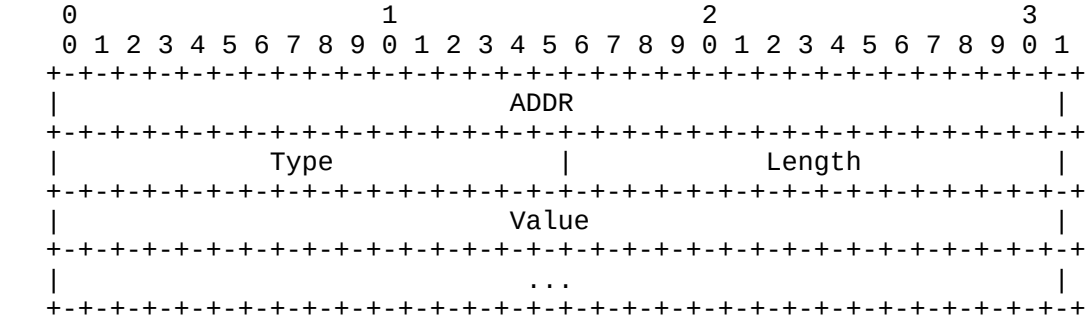

Figure 1: TLV Packet Format

Field Definitions:

ADDR (4 bytes):
Used to retrieve the semantic table at transit nodes, finding the application/user information that the TLV should act upon. A value of 0x00000000 is reserved and MUST NOT be used.

Type (2 bytes):
Indicates the type of high-frequency-changing service/resource that needs to be guaranteed for the application/user corresponding to ADDR, such as DSCP, queue priority, queue buffer depth, process priority, etc.

Length (2 bytes):
Indicates the length of the Value field in bytes.

Value (Length bytes):
Contains the specific value of the service/resource indicated by the Type field. The length is determined by the Length field.

5.3. TLV Type Definitions

This section defines the TLV Type field values. The Value field is dynamically specified based on the specific requirements of the user, service, or application.

Type (16bit)	Length (16bit)	Description
0x0000	-	Reserved
0x0001	0x0001	DSCP adjustment: Value specifies the target DSCP value (0-63) for the transmission path
0x0002	0x0001	Queue priority adjustment: Value specifies the target queue priority level at network node ports
0x0003	0x0004	Queue buffer depth: Value specifies the buffer depth in bytes
0x0004	0x0001	Process priority: Value specifies the computing process priority level
0x0005-0xFFFE	Variable	Reserved for future use
0xFFFF	0x0000	TLV terminator, payload follows

Table 3: TLV Type Definitions

The above Type values (0x0001-0x0004) are examples. Additional Type values can be defined based on deployment requirements and registered through IANA.

5.4. SRv6 Protocol Extension

The packet format can be carried over MPLS, IPv4/v6, SRv6, and other protocol data planes. This section describes the SRv6 protocol extension as an example.

SRv6 [RFC8754] is a source routing technology. The SRH extension header supports multiple SIDs, with each SID being 128 bits and containing Locator, Function, and Argument parts. The bit width of each part can be flexibly defined, providing good programmability.

In this mechanism:

* The ADDR in the packet occupies the Function and Argument parts of one SID, totaling 32 bits

* The remaining 96 bits are used for the Locator

* The TLV in the packet is carried via the Optional TLV variable field in the SRH extension header

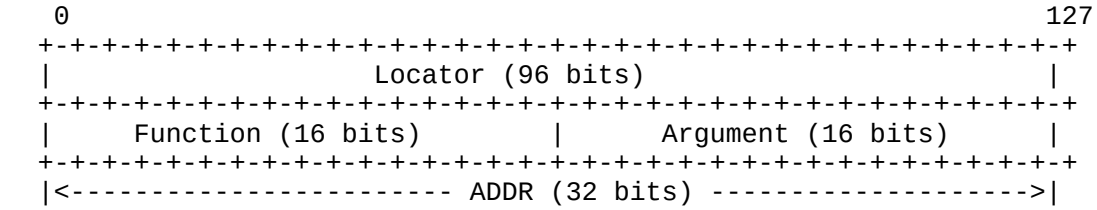

Figure 2: SRv6 SID Format with ADDR

6. Protocol Operations

6.1. Centralized Control Flow

Using a client-server pair with two intermediate network nodes and a centralized controller as an example:

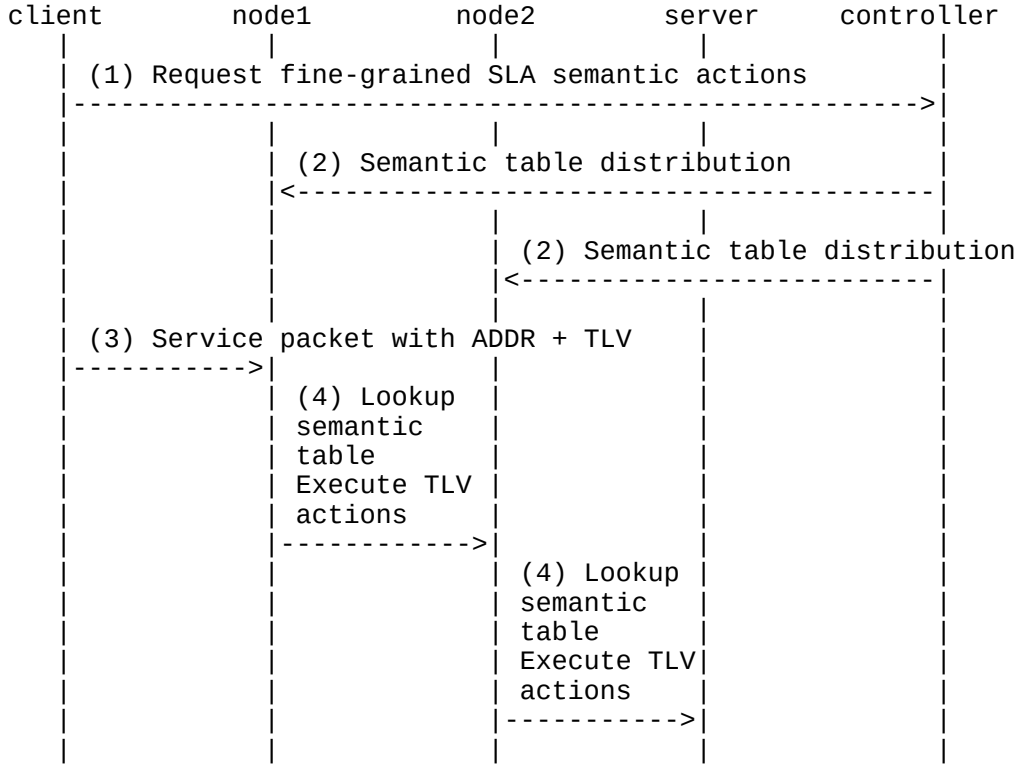

Figure 3: Centralized Control Flow

6.2. Detailed Operation Steps

1. Client Request Phase:

   The client sends a request to the controller, with the request type being "semantic action request for fine-grained or application-level SLA guarantee at transit nodes."

2. Semantic Table Distribution Phase:

   The controller sends semantic tables to each node.

3. Service Packet Transmission Phase:

   The client sends service packets carrying the mechanism header.

4. Node Processing Phase:

   Each node receives the packet, looks up the semantic table, and executes the actions indicated by the packet TLV.

6.3. Error Handling

Implementations MUST handle the following error conditions:

* Unknown ADDR: If a packet contains an ADDR that is not present in the local semantic table, the node SHOULD forward the packet using default QoS settings and MAY log the event.

* Invalid TLV: If a TLV with an unknown Type is encountered, the node SHOULD skip to the next TLV using the Length field and continue processing.

* Malformed Packet: If the packet structure is invalid (e.g., truncated TLV), the node SHOULD drop the packet and MAY increment an error counter.

7. Security Considerations

The semantic table mechanism introduces the following security considerations:

Semantic tables contain user and application identification information that may be sensitive. Unauthorized access to semantic table contents could reveal service topology and user behavior patterns. Implementations MUST enforce access control for semantic

table read and write operations. Semantic table distribution channels SHOULD be protected using authentication and encryption mechanisms.

The ADDR field carried in packets serves as an index into semantic tables. An attacker who can observe ADDR values may infer application or user group membership. When operating across trust domain boundaries, implementations SHOULD consider encrypting or obfuscating ADDR values.

Malicious injection of packets with crafted ADDR and TLV values could cause nodes to apply incorrect QoS policies. Implementations SHOULD validate that incoming packets originate from authorized sources before applying semantic table actions. BCP 38 ingress filtering SHOULD be applied at network boundaries.

8. IANA Considerations

This document requests IANA to create a new registry titled "Fine-grained QoE TLV Types" under an appropriate registry group.

The initial contents of the registry are defined in Section 5.3. The registration policy for new entries is Specification Required [RFC8126].

If the SRv6 extension defined in Section 5.4 is used, the SRv6 SID Function value used for semantic table lookup is allocated from the SRv6 Endpoint Behaviors registry defined in RFC 8986. This document does not request a specific allocation at this time; allocation will be requested when the mechanism is further specified.

9. References

9.1. 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>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/info/rfc8126>.

[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>.

[RFC8754] Filsfils, C., Ed., Dukes, D., Ed., Previdi, S., Leddy, J., Matsushima, S., and D. Voyer, "IPv6 Segment Routing Header (SRH)", RFC 8754, DOI 10.17487/RFC8754, March 2020, <https://www.rfc-editor.org/info/rfc8754>.

9.2. Informative References

[I-D.ietf-apn-framework] Liu, P., Peng, S., Li, Z., and C. Li, "Application-aware Networking (APN) Framework", Work in Progress, Internet-Draft, draft-ietf-apn-framework-10, October 2024, <https://datatracker.ietf.org/doc/html/draft-ietf-apn-framework>.

Acknowledgements

The authors would like to thank the members of the SPRING working group for their valuable feedback and discussions.

Contributors

The following individuals contributed to this document:

[Contributor Name]
[Organization]
Email: [email@example.com]

Authors' Addresses

Zhiqiang Li
China Mobile
32 Xuanwumen West Street
Beijing
100053
China
Email: lizhiqiangyjy@chinamobile.com

Zongpeng Du
China Mobile
32 Xuanwumen West Street
Beijing
100053
China
Email: duzongpeng@chinamobile.com

Wei Cheng
Centec Networks
Suzhou
215000
China
Email: chengw@centec.com

Junjie Wang
Centec Networks
Suzhou
215000
China
Email: wangjj@centec.com

Guoying Zhang
Centec Networks
Suzhou
215000
China
Email: zhanggy@centec.com