



## 基于信誉的二阶段溯源区块链共识策略

汪 澍<sup>1,2</sup>, 许翀寰<sup>2</sup>, 汤中运<sup>3</sup>

(1. 浙江工商大学 管理工程与电子商务学院, 杭州 310018; 2. 浙江工商大学 工商管理学院, 杭州 310018;  
3. 杭州电子科技大学 计算机学院, 杭州 310018)

**摘要:** 基于区块链技术的溯源系统在现代供应链系统中被广泛应用, 溯源区块链适合采用联盟链来构建, 其参与利益方多、共识网络差异化高的特性影响了此类区块链系统的性能和安全性。对区块链共识过程进行分析, 构建模拟溯源区块链的系统模型和信誉模型, 以排除拜占庭故障节点。在此基础上, 设计包含代表选择和代表共识两个阶段的共识过程, 并提出一种基于信誉的二阶段溯源区块链共识策略 RTsBFT。实验结果表明, 在相同的配置环境和条件下, 相比 CSBFT 和 PBFT 策略, RTsBFT 可取得更高的系统吞吐量、更短的延迟和更低的故障节点率, 能够有效提高联盟链场景下溯源系统的性能和安全性。

**关键词:** 区块链; 共识策略; 拜占庭容错; 溯源系统; 信誉模型

开放科学(资源服务)标志码(OSID):



中文引用格式: 汪澍, 许翀寰, 汤中运. 基于信誉的二阶段溯源区块链共识策略[J]. 计算机工程, 2021, 47(7): 109-116.

英文引用格式: WANG S, XU C H, TANG Z Y. Reputation-based two-stage traceability blockchain consensus strategy[J]. Computer Engineering, 2021, 47(7): 109-116.

## Reputation-Based Two-Stage Traceability Blockchain Consensus Strategy

WANG Shu<sup>1,2</sup>, XU Chonghuan<sup>2</sup>, TANG Zhongyun<sup>3</sup>

(1. School of Management Engineering and E-Business, Zhejiang Gongshang University, Hangzhou 310018, China;  
2. School of Business Administration, Zhejiang Gongshang University, Hangzhou 310018, China;  
3. School of Computer Science and Technology, Hangzhou Dianzi University, Hangzhou 310018, China)

**[Abstract]** Being widely used in modern supply chain systems, blockchain-based traceability systems usually choose consortium blockchain for construction. However, such blockchain-based traceability systems usually have multiple stakeholders and highly heterogeneous consensus networks, leading to a reduction in their performance and security. This paper analyzes the consensus process of blockchain, and constructs a reputation model as well as a system model that simulates blockchain-based traceability to eliminate Byzantine faulty nodes. Then the paper proposes the design of a consensus mechanism that includes the representative selection stage and the representative consensus stage, and on this basis proposes a reputation-based two-stage consensus strategy named RTsBFT for blockchain-based traceability systems. Experimental results show that compared with the CSBFT and PBFT protocols, RTsBFT can improve the system throughput, and reduce the delay as well as failure rate of nodes with the same configuration, improving the performance and security of the traceability systems in the consortium blockchain scenarios.

**[Key words]** blockchain; consensus strategy; Byzantine fault tolerance; traceability system; reputation model

DOI: 10.19678/j.issn.1000-3428.0060808

### 0 概述

传统溯源系统采用将溯源信息集中存放在单一节点上的中心化方案<sup>[1]</sup>, 这会影响系统的安全性、透明度和互操作性<sup>[2]</sup>, 而区块链技术的出现为溯源系统的构建提供了更可信、安全和高效的解决方案<sup>[3]</sup>。

由于公有链中实体可随意加入或离开区块链网络, 而参与溯源系统中代表各利益方的实体则受到约束, 因此与比特币<sup>[4]</sup>等公有链不同, 溯源系统更适合采用联盟链。溯源区块链包含来自生产、加工、运输、存储和消费等领域中兴趣各异的利益方<sup>[5]</sup>, 这种复杂性对该联盟链的共识策略提出了拜占庭容错能

基金项目: 国家自然科学基金(71702164)。

作者简介: 汪 澍(1989—), 男, 讲师、博士, 主研方向为智能信息处理; 许翀寰, 副教授、博士; 汤中运, 博士研究生。

收稿日期: 2021-02-04 修回日期: 2021-03-09 E-mail: eragon@zjgsu.edu.cn

力的要求。在区块链技术中,共识策略能够确保在部分节点失效的情况下整个区块链网络中的节点仍能达成一致。由于节点通常由不同利益方维护,所构成的共识网络具有显著的异构特性,因此溯源区块链的性能开销较大,受到内外部恶意攻击的安全风险高,其共识策略还需要保障系统的性能和安全性。然而,传统基于证明的共识策略主要应用在公有链场景下,伴随巨大的计算开销和性能瓶颈,导致它们不适用于联盟链场景下的溯源区块链。基于拜占庭容错状态机复制的策略(BFT)<sup>[6]</sup>可以在不引入额外开销的情况下使全体非故障节点最终都达成共识,其中实用拜占庭容错(PBFT)是联盟链场景下应用最广泛的共识策略。但是,由于PBFT采用主节点(primary node)和多轮相互广播的方式,并且缺少故障节点排除机制<sup>[7]</sup>,导致其在溯源区块链场景下依然存在通信开销大、可靠性低、易受恶意攻击等问题。

本文提出一种基于信誉的二阶段溯源区块链共识策略RTsBFT。通过对溯源区块链进行分析,构建将共识节点分散在多个本地组中的系统模型,以模拟溯源系统多利益方、共识网络异构的特性。同时建立一个信誉模型,将信誉作为节点参与共识过程的凭据,并根据节点信誉来排除故障节点。在此基础上,将共识设计为一个包含代表选择和代表共识两个阶段的过程,实现异构网络通信。

## 1 背景知识

### 1.1 溯源区块链

区块链技术可以解决供应链系统中信任相关的问题<sup>[8]</sup>,溯源区块链能够通过信息获取、传输、分享等过程,为供应链生产、加工、仓储、销售等各环节提供可追溯性的可靠信息<sup>[9]</sup>。溯源区块链继承了区块链技术的众多特性,如提供一种安全可共享的去中心化数据库,为物品、数据、金融等资源的交易提供基于透明性和可追溯性的信任强化。随着射频识别(RFID)、近场通信(NFC)等技术的发展,溯源区块链被应用到众多领域<sup>[10]</sup>。溯源区块链可采用无权限(公有链)和有权限(联盟链、私有链)系统,但无权限系统在设计上允许任意实体随时加入和退出区块链系统,并且拥有任意的读写权限,因此需要花费巨大的代价来保障安全性,如采用资源浪费极为严重的基于证明的共识策略<sup>[11]</sup>。同时,无权限系统的事务吞吐量受其设计的限制,无法满足供应链溯源过程中的高频交易需求,而有权限系统则可以通过优化共识策略来实现性能的显著提升<sup>[12]</sup>,因此,共识策略设计对溯源区块链十分重要。

### 1.2 共识策略

共识策略由传统分布式系统中的一致性算法演化而来,一致性算法最初由PEASE等<sup>[13]</sup>提出,考虑

到分布式系统中可能存在故障节点,研究者又在共识策略中引入了拜占庭将军问题。1985年证明的FLP不可能性定理指出,对于具有错误过程的异步系统的共识问题,没有有限时间的理论解决方案,只能探索可行的“工程解”<sup>[14]</sup>。因此,早期探索的共识策略几乎都是非拜占庭容错的。直到2008年比特币被提出,拜占庭容错共识策略才得到广泛关注,这类策略大致上可分为基于证明和基于投票两类。比特币区块链中采用一种基于证明的共识策略——工作量证明(Proof of Work, PoW)<sup>[15]</sup>。但是, PoW是一种十分浪费算力的共识策略<sup>[16]</sup>,因此KING提出了权益证明(Proof of Stake, PoS)<sup>[17]</sup>机制来改善PoW。自提出PoW和PoS之后,研究者在不同的应用场景下提出了一系列基于证明的共识策略,如改进原始PoW和PoS的策略、结合PoW和PoS优点的策略<sup>[18]</sup>等,这类机制要求共识节点提供某种凭据以竞争区块打包和上链并获得收益,这类共识策略将区块链视作有限状态机,对于给定的顺序输入,它们能保证分布式系统最终的一致性,代价是牺牲系统的吞吐量和延迟性能。同时,由于上述共识策略的设计主要考虑公有链场景,因此并不适用于溯源区块链的联盟链场景。在基于投票的共识策略的每轮共识中,共识节点“相互投票”,将得到超过半数节点投票的节点选为打包区块上链的节点。通常基于投票的共识协议在分布式一致性算法中更为常见,如Paxos和Raft<sup>[19]</sup>,当前应用最为广泛的基于投票的共识策略是实用拜占庭容错(PBFT)策略。

### 1.3 PBFT策略

PBFT是一种典型的BFT状态机复制策略,它将分布式一致性算法的复杂性由指数时间级降低到多项式时间级。PBFT采用了主节点(Primary Node)和视图转换(View Change)的概念,主节点负责对请求和共识过程进行排序,视图转换则用于选择新的主节点。PBFT最多可容忍 $\lfloor (N-1)/3 \rfloor$ 个故障节点( $N$ 为总节点数)。当前对PBFT的研究主要集中在对其性能和安全性改进方面,多数研究从降低通信开销的角度来改进PBFT的性能<sup>[20]</sup>,如基于信誉的策略<sup>[21]</sup>和信誉监督的策略CSBFT<sup>[22]</sup>。在PBFT安全性改进方面,则有提高运行环境和网络基础设施安全性、增加故障检测及配备冗余主节点<sup>[23]</sup>的方案。

溯源区块链吸引着众多利益方,共识策略需要满足联盟链对性能和安全性需求<sup>[12]</sup>,溯源区块链的特性对共识策略的设计十分重要。

## 2 共识策略设计

不同于比特币的公有链,溯源区块链通常采用联盟链,涉及的各利益方均可执行业务功能和权限管理,并通过投资区块链节点、计算能力、带宽等,与

其他利益方竞争话语权,这形成了溯源区块链多利益方和共识网络异构的特性,引起了性能和拜占庭容错的需求。针对上述需求,本文提出基于信誉的两阶段拜占庭容错共识策略 RTsBFT,其主要包括系统建模、信誉建模及共识过程3个部分,其中,共识过程分为代表选择阶段和代表共识阶段。

## 2.1 系统建模

将溯源区块链中的节点分为3类,分别为事务节点、共识节点( $\text{Nodes}^{\text{con}} = \{\text{node}_1^{\text{con}}, \text{node}_2^{\text{con}}, \dots, \text{node}_n^{\text{con}}\}$ )以及存储节点。为了模拟溯源区块链多利益方、共识网络异构的特性,共识节点被分为  $S_{\text{seg}}$  个本地组  $G^{\text{local}} = \{g_1^{\text{local}}, g_2^{\text{local}}, \dots, g_{S_{\text{seg}}}^{\text{local}}\}$ , 在同一个本地组  $g_i^{\text{local}}$  中的节点性能相近并通过高速本地网络相连。事务节点产生事务并向各本地组广播,组内共识节点在其自身的事务队列中缓存待处理的事务,共识节点验证事务并打包区块。在本地组信誉值最高的节点中,完成上述任务最快的节点被选为该组的代表节点。共识将由所有本地组的代表节点参与,结果将分别发送到存储节点和共识节点,以进行永久存储和信誉更新。溯源区块链系统模型如图1所示。

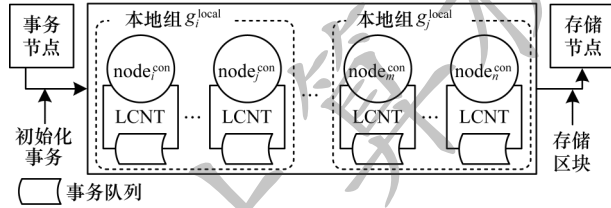


图1 溯源区块链系统模型

Fig.1 Traceability blockchain system model

当事务到达本地组时,组内所有共识节点会在进入共识过程前与一个可靠的NTP服务器同步时间。同一本地组中的每个共识节点都维护一个本地共识节点表  $\text{LCNT} = \langle \text{本地组标识}, \text{事务标识}, \text{节点标识}, \text{节点公钥}, \text{节点IP}, \text{端口对}, \text{节点信誉值}, \text{节点状态} \rangle$ , 整个系统中所有LCNT的初始配置都相同。

## 2.2 信誉模型

溯源区块链作为一种联盟链,比公有链更加可靠和安全,无需应用基于证明的共识策略以及增加额外的开销。但是,内部利益方和外部对手的恶意行为导致共识节点不能始终被视为值得信赖,从而需要考虑共识策略的拜占庭容错性。因此,本文建立一种信誉模型,以监督共识节点的行为,识别并排除有缺陷或恶意的故障节点,避免其影响共识过程。在该模型中,共识节点花费一定数量的自身信誉对共识过程中的提案进行签名,当代表共识阶段最终达成共识时,对共识提案签名的代表节点对应的本地组,可取回其在该轮共识中花费的信誉,这些本地组被标记为获胜组( $g_i^{\text{winning}} \in G^{\text{winning}}$ ),其代表被标记为获胜代表,其他本地组及其代表分别被标记为失

败组( $g_k^{\text{fail}} \in G^{\text{fail}}$ )和失败代表。所有签名为共识提案的节点均将取回其本轮花费的信誉,获胜组中签名共识提案的节点还将获得额外的信誉奖励,未签名共识提案的节点将受到信誉惩罚。

### 2.2.1 代表选择阶段的信誉变化

在系统初始化时,所有共识节点的信誉被设置为相同,对于共识节点  $\text{node}_i^{\text{con}}$ ,其初始信誉  $r_i = r^{\text{ini}}$ 。令  $L(X)$  表示  $X$  中的元素个数,则某一本地组  $g_i^{\text{local}}$  中节点的个数为  $L(g_i^{\text{local}})$ ,  $R(g_i^{\text{local}})$  表示  $g_i^{\text{local}}$  中所有节点在本轮共识中花费的信誉,其计算如下:

$$R(g_i^{\text{local}}) = \sum_{\text{node}_i^{\text{con}} \in g_i^{\text{local}}, r_i > T^{\text{rep}}} w^{\text{sep}} \quad (1)$$

其中,  $w^{\text{sep}}$  是共识节点签名一个提案花费的信誉值,  $T^{\text{rep}}$  是节点参与共识过程所需保有的最小信誉值。若  $r_i \leq T^{\text{rep}}$ , 则  $\text{node}_i^{\text{con}}$  将被视作为出现故障的拜占庭节点,无法签名任何新的提案,随后从本地组中移除或被新节点代替。在代表共识阶段前,共识节点  $\text{node}_i^{\text{con}}$  的信誉  $r_i$  的变化如下:

$$r_i = \begin{cases} r_i - w^{\text{sep}}, & r_i > T^{\text{rep}} \\ r_i, & r_i \leq T^{\text{rep}} \end{cases} \quad (2)$$

### 2.2.2 代表共识阶段的信誉变化

在代表共识阶段,本地组  $g_i^{\text{local}}$  的代表节点向其他本地组发送提案和  $R(g_i^{\text{local}})$  信息,该信息只能被其他本地组的代表节点获取。所有代表节点对提案进行验证,当最终达到共识时,获胜组将取回其花费的信誉,失败组所花费的信誉将用于获胜组奖励。

对于一个获胜组,其组内签名为共识提案的节点  $\text{Nodes}_{\text{sig}}^{\text{rep}}$  的信誉  $r_w$  的变化(包括获胜代表节点)如式(3)和式(4)所示,这些节点将取回它们花费的信誉,并从本组中未签名共识提案的节点  $\text{Nodes}_{\text{unsig}}^{\text{rep}}$  和其他失败组中获得一定的信誉奖励。

$$r_w = r_w + w^{\text{rep}} + R_{\text{RWD}} \quad (3)$$

$$R_{\text{RWD}} = \frac{\sum_{g_k^{\text{fail}} \in G^{\text{fail}}} R(g_k^{\text{fail}})}{L(G^{\text{winning}}) \times L(\text{Nodes}_{\text{sig}}^{\text{rep}})} + \frac{\sum_{\text{node}_u \in \text{Nodes}_{\text{unsig}}^{\text{rep}}} w^{\text{rep}}}{L(\text{Nodes}_{\text{sig}}^{\text{rep}})} \quad (4)$$

来自失败组的信誉首先平均分配给各获胜组,然后在每个获胜组内进行分配,  $R_{\text{RWD}}$  是给  $\text{Nodes}_{\text{sig}}^{\text{rep}}$  的信誉奖励,其由从失败组以及  $\text{Nodes}_{\text{unsig}}^{\text{rep}}$  花费的信誉中分配而来的信誉共同构成,  $\text{Nodes}_{\text{unsig}}^{\text{rep}}$  的信誉则维持本轮花费信誉后的值不变。

对于一个失败组,其组内共有3类节点,分别是失败代表本身  $\text{node}_i^{\text{rep}}$ 、未签名共识提案的节点  $\text{Nodes}_{\text{unsig}}^{\text{fail}} = \{\text{node}_1^{\text{fail}}, \text{node}_2^{\text{fail}}, \dots, \text{node}_u^{\text{fail}}\}$  以及签名共识提案的节点  $\text{Nodes}_{\text{sig}}^{\text{fail}} = \{\text{node}_1^{\text{sig}}, \text{node}_2^{\text{sig}}, \dots, \text{node}_s^{\text{sig}}\}$ , 分别用  $r_f$ 、 $r_u$  和  $r_s$  表示这3类节点的信誉,则其变化分别如式(5)~式(7)所示:



$$r_f = \begin{cases} r_f, r_f \leq T^{\text{rep}} \\ r_f - L(\text{Nodes}_{\text{sig}}^{\text{fail}}) \times w^{\text{rep}}, r_f > T^{\text{rep}} \end{cases} \quad (5)$$

$$r_u = r_u \quad (6)$$

$$r_s = \begin{cases} r_s, r_s \leq T^{\text{rep}} \\ r_s + w^{\text{rep}}, r_s > T^{\text{rep}} \end{cases} \quad (7)$$

若失败组中有节点签名了共识提案,则它们应该被视作比同组中其他未签名共识提案的节点更可靠,失败代表应该为这些节点的信誉损失负责,并从它们的信誉值中减去这些节点花费的信誉,以补偿这些节点的信誉损失,其他节点则不会得到任何形式的信誉补偿。

### 2.3 共识过程

基于PBFT策略中的主节点在共识过程中发挥着重要作用,但存在引入拜占庭节点作为主节点的风险,从而对系统性能造成影响。RTsBFT采用代表节点机制,每个代表节点仅对其本地组负责,一个代表节点的错误不会影响到整个共识网络。对于一个有  $S_{\text{seg}}$  个本地组的共识网络,只要该系统中还有  $2\left\lfloor \frac{S_{\text{seg}} - 1}{3} \right\rfloor + 1$  个正常代表节点,RTsBFT就可保证该系统的正常运作。整个共识过程在本地组内和组间分别进行,故可将共识过程分为代表选择和代表共识两个阶段。

#### 2.3.1 代表选择阶段

在代表选择阶段,事务被共识节点接收,并产生代表节点。RTsBFT中采用固定区块大小的方法来生成区块,当共识节点收到用于生成新区块的最后一个事务时,它与NTP服务器同步时钟,并对区块进行验证、摘要和签名。当完成上述工作且未收到来自其他节点的代表声名信息  $M^{\text{rep}}$  时,节点向本地组内广播包含自身节点id、完成签名时间、区块头、区块id的代表声名信息  $M^{\text{rep}} = \{N_{\text{id}}, T_{\text{sign}}, H_{\text{block}}, B_{\text{id}}\}$ 。若在发出自身  $M^{\text{rep}}$  前接收到其他节点  $M^{\text{rep}}$  的节点,则切换自身状态为INHIBITED,并不再发出  $M^{\text{rep}}$ 。所有节点将接收到的  $M^{\text{rep}}$  按照信息对应的签名完成时间和LCNT中的信誉值进行排序,从最高信誉值的节点中选出完成时间最早的节点作为代表节点。所有信誉高于  $T^{\text{rep}}$  的节点对自身提案签名,并根据信誉模型更新LCNT中节点的信誉,将信誉高于  $T^{\text{rep}}$  的节点状态标记为FUNCTION,其他节点被标记为MALFUNCTION,它们被禁止参与下一轮共识。代表节点在签名自身提案后,将  $R(g_i^{\text{local}})$  与提案信息一起发送给其他本地组。代表选择阶段可分为Broadcast、Claim、Preprepare 3个流程,如图2所示。

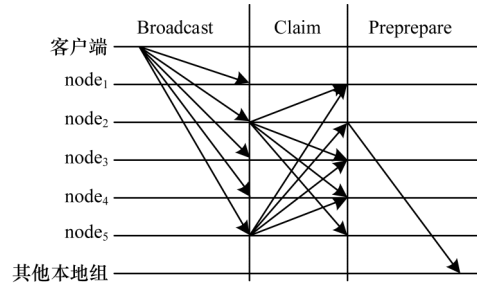


图2 代表选择阶段的共识过程

Fig.2 Consensus process of representative selection stage

代表选择阶段的详细步骤如下:

**步骤1** 进入Broadcast流程,用于生成新区块  $B_b$  的最后一个事务  $\text{Tran}_f^b$  被广播到本地组  $g_i^{\text{local}}$  中。

**步骤2**  $g_i^{\text{local}}$  中的共识节点验证  $\text{Tran}_f^b$ , 若验证错误,则回滚事务;若验证通过,则:

- 1) 与NTP服务器同步时钟。
- 2) 将  $\text{Tran}_f^b$  打包进  $B_b$ 。
- 3) 对区块进行摘要和签名。
- 4) 生成  $B_b$  的区块头  $H_{\text{block}}$ 。
- 5) 构造代表声名信息  $M^{\text{rep}}$ 。

**步骤3** 进入Claim流程,若节点未完成步骤1和步骤2且收到其他节点  $M^{\text{rep}}$ ,则将状态标记为INHIBITED;若完成步骤1和步骤2且未收到其他节点  $M^{\text{rep}}$ ,则进行信誉检查:

- 1) 若其信誉低于  $T^{\text{rep}}$  或其状态为INHIBITED,则不进行任何操作。
- 2) 若其信誉高于  $T^{\text{rep}}$  且状态为FUNCTION,则广播其  $M^{\text{rep}}$ 。

**步骤4** 进入Preprepare流程,在  $T^{\text{out}}$  时长后进行代表节点排序:

- 1) 从LCNT中检查发送过  $M^{\text{rep}}$  节点的信誉。
- 2) 根据信誉对上述节点进行排序。
- 3) 仅当多个节点信誉值相同时对其按照  $T_{\text{sign}}$  进行排序。
- 4) 将排序在第1位的节点状态标记为REPRESENT,选其作为代表节点。

**步骤5** 代表节点根据信誉模型计算  $R(g_i^{\text{local}})$ ,并向其他本地组发送提案和  $R(g_i^{\text{local}})$ ,所有节点根据信誉模型更新LCNT,将信誉值低于  $T^{\text{rep}}$  的节点状态标记为MALFUNCTION。

#### 2.3.2 代表共识阶段

与PBFT不同,RTsBFT中提案信息只会被本地组的代表节点接收,因此,共识过程会被限定在所有代表节点中进行。代表节点发送含有本地组id、代表节点信息、提案(含区块头)、本组  $R(g_i^{\text{local}})$  等内容

的消息  $M_{g_i}^{\text{prop}} = \{g_{\text{id}}, \text{INFO}^{\text{rep}}, \text{PROP}, R(g_i^{\text{local}})\}$ 。代表节点收到  $M_{g_i}^{\text{prop}}$  后,对内容进行验证,根据  $M_{g_i}^{\text{prop}}$  中的提案与自身提案是否相同,将该消息标记为 MATCH 或 MISMATCH,当 MATCH 消息的数量超过  $\left\lceil \frac{S_{\text{seg}}}{2} \right\rceil$  时,达成共识,相应提案为共识提案,并向所有代表节点提交该提案。签名共识提案的代表节点将提案相应的区块提交给存储节点,并同时开始反馈流程,共识提案被广播到各本地组,节点根据信誉模型更新信誉。代表共识阶段包含 Prepare、Commit、Store & Feedback 这3个流程,如图3所示。

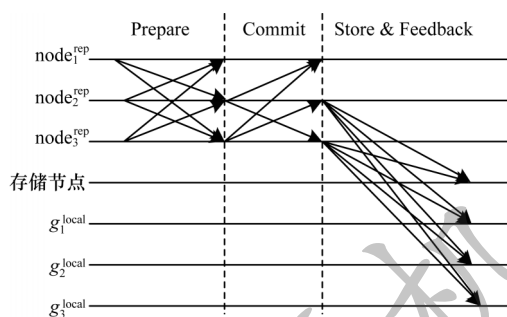


图3 代表共识阶段的共识过程

Fig.3 Consensus process of representative consensus stage

代表共识阶段的详细步骤如下:

**步骤1** 所有代表节点向其他本地组广播  $M_{g_i}^{\text{prop}}$ ,并由各本地组代表节点接收。

**步骤2** 进入 Prepare 流程,所有代表节点验证收到的  $M_{g_i}^{\text{prop}}$ ,若消息中的提案与本节点提案一致,则将其存入 MATCH 栈中;否则,存入 MISMATCH 栈中。

**步骤3** 进入 Commit 流程,收到 MATCH 消息多于  $\left\lceil \frac{S_{\text{seg}}}{2} \right\rceil$  的节点,根据  $\text{INFO}^{\text{rep}}$  中其他代表节点的信息将共识提案提交给其他代表节点。

**步骤4** 进入 Store & Feedback 流程,获胜代表节点将提案对应的区块发送给存储节点,并将共识提案广播给所有本地组,在各本地组内节点更新信誉值。

## 2.4 正确性分析

溯源区块链中的拜占庭节点不可靠,可能会通过恶意行为来阻碍共识的达成,RTsBFT 用信誉模型和二阶段过程设计识别这些节点,以避免拜占庭故障节点成为代表节点,从而保障共识过程的安全性。

**引理1** 在本地组中,正确节点的信誉始终高于故障节点的信誉。

**证明** 令  $\Phi > 0$  为初始信誉值,则在初始时对  $\forall \text{node}_i, C^{\text{rep}} = F^{\text{rep}} = \Phi$ ,其中,  $C^{\text{rep}}$  和  $F^{\text{rep}}$  分别为正确节点和故障节点的信誉。令签名提案信誉花费为  $\Delta > 0$ ,信誉奖励为  $\delta > 0$ ,信誉惩罚为  $\theta > 0$ ,则有:

1)若节点为  $\text{Nodes}_{\text{sig}}^{\text{fail}}$ ,则  $C^{\text{rep}} = \Phi - \Delta + \Delta = \Phi$ ;若节点为  $\text{Nodes}_{\text{sig}}^{\text{rep}}$ ,则  $C^{\text{rep}} = \Phi - \Delta + \Delta + \delta > \Phi$ 。因此,  $C^{\text{rep}} \geq \Phi$ 。

2)若节点为  $\text{Nodes}_{\text{unsig}}^{\text{fail}}$ ,则  $F^{\text{rep}} = F^{\text{rep}} - \Delta < \Phi$ ;若节点为  $\text{Nodes}_{\text{unsig}}^{\text{rep}}$ ,则  $F^{\text{rep}} = F^{\text{rep}} - \Delta - \theta < \Phi$ 。因此,  $F^{\text{rep}} < \Phi$ 。

综上所述,  $C^{\text{rep}} > F^{\text{rep}}$ ,即正确节点的信誉始终高于故障节点的信誉,因此,可以避免故障节点成为代表节点。

**引理2** 只要代表节点中的故障节点不超过  $\left\lfloor (S_{\text{seg}} - 1)/3 \right\rfloor$  个,RTsBFT 可保证运行于本文系统模型上的溯源区块链的安全性。

**证明**  $N^c$  和  $N^f$  分别表示正确代表节点和故障代表节点的个数。由  $N^f \leq \left\lfloor (S_{\text{seg}} - 1)/3 \right\rfloor$  可得,  $N^c \geq 2N^f + 1$ 。当故障节点均匀分布于所有本地组时,根据引理1,故障节点无法成为代表节点,  $N^f = 0$ ,故障节点提案不会进入 prepare 流程,因此,不会对最终共识的安全性产生影响。当故障节点非均匀分布时,最坏情况下  $\left\lfloor (S_{\text{seg}} - 1)/3 \right\rfloor$  个本地组中全部节点均为故障节点,根据引理1,此时  $N^c \geq 2N^f + 1$  依然成立,RTsBFT 在节点共识阶段并没有放宽 PBFT 中的限制,而 PBFT 已被证明在该条件下可以保证系统的安全性<sup>[24]</sup>。因此,RTsBFT 均可保证运行于本文系统模型上溯源区块链的安全性。

## 3 性能评估

本文构建一个模拟溯源区块链特性的原型系统,并在相同条件下运行 RTsBFT、PBFT 和 CSBFT,以验证本文 RTsBFT 模型的性能。

### 3.1 原型系统构建

本文构建一个简单的联盟链原型系统,包含一个事务模块和一个共识模块,原型系统架构如图4所示。事务模块生成事务,并将事务通过组间链路广播至共识模块的本地组内,该模块生成的事务大小一致(256 B),以确保达到 RTsBFT 设计中固定区块大小的要求。共识模块通过分别运行这些共识策略来监测其性能。为了模拟溯源区块链的共识节点分布于受不同利益方管理的异构网络的特性,将同一本地组内的节点用高速本地链路连接,各本地组间则通过相对低速的组间链路连接。整个原型系统的共识模块由10台多核计算机组成(Intel Core i5-9500, 32 GB 内存),每台计算机作为一个本地组宿主,在其上运行整个本地组的节点。组间链路由一个1 000 Mb/s 的交换网络实现,高速本地链路则由 Hypervisor 的 system bus 实现。

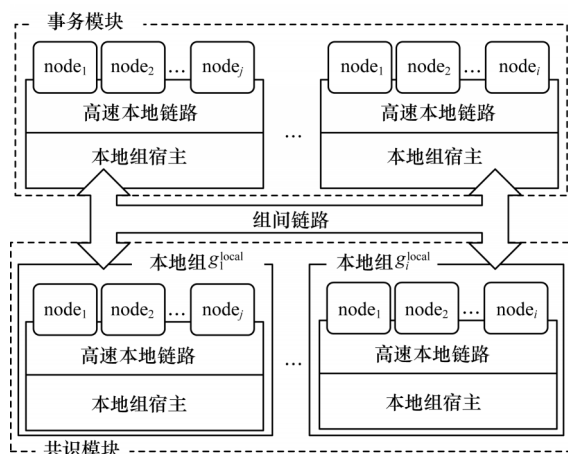


图4 原型系统架构

Fig.4 Prototype system architecture

### 3.2 评估指标

溯源区块链的特点对共识策略的性能和安全性提出了更高的要求。本文通过吞吐量、延迟和故障节点率3个指标<sup>[25]</sup>来评估共识策略的性能和安全性。

1) 吞吐量: 衡量共识策略在给定时间内处理的事务数量, 用每秒事务处理量 (Transaction Per Second, TPS) 来表示, TPS 越高, 系统性能越好。

2) 延迟: 从一个区块被生成到它的共识过程完成所需要的时间, 由新生成区块的时刻  $T_{\text{gen}}$  到共识达成的时刻  $T_{\text{con}}$  的时间差表示,  $\text{Delay} = T_{\text{con}} - T_{\text{gen}}$ , 延迟越低表示系统性能越高。

3) 故障节点率: 当前共识网络中故障节点的数量与总节点数量的比值, 故障节点率反映当前区块链系统共识策略的安全性和可靠性, 该指标越低, 系统安全性和可靠性越高。

由于 RTsBFT 采用固定区块大小的策略, 区块生成时间不可控, 因此可以考察区块大小对系统性能的影响。同时, 节点数量决定共识网络的复杂性, 本文将在不同区块大小和节点数量情况下对比 RTsBFT、PBFT 和 CSBFT 的吞吐量、延迟以及故障节点率。

### 3.3 吞吐量和延迟对比

本文分别设计 0.5 KB、1.0 KB、2.0 KB、4.0 KB 和 8.0 KB 这 5 个不同的区块大小, 原型系统配置为 5 个本地组宿主, 每宿主运行 10 个节点。评估指标取运行中某一段 20 s 时间内的平均值, 结果如图 5 所示。从图 5 可以看出, 在相同区块大小下, RTsBFT 的吞吐量总是高于 PBFT 和 CSBFT, 延迟低于 PBFT。随着区块大小的增加, 所有共识策略的吞吐量均有下降, 延迟均有升高, 说明区块大小会影响共识策略的性能。

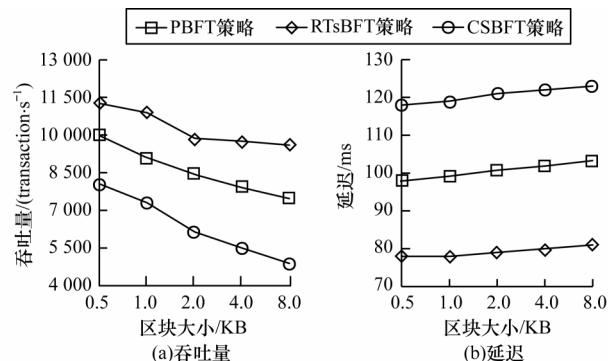


图5 不同区块大小下吞吐量和延迟的对比

Fig.5 Comparison of throughput and latency under different block sizes

将原型系统分别配置为运行 35 个、70 个、105 个和 140 个节点, 这些节点以两种方式分别分配到 5 个 (每宿主运行 7 个、14 个、21 个和 28 个节点) 和 7 个 (每宿主运行 5 个、10 个、15 个和 20 个节点) 本地组宿主上。区块大小配置为 1 KB, 评估指标也取运行中某一段 20 s 时间内的平均值, 结果如图 6 所示。从图 6 可以看出, 在不同节点数量和分组方式下, RTsBFT 的吞吐量均高于 PBFT 和 CSBFT, 延迟均低于 PBFT 和 CSBFT, 节点数量的增加降低了共识策略的吞吐量, 但 PBFT 和 CSBFT 的吞吐量下降更为显著, 同时, 节点数量的增加提高了共识策略的延迟, 但 PBFT 和 CSBFT 的延迟提高幅度更大。

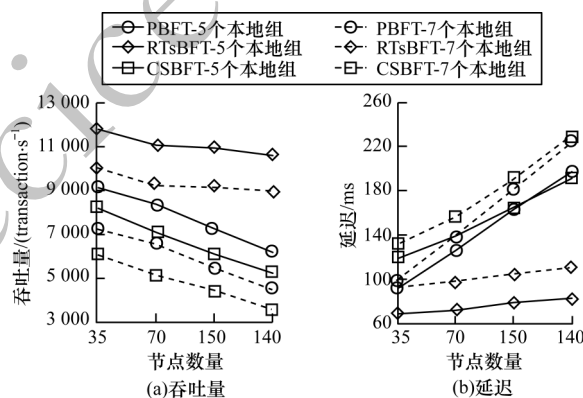


图6 不同节点数量下吞吐量和延迟的对比

Fig.6 Comparison of throughput and latency under different numbers of nodes

在相同区块大小和相同节点数量这两种情况下, RTsBFT 在吞吐量和延迟上的性能表现均优于 PBFT 和 CSBFT。尽管随着区块大小的增加和节点数量的增加, 各策略性能均有一定程度的下降, 但 RTsBFT 下降得更少。因此, RTsBFT 具有更高的性能, 且在系统复杂性增加的情况下能够更好地控制性能下降。研究表明, 通信复杂性对共识策略的性能有重要影响<sup>[26]</sup>, 在 RTsBFT 的设计中, 本地组内的通信发生在代表选择阶段, 组间通信只发生在代表



共识阶段,同时共识的核心过程仅在少数代表节点中发生,降低了节点间通信开销,特别是组间节点的通信开销。而在两阶段的共识过程中,不需要向所有节点广播区块并等待它们的响应,这与引入的信誉模型一起简化了共识过程,同时避免了故障节点带来的额外开销,降低了共识过程的复杂性。在不同的节点分组方式中,RTsBFT的延迟差异几乎是恒定的,而PBFT和CSBFT在不同节点分组下的延迟差异随着节点数量的增加而提高,这也证明对通信开销的控制给RTsBFT带来了性能提升。

### 3.4 性能与安全性随时间的变化情况

故障节点引入了拜占庭错误,本文对比2个共识策略在开始运行20 s内的吞吐量、延迟和故障节点率的变化情况。原型系统配置为在5个本地组宿主上运行50个共识节点,区块大小为1 KB,初始故障节点为12个。该过程中3个共识策略的吞吐量和延迟随时间变化的统计信息如表1所示,变化趋势如图7所示。

表1 3个共识策略吞吐量和延迟随时间变化的统计信息

Table 1 Statistics of the throughput and latency of three consensus strategies over time

策略	吞吐量/(transaction·s <sup>-1</sup> )			延迟/ms		
	最大值	最小值	均值 (标准差)	最大值	最小值	均值 (标准差)
RTsBFT	11 054	10 737	10 893(96)	91	72	80(5.31)
PBFT	9 377	8 654	9 079(183)	115	71	98(12.45)
CSBFT	8 271	7 634	8 005(180)	134	106	120(8.33)

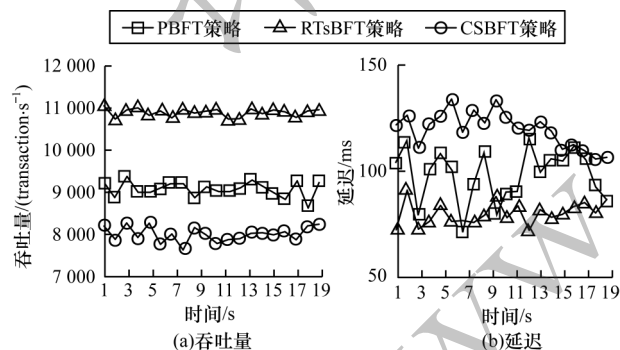


图7 吞吐量和延迟随时间的变化情况

Fig.7 The change of throughput and latency with time

从表1和图7可以看出,RTsBFT的吞吐量随时间变化的均值高于另两个策略,延迟低于另两个策略,且RTsBFT的吞吐量和延迟随时间变化的波动比另两个策略平缓。

3个共识策略故障节点率随时间的变化趋势如图8所示,可以看出,PBFT的故障节点率保持在24%左右,而RTsBFT几乎在前5 s就将所有故障节点排除在外,CSBFT大约在10 s完成故障节点的排除。RTsBFT具有排除故障节点的能力,因此相比PBFT,

其具有更好的安全性和可靠性,相比CSBFT的故障节点排除能力更强。RTsBFT在性能和安全性波动上的优势来自于其设计中的信誉模型所带来的排除拜占庭故障节点的能力,故障节点会加重系统的负载,从而对系统性能和安全性产生不利影响。RTsBFT能消除故障节点恶意行为对共识过程的影响,减少共识过程中需要通信的节点数量,从而提高系统性能和安全性。而CSBFT在没有摆脱PBFT主节点和视图转换弊端的情况下引入信誉模型,虽然具备一定的故障节点排除能力,但增加了大量额外开销<sup>[21]</sup>,因此,其性能较PBFT差。

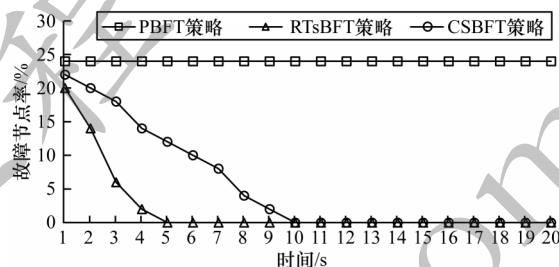


图8 故障节点率随时间的变化情况

Fig.8 The change of failure node rate with time

## 4 结束语

本文提出一种基于信誉的二阶段溯源区块链共识策略RTsBFT,在多利益方、共识网络异构的联盟链场景下实现对溯源区块链性能和安全性提升。通过建立溯源区块链系统模型与信誉模型及设计二阶段共识过程,使得RTsBFT具有排除拜占庭故障节点的能力并降低节点通信开销。实验结果表明,在不同区块大小和节点数量的情况下,RTsBFT的吞吐量、延迟和故障节点率均优于PBFT和CSBFT策略。下一步将对RTsBFT中的信誉模型进行优化,使其适用于更多类型的联盟链系统模型架构。

### 参考文献

- [1] KHAN M A, SALAH K. IoT security: review, blockchain solutions, and open challenges [J]. Future Generation Computer Systems, 2018, 82: 395-411.
- [2] HELO P, HAO Y. Blockchains in operations and supply chains: a model and reference implementation [J]. Computers & Industrial Engineering, 2019, 136: 242-251.
- [3] KAMBLE S S, GUNASEKARAN A, SHARMA R. Modeling the blockchain enabled traceability in agriculture supply chain [J]. International Journal of Information Management, 2020, 52: 101967-101969.
- [4] NAKAMOTO S, BITCOIN A. A peer-to-peer electronic cash system [EB/OL]. [2021-01-05]. <http://ondoc.logand.com/d/4354/pdf>.
- [5] PAPPA I C, ILIOPOULOS C, MASSOURAS T. What determines the acceptance and use of electronic traceability systems in agri-food supply chains? [J]. Journal of Rural

- Studies, 2018, 58: 123-135.
- [ 6 ] SOUSA J, BESSANI A. From Byzantine consensus to BFT state machine replication: a latency-optimal transformation [C]//Proceedings of 2012 European Dependable Computing Conference. Washington D. C. , USA: IEEE Press, 2012: 37-48.
- [ 7 ] JIANG Y J, LIAN Z. High performance and scalable Byzantine fault tolerance [C]//Proceedings of 2019 IEEE Information Technology, Networking, Electronic and Automation Control Conference. Washington D. C. , USA: IEEE Press, 2019: 1195-1202.
- [ 8 ] XU X W, LU Q H, LIU Y, et al. Designing blockchain-based applications a case study for imported product traceability [J]. Future Generation Computer Systems, 2019, 92: 399-406.
- [ 9 ] 张亮, 刘百祥, 张如意, 等. 区块链技术综述 [J]. 计算机工程, 2019, 45(5): 1-12.
- ZHANG L, LIU B X, ZHANG R Y, et al. Overview of blockchain technology [J]. Computer Engineering, 2019, 45(5): 1-12. (in Chinese)
- [ 10 ] 伏阳阳, 梁子婧. 基于区块链的护肤品跨境电商物流溯源体系研究 [J]. 中国商论, 2018(14): 7-9.
- FU Y Y, LIANG Z J. Blockchain based trace-ability system of cross-border skin care product e-commerce logistics [J]. China Business & Trade, 2018(14): 7-9. (in Chinese)
- [ 11 ] EYAL I, GENCER A E, SIRER E G, et al. Bitcoin-NG: a scalable blockchain protocol [EB/OL]. [2021-01-05]. <https://www.usenix.org/system/files/conference/nsdi16/nsdi16-paper-eyal.pdf>.
- [ 12 ] DEMESTICHAS K, PEPPE N, ALEXAKIS T, et al. Blockchain in agriculture traceability systems: a review [J]. Applied Sciences, 2020, 10(12): 4113-4115.
- [ 13 ] PEASE M, SHOSTAK R, LAMPORT L. Reaching agreement in the presence of faults [J]. Journal of the ACM, 1980, 27(2): 228-234.
- [ 14 ] FISCHER M J, LYNCH N A, PATERSON M S. Impossibility of distributed consensus with one faulty process [J]. Journal of the ACM, 1985, 32(2): 374-382.
- [ 15 ] JAKOBSSON M, JUELS A. Proofs of work and bread pudding protocols (extended abstract) [EB/OL]. [2021-01-05]. <http://www.arijueles.com/wp-content/uploads/2013/09/PoW.pdf>.
- [ 16 ] ULLRICH J, STIFTER N, JUDMAYER A, et al. Proof-of-blackouts? How proof-of-work cryptocurrencies could affect power grids [C]//Proceedings of International Symposium on Research in Attacks, Intrusions, and Defenses. Berlin, Germany: Springer, 2018: 184-203.
- [ 17 ] KING S, NADAL S. PPcoin: peer-to-peer cryptocurrency with proof-of-stake [EB/OL]. [2021-01-05]. <http://www.sysu.edu.cn/403.html>.
- [ 18 ] DUONG T, FAN L, KATZ J, et al. 2-hop blockchain: combining proof-of-work and proof-of-stake securely [C]//Proceedings of European Symposium on Research in Computer Security. Berlin, Germany: Springer, 2020: 697-712.
- [ 19 ] 王日宏, 张立峰, 周航, 等. 一种结合 BLS 签名的可拜占庭容错 Raft 算法 [J]. 应用科学学报, 2020, 38(1): 93-104.
- WANG R H, ZHANG L F, ZHOU H, et al. A Byzantine fault tolerance Raft algorithm combines with BLS signature [J]. Journal of Applied Sciences, 2020, 38(1): 93-104. (in Chinese)
- [ 20 ] BACH L, MIHALJEVIC B, ZAGAR M. Comparative analysis of blockchain consensus algorithms [C]//Proceedings of 2018 International Convention on Information and Communication Technology, Electronics and Microelectronics. Washington D. C. , USA: IEEE Press, 2018: 1545-1550.
- [ 21 ] LEI K, ZHANG Q C, XU L M, et al. Reputation-based Byzantine fault-tolerance for consortium block-chain [C]//Proceedings of 2018 IEEE International Conference on Parallel and Distributed Systems. Washington D. C. , USA: IEEE Press, 2018: 604-611.
- [ 22 ] 任守纲, 何自明, 周正己, 等. 基于 CSBFT 区块链的农作物全产业链信息溯源平台设计 [J]. 农业工程学报, 2020, 36(3): 279-286.
- REN S G, HE Z M, ZHOU Z J, et al. Design and implementation of information tracing platform for crop whole industry chain based on CSBFT-blockchain [J]. Chinese Society of Agricultural Engineering, 2020, 36(3): 279-286. (in Chinese)
- [ 23 ] AUBLIN P L, MOKHTAR S B, QUEMA V. RBFT: redundant Byzantine fault tolerance [C]//Proceedings of 2013 IEEE International Conference on Distributed Computing Systems. Washington D. C. , USA: IEEE Press, 2013: 297-306.
- [ 24 ] CASTRO M, LISKOV B. Practical Byzantine fault tolerance and proactive recovery [J]. ACM Transactions on Computer Systems, 2002, 20(4): 398-461.
- [ 25 ] BAMAKAN S M H, MOTAVALI A, BONDARTI A B. A survey of blockchain consensus algorithms performance evaluation criteria [J]. Expert Systems with Applications, 2020, 154: 113385-113390.
- [ 26 ] HAO Y, LI Y, DONG X H, et al. Performance analysis of consensus algorithm in private blockchain [C]//Proceedings of 2018 IEEE Intelligent Vehicles Symposium. Washington D. C. , USA: IEEE Press, 2018: 280-285.

编辑 吴云芳 薛晋栋