

# 6lowpan Test Suite Documentation

29 april 2012

## 1 Coverage

The target RFCs for this test suite are :

- RFC 4944    Transmission of IPv6 over IEEE 802.15.4
- RFC 6282    Header compression for 6lowpan

The test suite provides type definitions for the messages used by these two RFCs (except the HC1 compression scheme which is now deprecated), and a few sample testcases demonstrating how to use them.

The Codec covers these two RFCs and the types from the IPv6 library that were needed for the sample test cases.

The Platform Adapters implements the header compression and decompression functions for 6lowpan (RFC 6282) in stateless mode only.

The System Adapter provides a system port implementation that handles IEEE 802.15.4 link layer.

## 2 Architecture

### 2.1.1 Repository content

The content of the repository is detailed in the following table:

6lowpan-documentation.odt 6lowpan-documentation.pdf	The present document
Ats6lowpanParameters.ttcn	The file defining the module parameters for the runtime configuration of the test suite
Ats6lowpan.ttcn	The source file of the test suite
LibCommon/ LibIpv6/	Directories containing the files imported from the IPv6 library
LibIpv6/Lib6lowpan/LibIpv6_6lowpan_Interface.ttcn LibIpv6/Lib6lowpan/LibIpv6_6lowpan_TypesAndValues.ttcn	An extension to the IPv6 library to support 6lowpan
CD/6lowpan-codets.cpp CD/6lowpan.h CD/codec.cpp CD/codec.h CD/libipv6-codets.cpp CD/libipv6.h	The source files for the CoDec
PA/functions.cpp	The source files of the Platform Adapter
SA/6lowpan-port.cpp	The source files of the System Adapter

SA/6lowpan-port.h SA/802.15.4-framer.cpp SA/802.15.4-framer.h SA/init.cpp	
wscript	The waf build script used to compile the test suite
config.parfile	The runtime configuration file for tuning the module parameters (used by Rational Systems Tester only)
__build__/ 	A temporary directory created by waf ans which stores intermediate and output files

### 2.1.2 TTCN-3 Abstract Test Suite

The abstract test suite is written in TTCN-3, it is based on the LibIPv6 produced by the STF 276. Four new modules were written to support 6lowpan testing :

- **LibIPv6\_6lowpan\_TypesAndValues** contains the types definitions for RFC 4944 and RFC 6282 messages.
- **LibIPv6\_6lowpan\_Interface** contains the definition of the 6lowpan system port used to communicate with the Implementation Under Test and the definition of the configuration message for this port.
- **Ats6lowpanParameters** contains the TTCN-3 module parameters for the runtime configuration of the test suite (IUT hardware address + SUT port configurations)
- **Ats6lowpan** contains the 6lowpan abstract test suite

All other TTCN-3 source files were imported from the IPv6 library.

### 2.1.3 System Adapter

Figure 1 represents the framework used for the System Adapter (interaction between the TTCN-3 abstract test suite and the Implementation under Test).

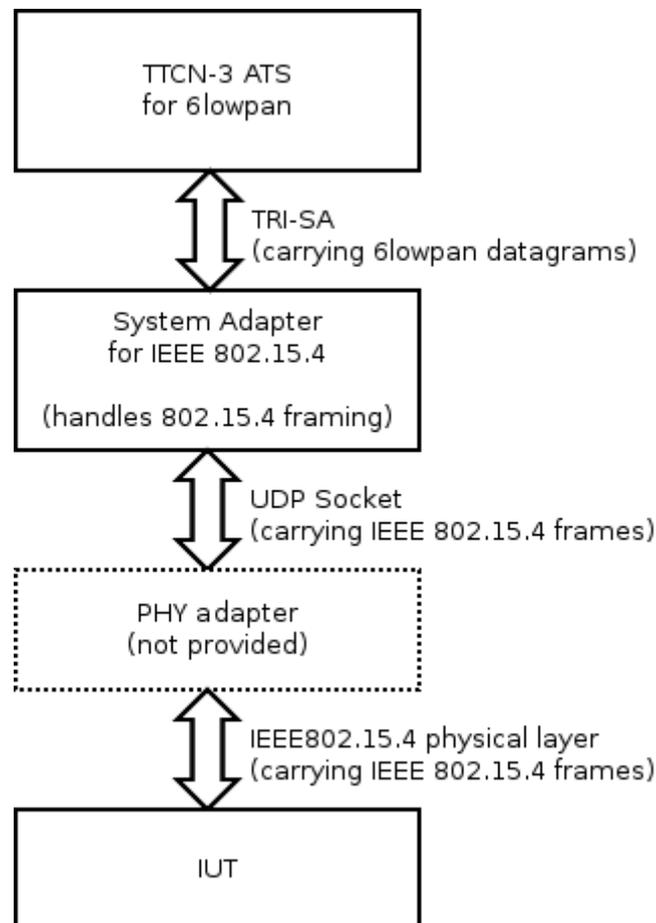


Figure 1: Framework of the System Adapter

The System Adapter which is included with this test suite implements the IEEE 802.15.4 link-layer. It provides a TTCN-3 system port that supports the encapsulation of 6lowpan packets into IEEE 802.15.4 frames.

The System Adapter relies on an external program to handle the physical layer. The communication between the TTCN-3 System Adapter and the Physical Layer adapter is performed over a UDP socket. The purpose of this separation is to let the test suite remain independent from the physical layer (since IEEE 802.15.4 can be used over various physical layers: 800MHz radio, 2.4GHz radio, PLC,...). Figure 2 shows how a 2.4Ghz radio adapter could be implemented using a Telos-B dongle.

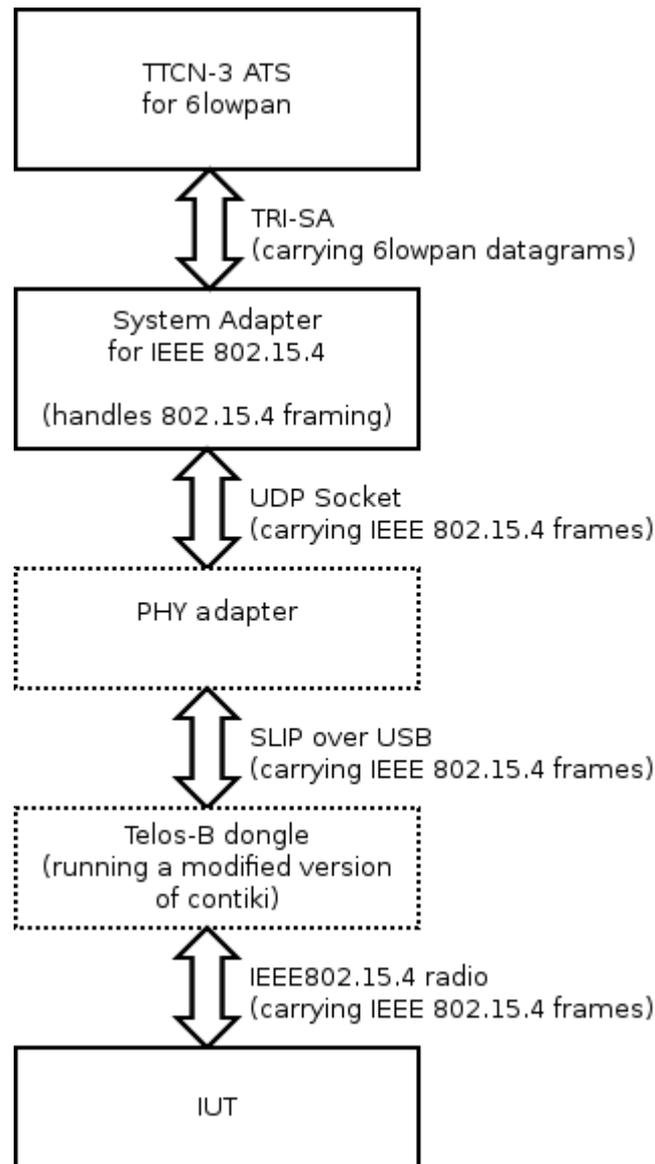


Figure 2: Example of physical adapter implementation using a Telos-B dongle

### 2.1.4 Codec

The CoDec implementation is based on the T3DevKit codec generator. This generator parses the TTCN-3 source files from the test suite and generates a C++ class for each structured type (record, set, record of, set of, union) defined in the ATS. These classes implement a default codec that can be customised by adding «codet » functions named PreDecode(), PostDecode(), etc. These additional functions implement the codec semantics specific to this type (eg. the field named PayloadType indicates the length of the payload).

More details can be found in T3DevKit's user manual: <http://t3devkit.gforge.inria.fr/doc/userref/>

## 3 Requirements

In order to use this test suite, the following software is required:

- T3DevKit (used for implementing the CoDec and adapters), which is available at:

<http://www.irisa.fr/tipi/wiki/doku.php/t3devkit>

The 0.11.0 release is not suitable. As of April 2012 it is necessary to work with the HEAD version from the subversion repository:

```
svn checkout svn://scm.gforge.inria.fr/svnroot/t3devkit/t3devkit/trunk
```

- A TTCN-3 tool which is supported by T3DevKit (currently this includes IBM Rational Systems Tester and Elvior Testcast T3)
- The toolchain and libraries needed by the TTCN-3 tool and by T3DevKit.
  - a C/C++ compiler (g++ or msvc (from the Windows SDK))
  - flex, bison
  - python
  - the boost libraries

If running on Debian or Ubuntu you just need to run:

```
apt-get install flex bison libboost-all-dev g++ gcc python2
```

The installation procedure for T3DevKit is detailed in section « Getting Started » from the user manual: <http://t3devkit.gforge.inria.fr/doc/userref/userrefch2.html>

## 4 Usage

The build process is fully automated by T3DevKit's build scripts based on waf. Before compiling the test suite, you can try with an example from the T3DevKit directory:

```
cd examples/HelloWorld
twaf configure          # configure the build
twaf                   # build the executable test suite
```

The procedure to run the Executable Test Suite depends on the tool being used.

- with Rational Systems Tester, you can just run the following command line, it will read the TTCN-3 runtime module parameters from the file config.parfile.

```
twaf --run
```
- with Testcast T3, the build will just compile the System Adapter and the Codec and link them separately as shared libraries that will be usable by the tool. These libraries will be located in the subdirectory `__build__/default/` from the 6lowpan test suite directory. You need to set-up a TTCN-3 project in Testcast's IDE and configure it properly to use the generated adapters and codecs.

The Test Suite needs to be configured in order to run properly. There are 4 module parameters defined in the `Ats6lowpanParameters` module.

- **IUT\_MAC\_ADDRESS** is an octetstring that must be set to the hardware address of the network interface of the IUT. Example:

```
Ats6lowpanParameters.IUT_MAC_ADDRESS := '0200000001020304'0
```

- **IEEE\_802\_15\_4\_PAN\_ID** is an octetstring that configures the IEEE 802.15.4 PAN ID to be used to communicate with the IUT. Example:

```
Ats6lowpanParameters.IEEE_802_15_4_PAN_ID := '0000'0;
```

- **SIXLOWPAN\_ADAPTER\_IP** and **SIXLOWPAN\_ADAPTER\_PORT** indicate the IP address and UDP port number of the PHY adapter to be contacted for transmitting IEEE 802.15.4 frames to the radio (or any other physical layer). Example:

```
Ats6lowpanParameters .SIXLOWPAN_ADAPTER_IP := "127.0.0.1";
```

```
Ats6lowpanParameters.SIXLOWPAN_ADAPTER_PORT := 13000;
```