kademlia发现协议

kademlia是p2p的一种节点发现协议，其核心是通过计算节点之间的逻辑距离来发现附近节点以实现节点查找的收敛。

**简化协议**

这里我们为了说明原理尽可能的简化协议。我们只实现三种请求:

1. 节点握手

2. 获取区块数据

3. 交易广播

为了方便，其中又将节点握手作为心跳发送，并根据心跳信息进行区块同步

网络协议方面，借鉴以太坊的做法，UDP做协议发现，TCP做数据传输。每当发现一个节点，就通过TCP建立连接，并发送心跳数据，以保证数据的一致性。

**消息**

定义消息类，分别定义了无意义回应和以上三种请求。为了方便处理，这里统一使用字符串而不是二进制数据进行数据传输。

TCP端比较简单，listen\_loop方法监听新的请求并开启一个新线程处理连接中的数据交互。handle\_loop方法调用了handle分发处理请求。

handle\_handshake处理握手请求，这里将最新块高度和创世块发送出去了，方便和本地数据进行比较，如果远程数据更新，那么就获取新的部分的区块。

handle\_get\_block获取对应的区块并将数据发送给客户端。

handle\_transaction 处理客户端发送来的交易信息。把客户端发送来的交易添加到未确认交易池，如果交易池满了就添加到区块。这里是方便处理才这么做的，实际上，比特币中并不是这样做的，而是由矿工根据情况进行打包区块的。

`handle\_transaction`处理服务器发送来的交易,将交易添加到交易池，如果交易池满了就添加到区块链中。

`handle\_get\_block`处理服务器发送来的区块，并将区块更新到链上。

`handle\_shake`处理服务器响应的握手信息，如果发现当前的的区块高度低于数据中响应的区块高高度，则发起请求获取新的几个区块。

`shake\_loop` 每间隔10秒发送一次握手信息（5秒同步一次区块),如果发现有需要广播的交易则进行交易的广播。

**P2P服务器**

p2p节点发现部分，使用了kademlia协议，并使用了kademlia库

其中`run`方法启动节点监听并连接一个初始节点，并运行p2p节点监听。

`get\_nodes`方法获取当前所有的节点。

**## 连接节点**

**`nodes\_find`为节点发现方法，每隔1秒进行查找当前是否有新的节点，并开启线程进行连接。**

**`broadcast\_tx`为广播交易的方法，将交易添加到待广播交易池。**