EDSA-403
ISA Security Compliance Institute –
Embedded Device Security Assurance –
Testing the robustness of implementations of the IETF IPv4 network protocol

Version 1.6

February 2015

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
1.31	2010.09.28	generalized test IPv4.T07; 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 (merging three tests and removing five); 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
1.6	2015.02.07	incorporate EDSA-102 v1.2 errata; change terminology essential services to essential functions; require pseudo random test generation where applicable; load tests run two minutes; clarify pass/fail in Clause 7; lower level protocols must be valid in fuzz tests; correct definition and meaning of NPDU header field TotalLength; require T06 to test each defined option; require T08 to test instances of each restricted source-address class; require T11 to include defects of R10 classes c), d) and e); extended T12 to include DUT discard of overly-long reassembled NPDU; corrected 4.2.4.6.1 b) and c); elaborated T13 to ensure Requirement IPv4.R16 and 6.7.3 are considered

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 IPv4 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.

Requirements are comprised of all the numbered Requirements in this document and any immediately following clarifying information, together with the tables in Clause 7 that describe individual tests. In the event of a conflict between these, the tables in Clause 7 take precedence.

NOTE 1 The IPv4 protocol is without distinction between server and client roles. The protocol is almost stateless; the sole exception is the transient per-instance state information used during reassembly of a fragmented NPDU while awaiting reception of additional NPDUs that together convey that unfragmented NPDU.

NOTE 2 Although conformance is explicitly NOT a goal of this testing, prior versions of this document included some aspects of conformance testing which have now intentionally been removed.

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] *ISA Security Compliance Institute – Embedded device security assurance – Requirements for embedded device robustness testing* [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 and number registries, http://www.iana.org/protocols/
registries of various assigned code points for standard Internet protocols

RFC791, *Internet protocol [version 4]*

RFC950, *Internet standard subnetting procedure*

RFC1042, *A standard for the transmission of IP datagrams over IEEE 802 networks*

RFC1122, *Requirements for internet hosts – communication layers*

NOTE 3 Only 3.2.1 and 3.2.2 are referenced.

RFC1191, *Path MTU discovery*

[1] to be published concurrently with this document

NOTE 4 Only Clause 4 is referenced.

RFC1393, *Traceroute using an IP option*

RFC1475, *TP/IX: The next Internet*

RFC1770, *IPv4 option for sender directed multi-destination delivery*

RFC1812, *Requirements for IP version 4 routers*

RFC2113, *IP router alert option*

RFC2474, *Definition of the differentiated services field (DS field) in the IPv4 and IPv6 headers*

RFC2475, *An architecture for differentiated services*

RFC3168, *The addition of explicit congestion notification (ECN) to IP*

RFC3260, *New terminology and clarifications for diffserv*

RFC5350, *IANA considerations for the IPv4 and IPv6 router alert options*

These publications of the Internet Engineering Task Force (IETF) are the controlling specifications for the higher-sublayer ICMPv4 protocol that may be used in testing the robustness of the IPv4 protocol that is the subject of this document:

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

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

IANA port numbers, as specified at http://www.iana.org/assignments/port-numbers

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
“Ethernet”
either the IETF Ethernet II protocol or IEEE 802 SNAP over IEEE 802.2 Type 1 LLC over IEEE 802.3

3.1.4
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.

3.1.5
inferior (protocol)
protocol at a lower layer or sublayer than the referenced protocol

3.1.6
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.

3.1.7
malformed (message or PDU)
PDU that violates syntactic rules on PDU structure

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

3.1.8
network service data unit
data block passed between the network and higher protocol layers, which is also the payload of the associated (possibly virtual) unfragmented NPDU

3.1.9
reassembling
post-reception function performed by IP to reconstruct one unfragmented NPDU from multiple fragmented NPDUs

3.1.10
superior (protocol)
protocol at a higher layer or sublayer than the referenced protocol

3.1.11
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.

3.1.12
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

3.1.13
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 protocol version 4
IEEE	Institute of Electrical and Electronic Engineers
IETF	Internet engineering task force
IPv4	Internet (network layer) protocol version 4
MTU	maximum transmission unit
MPDU	medium access control (MAC) sublayer protocol data unit
(*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
NSDU	network-layer service data unit
OUI	organizationally unique identifier, assigned by IEEE
SNAP	sub-network access protocol
TCP	transmission control protocol
TD	testing device

4 Elements of the protocol under test

4.1 General

This document specifies robustness testing for the IETF IPv4 protocol, which is a network protocol providing an unordered, unreliable end-to-end communications path, whose only state information is that used transiently during the reassembly of a fragmented NPDU while awaiting reception of the several NPDUs that together convey that original fragmented NPDU and its conveyed NSDU.

4.2 IPv4 NPDUs

4.2.1 IPv4 NPDU structure

An IPv4 NPDU is structured as shown in Figure 1, using a big-endian octet order.

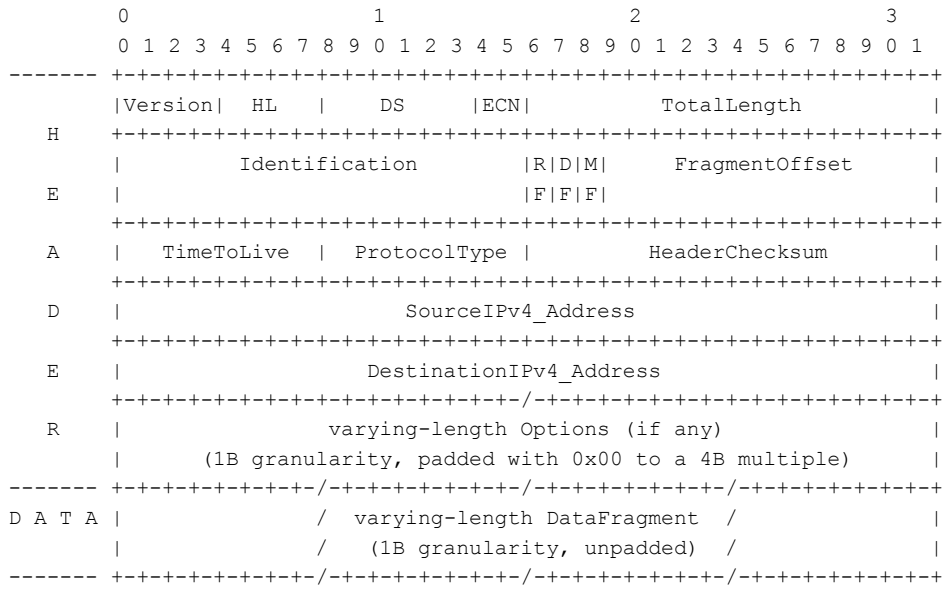


Figure 1 – IPv4 NPDU structure

4.2.2 Mandatory fields

NOTE 1 In the following, some of the field names have been expanded or, in one case, shortened to better express their meaning.

The following fields are mandatory components of each IPv4 NPDU (where field sizes are specified in octets (B) or bits (b)):

a) Version (V): (4b): 0b0100 (i.e., 0x4, which is the source of the “v4” in “IPv4” and “ICMPv4”)

b) HeaderLength (HL): (4b): Size of IP header in 4 B words, range 5..15

NOTE 2 In 1981 this field was called IHL (internet [sic] header length), the NPDU header size in 4 B words. The header length (in 4 B words) is also the offset to the start of the Data field, which contains NSDU data.

NOTE 3 In 1981 the following two fields were known collectively as the TOS (type of service) field. In 1998 that definition was made obsolete by RFC1349, which was subsequently replaced by RFC2474, RFC3168 and RFC3260.

c) DifferentiatedServices (DS): (6b): (per RFC2474, RFC3260)

d) ExplicitCongestionNotification (ECN): (2b): (per RFC3168)

— 0b00 = not-ECT (non ECN-capable transport)

— 0b01 = ECT(1)

— 0b10 = ECT(0)

— 0b11 = CE (congestion notification to end nodes)

e) TotalLength (TL): (2B): includes complete header and all data of the conveyed NPDU fragment (in units of 1 B); every value in the range 21 .. 576 SHALL be supported; support of values greater than 576 is optional

NOTE 4 TotalLength specifies the size of the octet sequence being conveyed in this specific PDU, not that of an eventual NPDU after any potential reassembly of multiple fragments.

f) Identification (ID): (2B): see 4.2.4.6

g) IP fragmentation flags (FF): (3b): see 4.2.4.6

— RF: RFU: (1b): 0=required, 1=reserved for future use; may not be used

— DF (don't fragment): (1b): 0=may-fragment, 1=don't-fragment

— MF (more fragments): (1b): 0=last-fragment, 1=more-fragments

h) FragmentOffset (FO): (13b): 0..(TL-1)/8 (in units of 8 B), see 4.2.4.6

i) TimeToLive (TTL): (1B): discard IPv4 NPDU at time of forwarding when =0, see 4.2.4.3

j) ProtocolType (PT): (1B): see 4.2.4.4

k) HeaderChecksum (HC): (2B): see 4.2.4.2

l) SourceIPv4_Address (SA): (4B): see 4.2.4.4

m) DestinationIPv4_Address (DA): (4B): see 4.2.4.4

n) Options (O): (4 × (HL-5)B): see 4.2.5

o) DataFragment: (<= (TL - 4 × HL)B, possibly null, granularity 1 B, not padded)

4.2.3 Mandatory protocol aspects

4.2.3.1 Conveying DPDU

This specification presumes that IPv4 NPDUs are conveyed by "Ethernet" DPDUs without DPDU segmentation.

NOTE Although this specification presumes that IPv4 NPDUs are being conveyed by IEEE 802.3 DPDUs, other means of conveying such IPv4 NPDUs, such as use of IEEE 802 SNAP over IEEE 802.2 Type 1 over IEEE 802.11, are not inherently precluded. Other than conveyance, IPv4 has no dependencies on these lower layer protocols.

4.2.3.2 Associated management protocol version

IPv4 requires ICMPv4.

NOTE Many embedded devices or their associated automation system devices (e.g., control firewalls) may preclude the use of ICMPv4, either directed to the DUT or as error messages sent by the DUT or both. Thus those aspects of IPv4 that require transmission or reception of ICMPv4 PDUs may not be testable in the control environment, and may even be absent from the IPv4 implementation.

4.2.4 Mandatory elements of procedure

4.2.4.1 Receipt of an DPDU with an incomplete NPDU header

The minimum size for the header of an IPv4 NPDU is 20 octets.

M1) Any received DPDU payload of less than 20 octets that is classified as an IPv4 NPDU SHALL be discarded without notification.

M2) Any received DPDU payload of 20 or more octets that is classified as an IPv4 NPDU, but which has fewer octets than 4 × HeaderLength octets, MAY cause an ICMPv4 ParameterProblem error PDU to be sent to the source IP address of the received NPDU, or MAY be discarded without notification.

4.2.4.2 Checksum

This is the 16-bit one's complement of the one's-complement sum, in big-endian octet order, of all 2 × <HL> 2B words in the IPv4 NPDU header of Figure 1, up through the Options field but not including the DataFragment field. While computing the checksum, the checksum field of the NPDU is set to zero.

M1) Received NPDUs whose computed checksum value differs from that conveyed by the checksum field of the NPDU SHALL be discarded without notification.

NOTE Since the checksum is a one's-complement computation the values +0 (0x0000) and –0 (0xFFFF) are equivalent.

4.2.4.3 TimeToLive (TTL)

The TimeToLive (TTL) field, which has a default value at origination of 64, must be decremented by at least one at each hop.

M1) NPDUs SHALL NOT be forwarded when TTL=0; instead an ICMPv4 TimeExceeded error PDU with reason code 0, meaning “time to live exceeded in transit”, SHOULD be sent to the source IP address of the received NPDU, but sending the ICMPv4 PDU MAY be suppressed as a matter of system policy.

NOTE 1 This field usually is used as a max-hop-count with no time consideration.

NOTE 2 ICMPv4 error reports SHOULD be sent, but MAY be suppressed as a matter of system policy.

M2) NPDUs SHOULD be received whether TTL is zero or not.

NOTE 3 Failure to accept received IPv4 NPDUs whose TTL field has the value zero is a common implementation error.

4.2.4.4 Protocol identifier

Protocol types of interest: ICMP= 0x01, TCP= 0x06, UDP= 0x11. In all cases these are the versions of the respective protocols that can be conveyed by IPv4:

ICMPv4;

UDP and TCP with IPv4-based checksums using IPv4 addresses.

M1) Any received NPDU specifying a protocol type that is not supported by the receiving device SHOULD cause an ICMPv4 DestinationUnreachable error PDU with an error code of 2, meaning “protocol unreachable”, to be sent to the source IP address of the received NPDU, but sending the ICMPv4 PDU MAY be suppressed as a matter of system policy.

NOTE It is likely that the above three protocol types are the only valid types for the DUT.

4.2.4.5 IPv4 addresses

4.2.4.5.1 Formats

Source & destination IPv4 addresses are structured as follows, and as shown in Figure 2, using a big-endian octet order: (4 B = 32 b):

Class E: 4 b: 0b1111, 28 b: reserved for experimental use

Class D: 4 b: 0b1110, 28 b: multicast address

Class ABC: (*N*) b: network ID, (32-*N*) b: local ID on network, 3 ≤ *N* ≤ 30, bits 00..02 ≠ 0b111

NOTE The designation Class ABC is used to represent the merging of Class A, Class B and Class C that results from the unification of IP addressing specified in RFC1812, 2.2.5, which amends RFC1122, 3.2.1.3, which amends RFC791, 3.2.

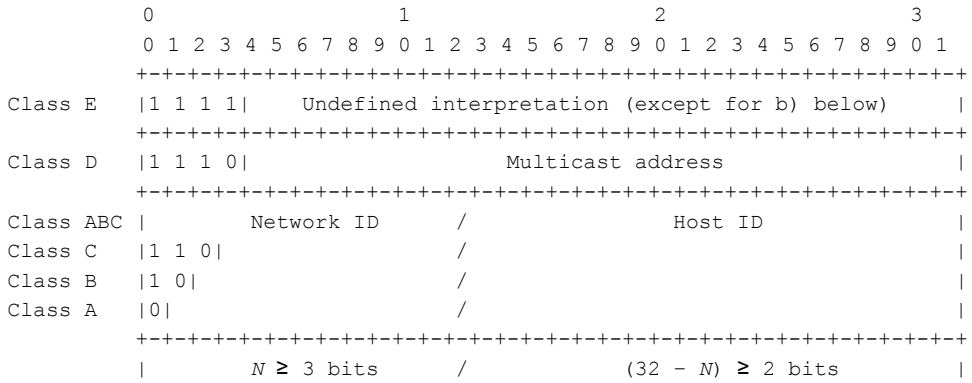


Figure 2 – IPv4 address classes

4.2.4.5.2 Restrictions

Using the following notation for an IP address:

{ <Network-ID>, <Host-ID> }

and the notation “0” for a field that contains all 0 bits and "-1" for a field that contains all 1 bits (i.e., two’, the important special cases for Class ABC unicast and limited-broadcast IP addresses are as follows:

a) Class A: { 0, 0 } – 0/32

Meaning: This host on the immediate physically connected network.

M1) This IP address SHALL NOT be sent, except as a source address as part of an initialization procedure by which the host learns its own IP address. (See ARP and DHCP)

b) Class E: { -1, -1 } – -1/32

Meaning: Broadcast limited to the immediate physically connected network.

M2) This IP address SHALL NOT be used as a source address. An NPDU with this destination address will be received by every host on the connected physical network but SHALL NOT be forwarded outside that network.

c) Class A: { 0x7F, <any> } – 128/8

Meaning: Internal host loopback address.

M3) Addresses of this form SHALL NOT appear on the network.

d) Class ABC: { 0, <Host-ID> }

Meaning: The specified host on the immediate physically connected network.

M4) This IP address SHALL NOT be sent, except as a source address as part of an initialization procedure by which the host learns its full IP address. (See ARP and DHCP)

e) Class ABC: { <Network-ID>, -1 }

Meaning: Directed broadcast to the specified network.

M5) This IP address SHALL NOT be used as a source address.

A number of additional constraints apply to IP addresses:

M6) IP addresses SHALL NOT have the value 0 or -1 for either the <Network-ID> or <Host-ID> fields, except in the special cases a) through e).

M7) When a host sends any NPDU, the IP source address SHALL be one of its own unicast IP addresses and not a broadcast or multicast address.

M8) A host SHALL discard, without notification, a received NPDU that is not destined for the host.

A received NPDU is destined for the host if the NPDU’s destination address field is:

1) (one of) the host's IP address(es); or
2) an IP broadcast address valid for the connected network; or
3) the address for a multicast group of which the host is a member on the receiving physical interface.

For most purposes, an NPDU addressed to a broadcast or multicast destination is processed as if it had been addressed to one of the host's IP addresses; the term "specific-destination address" is used for the equivalent local IP address of the host. The specific-destination address is defined to be the destination address in the IP header unless the header contains a broadcast or multicast address, in which case the specific-destination address is an IP address assigned to the physical interface on which the NPDU was received.

M9) A host SHALL discard, without notification, a received NPDU containing an IP source address that is invalid by these rules.

M10) A host SHOULD discard, without notification, a received NPDU containing an IP source address that is (one of) the host's IP address(es), unless the host intentionally sent that NPDU to itself (e.g., via a specified source route).

NOTE This simple measure provides uniform protection for higher-layer host protocols against “Land” attacks and similar attacks that spoof NPDUs which purportedly were sent by the host.

M11) The address ranges specified in Table 1 are subnet-local and SHALL NOT be forwarded by a router as either a source or destination IP-address:

Table 1 – Ranges of non-routable IP addresses

Prefix form	Range		
10/8	10.0.0.0	..	10.255.255.255
172.16/12	172.16.0.0	..	172.31.255.255
192.168/16	192.168.0.0	..	192.168.255.255

4.2.4.6 Fragmentation and reassembly

4.2.4.6.1 Supporting fields

IPv4 NPDU fields that support fragmentation:

Identification (ID): assigned by source, required to be unique to the source-destination IP address pair and specified protocol during the interval that the NSDU is in transit

TotalLength (TL) of the NPDU, thereby specifying (after header option processing) the total number of payload octets conveyed in the NPDU, where the number of payload octets should be a multiple of 8 B when the MF flag is set

FragmentOffset (FO) within the reassembled NSDU: as an integral 8 B multiple of the FO value

Don’t-fragment (DF): 0=may-fragment, 1=don’t-fragment

More-fragments (MF) flag: 0= last; 1=not-last

4.2.4.6.2 Constraints

When MF is set, the NPDU payload SHALL be an 8B multiple.

When subdividing a fragment during forwarding into a still-more-size-limited subnet, MF of the last created sub-fragment SHALL equal that of the received larger fragment that is being subdivided; the other fragments SHALL have the value MF=1.

Reassembly SHALL detect whether all needed fragments have been received, and SHOULD discard duplicate fragments. Each node SHALL be able to forward an IPv4 NPDU of <=68 B without requiring further fragmentation. Each node SHALL be able to discard a partially received fragmented IPv4 NPDU after an appropriate timeout. Each node SHALL be able to limit at any instant the number of partially received fragmented IPv4 NPDUs that are being held for reassembly.

NOTE 1 The choice to discard the earlier or later received versions of an 8 B fragment “line” is left to the implementation.

NOTE 2 The choice to restart the timeout after receipt of each fragment is left to the implementation.

4.2.4.6.3 Informal description of NPDU fragmentation process

Conceptually, each to-be-sent NSDU is assembled into a single IPv4 NPDU, together with any needed IPv4 options. A value is assigned to the NPDU's Identification field that is different than values recently assigned to other NPDUs with the same DestinationIPv4_Address, SourceIPv4_Address and ProtocolType. The NPDU's DF fragmentation flag is set or reset according to criteria not specified in the IPv4 standard.

If the resulting NPDU exceeds the expected maximum transmission unit (MTU) size for the expected path(s) from the sending device to the destination device with the specified DestinationIPv4_Address, and the NPDU's DF fragmentation flag is reset, then the NPDU is fragmented into multiple NPDUs before transmission, each of which contains

a) the same header (other than option and NPDU length and MF flag value);

b) the same IPv4 options, unless the NPDU being formed is not the first and the options are ones which are only present in a first NPDU of a series;

c) one or more 8 B lines of the unfragmented NDPU's payload, selected to start on a 0 (mod 8) B integral line boundary, where the last 8 B line can be truncated only if it is the last line of the unfragmented NDPU's payload;

d) the FragmentOffset field is set to reflect the relative line index within the original unfragmented NDPU's payload of the first line of the created NPDU's payload;

e) all created NPDUs except the one NPDU containing the last (full or partial) line of the unfragmented NDPU's payload have their MF fragmentation flag set; the last NPDU has that flag reset.

A router that receives an IPv4 NPDU, which needs to fragment the NPDU's payload to meet the MTU requirements of the forwarding path, and that is permitted to do so because the received NPDU's DF fragmentation flag is reset, does so using a variation on the above procedure:

a) the same header (other than option and NPDU length and MF flag value);

b) the same IPv4 options, unless the NPDU being formed is not the first and the options are ones which are only present in a first NPDU of a series;

c') one or more 8 B lines of the received NPDU's payload, selected to start on a 0 (mod 8) B integral line boundary, where the last 8 B line can be truncated only if it is the last line of the received NPDU's payload, that payload was identically truncated, and the received NPDU's MF flag is reset;

d') the FragmentOffset field is incremented relative to that in the received NPDU to reflect the relative line index within the received NPDU's payload of the first line of the created NPDU's payload;

e') all created NPDUs except the NPDU containing the last (full or partial) line of the received NPDU's payload have their MF fragmentation flag set; the last NPDU has that flag identical to the MF flag of the received NPDU.

Alternatively, if the sending device needs to fragment the NPDU's payload to meet the MTU requirements of the forwarding path, but is not permitted to do so because the received NPDU's DF fragmentation flag is set, then this SHOULD cause an ICMPv4 DestinationUnreachable error PDU with an error code of 4, meaning "fragmentation needed and DF set", to be sent to the source IP address of the received NPDU, but sending the ICMPv4 PDU MAY be suppressed as a matter of system policy.

4.2.4.6.4 Informal description of process to reassemble an unfragmented NPDU from fragmented NPDUs

Each instance of the ordered tuple (DestinationIPv4_Address, SourceIPv4_Address, ProtocolType, Identification) identifies a different transient NPDU reassembly FSM. Table 2 and Table 3 describe the conceptual states and transitions, respectively of that FSM.

Table 2 – IP NPDU reassembly states

State	Use
—	FSM is not instantiated
S1	reassembling
S1a	checking reassembly completion

Table 3 – IP NPDU reassembly transitions

Initial state	Conditions	Actions	Final state
—	unfragmented NPDU received	process options; deliver contained NSDU	—
—	fragmented NPDU received	allocate reassembly buffer and bit map for an NPDU with a header of size (4 × HL) B and an NSDU of size (TL - 4 × HL) B; set first ⌈TL/8 – HL/2⌉ bits in bit map for NSDU reassembly of 8 B NSDU lines; (a) write received fragment lines into appropriate 8 B line(s) of reassembly buffer; reset bit-map bits corresponding to lines written; start timeout monitor based on received NPDU's TTL value	S1
S1	fragmented NPDU received	process options; write received fragment lines into appropriate 8 B line(s) of reassembly buffer; (b) reset bit-map bits corresponding to lines written	S1a
S1a	some bit-map bits still set	optionally extend timeout based on received NPDU's TTL value (c)	S1
S1a	all bit-map bits reset	process options; deliver reassembled NSDU; discard NSDU reassembly state; terminate FSM	—
S1	unfragmented NPDU received (d)	process options; deliver reassembled NSDU; discard NSDU reassembly state; terminate FSM	—
S1	timeout	send ICMPv4 TimeExceeded error PDU with reason code 1, meaning "fragment reassembly time exceeded", to source IP address of the partially reassembled NPDU (e) discard NSDU reassembly state; terminate FSM	—

(a) ⌈…⌉ is the ceiling function, which rounds up to the next largest integer.

(b) The choice to discard the earlier or later received versions of an 8 B fragment "line" is left to the implementation.

(c) Restarting the timeout monitor on receipt of any fragment provides adaptive timeout for NSDUs with a large number of segments. Omitting this restart is also permitted, which would provide each NSDU with the same maximum reassembly time independent of the number of NPDUs required to convey the NSDU.

(d) This transition can occur only when a complete unfragmented NPDU with the same 4-component identification is received after prior receipt of one or more fragmented NPDUs containing fragments of the same NSDU.

(e) ICMPv4 error reports SHOULD be sent, but MAY be suppressed as a matter of system policy.

NOTE An alternate informal non-FSM description of the reassembly process can be found in RFC 791, 3.2.

4.2.5 Optional PDU components and elements of procedure

IPv4 defines two types of option, each structured generically as shown in Figure 3, using a big-endian octet order:

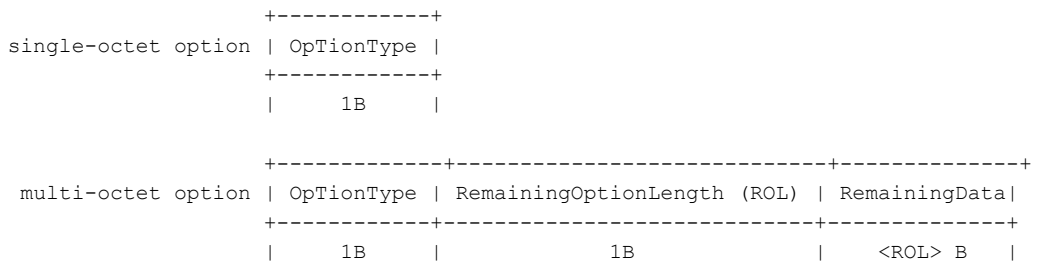
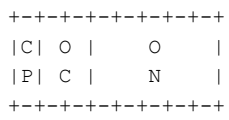


Figure 3 – Generic option structure

The 1B OptionType field is coded as shown in Figure 4:



- Copied flag (CF): (1b): 0=not-copied in first fragment only, 1=copied to each fragment
- Option class (OC): (2b): 0=control, 1=RFU, 2=debug & measurement, 3=RFU
- Option number (ON): (5b)

Figure 4 – OptionType substructure

Twelve specific options are defined for IP, as shown in Figure 5 and Figure 6.

EOS =0b0000 0000: (1 B): end-of-option-sequence, used to pad the Options field to a 4B multiple;

PAD =0b0000 0001: (1 B): intra-option padding used to force a specific starting alignment of a subsequent option;

LSRR =0b1000 0011 (<3+4*n*> B): Loose source & record route; copied to each fragment

SSRR =0b1000 1001 (<3+4*n*> B): Strict source & record route; copied to each fragment

RR =0b1000 0111 (<3+4*n*> B): Record route; first fragment only

TS =0b0100 0100 (<4+4*n*> B): Record timestamp and optionally route; first fragment only

TR =0b0101 0010 (12 B): Traceroute; first fragment only (experimental, RFC1393)

RA =0b1001 0100 0000 0100 (4 B): Router alert; copied to each fragment (RFC5350)

MD =0b1001 0101 (<2+4*n*> B): Multi-destination; copied to each fragment (RFC1770)

NOTE RFC1770 specifies concurrent use of a Broadcast destination IPv4 address for the NPDU.

DoDSEC =0b1000 0010 (<3+*n*> B): DoD basic security; copied to each fragment (RFC1108)

DoDEXT =0b1000 0101 (<3+*n*> B): DoD extended security; copied to each fragment (RFC1108)

DoDSN =0b1000 1000 (4 B): DoD satellite network stream ID; copied to each fragment

The previous three DoD options SHOULD NOT be used in non-DoD embedded devices. However, they are valid IPv4 options that conform to standard TLV (type, length, value) format; thus their presence SHOULD NOT be viewed as a robustness issue.

```
       +--------+
EOS    |00000000|
       +--------+
       |   1B   |
```

NOTE 1 This option should be used only at the end of the sequence of options, to pad to a 4B boundary.

```
       +--------+
PAD    |00000001|
       +--------+
       |   1B   |
```

NOTE 2 This option should be used only before other options, to pad to a desired boundary modulo 2B or 4B.

```
       +--------+--------+-----------------+
RA     |10010100|00010100|    alert code   |
       +--------+--------+-----------------+
       |   1B   |   1B   |        2B       |
```

```
       +--------+--------+-----------+--------------------+-------------------+-------------------------+
TR     |01010010|00001100| ID number | outbound hop count | return hop count  | originator IPv4 address |
       +--------+--------+-----------+--------------------+-------------------+-------------------------+
       |   1B   |   1B   |    2B     |---------2B---------+--------2B---------+------------4B------------+
```

- Length: option size, set at NPDU formation, unchanged as NPDU is forwarded
- ID number = arbitrary trace ID, unrelated to NPDU Identification field

```
LSRR   +--------+--------+--------+---------//--------+
SSRR   |1000X0Y1| length | offset |     routeData     |
  RR   +--------+--------+--------+---------//--------+
       |   1B   |   1B   |   1B   |        4n B       |
```

- Length: option size, set at NPDU formation, unchanged as NPDU is forwarded
- Offset: 1-origin index of next 4B IP address in routeData, originated as 4, incremented by 4 as NPDU is forwarded
- RouteData = one or more 4B IP addresses written by intermediate forwarding routers

NOTE 3 Attempting to use a partial final address field is an error that results in packet discard. The pre-coat value for the routeData field is not specified; all zeros for these IP addresses would be appropriate.

```
       +--------+--------+-----------------------+
MD     |10010100| length |  multipleDestinations |
       +--------+--------+-----------------------+
       |   1B   |   1B   |         4n B          |
```

- Type always specifies that the option is required in each segment of a segmented NPDU
- Length: option size, set at NPDU formation, unchanged as NPDU is forwarded
- MultipleDestinations = one or more 4B destination IPv4 addresses

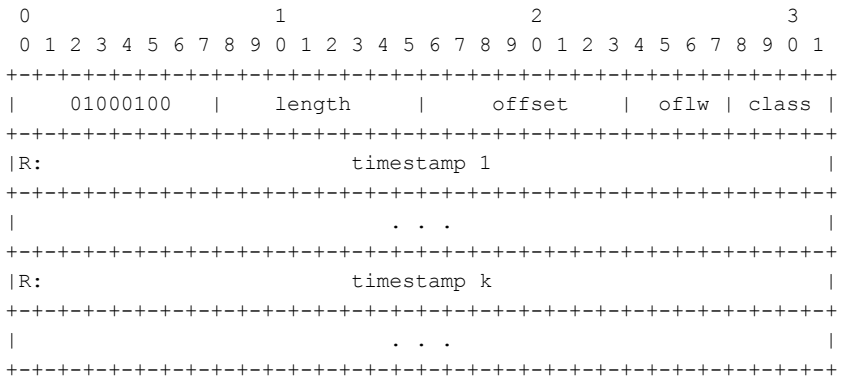
NOTE 4 Attempting to use a partial final address field is an error that results in packet discard

```
DODSEC +--------+--------+---------//--------+
DODEXT |  type  | length |       value       |
DODSN  +--------+--------+---------//--------+
       |   1B   |   1B   |        n B        |
```

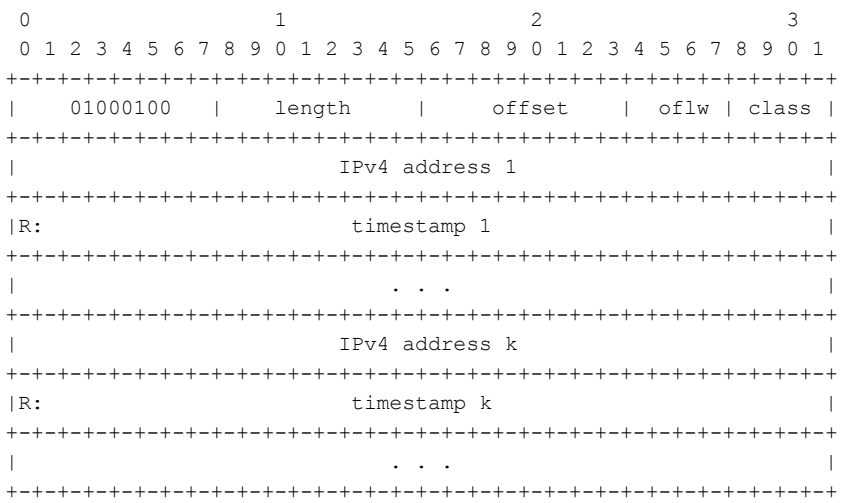
- Type always specifies that the option is required in each segment of a segmented NPDU
- Length: option size, set at NPDU formation, unchanged as NPDU is forwarded

Figure 5 – Specific structure of options other than timestamp

Timestamp (TS), Class 0:



Timestamp (TS), Class 1 or Class 2:



a) Length: 4n B, 2 <= n <= 10, other length values are invalid

b) Offset: 1-origin index of next 4 B or 8 B timestamp record, originated as 5, incremented by 4 or by 8 as NPDU is forwarded

c) Overflow (oflw): count of number of forwarding routers that could not record their timestamps due to lack of sufficient entries.

NOTE 5 Oflw should be originated as zero and should stick at 0xF, but those requirements are not in RFC791.

d) Class: format and handling of timestamp or IP-address/timestamp records within the option:

0: IP addresses are not present or recorded

1: IP addresses are recorded with timestamps

2: Pre-specified IP addresses select which routers insert timestamps (but only when router is at next specified IP address)

– other classes are undefined

e) IPv4 address: 4 B when present

f) Timestamp: 4 B: High-order bit (R) specifies reference for time

0: R =0: ms since midnight UTC

1: R =1: router-local ms time not referenced to UTC

NOTE 6 The pre-coat values for IP address and timestamp fields are not specified; all zeros for IP addresses and all ones for timestamps would be appropriate.

Figure 6 – Specific structure of timestamp options

Most of the required procedure within routers for handling IPv4 options is obvious and is covered in RFC791. However, RFC1393 added an experimental element to the forwarding of IPv4 NPDUs containing any of the Route Record options: RR, LSRR and TSRR. Compliance with RFC1393 is optional and unlikely to be encountered. Nevertheless, TDs SHALL NOT fail a DUT simply because it generates Traceroute ICMPv4 PDUs as specified in RFC1393.

5 Elements of other protocols required for the testing

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

This specification presumes that IPv4 NPDUs are being conveyed by “Ethernet” DPDUs. However, other protocols can replace “Ethernet”, such as IEEE 802 SNAP over IEEE 802.2 Type 1 over IEEE 802.11. In reality, IPv4 has no dependence (other than conveyance) on any of these lower-layer protocols.

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

5.2.1 General

The ICMPv4 protocol, a mandatory co-protocol that must co-exist with any IPv4 implementation, may 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.

Basic testing of IPv4 robustness requires only transmission of IPv4 test NPDUs to the DUT. Thus there is no requirement for use of ICMPv4 or a superior layer protocol during IPv4 robustness testing. However, TDs are permitted to observe ICMPv4 NPDUs generated by the DUT in response to NPDUs from the TD and, if desired, to use ICMPv4 Echo NPDUs to probe the DUT’s reassembly of fragmented IPv4 NPDUs. Thus those aspects of ICMPv4 that could be relevant to IPv4 robustness testing are summarized in 5.2.2.

5.2.2 ICMPv4 PDU composition

5.2.2.1 Generic structure of ICMPv4 PDUs

ICMPv4 PDUs are structured generically as shown in Figure 7, using a big-endian octet order; they are carried as the payload of IPv4 NPDUs.

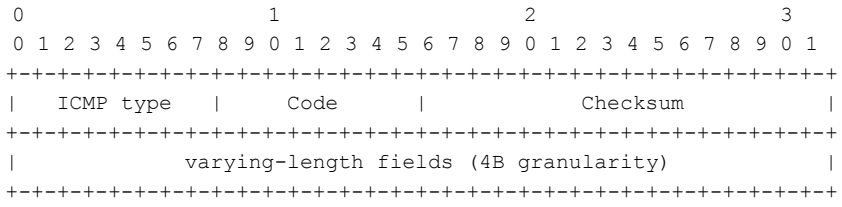


Figure 7 – ICMPv4 PDU structure

Sixteen 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
2) Route control command:
   d) Source quench, ICMP type = 0x04, specified in RFC792
   e) Redirect, ICMP type = 0x05, specified in RFC792
3) Testing query/reply:
   f) Echo, ICMP type = 0x08, specified in RFC792
   g) Echo reply, ICMP type = 0x00, specified in RFC792
   h) Timestamp, ICMP type = 0x0D, specified in RFC792
   i) Timestamp reply, ICMP type = 0x0E, specified in RFC792
4) Environment query/reply:

j) Information request, ICMP type = 0x0F, specified in RFC792, obsolesced by RFC1812

k) Information reply, ICMP type = 0x10, specified in RFC792, obsolesced by RFC1812

l) Address mask request, ICMP type = 0x11, specified in RFC950

m) Address mask reply, ICMP type = 0x12, specified in RFC950

n) Router solicitation, ICMP type = 0x0A, specified in RFC1256

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

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

Of these sixteen, the most useful for IPv4 testing are a) through g), which provide error reports, route control commands, and testing functionality similar to that offered by TCP Echo over TCP. Thus only a) through g) ICMPv4 PDUs are described in this subclause. A correct host IPv4 implementation SHOULD generate ICMPv4 PDU types a), b) and c) under appropriate error conditions and SHOULD be responsive to receipt of ICMPv4 PDU types d) and e).

A correct host IPv4 implementation MAY generate ICMPv4 PDUs of type p) upon receipt of an IPv4 NPDU with a trace route option. There is no requirement for robustness testing to test or interpret such ICMPv4 PDUs; however their presence is not a reason for failing a DUT.

5.2.3 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 8, followed by one or more extension objects that are structured generically as shown in Figure 9.

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.

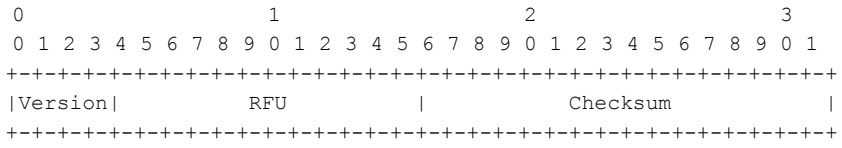


Figure 8 – ICMPv4 extension header

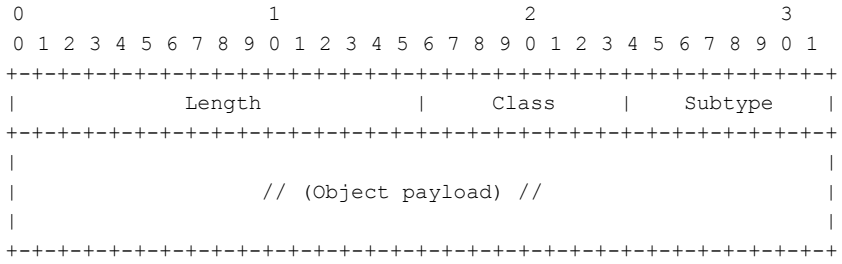


Figure 9 – ICMPv4 extension object header and payload

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.

5.2.3.1 Structure of DestinationUnreachable ICMPv4 PDUs

This PDU is structured as shown in Figure 10; it is used to report an unreachable destination, and is sent on the reverse path toward the original source of the conveyed packet.

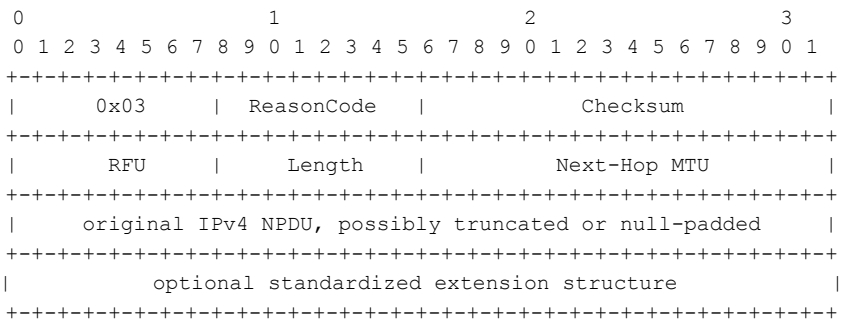


Figure 10 – DestinationUnreachable ICMPv4 PDU structure

Defined reason codes for DestinationUnreachable ICMP PDUs are specified in Table 4.

Table 4 – 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

5.2.3.2 Structure of TimeExceeded ICMPv4 PDUs

This PDU is structured as shown in Figure 11; 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.

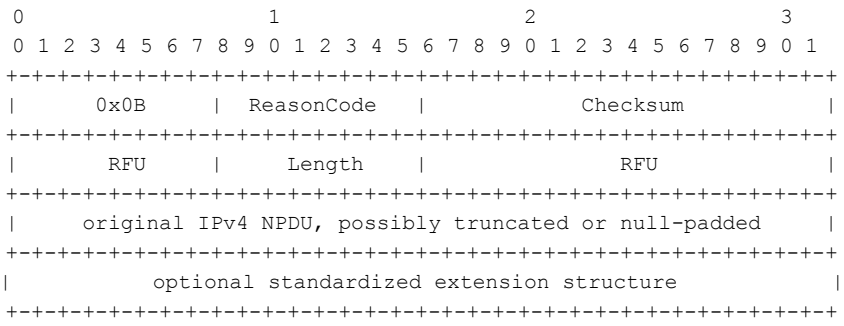


Figure 11 – TimeExceeded ICMPv4 PDU structure

Defined reason codes for TimeExceeded ICMP PDUs are specified in Table 5.

Table 5 – TimeExceeded reason codes

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

5.2.3.3 Structure of ParameterProblem ICMPv4 PDUs

This PDU is structured as shown in Figure 12; 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.

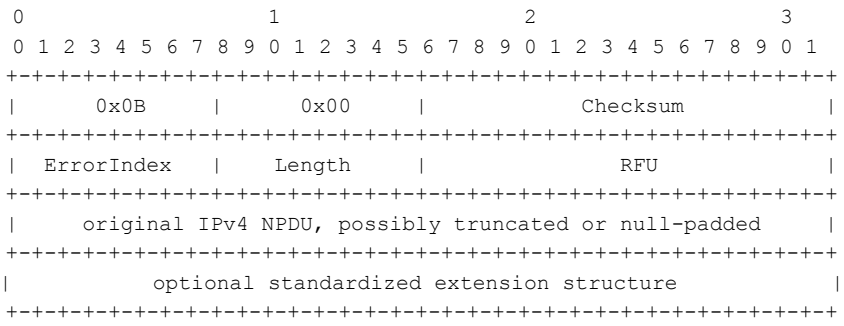


Figure 12 – ParameterProblem ICMPv4 PDU structure

5.2.3.4 Structure of SourceQuench ICMPv4 PDUs

This PDU is structured as shown in Figure 13; it is used to command that the source of a packet stop sourcing or forwarding those packets.

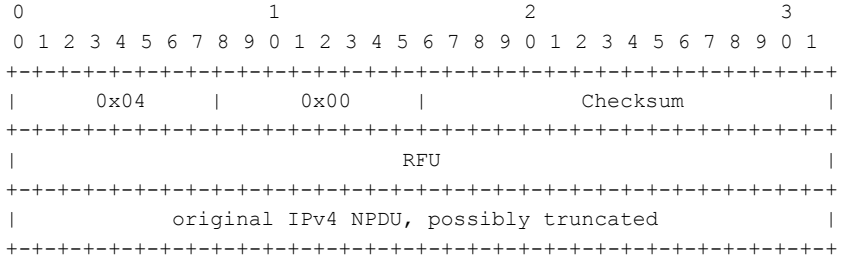


Figure 13 – SourceQuench ICMPv4 PDU structure

5.2.3.5 Structure of Redirect ICMPv4 PDUs

This PDU is structured as shown in Figure 14; it is used to command that the prior source or forwarder of a packet stop sourcing or forwarding those packets via a different route.

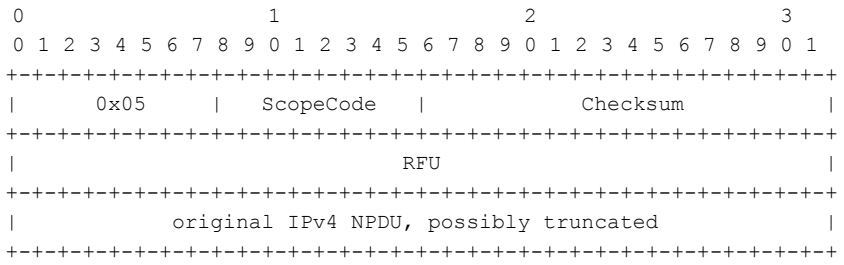


Figure 14 – Redirect ICMPv4 PDU structure

Defined scope codes for Redirect ICMP PDUs are specified in Table 6.

Table 6 – Redirect ScopeCodes

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

5.2.3.6 Structure of Echo and EchoReply ICMPv4 PDUs

These PDUs are structured as shown in Figure 15; they are used to test an IPv4 implementation by requiring that implementation to receive an ICMPv4 EchoRequest PDU with arbitrary payload and echo that PDU back to the sender as an EchoReply PDU.

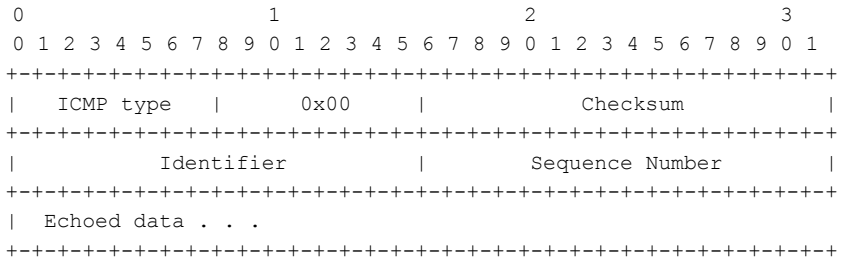


Figure 15 – Echo and EchoReply ICMPv4 PDU structure

5.2.3.7 Structure of Traceroute ICMPv4 PDUs

These PDUs are structured as shown in Figure 16. 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.

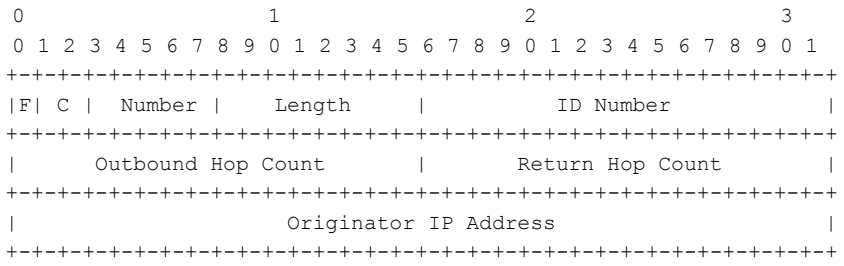


Figure 16 – Traceroute ICMPv4 PDU structure

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

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 must be preconditioned to support testing by

1) meeting the requirements of [EDSA-310] for demonstrating continued correct operation during testing;

2) if possible, for TDs that monitor DUT transmissions during robustness testing, preparing the DUT and other devices in its test environment to not block or discard generated ICMPv4 error PDUs, so that such error PDUs can be used for observing anomalies in the IPv4 implementation.

   NOTE 1 Such observation is not a requirement, since most faults in IPv4 implementations can be inferred even when ICMPv4 error report PDUs are blocked by a firewall.

3) if possible, for TDs that monitor DUT transmissions during robustness testing, preparing the DUT and other devices in its test environment to not block or discard ICMPv4 Echo PDUs, so that ICMP can be used for testing the IPv4 implementation, or

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 network communications load, before any protocol fuzzing or stress testing is attempted.

   NOTE 2 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 3 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 4 This conceptual phase focuses on complex protocol robustness/fuzzing tests as well as 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 IPv4.R1 – Criteria for robustness test failure

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

whether or not essential functions are adequately maintained under network traffic conditions created under these tests, as defined in [CRT.Essential_functions];

any particular conditions resulting in pass/fail mandated by the testing specified in this document.

The IPv4 protocol that is the subject of this specification is a stateless protocol without any query/ response mechanisms. Thus there is no requirement for a superior layer protocol during IPv4 robustness testing.

6.3 Protocol stack used for testing

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

Although this specification presumes that “Ethernet” is conveying IPv4 NPDUs, other means of conveying IPv4 NPDUs, such as the IEEE 802.11 (WiFi)data-link layer, are not inherently precluded.

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

IPv4 is a near-stateless network protocol that routes, and potentially segments and reassembles, higher-layer PDUs, providing transport protocol identification and end-device addressing, as well as detection of most non-deliberate changes to the conveying NPDU.

Testing of IPv4 robustness requires only transmission of IPV4 test NPDUs to the DUT. Thus there is no direct requirement for ICMPv4 or a superior layer protocol during IPv4 robustness testing. However, observations of the DUT’s response to received NPDUs is permitted, including generation of ICMPv4 error report NPDUs to notify the TD of problems detected during reception of the TD’s transmissions. Thus the expected ICMPv4 error responses are documented.

6.4 Phase 0: DUT preconditioning

Requirement IPv4.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) for TDs that monitor DUT transmissions during robustness testing, enabling the forwarding of ICMPv4 PDUs, from the DUT to the TD, 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 functions* of the embedded device under the test conditions, per the requirements in [CRT.Essential_functions];

Essential functions as defined in [CRT.Essential_functions] include control loops, 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, other automation system elements that have been engineered to communicate with and monitor the DUT.

6.5 Phase 1: Baseline operation

6.5.1 General

Requirement IPv4.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 IP component of the DUT's protocol stack is present and functioning, and that the DUT can maintain essential functions.

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, extra functionality for standard fields in standard messages, or extra option types or values in variable-content option fields.

NOTE Such protocol extensions are not common in IPv4 implementations, so these requirements related to discovery and robustness testing of proprietary protocol extensions usually are trivially satisfied.

Requirement IPv4.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.

Also included in this testing are out-of-order receipt of protocol messages, and receipt of related protocol messages (e.g, multiple IP fragments of the same aggregate payload) with inconsistent options or overlapping segments and segment gaps that must be detected and handled during fragment reassembly.

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;

      NOTE IPv4 traffic is stateless, except for the transient state used during reassembly of received fragmented NPDUs.

   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 functions under these network traffic conditions. These criteria depend upon the specific function as well as whether the function operates on the same network interface used for test traffic.

6.6.2 Basis for IPv4 robustness testing

6.6.2.1 Testing for inappropriate or malformed content

Incorrectly formed IPv4 NPDUs form the basis for robustness testing.

Requirement IPv4.R5 – Testing of each message field for sensitivity to invalid content

For basic robustness testing requiring erroneous messages or message sequences, valid IPv4 NPDUs or NPDU sequences from the TD to the DUT SHALL be altered so that one component of one or more of the IPv4 NPDUs in the sequence is erroneous, including cases where the IPv4 NPDU is in violation of 4.2.4 or 4.2.5.

Such alterations SHALL be applied to each field of the IPv4 NPDU where alteration might have an impact on the DUT (e.g., probably not the TTL field in an IPv4 NPDU, since all received values are valid).

During basic robustness testing, lower level PDUs employed to convey a protocol under test SHALL be valid.

NOTE 1 This type of testing can be described as single-message protocol “fuzzing”.

Requirement IPv4.R6 – Testing of DUT response to truncated NPDU headers or header options

Basic robustness testing SHALL include testing whether the DUT appears to accept and process NPDUs whose received length is too short for the specified header length, or which contain multi-octet options within the header’s option field that extend beyond the end of that field, either extending beyond the NPDU or extending into the DataFragment portion of the NPDU.

Requirement IPv4.R7 – Testing of DUT response to receipt of an NPDU with an invalid checksum

Basic robustness testing SHALL include testing whether the DUT appears to malfunction when receiving IPv4 NPDUs whose received checksum value is incorrect.

Requirement IPv4.R8 – Testing of DUT response to receipt of an NPDU with an invalid source IP address

Basic robustness testing SHALL include testing whether the DUT appears to malfunction when receiving IPv4 NPDUs whose source addresses violate the restrictions of 4.2.4.5.2.

Requirement IPv4.R9 – Testing of DUT response to malformed header options for apparently supported option types

Basic robustness testing SHALL include testing whether the DUT appears to malfunction when receiving IPv4 NPDUs where the NPDU header contains an option of an apparently supported option type, but the coding of that option instance is in violation of 4.2.5, such as a misaligned Offset field value or one inconsistent with the overall length of the option as coded in the NPDU.

6.6.2.2 Testing for inconsistent content among multiple NPDUs conveying the same NSDU

Requirement IPv4.R10 – Testing of DUT reassembly of fragmented NPDUs

Basic robustness testing SHALL include testing whether the DUT appears to malfunction when receiving and presumably reassembling fragmented IPv4 NPDUs before subsequent processing. Such testing SHALL determine the DUT's ability to continue functioning upon receipt of different fragmented NPDUs consisting of

a) multiple versions of the same NSDU line (whose maximum length is eight octets);

b) <deleted>;

c) fragmented NPDUs whose MF fragmentation flag has the value 1 (more fragments), but whose payload is not an 8-octet multiple;

d) fragmented NPDUs whose payload corresponds to the last lines of the original unfragmented NPDU, but whose fragmentation flag has the value 1 (more fragments);

e) NPDUs where the protocol type field has different values in at least two of the NPDUs, where the source and destination IPv4 addresses and the identification field have identical values in all of the NPDUs, and where at least all but one of the NPDUs are fragmented.

NOTE This last class of tests explores a situation where the DUT should refer the NPDUs with different protocol types to different NPDU reassembly FSMs. It is the purpose of this test to determine that the DUT correctly makes that distinction among received NPDUs, at least some of which are fragmented NPDUs, including the case where an unfragmented NPDU is received that has the same source and destination IPv4 addresses and same identification as a fragmented NPDU currently awaiting fragments for reassembly and delivery within the DUT.

6.6.2.3 Constituent elements in basic robustness tests

It is suggested that basic robustness testing proceed in phases, from simple to complex, as enumerated in 6.6.1 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 IPv4.R11 – Constituent elements in basic robustness tests

Basic IPv4 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 IPv4.R12 – 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 (i.e., "Ethernet" or IP, as appropriate), 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 IEEE 802 protocols are included in this list as an identifiable placeholder for any physical and data-link protocols used to convey IPv4 NPDUs.

Requirement IPv4.R13 – 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 IPv4.R14 – Testing of saturation rate-limiting mechanism(s)

Saturation rate testing SHOULD be for two minutes, long enough for any saturation effects to manifest. Tests that inherently involve a large number of NPDUs, such as protocol type sensitivity 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 IPv4.R15 – Reproducibility of robustness stress testing

Basic robustness testing SHALL use a deterministic selection process (which SHALL be a seeded pseudo-random process where applicable), 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 (which SHALL be a seeded pseudo-random process where applicable), that tests series of valid messages. See Clause 7 for specific required test cases.

NOTE 2 The above constraint to use 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.

Requirement IPv4.R16 – Concurrent activation of multiple IPv4 FSMs for reassembling fragmented NPDUs

Load stress testing of the DUT's IPv4 implementation SHOULD include NPDU sequences that activate or cause error sequencing in any one of multiple concurrently-operating NPDU reassembly FSMs, including attempts to overload the state management capabilities of the DUT's IPv4 implementation.

6.7.3 Specific robustness testing

Robustness testing requiring malformed messages or message sequences should include:

a) NPDU sequences from the TD to the DUT altered so that one or more components of the NPDU are erroneous or so that the NPDU is in violation of 4.2.4 or 4.2.5, including cases where the length of the NPDU or one of its contained options is inconsistent with other requirements for that NPDU or option, or where the value of an option field is inconsistent with the values in other fields of the NPDU;

b) multiple interleaved NPDU sequences from differing source-IP-addresses and/or different source ports;

c) sequences of fragmented NPDUs with out-of-order and/or missing and/or duplicated NSDU lines in their payloads;

d) NPDU sequences where an unfragmented NPDU is sent after sending **most but not all of** a fragmented NPDU with the same source and destination IPv4 addresses, same protocol type and same value of the NPDU's identification field.

Requirement IPv4.R17 – Specific focus of robustness testing

Specific focus SHOULD be put on reassembly of fragmented NPDUs. Particular focus SHOULD be given to the DUT's behavior on receipt of inconsistent NPDUs with the same 4-component identification, which nominally designate the same NPDU reassembly FSM within the DUT. Testing also SHOULD include cases where one of these four components varies while the others remain identical, as well as exploring the DUT's behavior when the NPDU options or field values or delivered NSDU lines are inconsistent or of incorrect length.

Load stress testing of the IPv4 protocol itself, with or without concurrent testing of other protocols, MAY be used to explore rate sensitivity and other aspects of the DUT's IPv4 protocol implementation.

6.8 Reproducibility

Requirement IPv4.R18 – 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 IPv4.R19 – Specific test cases

The tested suite of protocols SHALL be documented in at least the detail specified by Table 7.

Table 7 – IPv4: Protocols used in test process

Protocol layer tested	Permissible alternatives	Protocols tested	Maximum network communications load at which deliberate limiting occurs
Physical layer	IEEE 802.3		
Data-link layer	"Ethernet"		
Network layer	IPv4, optionally with receipt of ICMPv4 error NPDUs		

Requirement IPv4.R20 – Testing SHALL include at least that specified by Table 8 through Table 22

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. Tests where ***Results*** are indicated as Pass or Fail, SHALL pass if the indicated ***Expected response*** is observed. If the Results row in a table does not indicate "Pass/Fail," this means that the test provides security-relevant information about the DUT to be included in the test report, but cannot cause a device to fail certification as long as related documentation of compensating controls is provided by the vendor as indicated.

Table 8 – IPv4.T00: Baseline operation

Test ID	IPv4.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 is 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 IPv4.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 IPv4, 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 IPv4 component of the protocol stack is present and functioning and that the DUT can adequately maintain essential functions
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 9 – IPv4.T01: Bad checksum flood to exhaust stateful firewalls

Test ID	IPv4.T01
Test name	Bad checksum flood to exhaust stateful firewalls
Test description	A flurry of otherwise apparently valid IPv4 NPDUs whose checksums are made incorrect is sent to the DUT to attempt to overwhelm the storage resources of any interposed stateful firewalls that do not validate the checksums of forwarded IPv4 NPDUs. See [CRT.Rate_limiting] for additional requirements
Reference requirements	Requirement IPv4.R13
Test type	Baseline operation: behavior of interposed firewalls when under attack
Test status	Mandatory
Expected DUT behavior	Any stateful firewall within the DUT protects itself against being triggered into effecting a denial of service by a flood of NPDUs that the DUT's IP stack subsequently discards
Test object	To evaluate the DUT's in-situ capability to withstand an attack on any interposed stateful firewall
Test configuration	A TD is connected to the DUT by an underlying switched network that uses IPv4, as specified in [CRT.Test_configuration_1]
Test procedure	The TD sends a combination of valid NPDUs and NPDUs with incorrect checksums, addressed to the DUT, in an attempt to saturate any DUT firewall so that some valid NPDUs are discarded by that firewall.. Thus the valid NPDUs that the TD sends SHOULD result in observable behavior of the DUT, such as forwarding of the NPDU or generation of an NPDU conveying a response
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Determination of the need for compensating controls when the DUT appears to not defend against a flood of NPDUs, many of which have incorrect checksums. In this case the device vendor SHALL have documented the need for such controls. The results of this test SHALL be included in the test report
Remarks	This is a test for unmitigated vulnerabilities in the TD to DUT communication path

Table 10 – IPv4.T02: Truncated NPDU: truncated fixed header

Test ID	IPv4.T02
Test name	Truncated NPDU: truncated fixed header
Test description	A truncated IPv4 NPDU is sent as a DPDU payload, where the payload is the initial octets of a correctly formed IPv4 NPDU but where the total size of the payload is between 0 and 19 octets, inclusive, and thus less than the minimum IPv4 header size of 20 octets
Reference requirements	Requirement IPv4.R6
Test type	Basic robustness: PDU structural violations
Test status	Mandatory
Expected DUT behavior	The DUT checks the NPDU's length as received – the reported size of the DPDU's payload – before processing header fields to ensure that all of the fixed header fields are present. NOTE The minimum required DPDU payload length is 20 octets
Test object	To probe the robustness of the DUT's parsing of IPv4 NPDUs and protection against malformed NPDUs
Test configuration	A TD is connected to the DUT by an underlying switched network that uses IPv4, as specified in [CRT.Test_configuration_1]
Test procedure	The TD sends the initial octets of an otherwise-valid IPv4 NPDU, where the conveying DPDU's payload is truncated to less than 20 octets. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 11 – IPv4.T03: Invalid NPDU header IP version

Test ID	IPv4.T03
Test name	Invalid NPDU header IP version
Test description	Otherwise valid IPv4 NPDUs are sent whose header version field has a value not supported by the DUT (e.g., not equal to 4, and if the DUT supports IPv6 not equal to 6)
Reference requirements	Requirement IPv4.R5
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT validates the IP version specified in the NPDU header
Test object	To probe the robustness of the DUT's rejection of IP NPDUs of unsupported versions
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an otherwise valid IPv4 NPDU whose header version field has a value other than 4. If the DUT also supports IPv6, then the value 6 is also excluded from the testing. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 12 – IPv4.T04: Invalid IPv4 NPDU header checksum

Test ID	IPv4.T04
Test name	Invalid NPDU header checksum
Test description	IPv4 NPDUs are sent whose header checksum field value differs from the computed checksum for the NPDU's header
Reference requirements	Requirement IPv4.R5, violating 4.2.4.5.2, M4 and M5
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT validates IPv4 NPDU header checksums on receipt
Test object	To probe the robustness of the DUT's checksum processing for IPv4 NPDUs
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an otherwise valid IPv4 NPDU whose header checksum field differs from a correctly computed header checksum for the NPDU. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 13 – IPv4.T05: Truncated NPDU: truncated header options

Test ID	IPv4.T05
Test name	Truncated NPDU: truncated header options
Test description	A truncated IPv4 NPDU is sent as a DPDU payload, where the payload is the initial 20 or more octets of a correctly formed IPv4 NPDU, but where truncation occurs within a non-null header option field whose existence is indicated by an NPDU HeaderLength field whose value is greater than 5 (i.e., the stated length of an NPDU option positions its last octet outside the conveying NPDU header, as determined by the NPDU header's specified length)
Reference requirements	Requirement IPv4.R5, violating 4.2.4.5.2, M2
Test type	Basic robustness: PDU structural or content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT checks the size of the received DPDU payload, then the NPDU's header length and header checksum, before processing any conveyed NPDU options
Test object	To probe the robustness of the DUT's parsing of IPv4 NPDUs and protection against malformed NPDUs
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends the initial octets of an otherwise-valid IPv4 NPDU, where the conveying NPDU's payload is truncated to a number of octets of 20 or greater but less than the value of the NPDU's HeaderLength field times 4. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	The DUT MAY immediately queue a reply NPDU for transmission, containing an ICMPv4 ParameterProblem (type 0x04) PDU

Table 14 – IPv4.T06: NPDU options

Test ID	IPv4.T06
Test name	NPDU options
Test description	The TD includes non-null IPv4 option fields in some of its IPv4 NPDUs, such that the option field is either correct or erroneous. In the latter case, either the entire option filed is malformed, or one or more of the contained option substructures — are malformed (e.g., a truncated subfield), and/or — reference an undefined option, and are thus semantically invalid, and/or — are options that are appropriate only for the initial fragment of fragmented NDPUs but are conveyed in an NPDU whose FragmentOffset field is non-zero (implying that it is not the initial fragment of an NPDU). Such testing shall include all of the option types specified in 4.2.5 except the three DOD… options, whose inclusion within the tests is optional
Reference requirements	Requirement IPv4.R5,violating 4.2.5
Test type	Basic robustness: PDU content syntactic or semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT validates the IPv4 option field when present
Test object	To probe the robustness of the DUT's option processing for IPv4 NPDUs
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends an otherwise valid IPv4 NPDU whose option field is non-null and which may, or may not, be erroneous. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	The DUT MAY immediately queue a reply NPDU for transmission, containing an ICMP ParameterProblem (type 0x04) PDU, when the option field is malformed or contains an undefined option or one that is inappropriate for a non-initial fragment NDPU

Table 15 – IPv4.T07: Receipt of NPDUs with various TTL field values

Test ID	IPv4.T07
Test name	Receipt of NPDUs with various TTL field values
Test description	The TD sends some NPDUs to the DUT whose TTL field has values in the range 0 .. 63 inclusive, including some with the value zero (0), and some with the value one (1)
Reference requirements	4.2.4.3
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT accepts received NPDUs for all TTL field values
Test object	To probe the robustness of the DUT's processing of the TTL field in received IPv4 NPDUs
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends a valid IPv4 NPDU to the DUT whose TTL field has the value zero or one. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	The test for a zero TTL field applies to NPDU transmission, not reception (RFC1122, 3.2.1.7)

Table 16 – IPv4.T08: Rejection of NPDUs with invalid source IP addresses

Test ID	IPv4.T08
Test name	Rejection of NPDUs with invalid source IP addresses
Test description	ICMPv4 PDUs are sent with source IP addresses that are invalid per 4.2.4.5.2, M9 or M10. Testing shall include addresses of each class specified in 4.2.4.5.2
Reference requirements	4.2.4.5.2, M9 or M10, Requirement IPv4.R8
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT ignores and discard, without notification, received NPDUs that specify an invalid t source IP address, or one that is an IP address of the DUT when the DUT did not just send that NPDU
Test object	To probe the robustness of 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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends IPv4 NPDUs whose source IP address is invalid per 4.2.4.5.2, M9 or M10, then listens for any response NPDU from the DUT, addressed to any destination IP address. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 17 – IPv4.T09: Processing of NPDUs that reference undefined or supposedly non-implemented protocol types

Test ID	IPv4.T09
Test name	Processing of NPDUs that reference undefined or supposedly non-implemented protocol types
Test description	ICMPv4 PDUs are sent with protocol type field values that are undefined or supposedly not implemented in the DUT
Reference requirements	
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT responds to receipt of an NPDU specifying an invalid or unimplemented protocol type with an NPDU conveying an ICMPv4 Destination unreachable PDU whose DestinationUnreachable reason code has the value 0x02, ProtocolUnreachable. However, the DUT MAY ignore and discard, without notification, such received NPDUs
Test object	To probe the robustness of the DUT's protective measures in situations where the protocol type value in a received NPDU is one that the DUT is expected to ignore
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends IPv4 NPDUs whose protocol type field specifies a value other than those assigned to protocols that the DUT is claimed to implement, then listens for any response NPDU from the DUT addressed to the TD. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	The DUT MAY respond with an NPDU conveying an ICMPv4 Destination unreachable PDU whose DestinationUnreachable reason code has the value 0x02, ProtocolUnreachable

Table 18 – IPv4.T10: Illogical or inconsistent NPDU flag values

Test ID	IPv4.T10
Test name	Illogical or inconsistent NPDU flag values
Test description	The TD establishes IPv4 associations with the DUT, then intersperses with normal IPv4 traffic IPv4 NPDUs on the association whose IPv4 flags have various inconsistent or illogical states
Reference requirements	Requirement IPv4.R10
Test type	Basic robustness: PDU content semantic violations
Test status	Mandatory
Expected DUT behavior	The DUT recovers from receipt of NPDUs that specify such nonsensical intermixed IPv4 flag values, either continuing or resetting or terminating the association
Test object	To probe the robustness of the DUT's processing of IPv4 NPDUs that are inconsistent with current IPv4 state
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 SHOULD be enabled at any intervening firewall(s)
Test procedure	The TD sends otherwise valid IPv4 NPDUs whose flags are occasionally nonsensical or inconsistent with the state of the IPv4 association. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	It does not matter whether the DUT continues, resets or terminates the associations whose IPv4 flags assume inconsistent state; it only matters that the DUT does not itself lock up or irreversibly allocate IPv4 association state resources to associations that are terminated in error

Table 19 – IPv4.T11: NPDU fragment mis-reassembly

Test ID	IPv4.T11
Test name	NPDU fragment mis-reassembly
Test description	The TD sends a fragmented NPDU whose aggregate payload size is correct but which fails to specify all of the required 8 B lines of the conveyed NSDU. Conveyance defects of the classes specified in Requirement IPv4.R10 c) through e), or of those discussed in Requirement IPv4.R17, or both, SHALL be included in the testing.
Reference requirements	Requirement IPv4.R10
Test type	Basic robustness: NSDU reassembly from fragmented NPDUs
Test status	Mandatory
Expected DUT behavior	The DUT detects that the fragmented NSDU fails to complete reassembly
Test object	To probe the robustness of the DUT's ability to detect incompletely reassembling fragmented NPDUs whose total number of received payload octets is correct
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]
Test procedure	The TD creates a large NSDU that the DUT should echo on receipt, then fragments it incorrectly so that one 8 B line of the NSDU is duplicated between NPDU fragments and a different 8 B line of the NSDU is not contained in any NPDU fragment. (I.e., the total number of fragment lines and octets is correct, but one 8 B line is duplicated and another is missing.) The TD sends this set of fragmented NPDUs to the DUT to see whether the DUT will detect the missing line. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	The DUT is expected to detect that one line is missing from the reassembled NSDU. After a timeout, the DUT is expected to a) discard the NPDU fragments that failed to complete reassembly and b) send an ICMPv4 TimeExceeded error PDU with reason code 1, meaning "fragment reassembly time exceeded", to the source IP address specified in those fragmented NPDUs.

Table 20 – IPv4.T12: Large NPDU fragment assembly

Test ID	IPv4.T12
Test name	Large NPDU fragment assembly
Test description	The TD sends a large fragmented ICMP echo NPDU of 65,535 bytes or larger, sometimes called a "ping of death", which has been known to cause faults within IPv4 implementations on reassembly
Reference requirements	Requirement IPv4.R10
Test type	Basic robustness: NSDU reassembly from fragmented NPDUs
Test status	Mandatory
Expected DUT behavior	The DUT reassembles the NPDU correctly for parsing by the higher level protocol, or detects that the aggregated NPDU lines exceed the maximum that it supports and discards the NPDU
Test object	To probe the robustness of the DUT's ability to reassemble an overly long NPDU payload
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]
Test procedure	The TD sends several, large fragmented ICMP echo request NPDUs of 65,535 bytes or larger in size. The TD MAY monitor for any response from the DUT
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 21 – IPv4.T13: Maintenance of service under high load: NPDU fragment reassembly flood

Test ID	IPv4.T13
Test name	Maintenance of service under high load: NPDU fragment reassembly flood
Test description	A number of valid IPv4 NPDUs for multiple established IPv4 associations, each of which claims to be a fragment of a fragmented NDPU, are sent to the DUT in an interleaved manner, without sending all the fragments of the (perhaps virtual) underlying unfragmented NPDU, to attempt to overwhelm the DUT's fragment reassembly storage resources or storage management. See [CRT.Rate_limiting] for additional requirements
Reference requirements	Requirement IPv4.R16
Test type	Load stress
Test status	Mandatory
Expected DUT behavior	The DUT protects itself against a flood of fragmented NPDUs that fail to complete reassembly
Test object	To probe the robustness of the DUT's ability to receive, withstand and recover from a burst of incompletely reassembling fragmented NPDUs, attempting to fill the DUT's NSDU reassembly buffers
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]. The DUT vendor SHALL state a rate limit below which protective measures are not expected to be invoked
Test procedure	The TD repeatedly sends fragmented IPv4 NPDUs for which all the fragments necessary for reassembly are not sent, in an attempt to overwhelm the DUT's NPDU reassembly resources. Test sequences include each of the classes enumerated in 6.7.3
Expected DUT response	The DUT continues to adequately maintain essential functions
Results	Pass or fail
Remarks	

Table 22 – IPv4.T14: Maintenance of service under high load, including network saturation: Raw NPDU flood

Test ID	IPv4.T14
Test name	Maintenance of service under high load, including network saturation: Raw NPDU flood
Test description	A flurry of IPv4 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. During Phase 1, NPDUs are sent in three steps, first as a unicast flood, then as a broadcast flood, then as a multicast flood. See [CRT.Rate_limiting] for additional requirements
Reference requirements	Requirement IPv4.R13
Test type	Load stress
Test status	Mandatory
Expected DUT behavior	The DUT protects itself against a flood of received IPv4 NPDUs. • Phase 1: The DUT continues to function, adequately maintaining all essential functions, in the presence of a sudden burst of received IPv4 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 functions during the period of network overload.
Test object	To evaluate the DUT's ability to receive and withstand a burst of NPDUs addressed to it
Test configuration	A TD is connected to the DUT by an underlying switched network, as specified in [CRT.Test_configuration_1]. The DUT vendor SHALL state a rate limit below which protective measures are not expected to be invoked
Test procedure	The TD sends valid NPDUs that 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 two minutes. IPv4 NPDUs sent to the DUT MAY use any of the classes of explicit or implicit IPv4 addressing (i.e., unicast, broadcast and,multicast), in any combination. Testing SHALL use destination IPv4 addresses that the DUT is configured to recognize, including at least one of each class of recognized IPv4 address. During phase 1, testing SHALL proceed in three or more steps, first using unicast IPv4 destination addresses, then using the IPv4 broadcast address, then using IPv4 multicast addresses that the DUT is configured to recognize. An optional fourth test step MAY intermix these address classes.
Expected DUT response	• Phase 1: The DUT is expected to continue network communication even under high load while adequately maintaining essential functions. • 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

— — — — — —