# IOcopy

建议实现读取定长数据的世欧, 使用make定长slice + io.ReadFull的方式来代替 bytes.Buferr + io.CopyN, bytes.Buffer的初始长度很屌, 每次扩容长度\*2, 都需要拷贝数据, 大包就会影响性能, 发现包大了性能相对于其他框架性能下降, 就是因为io copy的多了.

测试pbrpc server性能的时候发现，如果消息大小比较大的场景下(>1MB)，性能会有明显的退化，跟C++版的brpc有比较大的差距，看了一下代码，gdp里面有些不必要的copy操作，改掉之后性能就能有所优化，例如packing/reader.go里的DiscardMessage，可以用io.ReadFull来代替两次copy操作

