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

Load Balancing Hash Polarization Mitigation Extension
draft-li-rtgwg-hash-polarization-mitigation-00

Abstract

This document defines a hash polarization mitigation extension for Link Aggregation (LAG) and Equal-Cost Multi-Path (ECMP) routing. This document specifies hash input field selection rules, Shift Factor definition and generation methods, hash value adjustment algorithms, and normative requirements for device processing procedures.

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

1.1. Background

Link Aggregation (LAG), defined in IEEE 802.1AX [IEEE802.1AX], and Equal-Cost Multi-Path (ECMP) routing, described in [RFC2991] and [RFC2992], are fundamental mechanisms for network load balancing. These mechanisms compute hash values from packet fields and map the hash values to one path within the set of available paths.

In multi-tier network topologies, when devices at each tier employ identical hash algorithms and identical input field configurations, packets with identical hash inputs produce identical hash values at each tier and are consequently mapped to the same relative path positions. This behavior causes traffic to persistently aggregate on specific physical paths, a phenomenon termed hash polarization.

1.2. Document Scope

This document defines the following:

* Hash input field selection and configuration rules

* Shift Factor definition, value range, and generation methods

* Shift Factor-based hash value adjustment algorithm

* Normative device packet processing procedures

* Error handling requirements

* Manageability requirements

This document does not define the following:

* Specific LAG or ECMP protocol specifications

* Data plane encapsulation formats

* Specific hash algorithm implementations

* Control plane protocols

1.3. 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. Terminology and Definitions

2.1. Abbreviations

ECMP: Equal-Cost Multi-Path routing.

HRNG: Hardware Random Number Generator.

LAG: Link Aggregation.

2.2. Term Definitions

Hash Input Data: The sequence of field values extracted from a packet and used for hash computation.

Initial Hash Value: The output value produced by the hash function operating on Hash Input Data.

Adjusted Hash Value: The value resulting from processing the Initial Hash Value with the Shift Factor, used for final path selection.

Shift Factor: An unsigned integer parameter used for circular bit rotation of the Initial Hash Value.

Hash Polarization: A phenomenon where traffic persistently aggregates on specific paths due to identical hash configurations across multi-tier load balancing devices.

Path Index: A non-negative integer identifying a specific path within the set of available paths, numbered starting from zero.

3. Hash Input Field Selection

3.1. Field Types

3.1.1. Mandatory Fields

Devices conforming to this specification MUST support the following hash input fields:

Layer 2 fields: Source MAC Address (48 bits), Destination MAC Address (48 bits), EtherType (16 bits), VLAN ID (12 bits).

Layer 3 fields: Source IP Address (32 bits for IPv4, 128 bits for IPv6), Destination IP Address (32 bits for IPv4, 128 bits for IPv6), Protocol or Next Header (8 bits), Flow Label (IPv6 only, 20 bits).

Layer 4 fields: Source Port (16 bits), Destination Port (16 bits).

3.1.2. Optional Fields

Devices conforming to this specification SHOULD support the following optional fields:

Inner tunnel fields: For packets with tunnel encapsulation such as VXLAN [RFC7348], GRE, or MPLS, support for extracting Layer 2, Layer 3, and Layer 4 fields from inner packets.

Devices conforming to this specification MAY support the following extended fields:

Custom offset fields: Byte sequences of specified length extracted from specified byte offset positions within packets.

3.2. Field Selection Configuration

Devices MUST provide a configuration mechanism allowing independent enabling or disabling of each field defined in Section 3.1 for hash computation participation.

For devices supporting inner tunnel fields, the configuration mechanism MUST allow specification of one of the following options: use outer fields only; use inner fields only; use a combination of both outer and inner fields.

Devices SHOULD support bitmask configuration, allowing specification of a mask value for each field. When a mask is configured, only bit positions corresponding to mask bits set to 1 participate in hash computation.

3.3. Hash Input Data Assembly

Devices MUST assemble Hash Input Data according to the following rules:

* Extract field values sequentially in the order specified by configuration. If field order is not configured, devices MUST use an implementation-defined deterministic order and MUST document this order.

* Field values are represented in network byte order (big-endian).

* Concatenate field values sequentially into a contiguous byte sequence, forming the Hash Input Data.

* If a field does not exist in the packet (e.g., IP address fields for non-IP packets, or port fields for non-TCP/UDP packets), that field position MUST be filled with all-zero values.

* If a field mask is configured, devices MUST perform a bitwise AND operation between the field value and the mask, then include the result in the Hash Input Data.

3.4. Default Configuration

When no explicit configuration is present, devices MUST use the following default field set: Source IP Address, Destination IP Address, Protocol, Source Port, Destination Port. This field set is commonly referred to as the five-tuple.

4. Shift Factor

4.1. Definition and Value Range

The Shift Factor is an unsigned integer used for circular bit rotation of the Initial Hash Value.

Let W denote the bit width of hash values. The valid value range for the Shift Factor is the closed interval [0, W-1].

Devices MUST support a hash value width of at least 16 bits. Devices SHOULD support a hash value width of 32 bits. Devices MAY support other hash value widths.

4.2. Generation Methods

Devices MUST support at least one of the following Shift Factor generation methods:

Static Configuration: Explicit specification of the Shift Factor value through the management interface. When static configuration is used, the configured value MUST be within the valid value range.

Random Generation: Automatic generation of a random value as the Shift Factor during device initialization. When random generation is used, devices SHOULD use a Hardware Random Number Generator (HRNG) or cryptographically secure pseudo-random number generator. Devices MUST NOT use predictable pseudo-random number generators such as Linear Congruential Generators.

Devices MAY support regeneration of the Shift Factor during runtime.

4.3. Persistence

After device restart, the Shift Factor behavior depends on the generation method:

If static configuration is used, devices MUST restore the configured Shift Factor value after restart.

If random generation is used, devices MAY generate a new random value after restart, or MAY persistently store and restore the previous value. Devices SHOULD document their behavior.

5. Hash Value Computation and Path Selection

5.1. Initial Hash Value Computation

Devices MUST input the Hash Input Data to a hash function and compute the Initial Hash Value.

Hash function selection is outside the scope of this document. Common choices include CRC polynomial families and XOR-based folding algorithms.

The hash function output bit width MUST equal the hash value width W defined in Section 4.1.

Hash functions SHOULD have good distribution uniformity, meaning that for randomly distributed inputs, output values are approximately uniformly distributed within the range [0, 2^W - 1].

5.2. Hash Value Adjustment Algorithm

Let H denote the Initial Hash Value, W denote the bit width, and S denote the Shift Factor. The Adjusted Hash Value H’ MUST be computed according to the following algorithm:

H’ = ROR(H, S, W)

Where ROR is the circular right rotation function, defined as:

ROR(H, S, W) = (H >> S) OR (H << (W - S))

Operators are defined as follows: ">>" is logical right shift with zero-fill of high-order bits; "<<" is logical left shift with zero-fill of low-order bits; "OR" is bitwise OR operation.

When S equals 0, H’ equals H. Implementations MUST correctly handle this boundary condition.

When S equals W, this case should not occur per the value range constraint. If S is greater than or equal to W due to configuration error, devices MUST treat S as 0 and SHOULD log an error.

5.3. Path Selection

Let N denote the number of available paths (N > 0). Devices MUST compute the Path Index P according to the following formula:

P = H’ mod N

Where "mod" is the modulo operation, with a non-negative integer result.

Path indices are numbered starting from 0, with valid range [0, N-1].

Devices MUST forward packets to the path corresponding to Path Index P.

When the path set changes (e.g., member port failure or recovery), the N value changes accordingly. Devices MUST use the updated N value to compute Path Index for subsequent packets.

6. Device Processing Procedures

6.1. Receive Processing

When a device receives a packet requiring LAG or ECMP load balanced forwarding, the device MUST execute processing in the following order:

First, parse packet headers. For tunnel-encapsulated packets, if configuration requires use of inner fields, devices MUST complete inner header parsing.

Second, extract hash input fields from the packet according to current configuration.

Third, assemble Hash Input Data according to the rules in Section 3.3.

Fourth, compute the Initial Hash Value H.

Fifth, compute the Adjusted Hash Value H’ using the current Shift Factor according to the algorithm in Section 5.2.

Sixth, compute the Path Index P according to the formula in Section 5.3 using the current number of available paths N.

Seventh, forward the packet to path P.

6.2. Error Handling

6.2.1. Parsing Failures

If packet header parsing fails (e.g., truncated headers, checksum errors, non-conformant format), devices SHOULD handle the situation as follows:

* If some fields have been successfully extracted, devices MAY continue computation using extracted fields, with non-extracted fields filled with all-zero values.

* If no configured fields can be extracted, devices SHOULD forward the packet to the default path (Path Index 0).

* Devices SHOULD maintain a parsing failure counter.

6.2.2. Configuration Errors

If the Shift Factor configuration value exceeds the valid range, devices MUST treat the Shift Factor as 0, SHOULD log a configuration error, and SHOULD notify the administrator through an alerting mechanism.

If the hash input field configuration is empty (no fields enabled), devices SHOULD use the default configuration defined in Section 3.4 and SHOULD log a configuration warning.

7. Manageability Considerations

7.1. Configuration Parameters

Devices conforming to this specification SHOULD provide the following configuration parameters through management interfaces:

* Hash input field enable state, allowing independent configuration for each field.

* Field bitmask values, if mask functionality is supported.

* Shift Factor generation method selection.

* Static Shift Factor configuration value, when static configuration is used.

7.2. Operational State

Devices conforming to this specification SHOULD provide the following operational state queries through management interfaces:

* Currently effective Shift Factor value.

* Currently effective hash input field configuration.

* Per-path traffic statistics, including packet counts and byte counts.

* Parsing error counts.

* Configuration error counts.

7.3. Logging and Notifications

Devices SHOULD log events when the following occur:

* Shift Factor value changes, including initial generation, static configuration changes, and runtime regeneration.

* Hash input field configuration changes.

* Configuration error detection.

* Parsing error rate exceeds an implementation-defined threshold.

8. Security Considerations

8.1. Configuration Access Control

Shift Factor and hash input field configuration affects traffic distribution. Unauthorized configuration modification may cause abnormal traffic aggregation, resulting in congestion or service degradation.

Implementations MUST enforce authentication for configuration operations. Implementations MUST enforce authorization control for configuration operations, allowing only administrators with appropriate privileges to modify configuration. Implementations SHOULD maintain configuration change audit logs, including operation time, operator identity, and change content.

8.2. Randomness Quality

When using random generation to produce the Shift Factor, weak randomness may make the Shift Factor predictable, allowing attackers to infer traffic distribution patterns.

Implementations SHOULD use a Hardware Random Number Generator (HRNG). If software random number generators are used, implementations MUST use cryptographically secure pseudo-random number generators (CSPRNG), such as those based on AES-CTR or ChaCha20. Implementations MUST NOT use Linear Congruential Generators, Mersenne Twister (non-cryptographic variants), or other predictable generators.

8.3. Traffic Analysis Attacks

Attackers may infer load balancing configuration by observing network traffic patterns.

In deployments with high security requirements, operators MAY consider periodic Shift Factor configuration updates.

8.4. Hash Collision Attacks

Attackers may construct packet sets with identical hash values, causing traffic to concentrate on specific paths and resulting in path congestion.

Implementations SHOULD select hash functions with good collision resistance. Operators SHOULD deploy traffic monitoring mechanisms to detect abnormal traffic patterns. Operators MAY deploy rate limiting mechanisms as a mitigation measure.

8.5. Multi-Tenancy Isolation

In multi-tenant environments, configuration for different tenants MUST be mutually isolated. Tenants MUST NOT be able to view or modify Shift Factor or hash input field configuration of other tenants.

9. IANA Considerations

This document has no IANA actions.

The mechanism defined in this document is local device behavior and does not involve protocol field allocation, port number registration, or parameter encoding registration.

10. References

10.1. 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>.

10.2. Informative References

[IEEE802.1AX] IEEE, "IEEE Standard for Local and Metropolitan Area Networks-- Link Aggregation", IEEE Std 802.1AX.

[RFC2991] Thaler, D. and C. Hopps, "Multipath Issues in Unicast and Multicast Next-Hop Selection", RFC 2991, DOI 10.17487/RFC2991, November 2000, <https://www.rfc-editor.org/info/rfc2991>.

[RFC2992] Hopps, C., "Analysis of an Equal-Cost Multi-Path Algorithm", RFC 2992, DOI 10.17487/RFC2992, November 2000, <https://www.rfc-editor.org/info/rfc2992>.

[RFC7348] Mahalingam, M., Dutt, D., Duda, K., Agarwal, P., Kreeger, L., Sridhar, T., Bursell, M., and C. Wright, "Virtual eXtensible Local Area Network (VXLAN): A Framework for Overlaying Virtualized Layer 2 Networks over Layer 3 Networks", RFC 7348, DOI 10.17487/RFC7348, August 2014, <https://www.rfc-editor.org/info/rfc7348>.

Appendix A. Algorithm Examples (Informative)

This appendix provides computation examples of the hash value adjustment algorithm for reference purposes.

A.1. Computation Examples

Given conditions: Initial Hash Value H = 0x12345678; Hash value width W = 32 bits; ECMP group member count N = 4.

Example computations are shown in the following table:

S	Circular Right Rotation	H’	Path Index
0	ROR(0x12345678, 0, 32)	0x12345678	0
4	ROR(0x12345678, 4, 32)	0x81234567	3
8	ROR(0x12345678, 8, 32)	0x78123456	2
16	ROR(0x12345678, 16, 32)	0x56781234	0

Table 1: Computation Examples

A.2. Multi-Device Scenario

Consider three devices deployed in series, each configured with a different Shift Factor:

Device A is configured with Shift Factor 0. Device B is configured with Shift Factor 4. Device C is configured with Shift Factor 8.

When traffic flows with identical five-tuples traverse these three devices sequentially, the path selection results at each device are as shown in Table 1. Because each device has a different Adjusted Hash Value, path selection results exhibit differentiated distribution.

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