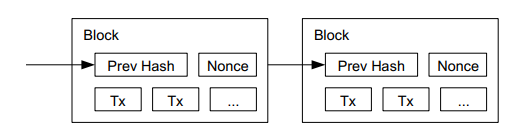
4. Proof-of-Work 工作量证明

为了在点对点的基础上实现一个分布式时间戳服务器，我们需要使用PoW（Proof of Work）系统来达成共识。PoW过程就是寻找一个目标值的过程，当对该目标值使用SHA-256之类Hash算法进行Hash运算时，要求得到的哈希值是以一定数目的0位开始的数。随着要求0位数目的增加，PoW过程的工作量是指数级增长的。通过对目标值进行一次哈希运算就可以验证目标是是否符合要求。

在PoW的实现中，我们在block中引入一个Nounce变量，我们通过Nounce的不断增大，来找到一个特定的Nounce值，使得整个block的哈希值以特定数目的0位开始。一旦CPU通过运算满足了PoW的要求，该block就不可以再被更改了，除非我们对该block及其后面的block都重新进行PoW过程。

|  |
| --- |
| // 要求hash值以5个0位开始  // 假设hash值都是8bit  func PoW (Nounce uint32) {  for **SHA256**.**Hash**(**prevHash** | **Nounce** | **Tx**) > 0b00000111 {  **Nounce**++  } |
| } |



PoW还解决了用什么方式表示大多数的问题。如果是基于每个IP一票的原则来获取大多数，这种方式很容易通过一个节点分配多个IP的方式来作弊。而PoW实际上是基于每个CPU一票的原则，大多数决策就用最长的链来表示，最长的链意味着耗费了最多的算力来进行PoW。如果大多数的CPU算力都被诚实的节点所掌握，那么诚实的链就是增长的最快的那个。如果攻击者想要更改某个区块，那么需要修改这个区块及其后续所有区块的内容，也就是需要对该区块及其后的所有区块都再次进行PoW过程，这超过了诚实节点的工作量。随着某个区块后面的区块链不断增长，改动该区块的可能性也就越低。

硬件计算能力是不断提高的，而且参与到网络中的节点也是越来越多（即贡献的算力越来越多），为了达到平衡，PoW的难度会相应的提高，我们通过控制平均每小时产生区块的数量来实现这种平衡。当区块产生的速度过快时，就通过提高难度来降低其速度。

5. Network 网络

区块链网络是按照如下的步骤运作的：

1. 生成新的交易并将交易广播给所有节点
2. 每个节点都收集这些交易，并将其打包成区块。
3. 每个节点都进行PoW过程，致力于找到满足要求的Nounce
4. 当一个节点完成PoW过程时，就将该区块广播到全网。
5. 只要区块中所有的交易都合法有效，节点就会接收该区块
6. 节点如果接收区块，就会开始创建下一个区块，用已接收区块的Hash值作为新区块的prevHash，进行新的一轮PoW的过程。

节点总是认为最长的链是正确的链，并致力于在最长的链上进行扩展。如果两个节点同时完成PoW过程并广播了不一样的区块，有的节点收到其中一个，而有的节点先会收到另一个。这种情况下，首先收到的区块会被节点作为下一个区块添加到链上，同时也会保存后到的区块。当下一个区块来临时，就会知道哪一个区块是比较长的，然后切换到较长的区块链上。

新交易的广播不一定要到达每一个节点，只要到达足够多的节点，这些节点就会在一定时间内将交易打包到区块中。区块的广播也是允许丢包的，如果节点没有收到某一个区块，当下一个区块到来时，该节点发现新区块的prevHash不是目前的最后一个区块，就会去请求丢失的区块。