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

Z. Li
Z. Du
China Mobile
W. Cheng
J. Wang
G. Zhang
Centec Networks
28 February 2026

Combined Hash Mechanism for ECMP/LAG Load Balancing
draft-li-rtgwg-ecmp-hash-combination-00

Abstract

This document specifies a combined hash mechanism for load balancing in Equal-Cost Multi-Path (ECMP) and Link Aggregation Group (LAG) environments. When processing tunneled traffic, traditional hash methods that operate on either outer or inner packet fields may result in uneven traffic distribution (polarization). This specification defines a method to extend effective hash entropy by combining multiple independently computed hash values, enabling network devices to consider multi-layer packet information for path selection, thereby improving traffic distribution uniformity. This mechanism is compatible with existing hash computation architectures and does not require packet format modifications or new protocol field definitions.

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.

Table of Contents

1. Introduction

Equal-Cost Multi-Path (ECMP) routing and Link Aggregation Groups (LAGs) are widely deployed technologies in modern networks for bandwidth scaling and redundancy. As described in [RFC2991] and [RFC2992], these technologies improve network performance and reliability by distributing traffic across multiple equal-cost paths.

In ECMP/LAG environments, routers typically employ hash-based methods to select forwarding paths. As analyzed in [RFC2992], hash algorithms extract key information from packet header fields to compute hash values used for path selection. Typical hash inputs include the 5-tuple (source/destination IP addresses, source/ destination ports, protocol type) or subsets thereof.

However, [RFC6438] identifies that in tunnel scenarios, when multiple distinct user flows are encapsulated within the same tunnel, hashing based solely on outer headers causes all encapsulated flows to be mapped to the same path, resulting in polarization. While RFC 6438 addresses this by setting the flow label in the outer IPv6 header, this requires tunnel endpoint cooperation and only applies to IPv6 outer encapsulation scenarios.

This document defines a complementary combined hash mechanism that extends available hash entropy by combining multiple independently computed hash values within the forwarding device, thereby improving load distribution for tunneled traffic without modifying packet formats. This mechanism can be used in conjunction with [RFC6438] and [RFC6790], or deployed independently.

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.

3. Terminology

This section defines terms used in this document. Some terms are inherited from [RFC2991], [RFC2992], [RFC6438], and [RFC6790].

Equal-Cost Multi-Path (ECMP): A routing strategy that allows packet forwarding to a single destination over multiple paths with equal routing cost. See [RFC2991].

Link Aggregation Group (LAG): A technology that bundles multiple physical links into a single logical link, providing bandwidth aggregation and link redundancy. See [IEEE802.1AX].

Hash Entropy: The amount of variability in the input to a hash function. Higher entropy typically leads to more uniform hash output distribution.

Polarization: In ECMP/LAG environments, the phenomenon where traffic concentrates on a subset of paths due to insufficient variability in hash inputs. See [RFC6438].

Base Hash Value: A hash value computed by a single hash computation unit based on a specific set of packet fields.

Combined Hash Value: A new hash value derived by applying a combination operation to two or more base hash values.

Hash Field Set: The set of packet fields used as hash input. This document defines standard field sets including L2 (source MAC, destination MAC, VLAN tag, EtherType), L3 (source IP, destination IP, protocol), L4 (source port, destination port), OUTER (outer header fields), and INNER (inner/payload header fields).

4. Problem Statement

4.1. Tunnel Traffic Polarization

As described in Section 1.1 of [RFC6438], in tunnel scenarios, when traffic from multiple sources is aggregated into a single tunnel, all encapsulated packets have the same outer source and destination addresses. If ECMP/LAG paths exist between intermediate routers, and path selection is based solely on hashing the outer 5-tuple (or its subset), all tunnel traffic will be mapped to the same path, preventing effective load sharing.

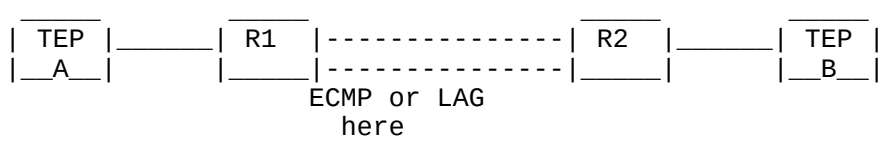

Figure 1: Tunnel Scenario with ECMP/LAG

4.2. Limitations of Existing Solutions

Existing solutions each have their applicable scenarios and limitations:

* *IPv6 Flow Label Approach [RFC6438]:* Requires IPv6 outer encapsulation and cooperation from the ingress tunnel endpoint (TEP) to set the flow label.

* *MPLS Entropy Label Approach [RFC6790]:* Only applicable to MPLS networks and requires all LSRs along the path to support entropy labels.

* *Deep Packet Inspection (DPI):* While some devices support inspecting inner packet fields for hashing, this increases processing complexity and latency, and is unavailable when packets are encrypted.

4.3. Motivation for Combined Hashing

In practical network devices, a common implementation approach is to configure multiple parallel hash computation units, each independently configurable with its hash field set. However, final path selection typically can only use one of these hash values. The combined hash mechanism defined in this document allows combining multiple base hash values to produce a combined hash value that considers multi-layer information for final path selection.

5. Combined Hash Mechanism Overview

5.1. Basic Principle

The core idea of the combined hash mechanism is to apply combination operations to multiple independently computed base hash values, producing a new combined hash value. Through this approach:

1. Each base hash value can be independently computed from different packet field sets (e.g., outer L3, inner L3, L2 fields).

2. The combination operation merges the entropy of multiple base hash values, producing a combined hash value containing more information.

3. Final path selection can be based on the combined hash value, thus indirectly achieving comprehensive consideration of multiple field sets.

5.2. Architecture

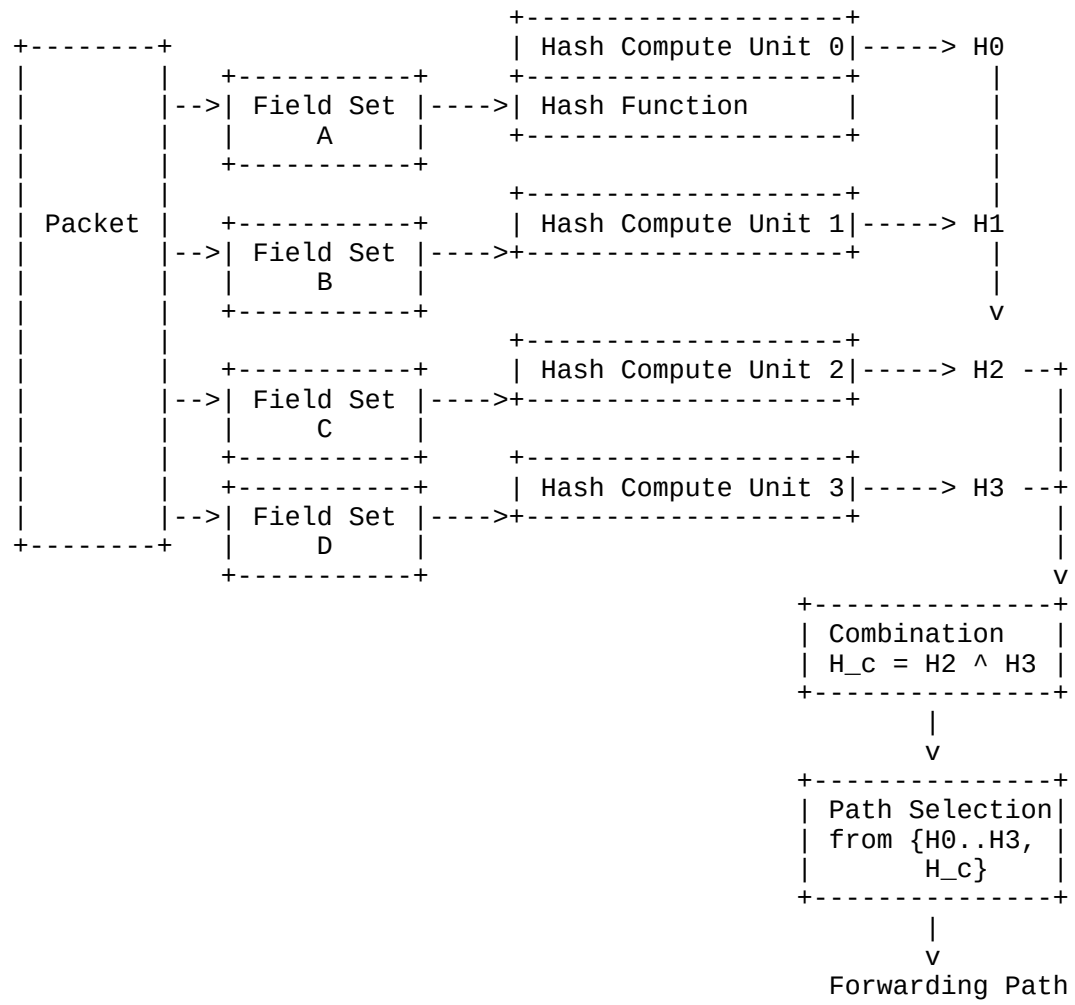

Figure 2: Combined Hash Mechanism Architecture

5.3. Relationship with Existing Mechanisms

This mechanism complements [RFC6438] and [RFC6790]. When the outer layer uses IPv6 and TEPs support RFC 6438, the flow label approach can be preferred. When MPLS networks are deployed with RFC 6790 support, the entropy label approach can be preferred. When the above conditions are not met, the combined hash mechanism provides a purely local implementation alternative. These approaches can be used together for optimal results.

6. Protocol Specification

6.1. Base Hash Computation

Devices implementing this specification MUST support at least two independent base hash computation units. Each computation unit SHOULD support independent configuration of its hash field set.

RECOMMENDED hash field set configuration options include:

Field Set ID	Included Fields	Typical Use Case
L2	Source MAC, Dest MAC, VLAN, EtherType	Ethernet switching
L3	Source IP, Dest IP, Protocol	IP routing
L4	Source Port, Dest Port	Transport layer differentiation
OUTER	Outer packet L2/L3/ L4 fields	Tunneled traffic
INNER	Inner packet L2/L3/ L4 fields	Tunneled traffic

Table 1: Hash Field Set Options

Base hash value bit-width SHOULD be at least 16 bits. Implementations MAY support larger bit-widths for better distribution characteristics.

6.2. Combination Operations

Implementations MUST support XOR combination operation on two base hash values:

H_combined = H_x XOR H_y

where H_x and H_y are any two selected base hash values.

Implementations MAY additionally support the following combination operations:

Addition combination:

H_combined = (H_x + H_y) mod 2^n

where n is the hash value bit-width.

Concatenation hash:

H_combined = Hash(H_x || H_y)

where || denotes bit-level concatenation and Hash() is the configured hash function.

When using XOR operation, implementations MUST detect whether H_x and H_y might be identical (computed from the same field set). If configured to XOR identical hash values, implementations SHOULD generate a warning, as this will result in a combined value of zero.

6.3. Path Selection

Implementations MUST support selecting the final hash value for path selection from any base hash value (H0, H1, H2, H3, ...) or the combined hash value (H_combined).

Final hash value selection SHOULD be configurable.

For ECMP path selection, the hash-threshold algorithm described in [RFC2992] or a similar deterministic algorithm SHOULD be used to ensure all packets of the same flow are forwarded to the same path.

7. Configuration Model

This section defines parameters required for configuring the combined hash mechanism. Specific configuration interfaces (CLI, NETCONF/ YANG, etc.) are outside the scope of this document.

7.1. Configuration Parameters

Parameter	Type	Range	Default
hash-unit-count	int	2-8	4
hash-unit[i].field-set	enum	L2,L3,L4,OUTER,INNER	-
combination.enabled	bool	true/false	false
combination.input-1	int	0 to (count-1)	-
combination.input-2	int	0 to (count-1)	-

combination.operation	enum	XOR,ADD,CONCAT-HASH	XOR
final-hash-select	int	0 to count	0

Table 2: Configuration Parameters

7.2. Configuration Constraints

The following configuration constraints MUST be enforced:

1. combination.input-1 and combination.input-2 SHOULD NOT be configured to the same value. If configured to the same value, implementations MUST generate a warning.

2. When combination.enabled is false, the valid range for final-hash-select is 0 to (hash-unit-count - 1).

3. When combination.enabled is true, final-hash-select MAY additionally select the combined hash value.

8. Operational Procedures

8.1. Initialization

Upon device startup, the following initialization procedure MUST be executed:

1. Read and validate configuration parameter validity.

2. Configure each hash computation unit with its corresponding field set.

3. If combined hashing is enabled, initialize the combination operation module.

4. Configure path selection logic to use the specified final hash value.

8.2. Packet Processing

For each received packet, the processing procedure is as follows:

Step 1: Base Hash Computation

* Each hash computation unit extracts configured fields from the packet in parallel.

* Execute hash function computation on extracted fields.

* Produce base hash values H0, H1, ..., Hn.

Step 2: Combined Hash Computation (if enabled)

* Select two base hash values according to configuration.

* Execute the configured combination operation.

* Produce combined hash value H_combined.

Step 3: Path Selection

* Select final hash value according to final-hash-select configuration.

* Execute path selection algorithm using final hash value (e.g., hash-threshold algorithm from [RFC2992]).

* Determine forwarding path and forward packet.

8.3. Configuration Change Handling

When configuration changes occur:

* Implementations SHOULD support online configuration changes without device restart.

* Configuration changes MUST take effect immediately for subsequent packets.

* Implementations SHOULD log configuration change events for audit purposes.

9. Relationship with Existing Mechanisms

9.1. Relationship with RFC 6438

[RFC6438] defines a method for using the flow label in IPv6 tunnel scenarios for ECMP/LAG load balancing. That method requires the ingress tunnel endpoint to set a flow label value in the outer IPv6 header based on information computed from the inner packet.

The combined hash mechanism defined in this document complements [RFC6438]: RFC 6438 executes at tunnel endpoints and affects all intermediate nodes along the tunnel path, while the combined hash mechanism executes locally at a single forwarding node and does not affect other nodes. When both mechanisms are used together, their effects can accumulate.

9.2. Relationship with RFC 6790

[RFC6790] defines the MPLS entropy label mechanism, allowing entropy information for load balancing to be carried in the MPLS label stack.

The mechanism defined in this document complements [RFC6790]: RFC 6790 requires nodes along the path to support entropy label parsing, while the combined hash mechanism does not require support from other nodes. In MPLS networks, both mechanisms can be used simultaneously.

9.3. Relationship with RFC 2992

[RFC2992] analyzes the hash-threshold algorithm for ECMP. The final hash value produced by the combined hash mechanism defined in this document can be used as input to the RFC 2992 algorithm. This document does not modify or replace the path selection algorithm defined in [RFC2992].

10. Security Considerations

10.1. Configuration Security

The combined hash mechanism is a local device behavior and does not introduce new attack surfaces beyond those inherent in hash-based load balancing. However, operators should be aware that hash configuration details, if exposed, could allow traffic engineering attacks. Implementations MUST enforce access control for hash configuration parameters. Implementations SHOULD log configuration changes for audit purposes.

11. IANA Considerations

This document does not require IANA to allocate any protocol parameters or create new registries. The combined hash mechanism is purely a local implementation optimization and does not involve protocol interactions.

If future working group consensus determines that configuration parameter namespaces need to be standardized (e.g., for YANG models), IANA action may be requested at that time.

12. References

12.1. Normative References

[RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, RFC 2119, DOI 10.17487/RFC2119, 28 February 2026, <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, 28 February 2026, <https://www.rfc-editor.org/info/rfc8174>.

12.2. Informative References

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, 28 February 2026, <https://www.rfc-editor.org/info/rfc2991>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, 28 February 2026, <https://www.rfc-editor.org/info/rfc2992>.

[RFC6438] Carpenter, B. and S. Amante, "Using the IPv6 Flow Label for Equal Cost Multipath Routing and Link Aggregation in Tunnels", RFC 6438, DOI 10.17487/RFC6438, 28 February 2026, <https://www.rfc-editor.org/info/rfc6438>.

[RFC6790] Kompella, K., Drake, J., Amante, S., Henderickx, W., and L. Yong, "The Use of Entropy Labels in MPLS Forwarding", RFC 6790, DOI 10.17487/RFC6790, 28 February 2026, <https://www.rfc-editor.org/info/rfc6790>.

[RFC7424] Krishnan, R., Yong, L., Ghanwani, A., So, N., and B. Khasnabish, "Mechanisms for Optimizing Link Aggregation Group (LAG) and Equal-Cost Multipath (ECMP) Component Link Utilization in Networks", RFC 7424, DOI 10.17487/RFC7424, 28 February 2026, <https://www.rfc-editor.org/info/rfc7424>.

[IEEE802.1AX] IEEE, "IEEE Standard for Local and metropolitan area networks - Link Aggregation", IEEE Std 802.1AX-2020, 28 February 2026.

Appendix A. Deployment Scenario Examples

A.1. Data Center Overlay Networks

In overlay networks such as VXLAN/NVGRE, large amounts of tenant traffic traverse shared underlay tunnels. Example configuration:

* Hash unit 0: OUTER field set (outer IP addresses)

* Hash unit 1: INNER field set (inner IP addresses)

* Combination operation: H0 XOR H1

* Final selection: Combined hash value

This configuration enables path selection to consider both outer tunnel information and inner tenant flow information.

A.2. Service Provider Core Networks

In MPLS backbone networks, entropy labels may not be fully deployed. Example configuration:

* Hash unit 0: L3 field set (IP 5-tuple)

* Hash unit 1: MPLS label stack

* Combination operation: H0 XOR H1

* Final selection: Combined hash value

This configuration provides improved load distribution without relying on entropy labels.

Authors' Addresses

Zhiqiang Li
China Mobile
32 Xuanwumen West Street
Beijing
100053
China
Email: lizhiqiangyjy@chinamobile.com

Zongpeng Du
China Mobile
32 Xuanwumen West Street

Beijing
100053
China
Email: duzongpeng@chinamobile.com

Wei Cheng
Centec Networks
Suzhou
215000
China
Email: chengw@centec.com

Junjie Wang
Centec Networks
Suzhou
215000
China
Email: wangjj@centec.com

Guoying Zhang
Centec Networks
Suzhou
215000
China
Email: zhanggy@centec.com