

汎用方式ASN.1ライトウェイト符号化規則
エンコーダ・デコーダの評価

6V-7

星加 修 古城 孝一 内山 光一
(株)東芝 情報処理・機器技術研究所

1. はじめに

OSIの応用層では、一般的にプロトコルデータ単位(PDU)のデータ構造がASN.1(抽象構文記法1)により抽象構文^[1]として定義されている。また、そのデータ構造から転送されるビット列への変換は符号化規則によって規定されている。現在ISOでは、ネットワークに対する高度化と多様化のニーズに応じて、ASN.1の大幅な見直し作業が行われている。符号化規則に関しては、現在国際標準として定められている基本符号化規則(BER)^[2]の他に新たに3つの符号化規則が、提案されている。そのうちの一つであるライトウェイト符号化規則(LWER)^[3]については、ASN.1の型毎にプログラムを生成するコンパイラ方式のエンコーダ/デコーダの評価が進められている^{[4][5]}。一方、報告者らはエンコーダ/デコーダの利用者が既存のアプリケーションをほとんど書き換える必要がない汎用的なエンコーダ/デコーダも有効であると考え、開発及びその評価を行ったので本稿で報告する。

2. 汎用方式エンコーダ/デコーダについて

2.1 開発の方針

今回の開発は、以下の方針に従い行った。

(1) エンコーダ/デコーダの利用者の符号化規則の増加による負担を軽減するために、報告者らにより既に開発されているBERを用いた汎用方式のエンコーダ/デコーダ^[6]とインターフェース、符号化/復号化情報のテーブルのデータ構造を同一にし、フラグによりLWERとBERのエンコーダ/デコーダを呼び分けることができるようにする。このことにより、利用者は既存のアプリケーションをほとんど書き換える必要はなくなる。

(2) BERを用いたエンコーダ/デコーダの内部で使用されている関数を、できる限り使用しプログラムのサイズが小さくなるようにする。

2.2 実装方式

符号化/復号化は2つのテーブルを与えることにより実現される。一方のテーブルは、ASN.1により記述された抽象構文が定義するデータ構造を示すテーブルで、アプリケーションにより一意であり変更されることのない静的なテーブル(ASIT:Abstract Syntax Information Table)である。もう一方のテーブルは、そのテーブルの一行がASITの一行に対応し、ASITの示す各要素に対応するデータの長さ及びデータの先頭ポイントにより構成される動的なテーブル(DAVIT:Data Value Information Table)である。このテーブルは、符号化の時は利用者により使用前に設定される。

図1に今回開発したエンコーダ/デコーダの処理の流れを示す。エンコーダはASITとDAVITを入力としPDUを作成する(図1の実線部分)。また、デコーダはASITとPDUを入力としDAVITを作成する(図1の点線部分)。

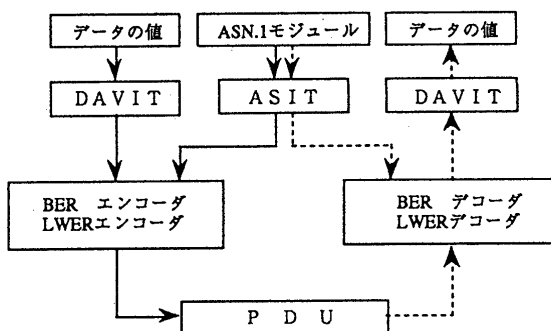


図1 エンコーダ・デコーダの処理の流れ

3. 評価

3.1 評価方法

今回開発したエンコーダ/デコーダの符号化/復号化の処理時間を、BERを用いた汎用方式のエンコーダ/デコーダ、LWERを用いて開発したコンパイラ方式のエンコーダ/デコーダと比較することによって評価を行った。処理時間はエンコーダについては、コ

ールしてからPDUが作成されるまでの時間、デコーダについては汎用方式ではコールしてからDAVITが作成されるまでの時間、コンパイラ方式ではコールしてから抽象構文に相当するC言語の型に値が設定されるまでの時間である。なお、測定に用いた計算機は当社のラップトップワークステーションSPARC LT (クロック周波数: 25 MHz) である。また、LWERの転送構文はSPARC LTに特有の32ビット/ワード長のBig Endianとした。

3.2 測定結果

測定に用いたサンプルデータのうち主なものを図2に示す。そのうちEx1の場合の符号化/復号化の結果を図3、4に示す。

```
Ex1::=SEQUENCE{
  int1[1] IMPLICIT INTEGER,
  int2[2] IMPLICIT INTEGER,
  ...
  intN[N]IMPLICIT INTEGER }
```

```
Ex2::=SEQUENCE OF SEQUENCE{
  int1[1] IMPLICIT INTEGER,
  int2[2] IMPLICIT INTEGER }
```

図2 測定のためのサンプルデータ

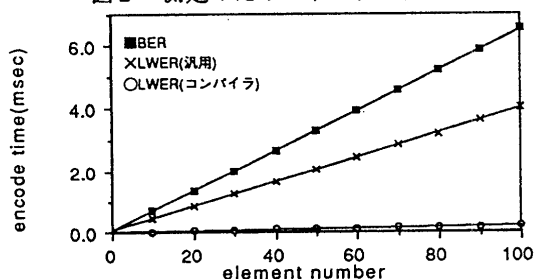


図3 符号化処理時間

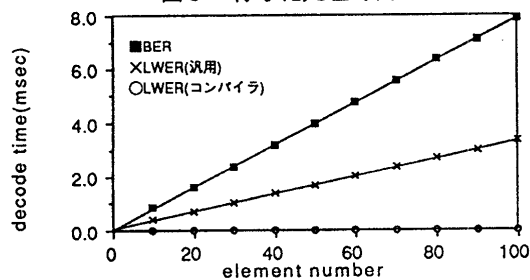


図4 復号化処理時間

3.3 結果の考察

図3、4より符号化/復号化ともコンパイラ方式のLWER、汎用方式のLWER、BERの順に処理時間を要することがわかる。まず、符号化において汎用方

式のLWERは、コンパイラ方式と比較して20~30倍処理時間がかかっている。これは、ASITの要素に対応するDAVITの一行を探す処理など、汎用性によるオーバーヘッドが生じていると考えられる。また、BERと比較すると2倍近く高速であるが、これはLWERにはないタグや長さの符号化が原因と考えられる。次に復号化においてコンパイラ方式では、Ex1の例は可変長の要素を含まないので、抽象構文に相当するC言語の型の先頭ポインタをPDUの先頭ポインタに代入するだけなので、ほとんどオーバーヘッドは生じない。一方汎用方式のLWERは、固定長符号化のメリットにより、BERと比較すると処理は簡単になるがDAVITを作成するという処理の分だけ、コンパイラ方式と比較して遅くなると考えられる。

4. まとめ

新しい符号化規則による既存のアプリケーションの書き換えをできる限り少なくすることを方針とした汎用方式のLWERエンコーダ/デコーダの評価を行った。しかし、汎用的にしたことによりLWERの特徴(プログラミング言語への対応、計算機の内部表現の考慮など)を生かした処理が出来ず、BERよりは高速となったが、コンパイラ方式と比較するとかなり処理時間を要した。

今後は、汎用方式のLWERについては長所を残しながら処理速度の改善を検討していく。一方、コンパイラ方式のLWERエンコーダ/デコーダは、LWERの特徴を生かした高速な処理が可能なので、分散処理の基盤技術として高速な処理が要求されているRPC(Remote Procedure Call)のプロトコルマシンに適用し、現在開発中である。

<参考文献>

- [1]ISO/IEC 8824 "Specification of Abstract Syntax Notation One(ASN.1)"
- [2]ISO/IEC 8825 "Specification of Basic Encoding Rules for Abstract Syntax Notation One(ASN.1)"
- [3]ISO/IEC WD 8825-4 "Specification of ASN.1 Encoding Rules-Part4:Light Weight Encoding Rules"
- [4]堀内、小花、鈴木、"OSI応用層プロトコル用ASN.1ライトウェイト符号化規則に関する一考察" 情処第43回全国大会、1-209~210
- [5]堀内、小花、鈴木、"ASN.1ライトウェイト符号化規則の評価" 情処第44回全国大会、1-163~164
- [6]内山、"OSI応用層におけるASN.1開発ツール" 情処第38回全国大会、1621~1622