基于信用投票共识的主从多链分层跨链模型

王瑞锦*，郭上铜，邱玮鸿，张凤荔

(电子科技大学信息与软件工程学院 成都 610054)

【摘要】该文提出了一种适用于联盟链的基于信用投票机制的共识算法 (PoVT)。该算法通过引入投票机制来决定记账权的归属，避免了节点之间的算力竞争，使系统中的节点能够公平地获得记账权；通过给节点赋予信用值，减小权益对系统的影响，同时对节点的行为进行量化评价能够更好地约束节点的行为，使其对恶意行为产生顾虑；此外，在 PoVT 的基础上提出了一个主从多链的分层跨链模型，对其性能进行了实验分析，结果表明系统的效率有了提高，且对双花攻击、自私挖矿、权益粉碎等攻击手段都有一定的防御能力。

关 键 词 区块链； 共识算法； 跨链； 信用值； 投票机制

中图分类号 TP311.13 **文献标志码** A doi:10.12178/1001-0548.2021103

A Master-Slave Multi-Chain Hierarchical Cross-Chain Model Based on Credit Voting Consensus

WANG Ruijin*, GUO Shangtong, QIU Weihong, and ZHANG Fengli

(School of Information and Software Engineering, University of Electronic Science and Technology of China Chengdu 610054)

Abstract This paper proposes a consensus algorithm based on the credit voting mechanism for alliance chain, named proof of vote and trust (PoVT). This algorithm introduces the voting mechanism to decide the ownership of the accounting privilege, avoids the competition of the computation power among nodes, and makes all nodes in the system get the accounting privilege fairly. By assigning credit value to nodes, the influence of both rights and interests on the system can be reduced. Meanwhile, quantitative evaluation on the behavior of nodes can better prevent nodes conducting malicious behaviors. Based on the proposed PoVT, a master-slave multi-chain layered cross-chain model is constructed and the performance is evaluated through experiments. The results show that the efficiency of the system is improved, and the system has a certain defense capability against attack methods such as double spending attack, selfish mining, and nothing-at-stake attack.

Key words blockchain; consensus algorithm; cross chain; trust values; voting mechanism

区块链作为一种分布式数据库，最基本也是最核心的要求是所有的节点能对所保存的数据达成一致，因此共识机制一直是被研究的重点。区块链共识算法[1-3]主要有工作量证明算法 (proof-of-work, PoW)、权益证明算法 (proof of stake, PoS)、股权权益证明算法 (delegated proof-of-stake, DPoS) 及拜占庭容错算法 (practical Byzantine fault tolerance, PBFT)。其中，PBFT 算法不适用于节点规模庞大的公共链；DPoS 算法存在周期过长、不能及时清理恶意节点的问题；PoW 算法存在大量算力浪费及电力浪费的问题，同时在交易吞吐量、延迟、确认时间等方面有较大缺陷；PoS 算法在一定程度上减轻了 PoW 算法资源浪费、出块时间长等问题，但“币龄”[4]的出现也带来了“富者愈富”的问题，同时该算法对代币有较大依赖。

现有的区块链可以分为公有链、私有链及联盟链 3 类。在公有链中，节点可以自由地加入或退出，无需身份审核，如比特币、以太坊等平台。私有链搭建在本地，不做公开用途。联盟链的节点接入需要获得许可，能在一定程度上保证网络中节点的可靠性。但现有的区块链平台几乎都是相互独立的，每一个平台之间的数据不能相互传递，形成事

收稿日期：2021－04－12；修回日期：2021－07－08

基金项目：国家自然科学基金 (61802033，61472064， 61602096)；四川省区域创新合作项目 (2020YFQ0018)；四川省科技计划重点研发项目 (2021YFG0027，2020YFG0475，2018GZ0087，2019YJ0543)；广东省国家重点实验室项目 (2017B030314131)

作者简介：王瑞锦 (1980－)，男，副教授，主要从事区块链、网络与信息安全、知识图谱及人工智能等方面的研究.

*通信作者：王瑞锦，E-mail：wrj8882003@163.com

实上的“信息孤岛”，不仅造成资源浪费也不利于区块链整体行业的发展，跨链因此成为区块链研究的一个重点方向。

针对上述情况，本文提出了一种基于信用投票共识机制 (proof of vote and trust, PoVT) 的跨链方案，并进行了测试，结果表明 PoVT 的交易处理性能有较大提高，同时对权益粉碎攻击、贿赂攻击及自私挖矿攻击等都有较好的防御能力。本文的主要工作有：1) 构建了一个主从多链的分层跨链模型，实现了从链与从链之间的数据跨链；2) 提出了一种基于信用投票机制的共识算法 (PoVT)，解决了 PoS 共识可能出现的权益粉碎攻击、“富者愈富”问题以及通过投票和随机选择出块者的方式解决了通过算力竞争记账权的问题；3) 从链的所有区块信息在主链上通过 PBFT 算法完成共识，保证了从链区块信息的安全性和不可篡改性，实现从链到主链的单向数据传递。

1 相关工作

PoW 算法通过算力竞争将记账权分配给全网所有节点，获得记账权的诚实节点能得到一定的数字货币作为贡献算力等资源的奖励。但 PoW 算法也浪费了大量的算力与电力资源，且 10 min 一个的出块速度也限制了其商业价值。除此之外，PoW 算法还容易遭到自私挖矿 (selfish mining)[5] 攻击，攻击者不需要掌握超过 51% 的算力，只需要掌握全网 1/3 的算力即可发起攻击。本文提出的基于信用投票机制的共识算法通过引入投票机制来决定记账权的归属问题，解决了 PoW 中自私挖矿、51% 攻击等问题，在保证系统的稳定性与公平性的同时解决了由算力竞争带来的资源浪费等问题。

PoS 共识的出现对解决 PoW 共识所消耗的大量算力与电力起到了一定的缓解作用，且加速出块时间也能提高对交易的处理速度和吞吐量，但 PoS 本质上还是需要通过哈希运算来竞争记账权，且“币龄”的存在也降低了数字货币的流通性。对 PoS 算法的研究有：文献 [6] 提出了 PoC(proof of credit) 共识机制，在 PoS 共识的基础上引入信用值作为一种特殊的权益，通过提出的选举人机制、自我审计机制以及混合激励机制使系统能够抵御双花攻击以及自私挖矿攻击；文献 [7] 提出了一种基于 PoS 的抽奖共识机制来选择出块者，节点将自己拥有的权益分成更小的单元，然后通过哈希算法得到一个哈希值，而拥有与哈希值最接近的单元的节点成为出块节点，通过这种方法彻底解决了 PoS 本质上还是要通过哈希运算来竞争记账权的缺点，同时也增加了 PoS 的可扩展性。

PoS 通过引入“币龄”来解决 PoW 中的 51% 攻击以及资源浪费等问题。原理是，持币越多的人越希望系统能够稳定从而使自己的利益不受损害，但“币龄”的存在同样也带来一些问题，如持币越多的人越容易得到记账权，导致“富者越富”，从而使得像权益粉碎攻击、贿赂攻击等更频繁发生。而本文提出的基于信用投票机制的共识算法通过给节点赋予信用值的方式降低了权益的影响，同时也降低了上述攻击对系统产生的影响。

2 基于信用度投票共识算法的主从多链跨链方案

本文提出了 PoVT 共识算法。该共识通过引入投票机制和信用值，提高了共识效率的同时保证了系统的安全性。在此基础上，构建了一个从链基于 PoVT 共识，主链基于 PBFT 共识的主从多链分层跨链模型。

2.1 模型描述

模型中的节点被分成 4 种角色：普通节点 N_o、投票节点 N_v、生产节点 N_p、候补节点 N_c、代表节点 N_m。同时将时间划分为一个个时间片段，每一个时间片为一个时隙，从链在每个时隙完成校验、出块到上链的全过程，而一个周期则由许多个时隙组成，此外在每一个周期的最后一个时隙结束后，代表节点将该周期所有已确认的区块数据上传至主链网络中。在每一个周期开始前会根据节点的权益以及信用值 (STrust 值) 从希望参与共识的节点中选择一些节点组成一个共识节点集合 N={(A_1, S_1, STrust_1), (A_2, S_2, STrust_2), ⋯, (A_n, S_n, STrust_n)}。该集合中的节点由生产节点、投票节点、候补节点 3 种角色组成。其中生产节点从交易池中取出交易并打包组装成区块；投票节点对数据进行验证并投票；候补节点负责在生产节点或投票节点因为某些原因无法继续提供服务时，递补成为该角色继续行使使命，保证了系统的安全性与稳定性。在集合 N 形成后，通过 PoVT 共识来决定每一个时隙产生区块的生产节点编号，同时在这个过程中通过运行梅森旋转算法[8] 生成一个伪随机数来产生组成主链的代表节点的编号。集合 N 中的节点允许同时拥

有主链和从链的双重身份，主链节点负责将其所在主体每一周期内已被确认的区块数据上传至主链网络中，主链节点之间再通过 PBFT 算法对数据达成共识，形成一条主链。系统的结构如图 1 所示。

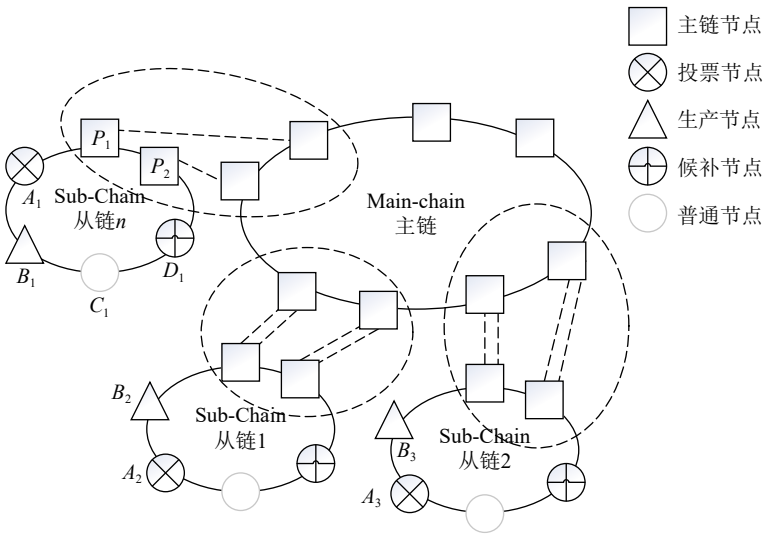

图 1　系统结构图

2.2　从链共识机制

2.2.1　准备阶段

在集合 $N=\{(A_1, S_1, \mathrm{STrust}_1), (A_2, S_2, \mathrm{STrust}_2), \cdots, (A_n, S_n, \mathrm{STrust}_n)\}$ 中，节点的个数 $|N|=\mathrm{Num_p}+\mathrm{Num_v}+\mathrm{Num_c}$，$A$ 为节点的公钥地址，S 为节点的权益，STrust 值为节点的信用值，$\mathrm{Num_p}$ 为生产节点的个数，$\mathrm{Num_v}$ 为投票节点的个数，$\mathrm{Num_c}$ 为候补节点的个数。将集合中的节点进行编号，编号 1~$\mathrm{Num_p}$ 的节点成为生产节点，编号 $\mathrm{Num_p}$+1~$\mathrm{Num_p}$+$\mathrm{Num_v}$ 的节点为投票节点，剩下的个数为 $\mathrm{Num_c}$ 的节点成为候补节点，普通节点 N_o 不参与共识但需同步最新数据块至本地。

2.2.2　基于投票和信用机制的 PoVT 共识

PoVT 共识中的投票机制[9] 避免了节点之间的算力竞争，引入的信用机制能够保证参与共识的节点的可靠性，同时降低权益对记账权分配的影响，从而增大对系统发起权益粉碎攻击、双花攻击、自私挖矿攻击等的难度。

1) 共识流程

网络中的普通节点产生交易数据，① 在周期开始前，根据节点的权益及 STrust 值形成一个共识节点集合。② 在集合中从范围为 $(1, 2, \cdots, \mathrm{Num_p})$ 的生产节点中选择编号与随机数 R 相同的节点成为出块节点，该节点从交易池中取出一些交易打包并组装成区块，随后将区块广播给投票节点并准备接受投票节点的反馈消息。在这一阶段，如果要产生的区块是创世区块的话，则 R 为 1。如果为非创世区块的话，随机数由上一生产节点在生成新区块的过程中产生。③ 投票节点在收到生产节点发出的区块验证请求后，对区块中的数据进行验证，验证无误后签名并加盖时间戳 (timestamp)，并广播确认信息。若发现数据有误，则广播一个拒绝消息。④ 生产节点在收到 $\mathrm{Num_v}/2+1$ 个投票节点的确认消息后则表明已对该区块达成共识，生成一个记录此时时间的时间戳，然后对该区块进行后续的签名、广播等操作。反之，若网络中存在超过 $\mathrm{Num_v}/2+1$ 个投票节点发出拒绝消息则判定该生产节点有恶意行为，立即取消该节点的共识资格并由编号为 R+1 的生产节点继续进行共识流程 (若 R+1>$\mathrm{Num_p}$，则从第一个生产节点开始)，同时在候补节点中根据编号顺序递补成为新的生产节点。⑤ 每个被选择的生产节点都被要求在一个时间 T_b 内完成出块，若超过这个时间还没能完成出块则由编号为 R+1(若 R+1>$\mathrm{Num_p}$，则从第一个生产节点开始) 的生产节点继续完成下一个区块的产生。⑥ 在每一轮周期结束后，成功参与共识的节点会获得 STrust 值奖励。相应地，有恶意行为的节点也会受到降低 STrust 值的惩罚，则该节点后续想要参与共识的难度增加。共识流程如图 2 所示。

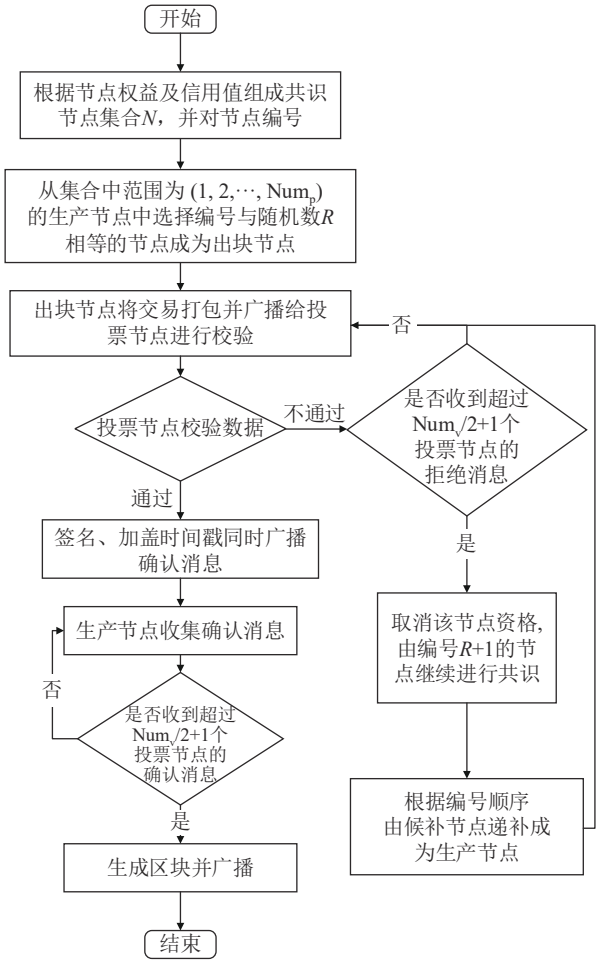

图 2 共识流程图

2) 随机数 R 的产生

除了生成创世区块的 R 值默认为 1 之外，每一个生产节点生成区块的同时也产生一个记录在区块中的随机数 R 来决定谁是下一个生产节点，随机数 R 生成的过程如下：

生产节点在向投票节点提交区块后同时收集投票节点的反馈消息，即 Signature[i]($1 \leqslant i \leqslant \mathrm{Num_v}$)，同时根据时间戳 TimeStamp 由式 (1) 得到 Rsource：

$$\mathrm{Rsource} = \mathop{\oplus}_{i=1}^{k} \mathrm{Signature}[i] \oplus \mathrm{TimeStamp} \quad \mathrm{Num_p}/2 < k < \mathrm{Num_v} \tag{1}$$

式中，⊕表示异或操作，然后对得到的 Rsource 进行哈希运算，取字符串的后 32 位将其转化成整数，得到 R'，如式 (2)：

$$R' = \mathrm{StrToInt}(\mathrm{SubStringEnd32}(\mathrm{Hash}(\mathrm{Rsource}))) \tag{2}$$

由式 (3) 可得随机数 R：

$$R = R' \bmod \mathrm{Num_p} \quad 1 \leqslant R \leqslant \mathrm{Num_p} \tag{3}$$

通过这种方式随机地选择生产节点，避免了当前生产节点出于利益考虑而干扰下一生产节点的选择，能够有效地预防节点之间地共谋攻击，也保证了共识过程的公平与稳定。

3 信用机制

信用机制通过综合考虑一个节点的有效出块数、有效投票数、参与度等因素，使用 STrust 值来定量地描述一个节点的可信度，再结合节点本身的权益来决定节点是否能够参与共识过程。

1) 节点有效出块数。即生产节点在整个周期内生成的有效区块的数量。若一个生产节点在属于它的时隙内成功生成一个通过验证的区块，则认定该节点生成一个有效区块。反之，为无效区块。节点有效投票数γ表示为：

$$\gamma = a \sum_{s=1}^{n} \frac{1}{2^{ct}} B_i^s \tag{4}$$

式中，B_i^s表示节点 i 在第 s 个时隙内是否成功生成区块，若成功生成区块则为 1，否则为 0；t 是节点生成区块的时间，γ随着 t 的增大而减小；$c(c \in [0,1])$是时间系数，可根据系统的实际考虑调节出块时间对系统的影响；a 是权重，可根据系统实际需要调节γ对系统的影响。

2) 节点有效投票数。即节点在整个周期内投出的有效票数。若节点在正确验证区块及数据无误后按要求签名确认则认定为有效投票。与此同时，若节点对某一区块投出了确认票，但在该时隙内网络中存在超过 $\mathrm{Num_v}/2+1$ 个拒绝消息，则认定该节点的此次投票无效。节点的有效投票数表示为：

$$\omega = b \sum_{s=1}^{n} \frac{1}{\sqrt{\frac{m}{n}}} V_i^s \tag{5}$$

式中，V_i^s表示节点 i 在第 s 个时隙内是否投出有效票，若是则为 1，不是则为−1；m 为该周期内总的时隙数；n 为节点i在该周期内实际参与的时隙数；$b(b \in [0,1])$ 为权重。

3) 节点参与度。即节点参与交易的情况。节点参与度λ可表示为：

$$\lambda = f \sum_{s=1}^{n} \mathrm{tran}_s \tag{6}$$

式中，tran_s 表示的是节点 i 在第 s 个时隙内产生的区块中参与的交易数，若节点 i 是交易发送者或接收者其中一方则为 1，否则为 0；$f(f \in [0,1])$ 为权重。

4) 历史信用影响度。节点的历史信用影响度是指节点的信用值受其历史信用值与权益的影响。节点的历史信用影响度ε可表示为：

$$\varepsilon = g\ln^{\text{stake}\times\text{trust}_i^{h-1}} \tag{7}$$

式中，stake 表示节点所拥有的权益；trust_i^{h-1}表示节点 i 在第 h 个周期之前的信用值；$g(g\in[0,1])$ 是权重；ε 的值随着trust_i^{h-1}的值增大而增大，目的是降低节点权益的影响占比。

5) 惩罚因子。为确保系统能够安全稳定地运行，需要采取措施对节点的一些恶意行为进行惩罚。惩罚因子θ表示为：

$$\theta = c\sum_{s=1}^{n} b_i^s + d\sum_{s=1}^{n} V_i^s \tag{8}$$

式中，b_i^s表示节点 i 在第 s 个时隙内是否产生了无效的区块，若是则为 1，否则为 0；V_i^s表示节点 i 在第 s 个时隙内是否发出了无效的投票，若是则为 1，否则为 0；c 和 d 分别为对应的权重，可根据系统实际需要灵活调整对无效块和无效票的惩罚力度。

6) 节点信任度更新公式

周期结束后，系统会根据公式评价共识节点的行为，更新其信用值。信用值更新公式可表示为：

$$\text{trust}_i^h = \begin{cases} \text{trust}_i^{h-1} + \sum_{s=1}^{n}(\gamma+\omega+\lambda-\theta)+\varepsilon & i\in\left[1,\text{Num}_\text{p}+\text{Num}_\text{v}\right] \\ \text{trust}_i^{h-1}+1 & i\in\left[\text{Num}_\text{p}+\text{Num}_\text{v}+1,|N|\right] \end{cases} \tag{9}$$

式中，trust_i^h表示节点 i 在第 h 个周期的信用值，为提高节点参与共识的积极性，若节点被选入集合 N 中但实际未参与共识 (如候补节点)，系统也会给予这些节点信用值奖励。

同时，若节点因为作恶而导致 STrust 值降至系统设定的阈值以下，则会被限制为每隔若干个周期才能参与一次共识，且若节点持续作恶直至 STrust 值降为 0，则该节点将会永久丧失参与共识的资格。

4　主链共识机制

4.1　主链节点选择

主链节点从集合 N 中选择，利用从链生产节点计算随机数 R 的过程中得到的中间数据 R'作为梅森旋转算法 (MT19937-32) 的种子得到一个随机数 R_m，再从集合 N 中选择编号与 R_m 相同的节点作为代表节点构成主链。R_m 的计算过程如下：

首先将 R'作为种子赋值给 MT[0]，根据式 (10) 递推得到剩下的 623 个状态，完成全部 624 个状态的填充。

$$\begin{aligned}\text{MT}[i] = &\text{lowest32bitsof}\\ &(f(\text{MT}[i-1]\oplus(\text{MT}[i-1])>>30)+i)\end{aligned} \tag{10}$$

然后对得到的旋转链进行遍历并根据式 (11) 对每一个状态位进行处理：

$$\begin{aligned}\text{MT}[i] = &\text{MT}[i+m]\oplus(\text{uper\_mask}(\text{MT}[i])\|\\ &\text{lower\_mask}(\text{MT}[i+1]))A\end{aligned} \tag{11}$$

式中，m 的取值为 397；$\|$表示将 MT[i] 的高 1 位和 MT[i+1] 的低 31 位组合，设组合后的数字为 x，则 x_A 的运算规则为：

$$x_A = \begin{cases} x>>1 & x_0=0 \\ (x>>1)\oplus a & x_0=1 \end{cases} \tag{12}$$

式中，a 的取值为 0x9908B0DF；x_0 表示的是该数的最低位。然后再经过式 (13) 的处理，得到一个伪随机数R'_m：

$$\begin{aligned} y &= x\oplus((x>>u)\&d)\\ y &= y\oplus((y<<s)\&b)\\ y &= y\oplus((y<<t)\&c)\\ R'_m &= y\oplus(y>>l)\end{aligned} \tag{13}$$

式中，令 x=MT[0]，(u, d)=(11, FFFFFFFF16)，(s,b)=(7, 9D2C568016)，(t, c)=(15, EFC6000016)，l = 18。最后将R'_m取模处理后得到 R_m：

$$R_m = R'_m \bmod |N| \quad 1\leqslant R_m\leqslant|N| \tag{14}$$

式中，$|N|$是指集合 N 中生产节点、投票节点、候补节点的数量总和。选取节点编号与 R_m 相同的节点成为主链节点，再由主链节点构建一条主链，完成主链上的事务处理。

4.2　主链共识

在从链的生产节点生成了随机数 R_m 后将之写进新生成的区块中，在每一周期最后一个从链区块产生后，集合 N 中所有编号与写入各区块中的 R_m 相同的节点成为代表节点，代表节点将自己所在从链中已确认的区块数据上传至主链网络中，随后参与共识并将主链区块保存至本地。为了保证主链上保存的从链区块数据都是真实完整、未被篡改的，代表节点在打包之前会检查被上传的从链区块数据的上传次数，只有被不少于其所在从链中一半

以上的代表节点上传过的从链区块数据才能被主链节点打包上链。当主链节点确保所打包信息都符合要求后，在主链上通过 PBFT 共识算法达成共识，完成信息在主链上的完整上链过程。

4.3 数据跨链

如图 1 所示，每一周期被选择成为主链节点的各从链代表节点 (P_1、P_2) 会将自己所在从链本周期产生的区块 (A_1、B_1、C_1、D_1) 上传至主链网络中，同时参与主链共识，在共识完成后各代表节点同样保存主链区块至本地网络中，每一个代表节点 (P_1、P_2) 保存的主链区块中都包含来自不同从链的区块数据 (A_2、B_2、A_3、B_3…) 供其所在从链其余节点 (A_1、B_1、C_1、D_1) 查询，从而完成不同从链之间的数据跨链。

5 攻击成本分析

5.1 权益粉碎攻击

在 PoS 网络中，拥有较低权益的节点挖出区块的可能性同样很低。另一方面，拥有较低权益的节点就更有可能去尝试分叉，因为在 PoS 网络中节点产生区块并不需要投入大量的算力等资源，即使分叉失败也只是损失较小的权益，而一旦攻击成功则能获取大量利益。在 PoVT 中，生产者只负责组装区块，还要经过投票节点的确认才能发布区块，且一个时隙内一个生产者最多只被允许产生一个区块，所以攻击者是无法发起权益粉碎攻击的。

5.2 自私挖矿攻击

自私挖矿的攻击者在挖出区块后选择不将挖出的区块发布出去，而是在已挖出的区块上继续挖出新的区块，当自己的秘密分叉超过主链的长度时，就将秘密分叉发布出来取代原来的主链成为新的最长链，使自己获得更多的收益。这种攻击不仅严重损害了原来主链上的诚实矿工的权利，而且还会造成数据丢失等危害区块链稳定性的情况。但在 PoVT 机制下，出块者是在生产节点中通过投票机制随机选择出来的，同时生产节点的选择则受到 STrust 值的影响，且生产节点如果不能在规定的时间内产生区块的话不仅得不到 STrust 值的奖励反而 STrust 值会下降，这样导致节点之后成为共识节点的概率降低，从而使攻击难以成功：

$$\lim_{n\to\infty}\frac{\mathrm{trust}_i^{h-1}+m}{\mathrm{trust}_i^{h-1}+n}=0 \tag{15}$$

式中，分母为节点正常出块的信用值；分子为节点自私挖矿时的信用值增长。$n=\sum_{s=1}^{n}(\gamma_1+\omega_1+\lambda_1-\theta_1)+\varepsilon_1$，而由于节点被检测出自私挖矿行为，$\gamma_2$=0，$\lambda_2<\lambda_1$，$\varepsilon_2<\varepsilon_1$，$\theta_2>\theta_1$，$m=\sum_{s=1}^{n}(\omega_2+\lambda_2-\theta_2)+\varepsilon_2$。节点的历史信用值$\mathrm{trust}_i^h$因为攻击行为而随着周期数的增加而下降，正常出块的节点则保持增加，因此自私挖矿攻击对于任何理性节点而言都是得不偿失的行为。

5.3 双花攻击

双花攻击是所有区块链网络都必须要解决的威胁。传统的双花攻击的步骤是：1) 攻击者发起一个交易；2) 在交易上链后，攻击者在该交易之前的一个区块上建立一个包含新交易的分叉；3) 而当分叉的长度超过原主链时，攻击者将其发布从而取代原主链成为新的最长链，此时原交易被新的交易所取代。而在主从多链分层跨链系统中，从链的区块数据不仅会被保存在从链节点上，还会通过代理节点上传至主链中，在经过主链共识后形成主链区块保存至主链节点中。若想发起双花攻击，不仅需要改变从链的区块信息，同时还要改变相应的主链节点上的区块信息，成本巨大，攻击难以实现。

6 仿真实验分析

实验在 Docker18.10 上搭建了一个从链基于 PoVT，主链基于 Fabric 的主从多链分层跨链原型系统，实验的操作系统为 Ubuntu18.04。实验的主要测试目的是测试该主从多链跨链分层共识机制的每秒交易处理速度、节点信用值增加以及节点作恶后信用值受到的惩罚。本文将实验的节点设置为 11 个，分别是 5 个投票节点、4 个生产节点、2 个候补节点，同时将一个周期划分为 10 个时隙，每个时隙经历出块到上链的全过程。将 STrust 的阈值设置为 30，当节点的 STrust 值低于 30 时，设定为节点必须要每隔 5 个周期才能参与以此共识。

6.1 信任度增长

为了能够更加灵活地调整系统节点的信用值增长，本文对参与信用值评价的参数都设置了权重。每一轮周期参与共识的节点以及节点参与共识所扮演的角色都不尽相同，所以选取了某一个节点在不同的权重下参与共识的信用值增长。

如图 3 所示，本文权重分别取值为 1、0.5、0.1。

可以得到，权重能对节点 STrust 值的增长有一个较好的调节，权重较高时，节点的 STrust 值能快速上升，最终达到设定的最大 STrust 值 300。同时，随着权重的减小，节点的 STrust 值的增长变得缓慢，但最终也能达到最大值。这保障了不同节点规模的系统的稳定性，当节点规模较小时，可以将权重加大，而在节点规模较大的网络则将权重降低，这样网络中的节点的 STrust 的增长速度能保持大致相同，不至于产生少数节点的 STrust 值增长过快从而危及系统安全性和稳定性的情况。

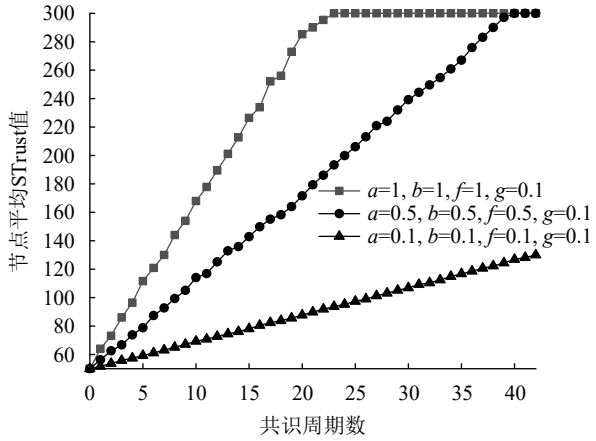

图 3　节点信用值增长

6.2　信任度惩罚

节点正常参与共识时 STrust 值会保持一个增长的状态，但如果节点在参与共识时有作恶行为，如投票节点未能诚实投票及出块节点打包的区块中包含被篡改的交易等情况时，节点会立即丧失本周期共识资格，同时在周期结束时受到 STrust 值降低的惩罚。如图 4 所示，Δ 线表示节点正常参与共识时 STrust 值的变化情况，可以看到在第 40 个周期时节点的 STrust 值达到最大值。*线在前 15 个周期时正常参与共识，而在第 15 个周期时节点开始作恶，这时节点的 Strust 值快速下降到 30 以下，此后节点被限制为每隔 5 个周期才能参与一次共识，此时若节点还是持续作恶，当 STrust 值降至 0 时节点将被永久剥夺参与共识的资格，而从●线可以看到节点在第 35 个周期时开始正常参与共识，但直到第 45 个周期才将 STrust值升至 30 以上，而此时正常参与共识的节点的 STrust 早已达到最大值，所以由此可以看出节点若作恶，将付出巨大的 STrust 值代价，从而影响节点参与共识的难度，是一种得不偿失的行为。

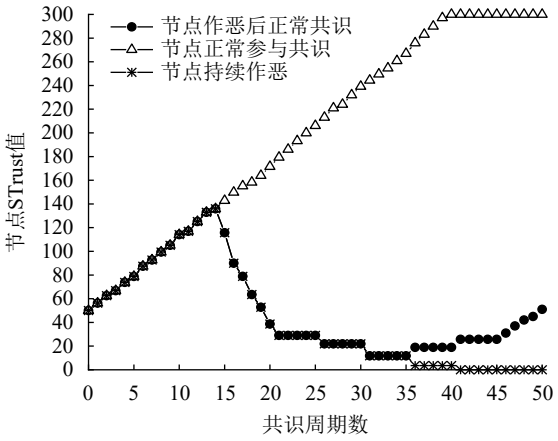

图 4　节点信任度惩罚对比

6.3　每秒交易处理量

每秒交易处理量 (transaction per second, TPS) 即用区块打包的交易数除以区块的生成时间，系统的每秒交易处理量能够衡量系统的交易处理性能。本文一共进行了 50 个周期共识，产生 500 个区块。实验中的以太坊网络采用标准的设置，平均每 15 s 生成一个区块，同时设置从链的 T_b 值为 10，即每 10 s 产生一个区块，实验结果如图 5 所示。在采用标准以太坊网络中的 TPS 平均值为 49，而在采用 PoVT 的单挑从链中的 TPS 在 60 左右波动，平均值为 63。而在采用 PBFT 共识的主链网络中的 TPS 在 45 左右波动，平均值为 47。可以看到采用 PoVT 的从链的交易处理速度是要优于以太坊网络的以及整个跨链模型的交易处理速度是比较理想的。

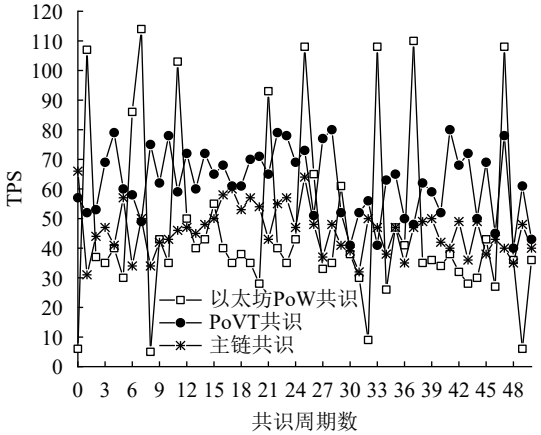

图 5　节点每秒交易处理量

7　结 束 语

近年来，区块链的发展取得了长足进步。但单一的共识机制仍然存在一定程度的缺陷，并且单链

形式已经不足以满足区块链的发展。本文提出了一种基于信用投票机制的共识算法，能够在避免算力竞争的条件下更加公平地分配节点的记账权，并且通过对节点赋予信用值来激励节点积极参与共识，同时也能对节点的恶意行为做出相应惩罚。但 PoVT 共识仍然存在一些不足，下一步的工作将针对信用值调节以及交易处理速度稳定性方面进行优化。

本文研究工作还得到网络与数据安全四川省重点实验室开放课题 (NDSMS201606) 的资助，在此表示感谢。

参 考 文 献

[1] LAMPORT L. The part-time parliament[J]. ACM Transactions on Computer Systems, 1998, 16(2): 133-169.
[2] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm[C]//The 2014 USENIX Conference on USENIX Annual Technical Conference. [S.l.]: USENIX Association, 2015: 305-320.
[3] LAMPORT L, SHOSTAK R, PEASE M. The Byzantine generals problem[J]. ACM Transactions on Programming Languages and Systems, 1982, 4(3): 382-401.
[4] LARIMER D. Transactions as proof-of-stake [EB/OL]. [2020-06-10]. https://bravenewcoin.com/assets/Uploads/TransactionsAsProofOfStake10.pdf.
[5] EYAL I. The miner's dilemma[C]//2015 IEEE Symposium on Security and Privacy. SanJose: IEEE, 2014: 89-103.
[6] HAN X, YUAN Y, WANG F Y. A fair blockchain based on proof of credit[J]. IEEE Transactions on Computational Social Systems, 2019, 6(5): 922-931.
[7] LEE S B, HWANG D Y, KIM J, et al. Proof of lottery: Design for block producing algorithm based on PoS for scalability[C]//International Conference on Information Networking(ICOIN). Barcelona: IEEE, 2020: 666-669.
[8] MATSUMOTO M, NISHIMURA T. Mersenne twister: A 623-dimensionally equidistributed uniform pseudo-random number generator[J]. ACM Transactions on Modeling and Computer Simulation(TOMACS), 1998, 8(1): 3-30.
[9] LI K J, LI H, HOU H X, et al. Proof of vote: A high-performance consensus protocol based on vote mechanism & consortium blockchain[C]//The 19th International Conference on High-Performance Computing and Communications. Bangkok: IEEE, 2017: 466-473.

编 辑 蒋 晓