RTGWG Working Group
Internet Draft
Intended status: Standards Track
Expires: 05 January 2026

W. Cheng
China Mobile
C. Lin
New H3C Technologies

July 4, 2025

# Enhanced ECMP for AI Cluster

draft-cheng-rtgwg-enhanced-ecmp-00

## Abstract

In AI training scenarios, the current mainstream load balancing technology is per-flow ECMP. However, hash collision issues lead to imbalanced traffic distribution, adversely affecting application performance.

To address this problem, this document proposes an enhanced ECMP method that resolves load imbalance caused by hash collisions. The proposed solution effectively improves load balancing efficiency, reduces network congestion, and enhances overall network performance.

Table of Contents



## 1. Introduction

Currently, there are two granularities for network load balancing: per-flow ECMP and per-packet forwarding.

As illustrated in Figure 1, the per-flow ECMP method employs flow characteristic-based hashing (typically using the five-tuple) to distribute traffic across multiple ECMP paths. This approach works effectively in environments with numerous small flows and absence of elephant flows. Its primary advantage is the elimination of packet reordering issues.

However, this method presents limitations when dealing with either:

A limited number of flows, or The presence of elephant flows.

In such cases, five-tuple-based hashing may lead to hash collisions, causing disproportionate mapping of oversized flows to the same path. This results in suboptimal load balancing performance.

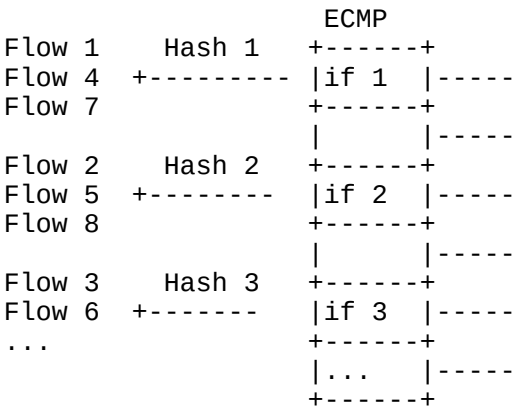


Figure 1 Per-flow ECMP

The other approach is per-packet forwarding. This method applies hashing to each individual packet, distributing traffic across different ECMP paths, as illustrated in Figure 2. Theoretically, it achieves optimal load-balancing granularity. However, it introduces severe packet reordering within the same flow, necessitating additional mechanisms (e.g., reordering buffers or sequence tracking) to handle out-of-order delivery. This imposes higher demands on network infrastructure.

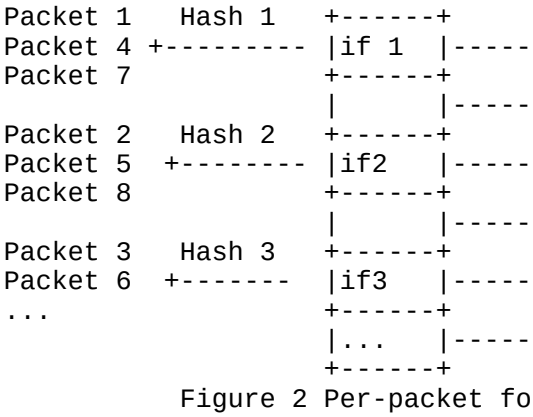


Figure 2 Per-packet forwarding

In AI training scenarios, the current mainstream load balancing technology is per-flow ECMP. However, hash collision issues lead to imbalanced traffic distribution, adversely affecting application performance.

To address this problem, this document proposes an enhanced ECMP method that resolves load imbalance caused by hash collisions. The proposed solution effectively improves load balancing efficiency, reduces network congestion, and enhances overall network performance.

### 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. Motivation

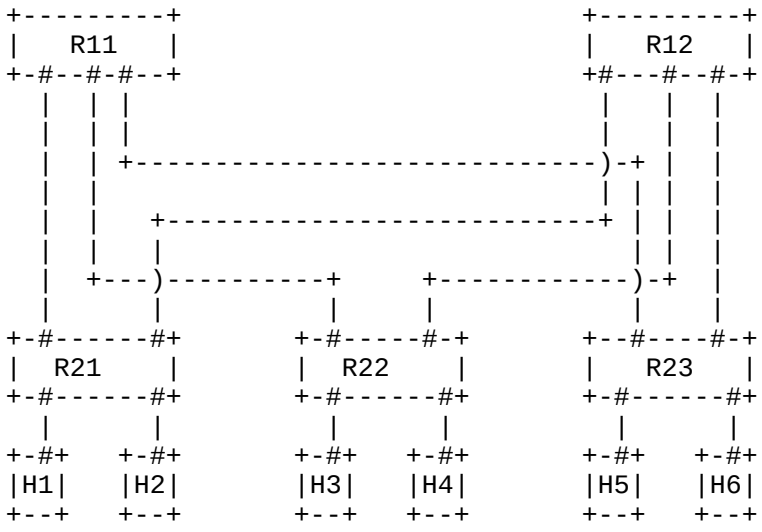


Figure 3 AI Network

Due to the unique traffic patterns in AI training networks - characterized by a limited number of flows - achieving balanced load distribution becomes challenging. Traditional flow-based load

balancing strategies often result in uneven traffic distribution, potentially leading to network congestion. While packet-based approaches can mitigate this imbalance to some degree, they introduce packet reordering issues as flow packets may traverse different paths, requiring additional network-level reordering mechanisms.

This document proposes two enhanced ECMP methods to address the load imbalance issue in AI training networks and improve the overall network performance.

## 3. Solution

### 3.1. ECMP based on source ingress interface

Group the ingress interfaces for traffic, assign an ECMP number to the interfaces within the same group, and then perform ECMP hashing based on this ECMP number. This method is suitable when the forwarding traffic size for each ingress interface is roughly the same.

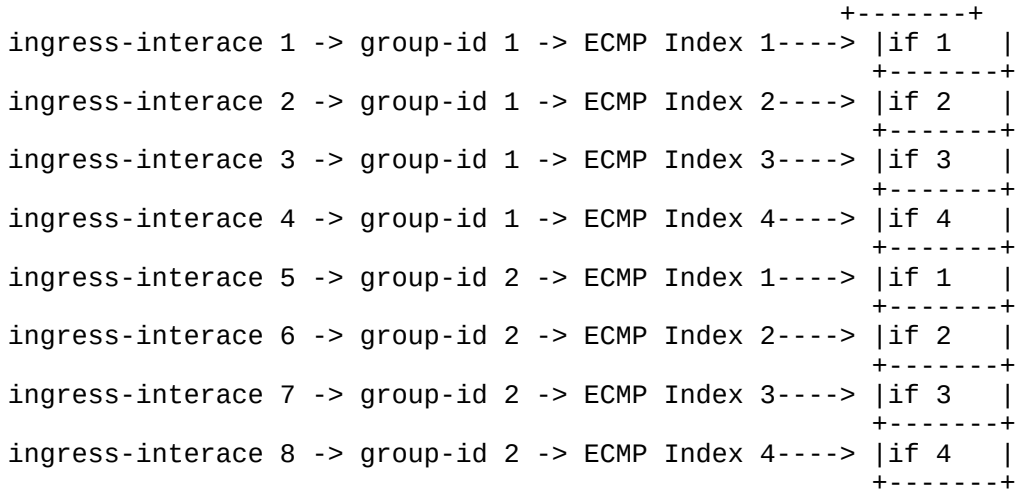


Figure 4 ECMP based on source ingress interface

As shown in Figure 4 above, the eight ingress interfaces are divided into two groups, with four interfaces in each group.

Within each group:

* The ingress interfaces are assigned ECMP numbers 1, 2, 3, and 4 respectively.

* ECMP hashing is performed based on these assigned ECMP numbers to select corresponding egress interfaces for forwarding.

For traffic entering through the four ingress interfaces in Group 1: Different egress interfaces are selected for forwarding (four distinct paths)

Similarly, for traffic entering through Group 2's four ingress interfaces: Different egress interfaces are selected for forwarding (four distinct paths)

## 3.2. ECMP based on egress Grouping

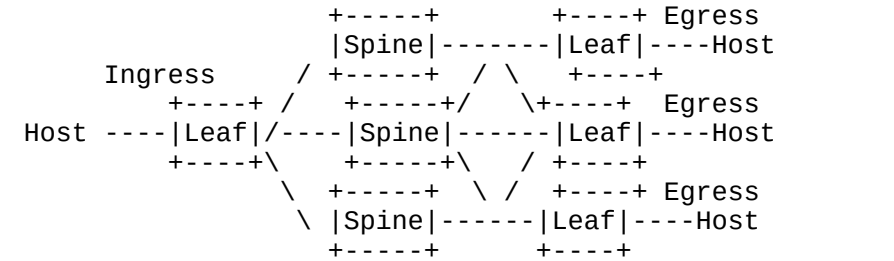


Figure 5 ECMP based on egress Grouping

As shown in the figure, the source HOST connects to the source Leaf via an Ingress interface, while the destination HOST connects to the destination Leaf through an Egress interface. Multiple ECMP (Equal-Cost Multi-Path) links exist between the Leaf switches and multiple Spine devices.

To improve load-balancing distribution uniformity, the ECMP interfaces connecting to multiple Spine devices are grouped on the source Leaf. This grouping can be configured on the Leaf device. For example, if there are 128 equal-cost links between the source Leaf and Spine devices, they can be divided into groups of 4 interfaces each (Group 1: interfaces 1-4; Group 2: interfaces 5-8, etc.).

For traffic load balancing, flows are first mapped to specific groups based on their location information (flows with the same location information are assigned to the same group), and then hash-based load balancing is performed within each group.

The location information used for group mapping can be either the source's Ingress interface or the destination's Egress interface.

By implementing fine-grained grouping of ECMP interfaces, this solution achieves more uniform traffic load distribution, thereby addressing current issues of imbalanced load sharing and flow collisions.

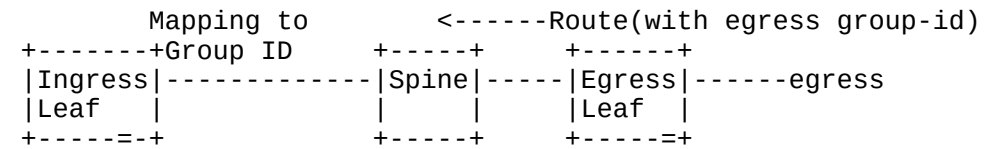


Figure 6 Route carries remote egress group-id attribute

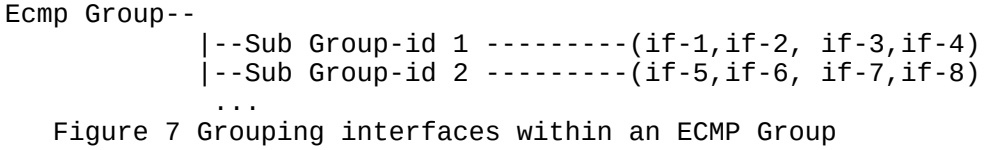


Figure 7 Grouping interfaces within an ECMP Group

First, as shown in Figure 6, when the Egress Leaf advertises a route, it carries the egress group index, which can be composed of the local device's router-id and group-id (see Section 4 for details).

On the Ingress Leaf, the ECMP egress interfaces toward the Spine devices are grouped (as illustrated in Figure 7).

When the Ingress Leaf receives a route, it extracts the remote egress group index carried in the route. It then maps this remote egress group index to a local ECMP subgroup index, effectively directing traffic to the corresponding subset of interfaces for forwarding.

This ensures that flows destined for different remote addresses are load-balanced across different ECMP subgroups, improving distribution granularity. Refer to Figure 8 for details.

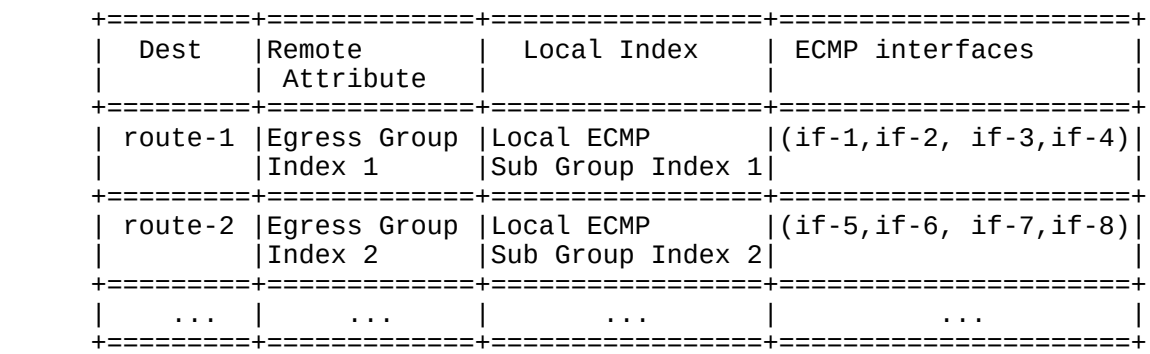


Figure 8

## 4. Protocol Extension

This document defines a new extended community attribute type to carry the ECMP ID associated with a route. The ID comprises a 4-byte Router ID and a 2-byte Group-ID. The format is as follows:

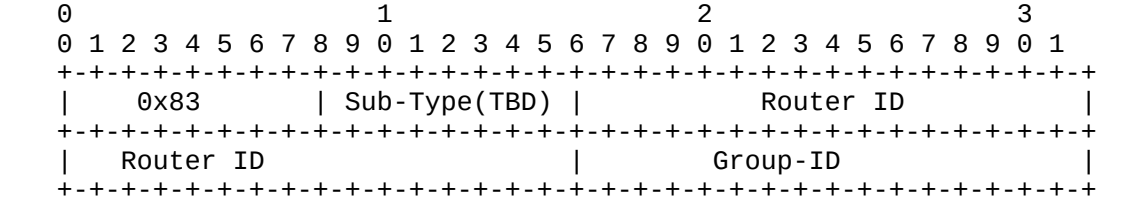


Figure 10 ECMP GroupID Extended Community

Sub-Type: BGP_EXT-COMM-ECMP-GROUPID (TBD)

Value Structure:

Local BGP RouterID (4 bytes)

ECMP-Group-ID Value (2 bytes)

## 5. Security Considerations

TBD.

## 6. IANA Considerations

Registry Name: Transitive BGP_EXT-COMM-ECMP-GROUPID Community Sub-Types

TBD: BGP_EXT-COMM-ECMP-GROUPID

## 7. References

### 7.1. Normative References

TBD.

### 7.2. Informational References

TBD

## Authors' Addresses

Weiqiang Cheng
China Mobile
Beijing
China
Email: chengweiqiang@chinamobile.com

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