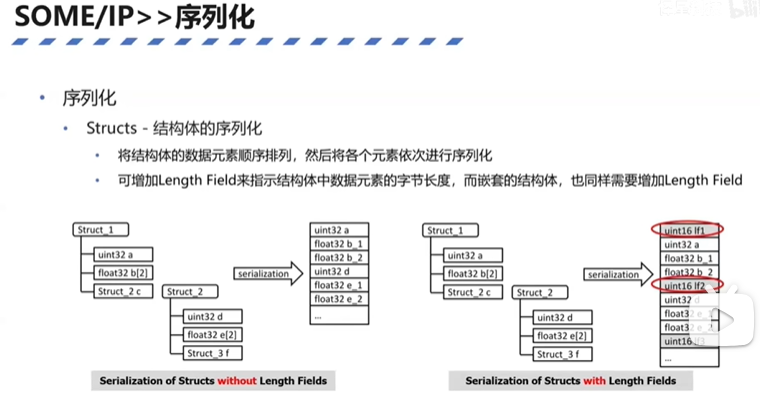
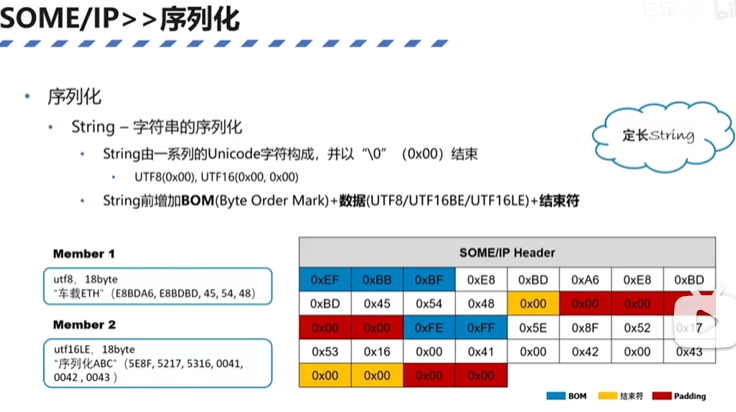
Someip协议介绍

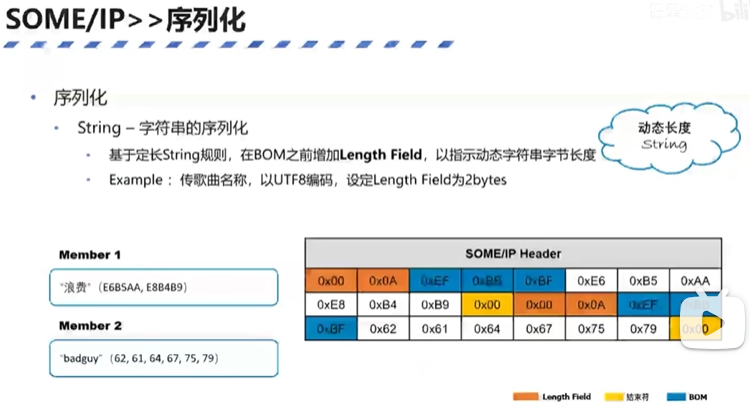


序列化结构体在前面有一个选择，就是可以加上一个length field字段，用来指示结构体的长度。就是我需要在序列化的时候加上这个长度字段，那么我需要在前期定义的时候就定义Length field，Length field本身会占到一定的长度，在Vlan里面有1个字节，2个字节，4个字节，三种情况。那么Length field对应的数据类型就是uint8，uint16，uint32。

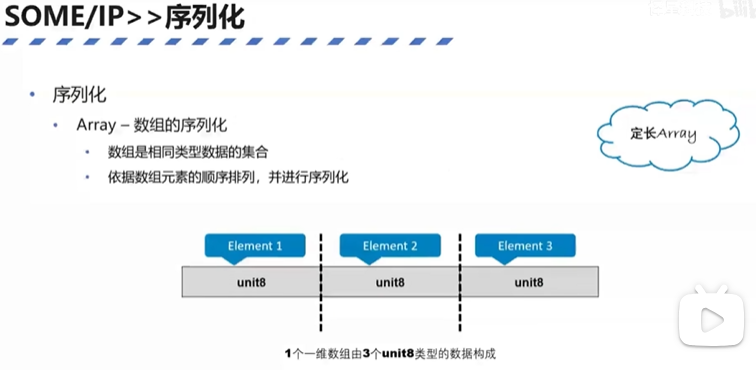


字符串的序列化，字符串的序列化相对而言比较特殊。计算机实际传输的是一些二进制数字，比如实际传输的是1首歌名称，有一些中文字符，或者英文字符，要转化成数字，那么需要一种转化方式。那么Autosar中有2中编码方式，utf-8和utf-16。utf-8和utf-16序列化的前面需要加上bom或者master来指示后面的编码到底是采用那种编码方式。

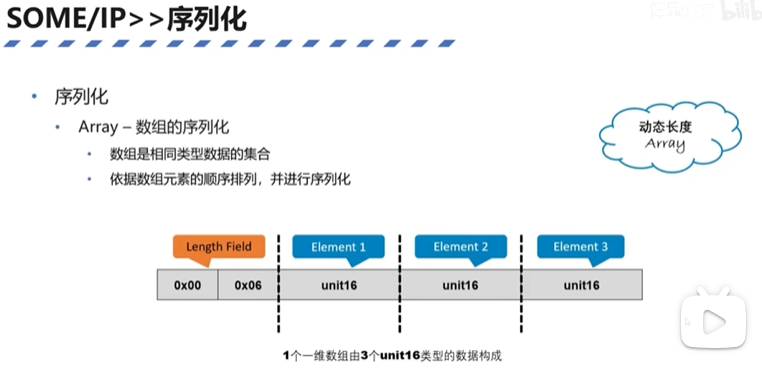
可以看下例子，第一个例子，车载ETH，是通过UTF-8的方式进行编码的，序列化的时候前面需要加上3个字节的Bom，即0xEF，0xBB，0xBF。那么当我反序列化的时候，获取到这3个信息，那么我就要判断后面以怎样的方式进行反序列化。字符串的序列化后需要有一个结束符，结束符就0x00。如果是utf-8的话，就是1个字节，utf16，2个字节。后面的红色代表Pedding，由于现在介绍的字符串是定长的，但是字符本身并没有把空间占用完。比如上述定义的18字节的字符串，如果没有占完，那么红色部分就做一些Pedding，即填充。



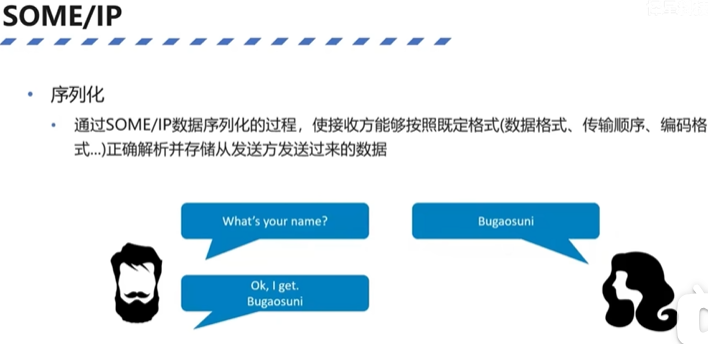
实际使用过程中，不一定所有的字节都是18个字节，那么动态的情况，比如一首个的名称可以不同，比如专辑的名称不同，那么需要定义一个数据类型，需要数据类型是可变的。不同于前面的定长的，需要在序列化的数据之前，加上一个Length field，就是Length field是前面长度的一个指示。在实际Length field里面，我会指示我实际传输的过程中，字符串传送的长度。这即是变长序列化的规则。



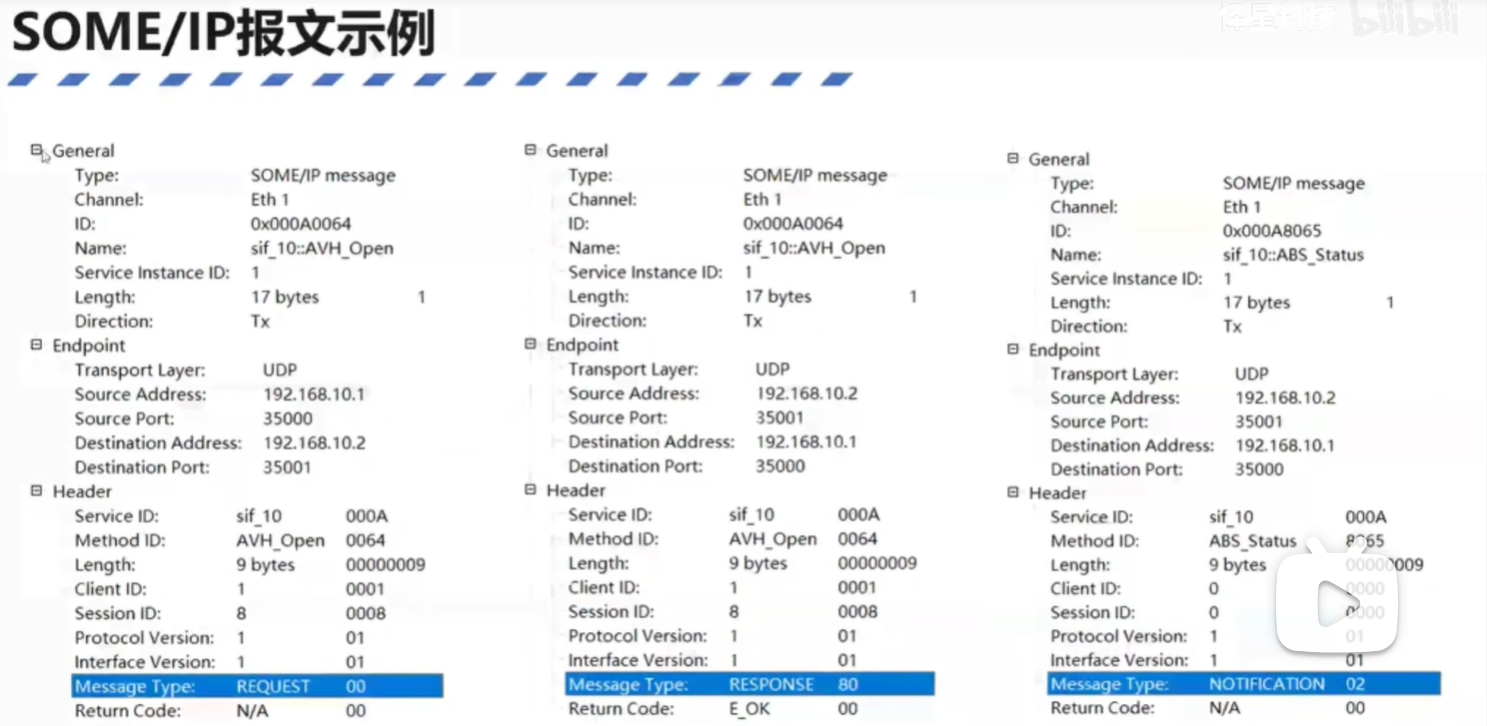
数组的序列化，数据是相同数据类型的集合，定长数组直接按顺序进行排布。



变长的情况，类似与前面的字符串，在前面加上一个length field，就是这样一个字段，来指示实际数组的长度。里面的一个数值代表一个单位长度，单位是一个字节，而不是一个数据元素。比如说3个元素，每一个元素是uint16，1个uint16是2个字节，那么长度字段里面需要写一个6，说明实际的数组占6个字节。







第一个是Request报文，前面是一个Header，代表一个bottom。通过解析，可以知道，报文的Service ID、Method ID。这些数值，比如AVH\_Open,是在以太网通信数据库Arxml里文件名称的描述，所以说可以描述出来。那么这个Method ID的名称、数值以及长度（从client ID开始大概占多长），以及Session ID、Protocol Version、Interface Version、Message Type、Return Code都可以看到。

Response报文，前面的这些结构都是一样的，那么区别在于，在Message Type这个地方，和Request报文不同，数值是80，Return Code代表这个响应是成功的，这里Payload没有展示出来。

第三个就是事件，事件的一个通知，这里的Message ID指示的是Event ID，最高位是1，转换位16进制是80。Message Type是NOTIFICATION，数值是02，代表报文是事件通知的报文。