Internet Area Working Group
Internet-Draft
Obsoletes: 5837 (if approved)
Intended status: Standards Track
Expires: 27 September 2026

J. Mitchell
Google LLC
March 2026

Extending ICMP for Interface and Next-Hop Identification
draft-mitchell-intarea-rfc5837bis-01

Abstract

This memo defines data structures that can be appended to selected ICMP messages. The ICMP extensions defined herein can be used to identify any combination of the following: the IP interface upon which a datagram arrived, the sub-IP component of an IP interface upon which a datagram arrived, the IP interface through which the datagram would have been forwarded had it been forwardable, the sub-IP component of an IP interface through which the datagram would have been forwarded had it been forwardable, and the IP next hop to which the datagram would have been forwarded.

Devices can use this ICMP extension to identify interfaces and their components by any combination of the following: ifIndex, IPv4 address, IPv6 address, name, and MTU. ICMP-aware devices can use these extensions to identify both numbered and unnumbered interfaces.

This document obsoletes RFC 5837. To preserve strict backward compatibility with legacy implementations, it preserves the original Interface Information Object (Class-Num 2) as defined in RFC 5837, and introduces a new Extended Interface Information Object to accommodate new interface roles.

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 2 September 2026.

Table of Contents

1. Introduction

IP devices use the Internet Control Message Protocol (ICMPv4 [RFC0792] and ICMPv6 [RFC4443]) to convey control information. In particular, when an IP device receives a datagram that it cannot process, it may send an ICMP message to the datagram's originator. Network operators and higher-level protocols use these ICMP messages to detect and diagnose network issues.

In the simplest case, the source address of the ICMP message identifies the interface upon which the datagram arrived. However, in many cases, the incoming interface is not identified by the ICMP message at all. Details follow:

According to [RFC1812], when a router generates an ICMPv4 message, the source address of that message MUST be one of the following:

* one of the IP addresses associated with the physical interface over which the ICMPv4 message is transmitted

* if that interface has no IP addresses associated with it, the device's router-id or host-id is used instead

If all of the following conditions are true, the source address of the ICMPv4 message identifies the interface upon which the original datagram arrived:

* the device sends an ICMPv4 message through the same interface upon which the original datagram was received

* that interface is numbered

However, the incoming and outgoing interfaces may be different due to an asymmetric return path, which can occur due to asymmetric link costs, parallel links, or Equal Cost Multipath (ECMP).

Similarly, [RFC1122] provides guidance for source address selection for multihomed IPv4 hosts. These recommendations, like those stated above, do not always cause the source address of an ICMPv4 message to identify the incoming interface.

ICMPv6 is somewhat more flexible. [RFC4443] states that for responses to messages sent to a non-local interface, the source address must be chosen as follows:

* the Source Address of the ICMPv6 packet MUST be a unicast address belonging to the node. The address SHOULD be chosen according to the rules that would be used to select the source address for any

other packet originated by the node, given the destination address of the packet. However, it MAY be selected in an alternative way if this would lead to a more informative choice of address reachable from the destination of the ICMPv6 packet.

When a datagram that cannot be processed arrives on an unnumbered interface, neither ICMPv4 nor ICMPv6 is currently capable of identifying the incoming interface. Even when an ICMP message is generated such that the ICMP source address identifies the incoming interface, the receiver of that ICMP message has no way of knowing if this is the case. ICMP extensions are required to explicitly identify the incoming interface.

Using the extension defined herein, a device can explicitly identify the incoming IP interface or its sub-IP components by any combination of the following:

* ifIndex

* IPv4 address

* IPv6 address

* name

* MTU

The interface name SHOULD be identical to the first 63 octets of the ifName, as defined in [RFC2863]. The ifIndex is also defined in [RFC2863].

Using the same extension, an IP device can explicitly identify by the above the outgoing interface over which a datagram would have been forwarded if that datagram had been deliverable.

The next-hop IP address, to which the datagram would have been forwarded, can also be identified using this same extension. This information can be used for creating a downstream map. The next-hop information may not always be available. There are corner-cases where it doesn’t exist and there may be implementations where it is not practical to provide this information. This specification provides an encoding for providing the next-hop IP address when it is available.

The extension defined herein uses the ICMP multi-part message framework defined in [RFC4884]. The same backward compatibility issues that apply to [RFC4884] apply to this extension.

2. 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.

3. Applications

3.1. Application to Traceroute

ICMP extensions defined in this memo provide additional capability to traceroute. An enhanced traceroute application, like older implementations, identifies nodes that a datagram visited en route to its destination. It differs from older implementations in that it can explicitly identify the following at each node:

* the IP interface upon which a datagram arrived

* the sub-IP component of an IP interface upon which a datagram arrived

* the IP interface through which the datagram would have been forwarded had it been forwardable

* the sub-IP component of an IP interface through which the datagram would have been forwarded had it been forwardable

* the IP next hop to which the datagram would have been forwarded

Enhanced traceroute applications can identify the above listed entities by:

* ifIndex

* IPv4 address

* IPv6 address

* name

* MTU

The ifIndex can be utilized within a management domain to map to an actual interface, but it is also valuable in public applications. The ifIndex can be used as an opaque token to discern whether or not two ICMP messages generated from the same router involve the same interface.

3.2. Policy and MTU Detection

A general application would be to identify which outgoing interface triggered a given function for the original packet. For example, if an access control list (ACL) drops the packet and Dest Unreachable/ Admin Prohibited denies the packet, being able to identify the outgoing interface might be useful. Another example would be to support Path MTU Discovery (PMTUD), since this would allow identification of which outgoing interface can't support a given MTU size. For example, knowledge of the problematic interface would allow an informed request for reconfiguration of the MTU of that interface.

4. Interface Information Object

This section defines the Interface Information Object, an ICMP extension object with a Class-Num (Object Class Value) of 2 that can be appended to the following messages:

* ICMPv4 Time Exceeded
* ICMPv4 Destination Unreachable
* ICMPv4 Parameter Problem
* ICMPv6 Time Exceeded
* ICMPv6 Destination Unreachable

For reasons described in [RFC4884], this extension cannot be appended to any of the currently defined ICMPv4 or ICMPv6 messages other than those listed above.

The extension defined herein MAY be appended to any of the above listed messages and SHOULD be appended whenever required to identify an unnumbered interface and when local policy or security considerations do not supersede this requirement.

A single ICMP message can contain as few as zero and as many as four instances of the Interface Information Object. It is illegal if it contains more than four instances, because that means that an interface role is used more than once (see Section 6.4).

A single instance of the Interface Information Object can provide information regarding any one of the following interface roles:

* the IP interface upon which a datagram arrived

* the sub-IP component of an IP interface upon which a datagram arrived

* the IP interface through which the datagram would have been forwarded had it been forwardable

* the IP next hop to which the datagram would have been forwarded

The following are examples of sub-IP components of IP interfaces upon which a datagram might arrive:

* Ethernet Link Aggregation Group Member

* Multilink PPP bundle member

* Multilink frame relay bundle member

To minimize the number of octets required for this extension, there are four different pieces of information that can appear in an Interface Information Object.

1. The ifIndex of the interface of interest MAY be included. This is the 32-bit ifIndex assigned to the interface by the device as specified by the Interfaces Group MIB [RFC2863].

2. An IP Address Sub-Object MAY be included if either of the following conditions is true: a) the eliciting datagram is IPv4 and the identified interface has at least one IPv4 address associated with it, or b) the eliciting datagram is IPv6 and the identified interface has at least one IPv6 address associated with it. The IP Address Sub-Object is described in Section 6.1 of this memo.

3. An Interface Name Sub-Object, containing a string of no more than 63 octets, MAY be included. That string, as specified in Section 6.2, is the interface name and SHOULD be the MIB-II ifName [RFC2863], but MAY be some other human-meaningful name of the interface.

4. A 32-bit unsigned integer reflecting the MTU MAY be included.

4.1. C-Type Meaning in an Interface Information Object

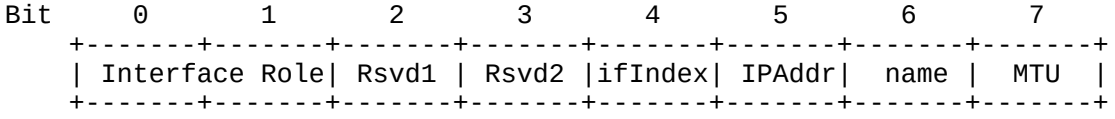

Figure 1: C-Type for the Interface Information Object

The following are bit-field definitions for C-Type:

Interface Role (bits 0-1): These bits indicate the role of the interface being identified. The enumerated values are given below:

Value 0: This object describes the IP interface upon which a datagram arrived

Value 1: This object describes the sub-IP component of an IP interface upon which a datagram arrived

Value 2: This object describes the IP interface through which the datagram would have been forwarded had it been forwardable

Value 3: This object describes the IP next hop to which the datagram would have been forwarded

Reserved 1 (bit 2): MUST be set to 0 and ignored on receipt.

Reserved 2 (bit 3): MUST be set to 0 and ignored on receipt.

ifIndex (bit 4): When set, the 32-bit ifIndex of the interface is included.

IP Addr (bit 5): When set, an IP Address Sub-Object is present. When clear, an IP Address Sub-Object is not present. The IP Address Sub-Object is described in Section 6.1 of this memo.

Interface Name (bit 6): When set, an Interface Name Sub-Object is included. When clear, it is not included. The Name Sub-Object is described in Section 6.2 of this memo.

MTU (bit 7): When set, a 32-bit integer representing the MTU is present. When clear, this 32-bit integer is not present.

The information included does not self-identify, so this specification defines a specific ordering for sending the information that must be followed.

If bit 4 (ifIndex) is set, then the 32-bit ifIndex MUST be sent first. If bit 5 (IP Address) is set, an IP Address Sub-Object MUST be sent next. If bit 6 (Name) is set, an Interface Name Sub-Object MUST be sent next. If bit 7 is set, an MTU MUST be sent next. The information order is thus: ifIndex, IP Address Sub-Object, Interface Name Sub-Object, and MTU. Any or all pieces of information may be present or absent, as indicated by the C-Type. Any data that follows these optional pieces of information MUST be ignored.

It is valid (though pointless until additional bits are assigned by IANA) to receive an Interface Information Object where bits 4, 5, 6, and 7 are all 0; this MUST NOT generate a warning or error.

5. Extended Interface Information Object

To extend the functionality in RFC 5837 without breaking legacy implementations that strictly validate the C-Type bitmask, this document defines the Extended Interface Information Object. It has a Class-Num (Object Class Value) of TBA1.

A single instance of the Extended Interface Information Object provides information regarding interface roles not covered by the original Interface Information Object.

5.1. C-Type Meaning in an Extended Interface Information Object

For this object, the C-Type is used to indicate both the role of the interface and the information that is included.

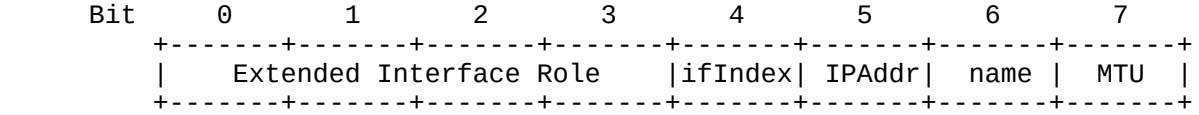

Figure 2: C-Type for the Extended Interface Information Object

The following are bit-field definitions for C-Type:

Extended Interface Role (bits 0-3): A 4-bit field indicating the role.

* Value 0: the sub-IP component of an IP interface through which the datagram would have been forwarded had it been forwardable.

* Values 1-15: Unassigned.

ifIndex (bit 4): When set, the 32-bit ifIndex is included.

IP Addr (bit 5): When set, an IP Address Sub-Object is present.

Interface Name (bit 6): When set, an Interface Name Sub-Object is included.

MTU (bit 7): When set, a 32-bit integer representing the MTU is present.

The ordering rules for appending sub-objects (ifIndex, IP Addr, Name, MTU) remain identical to those described in Section 4.

6. Shared Sub-Objects

Both the Interface Information Object and the Extended Interface Information Object utilize the exact same sub-objects.

6.1. Interface IP Address Sub-Object

Figure 3 depicts the Interface Address Sub-Object:

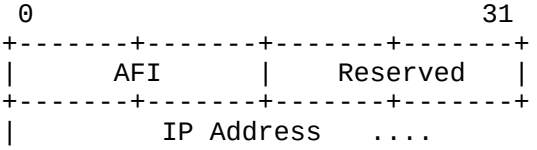

Figure 3: Interface Address Sub-Object

The IP Address Sub-Object contains the following fields:

* Address Family Identifier (AFI): This 16-bit bit field identifies the type of address represented by the IP Address field. It also determines the length of that field and the length of the entire sub-object. Values for this field represent a subset of values found in the IANA registry of Address Family Numbers (available from http://www.iana.org). Valid values are 1 (representing a 32-bit IPv4 address) and 2 (representing a 128-bit IPv6 address).

* Reserved: This 16-bit field MUST be set to zero and ignored upon receipt.

* IP Address: This variable-length field represents an IP address associated with the identified interface.

If the eliciting datagram was IPv4, the IP Interface Sub-Object MUST represent an IPv4 address. Likewise, if the eliciting datagram was IPv6, the IP Interface Sub-Object MUST represent an IPv6 address.

6.2. Interface Name Sub-Object

Figure 4 depicts the Interface Name Sub-Object:

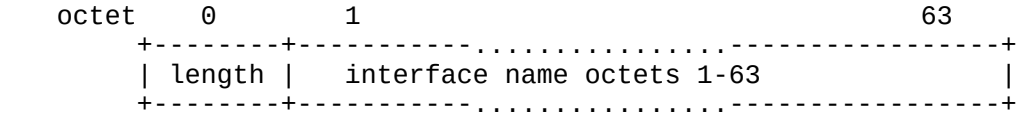

Figure 4: Interface Name Sub-Object

The Interface Name Sub-Object MUST have a length that is a multiple of 4 octets and MUST NOT exceed 64 octets.

The Length field represents the length of the Interface Name Sub-Object, including the length and the interface name in octets. The maximum valid length is 64 octets. The length is constrained to ensure there is space for the start of the original packet and additional information.

The second field contains the human-readable interface name. The interface name SHOULD be the full MIB-II ifName [RFC2863], if less than 64 octets, or the first 63 octets of the ifName, if the ifName is longer. The interface name MAY be some other human-meaningful name of the interface. It is useful to provide the ifName for cross-correlation with other MIB information and for human-reader familiarity. The interface name MUST be padded with ASCII NULL characters if the object would not otherwise terminate on a 4-octet boundary.

The interface name MUST be represented in the UTF-8 charset [RFC3629] using the Default Language [RFC2277].

6.3. Interface Information Object Examples

Figure 5 shows a full ICMPv4 Time Exceeded message, including the Interface Information Object, which MUST be preceded by an ICMP Extension Structure Header and an ICMP Object Header. Both are defined in [RFC4884].

Although examples show an Interface Name Sub-Object of length 64, this is only for illustration and depicts the maximum allowable length.

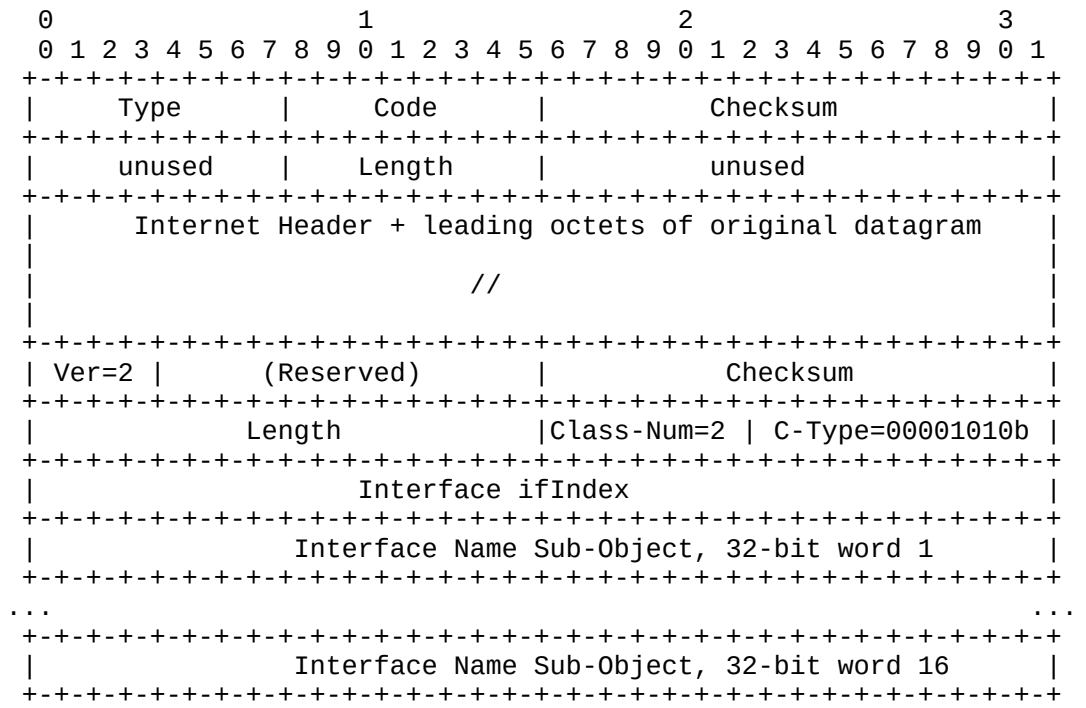

Figure 5: ICMPv4 Time Exceeded Message with Interface Information Object

Figure 6 depicts an Interface Information Object representing an incoming interface identified by ifIndex and Name.

```
Class-Num = 2
C-Type = 00001010b   // Indicates incoming interface
Length = 72 (4 + 4 + 64)
```

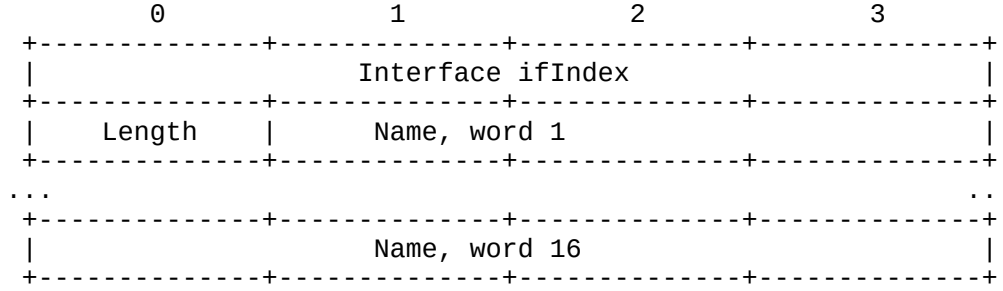

Figure 6: Incoming Interface: By ifIndex and Name

Figure 7 depicts an Interface Information Object representing an incoming interface identified by ifIndex, IPv4 Address, and Name.

```
      Class-Num = 2
      C-Type = 00001110b   // Indicates incoming interface
      Length = 80 (4 + 4 + 8 + 64)
```

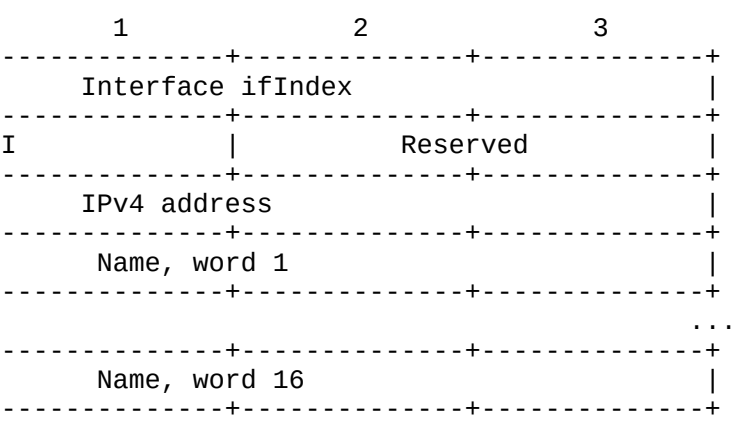

Figure 7: Incoming Interface: by ifIndex, IPv4 Address, and Name

Figure 8 depicts an Interface Information Object representing an incoming interface identified by ifIndex and IPv6 Address.

```
    Class-Num = 2
    C-Type = 00001100b   // Indicates incoming interface
    Length = 28 (4 + 4 + 20)
```

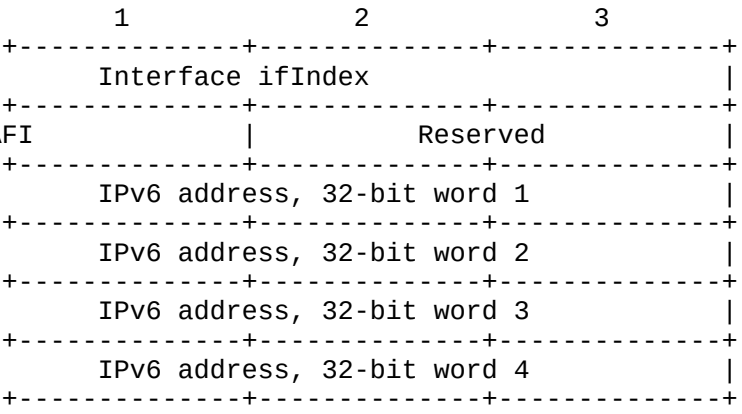

Figure 8: Incoming Interface: By ifIndex and IPv6 Address

Figure 9 depicts an Interface Information Object representing an outgoing interface identified by ifIndex and Name.

```
    Class-Num = 2
    C-Type = 10001010b   // Indicates outgoing interface
    Length = 72 (4 + 4 + 64)
```

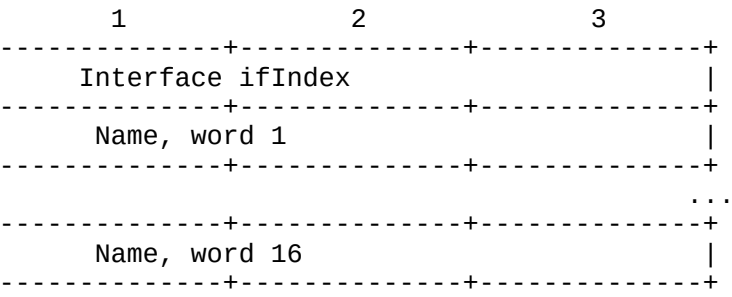

Figure 9: Outgoing Interface: By ifIndex and Name

6.4. Usage

Multiple Interface Information Objects (Class-Num 2) and Extended Interface Information Objects (Class-Num TBA1) MAY be included within a single ICMP message, provided that each object specifies a unique role.

A single ICMP message MUST NOT contain two Interface Information Objects (Class-Num 2) that specify the same role. A single ICMP message MUST NOT contain two Extended Interface Information Objects (Class-Num TBA1) that specify the same role.

Implementations sending detailed forwarding information SHOULD send both the Outgoing IP Interface (Class-Num 2, Role 2) and the Sub-IP Component of the Outgoing IP Interface (Class-Num TBA1, Role 0) when both are known. Legacy receivers will process the Class-Num 2 object and safely ignore the unknown Class-Num TBA1 object, as per the standard [RFC4884] multi-part message framework.

ifIndex, MTU, and name information MAY be included whenever it is available; more than one instance of each of these three information elements MUST NOT be included per Interface Information Object.

A single instance of IP Address information MAY be included in an Interface Information Object under the following circumstances:

* if the eliciting datagram is IPv4 and an IPv4 address is associated with the identified interface. In this case, if an IP Address Sub-Object is included, it MUST specify an IPv4 address.

* if the eliciting datagram is IPv6 and an IPv6 address is associated with the identified interface. In this case, if an IP Address Sub-Object is included, it MUST specify an IPv6 address.

In all other circumstances, IP address information MUST NOT be included.

An ICMP message that does not conform to these rules and contains multiple instances of the same information is considered illegal; specifically, an ICMP message containing more than one Interface Information Object or Extended Interface Information Object with the same role, as well as an ICMP message containing a duplicate information element in a given role are considered illegal. If such an illegal ICMP message is received, it MUST be silently discarded.

7. Network Address Translation Considerations

[RFC5508] encourages Traditional IP Network Address Translators (Traditional NATs; see [RFC3022]) to support ICMP extension objects. This document defines an ICMP extension that includes IP addresses and therefore contains realm-specific information, and consequently describes possible NAT behaviors in the presence of these extensions.

NAT devices MUST NOT translate or overwrite the ICMP extensions described herein. That is, they MUST either remove the extension entirely or pass it unchanged.

It is conceivable that a NAT device might translate an ICMP header without translating the extension defined herein. In this case, the ICMP message might contain two instances of the same address, one translated and the other untranslated. Therefore, application developers should not assume addresses in the extension are of the same realm as the addresses in the datagram’s header.

It also is conceivable that a NAT device might translate an ICMPv4 message into ICMPv6 or vice versa. If that were to occur, applications might receive ICMPv6 messages that contain IP Address Sub-Objects that specify IPv4 addresses. Likewise, applications might receive ICMPv4 messages that contain IP Address Sub-Objects that specify IPv6 addresses.

8. Security Considerations

This extension can provide the user of traceroute with additional network information that is not currently available. Implementations SHOULD provide configuration switches that suppress the generation of this extension based upon role (i.e., incoming interface, outgoing interface, sub-IP data). Implementations SHOULD also provide configuration switches that conceal various types of information (e.g., ifIndex, interface name).

It may be desirable to provide this information to a particular network's operators and not to others. If such policy controls are desirable, then an implementation could determine what sub-objects to include based upon the destination IP address of the ICMP message that will contain the sub-objects. The implementation of policy controls could also be based upon the mechanisms described in [TRACEROUTE-EXT] for those limited cases supported.

For instance, the IP address may be included for all potential recipients. The ifIndex and interface name could be included as well if the destination IP address is a management address of the network that has administrative control of the router.

Another example use case would be where the detailed information in these extensions may be provided to ICMP destinations within the local administrative domain, but only traditional information is provided to 'external' or untrusted ICMP destinations.

The intended field of use for the extensions defined in this document is administrative debugging and troubleshooting. The extensions herein defined supply additional information in ICMP responses. These mechanisms are not intended to be used in non-debugging applications.

This document does not specify an authentication mechanism for the extension that it defines. Application developers should be aware that ICMP messages and their contents are easily spoofed.

9. IANA Considerations

IANA is requested to take the following actions:

* Maintain the reservation of Class-Num 2 for the "Interface Information Object" in the ICMP Extension Object Classes registry.

* Maintain the C-Type values for Class-Num 2 as defined in [RFC5837], updating the reference to this document.

* Assign a new Class-Num TBA1 for the "Extended Interface Information Object" from the ICMP Extension Object Classes registry.

* Establish a new C-Type sub-registry for Class-Num TBA1:

  - Bit 0-3: Extended Interface Role

  - Bit 4: ifIndex included

  - Bit 5: IP Address Sub-Object included

  - Bit 6: Name Sub-Object included

  - Bit 7: MTU included

* Establish a new registry for "Extended Interface Roles" under Class-Num TBA1, reserving Value 0 for "Outgoing Sub-IP Component". Values 1-15 are Unallocated.

10. Acknowledgments

This document is an update of [RFC5837], and only minimally changes its text to support the additional use case. Thanks are therefore due to that document's authors: Alia K. Atlas, Ronald P. Bonica, Carlos Pignataro, Naiming Shen, and JR. Rivers. Additionally, the authors wish to thank Nachikethas Jagadeesan, Fabricio Pimenta de Avila, and Ronald P. Bonica for discussions on the approach.

11. References

11.1. 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>.

[RFC2863] McCloghrie, K. and F. Kastenholz, "The Interfaces Group MIB", RFC 2863, DOI 10.17487/RFC2863, June 2000, <https://www.rfc-editor.org/info/rfc2863>.

[RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, RFC 792, DOI 10.17487/RFC0792, September 1981, <https://www.rfc-editor.org/info/rfc792>.

[RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, RFC 3629, DOI 10.17487/RFC3629, November 2003, <https://www.rfc-editor.org/info/rfc3629>.

[RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", STD 89, RFC 4443, DOI 10.17487/RFC4443, March 2006, <https://www.rfc-editor.org/info/rfc4443>.

[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/info/rfc4884>.

[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>.

11.2. Informative References

[RFC1812] Baker, F., Ed., "Requirements for IP Version 4 Routers", RFC 1812, DOI 10.17487/RFC1812, June 1995, <https://www.rfc-editor.org/info/rfc1812>.

[RFC1122] Braden, R., Ed., "Requirements for Internet Hosts - Communication Layers", STD 3, RFC 1122, DOI 10.17487/RFC1122, October 1989, <https://www.rfc-editor.org/info/rfc1122>.

[RFC2277] Alvestrand, H., "IETF Policy on Character Sets and Languages", BCP 18, RFC 2277, DOI 10.17487/RFC2277, January 1998, <https://www.rfc-editor.org/info/rfc2277>.

[RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", RFC 3022, DOI 10.17487/RFC3022, January 2001, <https://www.rfc-editor.org/info/rfc3022>.

[RFC5508] Srisuresh, P., Ford, B., Sivakumar, S., and S. Guha, "NAT Behavioral Requirements for ICMP", BCP 148, RFC 5508, DOI 10.17487/RFC5508, April 2009, <https://www.rfc-editor.org/info/rfc5508>.

[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/info/rfc5837>.

[TRACEROUTE-EXT]
Shen, N., Pignataro, C., Asati, R., and E. Chen, "UDP Traceroute Message Extension", Work in Progress, Internet-Draft, draft-shen-udp-traceroute-ext-01, October 2008, <https://datatracker.ietf.org/doc/html/draft-shen-udp-traceroute-ext-01>.

Appendix A. Changes from RFC 5837

This document obsoletes [RFC5837]. The following technical changes have been made:

* Introduced the Extended Interface Information Object (Class-Num TBA1).

* Added support for the "Outgoing Sub-IP Component" role using the new Extended Interface Information Object.

* Added usage guidelines clarifying that multiple distinct extension classes can be appended to the same ICMP message to provide both the existing IP interface data and the extended interface data simultaneously.

* Updated Requirements Language to current IETF standards.

Appendix B. Changes from draft-mitchell-intarea-rfc5837bis-00

* Reverted the structural changes proposed to the Class-Num 2 C-Type bitmask. The 00 draft expanded the role field of the original object by utilizing a reserved bit. Based on working group feedback regarding backward compatibility risks, this approach was discarded.

* Adopted the approach of requesting a new ICMP Extension Object (Extended Interface Information Object) with a native 4-bit role field.

* Eliminated the requirement for modern senders to drop the "Outgoing IP" object to accommodate the new "Outgoing Sub-IP" object. Both can now be transmitted cleanly via standard [RFC4884] multi-part parsing.

Author’s Address

Jon Mitchell
Google LLC
1600 Amphitheatre Parkway
Mountain View, California 94043
United States of America
Email: jrmitche@puck.nether.net