

Standard Communication with Network Elements  
Internet-Draft  
Intended status: Standards Track  
Expires: 23 October 2025

H. Shi  
X. Geng  
Q. Gao  
Huawei  
Q. Wu  
J. Zhang  
Chinese Academy of Sciences  
21 April 2025

SCONE Real Time Communication Requirement  
draft-shi-scone-rtc-requirement-02

Abstract

In real-time communication (RTC) applications, low latency is essential, but unstable network conditions make it challenging. Traditional control loop reacts slowly and inaccurately to network changes. A new approach is proposed, communicating bandwidth and queue information from the bottleneck to the end-host for more accurate control.

Discussion Venues

This note is to be removed before publishing as an RFC.

Discussion of this document takes place on the Standard Communication with Network Elements Working Group mailing list ([scone@ietf.org](mailto:scone@ietf.org)), which is archived at <https://mailarchive.ietf.org/arch/browse/scone>.

Source for this draft and an issue tracker can be found at <https://github.com/VMatrix1900/draft-scone-rtc-requirement>.

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 23 October 2025.

## Copyright Notice

Copyright (c) 2025 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to BCP 78 and the IETF Trust's Legal Provisions Relating to IETF Documents (<https://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Revised BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Revised BSD License.

## Table of Contents

1. Introduction . . . . .	2
1.1. Terminology . . . . .	3
1.2. Requirements Language . . . . .	3
2. Network Assisted Real-Time Communication Optimization . . . . .	3
3. Requirement of the feedback signal . . . . .	4
4. Security Considerations . . . . .	5
5. Normative References . . . . .	5
Authors' Addresses . . . . .	5

## 1. Introduction

In real-time communication (RTC) transmission scenarios, low latency is one of the key requirements, especially in real-time interactive applications such as video conferencing, live streaming, and cloud gaming. For these applications, any significant delay can severely impact the user's quality of experience. However, the network environment especially in mobile networks is often changing dynamically, and factors such as bandwidth fluctuations, and delay jitters can affect transmission performance. Therefore, addressing these network fluctuations to ensure RTC with low latency is a significant technical challenge.

Traditional RTC transmission protocols typically rely on end-to-end network condition detection methods, such as ACKs or NACKs that adjust sending rates based on packet loss and latency changes. The Cubic, Reno and BBR are typical congestion control algorithms (CCAs) that adjusts transmission rates by estimating network congestion. However, end-to-end feedback mechanisms like them have several limitations:

- \* High feedback latency: Traditional end-to-end feedback mechanisms rely on detecting and adjusting to network conditions. For example, CCAs only react after detecting congestion or latency changes. This mechanism cannot predict sudden changes in network conditions, leading to delayed adjustments and reduced transmission efficiency.
- \* Insufficient feedback accuracy: Feedback mechanisms based on packet loss or latency measurements only provide rough estimates of network conditions. Due to varying network environments' sensitivity to packet loss and latency, relying solely on these indicators makes precise adaptation difficult, often resulting in inaccurate adjustment.

To overcome these limitations, Explicit Network Feedback mechanisms like ECN have been proposed. Unlike implicit end-to-end feedback, explicit network feedback obtains real-time status information from network devices (such as routers and switches), providing faster and more accurate feedback on network conditions. However, the limited number of bits in ECN feedback results in low precision, making it difficult to meet the feedback accuracy requirements of RTC applications. Therefore, this document proposes an explicit network feedback mechanism, using bandwidth and queue information to assist the end-host in fine-grained control and reduce RTC latency. Detailed solution is out of scope of this document.

### 1.1. Terminology

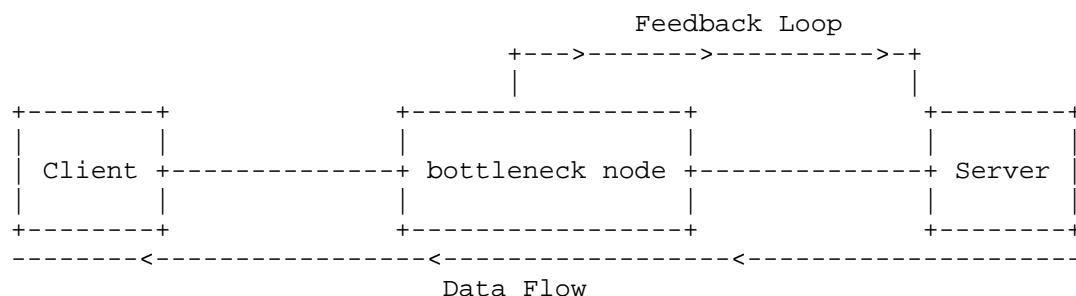
RTC: real-time communication.

### 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. Network Assisted Real-Time Communication Optimization

The process is shown in the diagram below: the end-to-end RTC stream is sent from the server to the client. The bottleneck node provides real-time feedback of stream-level Capacity and Queue Length to the sender, assisting the sender in rate control.



**Feedback Mechanism:** The bottleneck node sends network status information, including current network capacity and queue length, to the sender at fixed intervals (e.g., every 100ms). The feedback is transmitted using a lightweight way (such as embedding in IP header) to ensure low overhead and high timeliness.

**Rate Control Strategy:** The sender adjusts the transmission rate based on the network capacity feedback from the bottleneck node. If capacity is sufficient, the sender increases the transmission rate; if capacity is limited or the queue length increases, the sender reduces the rate to avoid network congestion. Additionally, the bottleneck node can prioritize scheduling key video frames to ensure smooth playback.

With explicit feedback from the bottleneck node, the sender can respond to network changes more quickly, reducing transmission delays caused by congestion. Compared to traditional end-to-end implicit feedback, explicit feedback is more accurate and better ensures the continuity and quality of the video stream.

### 3. Requirement of the feedback signal

To ensure that signaling is transmitted alongside the data flow, this scheme employs in-band signaling, where feedback is embedded in the headers of returning ACK packets.

The feedback contents includes:

- \* **Network Capacity:** The bottleneck node monitors the currently available network bandwidth in real-time and feeds back the available bandwidth capacity to the sender. The sender adjusts the RTC rate control (including sending rate and encoding bitrate) based on this information to maximize bandwidth utilization while avoiding network congestion.

- \* Queue Length: The bottleneck node also monitors the length of its internal buffer queue. When the queue length increases, it indicates growing transmission delays and potential congestion. Based on this information, the sender can timely reduce the transmission rate to prevent packet loss.

#### 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

Hang Shi  
Huawei  
Beijing  
China  
Email: shihang9@huawei.com

Xuesong Geng  
Huawei  
Beijing  
China  
Email: gengxuesong@huawei.com

Qiangzhou Gao  
Huawei  
Beijing  
China  
Email: gaoqiangzhou@huawei.com

Qinghua Wu  
Chinese Academy of Sciences  
Beijing  
China

Email: wuqinghua@ict.ac.cn

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