

IPPM
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

Packet-Triggered Statistics Reporting for In-situ OAM
draft-li-ippm-ioam-packet-triggered-reporting-00

Abstract

This document defines a packet-triggered statistics reporting extension for In-situ Operations, Administration, and Maintenance (IOAM). The extension specifies a 2-bit Cycle Identifier field that enables receiving nodes to trigger statistics reporting based on observed Cycle ID transitions in received packets. This document defines the Cycle ID field format, node processing procedures, and keepalive packet generation rules.

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

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	3
1.1. Requirements Language	3
2. Terminology	3
3. Protocol Overview	4
4. Cycle ID Field Format	4
4.1. Cycle ID Value Definitions	5
5. Encapsulating Node Procedures	5
5.1. Cycle ID Assignment	5
5.2. Statistics Maintenance	6
5.3. Statistics Reporting	6
5.4. Keepalive Packet Generation	7
5.5. Error Handling	8
6. Transit Node Procedures	8
6.1. Packet Processing	8
6.2. Error Handling	8
7. Decapsulating Node Procedures	9
7.1. State Maintenance	9
7.2. Report Triggering	9
7.3. Keepalive Packet Processing	9
7.4. Error Handling	10
8. Cycle ID State Machine	10
9. Manageability Considerations	11
9.1. Configuration Parameters	11
9.2. Monitoring and Statistics	11
10. Security Considerations	11
11. IANA Considerations	12
11.1. IOAM Option-Type Registry	12
11.2. IOAM Cycle ID Flags Registry	12
12. References	12
12.1. Normative References	12
12.2. Informative References	13
Appendix A. Processing Examples	13
A.1. Normal Operation	13
A.2. Keepalive Generation	14
Acknowledgements	14
Authors' Addresses	14

1. Introduction

In-situ Operations, Administration, and Maintenance (IOAM) [RFC9197] records operational and telemetry information within user data packets while they traverse a network path. Alternate marking methods for packet loss measurement rely on periodic color changes to delineate measurement intervals.

This document specifies a packet-triggered reporting extension that uses a four-value Cycle Identifier (Cycle ID) field. Receiving nodes observe Cycle ID transitions in arriving packets and use these transitions to trigger statistics reporting. This document defines the Cycle ID field encoding, procedures for encapsulating and decapsulating nodes, and the generation of keepalive packets to maintain measurement continuity.

The scope of this document is limited to the definition of the Cycle ID field and associated node behaviors. The transport of IOAM data and the encapsulation in specific protocols (e.g., IPv6, NSH, GRE) are outside the scope of this document.

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. Terminology

This document uses the terminology defined in [RFC9197]. The following additional terms are used in this document:

Cycle Identifier (Cycle ID): A 2-bit field that identifies the measurement interval to which a packet belongs. Valid values are 0, 1, 2, and 3.

Keepalive Packet: A packet generated by the encapsulating node when no user data packets are transmitted during a measurement interval, ensuring at least one packet per interval reaches the decapsulating node.

Encapsulating Node: The IOAM node that inserts the Cycle ID field into packets entering the IOAM domain.

Decapsulating Node: The IOAM node that processes the Cycle ID field and triggers statistics reporting upon Cycle ID transitions.

Transit Node: An IOAM node that processes packets within the IOAM domain, updating statistics based on the Cycle ID.

Measurement Interval: The time period associated with a single Cycle ID value during which packet statistics are accumulated.

3. Protocol Overview

This section provides a non-normative overview of packet-triggered statistics reporting.

The encapsulating node assigns a Cycle ID value (0-3) to each packet based on the current measurement interval. The Cycle ID value increments sequentially (0, 1, 2, 3, 0, ...) as measurement intervals advance. When the decapsulating node receives a packet with a Cycle ID different from the previously received Cycle ID, it triggers the reporting of statistics accumulated during the earlier interval.

To ensure that statistics reporting is triggered even during periods of no user traffic, the encapsulating node generates keepalive packets when no user data packets have been transmitted during an interval transition.

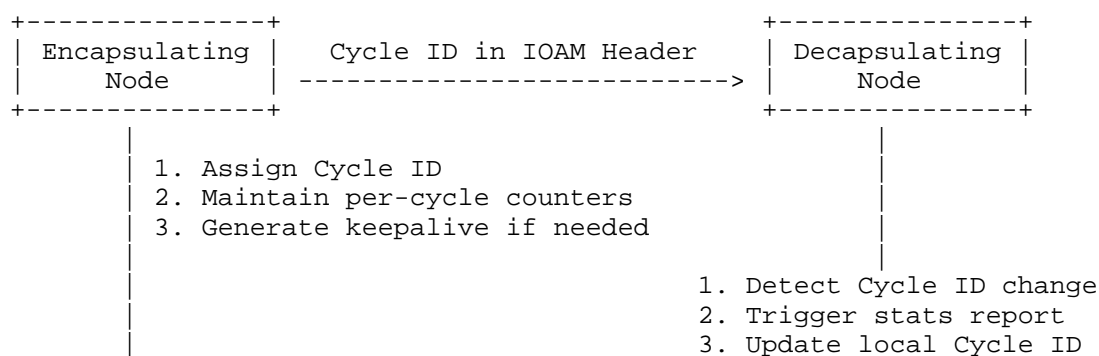


Figure 1: Packet-Triggered Reporting Overview

4. Cycle ID Field Format

The Cycle ID field is carried within the IOAM Option-Type header. The field format is defined as follows:

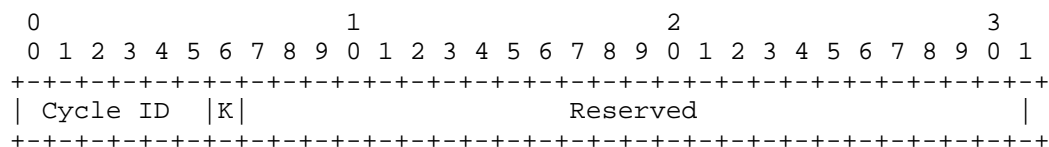


Figure 2: Cycle ID Field Format

The fields are defined as follows:

Cycle ID (2 bits): Identifies the measurement interval. Valid values are 0, 1, 2, and 3. The Cycle ID MUST increment sequentially modulo 4 (i.e., 0 -> 1 -> 2 -> 3 -> 0).

K (Keepalive Flag, 1 bit): When set to 1, indicates that this packet is a keepalive packet. When set to 0, indicates a user data packet. The K flag MUST be set to 0 for user data packets.

Reserved (29 bits): Reserved for future use. MUST be set to zero on transmission. MUST be ignored on reception.

4.1. Cycle ID Value Definitions

The following Cycle ID values are defined:

Value	Binary	Description
0	00	Measurement Interval 0
1	01	Measurement Interval 1
2	10	Measurement Interval 2
3	11	Measurement Interval 3

Table 1: Cycle ID Values

5. Encapsulating Node Procedures

This section specifies the normative procedures for encapsulating nodes.

5.1. Cycle ID Assignment

The encapsulating node MUST maintain a current Cycle ID value for each IOAM flow. The encapsulating node MUST assign the current Cycle ID value to all packets belonging to that flow.

The encapsulating node MUST increment the Cycle ID value at the beginning of each new measurement interval. The increment operation MUST be performed modulo 4:

$\text{new_cycle_id} = (\text{current_cycle_id} + 1) \bmod 4$

The duration of a measurement interval is a local configuration parameter. The default measurement interval duration is 1 second. Implementations SHOULD support configurable measurement interval durations in the range of 100 milliseconds to 60 seconds.

5.2. Statistics Maintenance

The encapsulating node MUST maintain per-flow, per-Cycle-ID statistics counters. At minimum, the following counters MUST be maintained:

- * Packet count: The number of packets transmitted with each Cycle ID value
- * Byte count: The total bytes transmitted with each Cycle ID value

The encapsulating node MUST maintain statistics for all four Cycle ID values (0, 1, 2, 3) simultaneously.

Flow ID	Current Cycle ID	Cyc 0 Stats	Cyc 1 Stats	Cyc 2 Stats	Cyc 3 Stats
<flow>	0 1 2 3	pkt/ byte	pkt/ byte	pkt/ byte	pkt/ byte

Figure 3: Per-Flow Statistics Structure

5.3. Statistics Reporting

The encapsulating node MUST report statistics with a two-interval delay relative to the current Cycle ID. This delay ensures that all packets from a given interval have been processed before reporting.

The reporting relationship is defined as follows:

Current Cycle ID	Report Statistics for Cycle ID
0	2
1	3
2	0
3	1

Table 2: Statistics Reporting Schedule

The reporting schedule MAY be expressed as:

$$\text{report_cycle_id} = (\text{current_cycle_id} + 2) \bmod 4$$

5.4. Keepalive Packet Generation

The encapsulating node MUST generate a keepalive packet when a Cycle ID transition occurs and no user data packets were transmitted during the ending interval.

The encapsulating node MUST track whether at least one user data packet has been transmitted during each measurement interval. The encapsulating node SHOULD implement this tracking using a per-flow, per-Cycle-ID flag.

Upon Cycle ID transition, if the tracking flag indicates no user data packets were transmitted during the ending interval, the encapsulating node MUST generate and transmit a keepalive packet with the following properties:

- * The Cycle ID field MUST be set to the ending interval's Cycle ID value
- * The K flag MUST be set to 1
- * The packet SHOULD use a cached packet template from the flow, if available
- * The packet MAY be a minimal probe packet if no template is available

Keepalive packets MUST NOT be counted in the packet statistics for packet loss calculation. Implementations MUST distinguish keepalive packets from user data packets in statistics counters.

5.5. Error Handling

If the encapsulating node is unable to insert the Cycle ID field due to insufficient header space, the node **MUST** forward the packet without IOAM information and **SHOULD** increment an error counter.

If the encapsulating node detects an invalid Cycle ID value in its local state (i.e., a value outside the range 0-3), the node **MUST** reset the Cycle ID to 0 and **SHOULD** log the error condition.

6. Transit Node Procedures

This section specifies the normative procedures for transit nodes.

6.1. Packet Processing

A transit node that supports packet-triggered reporting **MUST** perform the following operations for each received packet containing a Cycle ID field:

1. Parse the Cycle ID field from the IOAM header
2. Validate that the Cycle ID value is in the range 0-3
3. Update the statistics counter corresponding to the parsed Cycle ID value
4. Forward the packet according to normal forwarding procedures

Transit nodes **MUST NOT** modify the Cycle ID field or the K flag.

Transit nodes **SHOULD NOT** maintain timers for statistics reporting. Transit nodes **MAY** trigger local statistics reporting based on observed Cycle ID transitions, using the same procedures as decapsulating nodes.

6.2. Error Handling

If a transit node receives a packet with an invalid Cycle ID value (outside the range 0-3), the node **MUST** forward the packet without updating statistics and **SHOULD** increment an error counter.

If a transit node receives a packet with a malformed Cycle ID field (e.g., truncated header), the node **SHOULD** forward the packet according to local policy and **MUST** increment an error counter.

7. Decapsulating Node Procedures

This section specifies the normative procedures for decapsulating nodes.

7.1. State Maintenance

The decapsulating node MUST maintain the following per-flow state:

- * Last received Cycle ID: The Cycle ID value from the most recently received packet
- * Per-Cycle-ID statistics counters: Packet and byte counts for each of the four Cycle ID values
- * Initialization flag: Indicates whether the flow has received its first packet

Upon receiving the first packet for a flow, the decapsulating node MUST set the initialization flag and record the received Cycle ID as the last received Cycle ID. The decapsulating node MUST NOT trigger statistics reporting for the first received packet.

7.2. Report Triggering

For each received packet after initialization, the decapsulating node MUST compare the packet's Cycle ID with the last received Cycle ID.

If the Cycle ID values differ, the decapsulating node MUST:

1. Trigger statistics reporting for the interval that ended two cycles before the newly received Cycle ID, calculated as:
$$\text{report_cycle_id} = (\text{received_cycle_id} + 2) \bmod 4$$
2. Update the last received Cycle ID to the newly received value
3. Update the statistics counter for the newly received Cycle ID

If the Cycle ID values are equal, the decapsulating node MUST update only the statistics counter for the current Cycle ID.

7.3. Keepalive Packet Processing

When the decapsulating node receives a packet with the K flag set to 1, the node MUST:

- * Process the Cycle ID for report triggering as specified in Section 7.2

- * NOT include the keepalive packet in packet loss statistics
- * Optionally discard the packet after processing if no further forwarding is required

7.4. Error Handling

If the decapsulating node receives a packet with an invalid Cycle ID value, the node **MUST** discard the Cycle ID information from that packet, **MUST NOT** trigger statistics reporting based on that packet, and **SHOULD** increment an error counter.

If the decapsulating node detects a Cycle ID sequence gap (e.g., transition from 0 to 2, skipping 1), the node **SHOULD** log this condition and **MAY** trigger reporting for all skipped intervals. The specific behavior for sequence gaps is implementation-defined.

8. Cycle ID State Machine

The Cycle ID transitions through four states in a fixed sequence. The state machine is defined as follows:

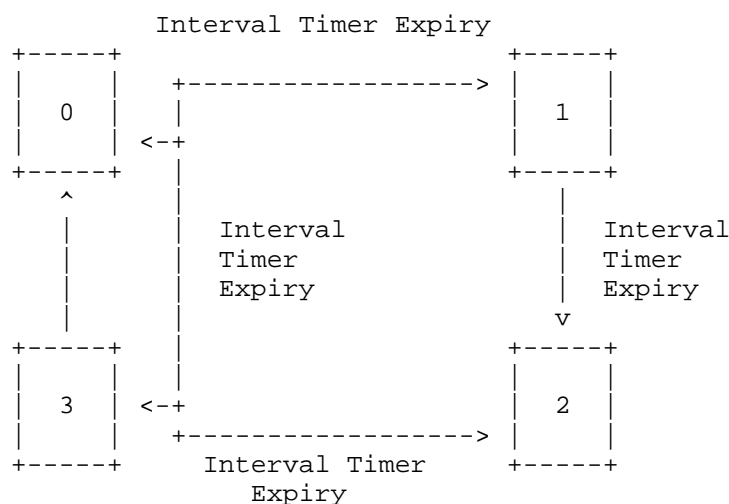


Figure 4: Cycle ID State Machine

The Cycle ID value **MUST** transition in the sequence: 0 -> 1 -> 2 -> 3 -> 0 (repeating). No other transition sequences are valid.

The transition from one Cycle ID value to the next **MUST** occur only at measurement interval boundaries as determined by the encapsulating node's local timer.

9. Manageability Considerations

9.1. Configuration Parameters

Implementations supporting this specification SHOULD expose the following configuration parameters:

- * Measurement interval duration (default: 1 second, range: 100ms - 60s)
- * Keepalive generation enable/disable (default: enabled)
- * Per-flow statistics reporting destination
- * Error counter thresholds for alarming

9.2. Monitoring and Statistics

Implementations SHOULD provide access to the following operational statistics:

- * Per-flow, per-Cycle-ID packet and byte counters
- * Keepalive packets generated (per flow)
- * Keepalive packets received (per flow)
- * Cycle ID errors detected
- * Cycle ID sequence gaps detected

10. Security Considerations

The security considerations of [RFC9197] apply to this extension. The following additional considerations are specific to the Cycle ID mechanism.

The Cycle ID field is a 2-bit value that indicates the measurement interval. An on-path attacker could modify the Cycle ID value, causing incorrect statistics reporting at the decapsulating node. Implementations operating outside a single administrative trust domain SHOULD protect IOAM data integrity using mechanisms such as those described in [RFC9197] Section 5.

The keepalive packet generation mechanism could be abused by an attacker who suppresses user data packets, causing the encapsulating node to generate keepalive packets continuously. Implementations SHOULD monitor keepalive generation rates and alert operators when abnormal patterns are detected.

Statistics counters maintained by transit and decapsulating nodes are local state and are not exposed in the data plane. Access to these counters through management interfaces SHOULD be protected by authentication and authorization mechanisms.

11. IANA Considerations

11.1. IOAM Option-Type Registry

This document requests IANA to allocate a new Option-Type from the "IOAM Option-Type" registry defined in [RFC9197].

Value	Description	Reference
TBD1	IOAM Cycle ID Option	[This Document]

Table 3: IOAM Option-Type Allocation Request

11.2. IOAM Cycle ID Flags Registry

This document requests IANA to create a new registry titled "IOAM Cycle ID Flags" under the "In-situ OAM (IOAM)" registry group.

The registry contains 1-bit flags. The initial contents of the registry are:

Bit Position	Name	Description	Reference
0	K	Keepalive Flag	[This Document]

Table 4: IOAM Cycle ID Flags Registry

Future assignments in this registry require Standards Action [RFC8126].

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, 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>>.
- [RFC9197] Brockners, F., Ed., Bhandari, S., Ed., and T. Mizrahi, Ed., "Data Fields for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9197, DOI 10.17487/RFC9197, May 2022, <<https://www.rfc-editor.org/info/rfc9197>>.

12.2. Informative References

- [RFC4303] Kent, S., "IP Encapsulating Security Payload (ESP)", RFC 4303, DOI 10.17487/RFC4303, December 2005, <<https://www.rfc-editor.org/info/rfc4303>>.
- [RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing an IANA Considerations Section in RFCs", BCP 26, RFC 8126, DOI 10.17487/RFC8126, June 2017, <<https://www.rfc-editor.org/info/rfc8126>>.
- [RFC9326] Song, H., Gafni, B., Brockners, F., Bhandari, S., and T. Mizrahi, "In Situ Operations, Administration, and Maintenance (IOAM) Direct Exporting", RFC 9326, DOI 10.17487/RFC9326, November 2022, <<https://www.rfc-editor.org/info/rfc9326>>.

Appendix A. Processing Examples

This appendix provides non-normative examples of packet-triggered reporting behavior.

A.1. Normal Operation

Consider a flow with continuous traffic. The encapsulating node operates with a 1-second measurement interval:

Time (s)	Cycle ID	Encap Action	Decap Action
-----	-----	-----	-----
0.0-1.0	0	Assign CID=0 Count packets	Receive CID=0 Count packets (Initialize)
1.0-2.0	1	Assign CID=1 Count packets	Receive CID=1 Trigger: report CID=3 (No data for CID=3) Count packets
2.0-3.0	2	Assign CID=2 Report CID=0 stats Count packets	Receive CID=2 Trigger: report CID=0 Count packets
3.0-4.0	3	Assign CID=3 Report CID=1 stats Count packets	Receive CID=3 Trigger: report CID=1 Count packets

A.2. Keepalive Generation

Consider a flow with intermittent traffic where no packets are transmitted during Cycle ID=1:

Time (s)	Cycle ID	Traffic	Encap Action
-----	-----	-----	-----
0.0-1.0	0	Present	Assign CID=0, count packets
1.0-2.0	1	None	Generate keepalive (CID=1, K=1)
2.0-3.0	2	Present	Assign CID=2, count packets

Acknowledgements

TBD

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