R-DAG：一种用于电子加密货币的规则化 DAG

Sichao Yang
ysc@canonchain.com

2018 年 2 月 27 日

摘要

本文提出了一种规则化的 DAG 电子加密货币，我们称之为 R-DAG (Regularized Directional Acyclic Graph)。通过规则化，R-DAG 可以让每个账户拥有自己独立的账户链，提供点对点之间无限的即时交易以及无限的可扩展性。同时，R-DAG 仍然可以通过 DAG 里的交易结算机制保持原有的安全性。

必要性申明：本文档将随 Canonchain（即“标准链”）项目进展随时修订，如果你是 Canonchain 技术爱好者，请及时留意变动情况，以便及时同步相关信息。

1 简介

区块链作为独立技术的概念在 2015 年开始普及。在此之前，它就是众所周知的比特币技术的数据结构。在中本聪的白皮书 [1] 中，两个词“块”和“链”一起出现，但它只被称为“一连串的区块”。随着比特币的流行，我们将比特币里面用到技术和概念分类为区块链 1.0。随着以太坊 [2]作为分布式应用程序的平台运行，人们开始将以太坊分类为区块链 2.0。现在市场正在争夺谁来命名区块链 3.0，而很多人认为 DAG 技术是区块链 3.0 的最佳选择。

在以比特币和以太坊为代表的传统区块链技术中，区块和交易是两个独立的概念。交易通过矿工打包进入区块获得确认，交易的吞吐量收到区块大小和区块生成速度的限制。由于区块链设计中所涉及的变量太多，一个社区对于区块链的理解不同很容易形成分裂，更糟糕的是没有人似乎同意哪些值是正确的从而造成链的分叉。在区块链中矿工有权决定进入区块的内容，矿工的逐利行为很容易造成区块链中算力或者是投票权的过分集中，从而失去离散化的特性。

基于 DAG 的数字货币就是为了解决这些问题应运而生的。RaiBlocks [3]，IOTA [4]和 Byteball [5]是目前在市场上比较热门的 DAG 项目。他们都具有 DAG 共同的优点，但是设计各有千秋，在性能、复杂度、可靠性和安全性上各有不同。

RaiBlocks

RaiBlocks 的设计从底层实现了零交易费用的可扩展的交易处理，优点非常突出。与许多其他加密货币中使用的传统区块链不同，RaiBlocks 使用块格结构。每个账户都有自己的区块链（账户链），账户链记录本账户的交易/余额历史，并通过委托权益证明(PoS)投票达成共识。每个账户链只能由账户所有者更新；这允许每个账户链立即异步地更新到块格的其余部分，从而实现快速交易。由于区块只能由每个账户链的所有者添加，因此将资金从一个账户转移到另一个账户需要两笔交易：发送交易扣除发件人余额中的金额以及将金额添加到收款账户余额的收款交易。接收交易可以随时执行；收件人在发送交易期间不需要在线。由于 RaiBlocks 协议非常轻便，节点运行成本几乎没有，因此 RaiBlocks 不收取交易费。一个交易适合单个 UDP 数据包，并且交易独立处理，消除了区块大小问题。RaiBlocks 钱包一次交易结束即可为下一个交易预先缓存 PoW 工作量证明，使交易瞬间完成，因为发送接收双方都有准备就绪的工作证明。RaiBlocks 允许修剪节点，甚至每个账户只需要保留账户链的最新区块，进一步减少查找时间和系统资源。由于具有以上这些优点，RaiBlocks 可以提供无限的即时交易以及无限的可扩展性，使得 RaiBlock 成为点对点交易的理想选择。

但是 RaiBlocks 中只有接收方负责确认交易的最终签名，交易缺少全局的结算，使得它具有一定的安全隐患。Raiblocks 中账户管理他们自己的区块链，以实现异步更新和扩展效率。缺点是除非接收方钱包在线，否则发送的交易将永远不会被确认和验证。这是第一个问题，交易可能无限期地未被确认。此外，Raiblocks 还将余额放入发送交易中。这意味着，交易的发送者和接收者可以共谋对帐户余额作假，除非账本已经经过了全局验证。因此 Raiblocks 通过牺牲全局账本验证的安全性而提高性能，这是第二个问题。结合上述两个漏洞，攻击者可以对网络进行 DoS 攻击以篡改账本。

IOTA/Byteball

IOTA 和 Byteball 均采用 DAG 的结构设计，有相当大的相似性，我们在这里一并阐述。在 DAG 中交易不会分组为块，而是直接连接；每个新交易通过引用的方式确认一个或多个以前的交易，并把自己连接到 DAG 上。在 DAG 中，由于不必序列化大小有限的区块中的交易并将它们全部放在单个通道中，对交易吞吐量没有理论限制。基于 DAG 的结构可以并行处理大量事务，并且更加灵活的适应当前负载。另外，DAG 中不需要通过昂贵的 PoW 来封装区块，使得系统的社会成本降低从而为外部世界创造价值。

和传统区块链技术相比，DAG 技术具有以下优点：

1）交易速度快，吞吐量大；

2）几乎无交易费，对小额支付友好；

3）不需要专门矿工参与；

4）可扩展性强。

但是 IOTA 和 Byteball 在细节上面还是有相当大的差异，以迎合不同的受众。IOTA 对每个交易赋于一定的权重，交易通过 PoW 机制接入，这样可以通过判断最大权重路径来防止分叉，同时 PoW 机制还可以防止垃圾交易的攻击。Byteball 没有采用 PoW 机制，而是通过收取少量的手续费来防止垃圾交易的产生，同时引入见证人的投票确定正确的交易。IOTA 和 Byteball 两种技术都有独到的优势，Byteball 的独特功能无疑是其集成的私人资产，提供类似于以太坊的智能合约，甚至进一步扩大了这些合同用于投注体育赛事或政治选举的能力。而 IOTA 的独特功能无疑是不收取交易费用，是目前唯一能够在全球范围内作为物联网主干的技术。

在本文里，我们提出了一种改良的 DAG 设计，该设计结合了 RaiBlocks 多链结构和 IOTA/Byteball DAG 的一些优点，我们称之为 R-DAG (Regularized Direct Acyclic Graph)。通过改良，我们在确保账本安全的前提下，赋予 DAG 更大的吞吐量和更快的处理能力；网络节点可以以更小的空间存储账本，并且在账本中快速的搜索自己的账户。

2 多链结构和 DAG 账本

我们在点到点交易的时候借鉴了 RaiBlocks 的多链结构。每个账户都有自己的一条链，这条链记录这个账户的支付和接收行为。如在图 1 中一共有 8 个账户，分别用 8 条链记录了账户发送和接收交易的记录。在图上，横向的坐标表示时间轴，纵向的坐标表示账户的索引。

将资金从一个账户转移到另一个账户需要两个交易：一个发送交易从发送方的余额中扣除金额，一个接收交易将该金额添加到接收账户的余额。不管在发送端账户还是接收端账户，添加新的交易到账户时都需要进行包含前面交易内容 Hash 的 PoW 工作证明。在账户链中，PoW 工作证明作为反垃圾交易工具，可以在几秒钟之内完成。在单一的账户链中，之前区块的 Hash 字段是已知的，可以预先生成后续块所需的 PoW。因此只要两笔交易之间的时间大于生成 PoW 所需的时间，用户的交易将在瞬时完成。

在这样的设计中，仅需要交易的接收端进行结算。接收端将接收到的交易签名并且放到账户链上，这样的交易就称之为已结算交易。结算后，接收端再将交易广播到其他节点的账本中。但是可能会出现接收端没有上线或者遭受到了 DoS 攻击的情况，导致接收端无法将接收端交易放上账户链，我们把这样的交易称之为未结算交易。如图 1 中的 X 符号表示的即为一个从账户 2 发送到账户 5 的未结算交易。

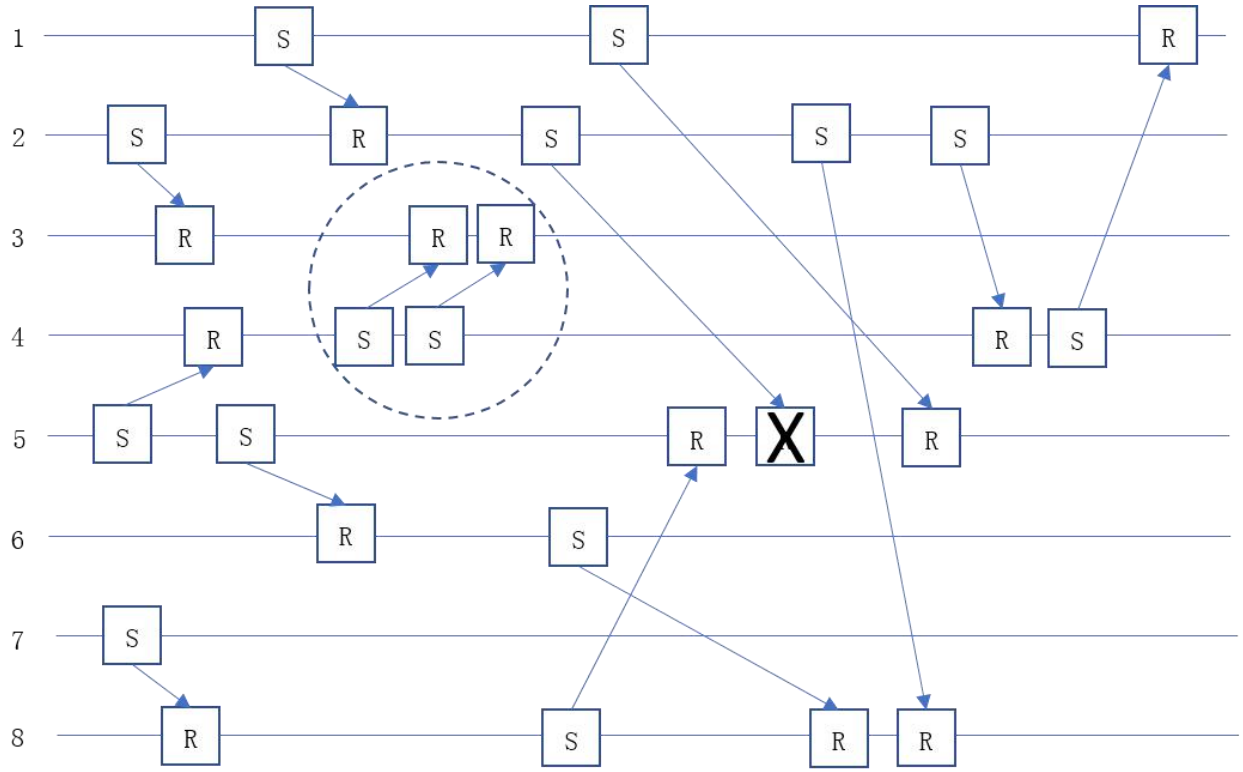

图 1

显然，因为只需要交易的发送端和接收端两者之间进行结算，这样的交易始非常轻量级的，所有的交易都可以在一个 UDP 包中传输，并且以极快的速度进行处理。同时一个账户所有的交易都保留在一个链上，具有很大的完整性，可以对账本进行最小化的修剪。一些节点对花费资源来存储账户的完整交易历史不感兴趣;他们只对每个账户的当前余额感兴趣。当账户进行交易时，会对其累计余额进行编码，这些节点只需要跟踪最新的块，这样可以在保持正确性的同时丢弃历史数据。这样的结算方式只有在发送端和接收端互相信任的情况下进行，并非是整个网络共识的最终结算。在发送端和接收端缺乏信任的情况下，或者是接收端遭受 DoS 攻击而发送端又不知情的情况下，具有安全隐患。对于如何进行最终结算，我们会在后面的章节中叙述。

我们观察到，虽然每个账户都拥有单独的链，但是整个账本却可以用 DAG 的形态来表示。如图 2 所示，即为图 1 中所有账户交易的 DAG 表示。

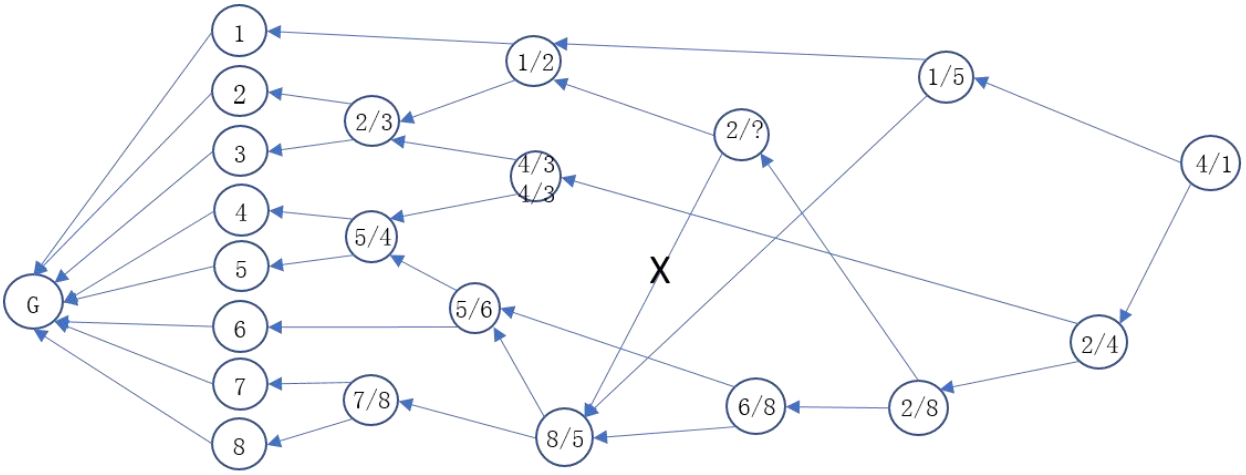

图 2

DAG 中的第一个单元为创世单元，后面的 8 个单元代表初始代币的分配，其他单元都对应于账户链之间的交易。我们用符号 a/b 来表示一个交易，其中发送者为 a，接收者为 b。如图 2 中最后的 4/1 单元就是对应于图 1 中的最后一笔交易 – 从账户 4 发送交易到账户 1。在图 1 中的一个交易是对交易双方账户链上最新区块或者说是最新交易的确认，反映到图 2 上就是一个单元对于交易双方账户链最新单元的引用。以 4/1 单元为例，在此之前账户 4 上最新区块是 2/4 交易的接收区块，账户 1 上的最新区块是 1/5 交易的发送区块。因此在 DAG 上面，4/1 单元引用 2/4 单元和 1/5 单元。

一个交易在通过验证后即被认为是合法的交易被添加到 DAG 中。交易的验证一般通过以下几个步骤：1）区块不能已经存在账本中（重复交易）。2）必须由账户所有者签名。3）前面的块是账户链的头块。如果它存在但不是在头部，那么这是一个分叉。4）账户必须有一个初始化区块。5）计算的哈希值符合 PoW 阈值要求。如果是一个接收区块，检查源区块哈希值是否处于待完成状态，这意味着它尚未被兑付。如果是一个发送区块，余额必须小于之前的余额。

在交易验证中可能会有两种情况需要特殊处理：1)在两个账户之前几乎同时发生了多次交易，这些交易在 DAG 上被合并到一个单元。如图 1 中虚线圆内的两次 4/3 交易。2）接收者没有接收到并结算一个交易，如 图 1 中的 2/5 交易。此时在 DAG 中就以单元符号 2/?表示一个未被结算的交易。而本来应该存在的从 2/5 单元到 8/5 单元的引用也被 1/5 单元到 8/5 单元的引用代替。如果随后 2/5 的交易被账户 5 成功结算，在 DAG 中添加从 2/5 单元到 1/5 单元的引用。

图 2 所示的 DAG 和用于 IOTA/Byteball 的 DAG 有所不同。在图 2 的 DAG 中，一个单元对于之前单元的引用并非是随机的，引用关系的存在与否取决于本单元和之前单元的交易之间是否有重合的帐户。例如，4/1 单元对 2/4 单元进行引用因为他们都有和账户 4 相关联的交易，4/1 单元对 1/5 单元进行引用因为他们都有和账户 1 相关联的交易。

3 R-DAG 账本

DAG 可以用作电子货币账本的本质是通过后到的交易为前面的交易做确认。通过一定的机制使得每一个交易随机选择之前的头部交易(tips)，可以保证所有的交易都能以一定的概率获得确认，同时保证 DAG 不断增长。等到 DAG 增长到一定长度，一笔交易获得足够多的后续交易的确认之后，我们可以认为这笔交易为不可逆的。因此，DAG 中交易的确认速度和交易的到达速度正相关。在这样的设计中，由于总体上在整个系统中交易到达的速率是非常快的，即使某些账户的交易频度不高，但还是能被那些高频的交易确认。

而在多链结构中（图 2），虽然多条账户链的设计可以完成无费用的快速交易，但是交易确认关系和交易的账户相关，并非随机。当两个账户之间的交易不够活跃时，他们的交易就无法快速的获得后续交易的确认。为了解决这个问题，我们让每个交易在最终连接到账本上时，在保留对自己相关账户之前交易的确认外，必须随机确认另外的一笔交易。在图 2 中添加用虚线表示的随机确认即得到图 3。

在图 3 所示的 DAG 中，每个单元所做的三个确认中只有一个是从 tips 里面随机选取的，其他两个都是固定和交易账户关联的。因此我们称之为具有一定规则的 DAG, 也即 R-DAG (Regularized Direct Cyclic Diagram) 。

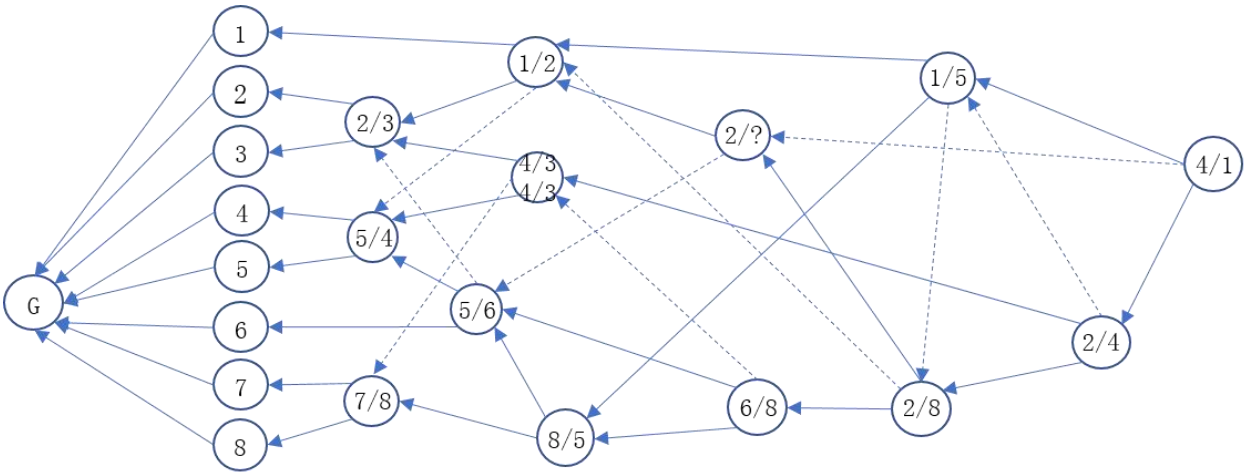

图 3

在 R-DAG 的设计中，两个互相信任的账户可以快速的在他们自己的链上结束交易。而对于有任何一方不可以信任的情况，对方可以到 DAG 上面寻找最终确认。同时两个账户之间也无法共谋作假，因为他们之间的交易最终需要在 DAG 上面和其他账户达成共识。

与 IOTA 相比，在 R-DAG 中节点确认交易的代价有所提高，因为每个单元需要总共做 3 次 PoW 的运算才能把自己放到 DAG 中。但是考虑到硬件性能的上升和价格的降低，增加的计算代价基本可以忽略不计。

R-DAG 交易查找和存储

在 IOTA 或者 Byteball 中，由于单元接入 DAG 的随机性，要查找一个交易记录很可能需要遍历整个 DAG。而在 R-DAG 中，由于互相引用的单元之间具有账户的关联性，我们只需要从创世单元开始对 DAG 进行深度优先搜索（Depth-First Search）即可以找到所需账户的特定交易记录。

在 R-DAG 中一个被完全确认的单元的入度(incoming degree)为 3，我们把没有被完全确认的单元也即入度小于 3 的单元称为头部单元(tips)。假设在 R-DAG 中一共有 n 个账户，很显然在 R-DAG 中 tips 的数量是少于 n 的。因此随着交易的增加，R-DAG 会朝着一个方向增长。这是 R-DAG 的另外一个优点，R-DAG 的 tips 数量只取决于账户的数量，而与交易的到达速度无关，具有良好的可扩展性。而在 IOTA 和 Byteball 中，tips 的数量是和交易速度正相关的，DAG 的宽度可以随着交易速度的提高而增加。

因此在 R-DAG 中查找交易的复杂度为 $O(3^m)$，其中 m 为查找的深度也即该交易在账号中对应的次序。图 4 显示了在 DAG 中通过深度优先查找账户 4 所有交易记录的过程，而查找所得

即对应账户 4 自身的账户链。轻量级的账户无需存储整个 DAG，只需要存储自身的账户链即可。在结算交易时，轻量级的账户可以通过可信节点调用 API 查询交易是否被最终结算。

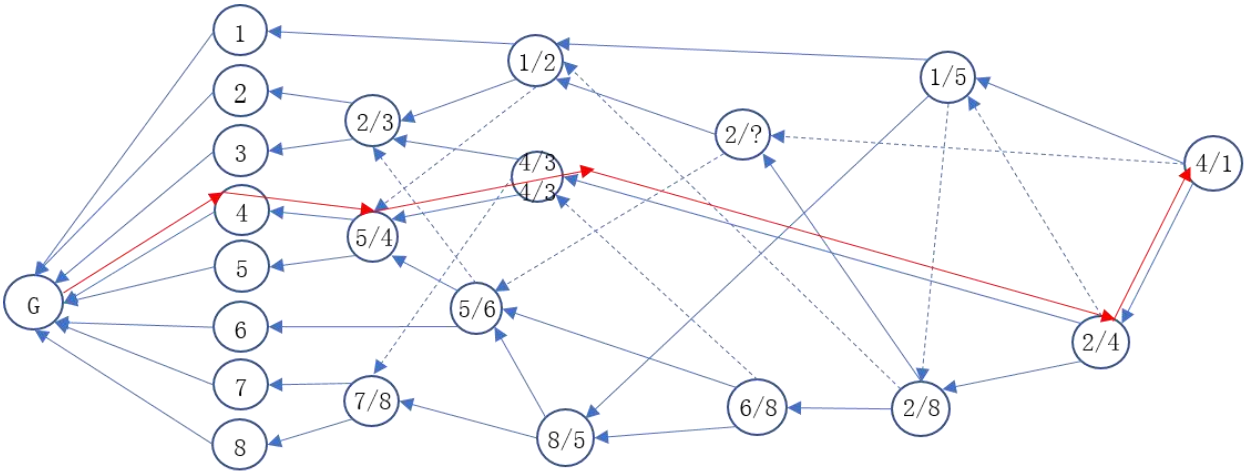

图 4

4 见证人

R-DAG 中的链定义了单元之间的相对次序，在同一条链上离创世单元较近的单元中的交易要比较远的单元中的交易发生的早。例如，在图 4 中账户 4 对应的链中，4/3 交易要比 2/4 交易早。但是当两个交易不在任何一条链上时，我们无法得知他们之间的相对次序，如图 4 中的 1/2 和 6/8。因此，和 Byteball 中的方法类似，我们在 R-DAG 中寻找一条主链来定义交易之间的绝对次序。

R-DAG 中的主链通过多数见证人(witness)投票确认，粗略的来说 R-DAG 中包含数量最多的被多数见证人确认单元的链即为主链。见证人是非匿名长期参与社区并拥有良好信誉的人，或是主动维护网络健康发展的组织，或者自身利益与 R-DAG 账本的正确性密切相关的人。虽然不能期望每个见证人都做出正确的判断，但是可以安全的假设多数见证人是诚实的。

见证人可以通过社区以不同的规则选举产生：如社区可以选举在 R-DAG 中代币最多的人作为见证人，形成 PoS（Proof-of-Stake）共识机制。在标准链(CanonChain)中，我们采用 PoP(Proof-of-Participation)基于节点贡献度的共识算法。关于 PoP 共识算法，我们会另开一文详细论述。

5 主链和交易序列值

在 R-DAG 中，一个单元的级别(level)定义为此单元和创世单元之间最长路径的长度。一个单元的主链从创世单元开始以递归的方式定义如图 5 所示：

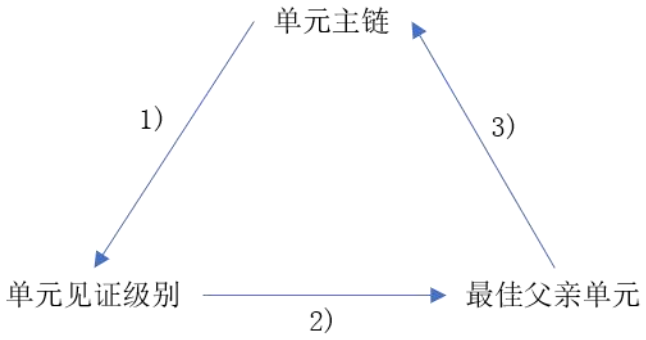

图 5

1) 从一个单元主链往创世单元方向追溯，如果经过多数个见证人，那么最后一个见证人在 R-DAG 中的级别即为该单元的见证级别；否则单元的见证级别为 0。
2) 一个单元的所有父亲单元里面见证级别最高的那个即为最佳父亲单元。
3) 一个单元主链 = 最佳父单元主链 + 该单元本身。

需要注意的是，主链是针对于每个单元定义的，不同单元的主链不一定相同。可以证明，把所有的主链组合在一起就是 DAG 上面的一棵树。

由于在 R-DAG 中基本上每个单元的入度和出度都为 3，整个图有非常好的连通性。从不同的 tips 开始的主链很快就会收敛到一条链上，我们把这条链表识为整个 R-DAG 的主链。如图 6 所示，所有账户里面帐户 1 和账户 6 为见证人账户，1/5 单元和 4/1 单元的主链在 2/8 单元处收敛。

在主链选定之后，就可以将所有单元关联到此链。所有的单元要么将直接位于主链上，要么主链上的单元存在引用关系。我们把主链上的单元按照和创世单元的距离建立序列值，离创世区块越近序列值越小，而非主链上的单元也可以根据它们和主链单元的引用关系建立序列值：非主链上的单元的序列值是对该单元进行直接或者间接引用的主链上单元的最小序列值。通过定义序列值可以确定 R-DAG 上所有交易的绝对次序，为图上无序的单元之间建立总序。

如果在 R-DAG 中出现包含有互相冲突的单元，具有较小序列值的单元被 DAG 接收，而另一条具有较大序列值得单元则被认为是比较后产生的双重支付而被判为无效。由一个特定的单元建

立的序列值告诉我们这个单元的对过去事件顺序的看法，即他对历史的记录。如前所述，该顺序会影响到对冲突单元的选择。

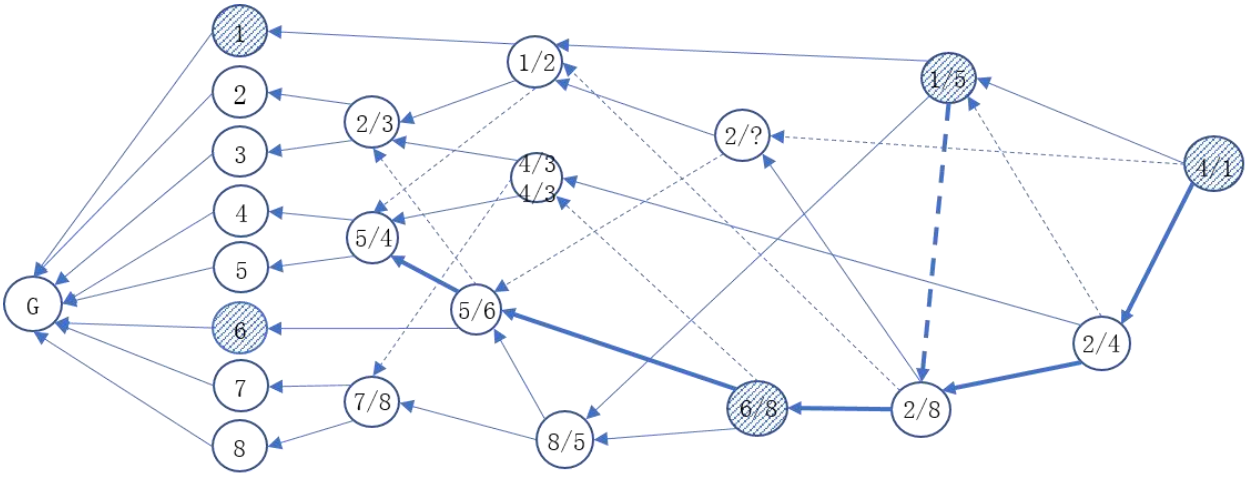

图 6

6 最终结算

如前面所述，R-DAG中的主链决定了各单元中交易的先后次序的判定。在主链中存在着一个“稳定点”，在稳定点之前的一段主链不会随着新单元的到达而改变，因此和此段主链相关联的所有单元的序列值不会改变，所有序列值小于该稳定点的交易都是不可逆的。很显然，创世单元就是第一个稳定点。如果我们能够证明稳定点随着新单元的到来会逐渐后移，稳定点的序列值不断变大，那么越来越多的交易就成为不可逆的。如果在当前的稳定点存在一个当前主链的分支和一个替代分支，如果新单元的到来无法让替代分支取代现有的主链分支而成为新的主链，那么稳定点就可以在主链上后移。

我们在此仅讨论主链的稳定点之后已有多数见证人的情况。随着时间的推移和见证人交易的叠加，这个条件是一定可以满足的。假设稳定点的级别为s；在多数见证人中，假设最低的级别为$\underline{w}$, 最高级别为$\overline{w}$。那么根据假设 $s \leq \underline{w} \leq \overline{w}$。我们需要找到一个替代分支将失去任何超过当前的主链 机会的条件。让我们从前面的例子中定义$\underline{w}$开始。在替代分支的所有单元中，我们则选择那些增加见证水平的人，即他们自己的见证水平大于每个父母单元的见证水平。在这些中，我们找到最大水平，然后即使所有剩余的（少数）证人聚集在替代分支上，替代分支上的见证等级将永远不会超过这个最大等级。因此，如果这个最大水平小于$\underline{w}$，对于替代分支来说则游戏结束，并且我们可以沿着当前主链前移稳定点。

因此，在当前主链上存在一点，在该点之前主链将永远不会改变（假设大多数证人不发布非序列单元）。因此，相对于该主链定义的总顺序也是最终的。如果我们有非序列单元，我们可以决定其中哪一个是有效的，以及最终的。如果新的非序列单元出现与已经稳定在主链上的任何内容发生冲突，则在旧对应单元之后，新的非序列单元将一定会被排序，并且新的非序列单元将被认为是无效的。因此，包括在稳定主链上的单元中进行的任何付款已经不可逆。

与比特币不同，交易最终性只是概率性的，在这里确定性交易的终极性。每个用户基于他所看到的单元构建他自己的（主观的）当前主链。由于新单元的传播不是即时的，并且它们可以以不同的顺序到达不同的用户，所以在任何给定时间内，用户将具有不同的当前主链以及关于主链最后稳定点的不同看法。

R-DAG中最终结算的方法和Byteball中的结算方法类似，需要了解更详细信息的读者可以参考Byteball白皮书。

7 安全性

像所有去中心化加密货币一样，R-DAG 可能会遭到恶意者攻击，企图获得经济利益或使系统崩溃。在本节中，我们讨论一些可能的攻击场景、这些攻击的后果以及 R-DAG 如何采取预防措施。

7.1 双花问题 (double spending)

和 RaiBlock 里面定义的接收者交易确认方式不同，在 R-DAG 中虽然每个账户仍然拥有自己的账户链，但是账户交易也被记录到 DAG 中被其他交易所确认。如前所述，根据见证人的观察，所有的交易在 R-DAG 上面都有自己的绝对序列号，系统可以判对两边交易的时间先后从而对双花问题进行裁决。

7.2 垃圾交易和小额交易攻击

由于在 R-DAG 里面采用零交易费的模式，恶意者可以以零费率在其控制下的账户之间发送许多不必要的但有效的交易，试图使网络饱和。R-DAG 通过 PoW 限制了恶意者在不显著投入计算资源的情况下可能产生的交易率。

为了防止存储空间被不必要的交易消耗，R-DAG 可以对非完整历史节点在一个统计指标下进行裁剪。如果节点想要进行更加激进的修剪，它们可以根据访问频率来算出统计分布，将不经常使用的账户委托给较慢的存储。

7.3 网络故障和 DoS 攻击

一个节点可以会由于网络故障或者遭受 DoS 攻击无法接受发送给它的交易。虽然接收账户无法确认交易，并把交易放到自己的账户链上。但是这个未结算交易仍然被保留在 R-DAG 中，在等待一段时间后接受者已经无法修改未结算交易，此时发送者可以重新发送一次新的交易。通过在 R-DAG 上保留未结算交易可以避免节点的重新同步所造成的各种问题。

7.4 >50% 攻击

R-DAG 的结算是通过见证者的投票进行的，拥有大多数见证者的路径成为 R-DAG 的主链，拥有判断 R-DAG 中交易次序的权力。因此，如果有超过半数的见证者达成共谋的话，整个 R-DAG 上的交易次序是可以被改变的。

有多种办法可以防止这种行为的发生。例如，把见证人的选择与投资挂钩的，见证人本质上倾向于去维护一个诚实的系统，以保护他们的投资。试图操纵账本将对整个系统造成破坏，从而造成投资损失。

8 参考文献

[1] S. Nakamoto, "Bitcoin: A Peer-to-Peer Electronic Cash System," https://bitcoin.org/bitcoin.pdf.

[2] V. Buterin, "Ethereum Whitepaper," https://github.com/ethereum/wiki/wiki/White-Paper.

[3] C. LeMahieu, "RaiBlocks: A Feeless Distributed Cryptocurrency Network," https://raiblocks.net/media/RaiBlocks_Whitepaper__English.pdf.

[4] S. Popov, "The Tangle," https://iota.org/IOTA_Whitepaper.pdf.

[5] A. Churyumov, "Byteball: A Decentralized System for Storage and Transfer of Value," https://byteball.org/Byteball.pdf.