1. 服务器之间内部rpc协议，1个消息头+ msgNum个消息体

1.1消息头

+---------------------+

| 32 Bit |

+---------------------+

|<= length of totle =>|

1.2消息体

+-----------+-----------+-----------+------------+--------------+---------------+

| 4 byte | 2 byte | 2 byte | 2byte | 4 byte | 4byte |

+-----------+-----------+-----------+------------+--------------+---------------+

|<= msgno =>|<=timeOut=>|<= local =>|<= remote= >|<= transaction =>|<== queryID ==>|

+------------+--------------------+--------------+---------------+

| 4byte | 3 bytes | 4byte | N bytes |

+------------+--------------------+--------------+---------------+

|<= userId= >|<= length of data =>|<== cmd ==>|<=== data ===>|

* 1. 头部由消息长度，最大允许2的31次方-1长度(31bit)未压缩消息发送。
  2. Msgno为自动递增uint32数值，由网关获取生成。其他服务需要发送消息时，这个值为0，经过网关时，网关获取并计算msgno的值，自动递增
  3. timeOut 当前消息的最大处理时间，超时host将会删除Msgno停止消息转发
  4. local由1byte服务序号+1byte服务ID号，记录的是请求发起方的信息。
  5. remote由1byte服务序号+1byte服务ID号，当ID号为0的时候则进入均衡负载由网关自动分配
  6. transaction事务号，由网关中心管理事务号和对事务进行提交与回滚
  7. cmd定义，在protocol里面编写go格式的struct结构体，由1byte服务序号+由结构体名字进行crc32生成4byte数值取前三位byte，格式为int32
  8. queryID是单一服务器生成的自增ID，用以匹配请求和回复消息
  9. userID 记录当前操作者的uid
  10. 结构体命名规则 MSG\_服务名\_方法名
  11. 按照结构体从上到下的顺序，进行序列化到2进制data区域
  12. int与uint，8bit到32bit，使用小端littleEndian编码，协议头的length使用uint32
  13. int,uint,int64,uint64使用varints编码
  14. string，将string强转为[]byte后， len()进行int编码+[]byte消息体
  15. []byte，规则如string
  16. bool，0=false或1=true
  17. []int与[]uint系列，长度len()进行int编码+len()个规则c编码，如[]int16{1,2}，将会编码为|02|01 00|02 00|，如[]int32{1,2},将会编码为|02|01 00 00 00|02 00 00 00
  18. 支持protocol内任意其他结构体指针，不支持非指针形式，由1byte+0或nbyte组成，当结构体为nil，第一个byte为0，不添加其他数据，当结构体不为nil，第一个byte为1，后续为结构体成员的编码。
  19. 支持[]protocol内任意其他结构体指针，编码格式为长度len()进行int编码+数个上述h格式编码，如[]\*A{nil,nil}，则为|02|00|00|
  20. Float因精度问题不予支持，转为string进行传递
  21. 当消息体所有的数据完成编码后，获取缓冲区长度，允许最大长度为16777216最大值(压缩前)，长度以3byte写到length of data区，后面紧跟data数据（未压缩）