# **[SNMP与MIB](https://www.cnblogs.com/luoxn28/p/5936405.html)**

　　简单网络管理协议（SNMP：Simple Network Management Protocol）是一套网络管理协议，注意，SNMP是一个强大的网络管理协议，而不是"简单"的。利用SNMP，一个管理工作站可以远程管理所有支持这种协议的网络设备，包括监视网络状态、修改网络设备配置、接收网络事件警告等。

　　TCP/IP网络管理由3部分组成：

* 一个管理信息库M I B（ Management Information Base）。管理信息库包含所有代理进程的所有可被查询和修改的参数。
* 关于 M I B的一套公用的结构和表示符号。叫做管理信息结构 S M I（ Structure ofManagement Information）。这个在RFC 1155 [Rose and McCloghrie 1990] 中定义。例如： S M I定义计数器是一个非负整数，它的计数范围是 0~4 294 967 295，当达到最大值时，又从 0开始计数。
* 管理进程和代理进程之间的通信协议，叫做简单网络管理协议 S N M P（ Simple NetworkManagement Protocol）。在RFC 1157 [Case et al. 1990]中定义。 S N M P包括数据报交换的格式等。尽管可以在运输层采用各种各样的协议，但是在 S N M P中，用得最多的协议还是 U D P。

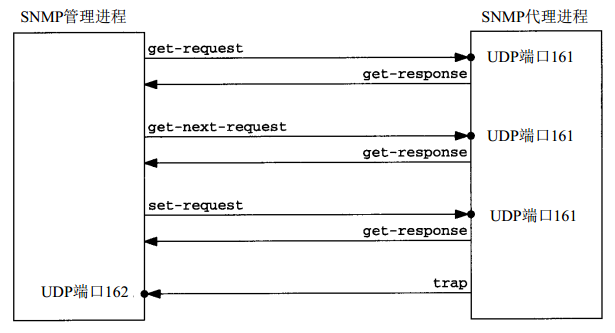
　　SNMP版本到目前为止有3个版本，分别是v1、v2、v3。

## **SNMP协议**

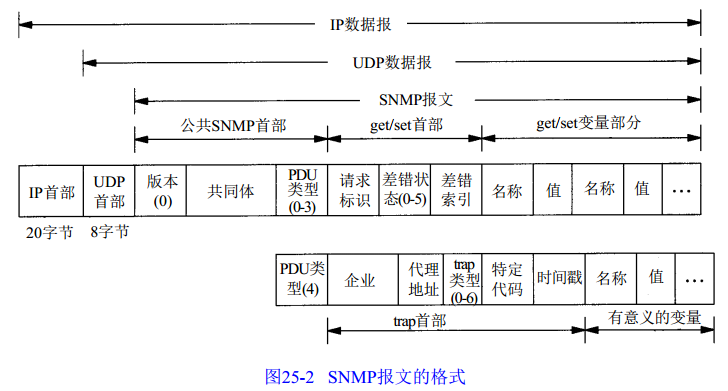
　　SNMP管理进程与代理进程之前为了交互信息，定义了5种报文：

* g e t - r e q u e s t操作：从代理进程处提取一个或多个参数值。
* g e t - n e x t - r e q u e s t操作：从代理进程处提取一个或多个参数的下一个参数值。
* s e t - r e q u e s t操作：设置代理进程的一个或多个参数值。
* g e t - r e s p o n s e操作：返回的一个或多个参数值。这个操作是由代理进程发出的。
* t r a p 操作：代理进程主动发出的报文，通知管理进程有某些事情发生。

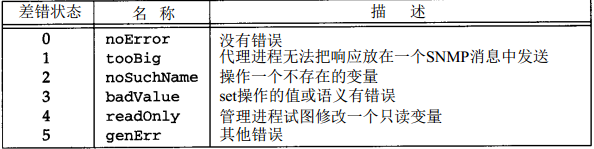
　　以上3个操作是由管理进程向代理进程发出的，后面2个是由代理进程发给管理进程的。管理进程发出的前面 3种操作采用 U D P的1 6 1端口。代理进程发出的 Tr a p操作采用 U D P的1 6 2端口。由于收发采用了不同的端口号，所以一个系统可以同时为管理进程和代理进程。



**SNMP报文格式**

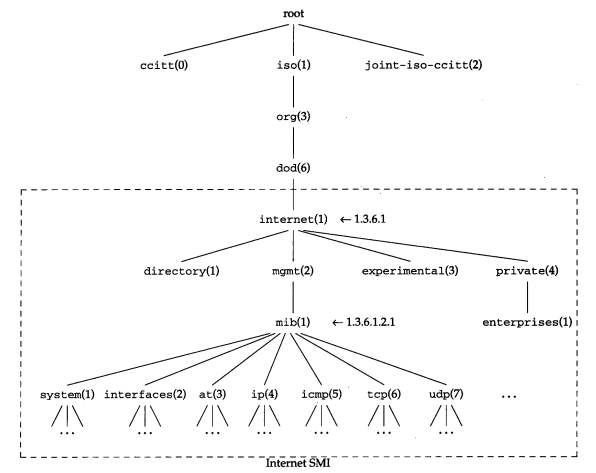


 　　PDU类型的取值范围是0-4，正好表示5种SNMP消息。差错状态字段是一个整数，它是由代理进程标注的，指明有差错发生。



## **MIB-管理信息库**

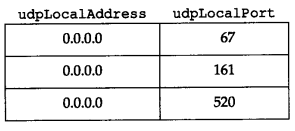
　　所谓管理信息库，或者M I B，就是所有代理进程包含的、并且能够被管理进程进行查询和设置的信息的集合。MIB是基于对象标识树的，对象标识是一个整数序列，中间以"."分割，这些整数构成一个树型结构，类似于 D N S或U n i x的文件系统。



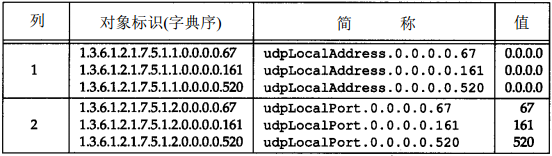
　　M I B被划分为若干个组，如s y s t e m、 i n t e r f a c e s、 a t（地址转换）和i p组等。i s o . o r g . d o d . i n t e r n e t .p r i v a t e . e n t e r p r i s e s（ 1 . 3 . 6 . 1 . 4 . 1）这个标识，是给厂家自定义而预留的，比如华为的为1.3.6.1.4.1.2011，华三的为1.3.6.1.4.1.25506。

　　当对MIB变量进行操作，如查询和设置变量的值时，必须对M I B的每个变量进行标识。注意，只有叶子结点是可操作的，上图中的m i b、 u d p等节点就不是叶子节点。

　　MIB表格都至少有一个索引，对于UDP监听表来说，M I B定义了包含两个变量的联合索引，这两个变量是： u d p L o c a l A d d r e s s，它是一个I P地址； u d p L o c a l P o r t，它是一个整数。下图是一个UDP监听表的索引示例：

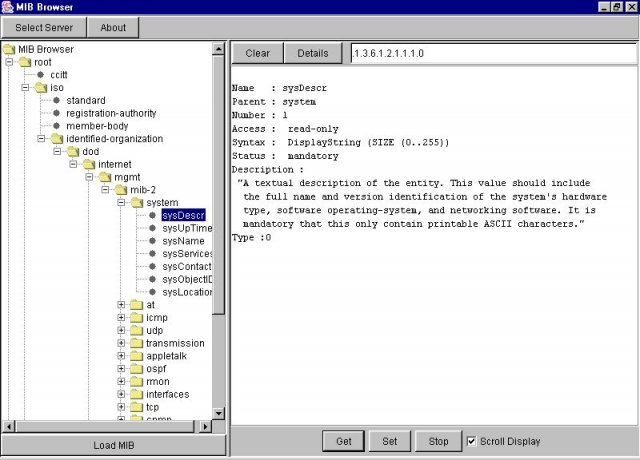


　　M I B表格是根据其对象标识按照字典的顺序进行排序的。如果表格有2个索引的话，首先根据第一索引来进行排序，如果第一索引相同，则根据第二索引来排序。下图是一个UDP监听表的字典序排列：

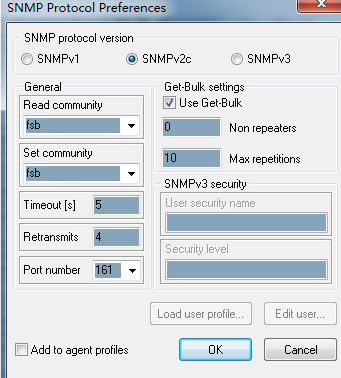


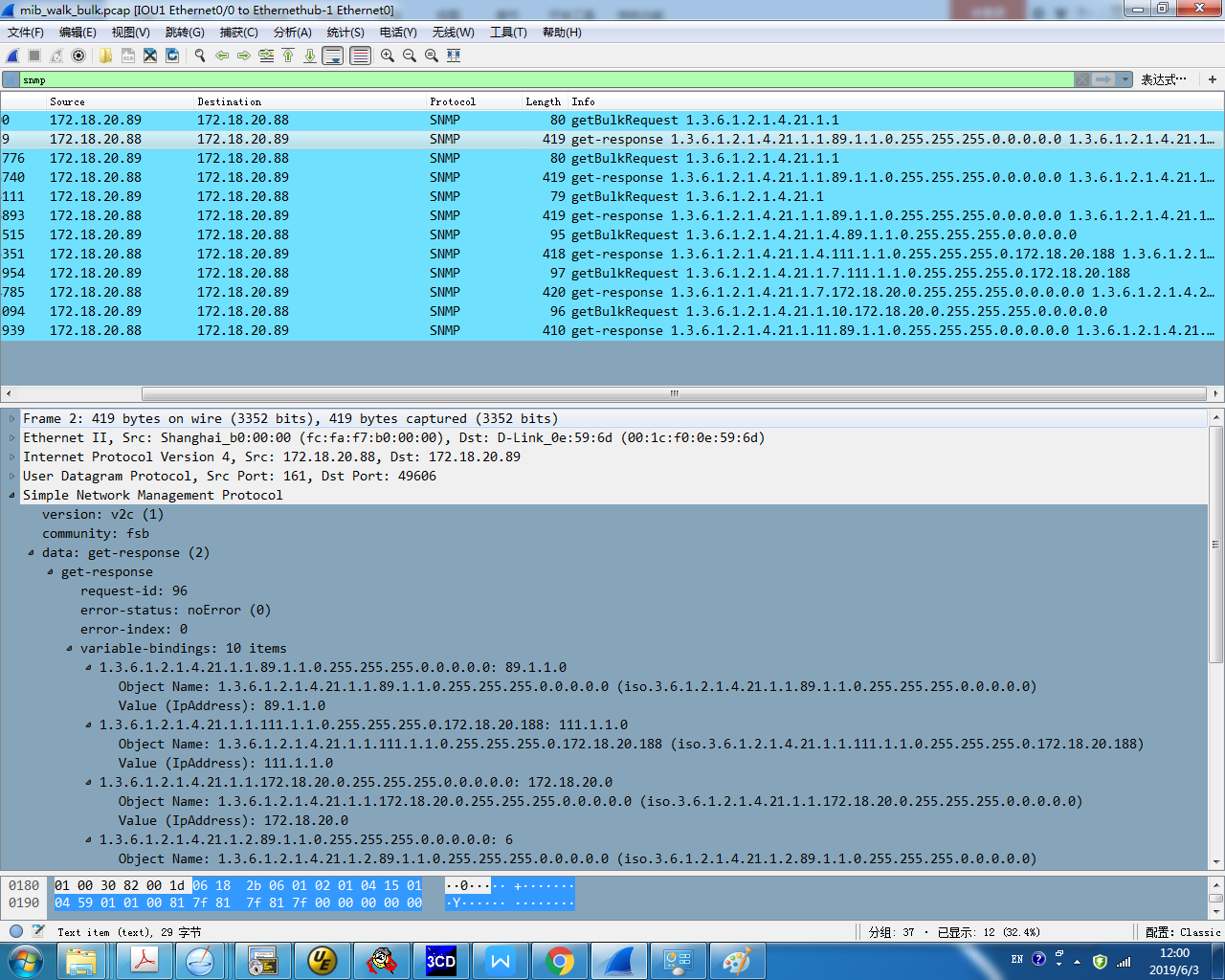
　　在上面的表格中，如果对udpLocalAddress.0.0.0.100进行get操作，则直接返回为空；如果对udpLocalAddress.0.0.0.100进行get next操作，因为udpLocalAddress.0.0.0.100是介于udpLocalAddress.0.0.0.67和udpLocalAddress.0.0.0.161之间，所以应该返回udpLocalAddress.0.0.0.161对应的数据值。

　　在表格中，一个给定变量（在这里指 u d p L o c a l A d d r e s s）的所有实例都在下个变量（这里指 u d p L o c a l P o r t）的所有实例之前显示。这暗示表格的操作顺序是“先列后行”的次序。这是由于对对象标识进行字典式排序所得到的，而不是按照人们的阅读习惯而排列的。



　　用MIB浏览器可以与网络设备进行通信，比如我们以sysDescr的最后一项数据为输入条件进行get next操作，则会返回sysUpTime的第一项数据。

表示一次性返回设置的10个条目，如果一个叶子节点只有3个条目，则会继续读下一个叶子节点，知道凑齐10个条目返回。可以修改这个参数，或者不采用USE getbulk选项。



## **MIB浏览器与设备通信完整流程**

　　SNMP（或者说MIB）开发中，使用MIB浏览器进行测试是必不可少的，它可以直观的显示和配置设备相关的数据。比如在MIB浏览器对UDP监听表进行walk操作（walk操作其实是get next操作，直到遍历完UDP监听表中全部数据为止），第一次进行get next操作，因为没有输入任何数据，所以一级索引和二级索引都为默认值，也就是非法值。然后设置SNMP协议类型为get next，通过网络将数据包发送到设备。

　　设备首先要运行SNMP程序，这里就暂时称为snmpd守护进程，当守护进程接收到数据包时，进行解析，发现是get next操作，并且一级索引和二级索引都是默认值，则表示需要获取UDP监听表中的第一项数据，在守护进程获取第一项数据完毕后，就把该数据封装到SNMP的get response数据包中，通过网络返回给MIB浏览器，这里注意数据的字节序问题。

　　在MIB浏览器中收到返回的数据包时，进行解析并显示出来。然后记录下返回的两个索引值，接着将这两个索引值作为数据数据进行下一次的get next操作，此时一级索引和二级索引都是有效值了。那什么时候结束呢，当守护进程返回END\_OF\_TABLE标志时表示整个UDP表数据都遍历完了。