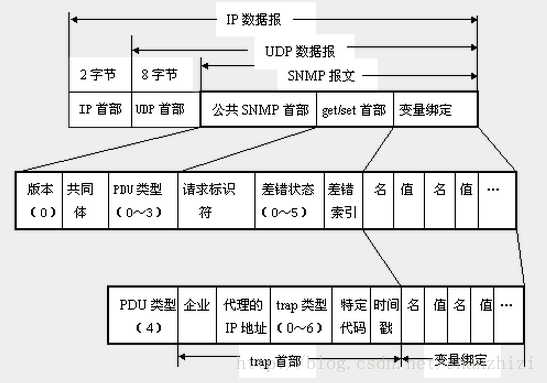
<https://blog.csdn.net/shanzhizi/article/details/11606767>

<https://blog.csdn.net/HandsomeHong/article/details/80016249>

<https://www.cnblogs.com/oloroso/p/4682502.html>



（1）公共SNMP首部

共三个字段：

版本

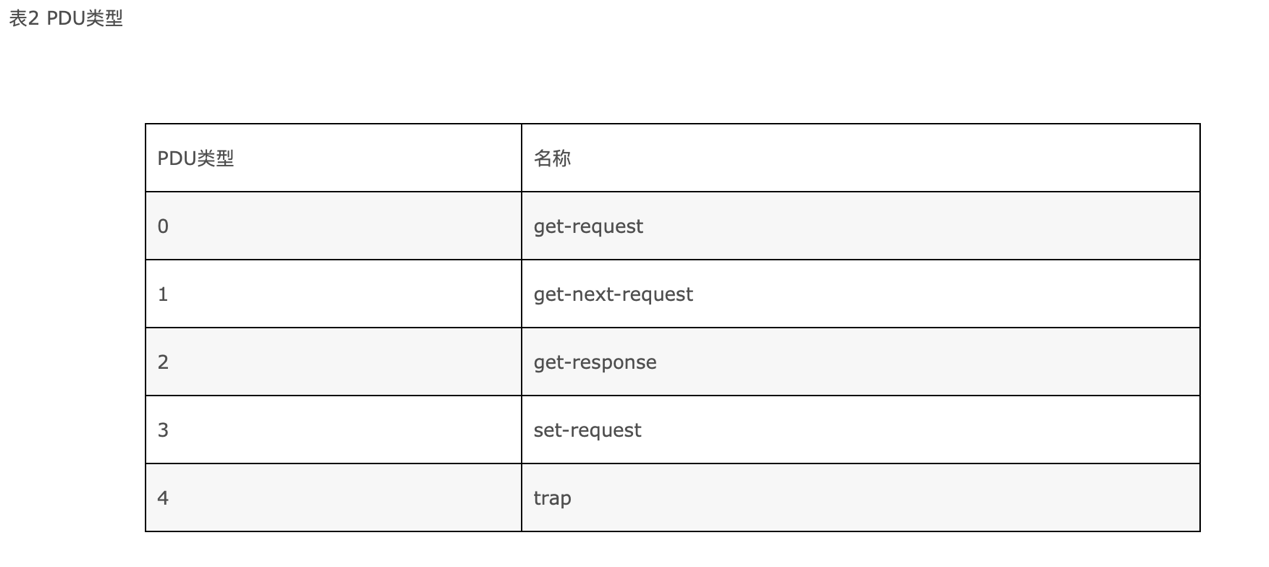
写入版本字段的是版本号减1，对于SNMP（即SNMPV1）则应写入0。

共同体（community）

共同体就是一个字符串，作为管理进程和代理进程之间的明文口令，常用的是6个字符“public”。

PDU类型

根据PDU的类型，填入0～4中的一个数字，其对应关系如表2所示意图。



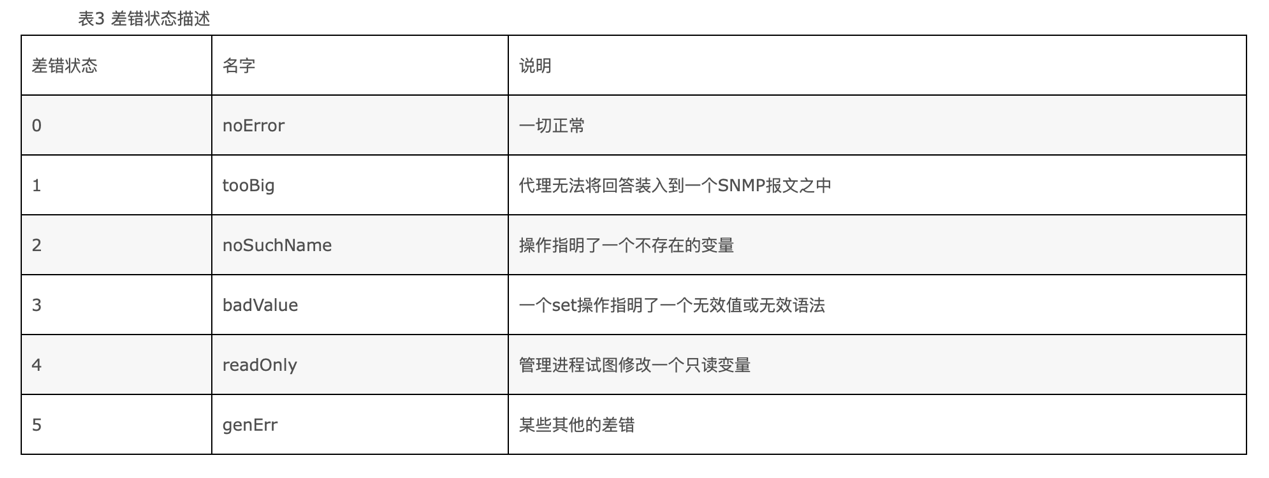
（2）get/set首部

1、请求标识符(request ID)

这是由管理进程设置的一个整数值。代理进程在发送get-response报文时也要返回此请求标识符。管理进程可同时向许多代理发出get报文，这些报文都使用UDP传送，先发送的有可能后到达。设置了请求标识符可使管理进程能够识别返回的响应报文对于哪一个请求报文

2、差错状态（error status）

由代理进程回答时填入0～5中的一个数字，见表3的描述



3、差错索引(error index)

当出现noSuchName、badValue或readOnly的差错时，由代理进程在回答时设置的一个整数，它指明有差错的变量在变量列表中的偏移。

（3）trap首部

1、企业（enterprise）

填入trap报文的网络设备的对象标识符。此对象标识符肯定是在图3的对象命名树上的enterprise结点{1.3.6.1.4.1}下面的一棵子树上。

2、trap类型

此字段正式的名称是generic-trap，共分为表4中的7种。



当使用上述类型2、3、5时，在报文后面变量部分的第一个变量应标识响应的接口。

3、特定代码(specific-code)

指明代理自定义的时间（若trap类型为6），否则为0。

4、时间戳(timestamp)

指明自代理进程初始化到trap报告的事件发生所经历的时间，单位为10ms。例如时间戳为1908表明在代理初始化后1908ms发生了该时间。

（4）变量绑定(variable-bindings)

指明一个或多个变量的名和对应的值。在get或get-next报文中，变量的值应忽略。

00 23 5a 9e 58 b9

00 4c 41 49 50 55

08 00

45 00 00 48 00 00 40 00 40 11 a5 4e c0 a8 0a 01 c0 a8 0a 05

0c 00 00 a2 00 34 ff e0

30 2a

02 01 00

04 06 70 75 62 6c 69 63

a4 1d

06 0a 2b 06 01 04 01 bf 08 03 02 0a

40 04 c0 a8 0a 01

02 01 00

02 01 00

43 01 0e

30 00

目的MAC：00 23 5a 9e 58 b9

源的MAC：00 4c 41 49 50 55

协议类型：08 00 ，为IP数据报

IP头：45 00 00 48 00 00 40 00 40 11 a5 4e c0 a8 0a 01 c0 a8 0a 05 0c

UDP头：0c 00 00 a2 00 34 ff e0

其余部分都为SNMP报文，接下来我们对照前面的报文结构体来逐个分析一下。

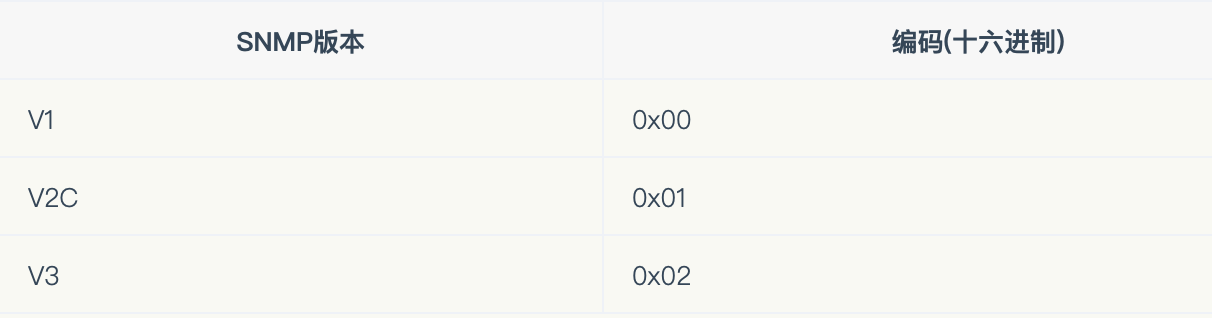
SNMP报文的**首部**指明了这个报文是SNMP协议报文，以及报文的字节数。

30 表示SNMP消息是ASN.1的SEQUENCE类型，第一个字节用于表示这是一个SNMP报文。

2a 表示该SNMP报文的总长度是42(0x2a)个字节，告知后面的SNMP报文的总字节数(不包括前面的0x30和这个长度域所占的字节数)。

**版本**

02 01 00 表示版本号，可见其确实为BER编码方式。02表示该字段是INTEGER类型；01表示该字段占1个字节；00表示版本号，该值为“版本号-1”；



**Community共同体表示**

04 06 70 75 62 6c 69 63 表示团体名，04表示该字段为OCTET STRING类型；06表示该字段占6个字节；70 75 62 6c 69 63 表示团体名的ANSII码的十六进制形式，这里是“public”；

**1、get/set类型PUD表示**

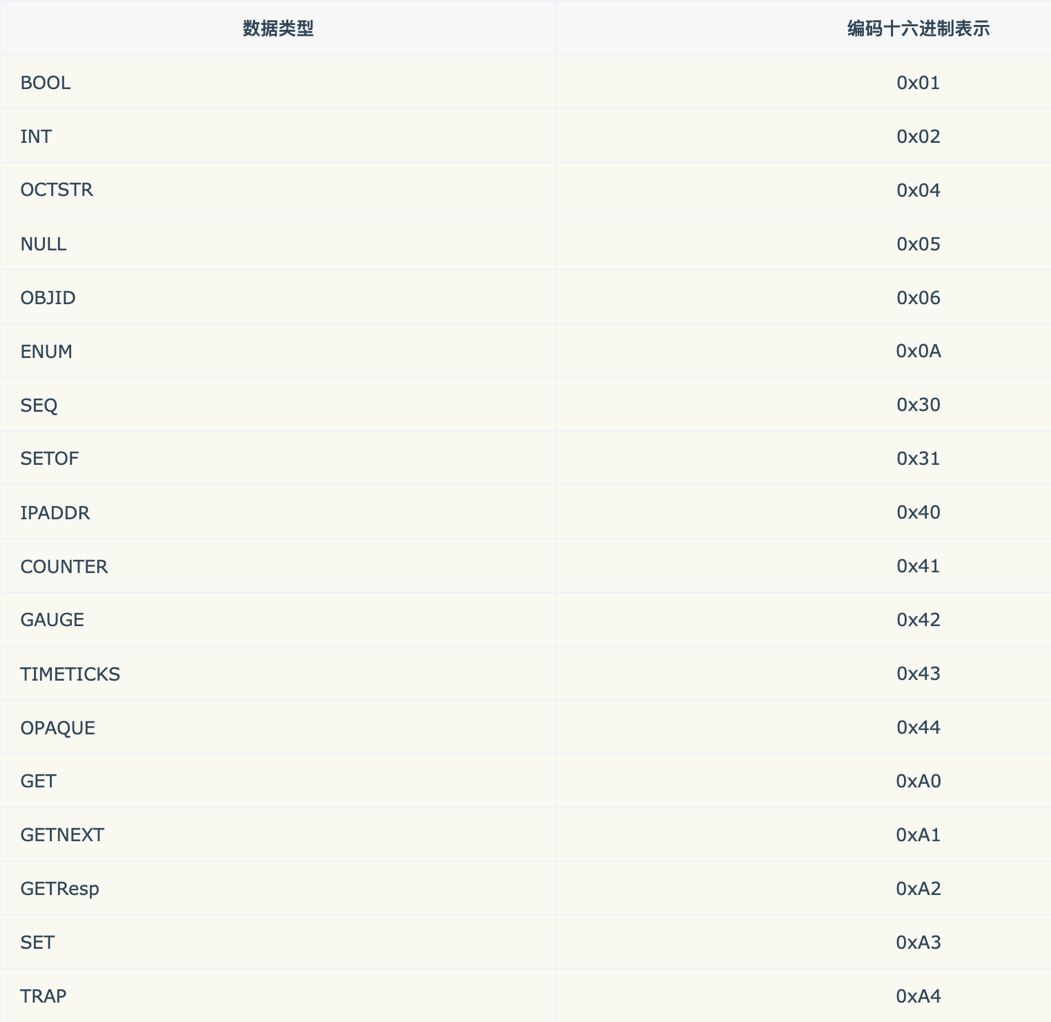
a4 1d 其中a4中的“4”表示这是一个trap报文，a4又叫报文的标签标记，PDU的类型；1d表示后面还有29(0x1d)个字节的数据；



**get/set的表示**

**请求标识符Request ID**

02 04 22 70 8b d4，0x02是标识域，表示值域为整型数据，0x04是长度域，表示值域长度为4个字节，0x22 70 8b d4是值域，就是一个整数而已。(577801172)



**错误状态error-state**

02 01 00，0x02是标识域，表示值域为整型数据，0x01是长度域，表示值域长度为1个字节。0x00是值域，表示没有出错。



**错误索引error-index**

02 01 00，0x02是标识域，表示值域为整型数据，0x01是长度域，表示值域长度为1个字节，0x00是值域，因为没有出错，所以这里为0。

**2、trap类型PDU**

**OID**

06 0a 2b 06 01 04 01 bf 08 03 02 0a，06表示标识符，OBJECT IDENTIFIER；0a表示该字段占10(0x0a)个字节；

关于SNMP的OID的编码方式有些奇特：例如1.3.6.1.2…. 取前两个数字分别记为x和y。编码时40 \* x + y，这里x=1，y=3，因此结果为40\*1+3=43，即表示十六进制的2b。因此，这里的企业OID编码即为1.3.6.1.4.1.8072.3.2.10；2b 06 01 04 01 bf 08 03 02 0a

8072 -> 0x1f 88 -> 011 1111 000 1000(七比特划分) -> 1011 1111 0000 1000(填充最高位, 前面的最高位填1，最后一个最高位填0) -> 0xbf 88

1.3.6.1.2.1.1.1

1、首两个ID被合并为一个字节X \* 40+Y。

例如：1.3合并为1x40+3 = 43 = 0x2B

2、后续的ID，如果在区间[1,127]内，直接编码表示，如果大于127，那么按照下面(3)所述方法编码。

1. 如果ID大于127，那么使用多个字节来表示。

a.这多个字节中除最后一个字节外，前面的字节最高位为1

b.这多个字节的最后一个字节的最高位为0

c.这里每个字节剩下的7个比特位用来表示实际的数值

例如这里的201566这个数，用十六进制表示是0x03 13 5e

那么用二进制表示是000 1100 010 0110 101 1110

注意上面是以7个比特位为单位进行分划的，现在我们来填充最高位

将前面的最高位填1，最后一个最高位填0即可得到

1000 1100 1010 0110 0101 1110用十六进制表示为0x8c a6 5e

**代理的IP地址**

40 04 c0 a8 0a 01，同样40表示该字段为OCTET STRING 类型；04表示IP地址占4个字节；c0 a8 0a 01表示IP地址，为192.168.10.1；

**trap类型，字段正式的名称是generic-tra**

02 01 00 其中00表示trap类型为coldStart；



**特定代码，指明代理自定义的时间**

02 01 00，其中00表示我们指定的trap即specific-trap也为coldStart类型；

**时间戳，指明自代理进程初始化到产生trap报告的事件发生所经历的时间**

43 01 0e，43表示为TimeTicks类型；01表示该字段占1个字节；0e即十进制的14表示时间标签为0.14秒，这里时间计数器以0.01秒递增；

**变量绑定**

30 00 30，表示“键-值”值对的编码类型为SEQUENCE；00表示该字段占0个字节，即没有该字段。

Dt[42]= 0x30;Dt[43]= 0x2a;

Dt[44]= 0x02;Dt[45]= 0x01;Dt[46]= 0x00;

Dt[47]= 0x04;Dt[48]= 0x06;Dt[49]= 0x70;Dt[50]= 0x75;Dt[51]= 0x62;Dt[52]= 0x6c;Dt[53]= 0x69;Dt[54]= 0x63;

Dt[55]= 0xa4;Dt[56]= 0x1d;

Dt[57]= 0x06;Dt[58]= 0x0a;Dt[59]= 0x2b;Dt[60]= 0x06;Dt[61]= 0x01;Dt[62]= 0x04;Dt[63]= 0x01;Dt[64]= 0xbf;Dt[65]= 0x08;Dt[66]= 0x03;Dt[67]= 0x02;Dt[68]= 0x0a;

Dt[69]= 0x40;Dt[70]= 0x04;Dt[71]= 0xc0;Dt[72]= 0xa8;Dt[73]= 0x0a;Dt[74]= 0x01;

Dt[75]= 0x02;Dt[76]= 0x01;Dt[77]= 0x00;

Dt[78]= 0x02;Dt[79]= 0x01;Dt[80]= 0x00;

Dt[81]= 0x43;Dt[82]= 0x01;Dt[83]= 0x0e;

Dt[84]= 0x30;Dt[85]= 0x00;

//---AFDX Payload

Dt[42] = 0x30; Dt[43] = 0x28; /\*snmp, length\*/

Dt[44] = 0x02; Dt[45] = 0x01; Dt[46] = 0x00; /\*version-v1\*/

Dt[47] = 0x04; Dt[48] = 0x06; /\*community, length\*/

Dt[49] = 0x70; Dt[50] = 0x75; Dt[51] = 0x62; Dt[52] = 0x6c; Dt[53] = 0x69; Dt[54] = 0x63; /\*public\*/

Dt[55] = 0xa0; Dt[56] = 0x1b; /\*get-request, length:0x1b\*/

Dt[57] = 0x02; Dt[58] = 0x02; /\*request-id,length\*/

Dt[59] = 0x57; Dt[60] = 0xc3; /\*request-id\*/

Dt[61] = 0x02; Dt[62] = 0x01; Dt[63] = 0x00; /\*error-status\*/

Dt[64] = 0x02; Dt[65] = 0x01; Dt[66] = 0x00; /\*error-index\*/

Dt[67] = 0x30; Dt[68] = 0x0f; /\*variable-binding编码方式, length\*/

Dt[69] = 0x30; Dt[70] = 0x0d; /\*variable-1编码, length\*/

Dt[71] = 0x06; Dt[72] = 0x09; /\*variable类型, length\*/

Dt[73] = 0x2b; Dt[74] = 0x06; Dt[75] = 0x01; Dt[76] = 0x02; Dt[77] = 0x01;

Dt[78] = 0x19;Dt[79] = 0x01; Dt[80] = 0x02; Dt[81] = 0x00; /\*1.3.6.1.2.1.25.1.2.0\*/

Dt[82] = 0x05; Dt[83] = 0x00;

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Dt[42]= 0x30;Dt[43]= 0x28; /\*snmp, length\*/

Dt[44]= 0x02;Dt[45]= 0x01;Dt[46]= 0x00; /\*version-v1\*/

Dt[47]= 0x04;Dt[48]= 0x06; /\*community, length\*/

Dt[49]= 0x70;Dt[50]= 0x75;Dt[51]= 0x62;Dt[52]= 0x6c;Dt[53]= 0x69;Dt[54]= 0x63; /\*public\*/

Dt[55]= 0xa0;Dt[56]= 0x1b; /\*get-request, length:0x1b\*/

Dt[57]= 0x48;Dt[58]= 0x21; /\*request-id\*/

Dt[59]= 0x02;Dt[60]= 0x01;Dt[61]= 0x00; /\*error-status\*/

Dt[62]= 0x02;Dt[63]= 0x01;Dt[64]= 0x00; /\*error-index\*/

Dt[65]= 0x30;Dt[66]= 0x0f; /\*variable-binding编码方式, length\*/

Dt[67]= 0x30;Dt[68]= 0x0d; /\*variable-1编码, length\*/

Dt[69]= 0x06;Dt[70]= 0x09; /\*variable类型, length\*/

Dt[71]= 0x2b;Dt[72]= 0x06;Dt[73]= 0x01;Dt[74]= 0x02;Dt[75]= 0x01;Dt[76]= 0x19;Dt[77]= 0x01;Dt[78]= 0x02;Dt[79]= 0x00; /\*1.3.6.1.2.1.25.1.2.0\*/

Dt[80]= 0x05;Dt[81]= 0x00;