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

Y. Liu
China Mobile
J. Yao
Huawei
C. Lin
New H3C Technologies
M. Xiao
ZTE

28 February 2026

Congestion Control Based on SRv6 Path
draft-liu-rtgwg-srv6-cc-01

Abstract

This document describes a congestion control solution based on SRv6. It defines mechanisms for congestion notification and flow control within an SRv6-based network, optimizing congestion handling through hierarchical congestion control messages along SRv6 paths.

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

The SRv6 network needs a reliable and efficient mechanism for handling congestion across different segments. Current congestion control techniques lack the ability to handle congestion in a fine-grained, per-path manner. This draft proposes a solution that uses SRv6 path segments and slicing to notify upstream nodes and take actions to reduce congestion. The key idea is to notify upstream nodes about congestion and enable flow control based on SRv6 segments (SID lists). This process is integrated with the SRv6 network's slicing capabilities to provide fine-grained control over network traffic, ensuring lossless transmission of data across SRv6 network.

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. Background and Motivation

Priority Flow Control (PFC) provides hop-by-hop, priority-based traffic control. Compared to the traditional Ethernet Pause mechanism, PFC enables more precise flow management by creating multiple virtual channels on a link, each of which can be paused or resumed independently, ensuring that traffic of different priorities does not interfere with one another.

With the growth of intelligent computing services, scenarios such as disaggregated computing and real-time inference require the lossless transmission of large volumes of bursty traffic. In interconnected wide-area networks (WANs), when network congestion occurs, the congestion status must be quickly propagated upstream to both head-end devices and edge devices, enabling hop-by-hop reduction of sending rates. These intelligent computing WANs typically use SRv6 Policies for transport. However, once traffic enters a policy, traditional PFC mechanisms face the following three major challenges:

1. Imprecise Congestion Notification: PFC propagates congestion information via Ethernet multicast frames. In WANs with complex topologies, multicast-based congestion signaling cannot accurately reach upstream SRv6 nodes, potentially leading to incorrect flow suppression and impacting unrelated services.

2. Long Path Latency: WAN paths are long and have significant latency. If congestion signals must be sent all the way back to the data center or edge devices, it results in prolonged traffic degradation. Therefore, upstream control on the SRv6 path is needed to respond promptly.

3. Control Overhead at the Head Node: A single head node in the WAN may manage numerous SRv6 paths. If all congestion messages are sent back to the head node, it could become a processing bottleneck. Performing distributed traffic control at intermediate nodes along the SRv6 path can alleviate the burden on the head node.

3. SRv6 congestion notification Mechanism

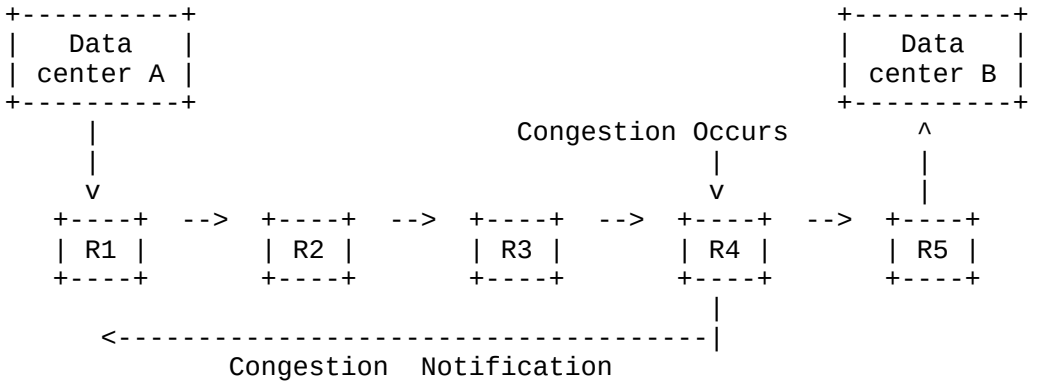

Figure 1: Congestion Notification in SRv6 Network

Consider two data centers, A and B, connected via an SRv6 path defined as R1 -> R2 -> R3 -> R4 -> R5, as shown in Figure 1. The process follows these steps:

1. The head node R1 encapsulates the SID list (SRv6 path) containing R2 -> R3 -> R4 -> R5. It may optionally carry an SRv6 path segment (PSID) and starts forwarding the data. The source address is R1, and the destination address is the SID of R2.

2. Transit nodes (R2, R3, R4) forward data according to the SID list, with each node checking its local SID table for forwarding and slice-related information.

3. When a node, such as R4, faces congestion(such as queue overload), it sends a congestion notification message to the previous node in the SID list (R3), including congestion-related information. For example, the priority queue where congestion occurs, congestion control parameter information (such as pause-time and/or target bandwidth), and slice ID of the suppressed tenant.

4. R3 receives the notification and adjusts the forwarding rate based on local capacity. If R3 cannot handle the congestion, the notification is forwarded further upstream to R2 and so on.

5. If no node can manage the congestion, the head node R1 adjusts the path load balancing or selects an alternate path to mitigate the congestion.

4. Congestion Notification Message Format

The congestion notification message can be encapsulated in either ICMPv6 [RFC4443] or UDP [RFC768] messages. Regardless of the encapsulation format, they contains following fields:

1. *Checksum*: Used for error-checking the packet.

2. *Flags*: Contains special flags. not defined.

3. *Priority*: Queue priority identifier, each priority queue occupies 1 bit (from high-order to low-order bits representing high priority to low priority respectively). If each bit is set to 1, it indicates that the priority queue is suppressed due to congestion control. If each bit is set to 0, it indicates that suppression is released from the priority queue.

4. *Argument[]*: Congestion control parameter information, each priority occupies 2 bytes, totaling 16 bytes. The use of arguments can be combined with flags, supporting flexible definition of congestion control parameter fields. By default(when all flag bits are 0), the meaning of argument is pause-time, measured in microseconds. When the upstream node's action execution exceeds the value of pausetime, traffic must be restored immediately.

5. *Target Bandwidth*: Indicates the target bandwidth information for expectation suppression. The default value is 0.

6. *Slice ID*: The identifier for the slice experiencing congestion.

4.1. ICMPv6 message format

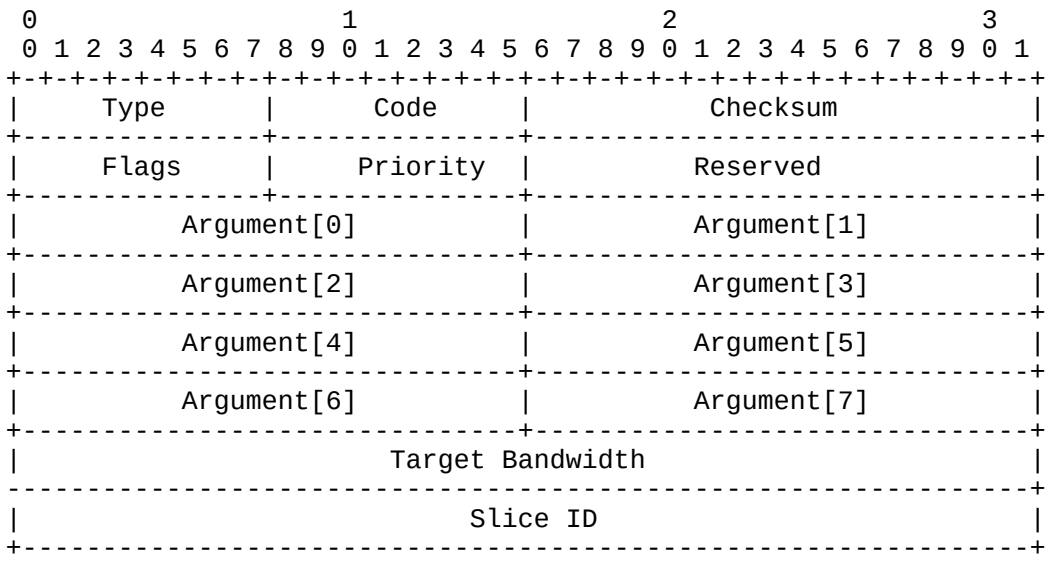

Figure 2: Congestion Notification in ICMPv6

Where:

Type and *Code*: These fields indicate the specific congestion notification type and its sub-type, providing details about the kind of congestion event being reported.

4.2. UDP packet

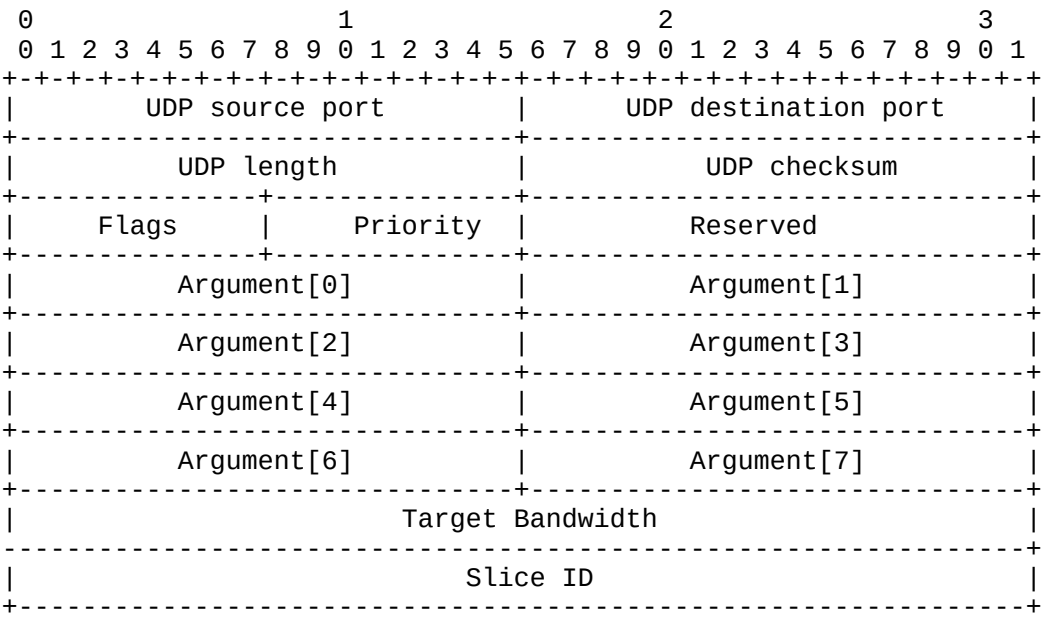

Figure 3: Congestion Notification in UDP

Where:

UDP Destination port: A new port indicates the congestion notification packet.

5. SRv6 congestion notification running process

The SID configuration of each node in the figure is as follows: End.X SIDs of nodes R1 to R5 are A::1:1,A::2:1,A::3:1,A::4:1,A::5:1, and the slice ID corresponding to each SID is 1. The VPN SID of the R5 node is A::5:F.

The running process of each node is as follows:

1. The data packet sent by R1 is encapsulated with the SRv6 Policy. The SID list is {A::2:1, A::3:1, A::4:1, A::5:F}. The source address is A::1::, and the destination address is A::2:1.

2. Packets are forwarded based on the SID list of the SRv6 path. The destination address of the packet forwarded by R4 is replaced with A::5:F. The forwarding plane selects the corresponding slice based on the slice ID carried in the packet and selects the priority queue to be used based on the service class of the packet. When the 6th priority queue corresponding to slice 1 on

the outbound interface of R4 is congested. current buffer usage exceeds the preset threshold (50%), a congestion notification message needs to be sent to the previous hop of the SRv6 path.

3. R4 constructs a congestion control packet in ICMPv6/UDP format and sends it to the previous-hop node. The packet carries the slice ID that identifies the tenant, the priority queue where congestion occurs, and the parameters that the tenant is expected to use for traffic control, such as pause-time. The source address of the IP packet can be the local address of R4, and the destination address can be the address of the previous-hop node R3, so that the data packet can reach R3 through routing. In this example, the slice ID is 1, the priority queue is 6, and the arguments parameter is used to identify the pause-time, which is 5 ms. This indicates that the upstream node R3 is expected to stop sending packets for 5 ms.

4. After receiving the congestion notification packet, R3 checks whether the destination address is local. R3 parses the ICMPv6/ UDP congestion control packet to obtain the tenant slice ID and the priority queue that identifies the congestion, and then performs traffic control on the priority queue of the tenant based on the traffic control parameter information carried in the packet. In this example, R3 performs traffic control on priority queue 6 of slice 1 for a duration of 5 ms.

5. If the local buffer of R3 is sufficient, the congestion on R4 is relieved at the previous hop R3. This method effectively alleviates small network bursts through congestion control packets, thereby avoiding packet loss due to congestion. When the local buffer of the priority queue of the tenant on R3 is insufficient (the usage exceeds the threshold), R3 constructs an ICMPv6/UDP congestion control packet (for details about how to construct congestion control parameters, see R4) to notify the previous-hop node R2. This process is repeated until the congestion status is transmitted to the ingress node of the tunnel through each hop. The ingress node then resolves the congestion status through multipath load balancing or selects an alternate path.

6. Security Considerations

This document does not introduce any new security considerations.

7. IANA Considerations

This document requests IANA to allocate a new ICMP message type and UDP port.

8. Normative References

[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/rfc/rfc8754>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/rfc/rfc4443>.

[RFC768] Postel, J., "User Datagram Protocol", STD 6, RFC 768, DOI 10.17487/RFC0768, August 1980, <https://www.rfc-editor.org/rfc/rfc768>.

[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

Yisong Liu
China Mobile
Beijing
China
Email: liuyisong@chinamobile.com

Junda Yao
Huawei
Beijing
China
Email: yaojunda@huawei.com

Changwang Lin
New H3C Technologies
Beijing
China
Email: linchangwang.04414@h3c.com

Min Xiao
ZTE Corporation
Nanjing
China
Email: xiao.min2@zte.com.cn