Abstract—How to encode/decode PDUs of protocols is always a time-wasted complicated work in the phase of developing the systems for the telecommunication or internet networks. As ASN.1’s well known efficient way to describe the complicated protocols, many protocols PDUs of the presentation layer or the application layer (e.g TCAP, INAP) in the telecommunication network and internet have been represented in ASN.1. One of the key protocols of SS7 network in the legacy telecommunication network is the ISDN user part (short named as ISUP). Its development and extension have an important influence on the related voice telecommunications. However, no ASN.1 presentation file of the ISUP PDUs appears in the ITU-T specifications. One important reason is the special structure of ISUP PDUs against ASN.1 BER which forms the base for the ASN.1 presentation of any protocol PDUs like those PDUs of the protocols (e.g TCAP, INAP, MAP, SNMP). Because of the wide usage of the ASN.1 and the importance of the ISUP, it makes sense to parse the ISUP PDUs with the ASN.1. What’s more, there are many open free tools available for parsing to compile ASN.1 into other languages easily, which can ease developing ISUP PDUs parsing tools. In this paper, we present a new effective approach to encode/decode it in the same way as the other protocol PDUs presented in ASN.1. By this approach, we have developed a library available under Linux for parsing ISUP PDUs via ASN.1.

I. INTRODUCTION

ISUP is a key protocol of SS7 network and was of prime importance in the telecommunications in the 20th century. As it will take a long way to evaluate the telecommunication network from the legacy SS7 network into the next generation network (NGN), ISUP protocol will be still developed further for passing the NGN network parameters or the IN parameters in the converged network of the fixed network with the mobile network. ISUP messages in the future can be divided as shown in Figure 1 and Figure 2. Of course encoding/decoding the newly developed ISUP like the 2002 version plays an important role in deploying the new messages or the new parameters in the given messages.
II. What's ASN.1 and BER?

ASN.1 is the acronym for Abstract Syntax Notation One, a notation for describing structured information data; typically, information data distributed in the networks which can be the homogeneous network or the heterogeneous network across different interfaces or communication medium. ASN.1 has been specified as the international standard by ITU-T with the X.680 to X.683. Because of its efficiency with the abstractive description of data structures, it is widely used in the specifications of communication protocols, and in particular, is employed in virtually all of the emerging standards for the application layer of Open Systems Interconnection (OSI).

Given any ASN.1 description of a message, a representation can be derived mechanically by applying a set of encoding rules. The ASN.1 Basic Encoding Rules (BER) is one of various ASN.1 encoding rules and is also the most widely used one. A clear advantage of the use of encoding rules rather than hand-crafting transfer syntaxes is that application designers do not need to be familiar with their details. All can be done by special ASN.1 parsing compiler tools.

III. Problems of ASN.1 Application in ISUP PDUs

Not like many with ASN.1 designed protocols (e.g. MAP, INAP), the structure of ISUP PDUs does not concoct with the ASN.1 BER, as you can see in Table 1. As we have already known, the BER is defined with the format of a triplet TLV (Tag, Length, Value). As the Figure 4 shows, ISUP PDUs can be divided into 4 parts: Routing label and CIC; mandatory fixed part; mandatory part of the parameters with variable length and optional parameter part. But for mandatory fixed part and mandatory variable part, they all conflict with TLV style. So if we want to represent it with ASN.1, we have to modify some octets in its PDU, and then parse it by one special compiler.

IV. Our Approach of Parsing ISUP PDUs via ASN.1

Because of the above reasons, we should reform the representation of ISUP PDUs, to make it consistent with the ASN.1 BER. Of course, that does not mean to modify the structure of ISUP PDUs, but to reconstruct its representation as TLV style just when we view it with our parser.

Related to the 4 parts of one ISUP PDU, only the last 3 parts need to encode/decode via ASN.1 (The length of the routing label and CIC is fixed). The optional part follows to the ASN.1 BER, so what we need to reconstruct are the secondary part and the third part of one ISUP PDU.

Table 2 shows how the ISUP PDUs are represented and then do the procedure (see in Figure 4) with the following steps:
Firstly, it is proposed that ISUP PDUs are represented abstractly with ASN.1 according to the reconstructed structure of ISUP PDUs.

ISUPPackageITU1999 DEFINITIONS ::= BEGIN
EXPORTS -- EVERYTHING
PackageType ;
PackageType ::= CHOICE
{  
aCM [6] IMPLICIT ACM,
aNswer [9] IMPLICIT ANSWER,
  
 blocking [19] IMPLICIT blocking,
  blockingAck [21] IMPLICIT blockingAck,
  callprogress [44] IMPLICIT callprogress,
  
  release [12] IMPLICIT release,
  releaseComplete [16] IMPLICIT releaseComplete,
  
  usrparttest [52] IMPLICIT usrparttest,
  usrToUsrInfo [45] IMPLICIT usrToUsrInfo
}
END

The above representation form includes all of the types of ISUP messages at the beginning of the ISUP ASN.1 representation. As you can see from the above ASN.1 definition, each ISUP message PDU is defined with one related new ASN.1 type and the structure for each new ASN.1 type is defined in its ASN.1 form, the example of which is the RELEASE message as explained in the following ASN.1 type definition.

RELEASE ::= SET {
  pointerMandatoryParameter PointerMandatoryParameter,  
  pointerOptionalParameter PointerOptionalParameter,  
  causeIndicators CauseIndicators,  
  redirectionNumber RedirectionNumber OPTIONAL,  
  redirectionInformation RedirectionInformation OPTIONAL  
}

END

Secondly, compile this ASN.1 file into C/C++ code by some compilers (we chose the compiler called SNACC here).

Thirdly, modify the generated C/C++ codec file, according to the structure of the ISUP message type. That is the key step of this approach. Here are the guidelines for modifying the codec file:

a) When decoding, the extra added parts in Table 2 against Table 1(e.g the length of that message type) should be added, because octets or bits of the message length do not exist in the real ISUP PDU structure.
b) When decoding, the contents of some additional parts in Table 2 (e.g., the pointer to the parameter M, etc.), which is just a pointer, other than parameters of ISUP message. We add it in ASN.1 file just for generation C/C++ via ASN.1 compiler) are not what we take care of. So the results of decoding these parts should be discarded and not put in the result buffer.

When encoding, the extra added part (e.g., many length octets except mandatory variable part and optional part) should be discarded and not put in the generated binary or bits streams of the PDUs.

All of the above steps are just one-time works and the same for all types of the ISUP PDUs, since ISUP PDUs for each kind of message are regular and can be manipulated in the same way.

Fourthly, compile the ISUP ASN.1 file with ASN.1 parsing library and then generate the executable files with application files.

V. CONCLUSION

In this paper, one approach for encoding/decoding ISUP PDUs with the ASN.1 BER had been described, and deployed in the software development of our ISUP protocol software. This is the first time to realize the process of encoding/decoding the ISUP PDUs in the same way as the GSM MAP or INAP protocols with the ASN.1 BER. The whole procedure is not complex and the modification to the generated C/C++ codes is just one-time work. In particular, we have developed a library (for C) available under Linux (See libisup.a in Figure 4) for encoding/decoding ISUP messages by this approach. Consequently this approach is obviously efficient, when the ISUP protocol is evaluated into the new version like that version ISUP (1999). In this new ISUP version, many parameters used in the INAP (e.g., RemoteOperations) are introduced and suitable for the ASN.1 description of them. This encourages us to make more research on the ASN.1 evaluation of the ISUP protocol, even though this is not the fully explicit ASN.1 definition of the ISUP. It is sure that the perfective ASN.1 definition of the ISUP PDUs will be available in the near future.

REFERENCES