

WG Working Group	A. Mayer
Internet-Draft	Univ. of Rome Tor Vergata / CNIT / Common Net
Intended status: Experimental	S. Salsano
Expires: 20 September 2026	Univ. of Rome Tor Vergata / CNIT
	19 March 2026

Global Opaque Block for IOAM Pre-allocated Trace Option  
draft-mayer-ioam-gob-01

## Abstract

Extensible metadata carried within the IOAM Pre-allocated Trace Option (PTO) can support packet-level, flow-level, or path-level processing beyond per-node trace data. This document defines the Global Opaque Block (GOB), an extension to the IOAM PTO that introduces a single pre-allocated global metadata region placed between the PTO fixed header and the node data list. The GOB carries an explicit length and schema identifier, preserves the pre-allocated PTO processing model, and can be used to transport Extensible In-band Processing (EIP) Information Elements or other structured metadata formats.

## About This Document

This note is to be removed before publishing as an RFC.

The latest revision of this draft can be found at <https://eip-home.github.io/ioam-gob/draft-mayer-ioam-gob.html>. Status information for this document may be found at <https://datatracker.ietf.org/doc/draft-mayer-ioam-gob/>.

Discussion of this document takes place on the EIP SIG mailing list (<mailto:eip@cnit.it>), which is archived at <http://postino.cnit.it/cgi-bin/mailman/private/eip/>.

Source for this draft and an issue tracker can be found at <https://github.com/eip-home/ioam-gob>.

## 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 20 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.

## Table of Contents

1. Introduction . . . . .	3
2. Conventions and Definitions . . . . .	3
3. Design Overview . . . . .	4
4. GOB Format . . . . .	5
4.1. PTO Layout with GOB . . . . .	5
4.2. GOB Header Fields . . . . .	5
4.3. Byte-Level Layout . . . . .	6
5. Length and Alignment Rules . . . . .	8
5.1. GOB Length . . . . .	8
5.2. Internal PTO Offsets . . . . .	8
5.3. IOAM Option Length and Hop-by-Hop Header Length . . . . .	9
5.4. Alignment Semantics . . . . .	9
5.5. Preservation of PTO Semantics . . . . .	10
6. Processing Semantics . . . . .	10
6.1. Ingress Node Behavior . . . . .	10
6.2. Transit Node Behavior . . . . .	11
6.3. Egress or Collector Behavior . . . . .	11
7. Relation to PTO Semantics . . . . .	11
8. Schema Semantics and Extensibility . . . . .	12
9. Integration with EIP . . . . .	12
10. Security Considerations . . . . .	13
11. IANA Considerations . . . . .	13
12. References . . . . .	14
12.1. Normative References . . . . .	14
12.2. Informative References . . . . .	15
Acknowledgments . . . . .	15
Authors' Addresses . . . . .	15

## 1. Introduction

The IOAM Pre-allocated Trace Option (PTO) defined in [RFC9197] and [RFC9486] provides a structured way to carry in-band operational and telemetry information along a packet path. In the PTO model, the encapsulating node pre-allocates space for node data that can be filled by IOAM-capable nodes along the path.

This document defines an extension to the IOAM PTO called the Global Opaque Block (GOB). The GOB is a global metadata region that appears once in the PTO, immediately after the PTO fixed header and before the per-node node data list. The GOB is identified by a presence bit in the PTO header and includes its own length and schema fields.

The GOB enables the carriage of structured global metadata within the IOAM PTO, while preserving the existing PTO processing model for per-node trace data. In particular, the GOB can carry Extensible In-band Processing (EIP) Information Elements or other schema-driven metadata formats, without modifying the semantics of the node data list.

The GOB is pre-allocated by the encapsulating node. Transit nodes MAY read or update the GOB payload, subject to the bounds established at encapsulation time, but they MUST NOT alter the overall GOB size or the layout of the enclosing PTO option.

This document specifies the GOB format, length and alignment rules, and processing behavior for encapsulating, transit, and egress or collector nodes.

An implementation and experimental evaluation of the proposed GOB approach within the Linux IOAM framework are described in [mayer26noms].

## 2. Conventions and Definitions

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.

The following terms are used in this document:

GOB: Global Opaque Block. A structured global metadata region inserted into the IOAM PTO.

GOB Header: The 4-octet fixed header of the GOB, containing the GOB Len and Schema fields.

**GOB Payload:** The payload that follows the GOB Header. Its size is determined by the GOB Len field.

**PTO:** The IOAM Pre-allocated Trace Option defined in [RFC9197] and [RFC9486].

**Node Data List:** The sequence of pre-allocated per-node data slots defined by the PTO format.

**Ingress Node:** The encapsulating node that inserts the IOAM PTO and, when present, allocates and initializes the GOB.

**Transit Node:** An IOAM-capable node along the path that may process the PTO and may read or update the GOB payload.

**Egress or Collector Node:** A node that removes, terminates, exports, or otherwise interprets the IOAM information carried in the packet.

**Schema:** A 24-bit identifier that specifies how the GOB Payload is to be interpreted.

### 3. Design Overview

The GOB extends the IOAM PTO with a global metadata area that is logically distinct from the per-node node data list. It is intended for metadata that applies to the packet, flow, or path as a whole, rather than to a specific transit node.

The GOB appears at most once in a PTO option. When present, it is placed immediately after the PTO fixed header and before the start of the node data list. The GOB consists of a fixed 4-octet header followed by a variable-length payload.

The GOB preserves the pre-allocated nature of the PTO model. The ingress node allocates the GOB space at encapsulation time and determines its size, its schema, and the overall PTO layout. Transit nodes MAY modify the contents of the GOB payload, but MUST NOT modify the GOB Len field, the Schema field, or any PTO fixed-header fields that precede the GOB.

The GOB is intended to support structured metadata formats. One relevant use case is the carriage of Extensible In-band Processing (EIP) Information Elements, but the mechanism is not restricted to EIP and may be used with other schema-driven payload formats.

The addition of the GOB does not change the semantics of the per-node node data list. The GOB complements the PTO by adding a global opaque area while leaving the existing per-node trace structure intact.

## 4. GOB Format

### 4.1. PTO Layout with GOB

When the Global Opaque Block (GOB) is present, the IOAM Pre-allocated Trace Option (PTO) is extended by inserting a single GOB between the PTO fixed header and the Node Data List.

The resulting layout is:

1. PTO fixed header
2. GOB Header
3. GOB Payload
4. Node Data List
5. Tail padding, if required to satisfy alignment of the enclosing Hop-by-Hop Options header

The GOB is indicated by the G bit in the PTO fixed header. If the G bit is clear, no GOB is present and PTO processing follows the base PTO format.

The GOB appears at most once in a PTO option.

### 4.2. GOB Header Fields

The GOB Header is 4 octets long and contains the following fields:

GOB Len: An 8-bit field that specifies the size of the GOB Payload in units of 4 octets.

Schema: A 24-bit field that identifies the format of the GOB Payload.

The total size of the GOB is therefore:

$\text{GOB\_PLEN} = \text{GOB\_Len} * 4$   
 $\text{GOB\_SIZE} = 4 + \text{GOB\_PLEN}$

where the first 4 octets correspond to the GOB Header itself.

The Schema field allows the GOB Payload to be interpreted according to a structured format. One important use case is the carriage of Extensible In-band Processing (EIP) Information Elements, but the GOB is not limited to EIP and may carry other schema-driven metadata formats.

### 4.3. Byte-Level Layout

The following figure shows the byte-level layout of the IOAM PTO with the GOB present.

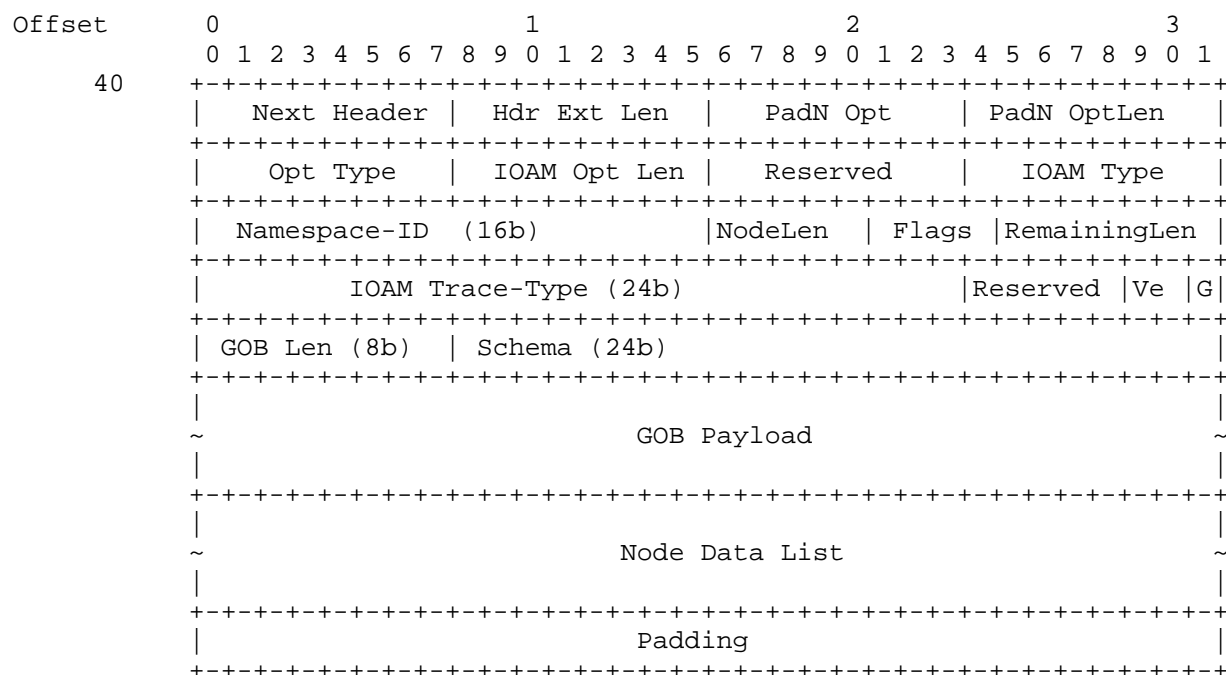


Figure 1: IOAM PTO with Global Opaque Block (GOB)

More explicitly, the internal structure is:

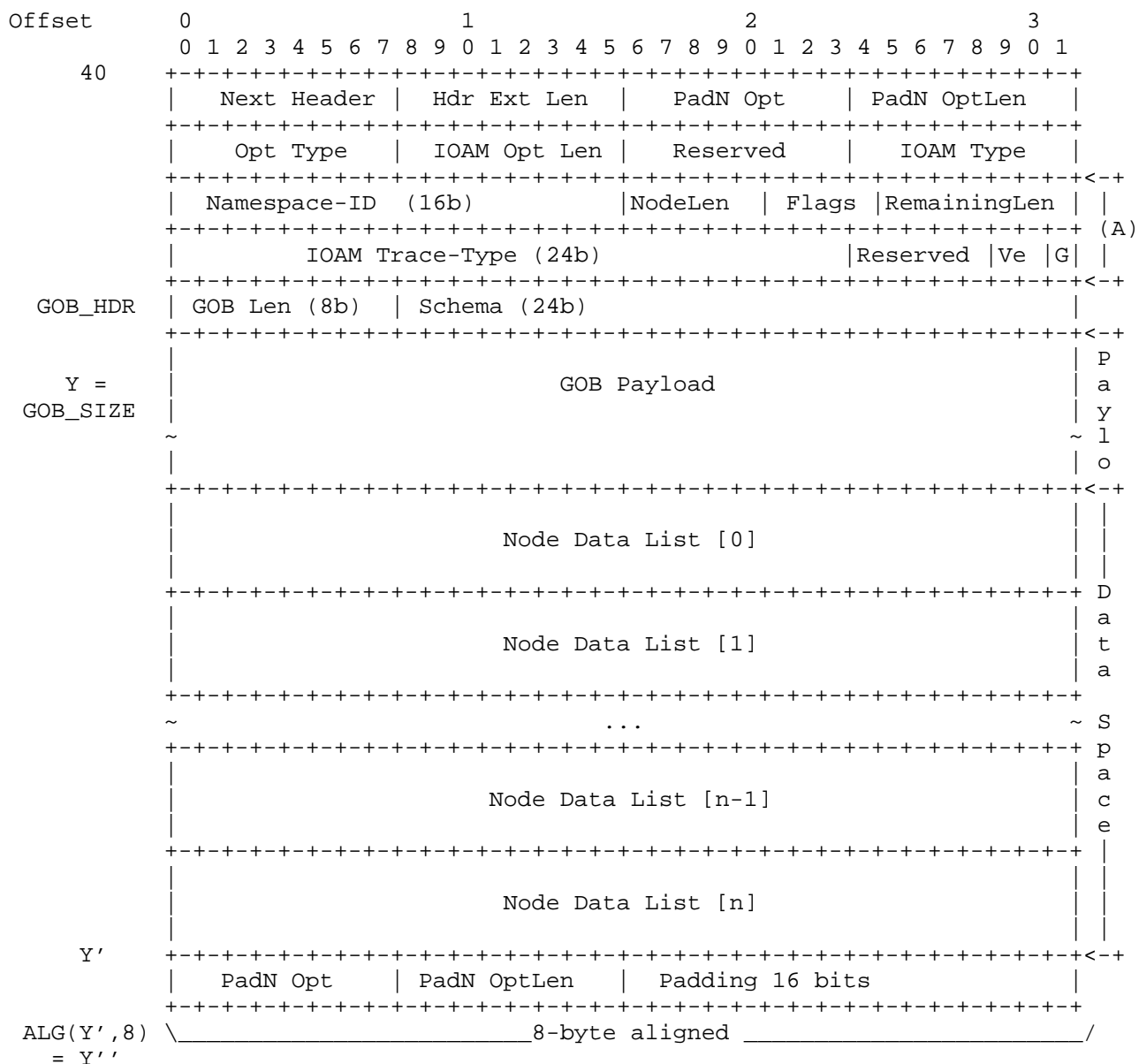


Figure 2: Detailed PTO layout with GOB, Node Data List, and tail padding

The GOB Header is placed immediately after the PTO fixed header. The GOB Payload immediately follows the GOB Header. The Node Data List begins immediately after the GOB Payload. Any final tail padding is added only to satisfy alignment requirements of the enclosing Hop-by-Hop Options header.

## 5. Length and Alignment Rules

This section defines the length relations used when the GOB is present in the IOAM PTO.

### 5.1. GOB Length

The GOB Payload length is determined by the GOB Len field:

$$\text{GOB\_PLEN} = \text{GOB\_Len} * 4$$

The total GOB size, including the fixed 4-octet GOB Header, is:

$$\text{GOB\_SIZE} = 4 + \text{GOB\_PLEN}$$

The GOB Len field is set by the ingress node at encapsulation time.

Transit nodes MAY modify the content of the GOB Payload, but MUST NOT modify the GOB Len field and therefore MUST NOT alter the total GOB size.

### 5.2. Internal PTO Offsets

Let:

- \* Y be the offset immediately following the GOB Payload;
- \* Y' be the offset immediately following the Node Data List;
- \* Y'' be the IOAM option length rounded up to the next multiple of 8 octets.

Then:

$$\begin{aligned} Y &= (\text{sizeof}(\text{IOAM\_TLV}) - 2) + \text{sizeof}(\text{IOAM\_PTO\_HDR}) + \text{GOB\_SIZE} \\ Y' &= Y + \text{RemainingLen} * 4 \\ Y'' &= \text{ALIGN}(Y', 8) \end{aligned}$$

In these expressions:

- \*  $\text{sizeof}(\text{IOAM\_TLV}) - 2$  excludes the Option Type and Option Length octets already accounted for in the option layout;



- \* `sizeof(IOAM_PTO_HDR)` is the fixed PTO header size;
- \* `RemainingLen` retains its PTO meaning and determines the amount of pre-allocated space associated with the Node Data List.

The Node Data List therefore starts immediately after the GOB and spans `RemainingLen * 4` octets.

### 5.3. IOAM Option Length and Hop-by-Hop Header Length

The IOAM Option Length field MUST be set to `Y''`, that is, to the total internal length of the IOAM option after rounding up to the required alignment.

The enclosing Hop-by-Hop Options header length is derived from the total size of the header, including:

- \* the 2-octet Next Header and Hdr Ext Len fields;
- \* any PadN option used to align the start of the IOAM option;
- \* the IOAM option itself, including any final tail padding.

Let `HBH_LEN` denote the total Hop-by-Hop Options header size in octets. Then:

$$\text{HBH\_LEN} = \text{ALIGN}(4 + \text{sizeof}(\text{IOAM\_PTO\_HDR}) + \text{GOB\_SIZE} + \text{RemainingLen} * 4, 8)$$

where the leading 4 octets account for:

- \* 2 octets for Next Header and Hdr Ext Len;
- \* 2 octets for the PadN option that aligns the IOAM option start.

The IPv6 Hdr Ext Len field is encoded as:

$$\text{Hdr Ext Len} = \text{HBH\_LEN} / 8 - 1$$

### 5.4. Alignment Semantics

As specified in [RFC8200], the Hop-by-Hop Options header is encoded so that its total length is a multiple of 8 octets, and the Hdr Ext Len field carries that length in 8-octet units, not including the first 8 octets.

The final tail padding used to align the enclosing Hop-by-Hop Options header to an 8-octet boundary is not part of the GOB and is not part of the Node Data List.

More specifically:

- \* Y identifies the end of the GOB;
- \* Y' identifies the end of the Node Data List;
- \* Y'' identifies the aligned end of the IOAM option.

The difference between Y' and Y'' is due only to final alignment padding and does not carry GOB or node-data semantics.

### 5.5. Preservation of PTO Semantics

The presence of the GOB does not alter the semantics of RemainingLen or the pre-allocated nature of the PTO.

The ingress node determines, at encapsulation time:

- \* the GOB size;
- \* the start of the Node Data List;
- \* the size of the Node Data List;
- \* the final aligned IOAM option length.

Transit nodes operate only within the pre-allocated bounds of the GOB Payload and the Node Data List, and MUST NOT change the overall layout of the PTO option.

In this way, the pre-allocated model of the PTO is preserved while extending the option with a structured global metadata region.

## 6. Processing Semantics

### 6.1. Ingress Node Behavior

An ingress node that inserts a GOB into the PTO:

- \* MUST set the G bit to 1;
- \* MUST allocate the full GOB space at encapsulation time;
- \* MUST set the GOB Len field;

- \* MUST set the Schema field;
- \* MUST determine the final PTO layout, including the GOB, the Node Data List, and any required tail padding.

The ingress node MAY initialize the GOB Payload with structured metadata, zero values, or other schema-defined initial content.

## 6.2. Transit Node Behavior

A transit node that processes a PTO containing a GOB:

- \* MAY read the GOB Payload;
- \* MAY update fields within the GOB Payload, subject to the schema in use;
- \* MUST NOT modify the GOB Len field;
- \* MUST NOT modify the Schema field;
- \* MUST NOT modify PTO fixed-header fields that precede the GOB;
- \* MUST NOT change the overall size or layout of the PTO option.

A transit node MUST confine any GOB update to the pre-allocated GOB Payload bounds established by the ingress node.

## 6.3. Egress or Collector Behavior

An egress or collector node that processes a PTO containing a GOB:

- \* MAY parse the GOB Payload according to the Schema field;
- \* MAY export the GOB content as structured metadata or as raw bytes;
- \* MAY ignore the GOB Payload if the Schema value is unknown or not supported.

If the Schema value is unknown, the node MUST NOT infer semantics beyond the presence and length of the GOB.

## 7. Relation to PTO Semantics

The GOB preserves the pre-allocated semantics of the IOAM PTO.

The ingress node continues to determine the overall PTO allocation at encapsulation time. This includes the size of the node data list and, when present, the size of the GOB. Transit nodes operate only within the pre-allocated space and do not change the overall layout of the PTO.

The Node Data List retains its existing semantics and processing model. The GOB does not redefine the per-node trace data structure and does not replace the node data list. Instead, it introduces a distinct global metadata region placed before the node data list.

In this sense, the pre-allocated nature of the PTO applies both to the per-node trace space and to the GOB. The GOB extends the PTO with a global opaque area while preserving the original PTO processing model.

## 8. Schema Semantics and Extensibility

The Schema field identifies the format of the GOB Payload.

A Schema value may identify:

- \* a sequence of EIP Information Elements;
- \* another structured metadata format defined by a specification;
- \* a locally defined or domain-specific format.

This document defines the GOB container, but does not define the internal syntax of individual schema-specific payloads. The semantics of the GOB Payload therefore depend on the specification associated with the Schema value in use.

Future specifications may define registries, allocation policies, or schema-specific processing rules for GOB Payload formats.

A Schema value of 0 indicates a locally defined or domain-specific format. Future specifications that define registries or allocation policies for Schema values MUST preserve this reservation.

## 9. Integration with EIP

One important use of the GOB is the carriage of Extensible In-band Processing (EIP) Information Elements.

The GOB provides a natural container for EIP within the IOAM PTO because it offers a single global metadata region with explicit length and schema identification, distinct from the per-node trace data. This makes it suitable for metadata that is flow-level, packet-level, or path-level rather than node-local.

Earlier work explored the integration of EIP into the IOAM framework through a different approach based on IOAM data fields [salsano25cscn]. The GOB-based approach defined in this document provides an alternative integration model based on a pre-allocated global metadata region within the PTO.

The GOB is a general container for structured metadata. It can carry EIP Information Elements, but it is not limited to EIP.

The GOB-based integration has also been described and experimentally evaluated in [mayer26noms].

## 10. Security Considerations

The GOB introduces a global metadata region that may be read or modified by on-path nodes. Unauthorized access to, or modification of, the GOB Payload can affect telemetry interpretation, service behavior, or policy enforcement.

For this reason, deployments of the GOB SHOULD be limited to controlled or trusted domains, consistent with the limited-domain assumptions often applied to IOAM deployments.

The security implications of the GOB Payload also depend on the Schema in use. Some schemas may carry purely observational metadata, while others may influence packet treatment or convey more sensitive information. Schema-specific specifications SHOULD describe any additional confidentiality, integrity, or authorization requirements.

## 11. IANA Considerations

This document defines the Global Opaque Block (GOB) as an extension to the IOAM Pre-allocated Trace Option.

Two possible standardization approaches can be considered.

A first approach is to reuse the existing codepoint of the IOAM Pre-allocated Trace Option and extend its format by defining the G bit and the associated GOB fields. In general, such an approach would be questionable, because it introduces a format extension that is not backward compatible with implementations that interpret the PTO according to the current specification and do not support the GOB extension.

However, IOAM is typically intended for deployment in limited domains, under the operational control of a single administrative entity. In such environments, the operator can control software versions and protocol support across the involved devices. For this reason, a non-backward-compatible extension of the PTO format may still be a viable standardization choice in this specific context, provided that the deployment domain is managed in a coordinated manner.

A second approach is to define a new codepoint for a PTO variant that supports the Global Opaque Block. In this case, the presence or absence of the GOB can still be indicated by the G bit, while the use of a distinct codepoint provides explicit protocol separation from the current PTO format and avoids ambiguity for non-updated implementations.

This version of the document does not request IANA actions. Future versions may request updates to the relevant IOAM or IPv6 registries, depending on which standardization approach is selected.

## 12. References

### 12.1. Normative References

[IANA-ioam-types]

IANA, "IOAM Data Field Types", n.d.,  
<<https://www.iana.org/assignments/ioam/ioam.xhtml#data-field-types>>.

[IANA-ipv6-parameters]

IANA, "Internet Protocol Version 6 (IPv6) Parameters", n.d., <<https://www.iana.org/assignments/ipv6-parameters/ipv6-parameters.xhtml>>.

[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/rfc/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/rfc/rfc8174>>.
- [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>>.
- [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>>.
- [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>>.

## 12.2. Informative References

### [draft-eip-arch]

Salsano, S., ElBakoury, H., and D. R. Lopez, "Extensible In-band Processing (EIP) Architecture and Framework", Internet-Draft draft-eip-arch, 2026, <<https://datatracker.ietf.org/doc/draft-eip-arch/>>.

### [mayer26noms]

Mayer, A., Salsano, S., Sidoretti, G., Loreti, P., Bracciale, L., ElBakoury, H., and D. R. Lopez, "Integrating the Global Opaque Block into the IOAM Framework: A Unified Approach to In-Packet Telemetry and Metadata", IEEE/IFIP NOMS 2026, 2026.

### [salsano25cscn]

Salsano, S., Mayer, A., Sidoretti, G., Loreti, P., Bracciale, L., ElBakoury, H., and D. R. Lopez, "Integrating Extensible In-Band Processing (EIP) into the IOAM Framework: A Unified Approach to In-Packet Telemetry and Metadata", IEEE CSCN 2025 Conference, 2025.

## Acknowledgments

TBD.

## Authors' Addresses

Andrea Mayer  
Univ. of Rome Tor Vergata / CNIT / Common Net  
Email: andrea.mayer@uniroma2.it

Stefano Salsano  
Univ. of Rome Tor Vergata / CNIT  
Email: stefano.salsano@uniroma2.it