Internet Area Working Group
Internet-Draft
Intended status: Standards Track
Expires: 19 February 2026

J. Rajamanickam, Ed.
D. Dukes
M. Sankaranarayanan
Cisco Systems, Inc.

14 October 2025

ICMP extension to include underlay information
draft-jags-intarea-icmp-ext-underlay-info-03

Abstract

Network operators managing overlay networks require visibility into underlay network hops during traceroute operations from overlay endpoints. This document defines an ICMP extension object, the Underlay Information Object (UIO), which allows underlay head-end nodes to encapsulate underlay error information within ICMP error messages. This mechanism provides overlay operators with crucial visibility into underlay network paths for troubleshooting.

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 19 February 2026.

Table of Contents

1. Introduction

The mechanism for ICMP messages to carry additional information is defined in [RFC4884]. ICMP message extensions that enable ICMP messages to carry additional information about the system where an error occurred are defined in [RFC5837], [RFC8335], and [RFC8883]. These extensions transmit enhanced diagnostic information to the source node.

Network operators who manage both overlay and underlay networks, such as those operating VPN segments connected through an SRv6 core network, require the ability to trace paths through the underlay infrastructure. Currently, when performing traceroute operations from an overlay endpoint, operators lack visibility into the underlay path and cannot identify the specific underlay node where a failure occurred. For instance, imagine a VPN service (overlay) running over an SRv6 network (underlay). If a packet gets dropped within the SRv6 network, the VPN operator currently has no direct way to pinpoint the exact underlay node causing the issue.

The Underlay Information Object (UIO) defined in this document addresses this operational requirement by enabling underlay head-end nodes to include underlay-specific diagnostic information in ICMP error messages sent to overlay endpoints, thereby providing crucial visibility for troubleshooting.

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.

This document uses the following terms:

Overlay Network: A virtual network built on top of an existing underlying network infrastructure, often providing services like VPNs or tunnels.

Underlay Network: The physical or logical network infrastructure over which an overlay network operates, responsible for forwarding packets between overlay endpoints.

Overlay Endpoint: A device or system that terminates an overlay network segment and originates or receives traffic for the overlay.

Underlay Head-End Node: The node in the underlay network responsible for encapsulating overlay traffic and often the first point of contact for an overlay packet entering the underlay.

3. Underlay Information Object

This section defines a new ICMP extension object called Underlay Information Object (UIO) that is encoded as part of ICMP extension message. A new Class-Num value TBA (To Be Assigned) is assigned to identify the UIO. As per [RFC4884], this object MAY be appended to one of the following ICMP messages:

ICMPv4 Time Exceeded

ICMPv4 Destination Unreachable

ICMPv4 Parameter Problem

ICMPv6 Time Exceeded

ICMPv6 Destination Unreachable

3.1. UIO Object Format

The UIO ICMP extension object has the following format:

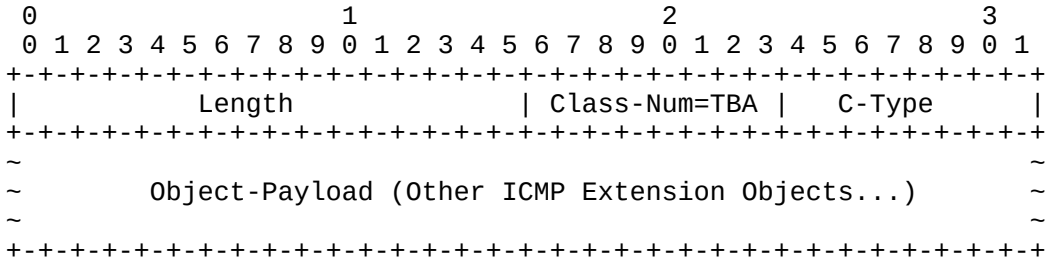


Figure 1: Underlay Information Object Format

Length (16 bits):

The length of this object, measured in octets, including the object header and object payload. The length MUST be a multiple of 4 octets and MUST be at least 8 octets.

Class-Num (8 bits):

The ICMP extension object class number that identifies this as a UIO object. IANA is requested to assign a value from the "ICMP Extension Object Classes and Class Sub-types" registry (see Section 5).

C-Type (8 bits):

The object sub-type. This document defines C-Type value 0. Additional C-Type values may be defined in future documents. Implementations MUST set this field to 0 and SHOULD ignore the value upon receipt.

Object-Payload (variable length):

Contains one or more ICMP Extension Objects that provide information about underlay nodes. The payload MUST contain at least one ICMP extension object. Each encapsulated ICMP extension object MUST be formatted according to [RFC4884] and the specifications for that particular object class.

This ICMP extension object acts as an envelope to carry other ICMP extension objects related to the underlay. Primarily, the UIO ICMP extension object is encoded in the ICMP extension message by the underlay head-end when it receives an ICMP error message from one of its intermediate nodes.

This UIO ICMP extension object can encapsulate one or more relevant ICMP extension objects that are related to the underlay node. When the underlay head-end encodes its ICMP extension object, the first object MUST contain the ICMP extension object that carries IP address or the hostname of the node where the initial ICMP error was generated. The ICMP extension objects encoded within the UIO ICMP extension objects can belong to any address family, irrespective of the address family of the source node that decapsulates the UIO ICMP extension objects, as opposed to what is stated in [RFC5837] Section 4.2.

If the node decoding the ICMP extension header does not recognize the UIO ICMP extension object, it SHOULD ignore this object and continue processing the other objects.

3.2. Underlay Information Object Encoding Process

When an underlay head-end node receives an ICMP error message from an underlay node and needs to forward information about this error to an overlay endpoint, it follows this process:

1. The underlay head-end node constructs an ICMP error message destined for the overlay endpoint.

2. The node appends a UIO ICMP extension object to this ICMP error message according to the procedures defined in [RFC4884].

3. Within the UIO object payload, the node includes one or more ICMP extension objects that carry information about the underlay node where the original error occurred.

4. The first ICMP extension object within the UIO payload MUST contain addressing information (e.g., using the Interface Information Object defined in [RFC5837]) that identifies the underlay node that generated the original error. This ensures that the most critical diagnostic information for pinpointing the failure source is immediately available.

5. Additional ICMP extension objects MAY be included to provide supplementary diagnostic information about the underlay path.

6. The encapsulated ICMP extension objects within the UIO may belong to any address family, regardless of the address family used between the underlay head-end and the overlay endpoint.

7. The total length of the ICMP message, including all extensions, MUST NOT exceed 576 octets for IPv4 or 1280 octets for IPv6 (the minimum reassembly buffer sizes defined in [RFC791] and [RFC8200], respectively).

Implementations SHOULD provide configuration options to control which underlay information is included in UIO objects, considering security and privacy implications discussed in Section 4.

4. Security Considerations

The UIO extension introduces several security considerations that implementations and operators must address:

4.1. Information Disclosure

The UIO extension reveals information about the underlay network topology and addressing to overlay endpoints. In many deployments, the overlay and underlay networks are operated by different administrative entities, and underlay topology information may be considered sensitive.

Implementations MUST provide configuration options to control the generation of UIO extensions. The default configuration MUST disable UIO generation. Operators SHOULD enable UIO only for authenticated and authorized overlay endpoints or networks. The specific mechanisms for such authentication and authorization are outside the scope of this document but are crucial for secure deployment.

4.2. Privacy Considerations

Underlay information may reveal details about network architecture, capacity, and routing that could be exploited for reconnaissance or targeted attacks. Operators SHOULD carefully consider which underlay information to expose through UIO extensions.

4.3. Message Size and Amplification

Including UIO extensions increases ICMP message size. Implementations MUST enforce the message size limits specified in Section 3.2 to prevent fragmentation issues and potential amplification attacks.

4.4. Spoofing and Forgery

As with all ICMP messages, UIO extensions are subject to spoofing attacks. The authenticity and integrity of UIO information cannot be guaranteed without additional security mechanisms. Implementations and operators SHOULD NOT use UIO information for security-critical decisions.

4.5. Intended Use

The extensions defined in this document are intended exclusively for administrative debugging and troubleshooting purposes. They provide diagnostic information in ICMP responses and are not designed for use in production protocols, automation systems, or non-debugging applications.

4.6. Rate Limiting

Implementations SHOULD apply rate limiting to the generation of ICMP messages containing UIO extensions to prevent resource exhaustion and potential denial-of-service conditions.

5. IANA Considerations

5.1. ICMP Extension Object Class

IANA is requested to assign a new value from the "ICMP Extension Object Classes and Class Sub-types" registry (https://www.iana.org/assignments/icmp-parameters/) for the Underlay Information Object (UIO) as follows:

Class Value: TBA (suggested value: TBD by IANA)
Class Name: Underlay Information Object
Reference: [This RFC]

5.2. C-Type Values

IANA is requested to establish a new sub-registry titled "Underlay Information Object C-Types" under the "ICMP Extension Object Classes and Class Sub-types" registry.

Initial values for this registry are as follows:

C-Type Value	Description	Reference
0	Reserved/Unspecified	[This RFC]
1-246	Unassigned	
247-255	Reserved for Private or Experimental Use	[This RFC]

The registration procedure for values 1-246 is Standards Action or IESG Approval as defined in [RFC8126].

6. Operational Considerations

6.1. Configuration

Operators SHOULD carefully configure which overlay endpoints or networks are authorized to receive UIO information. To effectively manage the security and operational aspects of UIO, implementations SHOULD provide configuration options, including but not limited to:

- Enable/disable UIO generation (default: disabled)
- Whitelist of authorized overlay prefixes
- Maximum UIO object payload size
- Rate limiting parameters

6.2. Troubleshooting Workflow

The intended use case for UIO is as follows:

1. An overlay operator performs traceroute from an overlay endpoint
2. The traceroute reveals a failure point in the path
3. ICMP error messages include UIO extensions with underlay details
4. The overlay operator uses this information to coordinate with the underlay operator for problem resolution

6.3. Multi-Vendor Interoperability

Implementations SHOULD be tested for interoperability, particularly when overlay and underlay equipment are from different vendors.

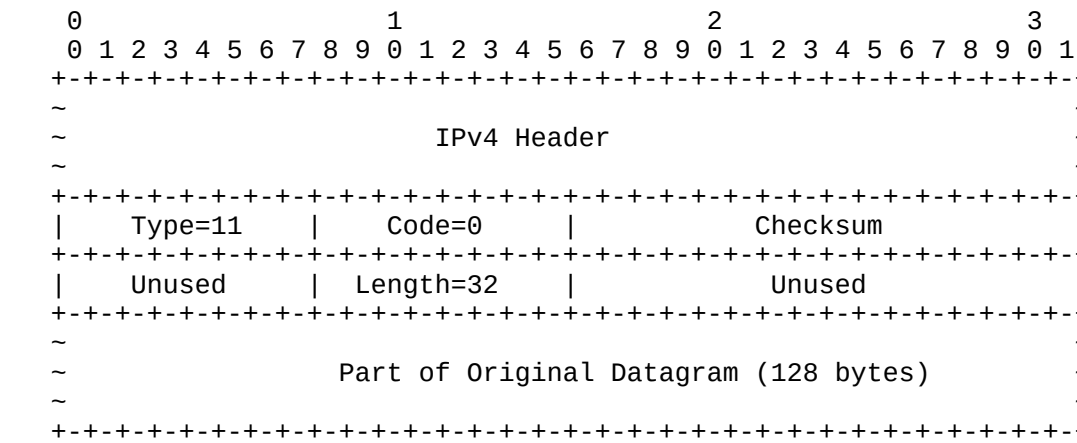
7. Appendix

7.1. UIO ICMP Extension Message Examples

This section lists examples of UIO encoding.

7.1.1. UIO carrying IPv6 information to the IPv4 source

In this example, a host receives an IPv4 ICMPv4 Time Exceeded error message in response to an ICMP Echo Request as part of the traceroute application. It also contains an UIO ICMP extension object with IPv6 interface address information as follows.



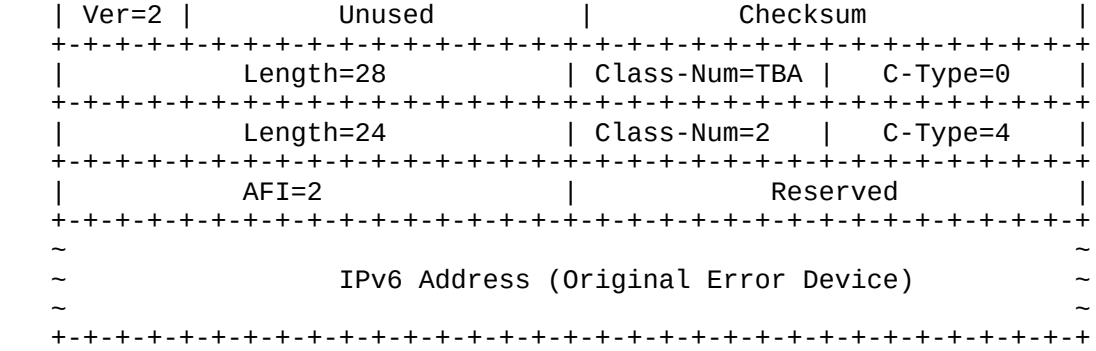
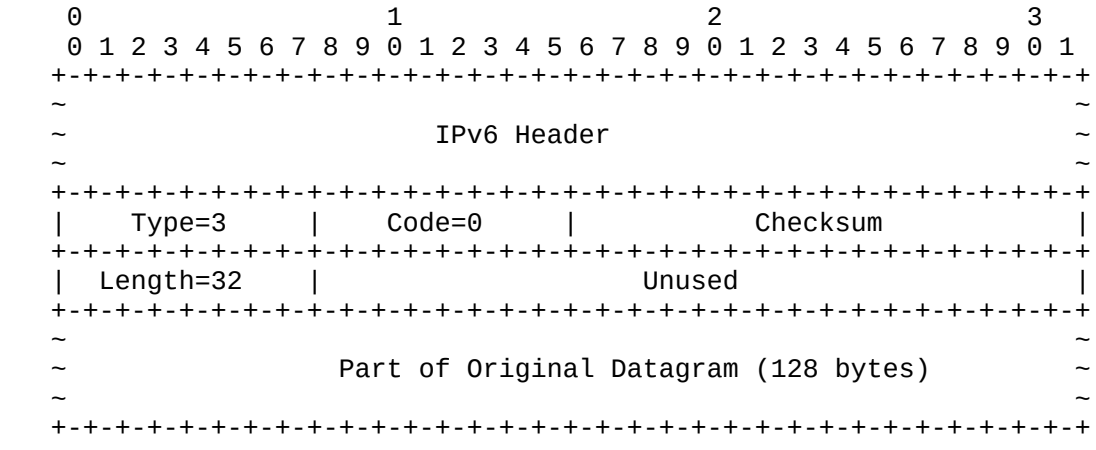


Figure 2: ICMPv4 packet carrying UIO ICMP extension

The traceroute application displays the IPv6 Address in the UIO to allow an administrator to trace the underlay path of the route being traced.

7.1.2. UIO carrying IPv4 information to the IPv6 source

In this example, a host receives an IPv6 ICMPv6 Time Exceeded error message in response to an ICMP Echo Request as part of the traceroute application. It contains a UIO ICMP extension object with IPv4 interface address information as follows.



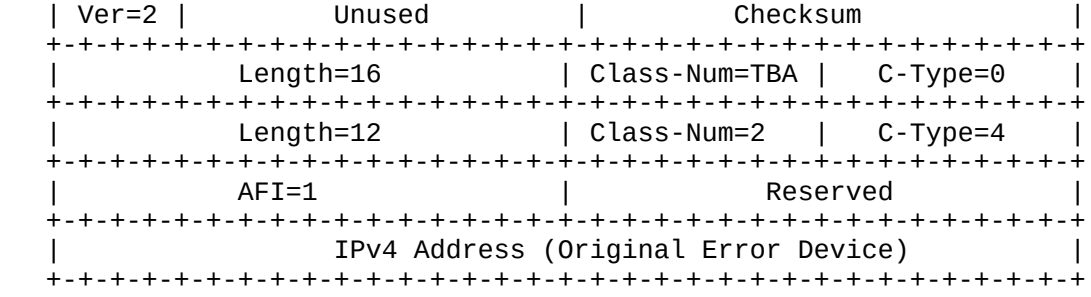


Figure 3: UIO carrying IPv4 information to the IPv6 source

The traceroute application displays the IPv4 Address in the UIO to allow an administrator to trace the underlay path of the route being traced.

8. Normative References

[This RFC] This document.

[RFC791] Postel, J., "Internet Protocol", STD 5, RFC 791, DOI 10.17487/RFC0791, September 1981, <https://www.rfc-editor.org/rfc/rfc791>.

[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>.

[RFC4884] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "Extended ICMP to Support Multi-Part Messages", RFC 4884, DOI 10.17487/RFC4884, April 2007, <https://www.rfc-editor.org/rfc/rfc4884>.

[RFC5837] Atlas, A., Ed., Bonica, R., Ed., Pignataro, C., Ed., Shen, N., and JR. Rivers, "Extending ICMP for Interface and Next-Hop Identification", RFC 5837, DOI 10.17487/RFC5837, April 2010, <https://www.rfc-editor.org/rfc/rfc5837>.

[RFC8126] Cotton, M., Leiba, B., and T. Narten, "Guidelines for Writing RFCs with Nits", BCP 14, RFC 8126, DOI 10.17487/RFC8126, June 2017, <https://www.rfc-editor.org/rfc/rfc8126>.

[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>.

[RFC8335] Bonica, R., Thomas, R., Linkova, J., Lenart, C., and M. Boucadair, "PROBE: A Utility for Probing Interfaces", RFC 8335, DOI 10.17487/RFC8335, February 2018, <https://www.rfc-editor.org/rfc/rfc8335>.

[RFC8883] Herbert, T., "ICMPv6 Errors for Discarding Packets Due to Processing Limits", RFC 8883, DOI 10.17487/RFC8883, September 2020, <https://www.rfc-editor.org/rfc/rfc8883>.

9. Informative References

[IANA.address-family-numbers] IANA, "Address Family Numbers", <http://www.iana.org/assignments/address-family-numbers>.

Acknowledgments

The authors thank the contributors listed below for their substantial input and review.

Contributors

Tamilselvan Murugan
Cisco Systems, Inc.
Email: tammurug@cisco.com

Dhilip Sekar
Email: dhilipsekar1998@gmail.com

Authors' Addresses

Jaganbabu Rajamanickam (editor)
Cisco Systems, Inc.
Canada
Email: jrajaman@cisco.com

Darren Dukes
Cisco Systems, Inc.
Canada
Email: ddukes@cisco.com

Madhan Sankaranarayanan (editor)
Cisco Systems, Inc.
India
Email: madsanka@cisco.com