

## 快速传送语法的设计与实现\*

龚正虎

(电子计算机系)

**摘要** 国际标准化组织提出的抽象语法 ASN-1 和传送语法 ASN-BER 功能强, 通用灵活, 但在一般实现中存在编码效率低和开销大的缺点。本文讨论提高编码效率和速度的一般方法, 以及描述快速传送语法的设计与实现技术。

**关键词** 计算机网络, 开放系统互联参考模型, 表示层, 抽象语法, 传送语法

**分类号** TP393. 1

## 1 ASN-1 和 ASN-BER 简介

在国际标准化组织(ISO)所提出的开放系统互联参考模型(OSI/RM)中, 两个应用层实体(文件传输访问, 分布数据库管理)需要交换多种类型的数据。这些数据的类型用一组标记规则来定义, 称之为抽象语法(Abstract Syntax)。为了使收方能准确地解释发方的数据, 在 OSI 环境中传送的应用层数据必须有严格定义的传送格式, 这就是传送语法(Transfer Syntax), 一套用二进制位串表示不同类型数据的编码规则。ISO 颁布的抽象语法和传送语法分别为 ASN-1(Abstract Syntax Notation One)和 ASN - BER(Basic Encoding Rules for ASN - 1), 它们的前身是 CCITT 的 X. 409<sup>[1,2,3]</sup>。

如程序设计语言, ASN-1 的类型分基本类型和构造类型两种。基本类型包括 integer, boolean, bitstring 等, 构造类型包括 CHOICE, SEQUENCE, SET 等结构。构造类型的分项可以是构造类型(嵌套性)。类型用标签号(number of tag)来标识, 它分四大类(class): universal, application, context-specific, private。限定词 IMPLICIT 定义隐含类型, 表明通讯双方都明了的类型。DEFAULT 定义省缺类型, 当发方省略该数值时, 收方可取省缺值。OPTIONAL 定义可选类型。下面定义的 personal 类型由三项(name, age, sex)组成, 方括号内的数字为标签号。

```
personal ::= [application 3] SEQUENCE{
    name    [0] IMPLICIT visitstring,
    age     [1] integer OPTIONAL,
    sex     [2] boolean DEFAULT(male)}
```

\* 1992年4月25日收稿

ASN-BER 是一种面向字节 (Octet-Oriented) 的, 递归式的 “tag — length — content” 编码结构。tag 标识数值类型, length 指明数值长度, content 包含数值内容, content 本身又可以是 “tag — length — content” (嵌套性)。设 WANG FANG (女, 28岁) 为 personal 类型的一个实例, 按 ASN-BER 规则编码所得结果如下 (IC 表示 28, EF 表示女):

```

personal length (content)
  63    23
    SEQUENCE length (content)
      36    21
        name length content
          80    9 "WANG FANG"
        age length (content)
          A1    3
            integer length content
              02    1    "IC"
        sex length (content)
          A2    3
            boolean length content
              01    1    "FF"

```

## 2 提高编码效率和速度的途径

ASN-BER 功能强, 通用灵活, 但是, 在它的一般实现中存在编码效率低, 开销大的缺点, 阻碍了它的应用和推广。人们普遍反映, ISO 表示层的效率低, 其主要原因就在于此。编码效率定义为有用信息在编码信息中所占百分比。ASN-BER 的编码效率一般为 20—50%<sup>[7]</sup>。上节实例中, 有用信息为 11 字节, 编码后数据为 25 字节, 效率为 44%。编码速度低意味着占用 CPU 时间多。有人试验<sup>[8]</sup>, 按 ASN-BER 对一个数组编码所用 CPU 时间为将数组从内存一个地方复制到另外一个地方所用 CPU 时间的 20 多倍。在计算机网络的实际设计中, 这种编码效率和速度是不可接受的。

提高编码效率的关键是最大限度减少冗余信息 (tag 和 length), 在不改变 ASN-1 前提下, 可采用下述两种方法减少冗余信息:

(1) 充分利用限定词 IMPLICIT 定义数据类型。例如, personal 类型中的 name 使用 IMPLICIT 之后, 编码时就省去了 visitstring 的 tag 和 length 信息。这个方法的优点是保持了 ASN-BER 不变, 但效率提高有限。

(2) 简化编码规则: 采用 ASN-BER 子集, 或重新定义快速传送语法 (Fast Transfer Syntax)。根据 ISO 表示层协议, 两个表示层实体在联接建立时可协商选择传送语法。这使得它们有机会采用快速传送语法交换数据<sup>[7,8,9]</sup>。这个方法可使编码效率大幅度提高, 缺点是表示层要多安装一套编码规则。

编码效率的提高意味着编码速度的提高, 但是, 提高编码速度的关键是 ASN-BER 实

现技术的改进。

作者在 OSI 协议实现技术研究中，定义了一种新的快速传送语法，实现中采用了直接变量编码技术，获得很高的编码效率和速度。

### 3 快速传送语法的设计

快速传送语法应该根据 OSI 环境来设计。如果通讯双方都明了待交换数据的类型，那么“tag”信息可以略去，采用“tag——content”编码结构。对于给定类型的数据，如果其长度是固定的，那么“Length”信息可以去掉，采用“tag-content”编码结构。ASN-BER 按字节编码，如果改写为按位编码，也能提高编码效率。

本文采用的快速传送语法是根据 YHNET 网络环境设计的，它是 ASN-BER 的一个子集，其要点如下：

- (1) SEQUENCE-OF, SET, SET-OF 按 SEQUENCE 方法处理。
- (2) 带标签号的 SEQUENCE 处理为 IMPLICIT SEQUENCE，其“length”等于各分项“length”形式。
- (3) CHOICE 的分项按顺序赋予 context-specific 标签号。分项的 tag 按 IMPLICIT 处理，其“length”等于数值的实际长度。
- (4) “length”不支持 indefinite form。
- (5) bitstring 和 octetstring 不支持结构型编码。
- (6) “tag”的第1字节中，P/C 位去掉，标签号由5位扩大为6位。
- (7) “tag”的第7、6位定义如下：

- 00 primitive (不包括 SEQUENCE 等结构类型)
- 01 SEQUENCE
- 10 application
- 11 context-specific

由于 tag 的第7、6位已说明了类型是基本类型(primitive)还是结构类型，因此 P/C 位可去掉。按上述编码规则，对 personal 类型的实例(WANG FANG 女28岁)编码，其效率由44%提高到68.75%。

```
personal length (content)
  63  14
      length content
        9  "WANG FANG"
          length content
            1  "IC"
              length content
                1  "FF"
```

### 4 直接变量编码方法

本文采用的直接变量编码方法如图1所示。所谓直接变量编码是指应用层协议实体的

变量纳入 APDU 定义, 编码/解码程序按 APDU 定义直接取变量之值编码或解码, 编码过程和 APDU 组装过程合二为一, 解码过程和 APDU 拆卸合二为一。APDU (Application Protocol Data Unit) 为应用层协议实体之间交换的数据单元。假定 personal 为一个 APDU 定义。当应用层请求表示层对 personal 的实例 (WANG FANG, 女, 28) 编码之前, 它将有关数据 (WANG FANG, 女, 28) 放在指定变量中。编码请求发出之后, 表示层根据 personal 定义直接取变量之值编码, 并组装报文放入发送缓冲器之中。解码过程相反。

APDU 用 VAX C 语言数组定义, 数组的每个元素为一个描述 SEQUENCE 分项的数据结构。personal 的定义为:

```

struct array_item {
    /* SPECIFY THE SIZE OF THE
    NEXT ARRAY (SUBTREE). */
    short          next_size;
    /* SPECIFY ASN.1 IDENTIFIER:
    CLASS, TAG. */
    unsigned char  asnl_type;
    short          asnl_tag;
    /* SPECIFY LOCAL VARIABES; TYPE, LENGTH AND ADDRESS
    */
    unsigned char  var_type;
    short          var_length;
    int            encode_addr;
    int            decode_addr;
}

struct array_item  personal [3];
    
```

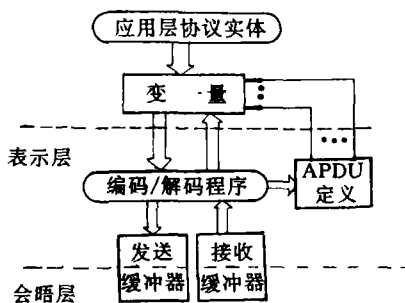


图1 直接变量编码模型

这里, personal 类型用数组 personal [3] 定义, 初始化, 它的三个元素 (array\_item) 分别描述 name, age, sex 的 ASN-1 定义 (class, tag) 以及放置这些数值的变量特性 (type, length, address)。

直接变量编码方法和其它方法<sup>[5, 8]</sup>相比, 编码速度大幅度提高。但是, 它要求应用层协议和表示层协议拟合在一起实现。为了提高 OSI 协议的执行效率, 绝大多数协议实现者将应用层和表示层放在一个进程或任务中实现。这种设计思想和直接变量编码方法的出发点是一致的。

## 5 结束语

本文已讨论了提高 ASN-1 和 ASN-BER 编码效率和速度的一般方法, 这些方法都没有改变 ASN-1 的语法规则。虽然快速传送语法改变了 ASN-BER 的编码规则, 但是, 由于表示层协议有协商选择传送语法的功能, 因此它没有损害 OSI 协议的开放性和标准性。本文最后描述了直接变量编码方法, 很显然它是一种开销最小的 ASN-BER 实现方法。

## 参 考 文 献

- 1 ISO/DIS 8822 Presentation Service Definition  
ISO/DIS 8823 Presentation Protocol Specification
- 2 ISO/DIS 8824. 2 specification of Abstract Syntax Notation One
- 3 ISO/DIS 8825. 2 Basic Encoding rules for ASN. 1
- 4 CCITT X. 409 Presentation Transfer Syntax and Notation
- 5 Pope A R. Encoding X. 409 Presentation Transfer Syntax. ACM comput commun rev 1984, (4)
- 6 Cleghorn C and Ural H. ASNST: An Abstract Syntax Notation One Support Tool. comput commun 1989, 12(5)
- 7 Pimentel J R. Efficient Encoding of Application Layer PDU's for Fieldbus Network. ACM comput commun rev 1988, 18(3)
- 8 Huitema C and Doghri A. Defining Fast Transfer Syntax for the OSI Presentation Protocol. ACM comput commun rev 1989, 19(5)
- 9 Fong K and Reinstedler J. The Development of the OSI Application Layer Protocol Interface. ACM comput commun rev 3 1989, 19(3)
- 10 龚正虎. 关于 ISO/OSI ASN. 1 编码规则实现, 全国第三次分布会议, 1988

## Design and Implementation of a Fast Transfer Syntax

Gong Zhenghu

(Department of Computer Science)

### Abstract

ISO-proposed ASN. 1 (Abstract Syntax Notation One) and ASN-BER (Basic Encoding Rules for ASN. 1) proved to be highly powerful and flexible, but with low efficiency and large overheads in normal implementations. This paper first discusses general methods to improve the encoding efficiency and speed, then describes the design and implementation techniques of a fast transfer syntax.

**Key words** computer network, open system, interconnecting reference model, abstract syntax, transfer syntax