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

Endre Kulcsár

Version 198 17-CNL 113 675, Rev. A, 2010-07-01

Table of Contents

About This Document . . . 1

How to Read This Document . . . 1

Presumed Knowledge . . . 1

System Requirements . . . 1

Protocol Modules . . . 1

Overview . . . 1

Installation . . . 1

Configuration . . . 2

Encoding, Decoding, Checksum Calculation . . . 2

Demo . . . 3

Terminology . . . 3

Abbreviations . . . 3

References . . . 3

About This Document

How to Read This Document

This is the User Guide for the TCP protocol module. The TCP protocol module is developed for the TTCN-3 Toolset with TITAN. This document should be read together with Function Specification [4].

Presumed Knowledge

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

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 3, 1.8.pl0.pre2 or higher installed. For installation guide see [3].

NOTE This version of the protocol module can not be used for defining 32bit unsigned or larger integers with TITAN versions earlier than 1.8.pl0.pre2. This version of the protocol module is not compatible with TITANreleases earlier than R7A.

Protocol Modules

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 file *TCP_Types.ttcn* includes the top level PDU_TCP and it subtypes which correspond to the structure given in [5]. Using these types, templates can be defined to send and receive a given message.

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 TCP 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].

Configuration

None.

Encoding, Decoding, Checksum Calculation

Implemented encoding/decoding functions:

Name	Type of formal parameters	Type of return value
f_enc_PDU_TCP	IP_Address, // src. IPv4 or IPv6 address IP_Address, //dest. IPv4 or IPv6 address PDU_TCP, boolean // automatically calculate data offset boolean // automatically calculate checksum	octetstring
f_dec_PDU_TCP	octetstring	PDU_TCP

The encoding function f_enc_PDU_TCP performs basic RAW encoding [2]. The data offset field is automatically calculated if the user sets the first boolean parameter (pl_autoDataOffset) to true. The checksum is automatically calculated during encoding if the user sets the second boolean parameter (pl_autoCheckSum) to true. For calculating the correct checksum the user needs to provide 2 additional parameters to the encoding function. These parameters are the pl_ip_source and pl_ip_dest which are the source and destination IP addresses (either a pair of IPv4 or a pair of IPv6 addresses).

The decoding function f_dec_PDU_TCP 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 TCP message:

Name	Type of formal parameters	Type of return value
f_TCP_verify_checksum	octetstring IP_Address, IP_Address	boolean

The inputs into this function are the encoded TCP message and the source and destination addresses (either a pair of IPv4 or a pair of IPv6 addresses). The function returns true if the checksum is correct and false if it is incorrect.

Demo

The demo directory contains the file *Handle_TCP_Connections.ttcn*. This file includes some basic functions which can be used to send and receive TCP packets. The function `f_TCP_CreatePayload` creates a TCP/IP packet which carries the payload `pl_data`. The function `f_TCP_VerifyHeader` verifies the TCP and IP headers in a received message. It also verifies the TCP checksum and saves the Sequence Number and Acknowledge Number in a local variable of type ConnectionList.

Terminology

TITAN:
TTCN-3 Test Executor.

Abbreviations

IETF
: Internet Engineering Task Force

IP
: Internet Protocol

IPv6
: Internet Protocol Version 6

RFC
: Request for Comments

TCP
: Transmission Control Protocol

TTCN-3
: Testing and Test Control Notation version 3

References

[1] ETSI ES 201 873–1 v.4.1.1 (2009-06)
The Testing and Test Control Notation version 3. Part 1: Core Language

[2] Programmer’s Technical Reference for the TITAN TTCN-3 Test Executor

[3] Installation Guide for the TITAN TTCN-3 Test Executor

[4] TCP Protocol Modules for TTCN-3 Toolset with TITAN, Function Specification

[5] IETF RFC 793