1. 子类协议对象的设计,重点在于Run函数(即handle函数)。需要测试节点之间能否通过此协议完成handshake以及各类消息的收发以及对peer对象的更新(需测试peer.go中的所有方法能否正常通信)
2. 测试某节点产生交易后，能否成功广播并被其他节点接收(测试pm.txBroadcastLoop())
3. 测试某节点挖到区块后，能否成功广播并被其他节点接收(测试pm.minedBroadcastLoop())
4. 测试在区块同步之前，节点的交易池需要将存储的所有交易信息发送给对端peer节点，即需要在handle函数中调用pm.syncTransactions(p)进行绑定，同时调用pm.txsyncLoop()完成交易消息的发送，对端peer在收到TxMsg之后需要将其中的交易添加到本地。此过程需要进行完整的测试(测试交易池能否得到同步)
5. 测试不同节点之间能否通过运行pm.syncer()完成区块同步(优先选择eth61同步策略)

注：

1. 测试时可以通过self.eventMux.Post(TxPreEvent{tx})产生一个交易事件
2. 可以通过self.eventMux.Post(NewMinedBlockEvent{block})产生一个挖矿事件
3. 对于新加入的没有创世区块的peer节点，从所有连接peer中选择一个具有最大td值的节点，拷贝其整个链(此方案效率比较高)
4. 对于新加入的没有创世区块的peer节点，也可以先只获取一个创世区块(对于区块同步来说,至少需要有一个创世区块)，后续也可以再与其他peer节点进行同步获取(更加安全，但是比较慢)

外部的接口与函数：

1. Handler.go:
   1. chainman \*core.ChainManager相关的一系列方法
   2. txSub event.Subscription相关方法
   3. minedBlockSub event.Subscription相关方法
2. protocal.go
   1. type txPool interface
   2. type chainManager interface