Standard Communication with Network Elements
Internet-Draft
Intended status: Standards Track
Expires: 29 May 2026

J. Zhang
G. Lv
Q. Wu
Z. Li
Chinese Academy of Sciences
25 November 2025

QoE-Driven Application-Transport Cooperation Requirements
draft-zhang-qoe-driven-transport-requirement-01

Abstract

This document specifies requirements for a QoE-driven transport system, which enables network stack to adjust its transport strategy according to the QoE intent expressed by applications. The Application Layer MUST provide a structured QoE Intent Signal detailing performance objectives to the transport layer. A QoE Mapping Engine is required to translate this intent into adaptive transport strategies. The Transport Protocol Stack SHOULD continuously feed a Transport Feedback Signal on current performance and network status back to the Application Layer, thereby closing the control loop essential for continuous QoE optimization.

Table of Contents

1. Introduction 2
1.1. Terminology 3
1.2. Requirements Language 3
2. System Architecture 3
2.1. Application Layer Requirements: Intent Declaration . . . 3
2.2. QoE Mapping Engine Requirements: Adaptive Strategy . . . 3
2.3. Transport Protocol Stack Requirements: Feedback and Optimization 4
3. Requirements for the QoE-Driven Cooperation Flow 5
3.1. Intent Flow (Application -> Transport System) 5
3.2. Strategy Flow (Mapping Engine -> Transport Protocol Stack) 5
3.3. Feedback Flow (Transport Protocol Stack -> Application Layer) 6
4. Security Considerations 6
5. Normative References 6
Authors' Addresses 7

1. Introduction

The rigid separation inherent in traditional networking models between the Application Layer and the Transport Layer frequently results in suboptimal Quality of Experience (QoE) for end-users. The Transport Layer's reliance on generic network metrics (e.g., RTT, throughput) is insufficient, as these metrics fail to correlate accurately with application-specific QoE contexts (e.g., video stall rate or guaranteed low-latency stability required for real-time gaming).

This structural deficiency is incapable of meeting the diverse and dynamic QoE demands of modern applications. A common conflict arises, for instance, when strategies focused on maximizing aggregate throughput actively undermine the requirements for stable low-latency. To fundamentally resolve this performance gap, the Transport Layer SHOULD evolve to become QoE-aware and cooperate directly with the Application Layer. This document specifies the requirements for a QoE-Driven Transport Paradigm, establishing a

closed-loop control system defined by three core functional flows essential for cross-layer cooperation: Intent Declaration, Adaptive Strategy, and Feedback Optimization. This architecture mandates a fundamental shift in responsibility to ensure continuous QoE management.

1.1. Terminology

QoE: Quality of Experience. QoE Intent Signal: The structured, encoded declaration of the application's real-time QoE goals and priorities. QoE Mapping Engine: The logical component responsible for translating the abstract QoE Intent Signal into concrete, executable Transport Strategy. Transport Strategy: The specific, low-level control commands generated by the Mapping Engine that modify the Transport Protocol Stack's behavior. Transport Feedback Signal: The signal containing real-time performance metrics (e.g., RTT, achieved throughput, loss rate) provided by the Transport Stack back to the Application Layer.

1.2. 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.

2. System Architecture

The QoE-Driven Transport Paradigm is based on a closed-loop system where responsibility is fundamentally shifted to achieve Application-Aware transport. This architecture is composed of three interconnected requirements for the system's components:

2.1. Application Layer Requirements: Intent Declaration

The Application Layer MUST provide a structured QoE Intent Signal to the transport system, explicitly declaring real-time performance objectives and shifting the optimization target from generic metrics (e.g., RTT) to user-centric QoE goals (e.g., stall avoidance).

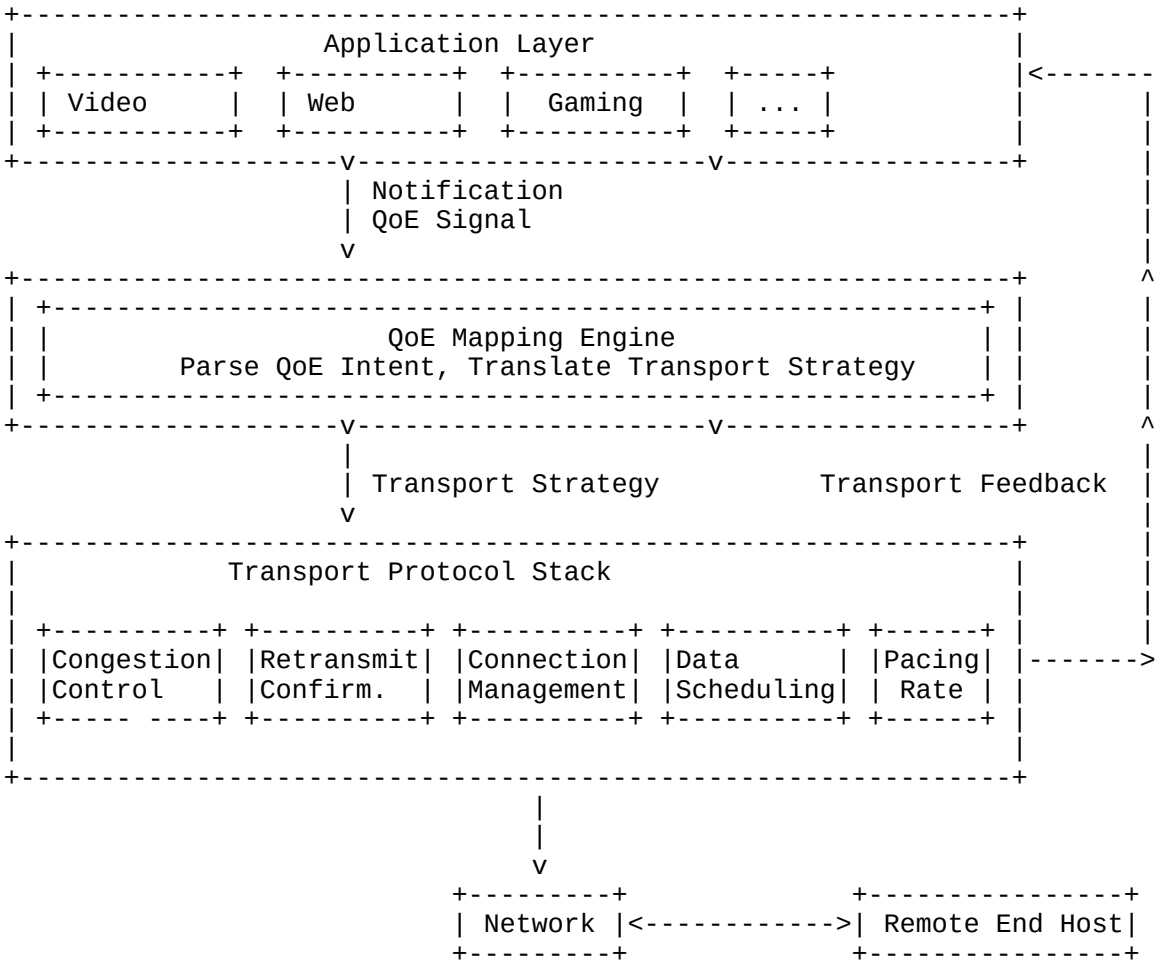
2.2. QoE Mapping Engine Requirements: Adaptive Strategy

A QoE Mapping Engine is fundamentally required. It MUST process the QoE Intent Signal and translate those high-level goals into concrete, adaptive transport strategies, including dynamic adjustments to congestion control algorithms, scheduling policies, and pacing rates.

2.3. Transport Protocol Stack Requirements: Feedback and Optimization

The Transport Protocol Stack SHOULD continuously furnish a Transport Feedback Signal on current performance and network status back to the Application Layer. This explicit linkage enables the Application Layer to accurately assess the effectiveness of the current transport strategy relative to the desired subjective QoE goal. This feedback closes the control loop, enabling the Application Layer to adjust its subsequent QoE Intent Signal, ensuring continuous QoE optimization.

The flow diagram below illustrates the relationship between these components:



3. Requirements for the QoE-Driven Cooperation Flow

This section details the specific requirements for the functional flows of the Application-Transport Cooperation system, mirroring the closed-loop diagram presented in Section 2.

3.1. Intent Flow (Application -> Transport System)

The Intent Flow carries the demand for QoE optimization from the Application Layer.

R3.1.1. QoE Signal Provision The Application Layer MUST generate and transmit the QoE Intent Signal. This signal SHALL explicitly declare the application’s real-time performance objectives, including objective metrics (e.g., maximum latency) and qualitative priorities (e.g., urgency weighting).

R3.1.2. Intent Processing and Validation The QoE Mapping Engine MUST receive and process the QoE Intent Signal. This processing SHALL include:

* Conflict Resolution: Utilizing priority fields to resolve competition among multiple concurrent intents.

* Security Sanitization: Enforcing constraints to ensure the requested intent does not violate network fairness principles or security policies.

3.2. Strategy Flow (Mapping Engine -> Transport Protocol Stack)

The Strategy Flow is where abstract QoE intent is converted into executable protocol actions.

R3.2.1. Strategy Translation The QoE Mapping Engine MUST translate the parsed QoE Intent into a set of specific Transport Strategies. This translation SHALL map high-level goals into the available low-level control knobs of the Transport Protocol Stack.

R3.2.2. Strategy Execution (Transport Protocol Stack) The Transport Protocol Stack MUST expose controllable components that accept and execute the Transport Strategies. Execution SHALL involve dynamic adjustments to the following functions based on the input from the Mapping Engine:

* R3.2.2.1. Congestion Control: Dynamically adjusting the active CC algorithm or modifying its parameters (e.g., aggression level) to Select Algorithm.

* R3.2.2.2. Retransmission Confirmation: Adjusting the retransmission policy to Set Mode, balancing recovery speed against latency tolerance.

* R3.2.2.3. Connection Management: Controlling parameters related to connection establishment, migration, or resource partitioning to Control Behavior.

* R3.2.2.4. Data Scheduling: Modifying the criteria used to sequence data packets or streams for transmission based on their relative priority, thus Tuning Policy.

* R3.2.2.5. Pacing: Adjusting the rate at which data is injected into the network to enforce a specific sending rate or smoothing behavior, fulfilling the requirement to Modify Rate.

3.3. Feedback Flow (Transport Protocol Stack -> Application Layer)

The Feedback Flow is essential for closing the control loop and enabling continuous adaptation.

R3.3.1. Performance Measurement The Transport Protocol Stack MUST continuously measure the real-time performance metrics of the current connection. This measurement SHALL include fundamental indicators such as RTT, packet loss rate, and achieved throughput.

R3.3.2. Feedback and Notification Provision The Transport Protocol Stack SHOULD provide this performance information as a Transport Feedback Signal back to the Application Layer. Additionally, the transport system SHOULD provide Notification when critical events occur (e.g., inability to meet the declared QoE Intent) or when a significant change in strategy is executed. This signal SHALL enable the Application Layer to assess the success of the current Transport Strategy relative to its declared QoE Intent.

4. Security Considerations

This document does not introduce any new security considerations.

5. 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/rfc/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/rfc/rfc8174>.

Authors' Addresses

Jiaxing Zhang
Chinese Academy of Sciences
Beijing
China
Email: zhangjiaxing20g@ict.ac.cn

Gerui Lv
Chinese Academy of Sciences
Beijing
China
Email: lvgerui@ict.ac.cn

Qinghua Wu
Chinese Academy of Sciences
Beijing
China
Email: wuqinghua@ict.ac.cn

Zhenyu Li
Chinese Academy of Sciences
Beijing
China
Email: zyli@ict.ac.cn