EDSA-404
ISA Security Compliance Institute – Embedded Device Security Assurance –
Testing the robustness of implementations of the IETF ICMPv4 network protocol

Version 1.3

September 2010

A. DISCLAIMER

ASCI and all related entities, including the International Society of Automation (collectively, “ASCI”)provide all materials, work products and, information (‘SPECIFICATION’) AS IS, WITHOUT WARRANTY AND WITH ALL FAULTS, and hereby disclaim all warranties and conditions, whether express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of reliability or availability, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence, all with regard to the SPECIFICATION, and the provision of or failure to provide support or other services, information, software, and related content through the SPECIFICATION or otherwise arising out of the use of the SPECIFICATION. ALSO, THERE IS NO WARRANTY OR CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION, OR NON-INFRINGEMENT WITH REGARD TO THE SPECIFICATION.

WITHOUT LIMITING THE FOREGOING, ASCI DISCLAIMS ALL LIABILITY FOR HARM TO PERSONS OR PROPERTY, AND USERS OF THIS SPECIFICATION ASSUME ALL RISKS OF SUCH HARM.

IN ISSUING AND MAKING THE SPECIFICATION AVAILABLE, ASCI IS NOT UNDERTAKING TO RENDER PROFESSIONAL OR OTHER SERVICES FOR OR ON BEHALF OF ANY PERSON OR ENTITY, NOR IS ASCI UNDERTAKING TO PERFORM ANY DUTY OWED BY ANY PERSON OR ENTITY TO SOMEONE ELSE. ANYONE USING THIS SPECIFICATION SHOULD RELY ON HIS OR HER OWN INDEPENDENT JUDGMENT OR, AS APPROPRIATE, SEEK THE ADVICE OF A COMPETENT PROFESSIONAL IN DETERMINING THE EXERCISE OF REASONABLE CARE IN ANY GIVEN CIRCUMSTANCES.

B. EXCLUSION OF INCIDENTAL, CONSEQUENTIAL AND CERTAIN OTHER DAMAGES

TO THE MAXIMUM EXTENT PERMITTED BY APPLICABLE LAW, IN NO EVENT SHALL ASCI OR ITS SUPPLIERS BE LIABLE FOR ANY SPECIAL, INCIDENTAL,PUNITIVE, INDIRECT, OR CONSEQUENTIAL DAMAGES WHATSOEVER (INCLUDING, BUT NOT LIMITED TO, DAMAGES FOR LOSS OF PROFITS OR CONFIDENTIAL OR OTHER INFORMATION, FOR BUSINESS INTERRUPTION, FOR PERSONAL INJURY, FOR LOSS OF PRIVACY, FOR FAILURE TO MEET ANY DUTY INCLUDING OF GOOD FAITH OR OF REASONABLE CARE, FOR NEGLIGENCE, AND FOR ANY OTHER PECUNIARY OR OTHER LOSS WHATSOEVER) ARISING OUT OF OR IN ANY WAY RELATED TO THE USE OF OR INABILITY TO USE THE SPECIFICATION, THE PROVISION OF OR FAILURE TO PROVIDE SUPPORT OR OTHER SERVICES, INFORMATON, SOFTWARE, AND RELATED CONTENT THROUGH THE SPECIFICATION OR OTHERWISE ARISING OUT OF THE USE OF THE SPECIFICATION, OR OTHERWISE UNDER OR IN CONNECTION WITH ANY PROVISION OF THIS SPECIFICATION, EVEN IN THE EVENT OF THE FAULT, TORT (INCLUDING NEGLIGENCE), MISREPRESENTATION, STRICT LIABILITY, BREACH OF CONTRACT OF ASCI OR ANY SUPPLIER, AND EVEN IF ASCI OR ANY SUPPLIER HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

Revision history

version	date	changes
1.1	2010.06.15	Initial version published to http://www.ISASecure.org
1.3	2010.09.17	Create distinct test criteria at high but supported rate and full auto-negotiated link rate; removed protocol conformance aspects of tests since covered by other industry efforts (six tests removed); removed discovery phase since not required to perform uniform testing over all devices; removed mixing of valid and invalid messages in load testing since valid messages create more load on device

Contents

Foreword

NOTE This is one of a series of robustness test specifications for embedded devices. The full current list of documents related to embedded device security assurance can be found on the web site of the ISA Security Compliance Institute, http://www.ISASecure.org.

1 Scope

This document is intended to provide requirements for testing the robustness of embedded device implementations of the IETF ICMPv4 protocol, as a measure of the extent to which such implementations provide required "host" (e.g., non-router) functionality and defend themselves against

- correctly formed messages and sequences of such messages;
- single erroneous messages; and
- inappropriate sequences of messages;

where failure of the device to continue to provide concurrent automation system control and reporting functions demonstrates potential security vulnerabilities within the device. This document is not intended to serve as a guide for testing the correctness of implementations or conformance to mandatory provisions of the controlling standard(s), which cannot be determined solely by observing a device's response to external stimuli.

NOTE Parts of the ICMPv4 protocol have distinct server and client roles, while other parts do not.

2 Normative references

This associated specification contains requirements common to this and similar robustness tests for other protocols for embedded devices, including requirements on test configurations.

[EDSA-310] *ISASecurity Compliance Institute – Embedded device security assurance – Common requirements for communication robustness testing of IP-based protocol implementations*[1], as specified at http://www.ISASecure.org

NOTE 1 Within this document, references to specific subclauses of this normative reference are made through symbolic tags of the form [CRT.Symbolic_tag]; the resolution of those tags is made in [EDSA-310], Table 1.

These publications of the Internet Engineering Task Force (IETF) are the controlling specifications for the protocol whose robustness testing is the subject of this document:

NOTE 2 For each RFC*nnn*, the controlling version can be found at http://tools.ietf.org/html/rfc*nnn*.

IANA protocol numbers

RFC792, *Internet control message protocol [version 4]*

RFC950, *Internet standard subnetting procedure*

RFC1122, *Requirements for internet hosts – communication layers*

NOTE 3 Only 3.2.2 is referenced.

RFC1191, *Path MTU discovery*

NOTE 4 Only Clause 4 is referenced.

RFC1256, *ICMP router discovery messages*

RFC1393, *Traceroute using an IP option*

RFC1788, *ICMP domain name messages*

RFC1812, *Requirements for IP version 4 routers*

[1] to be published concurrently with this document

NOTE 5 Only 4.3 is referenced, and then only for DUTs that also act as IP routers.

RFC2521, *ICMP security failure messages*

RFC4884, *Extended ICMP to support multi-part messages*

NOTE 6 Other IETF specifications related to the above can be found in the Bibliography.

3 Definitions and abbreviations

3.1 Definitions

3.1.1
device under test
device that is being stimulated and observed during testing to demonstrate the characteristics and behavior of the device when presented with the selected sequence of test stimuli

3.1.2
erroneous (message or PDU or option)
PDU that violates either syntactic rules on PDU structure or semantic rules on PDU content or both, or PDU option that violates either syntactic rules on PDU option structure or semantic rules on PDU option content or both

NOTE 1 Semantic and syntactic rule violations can interact, as when the value of one field determines the size of another field.

NOTE 2 The term erroneous includes syntactic malformation, semantically invalid values, and contextually invalid values and sequences.

NOTE 3 This is addressed further in [CRT.Terminology_of_Erroneous].

3.1.3
fragmenting
function performed by IPv4 to map one unfragmented NPDU into multiple smaller fragmented NPDUs before transmission

NOTE The equivalent OSI terms is segmenting, as specified in ISO/IEC 7498 1:1994, 5.8.1.9.

3.1.4
inferior (protocol)
protocol at a lower layer or sublayer than the referenced protocol

3.1.5
lower tester
tester that controls and observes a protocol layer implementation in a DUT through stimulus and observation via lower protocol layers and a physical interconnection to the TD

NOTE This is the only type of testing used in the ISCI EDSA robustness tests.

3.1.6
malformed (message or PDU)
PDU that violates syntactic rules on PDU structure

NOTE This is addressed further in [CRT.Terminology_of_Erroneous].

3.1.7
reassembling
post-reception function performed by IP to reconstruct one unfragmented NPDU from multiple fragmented NPDUs

3.1.8
security parameters index
unstructured opaque index used in conjunction with a PDU's destination address to identify a particular security association and the related set of security information for a given network association or connection

NOTE This term is referenced only in 4.2.5 and 4.3.4.

3.1.9
superior (protocol)
protocol at a higher layer or sublayer than the referenced protocol

3.1.10
testing device
conceptual single network-connected device, possibly consisting of multiple physical network-connected devices, used to test the robustness of the device under test

NOTE This could be any programmable network-connected device capable of processing PDUs at the rate required for testing.

3.1.11
upper tester
tester that controls and observes a protocol layer implementation in a DUT through stimulus and observation via a DUT-internal service interface between test software and the protocol layer under test

3.1.12
vulnerability
flaw or weakness in a system's design, implementation, operation, or management that could be exploited to violate the system's integrity or security policy

3.2 Abbreviations

The following abbreviations are used in this document

AKA	also known as
CRT	communication robustness testing
DPDU	data-link-layer protocol data unit
DUT	device under test
FSM	finite state machine
IANA	Internet assigned numbers authority
ICMPv4	Internet control message protocol, version 4
IEEE	Institute of Electrical and Electronics Engineers
IETF	Internet engineering task force
IP	Internet (network layer) protocol
IPv4	IP version 4 (uses 32-bit network layer addresses)
(*N*)PDU	(*N*-layer) protocol data unit, where *N* = D (data-link), N (network), T (transport), A (application), etc
NPDU	network-layer protocol data unit
SNAP	sub-network access protocol
SPI	security parameters index
TD	testing device

4 Elements of the protocol under test

4.1 General

This document specifies robustness testing for the IETF ICMPv4 protocol, which is a mandatory, stateless, network-layer management and multi-layer error-reporting protocol that must co-exist with any IPv4 implementation.

NOTE 1 Although ICMPv4 is a mandatory protocol, many embedded devices either do not implement it, or are blocked from receiving ICMPv4 PDUs by an interposed control firewall. Some firewalls with outbound filtering rules also may block ICMPv4 error report PDUs that devices issue in response to detected errors, or permit outbound ICMPv4 error PDUs but block all other outbound ICMPv4 PDUs. Thus testability of ICMPv4 is not assured.

ICMPv4 also can be used to stimulate the DUT as a client so that responses can be observed. ICMPv4 uses IPv4, thereby providing observability of the DUT's reception and generation of IPv4 NPDUs.

NOTE 2 ICMPv4 cannot be used with IPv6; ICMPv6 is the required co-protocol for IPv6.

The mandatory EchoRequest and EchoReply PDUs of ICMPv4 provide test functionality similar to that provided by the Echo protocol (type 7) over UDP or TCP: the ability of the TD to send an almost arbitrary payload to the DUT which the DUT is obligated to return. In the case of ICMPv4 and IPv4, this capability extends to the TD sending segmented IPv4 NPDUs to the DUT, such that the DUT must reassemble them before replying, thus enabling observation of the DUT's ability to perform such reassembly correctly.

NOTE 3 Such observations are not necessary for robustness testing, but may provide useful for diagnosing other implementation faults.

4.2 ICMPv4 PDU composition

4.2.1 Generic structure of ICMPv4 PDUs

ICMPv4 PDUs are structured generically as shown in Figure 1, using a big-endian octet order; they are carried as the payload of IPv4 NPDUs.

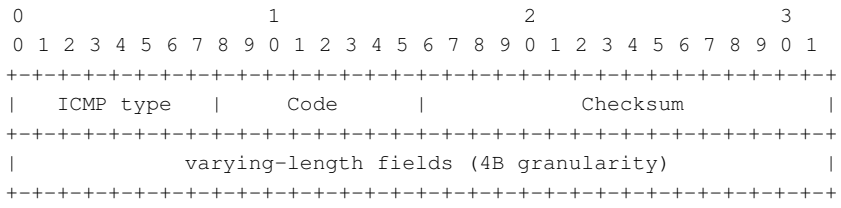


Figure 1 – ICMPv4 PDU structure

Nineteen types of PDU have been defined for ICMPv4, which fall roughly into four categories:

1) Error report:
   a) Destination unreachable, ICMP type = 0x03, specified in RFC792, RFC1191 and RFC4884
   b) Time exceeded, ICMP type = 0x0B, specified in RFC792 and RFC4884
   c) Parameter problem, ICMP type = 0x0C, specified in RFC792 and RFC4884
   d) Security failure, ICMP type = 0x40, specified as experimental in RFC2521
2) Route control command:
   e) Source quench, ICMP type = 0x04, specified in RFC792
   f) Redirect, ICMP type = 0x05, specified in RFC792
3) Testing query/reply:
   g) Echo (AKA Echo request), ICMP type = 0x08, specified in RFC792
   h) Echo reply, ICMP type = 0x00, specified in RFC792
   i) Timestamp (AKA Timestamp request), ICMP type = 0x0D, specified in RFC792
   j) Timestamp reply, ICMP type = 0x0E, specified in RFC792
4) Environment query/reply:
   k) Information request, ICMP type = 0x0F, specified in RFC792, obsolesced by RFC1812
   l) Information reply, ICMP type = 0x10, specified in RFC792, obsolesced by RFC1812
   m) Address mask request, ICMP type = 0x11, specified in RFC950
   n) Address mask reply, ICMP type = 0x12, specified in RFC950
   o) Router solicitation, ICMP type = 0x0A, specified in RFC1256

p) Router advertisement, ICMP type = 0x09, specified in RFC1256

q) Domain name request, ICMP type = 0x37, specified as experimental in RFC1788

r) Domain name reply, ICMP type = 0x38, specified as experimental in RFC1788

s) Traceroute, ICMP type = 0x52, specified as experimental in RFC1393

4.2.2 Structure of standardized extensions to ICMPv4 error report PDUs

RFC4884 recently (April 2007) introduced a standardized ICMPv4 extension structure for message types a) through c). The Extension Structure contains exactly one Extension Header, structured as shown in Figure 2, followed by one or more extension objects that are structured generically as shown in Figure 3.

When present in message types a) through c), a new Length field that specifies the length of the quoted triggering NPDU, in multiples of 4 octets, replaces a previously zero RFU field in those PDU types. The minimum required value for that Length field, when the extension structure is present, is 32, representing 128 octets of quoted NPDU that may have been zero-padded to that minimum required size.

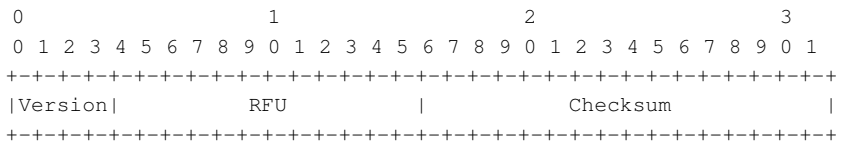


Figure 2 – ICMPv4 extension header

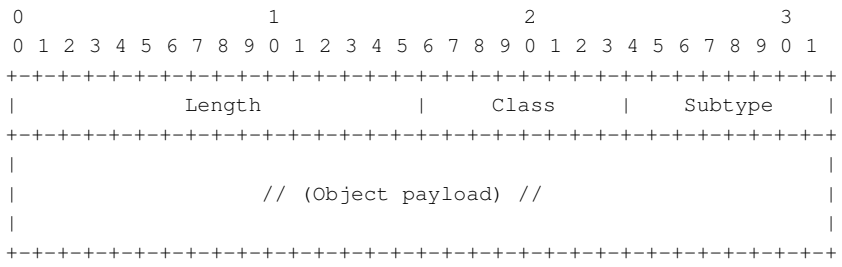
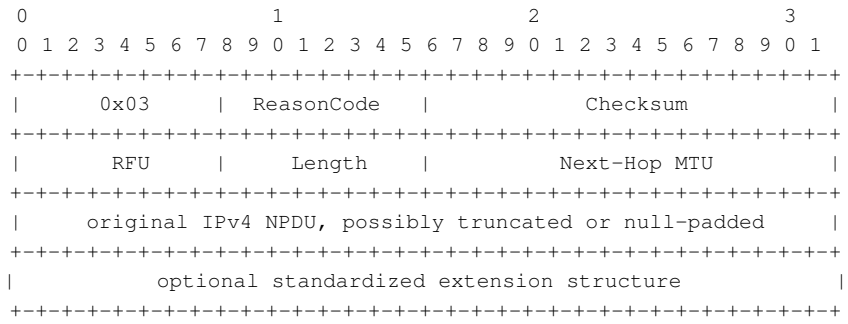


Figure 3 – ICMPv4 extension object header and payload

There is no requirement for robustness testing to test or interpret these optional extension headers; however their presence in PDU types a) through c), as indicated by a non-zero Length field whose span (in 4 B words) is 128 B or greater but is less than the size of the received ICMPv4 PDU, is not a reason for failing a DUT.

4.2.3 Structure of DestinationUnreachable ICMPv4 PDUs

This PDU is structured as shown in Figure 4; it is used to report an unreachable destination, and is sent on the reverse path toward the original source of the conveyed packet.



NOTE 1 The Next-Hop MTU field was added by RFC1191. The Length field was added by RFC4884.

NOTE 2 RFC1812 increased the amount of returned “original IPv4 NPDU” to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets. Where the optional extension structure of RFC4884 is present, the “original IPv4 NPDU” may be truncated to as little as 128 octets; if it is shorter it is required to be zero-padded to that 128-octet minimum.

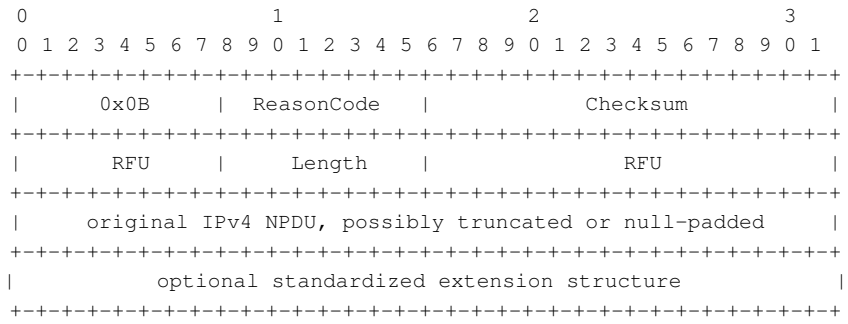
NOTE 3 The optional extension structure of RFC4884 is present when the Length field has a value of 32 or greater and the size of the received NPDU payload is greater than 4 × (*Length* + 2) B.

Figure 4 – DestinationUnreachable ICMPv4 PDU structure

Defined reason codes for DestinationUnreachable ICMP PDUs are specified in Table 1. Support for sending and receiving DestinationUnreachable error PDUs is mandatory.

4.2.4 Structure of TimeExceeded ICMPv4 PDUs

This PDU is structured as shown in Figure 5; it is used to report that a packet was discarded due to excess delay or too many hops, and is sent on the reverse path toward the original source of the conveyed packet.



NOTE 1 The Length field was added by RFC4884.

NOTE 2 RFC1812 increased the amount of returned “original IPv4 NPDU” to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets. Where the optional extension structure of RFC4884 is present, the “original IPv4 NPDU” may be truncated to as little as 128 octets; if it is shorter it is required to be zero-padded to that 128-octet minimum.

NOTE 3 The optional extension structure of RFC4884 is present when the Length field has a value of 32 or greater and the size of the received NPDU payload is greater than 4 × (*Length* + 2) B.

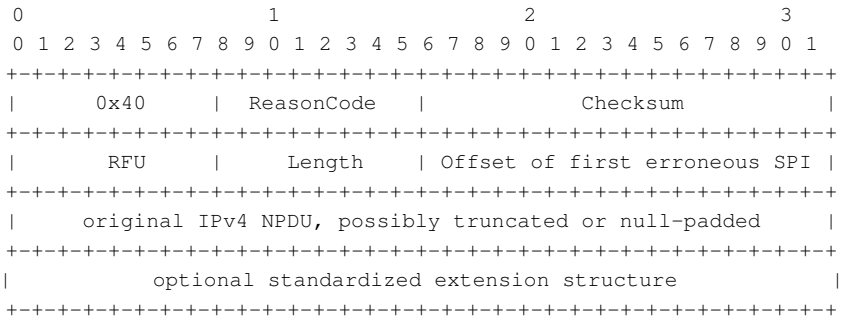
Figure 5 – TimeExceeded ICMPv4 PDU structure

Defined reason codes for TimeExceeded ICMP PDUs are specified in Table 2. Support for sending and receiving TimeExceeded error PDUs is mandatory.

4.2.5 Structure of SecurityFailure ICMPv4 PDUs

This PDU is structured as shown in Figure 6; it is based on RFC2521 for use with Internet Security Protocols [RFC-1825 et sequitur] for authentication and privacy. For statically configured Security Associations, this PDU indicates an attempted unauthorized operation or a need for reconfiguration. It also may be used to trigger automated session-key management.

NOTE Use of these PDUs is considered experimental; they are unlikely to be implemented by a DUT.



NOTE 1 The Length field was added by RFC4884.

NOTE 2 RFC1812 increased the amount of returned “original IPv4 NPDU” to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets. Where the optional extension structure of RFC4884 is present, the “original IPv4 NPDU” may be truncated to as little as 128 octets; if it is shorter it is required to be zero-padded to that 128-octet minimum.

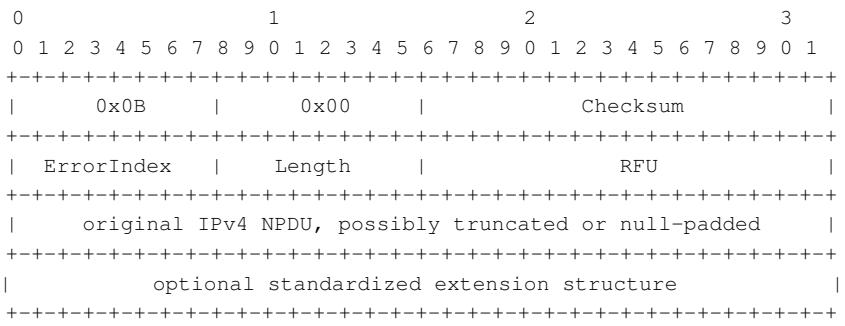
NOTE 3 The optional extension structure of RFC4884 is present when the Length field has a value of 32 or greater and the size of the received NPDU payload is greater than 4 × (*Length* + 2) B.

Figure 6 – SecurityFailure ICMPv4 PDU structure

Defined reason codes for SecurityFailure ICMP PDUs are specified in Table 2. Support for sending and receiving SecurityFailure error PDUs is optional.

4.2.6 Structure of ParameterProblem ICMPv4 PDUs

This PDU is structured as shown in Figure 7; it is used to report that a packet was discarded due to a problem with its parameters, and is sent on the reverse path toward the original source of the conveyed packet.



NOTE 1 The Length and ErrorIndex (AKA/ “pointer”) fields were added by RFC4884.

NOTE 2 RFC1812 increased the amount of returned “original IPv4 NPDU” to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets. Where the optional extension structure of RFC4884 is present, the “original IPv4 NPDU” may be truncated to as little as 128 octets; if it is shorter it is required to be zero-padded to that 128-octet minimum.

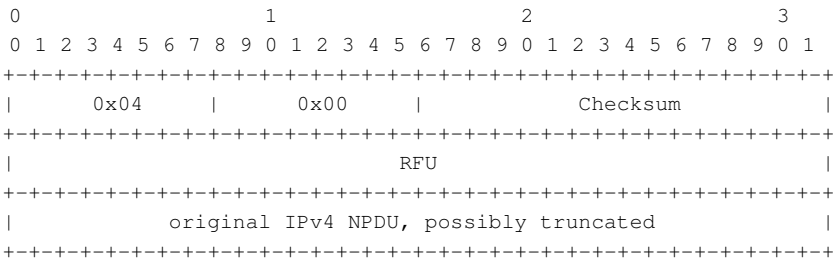
NOTE 3 The optional extension structure of RFC4884 is present when the Length field has a value of 32 or greater and the size of the received NPDU payload is greater than 4 × (*Length* + 2) B.

Figure 7 – ParameterProblem ICMPv4 PDU structure

Support for sending and receiving ParameterProblem error PDUs is mandatory.

4.2.7 Structure of SourceQuench ICMPv4 PDUs

This PDU is structured as shown in Figure 8; it is used to command that the source of a packet stop sourcing or forwarding those packets.



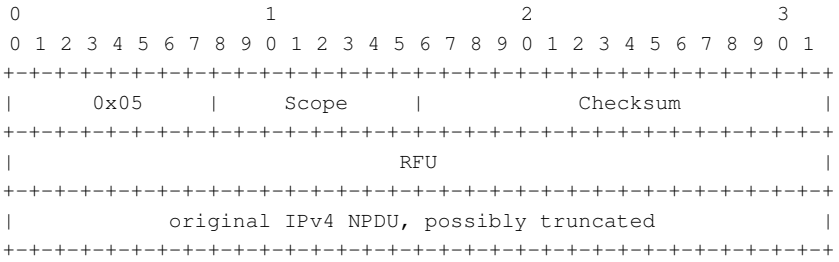
NOTE RFC1812 increased the amount of returned "original IPv4 NPDU" to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets.

Figure 8 – SourceQuench ICMPv4 PDU structure

Support for receipt of SourceQuench PDUs with appropriate consequent action is mandatory.

4.2.8 Structure of Redirect ICMPv4 PDUs

This PDU is structured as shown in Figure 9; it is used to command that the prior source or forwarder of a packet stop sourcing or forwarding those packets via a different route.



NOTE RFC1812 increased the amount of returned "original IPv4 NPDU" to as many octets as possible without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets.

Figure 9 – Redirect ICMPv4 PDU structure

Defined scope codes for Redirect ICMP PDUs are specified in Table 4. Support for receipt of Redirect PDUs with appropriate consequent action is mandatory.

4.2.9 Structure of Echo (AKA EchoRequest) and EchoReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 10; they are used to test an IPv4 implementation by requiring that implementation to receive an ICMPv4 EchoRequest (AKA Echo) PDU with arbitrary payload and echo that PDU back to the sender as an EchoReply PDU.

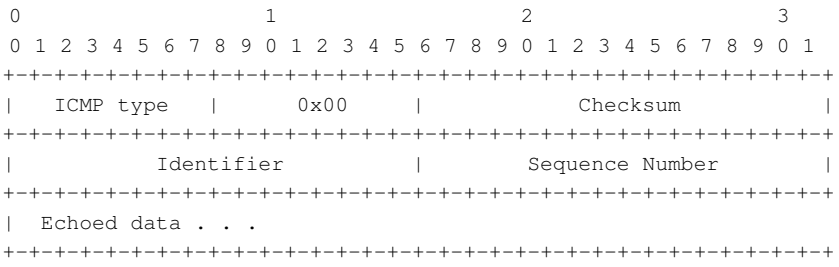


Figure 10 – Echo (AKA EchoEchoRequest) and EchoReply ICMPv4 PDU structure

Support for receiving Echo PDUs and sending EchoReply PDUs is mandatory.

4.2.10 Structure of Timestamp (AKA TimestampRequest) and TimestampReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 11; they are used to measure round-trip delays through an IPv4 implementation and the interconnecting data-link mechanisms. The Originate timestamp is added by the originating implementation at NPDU formation or when the NPDU is queued for transmission; the Receive and Transmit timestamps are added by the replying implementation at NPDU receipt and when the NPDU is queued for transmission as a reply, respectively.

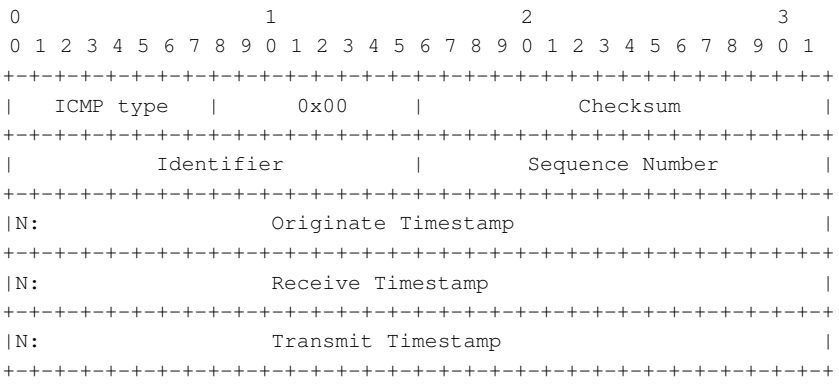


Figure 11 – Timestamp (AKA TimestampRequest) and TimestampReply ICMPv4 PDU structure

Support for receiving Timestamp PDUs and sending TimestampReply PDUs is recommended but optional.

4.2.11 Structure of InformationRequest and InformationReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 12; they are used to request (from a server) specific information about the subnetwork to which the device is attached, and to receive the requested information in reply. Per RFC1812, 4.3.3.7, they are now considered obsolete and should not be supported.

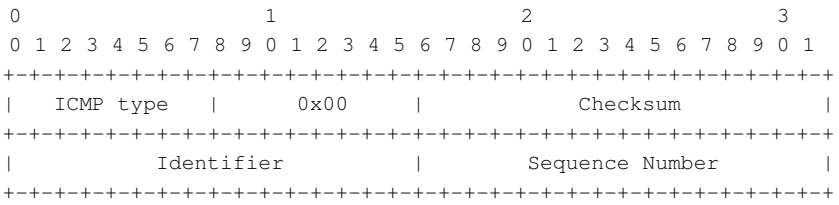


Figure 12 – InformationRequest and InformationReply ICMPv4 PDU structure

4.2.12 Structure of AddressMaskRequest and AddressMaskReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 13; they are used to request (from a server) the address mask for the subnetwork to which the device is attached, per RFC950, and to receive the requested address mask in reply.

Support for receiving AddressMaskReply PDUs and sending AddressMaskRequest PDUs is recommended but optional.

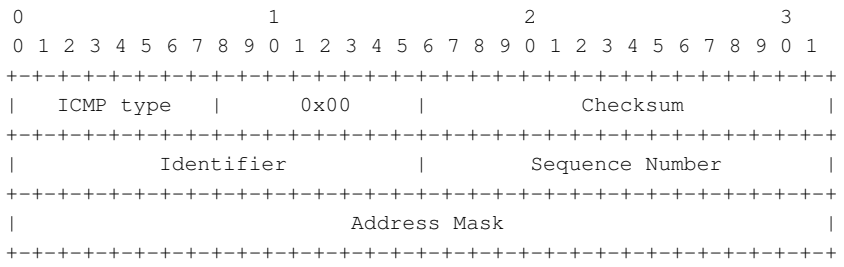


Figure 13 – AddressMaskRequest and AddressMaskReply ICMPv4 PDU structure

4.2.13 Structure of RouterSolicitation ICMPv4 PDUs

These PDUs are structured as shown in Figure 14; they are used to request (from servers) the addresses of routers on the subnetwork to which the device is attached, per RFC1256.

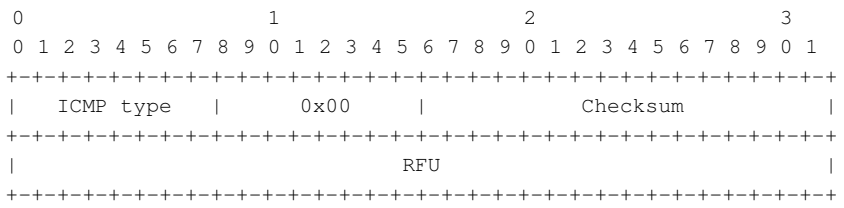


Figure 14 – RouterSolicitation ICMPv4 PDU structure

Support for sending RouterSolicitation PDUs is recommended but optional.

4.2.14 Structure of RouterAdvertisement ICMPv4 PDUs

These PDUs are structured as shown in Figure 15; servers use them to send addresses and relative preference weights of routers on the attached subnetwork, per RFC1256.

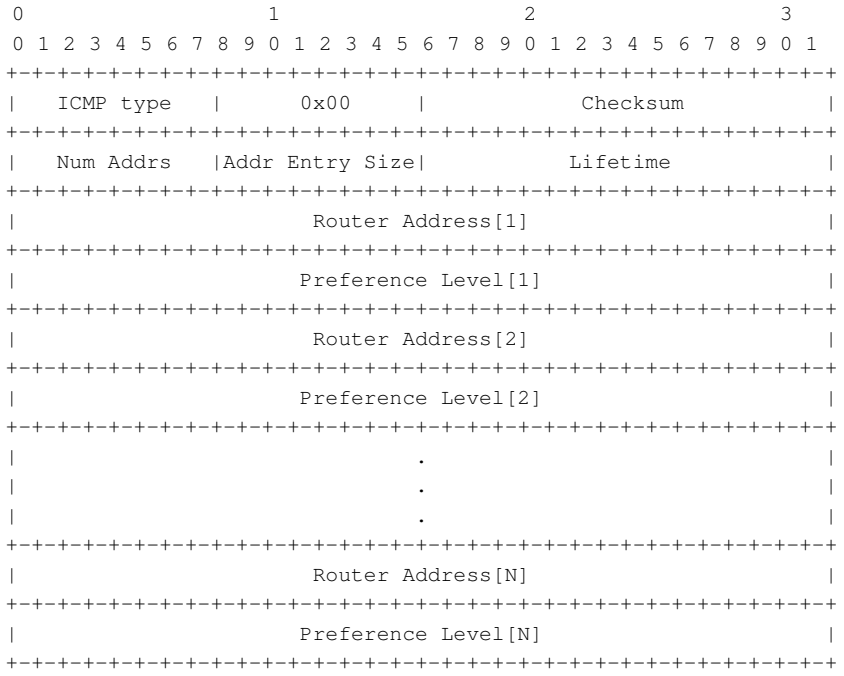


Figure 15 – RouterAdvertisement ICMPv4 PDU structure

Support for receiving RouterAdvertisement PDUs is recommended but optional.

4.2.15 Structure of DomainNameRequest and DomainNameReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 16 and Figure 17; they are used to request (from a server) the address mask for the subnetwork to which the device is attached, per RFC1788, and to receive the requested address mask in reply.

NOTE Use of these PDUs is considered experimental.

Support for receiving DomainNameReply PDUs and sending DomainNameRequest PDUs is recommended but optional. It would be mandatory if they were an approved standard rather than an experimental one.

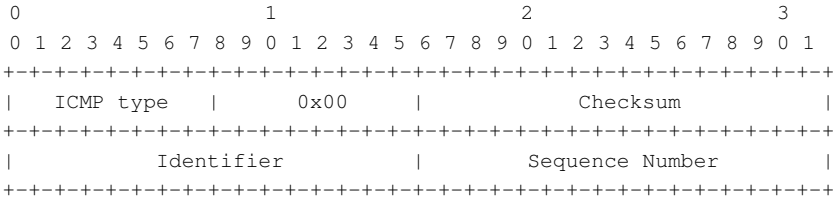


Figure 16 – DomainNameRequest ICMPv4 PDU structure

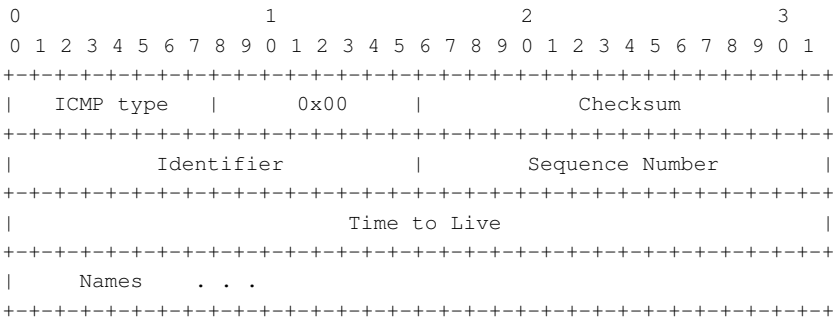


Figure 17 – DomainNameReply ICMPv4 PDU structure

4.2.16 Structure of Traceroute ICMPv4 PDUs

These PDUs are structured as shown in Figure 18. Each device that forwards an IPv4 packet, where the packet specifies a trace route option, may use this PDU to reply to the packet originator with additional route tracing information, per RFC1393.

NOTE Use of these PDUs is considered experimental; they are unlikely to be implemented by a DUT.

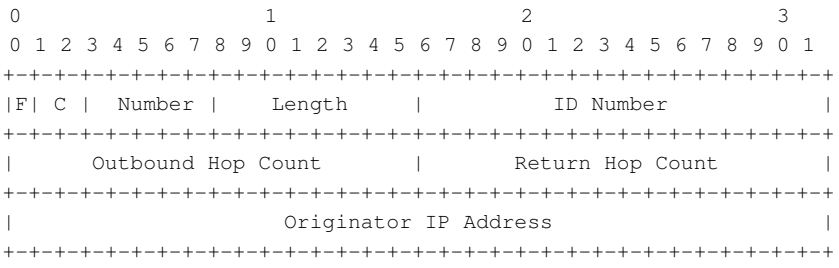


Figure 18 – Traceroute ICMPv4 PDU structure

Support for sending and receiving Traceroute PDUs is optional but not recommended.

4.3 ICMPv4 elements of procedure

4.3.1 Common elements of procedure

Common mandatory and optional elements of procedure are:

M1) Any received ICMPv4 PDU of unknown or unsupported type SHALL be discarded without notification.

M2) Any received ICMPv4 PDU with a multicast or broadcast source IP address SHALL be discarded without notification.

M3) Any sent ICMPv4 error report or route control command PDU, as listed in 4.2.1, 1) or 2), SHALL include the entire IPv4 header and at least the first 8 data octets (if present) of the IPv4 NPDU that triggered the error report.

C1) Any sent ICMPv4 error report or route control command PDU, as listed in 4.2.1, 1) or 2), SHOULD include as many additional data octets as possible of the IPv4 NPDU that triggered the error report, without causing the ICMP message to exceed the minimum IPv4 reassembly buffer size of 576 octets. (RFC1812)

C2) An ICMPv4 error report or route control command PDU, as listed in 4.2.1, 1) or 2), MAY contain a standardized extension header. (RFC4884) No processing of such a header is required.

M4) Any sent ICMPv4 error report or route control command PDU, as listed in 4.2.1, 1) or 2), SHALL be sent in an IPv4 NPDU whose DifferentiatedServices field has the value zero.

M5) Any received ICMPv4 error report that is passed to the transport layer SHALL be passed to the transport entity specified by the ProtocolType field specified in the echoed IPv4 header.

M6) An ICMPv4 error report PDU, as listed in 4.2.1, 1), SHALL NOT be sent as a result of receiving:

a) an ICMP error report;

b) an IPv4 NPDU that fails IP header validation (except where the relevant IETF specification text specifically permits sending an ICMP error PDU);

c) an IPv4 NPDU addressed to an IP multicast (including broadcast) address;

d) an IPv4 NPDU received on a data-link layer broadcast;

NOTE This probably should apply also to data-link layer multicast, which the IETF specifications do not address.

e) an IPv4 NPDU where ‘silent discard’ is the required response action;

f) an IPv4 NPDU containing a non-initial fragment (i.e., one whose FragmentOffset is not zero);

g) an IPv4 NPDU whose SourceIPv4_Address has a network prefix of zero or is an invalid source address;

h) an IPv4 NPDU whose DestinationIPv4_address does not define a unique host (e.g., a zero address, a loopback address, a multicast (including broadcast) address, or a Class E address).

M7) If the received IPv4 NPDU that provokes the sending of an ICMP error PDU contains a source-route option, the ICMP error NPDU SHOULD also contain a source-route option of the same (strict or loose) type, created by reversing the portion before the pointer of the route recorded in the source-route option of the original packet, unless

— the ICMP error message is an ICMP Parameter Problem reporting a source-route option problem in the original packet, or

— the router is aware of policy that would prevent the delivery of the ICMP error PDU.

4.3.2 Elements of procedure for DestinationUnreachable ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) The ReasonCode field of the PDU SHALL take one of the values specified in Table 1.

Table 1 – DestinationUnreachable reason codes

Value	Meaning
0	Net unreachable
1	Host unreachable
2	Protocol unreachable
3	Port unreachable
4	Fragmentation needed and DF set
5	Source route failed
6	Destination network unknown
7	Destination host unknown
8	Source host isolated
9	Communication with destination network administratively prohibited
10	Communication with destination host administratively prohibited
11	Network unreachable for type of service
12	Host unreachable for type of service

C1) A host SHOULD generate Destination Unreachable PDUs with code

2 (protocol unreachable): when the designated transport ProtocolType is not supported; or

3 (port unreachable): when the designated ProtocolType (e.g., UDP or TCP) is unable to demultiplex the received IPv4 NPDU (e.g., the destination port is CLOSED) but has no protocol-specific mechanism to inform the sender.

M2) A DestinationUnreachable message that is received with code

0 (net unreachable),

1 (host unreachable), or

5 (source route failed)

may result from a routing transient and therefore SHALL be interpreted as only a hint, not proof, that the specified destination is unreachable.

4.3.3 Elements of procedure for TimeExceeded ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) The ReasonCode field of the PDU SHALL take one of the values specified in Table 2.

Table 2 – TimeExceeded reason codes

Value	Meaning
0	time to live exceeded in transit
1	fragment reassembly time exceeded

4.3.4 Elements of procedure for SecurityFailure ICMPv4 PDUs

NOTE Per RFC1788, these PDUs are experimental and have been since 1995. Since they have not yet been approved as an IETF standard, it is likely that they are seldom implemented. They are included here only to explain their existence should a DUT originate or receive them.

PDU-class specific mandatory and optional elements of procedure are:

M1) The ReasonCode field of the PDU SHALL take one of the values specified in Table 3.

Table 3 – SecurityFailure reason codes

Value	Meaning
0	bad SPI
1	authentication failed
2	decompression failed
3	decryption failed
4	need authentication
5	need authorization

M2) The “Offset of first erroneous SPI” field of the PDU SHALL specify the offset into the “original IPv4 NPDU” field of the PDU of the (first) erroneous SPI, or zero when no SPI is present or when the offset would be greater than the number of original IPv4 NPDU octets conveyed in this error report.

4.3.5 Elements of procedure for ParameterProblem ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

C1) A host SHOULD generate ICMPv4 parameter-problem PDUs.

4.3.6 Elements of procedure for SourceQuench ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) If an ICMP source-quench PDU is received, the IPv4 implementation SHALL report it to the superior protocol entity addressed by the ProtocolType of the IPv4 NPDU.

C1) A host MAY send an ICMP source-quench PDU if it is approaching, or has reached, the point at which it is forced to discard received IP NPDUs due to a shortage of reassembly buffers or other resources.

4.3.7 Elements of procedure for Redirect ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) The ScopeCode field of the PDU SHALL take one of the values specified in Table 4

Table 4 – Redirect scope

Value	Meaning
0	redirect for the network
1	redirect for the host
2	redirect for the type of service and network
3	redirect for the type of service and host

M1) A host receiving an ICMP Redirect PDU SHALL update its routing information accordingly. Every host SHALL be prepared to accept both ICMP host-redirect and network-redirect PDUs and to process them as described in RFC1122, 3.3.1.2.

C1) A host SHOULD NOT send an ICMP Redirect PDU; Redirect PDUs are to be sent only by gateways.

C2) An ICMP Redirect PDU SHOULD be discarded without notification if the new gateway address it specifies is not on the same connected (sub)net through which the Redirect PDU arrived, or if the source of the Redirect PDU is not the current first-hop gateway for the specified destination (see RFC1122, 3.3.1).

4.3.8 Elements of procedure for Echo (AKA EchoRequest) and EchoReply ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) Every host SHALL implement an ICMP Echo server function that receives ICMP Echo Request PDUs and sends corresponding Echo Reply PDUs.

C1) A host SHOULD also implement an application-layer interface for sending an Echo Request PDU and receiving an Echo Reply PDU, for diagnostic purposes.

C2) An ICMP Echo Request (AKA Echo) PDU destined to an IP broadcast or multicast address MAY be discarded without notification.

M2) The SourceIPv4_Address in an ICMP Echo Reply PDU SHALL be the same as the specific-destination address (defined in RFC1122, 3.2.1.3) of the corresponding ICMP Echo Request PDU.

M3) All data received in an ICMP Echo Request PDU SHALL be included in the resulting Echo Reply PDU. However, if sending the Echo Reply PDU requires sender fragmentation and fragmentation is not implemented in the replying device, the Echo Reply PDU SHALL be truncated to the maximum transmission size of the replying device before sending.

M4) An Echo Reply PDU that is received in response to an Echo Request PDU that was originated at a user interface SHALL be passed to that requesting user at that interface.

C3) If an ICMP record-route option, with or without a time-stamp component, is received in an ICMP Echo Request PDU, this option SHOULD be updated to include the current host and included in the IP header of the Echo Reply PDU without "truncation", thus recording the entire round-trip route.

M5) If a source-route option is received in an ICMP Echo Request PDU, the contained traversed route SHALL be reversed and used as a source-route option for the ICMP Echo Reply PDU.

4.3.9 Elements of procedure for Timestamp (AKA TimestampRequest) and TimestampReply ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

C1) A host MAY implement IPv4 Timestamp Request and Timestamp Reply PDUs.

If these PDUs are implemented, the following rules SHALL be followed:

C2) The ICMP Timestamp server function returns a Timestamp Reply PDU to every Timestamp PDU that is received. If this function is implemented, it SHOULD be designed for minimum variability in delay (e.g., implemented in the kernel to avoid delay in scheduling a user process).

NOTE 1 The following cases for Timestamp PDUs are handled similar to the corresponding rules for ICMP Echo PDUs:

C3) An ICMP Timestamp Request PDU to an IP broadcast or multicast address MAY be discarded without notification.

M1) The source IP address in an ICMP Timestamp Reply PDU SHALL be the same as the specific-destination address (defined in RFC1122, 3.2.1.3) of the corresponding Timestamp Request PDU.

M2) If an ICMP record-route option, with or without a time-stamp component, is received in an ICMP Timestamp Request PDU, this option SHOULD be updated to include the current host and included in the IP header of the Timestamp Reply PDU without "truncation", thus recording the entire round-trip route.

M3) If a Source-route option is received in an ICMP Timestamp Request PDU, the contained traversed route SHALL be reversed and used as a Source Route option for the Timestamp Reply PDU.

M4) Received Timestamp Reply PDUs SHALL be passed up to the ICMP user interface.

M5) A timestamp value SHALL be expressed as milliseconds since midnight.

C4) The timestamp SHOULD be referenced to midnight UTC.

M6) The clock used for the timestamp SHALL be updated at a rate of at least 12,5 Hz.

NOTE 2 The original IETF specification of 15 Hz is U.S/Canada/Japan-centric. The replacement minimum rate of 12,5 Hz provides equivalent functionality whether the site power grid is 50 Hz or 60 Hz

4.3.10 Elements of procedure for InformationRequest and InformationReply ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

C1) A non-router host SHOULD NOT implement these PDUs.

NOTE Per RFC1812, 4.3.3.7, these PDUs are now considered obsolete.

4.3.11 Elements of procedure for AddressMaskRequest and AddressMaskReply ICMPv4 PDUs

PDU-class specific mandatory and optional elements of procedure are:

M1) A host SHALL support the first, and MAY implement all three, of the following methods for determining the address mask(s) corresponding to its IP address(es):

1) static configuration information;

2) obtaining the address mask(s) dynamically as a side-effect of the system initialization process;

3) sending ICMP AddressMaskRequest PDUs and receiving ICMP AddressMaskReply PDUs.

The choice of method to be used in a particular host SHALL be configurable.

M2) When method (3), the use of Address Mask PDUs, is enabled, then:

a) When it initializes, the host SHALL broadcast an AddressMaskRequest PDU on the connected network corresponding to the IP address. It SHALL retransmit this PDU a small number of times if it does not receive an immediate AddressMaskReply PDU.

b) Until it has received an Address Mask Reply PDU, the host SHOULD assume a mask appropriate for the address class of the IP address, i.e., assume that the connected network is not subnetted.

c) The first AddressMaskReply PDU received SHALL be used to set the address mask corresponding to the particular local IP address. This is true even if the first Address Mask Reply PDU is "unsolicited", in which case it will have been broadcast and may arrive after the host has ceased to retransmit Address Mask Request PDUs. Once the host's address mask has been set by an Address Mask Reply PDU, later Address Mask Reply PDUs SHALL be ignored, without notification.

NOTE Clearly this mandated behavior – unconditional trust in the first address mask received in an Address Mask Reply PDU, whether solicited or not – provides an opportunity for (perhaps meaningless) attack.

M3) Conversely, if Address Mask PDUs are disabled, then no ICMP AddressMaskRequest PDUs SHALL be sent, and any ICMP AddressMaskReply PDUs received for that local IP address SHALL be ignored, without notification.

M4) A host SHOULD make some reasonableness check on any address mask it installs.

M5) A system SHALL NOT send an AddressMaskReply PDU unless it is an authoritative agent for address masks. An authoritative agent may be a host or a gateway, but it SHALL be explicitly configured as an address mask agent. Receiving an address mask via an AddressMaskReply PDU does not give the receiver authority and SHALL NOT be used as the basis for issuing other AddressMaskReply PDUs.

M6) With a statically configured address mask, there SHOULD be additional configuration data that determines whether the host is to act as an authoritative agent for this mask, i.e., whether it will answer AddressMaskRequest PDUs with AddressMaskReply PDUs using this mask.

M7) If it is configured as an agent, the host SHALL broadcast an AddressMaskReply PDU for the mask on the appropriate interface when the host initializes on that interface.

M8) The Identifier and Sequence Number in the AddressMaskReply PDU SHALL equal that in the corresponding ICMP AddressMaskRequest PDU.

4.3.12 Elements of procedure for RouterSolicitation ICMPv4 PDUs

Per RFC1256, PDU-class specific mandatory and optional elements of procedure are:

M1) These PDUs SHALL be addressed to the IPv4 multicast address 224.0.0.2 (all-routers) or the IPv4 limited-broadcast address 255.255.255.255.

M2) A router SHALL reply to receipt of such a PDU at one of the two addresses specified in M1) by sending a valid RouterAdvertisement PDU, either

d) as a unicast reply sent to the requesting source address or

e) as a multicast reply sent to the router's configured AdvertisementAddress, which SHOULD BE either the IPv4 multicast address 224.0.0.1 (all-systems) or the IPv4 limited-broadcast address 255.255.255.255.

In either case, the replying router SHALL apply a small delay chosen from a uniform-random distribution before replying, to minimize collisions should multiple routers reply to the same request.

M3) A non-router host SHALL NOT reply to these messages.

4.3.13 Elements of procedure for RouterAdvertisement ICMPv4 PDUs

Per RFC1256, PDU-class specific mandatory and optional elements of procedure are:

M1) A non-router host SHALL NOT originate these PDUs.

M2) A router SHALL initialize its IPv4 interface to receive multicast address 224.0.0.2 (all-routers).

M3) A router SHALL respond to receipt of a RouterSolicitation PDU at the multicast address 224.0.0.2 (all-routers) via either

f) a unicast reply sent to the requesting source address, or

g) a multicast reply sent to the router's configured AdvertisementAddress.

M4) A router SHALL respond to receipt of a RouterSolicitation PDU at the limited-broadcast address 255.255.255.255 via either

a) a unicast reply sent to the requesting source address, or

b) a multicast reply sent to the IPv4 limited-broadcast address 255.255.255.255.

In either case, the replying router SHALL apply a small delay chosen from a uniform-random distribution before replying, to minimize collisions should multiple routers reply to the same request.

M5) Quasi-periodically, a router SHALL send RouterAdvertisement PDUs to its configured AdvertisementAddress, which SHOULD BE either the IPv4 multicast address 224.0.0.1 (all-systems) or the IPv4 limited-broadcast address 255.255.255.255.

M6) The router SHALL dither the periodicity of sending RouterAdvertisement PDUs, using a uniform-random distribution as configured for the router, except that the maximum periodicity SHALL be reduced for the first few such PDUs sent after router startup.

M7) The minimum periodicity distribution SHALL be [3..4] s; the default distribution SHALL be [450..600] s.

M8) The value of the Lifetime field in each such PDU SHALL be at least as great as the configured maximum advertisement interval, but in all cases no greater than 9000 s; the default Lifetime shall be three times the configured maximum advertisement interval.

4.3.14 Elements of procedure for AddressMaskRequest ICMPv4 PDUs

NOTE Per RFC2521, these PDUs are experimental and have been since 1999. Since they have not yet been approved as an IETF standard, it is likely that they are seldom implemented. They are included here only to explain their existence should a DUT originate or receive them.

PDU-class specific mandatory and optional elements of procedure are:

M1) The destination address of the conveying IPv4 NPDU shall be a unicast address.

4.3.15 Elements of procedure for AddressMaskReply ICMPv4 PDUs

NOTE Per RFC2521, these PDUs are experimental and have been since 1999. Since they have not yet been approved as an IETF standard, it is likely that they are seldom implemented. They are included here only to explain their existence should a DUT originate or receive them.

PDU-class specific mandatory and optional elements of procedure are:

M1) The destination address in the conveying AddressMaskReply NPDU SHALL equal the source address in the IPv4 NPDU that conveyed the corresponding AddressMaskRequest ICMPv4 PDU.

M2) The Identifier and Sequence Number in the PDU SHALL equal that in the corresponding AddressMaskRequest ICMPv4 PDU.

M3) Since the “Time to Live” value is a 2’s-complement representation, values of zero or less are invalid and SHALL NOT be sent.

M4) The Names field SHALL consist of zero or more complete fully-qualified domain names.

4.3.16 Elements of procedure for Traceroute ICMPv4 PDUs

NOTE Per RFC1393, these PDUs are experimental and have been since 1993. Since they have not yet been approved as an IETF standard, it is likely that they are seldom implemented. They are included here only to explain their existence should a DUT originate or receive them.

PDU-class specific mandatory and optional elements of procedure are:

C1) Per RFC1393, a router that forwards an IPv4 NPDU that contains a Traceroute option MAY send a Traceroute ICMPv4 PDU that does not contain a Traceroute option to the source address of that original triggering NPDU.

C2) Receipt of a Traceroute PDU, as a response by an intermediate router when forwarding an IPv4 NPDU that contains a Traceroute option, SHOULD NOT be considered an error by the receiving device.

5 Elements of other protocols required for the testing

5.1 Protocol(s) from inferior layers used by this protocol

This specification requires that IMCPv4 PDUs are conveyed by IPv4 NPDUs. An ICMPv4 implementation is expected to be insensitive to the selection of protocols below IPv4.

5.2 Protocol(s) from superior layers used to test this protocol

none

6 Robustness testing

6.1 Goals that drive testing requirements

The goal of the tests described in this document is to assess:

a) the robustness of an embedded control device with an implemented set of protocols, and

b) the device’s resistance to attack, including the impact on the device’s reporting and control functions while sustaining such an attack.

It is not a goal to determine the correctness of the implementation of those protocols, which would be a measure of their conformance to the requirements of the various protocol specifications.

This atypical testing goal interacts with vendor decisions to provide only partial implementations of protocols that are used within a proprietary or constrained context, such that those implementations are completely functional within the usage limits imposed by that context but are not conformant to the mandatory requirements of the controlling protocol standard.

As described by specific requirements in [EDSA-310], the consequent requirement is for this testing to

1) ascertain whether the DUT and other parts of the test configuration meet normal operational expectations before testing commences;

2) determine whether the DUT can survive receipt of invalid frames while continuing to function as expected in an automation environment; and

3) determine whether the DUT can sustain intervals of high and excessive communications load.

6.2 Testing overview

The DUT and its communications environment (e.g., any intervening firewalls) must be preconditioned to support testing by

1) meeting the requirements of [EDSA-310] for demonstrating continued correct operation during testing;
2) preparing the DUT and other devices in its test environment to not block or discard generated ICMPv4 PDUs, .

Robustness testing occurs in three conceptual phases that may overlap, plus a test environment preconditioning phase.

a) The first conceptual phase, Baseline operation, attempts to demonstrate that the selected DUT protocol suite used for testing appears to operate properly for simple test cases under low load, before any protocol fuzzing or stress testing is attempted.

   NOTE 1 This initial demonstration of apparently correct behavior establishes the presumption that failure during additional testing is due to vulnerabilities of the specific protocol under test, rather than other protocols in the test suite.

b) The second conceptual phase, Basic robustness testing, probes the implementation for its ability to not evidence harm due to receipt of arbitrary erroneous frames, either singly or in combination.

   NOTE 2 This conceptual phase focuses on simple protocol robustness/fuzzing tests.

c) The third conceptual phase, Load stress testing, probes the implementation's response to high traffic rates incorporating valid PDUs.

   NOTE 3 This conceptual phase focuses on load/performance tests. The latter are always capable of driving the communications stack into overload and functional collapse.

Although the robustness testing of this specification is conceptualized as occurring in distinct logical phases that progress from simple single-factor testing to more complex load testing incorporating PDUs with varying characteristics, there is no requirement that an actual robustness test process work in this ordered, sequential manner; any order of testing is permitted provided that the selected order does not lead to incorrect conclusions about robustness.

Requirement ICMPv4.R1 – Criteria for robustness test failure

Pass or fail of basic robustness and load stress testing SHALL be determined by:

- whether or not essential services are adequately maintained under network traffic conditions created under these tests, as defined in [CRT.Essential_services];
- any particular conditions resulting in pass/fail mandated by the testing specified in this document.

The ICMPv4 protocol that is the subject of this specification is a stateless protocol, part of which uses a simple client/server query/immediate-response mechanism. For that portion of the protocol, the DUT responds to remote-originated received ICMPv4 PDUs, enabling its network behavior to be observed.

Some received ICMPv4 PDUs are intended to affect the routing database used by the related IPv4 protocol implementation. In general, the TD can observe whether the DUT has made each such change by noting the DUT's routing behavior both before and after it has been sent the ICMPv4 message that should have effected that change.

NOTE 4 Such observations are not necessary for robustness testing, but may provide useful for diagnosing other implementation faults.

6.3 Protocol stack used for testing

6.3.1 Protocol(s) from inferior layers used by this protocol

IPv4, the co-protocol of ICMPv4, is used to convey ICMPv4 PDUs. There are no other dependencies.

6.3.2 Protocol(s) from superior layers used to test this protocol

Testing of ICMPv4 robustness requires only transmission of ICMPv4 test NPDUs to the DUT. Thus there is no requirement for a superior layer protocol during ICMPv4 robustness testing.

6.4 Phase 0: DUT preconditioning

Requirement ICMPV4.R2 – Preconditioning of DUT,TD and any firewalls between the DUT and TD

The DUT SHALL be preconditioned for robustness testing, typically by

a) configuring the DUT's IPv4 implementation with appropriate IPv4 network addresses;

b) enabling the bidirectional forwarding of ICMPv4 PDUs through any intermediary hardware or software firewalls, if such forwarding is normally disabled as a security precaution;

c) configuring the DUT, the TD(s) and possibly other devices in the test system to allow observation of the performance of *essential services* of the embedded device under the test conditions, per the requirements in [CRT.Essential_services].

Essential services as defined in [CRT.Essential_services] include the control loop, commands to control device configuration such as setpoints, and process alarms. A key approach to obtain observability is to use, as part of the test configuration, existing higher layer system elements that have been engineered to communicate with and monitor the DUT.

6.5 Phase 1: Baseline operation

6.5.1 General

Requirement ICMPV4.R3 – Demonstration of baseline operation

Before the TD commences robustness testing, the DUT shall demonstrate its ability to operate as expected in the test environment, including that the ICMPv4 and IPv4 components of the DUT's protocol stack are present and functioning, and that the DUT can maintain essential services.

6.5.2 Presence of proprietary protocol extensions

It is common practice for vendors to extend a standard protocol in a proprietary manner to provide functionality not covered by the standard protocol, or to provide more efficient or more constrained data transport for specific device information (e.g., when multiple device parameters require atomic update or readout as a group to maintain their inter-parameter consistency). Such extensions may take the form of extra message types, extra fields in standard messages, or extra functionality for standard fields in standard messages.

ICMPV4 already includes a standard mechanism for extending standard messages as specified in 4.2.2. However, this is a relative recent addition to ICMPv4, so it is possible that alternate extensions also may be encountered.

NOTE Robustness testing is not required to include specialized testing of proprietary protocol extensions. Rather, vendor disclosure of such extensions is intended to provide a basis for explanation of otherwise anomalous test results.

Requirement ICMPV4.R4 – Equipment vendor disclosure of proprietary protocol extensions

When a protocol offered for testing has been implemented with deliberate proprietary extensions, the vendor SHALL document the extensions in a manner similar to that of Clause 4, such that robustness testing can explore the intended and unintended consequences of those protocol extensions. It is acceptable that access to this proprietary information be covered by a non-disclosure agreement (NDA) between the equipment vendor and the organization that is providing the ISCI robustness testing service.

6.6 Phase 2: Basic robustness testing

6.6.1 General

Areas of specific robustness testing are identified by analysis of the controlling protocol standards. These include identification of all field value ranges and of the bounding values of the underlying message representation (e.g., a range of 10..100 in a one-byte field, whose underlying representational bounding values are 0..255). Basic robustness testing includes testing the acceptability of each of these bounding values, and of the acceptance or rejection of adjacent values to those bounding values when such adjacent values can be represented in the message encoding. It also includes testing whether fields specified to convey signed or unsigned values are distinguished and processed appropriately.

Conceptually, basic robustness testing consists of the following, where volume or rate of message traffic is not a factor:

a) tests of valid message traffic:
   1) in expected sequences, sent at a low rate;
   2) in unexpected but valid sequences sent at a low rate (i.e., where the messages would be considered valid for the protocol under some conditions, but are not expected for the particular protocol state, message sequence or relative time);
b) tests of low rate erroneous message traffic (e.g., the ability of the device to function after receiving erroneous messages), including:
   1) single erroneous messages, including messages with inconsistent field values;
   2) properly formed messages in erroneous sequences
   3) sequences of erroneous messages.

[EDSA-310] describes the criteria for adequate performance of device essential services under these network traffic conditions. These criteria depend upon the specific service as well as whether the service operates on the same network interface used for test traffic.

6.6.2 Specific basic robustness testing

6.6.2.1 DUT receipt of erroneous ICMPv4 PDUs

Requirement ICMPv4.R5 – Non-failure after receipt of erroneous ICMPv4 PDUs

RFC792 states that an ICMPv4 error report PDU should not be sent in response to receipt of another ICMPv4 PDU. This limits the extent to which ICMPv4 PDUs alone can be used to test the DUT's ICMPv4 implementation.

The TD SHALL explore the DUT's apparent resilience to receipt of pseudo ICMPv4 PDUs whose PDU type is not one of the defined ICMPv4 PDU types (i.e., the PDU type is an invalid value).

NOTE 1 Per RFC792, a response SHOULD NOT be observed.

The TD SHALL explore the DUT's apparent resilience to receipt of malformed ICMPv4 PDUs of defined ICMPv4 PDU types, and of properly formed ICMPv4 PDUs with field contents that violate field value range restrictions.

NOTE 2 Ibid

The TD SHALL explore the DUT's apparent resilience to receipt of correctly formed Destination unreachable and Time exceeded PDUs with various invalid values for the Reason code, and of Parameter problem PDUs with various invalid values for the Error index.

Requirement ICMPv4.R6 – Non-failure after receipt of PDUs of contextually inappropriate PDU type

The TD SHALL explore the DUT's apparent resilience to receipt of correctly formed unsolicited Echo reply, Timestamp reply, Address mask request, Router solicitation, Information request, Information reply and Traceroute PDUs.

NOTE 3 Ibid

Requirement ICMPv4.R7 – Non-failure after receipt of contextually inappropriate error PDUs

The TD SHALL explore the DUT's apparent resilience to receipt of correctly formed and malformed Destination unreachable, Time exceeded and Security failure PDUs with various valid and invalid values for the Reason code, and of Parameter problem PDUs with various valid and invalid values for the Error index,

Requirement ICMPv4.R8 – Non-failure after receipt of contextually inappropriate error PDUs with invalid field values

The TD SHALL explore the DUT's apparent resilience to receipt of correctly formed Destination unreachable, Time exceeded and, optionally, Security failure PDUs with various invalid reason codes, and of Parameter Problem PDUs with various invalid Error index values,

6.6.3 Basis for ICMPv4 robustness testing

Correctly and incorrectly formed ICMPv4 NPDUs sent to the DUT from the DUT form the basis for ICMPv4 robustness testing.

Requirement ICMPv4.R9 – Testing of each message field for sensitivity to malformed content

For basic robustness testing requiring malformed messages or message sequences, valid ICMPv4 PDUs or TPDU sequences from the TD to the DUT SHALL be altered so that one component of the ICMPv4 PDU is erroneous; or so that the ICMPv4 PDU is in violation of the relevant requirements of Clause 4; or that it is both erroneous and in violation. Any response from the DUT, other than possibly an ICMP error message addressed to the TD, indicates that the specific error did not result in the DUT's required rejection of the ICMPv4 PDU, resulting in failure of the robustness test.

Such alterations SHALL be applied to each field of the ICMPv4 PDU where alteration might have an impact on the DUT.

NOTE This type of testing can be described as single-message protocol "fuzzing".

It is suggested that basic robustness testing proceed in stages, from simple to complex, as enumerated in the relevant requirements of Clause 4 and indicated by the following list. In general, such ordering simplifies the task of locating the source(s) of software or hardware problems should they be uncovered by the testing. However, such ordering is not a requirement.

Requirement ICMPv4.R10 – Constituent elements in basic robustness tests

Basic ICMPv4 robustness testing SHALL include the following elements, at low traffic rates, either in distinct test phases or intermixed in a form of the test supplier's choosing:

a) valid message traffic;

b) erroneous messages.

6.7 Phase 3: Load stress testing

6.7.1 General

NOTE 1 This testing phase is used to ascertain resistance to busy plant conditions as well as deliberate attacks.

Conceptually, load stress testing consists of tests of valid message traffic sent in two distinct phases:

Phase 1 – Valid message traffic is sent at a high rate less than the saturation rate threshold specified by the DUT vendor (e.g., simulating normal but busy plant conditions);

Phase 2 – Valid message traffic is sent at up to the full auto-negotiated link rate (e.g., simulating an attack or malfunction of some kind);

Attacks against a protocol implementation take the form of repeated probing by malformed messages, or by correctly formed messages whose arrival sequence and relative timing are controlled by the attacker, or (more usually) by combinations thereof, all with the intent of exploiting some oversight or error in the specific protocol implementation(s), or of activating some intertwining aspects of a multi-layer protocol stack that were unconsidered by the implementing organization.

NOTE 2 Self-induced accidental attacks are also possible, due to designer or operator oversight.

Common examples of exploited oversights and errors are deliberate buffer overflows where the implementer had neglected to detect excessive message or field size, or recursive activation of character escape encoding when the implementer had not considered recursion. Implementation interactions within a multi-layer protocol stack may occur when an initial resource allocation (e.g., memory buffering) made by one protocol layer implementation is driven into an adjustment phase that conflicts with a resource allocation already made by a paired protocol layer implementation.

6.7.2 Basis for load stress testing

Device defenses against high traffic rates impact load stress testing, and are documented by the device vendor per the following requirement.

Requirement ICMPv4.R11 – Documentation of self-protective rate limiting behavior

Where the DUT vendor imposes rate limiting on one or more of the protocols in the test process (e.g., “Ethernet”, IPv4 or ICMPv4), the DUT vendor SHALL document that rate limiting occurs for that identified protocol when message rates exceed a perhaps-unspecified rate, as required by [CRT.Rate_limiting].

NOTE 1 The “Ethernet” protocol is included in this list as an identifiable placeholder for any physical and data-link protocols used to convey IPv4 NPDUs.

Requirement ICMPv4.R12 – Constituent elements in load stress tests

Load stress testing SHALL include the following elements, either in distinct test phases or intermixed in a form of the test supplier’s choosing:

a) high-rate valid message traffic;

b) over-saturation-rate version of a), at the maximum auto-negotiated link rate that the TD can support.

Requirement ICMPv4.R13 – Testing of saturation rate-limiting mechanism(s)

Saturation rate testing SHOULD be for durations of at least tens of seconds for each phase, long enough for any saturation effects to manifest. Tests that inherently involve a large number of PDUs, such as port scans, may need to run for much longer durations so that they do not cause other untoward impact on the test environment, which inherently involves the DUT, the TD and any other devices used in ascertaining the continuing performance of the DUT’s other normal functionality (e.g., interactions with superior or peer automation system components).

Requirement ICMPv4.R14 – Reproducibility of robustness testing

Basic robustness testing SHALL use a deterministic selection process (e.g., an offline test case generator or a seeded pseudo-random selection process) that tests combinations of valid and erroneous messages. See Clause 7 for specific required test cases.

Load stress testing SHALL use a deterministic selection process (e.g., an offline test case generator or a seeded pseudo-random selection process) that tests series of valid messages. See Clause 7 for specific required test cases.

NOTE 2 The above constraint to use of a deterministic selection process does not prohibit use of feedback from analysis of DUT responses (and non-responses) as a means of further varying and focusing testing. Nor does it prohibit use of tester-selectable options and modes to determine the aggressiveness of the test process. Rather, it is merely an attempt to facilitate reproducibility by requiring use of reproducible means to select the order, sequence and components of each test.

6.8 Reproducibility

Requirement ICMPv4.R15 – Overall reproducibility

Discovery, basic robustness testing, and load stress testing SHALL be reproducible per the requirements of [CRT.Reproducibility].

Those requirements recognize that deterministic behavior of the DUT itself is not under the control of the tester and must be assumed. Further, it is acceptable to branch a test process based upon prior results. Thus a change to the DUT may impact repeatability of a test even if the change does not intentionally cause variance for that test.

7 Specific test cases

Requirement ICMPv4.R16 – Specific test cases

The tested suite of protocols SHALL be documented in at least the detail specified by Table 5.

Table 5 – ICMPv4: Protocols used in test process

Protocol layer tested	Permissible alternatives	Protocols tested	Maximum load at which deliberate limiting occurs
Physical layer	IEEE 802.3		
Data-link layer	“Ethernet”		
Network layer	IPv4 plus ICMPv4		

Requirement ICMPv4.R17 – Testing SHALL include at least that specified by Table 6 through Table 14

These tables are descriptive, not proscriptive – there is no requirement that conforming robustness testing actually employ test sequences that are ordered or grouped as described in these tables.

Table 6 – ICMPv4.T00: Baseline operation

Test ID	ICMPv4.T00
Test name	Baseline operation
Test description	The basic operational aspects of the protocol under test, and of any inferior or selected superior supporting protocols used in the testing, shall be demonstrated as a means of checking that gross configuration or other errors are not interfering with the testing process, that IPv4 and ICMPv4 are a functioning part of the DUT's protocol stack, and that the protocol implementation under test performs approximately as expected when not under test
Reference requirements	Requirement ICMPV4.R3
Test type	Baseline operation
Test status	Mandatory
Expected DUT behavior	The DUT demonstrates basic protocol operability in the test configuration
Test object	To validate the lack of major errors in the configuration of the DUT and test environment
Test configuration	A TD is connected to the DUT by an underlying switched network that uses IEEE 802 and IP addressing, as specified in [CRT.Test_configuration_1]
Test procedure	The TD establishes that DUT is reachable and functions normally in the test environment, before protocol-specific testing commences
Expected DUT response	The DUT demonstrates expected behavior in its "automation" environment, including that the UDP component of the protocol stack is present and functioning and that the DUT can adequately maintain essential services
Ultimate results	Pass or fail
Remarks	Initial failure of this test indicates a probable problem with the configuration of the TD or the test environment, including configuration of any intervening firewalls to pass ICMP error reports to the TD

Table 7 – ICMPv4.T01: Undefined ICMPv4 PDU types

Test ID	ICMPv4.T01
Test name	Undefined ICMPv4 PDU types
Test description	The TD sends ICMPv4 PDUs of undefined PDU types to the DUT
Reference requirements	Requirement ICMPv4.R5
Test type	Basic robustness: PDU structural or content violations
Test status	Mandatory
Expected DUT behavior	Receipt of an ICMPv4 PDU of an undefined type results in no action by the DUT, either a change to a DUT-internal database or sending an ICMPv4 PDU in response
Test object	To evaluate the DUT's parsing of ICMPv4 PDUs and protection against malformed PDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses ICMPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs
Test procedure	The TD sends to the DUT ICMPv4 PDUs with malformed content appropriate for the desired test (in this case, with an undefined type field value). The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	Any ICMPv4 error PDU sent as a response by the DUT is generally an error

Table 8 – ICMPv4.T02: Malformed ICMPv4 PDUs of defined PDU types

Test ID	ICMPv4.T02
Test name	Malformed ICMPv4 PDUs of defined PDU types
Test description	The TD sends malformed ICMPv4 PDUs of each of the defined PDU types to the DUT, with varying types of malformation for each defined type of PDU
Reference requirements	Requirement ICMPv4.R5
Test type	Basic robustness: PDU structural or content violations
Test status	Mandatory
Expected DUT behavior	Receipt of a malformed ICMPv4 PDU results in no action by the DUT, either a change to a DUT-internal database or sending an ICMPv4 PDU in response
Test object	To evaluate the DUT's parsing of ICMPv4 PDUs and protection against malformed PDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses ICMPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs
Test procedure	The TD sends to the DUT ICMPv4 PDUs with malformed content appropriate for the desired test (in this case, with undefined values in a field with a limited set of defined values, or where a variable-size structure). The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	Any ICMPv4 error PDU sent as a response by the DUT is generally an error

Table 9 – ICMPv4.T03: ICMPv4 PDUs of contextually inappropriate PDU type

Test ID	ICMPv4.T03
Test name	ICMPv4 PDUs of contextually inappropriate PDU type
Test description	The TD sends ICMPv4 PDUs of contextually inappropriate PDU types (e.g., Echo reply, Timestamp reply, Address mask request, Information request or reply, Router solicitation, Traceroute) to the DUT, with varying apparently valid field content for each type of PDU
Reference requirements	Requirement ICMPv4.R6
Test type	Basic robustness: PDU structural or content violations
Test status	Mandatory
Expected DUT behavior	Receipt of a PDU of a contextually inappropriate PDU type results in no action by the DUT, either a change to a DUT-internal database or sending an ICMPv4 PDU in response
Test object	To evaluate the DUT's parsing and processing of ICMPv4 PDUs of inappropriate PDU types
Test configuration	A TD is connected to the DUT by an underlying switched network that uses ICMPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs
Test procedure	The TD sends to the DUT ICMPv4 PDUs of an inappropriate PDU type. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	Any ICMPv4 error PDU sent as a response by the DUT is generally an error

Table 10 – ICMPv4.T04: ICMPv4 PDUs of appropriate PDU type but with invalid field content

Test ID	ICMPv4.T04
Test name	ICMPv4 PDUs of appropriate PDU type but with invalid field content
Test description	The TD sends ICMPv4 PDUs of appropriate PDU types (e.g., Echo, Timestamp, Address mask reply, Router advertisement, Destination unreachable, Time exceeded, Parameter problem, Redirect and Source quench) to the DUT, with varying invalid field content for each type of PDU
Reference requirements	Requirement ICMPv4.R5
Test type	Basic robustness: PDU structural or content violations
Test status	Mandatory
Expected DUT behavior	Receipt of a PDU of an appropriate PDU type with invalid field content results in no action by the DUT, either a change to a DUT-internal database or sending an ICMPv4 PDU in response
Test object	To evaluate the DUT's parsing and processing of ICMPv4 PDUs with erroneous field content
Test configuration	A TD is connected to the DUT by an underlying switched network that uses ICMPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs
Test procedure	The TD sends to the DUT ICMPv4 PDUs of appropriate PDU type but with invalid field content. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	Any ICMPv4 error PDU sent as a response by the DUT is generally an error

Table 11 – ICMPv4.T05: Rejection of NPDUs with multicast or broadcast source IP addresses

Test ID	ICMPv4.T05
Test name	Rejection of NPDUs with multicast or broadcast source IP addresses
Test description	ICMPv4 PDUs are sent with multicast or broadcast source IP addresses
Reference requirements	Requirement ICMPv4.R5, violating 4.3.1 M2
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT ignores and discard, without notification, NPDUs sent with a multicast or broadcast source IP address
Test object	To evaluate the DUT's protective measures in situations where the source IP address of a received NPDU is determinably invalid
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHALL be enabled at any intervening firewall(s)
Test procedure	The TD sends ICMPv4 PDUs in IP NPDUs whose source address is a multicast or broadcast IP address. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	This test exposes failures to check received NPDUs for invalid source IP addresses

Table 12 – ICMPv4.T06: Rejection of IP multicasts and broadcasts

Test ID	ICMPv4.T06
Test name	Rejection of IP multicasts and broadcasts
Test description	ICMPv4 PDUs are sent with multicast or broadcast destination IP addresses
Reference requirements	Requirement ICMPv4.R5, violating 4.3.1 M6 c)
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT ignores and discard, without notification, ICMPv4 PDUs received at a multicast or broadcast IP address
Test object	To evaluate the DUT's protective measures in situations where the destination IP address of a received NPDU from a spoofed source could cause the DUT to participate as an unintentional subordinate attacker in a distributed denial-of-service attack
Test configuration	A TD is connected to the DUT by an underlying switched network that uses either IPv4 or IPv6 addressing, as specified in [CRT.Test_configuration_1]. ICMP error reporting by the DUT SHALL be enabled at any intervening firewall(s)
Test procedure	The TD sends ICMPv4 PDUs in IP NPDUs whose destination address is a multicast or broadcast address. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	This test exposes failures to react protectively to receipt of ICMPv4 PDUs in NPDUs addressed to a potentially large number of network nodes

Table 13 – ICMPv4.T07: Contextually inappropriate error PDUs

Test ID	ICMPv4.T07
Test name	Inappropriate error PDUs
Test description	The TD sends ICMPv4 error PDUs to the DUT that are contextually inappropriate, in that they are not correct responses to DUT activity
Reference requirements	Requirement ICMPv4.R5, Requirement ICMPv4.R6 and Requirement ICMPv4.R8
Test type	Basic robustness: contextually inappropriate PDUs
Test status	Mandatory
Expected DUT behavior	Receipt of a contextually inappropriate ICMPv4 PDU results in no action by the DUT, either a change to a DUT-internal database or sending an ICMPv4 PDU in response
Test object	To evaluate the DUT's parsing of ICMPv4 PDUs and defense against contextually inappropriate PDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses ICMPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs
Test procedure	The TD to the DUT sends contextually inappropriate ICMPv4 PDUs. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential services
Results	Pass or fail
Remarks	Any ICMPv4 error PDU sent as a response by the DUT is generally an error

Table 14 – ICMPv4.T08: Maintenance of service under high load, including network saturation: Raw ICMPv4 NPDU flood

Test ID	ICMPv4.T08
Test name	Maintenance of service under high load, including network saturation: Raw ICMPv4 NPDU flood
Test description	A flurry of ICMPv4 NPDUs is sent to the DUT to attempt to overwhelm the DUT's receive processing and storage resources. This test proceeds in two phases: • Phase 1: as a high load test during which the DUT SHOULD respond normally to received messages • Phase 2: as a network saturation test during which the DUT MAY invoke protective behaviors such as blocking network reception but SHOULD otherwise function normally. See [CRT.Rate_limiting] for additional requirements
Reference requirements	Requirement ICMPv4.R12
Test type	Load stress
Test status	Mandatory
Expected DUT behavior	The DUT protects itself against a flood of received ICMPv4 NPDUs. • Phase 1: The DUT continues to function, adequately maintaining all essential services, in the presence of a sudden burst of received ICMPv4 NPDUs, provided that the load thus induced is less than that claimed as supportable by the DUT vendor; • Phase 2: The DUT adequately maintains essential control, even if it must reduce or cease other essential services during the period of network overload.
Test object	To evaluate the DUT's ability to receive and withstand a burst of ICMPv4 NPDUs addressed to it
Test configuration	A TD is connected to the DUT by an underlying switched network that uses IPv4 addressing, as specified in [CRT.Test_configuration_1]. During testing, the filtering rules for ICMPv4 PDUs of each interposed firewall SHALL be modified to not filter such PDUs. The DUT vendor SHALL state a rate limit below which protective measures are not expected to be invoked
Test procedure	The TD sends valid ICMPv4 NPDUs of varying types, type sub-codes and options, which are either explicitly or implicitly addressed to the DUT • Phase 1: at a rate less than that at which the DUT's manufacturer claims DUT protective measures will be invoked; • Phase 2: at a rate up to the auto-negotiated maximum rate of the underlying network, maintains that high load rate for a few seconds, then gradually reduces its sending rate to zero.
Expected DUT response	• Phase 1: The DUT is expected to continue network communication even under high load while adequately maintaining essential services. • Phase 2: The DUT is expected to activate protective measures at some (vendor unspecified) level of resource demand, and to recover some reasonable time interval after that demand for resources is reduced substantially below the level at which the protective measures were triggered. The DUT is expected to adequately maintain essential control throughout the test
Results	Pass or fail
Remarks	The DUT vendor is not required to be able to predict the messaging rate at which such protective measures are invoked, but SHOULD be able to put an upper bound on time after the stimulus ceases before the recovery is complete

Bibliography

IANA protocol and number registries, http://www.iana.org/protocols/
registries of various assigned code points for standard Internet protocols

— — — — — —