

Network Working Group  
Request for Comments: 4884  
Updates: 792, 4443  
Category: Standards Track

R. Bonica  
Juniper Networks  
D. Gan  
Consultant  
D. Tappan  
Consultant  
C. Pignataro  
Cisco Systems, Inc.  
April 2007

## Extended ICMP to Support Multi-Part Messages

### Status of This Memo

This document specifies an Internet standards track protocol for the Internet community, and requests discussion and suggestions for improvements. Please refer to the current edition of the "Internet Official Protocol Standards" (STD 1) for the standardization state and status of this protocol. Distribution of this memo is unlimited.

### Copyright Notice

Copyright (C) The IETF Trust (2007).

### Abstract

This document redefines selected ICMP messages to support multi-part operation. A multi-part ICMP message carries all of the information that ICMP messages carried previously, as well as additional information that applications may require.

Multi-part messages are supported by an ICMP extension structure. The extension structure is situated at the end of the ICMP message. It includes an extension header followed by one or more extension objects. Each extension object contains an object header and object payload. All object headers share a common format.

This document further redefines the above mentioned ICMP messages by specifying a length attribute. All of the currently defined ICMP messages to which an extension structure can be appended include an "original datagram" field. The "original datagram" field contains the initial octets of the datagram that elicited the ICMP error message. Although the original datagram field is of variable length, the ICMP message does not include a field that specifies its length. Therefore, in order to facilitate message parsing, this document allocates eight previously reserved bits to reflect the length of the "original datagram" field.

The proposed modifications change the requirements for ICMP compliance. The impact of these changes on compliant implementations is discussed, and new requirements for future implementations are presented.

This memo updates [RFC 792](#) and [RFC 4443](#).

## Table of Contents

1. Introduction .....	3
2. Conventions Used in This Document .....	4
3. Summary of Changes to ICMP .....	4
4. ICMP Extensibility .....	4
4.1. ICMPv4 Destination Unreachable .....	7
4.2. ICMPv4 Time Exceeded .....	8
4.3. ICMPv4 Parameter Problem .....	8
4.4. ICMPv6 Destination Unreachable .....	9
4.5. ICMPv6 Time Exceeded .....	9
4.6. ICMP Messages That Can Be Extended .....	10
5. Backwards Compatibility .....	10
5.1. Classic Application Receives ICMP Message with Extensions .....	12
5.2. Non-Compliant Application Receives ICMP Message with No Extensions .....	12
5.3. Non-Compliant Application Receives ICMP Message with Compliant Extensions .....	13
5.4. Compliant Application Receives ICMP Message with No Extensions .....	14
5.5. Compliant Application Receives ICMP Message with Non-Compliant Extensions .....	14
6. Interaction with Network Address Translation .....	14
7. The ICMP Extension Structure .....	15
8. ICMP Extension Objects .....	16
9. Security Considerations .....	16
10. IANA Considerations .....	17
11. Acknowledgments .....	17
12. References .....	17
12.1. Normative References .....	17
12.2. Informative References .....	17

## 1. Introduction

This document redefines selected ICMPv4 [RFC0792] and ICMPv6 [RFC4443] messages to include an extension structure and a length attribute. The extension structure supports multi-part ICMP operation. Protocol designers can make an ICMP message carry additional information by encoding that information in the extension structure.

This document also addresses a fundamental problem in ICMP extensibility. All of the ICMP messages addressed by this memo include an "original datagram" field. The "original datagram" field contains the initial octets of the datagram that elicited the ICMP error message. Although the "original datagram" field is of variable length, the ICMP message does not include a field that specifies its length.

Application software infers the length of the "original datagram" field from the total length of the ICMP message. If an extension structure were appended to the message without adding a length attribute for the "original datagram" field, the message would become unparseable. Specifically, application software would not be able to determine where the "original datagram" field ends and where the extension structure begins. Therefore, this document proposes a length attribute as well as an extension structure that is appended to the ICMP message.

The current memo also addresses backwards compatibility with existing ICMP implementations that either do not implement the extensions defined herein or implement them without adding the required length attributes. In particular, this document addresses backwards compatibility with certain, widely deployed, MPLS-aware ICMPv4 implementations that send the extensions defined herein without adding the required length attribute.

The current memo does not define any ICMP extension objects. It defines only the extension header and a common header that all extension objects share. [UNNUMBERED], [ROUTING-INST], and [MPLS-ICMP] provide sample applications of the ICMP Extension Object.

The above mentioned memos share a common characteristic. They all append information to the ICMP Time Expired message for consumption by TRACEROUTE. In this case, as in many others, appending information to the existing ICMP Time Expired Message is preferable to defining a new message and emitting two messages whenever a packet is dropped due to TTL expiration.

## 2. Conventions Used in This Document

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

## 3. Summary of Changes to ICMP

The following is a summary of changes to ICMP that are introduced by this memo:

An ICMP Extension Structure MAY be appended to ICMPv4 Destination Unreachable, Time Exceeded, and Parameter Problem messages.

An ICMP Extension Structure MAY be appended to ICMPv6 Destination Unreachable, and Time Exceeded messages.

The above mentioned messages include an "original datagram" field, and the message formats are updated to specify a length attribute for the "original datagram" field.

When the ICMP Extension Structure is appended to an ICMP message and that ICMP message contains an "original datagram" field, the "original datagram" field MUST contain at least 128 octets.

When the ICMP Extension Structure is appended to an ICMPv4 message and that ICMPv4 message contains an "original datagram" field, the "original datagram" field MUST be zero padded to the nearest 32-bit boundary.

When the ICMP Extension Structure is appended to an ICMPv6 message and that ICMPv6 message contains an "original datagram" field, the "original datagram" field MUST be zero padded to the nearest 64-bit boundary.

ICMP messages defined in the future SHOULD indicate whether or not they support the extension mechanism defined in this specification. It is recommended that all new messages support extensions.

## 4. ICMP Extensibility

RFC 792 defines the following ICMPv4 message types:

- Destination Unreachable
- Time Exceeded

- Parameter Problem
- Source Quench
- Redirect
- Echo Request/Reply
- Timestamp/Timestamp Reply
- Information Request/Information Reply

[RFC1191] reserves bits for the "Next-Hop MTU" field in the Destination Unreachable message.

RFC 4443 defines the following ICMPv6 message types:

- Destination Unreachable
- Packet Too Big
- Time Exceeded
- Parameter Problem
- Echo Request/Reply

Many ICMP messages are extensible as currently defined. Protocol designers can extend ICMP messages by simply appending fields or data structures to them.

However, the following ICMP messages are not extensible as currently defined:

- ICMPv4 Destination Unreachable (type = 3)
- ICMPv4 Time Exceeded (type = 11)
- ICMPv4 Parameter Problem (type = 12)
- ICMPv6 Destination Unreachable (type = 1)
- ICMPv6 Packet Too Big (type = 2)
- ICMPv6 Time Exceeded (type = 3)
- ICMPv6 Parameter Problem (type = 4)

These messages contain an "original datagram" field which represents the leading octets of the datagram to which the ICMP message is a response. RFC 792 defines the "original datagram" field for ICMPv4 messages. In RFC 792, the "original datagram" field includes the IP header plus the next eight octets of the original datagram. [RFC1812] extends the "original datagram" field to contain as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size (i.e., 576 octets). RFC 4443 defines the "original datagram" field for ICMPv6 messages. In RFC 4443, the "original datagram" field always contained as many octets as possible without causing the ICMP message to exceed the minimum IPv6 MTU (i.e., 1280 octets).

Unfortunately, the "original datagram" field lacks a length attribute. Application software infers the length of this field from the total length of the ICMP message. If an extension structure were appended to the message without adding a length attribute for the "original datagram" field, the message would become unparseable. Specifically, application software would not be able to determine where the "original datagram" field ends and where the extension structure begins.

In order to solve this problem, this memo introduces an 8-bit length attribute to the following ICMPv4 messages.

- Destination Unreachable (type = 3)
- Time Exceeded (type = 11)
- Parameter Problem (type = 12)

It also introduces an 8-bit length attribute to the following ICMPv6 messages.

- Destination Unreachable (type = 1)
- Time Exceeded (type = 3)

The length attribute MUST be specified when the ICMP Extension Structure is appended to the above mentioned ICMP messages.

The length attribute represents the length of the "original datagram" field. Space for the length attribute is claimed from reserved octets, whose value was previously required to be zero.

For ICMPv4 messages, the length attribute represents 32-bit words. When the length attribute is specified, the "original datagram" field MUST be zero padded to the nearest 32-bit boundary. Because the

sixth octet of each of the impacted ICMPv4 messages was reserved for future use, this octet was selected as the location of the length attribute in ICMPv4.

For ICMPv6 messages, the length attribute represents 64-bit words. When the length attribute is specified, the "original datagram" field MUST be zero padded to the nearest 64-bit boundary. Because the fifth octet of each of the impacted ICMPv6 messages was reserved for future use, this octet was selected as the location of the length attribute in ICMPv6.

In order to achieve backwards compatibility, when the ICMP Extension Structure is appended to an ICMP message and that ICMP message contains an "original datagram" field, the "original datagram" field MUST contain at least 128 octets. If the original datagram did not contain 128 octets, the "original datagram" field MUST be zero padded to 128 octets. (See [Section 5.1](#) for rationale.)

The following sub-sections depict length attribute as it has been introduced to selected ICMP messages.

#### 4.1. ICMPv4 Destination Unreachable

Figure 1 depicts the ICMPv4 Destination Unreachable Message.

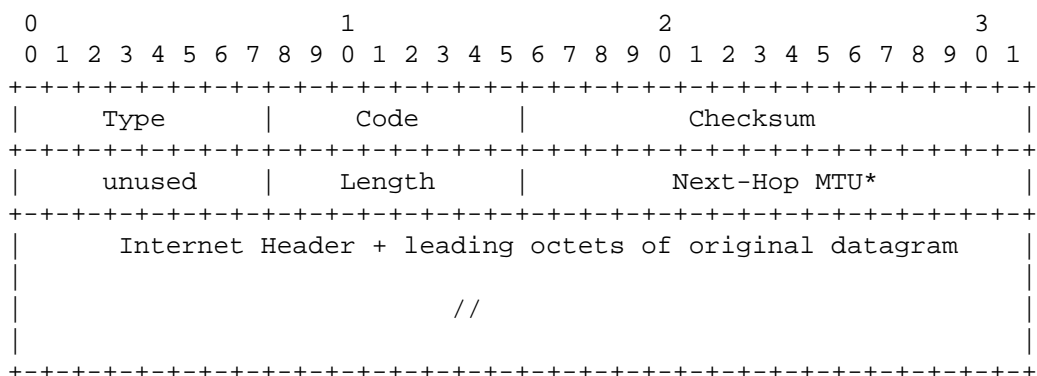


Figure 1: ICMPv4 Destination Unreachable

The syntax and semantics of all fields are unchanged from [RFC 792](#). However, a length attribute is added to the second word. The length attribute represents length of the padded "original datagram" field, measured in 32-bit words.

\* The Next-Hop MTU field is not required in all cases. It is depicted only to demonstrate that those bits are not available for assignment in this memo.

#### 4.2. ICMPv4 Time Exceeded

Figure 2 depicts the ICMPv4 Time Exceeded Message.

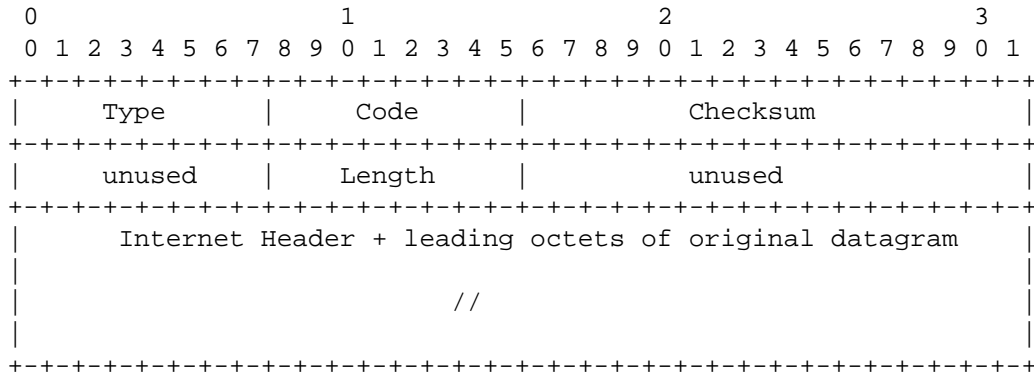


Figure 2: ICMPv4 Time Exceeded

The syntax and semantics of all fields are unchanged from [RFC 792](#), except for a length attribute which is added to the second word. The length attribute represents length of the padded "original datagram" field, measured in 32-bit words.

#### 4.3. ICMPv4 Parameter Problem

Figure 3 depicts the ICMPv4 Parameter Problem Message.

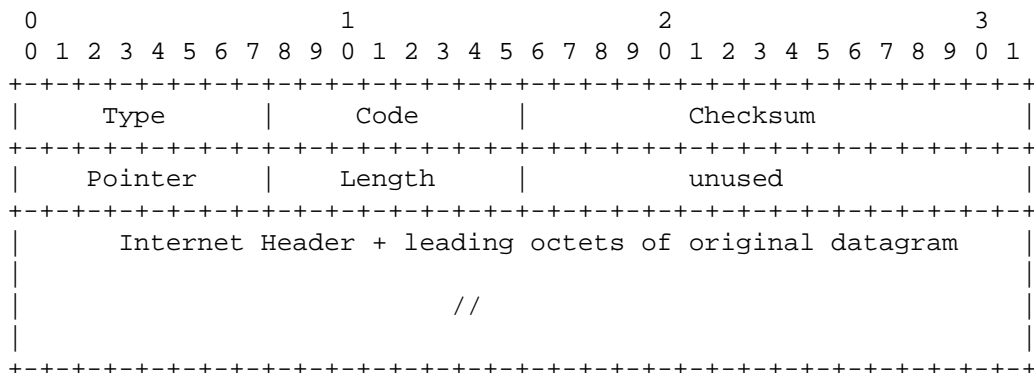


Figure 3: ICMPv4 Parameter Problem

The syntax and semantics of all fields are unchanged from [RFC 792](#), except for a length attribute which is added to the second word. The length attribute represents length of the padded "original datagram" field, measured in 32-bit words.



#### 4.4. ICMPv6 Destination Unreachable

Figure 4 depicts the ICMPv6 Destination Unreachable Message.

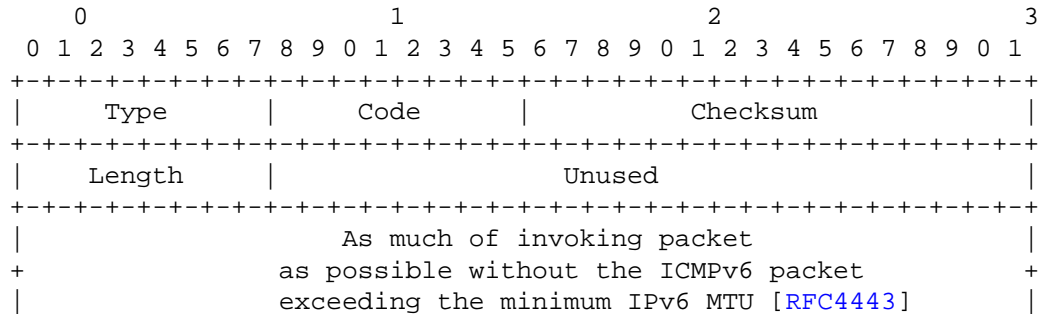


Figure 4: ICMPv6 Destination Unreachable

The syntax and semantics of all fields are unchanged from RFC 4443. However, a length attribute is added to the second word. The length attribute represents length of the padded "original datagram" field, measured in 64-bit words.

#### 4.5. ICMPv6 Time Exceeded

Figure 5 depicts the ICMPv6 Time Exceeded Message.

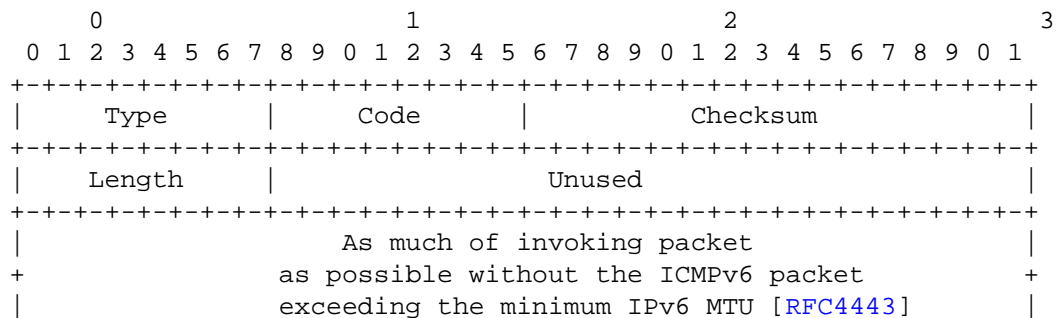


Figure 5: ICMPv6 Time Exceeded

The syntax and semantics of all fields are unchanged from RFC 4443, except for a length attribute which is added to the second word. The length attribute represents length of the padded "original datagram" field, measured in 64-bit words.

#### 4.6. ICMP Messages That Can Be Extended

The ICMP Extension Structure MAY be appended to messages of the following types:

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

The ICMP Extension Structure MUST NOT be appended to any of the other ICMP messages mentioned in [Section 4](#). Extensions were not defined for the ICMPv6 "Packet Too Big" and "Parameter Problem" messages because these messages lack space for a length attribute.

#### 5. Backwards Compatibility

ICMP messages can be categorized as follows:

- Messages that do not include any ICMP extensions
- Messages that include non-compliant ICMP extensions
- Messages that includes compliant ICMP extensions

Any ICMP implementation can send a message that does not include extensions. ICMP implementations produced prior to 1999 are not known to send ICMP extensions.

Some ICMP implementations, produced between 1999 and the time of this publication, may send a non-compliant version of ICMP extensions described in this memo. Specifically, these implementations may append the ICMP Extension Structure to the Time Exceeded and Destination Unreachable messages. When they do this, they send exactly 128 octets representing the original datagram, zero padding if required. They also calculate checksums as described in this document. However, they do not specify a length attribute to be associated with the "original datagram" field.

It is assumed that ICMP implementations produced in the future will send ICMP extensions that are compliant with this specification.

Likewise, applications that consume ICMP messages can be categorized as follows:

- Classic applications
- Non-compliant applications
- Compliant applications

Classic applications do not parse extensions defined in this memo. They are insensitive to the length attribute that is associated with the "original datagram" field.

Non-compliant implementations parse the extensions defined in this memo, but only in conjunction with the Time Expired and Destination Unreachable messages. They require the "original datagram" field to contain exactly 128 octets and are insensitive to the length attribute that is associated with the "original datagram" field. Non-compliant applications were produced between 1999 and the time of publication of this memo.

Compliant applications comply fully with the specifications of this document.

In order to demonstrate backwards compatibility, Table 1 describes how members of each application category would parse each category of ICMP message.

	No Extensions	Non-compliant Extensions	Compliant Extensions
Classic Application	-	<a href="#">Section 5.1</a>	<a href="#">Section 5.1</a>
Non-compliant Application	<a href="#">Section 5.2</a>	-	<a href="#">Section 5.3</a>
Compliant Application	<a href="#">Section 5.4</a>	<a href="#">Section 5.5</a>	-

Table 1

In the table above, cells that contain a dash represent the nominal case and require no explanation. In the following sections, we assume that the ICMP message type is "Time Exceeded".

### 5.1. Classic Application Receives ICMP Message with Extensions

When a classic application receives an ICMP message that includes extensions, it will incorrectly interpret those extensions as being part of the "original datagram" field. Fortunately, the extensions are guaranteed to begin at least 128 octets beyond the beginning of the "original datagram" field. So, only those ICMP applications that process the 129th octet of the "original datagram" field will be adversely effected. To date, only two applications falling into this category have been identified, and the degree to which they are effected is minimal.

Some TCP stacks, when they receive an ICMP message, verify the checksum in the original datagram field [[ATTACKS](#)]. If the checksum is incorrect, the TCP stack discards the ICMP message for security reasons. If the trailing octets of the original datagram field are overwritten by ICMP extensions, the TCP stack will discard an ICMP message that it would not otherwise have discarded. The impact of this issue is considered to be minimal because many ICMP messages are discarded for other reasons (e.g., ICMP filtering, network congestion, checksum was incorrect because original datagram field was truncated.)

Another theoretically possible, but highly improbably scenario occurs when ICMP extensions overwrite the portion of the original datagram field that represents the TCP header, causing the TCP stack to operate upon the wrong TCP connection. This scenario is highly unlikely because it occurs only when the TCP header appears at or beyond the 128th octet of the original datagram field and then only when the extensions approximate a valid TCP header.

### 5.2. Non-Compliant Application Receives ICMP Message with No Extensions

When a non-compliant ICMPv4 application receives a message that contains no extensions, the application examines the total length of the ICMPv4 message. If the total ICMPv4 message length is less than the length of its IP header plus 144 octets, the application correctly determines that the message does not contain any extensions.

The 144-octet sum is derived from 8 octets for the first two words of the ICMPv4 Time Exceeded message, 128 octets for the "original datagram" field, 4 octets for the ICMP Extension Header, and 4 octets for a single ICMP Object header. All of these octets would be required if extensions were present.

If the ICMPv4 payload contains 144 octets or more, the application must examine the 137th octet to determine whether it represents a valid ICMPv4 Extension Header. In order to represent a valid Extension Header, it must contain a valid version number and checksum. If it does not contain a valid version number and checksum, the application correctly determines that the message does not contain any extensions.

Non-compliant applications assume that the ICMPv4 Extension Structure begins on the 137th octet of the Time Exceeded message, after a 128-octet field representing the padded "original datagram" message.

It is possible that a non-compliant application will parse an ICMPv4 message incorrectly under the following conditions:

- the message does not contain extensions
- the original datagram field contains 144 octets or more
- selected octets of the original datagram field represent the correct values for an extension header version number and checksum

Although this is possible, it is very unlikely.

A similar analysis can be performed for ICMPv6. However, the numeric constants would change as appropriate.

### 5.3. Non-Compliant Application Receives ICMP Message with Compliant Extensions

When a non-compliant application receives a message that contains compliant ICMP extensions, it will parse those extensions correctly only if the "original datagram" field contains exactly 128 octets. This is because non-compliant applications are insensitive to the length attribute that is associated with the "original datagram" field. (They assume its value to be 128.)

Provided that the entire ICMP message does not exceed the minimum reassembly buffer size (576 octets for ICMPv4 or 1280 octets for ICMPv6), there is no upper limit upon the length of the "original datagram" field. However, each implementation will decide how many octets to include. Those wishing to be backward compatible with non-compliant TRACEROUTE implementations will include exactly 128 octets. Those not requiring compatibility with non-compliant TRACEROUTE applications may include more octets.

#### 5.4. Compliant Application Receives ICMP Message with No Extensions

When a compliant application receives an ICMP message, it examines the length attribute that is associated with the "original datagram" field. If the length attribute is zero, the compliant application MUST determine that the message contains no extensions.

#### 5.5. Compliant Application Receives ICMP Message with Non-Compliant Extensions

When a compliant application receives an ICMP message, it examines the length attribute that is associated with the "original datagram" field. If the length attribute is zero, the compliant application MUST determine that the message contains no extensions. In this case, that determination is technically correct, but not backwards compatible with the non-compliant implementation that originated the ICMP message.

So, to ease transition yet encourage compliant implementation, compliant TRACEROUTE implementations MUST include a non-default operation mode to also interpret non-compliant responses. Specifically, when a TRACEROUTE application operating in non-compliant mode receives a sufficiently long ICMP message that does not specify a length attribute, it will parse for a valid extension header at a fixed location, assuming a 128-octet "original datagram" field. If the application detects a valid version and checksum, it will treat the octets that follow as an extension structure.

### 6. Interaction with Network Address Translation

The ICMP extensions defined in this memo do not interfere with Network Address Translation. [RFC3022] permits traditional NAT devices to modify selected fields within ICMP messages. These fields include the "original datagram" field mentioned above. However, if a NAT device modifies the "original datagram" field, it should modify only the leading octets of that field, which represent the outermost IP header. Because the outermost IP header is guaranteed to be contained by the first 128 octets of the "original datagram" field, ICMP extensions and NAT will not interfere with one another.

It is conceivable that a NAT implementation might overstep the restrictions of RFC 3022 and overwrite the length attribute specified by this memo. If a NAT implementation were to overwrite the length attribute with zeros, the resulting packet will be indistinguishable from a packet that was generated by a non-compliant ICMP implementation. See Section 5.5 for packet details and a discussion of backwards compatibility.

## 7. The ICMP Extension Structure

This memo proposes an optional ICMP Extension Structure that can be appended to the ICMP messages referenced in [Section 4.6](#) of this document.

The Extension Structure contains exactly one Extension Header followed by one or more objects. Having received an ICMP message with extensions, application software MAY process selected objects while ignoring others. The presence of an unrecognized object does not imply that an ICMP message is malformed.

As stated above, the total length of the ICMP message, including extensions, MUST NOT exceed the minimum reassembly buffer size. Figure 6 depicts the ICMP Extension Header.

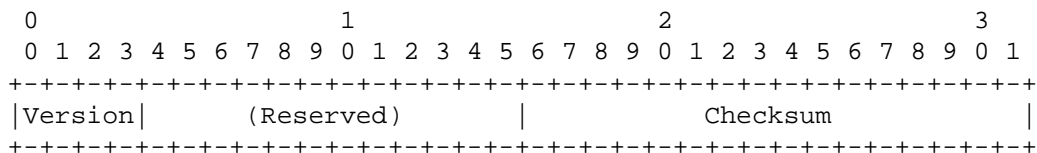


Figure 6: ICMP Extension Header

The fields of the ICMP Extension Header are as follows:

Version: 4 bits

ICMP extension version number. This is version 2.

Reserved: 12 bits

Must be set to 0.

Checksum: 16 bits

The one's complement of the one's complement sum of the data structure, with the checksum field replaced by zero for the purpose of computing the checksum. An all-zero value means that no checksum was transmitted. See [Section 5.2](#) for a description of how this field is used.

## 8. ICMP Extension Objects

Each extension object contains one or more 32-bit words, representing an object header and payload. All object headers share a common format. Figure 7 depicts the object header and payload.

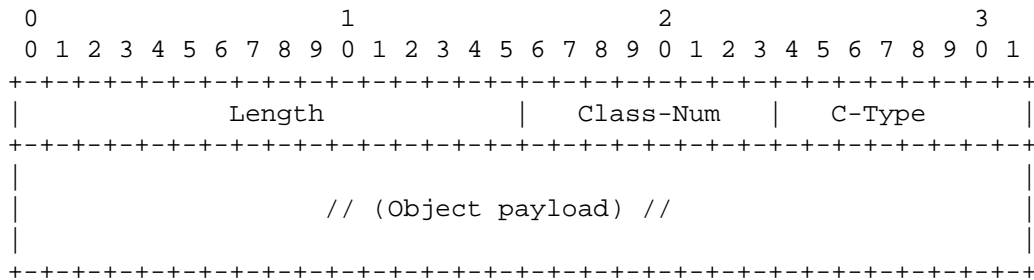


Figure 7: Object Header and Payload

An object header has the following fields:

Length: 16 bits

Length of the object, measured in octets, including the object header and object payload.

Class-Num: 8 bits

Identifies object class.

C-Type: 8 bits

Identifies object sub-type.

## 9. Security Considerations

Upon receipt of an ICMP message, application software must check it for syntactic correctness. The extension checksum must be verified. Improperly specified length attributes and other syntax problems may result in buffer overruns.

This memo does not define the conditions under which a router sends an ICMP message. Therefore, it does not expose routers to any new denial-of-service attacks. Routers may need to limit the rate at which ICMP messages are sent.



## 10. IANA Considerations

The ICMP Extension Object header contains two 8-bit fields: The Class-Num identifies the object class, and the C-Type identifies the class sub-type. Sub-type values are defined relative to a specific object class value, and are defined per class.

IANA has established a registry of ICMP extension objects classes and class sub-types. There are no values assigned within this document to maintain. Object classes 0xF7 - 0xFF are reserved for private use. Object class values are assignable on a first-come-first-serve basis. The policy for assigning sub-type values should be defined in the document defining new class values.

## 11. Acknowledgments

Thanks to Pekka Nikander, Mark Doll, Fernando Gont, Joe Touch, Christian Voiqt, and Sharon Christholm for their comments regarding this document.

## 12. References

### 12.1. Normative References

- [RFC0792] Postel, J., "Internet Control Message Protocol", STD 5, [RFC 792](#), September 1981.
- [RFC1191] Mogul, J. and S. Deering, "Path MTU discovery", [RFC 1191](#), November 1990.
- [RFC1812] Baker, F., "Requirements for IP Version 4 Routers", [RFC 1812](#), June 1995.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC4443] Conta, A., Deering, S., and M. Gupta, Ed., "Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification", [RFC 4443](#), March 2006.

### 12.2. Informative References

- [UNNUMBERED] Atlas, A., Bonica, R., Rivers, JR., Shen, N., and E. Chen, "ICMP Extensions for Unnumbered Interfaces", Work in Progress, March 2007.

- [MPLS-ICMP] Bonica, R., Gan, D., Tappan, D., and C. Pignataro, "ICMP Extensions for MultiProtocol Label Switching", Work in Progress, January 2007.
- [ATTACKS] Gont, F., "[ICMP attacks against TCP](#)", Work in Progress, October 2006.
- [ROUTING-INST] Shen, N. and E. Chen, "ICMP Extensions for Routing Instances", Work in Progress, November 2006.
- [RFC3022] Srisuresh, P. and K. Egevang, "Traditional IP Network Address Translator (Traditional NAT)", [RFC 3022](#), January 2001.

## Authors' Addresses

Ronald P. Bonica  
Juniper Networks  
2251 Corporate Park Drive  
Herndon, VA 20171  
US

EMail: [rbonica@juniper.net](mailto:rbonica@juniper.net)

Der-Hwa Gan  
Consultant

EMail: [derhwagan@yahoo.com](mailto:derhwagan@yahoo.com)

Daniel C. Tappan  
Consultant

EMail: [Dan.Tappan@gmail.com](mailto:Dan.Tappan@gmail.com)

Carlos Pignataro  
Cisco Systems, Inc.  
7025 Kit Creek Road  
Research Triangle Park, NC 27709  
US

EMail: [cpignata@cisco.com](mailto:cpignata@cisco.com)

## Full Copyright Statement

Copyright (C) The IETF Trust (2007).

This document is subject to the rights, licenses and restrictions contained in [BCP 78](#), and except as set forth therein, the authors retain all their rights.

This document and the information contained herein are provided on an "AS IS" basis and THE CONTRIBUTOR, THE ORGANIZATION HE/SHE REPRESENTS OR IS SPONSORED BY (IF ANY), THE INTERNET SOCIETY, THE IETF TRUST AND THE INTERNET ENGINEERING TASK FORCE DISCLAIM ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

## Intellectual Property

The IETF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the procedures with respect to rights in RFC documents can be found in [BCP 78](#) and [BCP 79](#).

Copies of IPR disclosures made to the IETF Secretariat and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this specification can be obtained from the IETF on-line IPR repository at <http://www.ietf.org/ipr>.

The IETF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights that may cover technology that may be required to implement this standard. Please address the information to the IETF at [ietf-ipr@ietf.org](mailto:ietf-ipr@ietf.org).

## Acknowledgement

Funding for the RFC Editor function is currently provided by the Internet Society.