IPv6 Maintenance
Internet-Draft
Intended status: Informational
Expires: 26 November 2026

Jiayuan. Hu, Ed.
Xia. Gong, Ed.
China Telecom

25 May 2026

A RoCEv2 Flow-Level Load Balancing Method Based on the IPv6 Flow Label
draft-hu-6man-ipv6-flowlabel-load-balancing-rdma-00

Abstract

This document proposes a method for achieving flow-level load balancing in RoCEv2 (RDMA over Converged Ethernet version 2) networks. Traditional per-flow load balancing based on the 5-tuple cannot distinguish between different RDMA sessions that share the same 5-tuple. This causes "elephant flows" to be hashed to the same path, leading to network congestion. This method resolves this issue by parsing the QP (Queue Pair) information from the IB BTH (Base Transport Header) and IB DETH (Datagram Extended Transport Header) headers of the RoCEv2 packet. By combining this with portions of the IPv6 source and destination addresses as an entropy source, a CRC32 hash algorithm generates a 20-bit value, which is then written into the Flow Label field of the IPv6 header. Network devices can subsequently use the updated "5-tuple + Flow Label" for more granular flow-level load balancing, thereby effectively improving transmission efficiency in high-performance networks such as AI computing.

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 26 November 2026.

Copyright Notice

Copyright (c) 2026 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

The rapid advancement of Artificial Intelligence (AI) and High-Performance Computing (HPC) has driven the widespread adoption of Remote Direct Memory Access (RDMA) over Converged Ethernet (RoCEv2) in data center and intelligent computing networks. RoCEv2 enables high-throughput, low-latency data transfers that are critical for distributed training and storage workloads. However, the effective operation of these networks is challenged by the inherent characteristics of RDMA traffic, particularly the "elephant flow" problem.

Traditional load balancing mechanisms in IP networks typically rely on a 5-tuple (source/destination IP address, source/destination port, and protocol number) to identify and distribute traffic flows. In RoCEv2 networks, a significant limitation arises: multiple distinct RDMA sessions or flows generated by the same upper-layer application may share an identical 5-tuple. This is because the RDMA Queue Pair (QP) information, which uniquely identifies a session, is encapsulated within the InfiniBand Base Transport Header (IB BTH) and

Datagram Extended Transport Header (IB DETH) of the RoCEv2 packet. Consequently, conventional 5-tuple-based hashing treats these distinct RDMA flows as a single entity and forwards them to the same network path, leading to severe congestion, packet loss, and a significant degradation in overall network throughput and performance.

To address this problem, this document introduces a novel method for flow-level load balancing that leverages a standard IPv6 extension mechanism. The core idea is to enable network devices, such as routers and switches, to extract the QP pair information (source QP and destination QP) from the RoCEv2 packets. This extracted QP pair information is then used as input to a CRC32-based hash function to generate a unique per-flow identifier. This identifier is subsequently mapped into the Flow Label field of the IPv6 header.

By combining the traditional 5-tuple with this dynamically generated Flow Label, the proposed method creates a fine-grained "5-tuple + Flow Label" flow identification key. This allows network devices to effectively distinguish between different RDMA sessions that were previously indistinguishable, thereby achieving true flow-level load balancing. This approach minimizes path collisions, reduces congestion, and enhances the utilization of multi-path network topologies within RoCEv2 environments.

This document outlines the concept, details the packet processing method, and describes the mapping of the QP pair to the IPv6 Flow Label field. The subsequent sections will cover the mechanism in detail, discuss its advantages over existing solutions, and present use cases for its implementation in intelligent computing and data center networks.

2. Conventions Used in This Document

2.1. 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.

2.2. Abbreviations

AIDC: Artificial Intelligence Data Center

RoCEv2: RDMA over Converged Ethernet version 2

RDMA: Remote Direct Memory Access

QP: Queue Pair

IB BTH: InfiniBand Base Transport Header

IB DETH: InfiniBand Datagram Extended Transport Header

CRC32: Cyclic Redundancy Check 32-bit algorithm.

3. Flow-Level Load Balancing Based on the IPv6 Flow Label

3.1. Construction of the Hash Input

Ensuring the generated Flow Label can uniquely identify an RDMA flow while possessing sufficient randomness to minimize collision probability is critical. The procedure for constructing the hash input is as follows:

1. Extract the QP Pair:

* Src_QP: Extracted from the IB DETH header, 24 bits long (e.g., 0x123456).

* Dst_QP: Extracted from the IB BTH header, 24 bits long (e.g., 0xABCDEF).

2. Generate the Entropy Source:

To increase hash randomness, an entropy source is introduced. This scheme recommends using portions of the IPv6 addresses.

* Take the lower 16 bits of the IPv6 source address as the first entropy source, Entropy_Src.

* Take the lower 16 bits of the IPv6 destination address as the second entropy source, Entropy_Dst.

3.2. Hash by CRC32 Algorithm

This draft uses CRC32 as the core hash algorithm and Initialize the CRC register to 0xFFFFFFFF. CRC32 offers advantages such as fast computation, hardware-friendly implementation, and a low collision rate, making it highly suitable for line-rate forwarding in network devices.

First step is Byte-wise Split (using Hash_Input = 0x123456ABCDEF00010002): 0x12, 0x34, 0x56, 0xAB, 0xCD, 0xEF, 0x00, 0x01, 0x00, 0x02

Second step is iterative Processing per Byte (using the first byte 0x12 as an example):

Step 1 (XOR): XOR the lower 8 bits of the CRC register with the byte 0x12.

Step 2 (8-bit Shift-XOR Loop): Process the result from Step 1 bit-by-bit for 8 iterations. In each iteration:

a. Check the least significant bit (LSB) of the CRC register.

b. Shift the CRC register right by one bit (pad the high bit with 0).

c. If the LSB was 1, XOR the result with the generator polynomial 0x04C11DB7.

Repeat Steps 1 and 2 for all subsequent bytes.

After processing all bytes, the value in the CRC register is the final 32-bit hash result (e.g., 0x8E4D7A2F).

3.3. Flow Label Field Population

From the 32-bit CRC32 hash result, take the lower 20 bits as the Flow Label value and write this 20-bit value into the Flow Label field of the IPv6 header.

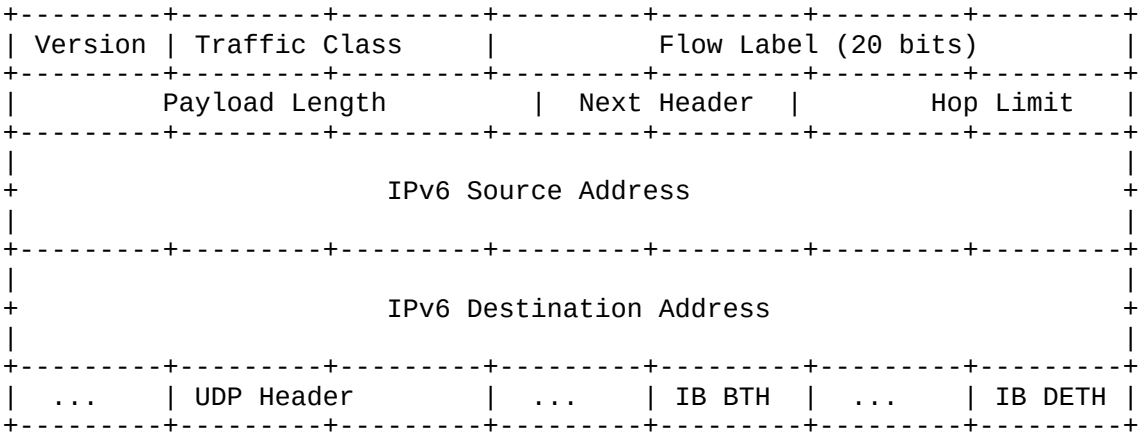

Figure 1: Updated IPv6 Header Structure Showing the Newly Populated Flow Label Field

4. IANA Considerations

This document makes no request to IANA.

5. Security Considerations

5.1. Security issue

This scheme only modifies the Flow Label field of the IPv6 header, which is performed by the ingress network device. It does not involve altering the packet payload and does not affect end-to-end application-layer security (e.g., IPsec). The modification does not change IP addresses or port numbers, thus imposing no additional processing burden on existing stateful firewalls or NAT devices.

5.2. Compatibility issue

End-to-End Protocol: The receiving device typically ignores the Flow Label field, making the scheme completely transparent to terminals that support standard IPv6.

Intermediate Devices: All network devices supporting the IPv6 Flow Label field can benefit from this scheme. For legacy devices that do not support the Flow Label, they can still forward packets based on the traditional 5-tuple. The scheme will not cause connectivity issues, but the full performance benefits will not be realized.

Hardware-Friendly Implementation: The CRC32 algorithm is widely supported in existing network ASICs. Implementing the required logic (parsing BTH/DETH headers, performing the hash, and modifying the Flow Label) is relatively straightforward and requires minimal changes to existing hardware.

6. References

6.1. 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>.

[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>.

Contributors

Thanks to all the contributors.

Authors' Addresses

Jiayuan Hu (editor)
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou
Guangzhou, 510000
China
Email: hujy5@chinatelecom.cn

Xia Gong (editor)
China Telecom
109, West Zhongshan Road, Tianhe District
Guangzhou
Guangzhou, 510000
China
Email: gongxia@chinatelecom.cn