

一种基于“自我中心主义”的共识机制

2021-10-29

简介

“自我中心主义”的含义是，对于每个人来说，世界的大小取决于他能够接触到多大的世界，世界很大，和我无关，世界很小，全和我有关。一个人认识的人、了解的事、接收到的信息，无论是否命中注定，一定是有限的，你不可能认识世界上所有的人，知道世界上所有的事。才学渊博、见多识广，又怎样呢。

在区块链中，共识机制是用来保证数据一致性的关键手段，也给区块链带来了最核心的去中心化的特点。共识机制是强一致性的，或者是在拥有一定容错能力的情况下，达到大多数一致的效果。有没有一种可能，存在一种共识机制，不以数据一致为首要目标呢？

世界本就是复杂的，试图将所有节点同步到仅仅一种数据状态，其实是违反直觉的。而且，无论是不需要授权的大范围共识，还是基于身份授权的小范围共识，最终实现数据一致的方法，都是“多点变单点”，也就是同一时间只有一个节点在处理数据，其他节点可能是在共识后接受满足条件的数据，也可能在确认数据前投票是否同意对数据的操作，总之都需要有一个“英雄”一样的节点，在关键的变更数据的时刻，做一些事情。

有的英雄实力强大，先斩后奏，改过数据后过来跟你说，“我改了数据”，你一开始不满意，但是接触后发现英雄确实厉害，能做出你没有解决的难题，于是你就认可了英雄的行动。

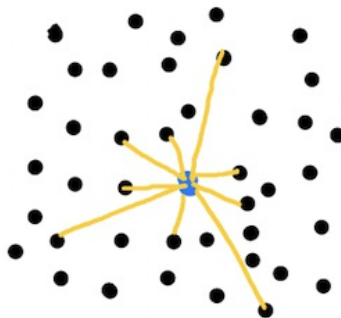
还有的英雄被公众授予权力，行动之前作为代表被选举出来，行动的时候会万分小心，挨个问民众，“改动这里的数据，你同意吗？”如果大多数人同意，英雄就会行动。

当然，每个人都有平等地享有做英雄的机会，虽然有的人天生神力，有的人八面玲珑，但机会总是有的，题放在那里，你做不出来，怪谁？每个人就可以被选举，别人不选你，怪谁？

所以，为什么我们不能做自己的英雄？为什么我们要屈就于别人的光环之下？每个人都是自己的英雄，在我们的世界里，在大小受限于个人接触范围的世界里。一种共识机制，节点的边界受限于其触及的网络规模。

网络概况

在非结构化的点对点网络中，路由表是必不可少的组成部分，节点能够接触到的网络大小，就取决于路由表中存着多少“联系方式”。共识对数据的处理，就以路由表中的节点为依据，路由表中有 10 个节点，就争取和这 10 个节点达成一致，路由表中有 10,000 个节点，就和 10,000 个节点达成一致。也没有必要使用分布式路由表，就普通的数组就可以。在这种情况下，网络中的节点会是这种样子：

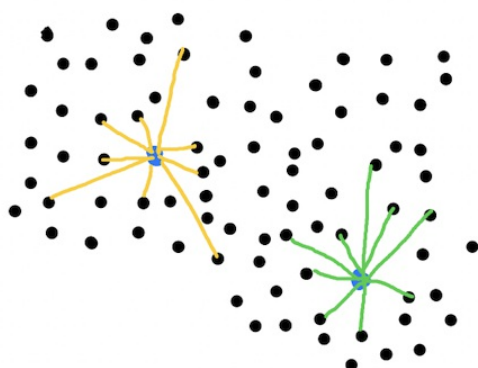


以当前节点为中心，连接到的节点数量可多可少，有的很远，有的很近。弱水三千，只取一瓢。在路由发现的问题上，节点也是需要种子地址的，比如节点启动的时候先解析种子地址的记录：

```
lookup("seek.domain")
-> 127.0.0.1
-> 127.0.0.2
```

然后依次请求解析出来的节点地址，去得到他们路由表中的内容，将其添加到自己的路由表。这其实是常规做法，不过这样有可能引起的后果是，节点会瞬间获取到整个网络的路由信息。这并非不好，只是感觉有点快了，我们认识一个人是需要时间的，和人交谈也是需要时间的，你无法同时和三个人交谈，或者无法同时听三个人说话。即使拿到了很多人的联系方式，也没办法“多线程”联系每个人。我们处理信息的“带宽”有限，节点也一样。我们甚至可以对网络发现的速度稍微做一点限制，比如串行处理路由表新增记录的动作，先与节点建立连接然后再添加信息。

让路由发现慢一点，似乎听起来不太正常，难道是想让网络处于不同步的状态吗？很多共识的瓶颈就在网络带宽上，就在协议交互的复杂上。如果降低节点间的交流成本，共识的容错能力也会随之降低。网络可能会被划分为不同的区域，可能形成大大小小的圈子。

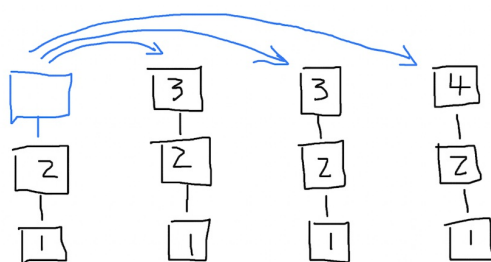


对于共识算法来说，脑裂是要尽可能避免的问题，但其实网络分割是再正常不过的事情，是自然存在的情况。我们人类的思想是分裂的，有可能是对立的，但是经过一些事件后又可能达成一致。所以在一个网络中出现割裂是完全允许的情形，形成的小规模网络可能是互联互通开放的，也可能是保守封闭与外界隔离的。重点是要有一种机制能够“修正”这种分裂，也就是在某种条件下，割裂的两个网络可以相互合并。

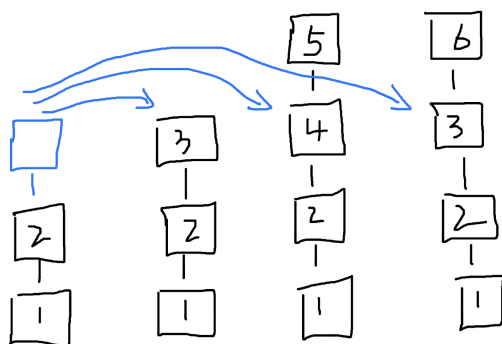
同步数据

主动

这里保留区块链创世块的概念，所有节点的第一个块内容是相同的。新加入网络的节点，会从创世块开始启动，此时其他节点的块高度已经有很多了。比如当前节点的块高度是2，想要从网络中同步第3个块高度的内容，节点的路由表中有其他节点的地址，其他节点块高度均等于3。当前节点会发起一个对块高度3的请求，依次到其他节点。

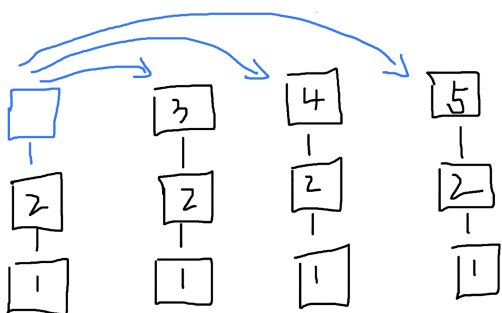


请求过后，发现有 2 个节点块高度为 3 的块一样，块的内容一样、块哈希一样、前块哈希也一样，那当前节点就把这个多数节点都存在的块作为第 3 个块。



如果请求的时候发现有的节点的块高度已经大于 3 了，那么是不是应该块高度最高的优先呢？如果考虑时间尺度的话，就会觉得事物发展是需要遵次序的，你不能跳过 3 岁直接过起 4 岁的人生，区块链的数据也应该有先后次序。况且，当前节点需要的块高度是 3，请求的块高度是 3，管你有没有其他高度的块？你的块高度再高，我就要 3 的，你说你多高有什么用？

那么如果请求过所有节点，发现每个节点的块都不一样呢？该信谁？

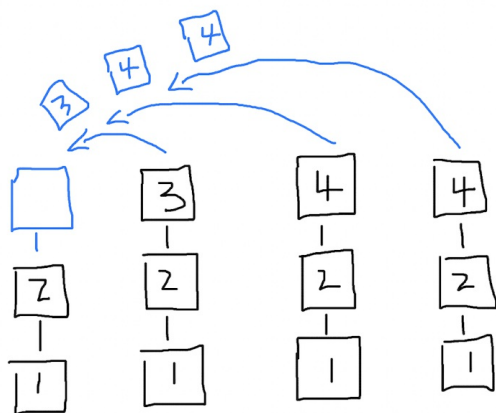


总得挑一个吧。如果不能确定哪个节点或者哪个内容可信的话，就随机选吧。最好是选择最后一个请求的节点，因为错过的节点就已经错过了，此时最后一个节点是距离你最近的节点，并且在此之前你并不能判断，是否存在块内容相同的节点。所以在放弃之前的节点后，最后一个节点就是你不可以放手的选择。

对于主动请求块数据的情况，很关键的地方是，请求一定是按照路由表顺序依次进行的，在得到第 1 个节点的响应之前，绝不向第 2 个节点发起请求，做人不能太三心二意了。如果有节点就刻意加速、同时请求多组数据呢？其实也无伤大雅，毕竟只是同步数据，有的人喜欢快点，有的人喜欢慢点，有的人喜欢快生活，有的人享受慢生活。

被动

除了主动请求某一高度的块数据，节点也会收到其他节点的广播消息，比如当前节点的块高度是 2，收到了来自其他节点的内容分别是 3、4、4 的块。



按大多数一致的原则，是不是应该选择内容是4的块放在自己的第3个块高度上？但是这样存在一个问题是，你无法预测自己会收到多少个块，没办法计算块内容的总量和占比。所以对于被动接收的块，可以以第一个收到的块内容为准。

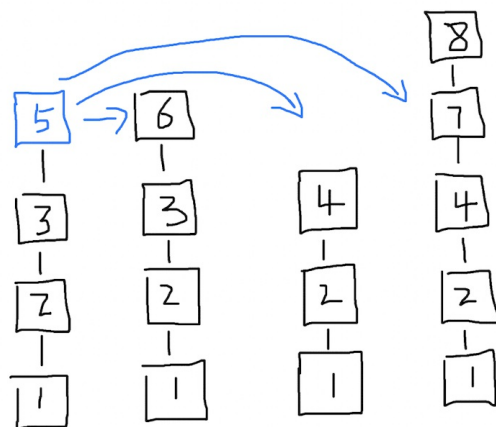
人生的出场顺序很重要，如果正好需要的块高度是3，接收到广播的块高度也是3，那就它吧，遇到哪个算哪个。实在不行后面遇到更合适的再换。如果有节点很激进，为了自己的块能够被大范围接受，把同一个块标记为从1到很大块高度广播出去，就为了碰运气，让正好缺块的节点接收，那也就随他吧。因为节点在主动同步块的时候，是按照高度获取内容的，这种激进一点的做法并不能带来很好的收益。

新增数据

网络中的数据由谁产生？为了解决这个问题，可以先定义为，每个节点都可以产生数据。一种极端的情况是，每个人都只相信自己的数据，各玩各的，整个网络就变成单机版了。所以节点也需要将自己产生的数据散播出去，发送给其他节点。对于其他节点来说，就是“被动同步数据”的情况了。

有节点需要

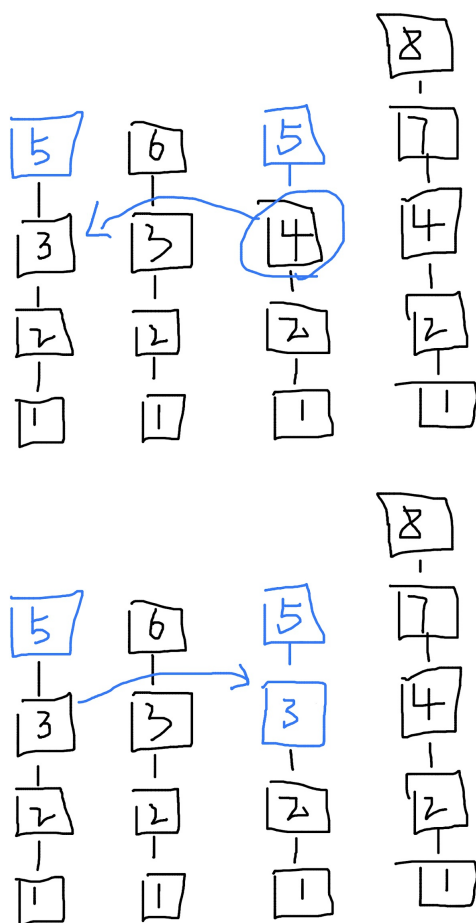
主动广播块数据分两种情况，一种是有节点正好需要块，你正好发送给他了。



当前节点接收到内容的请求，新增了块高度为4的块，这时会直接把块持久化到主链上。接着开始对块高度为4的块进行广播，广播按照路由表依次进行，在广播结束之前，当前节点不会打包下一个块。广播开始后，有节点块高度为3，说明你是第一个发送给他块高度4的节点，它一定

会接收你的块，同时给你一个响应消息。在收到响应后你就可以知道，当前网络至少有一个节点接收了你的块，你可以继续处理下一个块了。当然，在路由表遍历结束之前，节点即使收到响应消息也不会停止这一轮广播，这是理所当然的，希望有更多节点可以接收块内容。

如果对方节点在收到块后，发现块内容是5，前块哈希是3，并不对应它自己的前块哈希4，对方依然会接收这个块，并且依次替换自己之前的块，直到哈希一致。

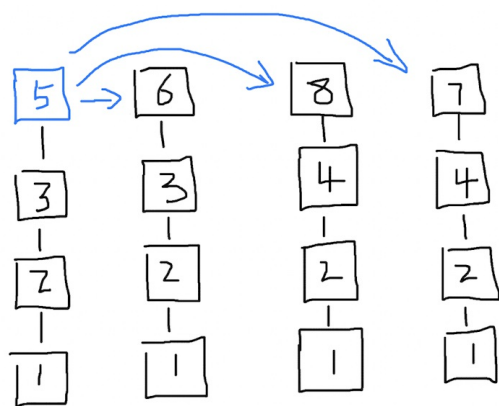


这种机制有可能带来的风险是，接收到一个块，然后把整条链都替换掉了，这是非常严重的不能接受的开销。但确实存在这样的可能，你遇到了一个坏人，这个坏人乘虚而入，他的思想颠覆了你的的人生观，让你误入歧途，六亲不认。倒是你需要反思一下，你的路由表里为什么会有这样的坏人。而且这样的坏人多吗？如果一个恶意节点用一个块替换了你的整条链，但是接下来会有很大概率有好人来把你的整条链置换到大多数一致的情况。

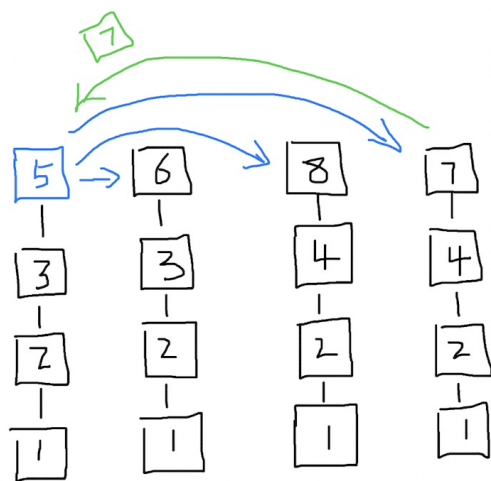
这里暴露出了一点问题，接收到第一个块就认可，是不是太草率了？如果是坏人怎么办？为了增加节点作恶的难度，在接收到块内容为5发现前块哈希对不上的时候，应该不止向第一条链请求块内容，而是走完整的“主动同步数据”的逻辑，根据块高度把路由表里的所有节点都请求一遍。如果块内容为5的块，前块哈希和大多数节点不一致，就直接把5抛弃掉。如果一致，就说明它不是有害内容。

没有节点需要

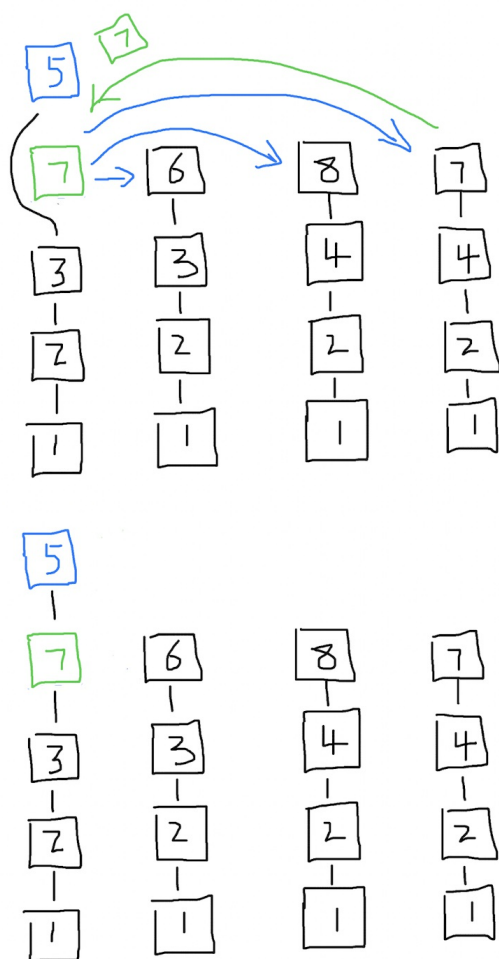
节点新增块后，可能遇到没有节点需要当前块高度的情况，



其他节点的块高度都大于等于广播出去的块。这种时候，当前节点就有必要做一点点妥协，为了让别人接收自己，为了其他节点接收自己的块数据，只好先从其他节点同步数据，和其他节点保持一致。

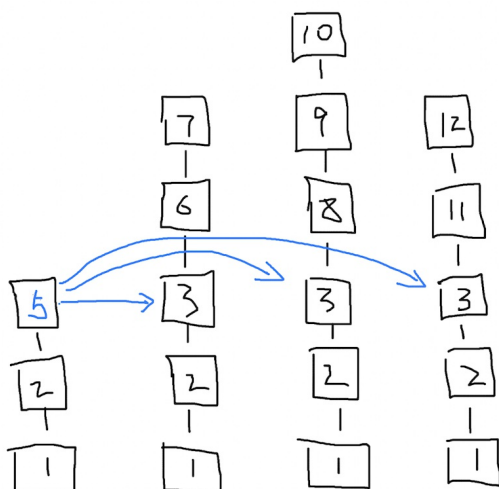


在块高度为4的块上，当前节点广播了一圈发现没有节点愿意接收这个块，那当前节点就把最后一个访问的节点的，当前块高度的块，请求过来。为什么是最后一个请求的节点？这里也可以走一遍完整的“主动同步数据”的流程，但为了提高效率，减少网络交互，可以先随意接收一个块内容，再做后续的判断。选择最后一个节点，是因为离得近。在错过了万丈红尘纷纷扰扰之后，恍然回首，发现最后一个节点是你此时最亲近的伙伴。



收到最后一个节点的块内容是7的块后，当前节点继续广播块内容为5的块。

如果不幸遇到了其他节点的块高度都远高于自己的情况，那说明自己确实落后了，先把其他节点的内容都同步过来再说。想要创新，想要新增内容，至少要先到达某一种顶端，

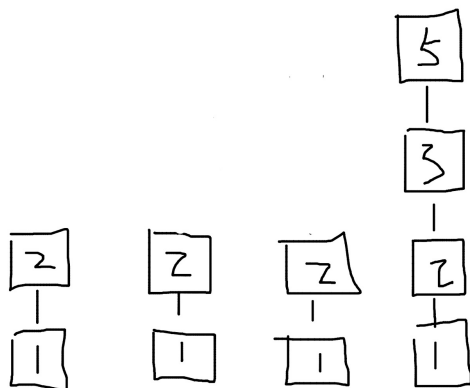


不一定是整个网络的顶端，至少是某种圈子的顶端。

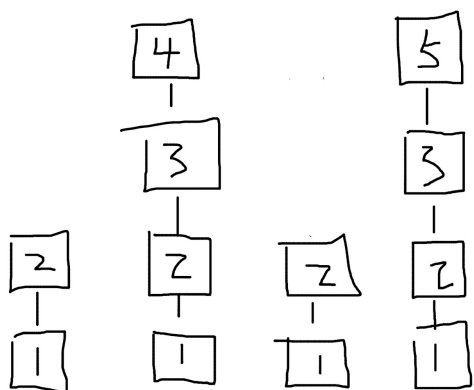
交换数据

类型

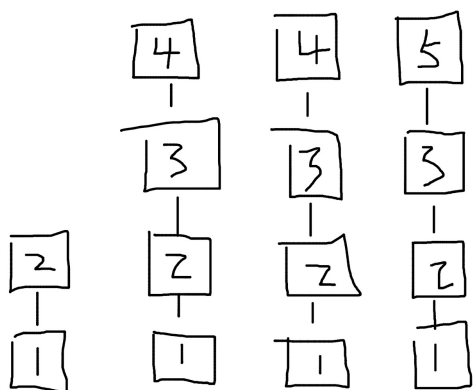
在目前的机制下，网络可能是混乱、不同步的。虽然对数据同步的速度预设是慢的，但如果有的节点就喜欢快呢，用快的计算、大的网络带宽，就是要达到整个网络的最前沿。



也就是大多数节点慢，少数节点快的情况。每个节点都是按照块高度平行更新内容的，也都是按照块高度广播内容的，在一定程度上会缓解这种问题。你想快就快，和我们没有关系，我们慢的自成一派，我们遵循大多数一致的原则，不是谁块高度高就听谁的。你想内卷就尽力去卷，我们不跟你玩。



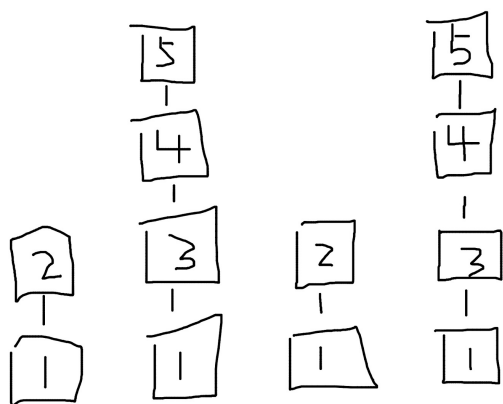
另一种是一半节点慢，一半节点快的情况，也没有什么好担心的，最坏就是形成两个网络，无关痛痒。

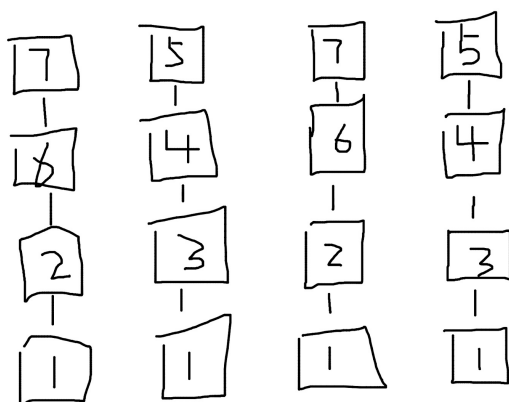


至于少数节点慢，多数节点快，属于最正常的情况了。

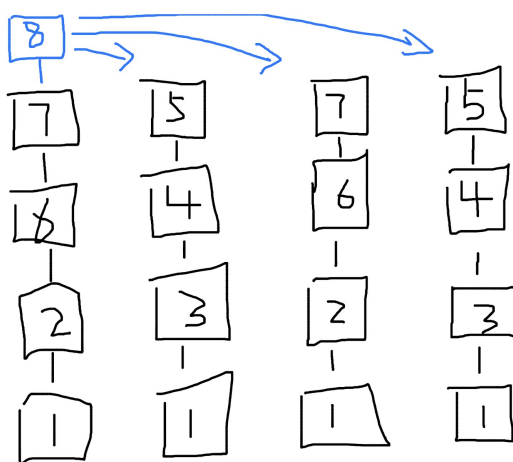
融合

在一半节点慢，一个节点快的情况下，很容易造成这样数据对立的情况，即使块高度一致，也是两种数据。

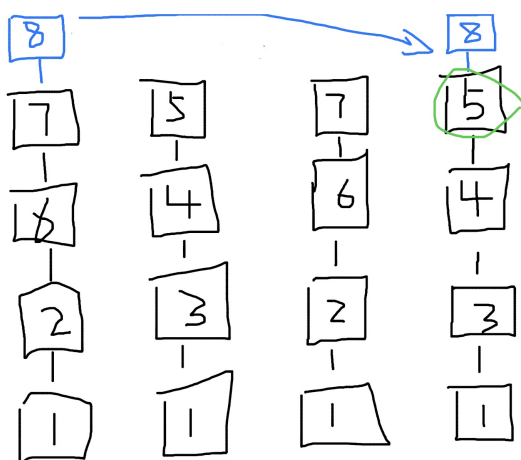




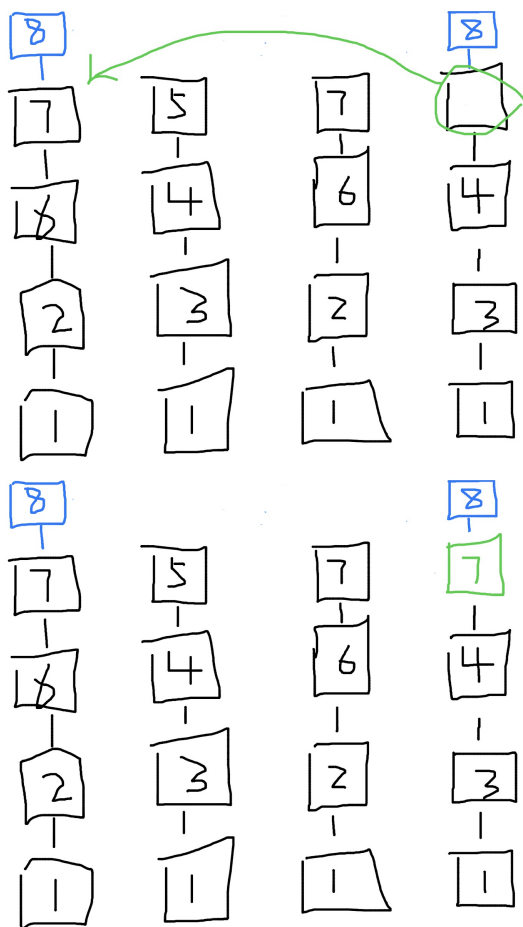
这个时候就不得不有一方妥协了。如果两个网络想要融合，就必须有一方做出一些牺牲。在块高度一致的情况下，假设路由表互通，产生新的块数据后，其实就是“主动广播数据”的过程，当前节点先产生一个块：



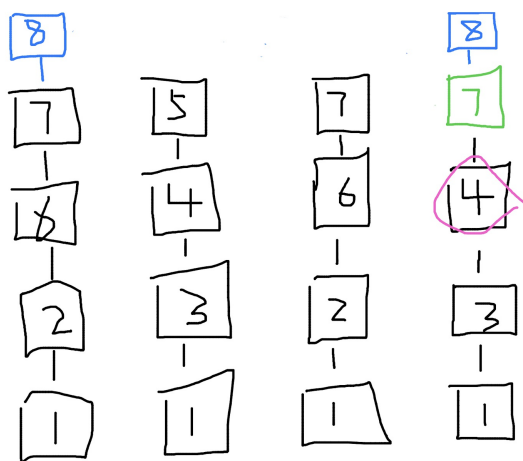
例如最后一个节点在收到块后，发现前块哈希和自己的对不上：



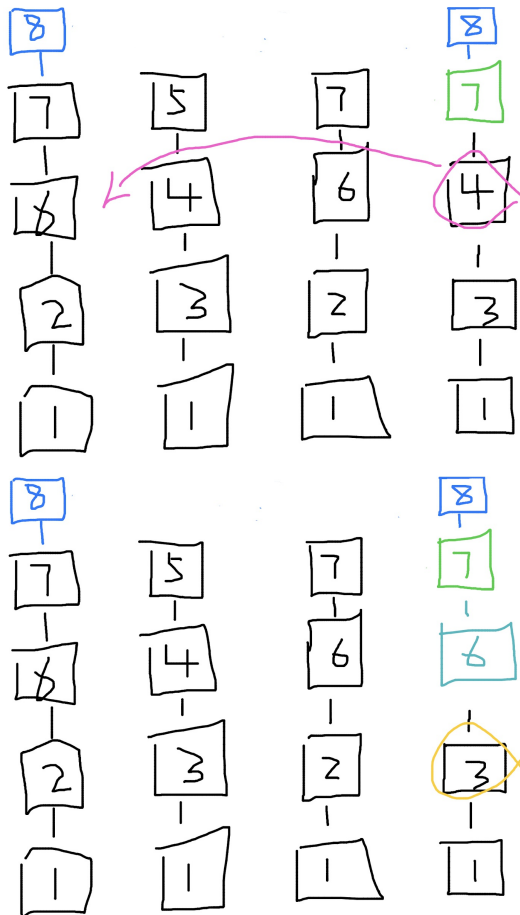
就去其他节点请求上一个块高度的内容：



发现上上个块的哈希对不上:



继续请求其他节点对应块高度的内容;



依次类推，直到整条链完全相同。

被替换掉的块内容可以放到一个缓存队列，作为新块的内容，继续向外广播，减少节点内容的丢失。

总结

这是一种没有经过实践考验的、也无法简单用公式来建模的共识机制的设想，共识以自身立场为出发点，关心自己如何应对网络中其他节点的不同行为，而不是从整个网络的角度“上帝式”地设计交互协议。这样的机制会给网络带来不确定性，但也会带来很多可能性。我们只能考虑节点基本的行为规则，就像我们学习生活规则一样，我们很难预测整个网络的走向，就像我们无法预测世界会向什么趋势发展。这种共识机制并不是而且也许不能解决特定的问题，比如建立电子现金系统或者提供图灵完备的运行平台，它关注在更基础一点的层面，提供一种实现数据一致性的方法和思路。

计算机通过网络组成的虚拟世界，一定也很精彩。