IPPM
Internet-Draft
Intended status: Standards Track
Expires: 26 October 2025

A. Clemm
Sympotech
L. Metzger
R. Bister
S. Dellsperger
Ostschweizer Fachhochschule - OST

24 April 2025

Aggregation Trace Option for In-situ Operations, Administration, and Maintenance (IOAM)
draft-cxx-ippm-ioamaggr-03

Abstract

The purpose of this memo is to describe a new option type for In-Situ Operations, Administration, and Maintenance (IOAM). This option type allows to aggregate IOAM data along a network path. Aggregates include functions such as the sum, average, minimum, or maximum of a given data parameter.

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 October 2025.

Copyright Notice

Table of Contents

1. Introduction

This memo proposes a new option type for In-Situ Operations, Administration, and Maintenance (IOAM) [RFC9197]. The IOAM Aggregate option type allows aggregating IOAM data along a network path. Aggregates include functions such as the sum, average, minimum, or maximum of a given data parameter.

Many applications interested in telemetry data across a path are not so much interested in each individual node's telemetry, but an aggregate to paint a more holistic picture. An example of an aggregate could be a sum (for example, the sum of packet dwell times experienced across a path), an average (for example, the average dwell time experienced across a path), or a minimum or maximum (for example, of the dwell time experienced on any hop across the path, along with the node ID where the extreme was experienced). Other applications include sustainable networking, where (for example) the carbon-intensity of a path as a whole needs to be determined as an input to applications that attempt to minimize pollution attributable to specific networking traffic.

The aggregation option type proposed in this memo addresses the needs of those applications. Rather than collecting individual IOAM data parameters at each node and exporting them for further processing, IOAM Aggregate allows preprocessing telemetry data into an aggregate as a packet traverses a path. Aggregating parameters along the path, instead of merely collecting them, offers the following advantages:

* It keeps the packet size constant. This avoids problems such as the possibility of packets exceeding their MTU and need for fragmentation and reassembly in case of longer data paths, or deteriorating packet delays as packets grow in size along a path.

* It reduces the volume of data to be exported.

* It obviates the need to correlate data exported from individual nodes as belonging to the same flow, when compared with processing of postcard telemetry data [RFC9326].

* It significantly reduces the amount of processing that needs to be done by applications, simplifying their development and deployment.

* It enables greater network intelligence, such as taking actions on aggregates when certain thresholds are exceeded.

Aggregating parameters does require a small amount of processing (such as, arithmetic operations to add to a sum, or a comparison operation to determine a minimum) at each hop, requiring some additional processing cycles. This is a small tradeoff to be aware of when choosing this option. We believe that this tradeoff will be acceptable in many implementations and deployment scenarios.

2. Background

[RFC9197] defines the scope of IOAM as well as the different IOAM nodes. The following section reiterates those roles and explains how they are applied in the context of IOAM Aggregation.

IOAM is focused on "limited domains", as defined in [RFC8799]. IOAM is not targeted for a deployment on the global Internet, which would incur additional considerations such as the crossing of Trust Boundaries, authentication of IOAM data, or the desire to obfuscate domain internals to outside parties. The part of the network that employs IOAM is referred to as the "IOAM-Domain".

An IOAM-Domain consists of "IOAM encapsulating nodes", "IOAM decapsulating nodes", and "IOAM transit nodes", as depicted in Figure 1.

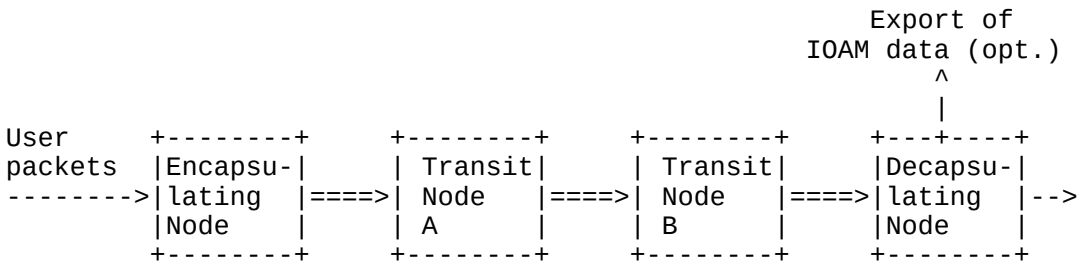

Figure 1: Roles of IOAM nodes

The role of these nodes is as follows:

* The Encapsulating Node originates the IOAM aggregation. It adds the IOAM Aggregation Option to the packet for which telemetry data is to be aggregated across the path and populates the fields with their initial values.

* The Transit Node is an IOAM-enabled node that aggregates the value of its own telemetry with the aggregate in the packet, updating the aggregation data as needed.

* The Decapsulating Node terminates the IOAM aggregation. It aggregates the value of its own telemetry with the aggregate in the packet and updates the aggregation data as needed. It subsequently exports the aggregated data, specifically, including the value of the aggregate itself as well as auxiliary data as applicable (e.g. node ID for min, max, and in case of errors).

3. IOAM Aggregation Option-Type

3.1. Overview

This section defines the data fields for the IOAM Aggregation Option Type format. Like other IOAM Aggregation Option Types, these data fields can be mapped into a number of transport protocols [RFC9378]. For example, transport over IPv6 [RFC8200] has been defined in [RFC9486].

The format of the IOAM Aggregation Option Type data fields is depicted in Figure 2.

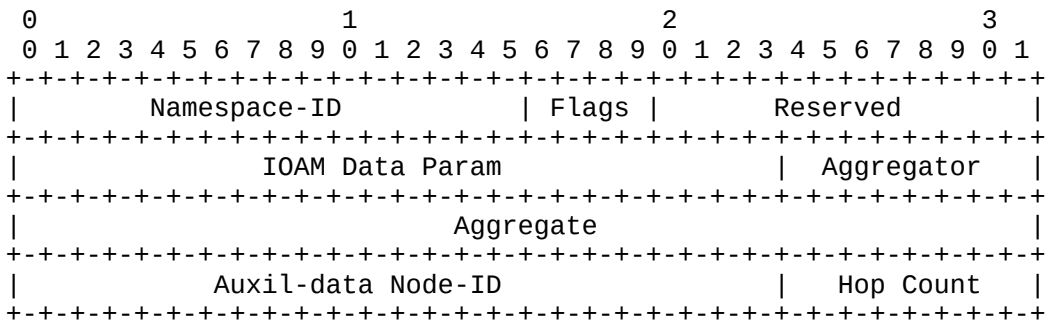

Figure 2: IOAM Aggregation Option Type Format

The total length of the IOAM Aggregation Option Type data fields is fixed at 16 octets (word-aligned). These 16 octets hold header information as well as aggregation data in the following fields:

* Namespace-ID: 16-bit identifier of an IOAM-Namespace, as defined in [RFC9197]. The Namespace-ID is populated by the encapsulating node and MUST NOT be changed by any of the intermediate nodes. The Namespace-ID value of 0x0000 is defined as the "Default-Namespace-ID" and MUST be known to all the nodes implementing IOAM. For any other Namespace-ID value that does not match any Namespace-ID the node is configured to operate on, the node MUST NOT change the contents of the IOAM-Data-Fields except for the Namespace Flag (see below).

* Flags: 4-bit field to indicate errors that were encountered when attempting to process the IOAM Aggregation Option along the path. Once a flag is set, no further aggregation occurs along the path. An intermediate node that encounters an error during processing of the IOAM Aggregation that prevents it from updating the aggregate as requested MUST set the corresponding flag to 1. In order to facilitate troubleshooting, it also MUST set the value of the Auxil-data Node-ID field to its own Node-ID. The encapsulating node MUST set the value of Flags to zero upon transmission. When an intermediate node encounters receives a packet in which any of the Flags are non-zero, the node MUST NOT perform further IOAM operations on that packet; instead, the IOAM data MUST be forwarded as-is unchanged. The following flags are defined:

  - Flag 1: Aggregator not supported

  - Flag 2: Unsupported IOAM data parameter

  - Flag 3: Unsupported Namespace

  - Flag 4: Any other error

* Reserved: An IOAM encapsulating node MUST set the value to zero upon transmission. IOAM transit nodes MUST ignore the received value.

* IOAM Data Param: This field identifies the data parameter that is to be aggregated across the nodes. It MUST be set by the IOAM encapsulating node. IOAM transit nodes MUST NOT change it. Contrary to IOAM Trace-Type in the pre-allocated and incremental trace option types, only a single parameter is aggregated at a time. Accordingly, the data parameter to be aggregated is not identified by a particular bit, but by a value.

* Aggregator: This 8-bit field identifies the aggregation function that is to be applied. Its value MUST be set by the IOAM encapsulating node. IOAM transit nodes MUST not change it. The following aggregators are defined:

  - Sum (value: 0b1)

  - Min (value: 0b10)

  - Max (value: 0b100)

  - Average (value: 0b1000)

* Aggregate: This 32-bit field contains the aggregated value. Its value is initialized by the encapsulating node,in general by simply recording the value of its data parameter that is to be aggregated. The field is updated by each subsequent node pre the requested aggregation, including IOAM transit nodes as well as the IOAM decapsulating node (prior to performing decapsulation).

* Auxil-data Node-ID: This 24-bit field contains a Node-ID. It MUST be set by the encapsulating node to its own Node ID. Subsequent nodes (IOAM transit nodes, as well as the IOAM decapsulating node prior to performing decapsulation) MUST update the value to their own Node-ID IF AND ONLY IF one of the following conditions hold, otherwise they MUST NOT change its value:

  - When a flag is set by the node (i.e., the first time any type of error is encountered along the path)

  - When the aggregator is one of Min or Max, and a new minimum respectively maximum is encountered. The value of the Auxil-data Node-ID field is hence used to record where the minimum respectively maximum value was first encountered. (When a node matches an existing minimum or maximum but does not beat it, the Node-ID is not updated.)

* Hop Count. This 8-bit fields contain a hop count to record the number of nodes along the path that successfully processed the IOAM Aggregation. The encapsulating node MUST set the value to 1, and each subsequent node (transit nodes, as well as the decapsulating node prior to performing decapsulation) MUST increment its value by 1. If the Hop Count at a node exceeds 255, that node MUST set the Hop Count to 0 and set Flag 4 ("any other error") to prevent further processing of the IOAM Aggregation.

3.2. Discussion

This section explains some design choices and points out items that may be subjected to further discussion.

* Single versus multiple parameters. The Aggregation Option Type allows to only aggregate one data parameter at a time. This allows to keep the format of the data structure simple and of fixed size. This facilitates processing. It also limits the number of processing cycles that need to be spent for aggregation at each node. An application seeking to perform multiple types of aggregation at a time will need to apply different types of aggregation for different packets.

* IOAM data parameter identification. Unlike other IOAM Option Types, data parameters are not represented by a bit position in a field but by a 24-bit identifier. This allows to support a greater number of parameters. In order to facilitate compatibility, initially only identifiers SHOULD be used that utilize bits 12 through 22, with other bits set to 0. The assignment of IOAM data parameter identifiers is at this point up to the network operator, with IOAM data parameters being specific to an IOAM name space. It is conceivable that a global namespace and a corresponding IANA registry for IOAM data parameters would be introduced at a later point in time.

* Average aggregator. An average can be easily derived from dividing a sum obtained across all nodes by the hop count. Avoiding division operations along the path can save considerable processing cycles. It is FFS if the average aggregator is really required.

* Simultaneous use with other IOAM Option Types. There are use cases conceivable that would benefit from also adding a trace of which nodes were actually traversed on the path. The possibility to do so is already provided with other IOAM Option Types and does not need to be added here. In order to use multiple IOAM Option Types simultaneously, applications can use one of several alternatives. In one alternative, multiple IOAM Option Types with their corresponding data structures are simultaneously used in the same packet. In another alternative, different packets of the same flow are each send with a different IOAM Option Type, a form of sampling which however provides no absolute guarantees of path congruency (i.e., different packets traversing the exact same path).

4. Use Cases

The following describes a number of use cases in which the IOAM Aggregation Trace Option can be applied in order to illustrate its use. Many more such use cases can be identified.

* Determination of energy-related metrics along a path. Energy metrics related to networking paths have been proposed as a way to optimize routing decisions for more sustainable networking (e.g. [I-D.petra-path-energy-api]). IOAM Aggregation Trace Option allows to easily determine such metrics by aggregating the contributions of nodes along the path without need for post-processing or coordination by controllers. An implementation of this has been successfully demonstrated [NetSoft2024].

* Identification of outliers to aid in diagnosing and troubleshooting of performance issues in the delivery of service instances. One use case for IOAM concerns collecting parameters such as the dwell time of packets at each node to aid in diagnosing latency issues. Of particular interest is the determination of the respective outlier on the path in order to identify if any node in particular might be the culprit. Using the IOAM Aggregation Trace Option allows delivering data about the particular outlier without the need to collect and then process a larger number of data items from each node across lengthy paths, making diagnosis a lot more efficient.

* Aggregation of packet dwell times across networking paths as a way to optimize networks to better meet service level objectives related to latency. Providing networking services that meet latency-related service level objectives is a well-documented problem and focus of the networking community. Some approaches focus on the dwell time of packets as one component of their solution, i.e. the time spent by routers in processing packets

[I-D.eckert-detnet-glbf]. Using the IOAM Aggregation Trace Option allows to directly act on an aggregate, i.e. the data of interest, without having to go through additional steps to first collect and process large numbers of individual raw data items.

5. Security Considerations

A malicious node along the path could attempt to forge the aggregate, resulting in a wrong aggregate to be reported. This might mislead applications. Likewise, a malicious node along the path could set a flag to trick other nodes not to process the aggregate any further, or clear a flag to make a distorted result appear legitimate. To avoid this, network operators need to ensure that their network nodes can be trusted and are not compromised.

6. IANA Considerations

IANA requests are TBD. Future versions of this document may request the establishment of a registry for Aggregators as well as for IOAM Data Parameters.

7. Contributors

* Reto Furrer, OST

8. References

8.1. Normative References

[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/rfc/rfc9197>.

8.2. Informative References

[I-D.eckert-detnet-glbf] Eckert, T. T., Clemm, A., Bryant, S., and S. Hommes, "Deterministic Networking (DetNet) Data Plane - guaranteed Latency Based Forwarding (gLBF) for bounded latency with low jitter and asynchronous forwarding in Deterministic Networks", Work in Progress, Internet-Draft, draft-eckert-detnet-glbf-04, 3 March 2025, <https://datatracker.ietf.org/doc/html/draft-eckert-detnet-glbf-04>.

[I-D.petra-path-energy-api]
Rodriguez-Natal, A., Contreras, L. M., Muniz, A., Palmero, M., Munoz, F., and J. Lindblad, "Path Energy Traffic Ratio API (PETRA)", Work in Progress, Internet-Draft, draft-petra-path-energy-api-02, 8 July 2024, <https://datatracker.ietf.org/doc/html/draft-petra-path-energy-api-02>.

[NetSoft2024]
Bister, R., Clemm, A., Dellsperger, S., and R. Furrer, "Towards Sustainable Networking: Unveiling Energy Efficiency Through Hop and Path Efficiency Indicators in Computer Networks.", IEEE 10th International Conference on Network Softwarization (NetSoft) , June 2024.

[RFC8200] Deering, S. and R. Hinden, "Internet Protocol, Version 6 (IPv6) Specification", STD 86, RFC 8200, DOI 10.17487/RFC8200, July 2017, <https://www.rfc-editor.org/rfc/rfc8200>.

[RFC8799] Carpenter, B. and B. Liu, "Limited Domains and Internet Protocols", RFC 8799, DOI 10.17487/RFC8799, July 2020, <https://www.rfc-editor.org/rfc/rfc8799>.

[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/rfc/rfc9326>.

[RFC9378] Brockners, F., Ed., Bhandari, S., Ed., Bernier, D., and T. Mizrahi, Ed., "In Situ Operations, Administration, and Maintenance (IOAM) Deployment", RFC 9378, DOI 10.17487/RFC9378, April 2023, <https://www.rfc-editor.org/rfc/rfc9378>.

[RFC9486] Bhandari, S., Ed. and F. Brockners, Ed., "IPv6 Options for In Situ Operations, Administration, and Maintenance (IOAM)", RFC 9486, DOI 10.17487/RFC9486, September 2023, <https://www.rfc-editor.org/rfc/rfc9486>.

Authors' Addresses

Alexander Clemm
Sympotech
Email: ludwig@clemm.org

Laurent Metzger
Ostschweizer Fachhochschule - OST
Email: laurent.metzger@ost.ch

Ramon Bister
Ostschweizer Fachhochschule - OST
Email: ramon.bister@ost.ch

Severin Dellsperger
Ostschweizer Fachhochschule - OST
Email: severin.dellsperger@ost.ch