ETN

Enrico Deiana, Emanuele Del Sozzo

Introduction

This paper describes the main data structures contained in the header file *etn.h*. In particular, it focuses on *Encoder* and *Decoder* data structures and how they are related to data types.
Header *etn.h* is part of *libetn*, a C library for encoding, decoding, and verifying **ETN** types. The library's interface is build around readers and writers, which can be overloaded to encode/decode/verify memory buffers, file descriptors, etc.
Library *libetn* interacts with code generated using *eg2source*, that generates C code containing structures that describe types, and *libetn* consumes these definitions to perform encoding/decoding/verifying.

Encoder

EtnEncoder is a "writer" and can be considered a base class. This data structure contains two function pointers, a *red-black tree* and an index.
The two function pointers are one to a *write* function and one to a *flush* function. The actual implementations of both are declared in the header file *packetEncoder.h* and defined in file *packetEncoder.c*.
The *write* function writes data to a packet sending fragments as maximum size reached. The final (non-full) fragment is not sent because there may be remaining data to write to it.
The *flush* function flushes packet encoder, sending the packet as the final fragment. Note that if the write does not fill the first fragment, then nothing will be sent until flush is called.

The *red-black tree* structure is used to

the *index* is used to

EtnBufferEncoder is a specialization of *Encoder*. It encodes to provided memory range. In contains an *encoder* and two pointers to *uint8_t* type that point to a memory range.
EtnNullEncoder is a specialization of *Encoder*. It is actually a quite useless structure since it contains only an *encoder*. *PacketEncoder* is specialization of *Encoder*. It encodes to packet and sends it on flush. It contains a *EtnBufferEncoder*, a *Packet*, the total length of the packet, a *boolean* that indicates whether a packet was sent and a *Connection*.

Decoder

EtnDecoder is a "reader" and can be considered a base class. This data structure contains a function pointer to a *read*, a pointer to the data, the data length and an index to the next data. The actual implementation of *read* is defined in file *decoder.c*.
EtnBufferDecoder is a specialization of *Decoder*. It encodes to provided memory range. In contains an *decoder* and two pointers to *uint8_t* type that point to a memory range.

PathDecoder is a specialization of *Decoder*. it decodes from provided path. It contains a *EtnBufferDecoder* and a *uint8_t* to the original data.