Prepared (also subject responsible if other)		No.			
ETH/RZX Endre Kulcsár +36 1 437 7918		198 17-CNL 113 631 Uen			
Approved	Checked	Date	Rev	Reference	
ETH/RZXC (Elemer Lelik)		2009-02-24	PA1	GASK2	

ICMPv6 Protocol Modules for TTCN-3 Toolset with TITAN, User Guide

Contents

Prepared (also subject responsible if other)		No.			
ETH/RZX Endre Kulcsár +36 1 437 7918		198 17-CNL 113 631 Uen			
Approved	Checked	Date	Rev	Reference	
ETH/RZXC (Elemer Lelik)		2009-02-24	PA1	GASK2	

1 Introduction

1.1 Revision history

Date	Rev	Characteristics	Prepared
2009-02-24	PA1	First draft version	ETHEKR

1.2 About this Document

1.2.1 How to Read this Document

This is the User Guide for the ICMPv6 protocol module. The ICMPv6 protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Product Revision Information [4] and Function Specification [5].

1.2.2 Presumed Knowledge

To use this protocol module the knowledge of the TTCN-3 language [1] is essential.

1.2.3 References

[1]   ETSI ES 201 873–1 v.3.1.1 (06/2005)
      The Testing and Test Control Notation version 3. Part 1: Core Language

[2]   2/198 17-CRL 113 200 Uen
      Programmer's Technical Reference for the TITAN TTCN-3 Test Executor

[3]   1/1531-CRL 113 200 Uen
      Installation Guide for the TITAN TTCN-3 Test Executor

[4]   109 21-CNL 113 631–1 Uen
      ICMPv6 Protocol Modules for TTCN-3 Toolset with TITAN, Product Revision Information

[5]   155 17-CNL 113 631
      ICMPv6 Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification

[6]   IETF RFC 4443
      Internet Control Message Protocol (ICMPv6) for the Internet Protocol Version 6 (IPv6) Specification

1.2.4 Abbreviations

IETF   Internet Engineering Task Force
IP   Internet Protocol
ICMPv6 Internet Control Message Protocol for IPv6
IPv6   Internet Protocol Version 6
RFC   Request for Comments
TTCN-3 Testing and Test Control Notation version 3

1.2.5 Terminology

TITAN   TTCN-3 Test Executor

1.3 System Requirements

Protocol modules are a set of TTCN-3 source code files that can be used as part of TTCN-3 test suites only. Hence, protocol modules alone do not put specific requirements on the system used. However in order to compile and execute a TTCN-3 test suite using the set of protocol modules the following system requirements must be satisfied:

- TITAN TTCN-3 Test Executor 1.8.pl0.pre2 or higher installed. For installation guide see [3]. Please note:

  1. This version of the protocol module can not be used for defining 32 bit unsigned or larger integers with TITAN versions earlier than 1.8.pl0.pre2.

  2. This version of the protocol module is not compatible with TITAN releases earlier than R7A.

2 Protocol Modules

2.1 Overview

Protocol modules implement the message structures of the corresponding protocol in a formalized way, using the standard specification language TTCN-3. This allows defining of test data (templates) in the TTCN-3 language [1] and correctly encoding/decoding messages when executing test suites using the TITAN TTCN-3 test environment.

Protocol modules are using TITAN's RAW encoding attributes [2] and hence are usable with the TITAN test toolset only.

The table below contains the implemented ICMPv6 messages and the corresponding TTCN-3 type records. Using those type records, templates can be defined to send and receive a given message.

Prepared (also subject responsible if other)			No.		
ETH/RZX Endre Kulcsár +36 1 437 7918			198 17-CNL 113 631 Uen		
Approved	Checked		Date	Rev	Reference
ETH/RZXC (Elemer Lelik)			2009-02-24	PA1	GASK2

Message name	Reference	Corresponding type record in **ICMPv6_Types.ttcn**
Destination Unreachable	[6] 3.1	ICMPv6_DestinationUnreachable
Packet Too Big	[6] 3.2	ICMPv6_PacketTooBig
Time Exceeded Message	[6] 3.3	ICMPv6_TimeExceeded
Parameter Problem	[6] 3.4	ICMPv6_ParameterProblem
Echo Request	[6] 4.1	ICMPv6_EchoRequest
Echo Reply	[6] 4.2	ICMPv6_EchoReply

2.2 Installation

The set of protocol modules can be used in developing TTCN-3 test suites using any text editor. However to make the work more efficient a TTCN-3-enabled text editor is recommended (e.g. nedit, xemacs). Since the ICMPv6 protocol module is used as a part of a TTCN-3 test suite, this requires TTCN-3 Test Executor and a C compiler be installed before the module can be compiled and executed together with other parts of the test suite. For more details on the installation of TTCN-3 Test Executor see the relevant parts of [2]
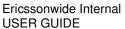
2.3 Configuration

None.

2.4 Encoding, Decoding, Checksum Calculation

Implemented encoding/decoding functions:

Name	Type of formal parameters	Type of return value
f_enc_PDU_ICMPv6	**PDU_ICMPv6, OCT16, OCT16**	**octetstring**
f_dec_PDU_ICMPv6	**octetstring**	**PDU_ICMPv6**

The encoding function **f_enc_PDU_ICMPv6** performs basic RAW encoding [2]. In addition to encoding functionality this function can calculate the checksum field. The checksum is automatically calculated during encoding if the user sets the PDU_ICMPv6's checksum field to '0000'O. For calculating the correct checksum the user needs to provide 2 additional parameters to the encoding function. These parameters are the srcaddr and dstaddr. The value of srcaddr is the source address in the IPv6 packet. The value of dstaddr is the destination address in the IPv6 packet.
The default value for srcaddr is the dummy value '00000000000000000000000000000000'O.
The default value for dstaddr is the dummy value '00000000000000000000000000000000'O.
If the user sets the PDU_ICMPv6's checksum field to a value different from '0000'O then this user defined value will appear in the encoded message.

Prepared (also subject responsible if other)		No.		
ETH/RZX Endre Kulcsár +36 1 437 7918		198 17-CNL 113 631 Uen		
Approved	Checked	Date	Rev	Reference
ETH/RZXC (Elemer Lelik)		2009-02-24	PA1	GASK2

The decoding function **f_dec_PDU_ICMPv6** performs basic RAW decoding [2]. The checksum field is not verified during decoding.

There is an additional function which verifies the checksum field in an encoded ICMPv6 message:

Name	Type of formal parameters	Type of return value
f_ICMPv6_verify_checksum	**octetstring, OCT16, OCT16**	**boolean**

The inputs into this function are the encoded ICMPv6 message and the source and destination addresses of the IPv6 packet.
The default value for srcaddr is the dummy value '00000000000000000000000000000000'O.
The default value for dstaddr is the dummy value '00000000000000000000000000000000'O.
The function returns "true" if the checksum is correct and "false" if it is incorrect.

Prepared (also subject responsible if other)		No.		
ETH/RZX Endre Kulcsár +36 1 437 7918		198 17-CNL 113 631 Uen		
Approved	Checked	Date	Rev	Reference
ETH/RZXC (Elemer Lelik)		2009-02-24	PA1	GASK2