·区块链理论与技术·　　文章编号：1000-3428(2022)06-0042-08　　文献标志码：A　　中图分类号：TP311

# 基于信誉值投票与随机数选举的PBFT共识算法

陈润宇，王伦文，朱然刚

（国防科技大学 电子对抗学院，合肥 230037）

**摘　要：**实用拜占庭容错(PBFT)算法在Raft和Paxos共识算法的基础上，解决了分布式系统中恶意节点向其他节点发送错误消息以扰乱系统正常运行的问题，但PBFT算法由于主节点选举随意导致共识效率低下，而现有PBFT改进算法普遍通信复杂度较高且容易出现系统集中化趋势。针对上述问题，提出一种基于信誉值投票与随机数选举的RN-VPBFT共识算法。通过增设监督节点，实现权力分散和信息中转，保证系统安全运行。在投票确定初始信誉值的过程中，引入随机参数使得满足条件的节点均有机会当选主节点，缓解系统集中化趋势。建立节点动态信誉模型，区分系统中的诚实节点与恶意节点，简化共识算法的一致性协议，降低算法通信复杂度。实验结果表明，与PBFT算法和基于信誉投票的PBFT改进算法相比，RN-VPBFT算法将通信复杂度由$O(N^2)$降至$O(N)$，并且所有诚实节点的信誉值之差仅为0.02，具有更低的通信复杂度及更好的去中心化特性。

**关键词：**区块链；共识算法；实用拜占庭容错算法；信誉值投票；随机数选举

**开放科学（资源服务）标志码（OSID）：**

**中文引用格式：**陈润宇，王伦文，朱然刚.基于信誉值投票与随机数选举的PBFT共识算法[J].计算机工程，2022，48(6)：42-49，56.

**英文引用格式：**CHEN R Y，WANG L W，ZHU R G.PBFT consensus algorithm based on reputation value voting and random number election[J].Computer Engineering，2022，48(6)：42-49，56.

# PBFT Consensus Algorithm Based on Reputation Value Voting and Random Number Election

CHEN Runyu，WANG Lunwen，ZHU Rangang

（Institute of Electronic Countermeasure，National University of Defense Technology，Hefei 230037，China）

**【Abstract】**Compared with Raft and Paxos，the original consensus algorithm，i.e.，the Practical Byzantine Fault Tolerant（PBFT）algorithm，is mainly aimed at malicious nodes in a distributed system used to send error messages to other nodes，disturbing the normal operation of the entire system.This algorithm fundamentally solves the Byzantine problem of the system.However，the PBFT algorithm itself has the problem of a low consensus efficiency caused by an arbitrary primary node election.Meanwhile，existing improved algorithms generally have problems of a high communication complexity and centralization tendency.Aiming at these problems，an improved consensus algorithm，RN-VPBFT，is proposed based on reputation value voting and random number election.First，by adding supervision nodes，the algorithm realizes a decentralization of rights and an information transfer and ensures the safe operation of the system.A random parameter is then introduced during the process of voting to determine the initial reputation value，and thus the reputation value is no longer the only criterion for selecting the master node.All nodes that meet certain conditions have an equal chance to be selected as the master node.This can reduce the trend of system centralization.Finally，the algorithm establishes the dynamic reputation model of the nodes to distinguish between honest and malicious nodes in the system，simplify the consistency protocol of the consensus algorithm，and reduce the communication complexity of the entire algorithm.Experimental results show that，compared with the original PBFT algorithm and the existing improved PBFT algorithm based on reputation voting，the RN-VPBFT algorithm can reduce the communication complexity from $O(N^2)$ to $O(N)$.The final difference in reputation of all honesty nodes is only 0.02，which is almost negligible.Therefore，the proposed algorithm has a lower communication complexity and better decentralization.

**【Key words】**blockchain；consensus algorithm；Practical Byzantine Fault Tolerant（PBFT）algorithm；reputation value voting；random number election

**DOI：**10. 19678/j. issn. 1000-3428. 0063904

**基金项目：**国家自然科学基金“图像渐进式秘密分享评价体系和算法研究”（61602491）。

**作者简介：**陈润宇（1998—），男，硕士研究生，主研方向为区块链共识机制；王伦文，教授；朱然刚，副教授。

**收稿日期：**2022-02-12　　**修回日期：**2022-03-15　　**E-mail：**583249094@qq.com

## 0　概述

区块链作为数字货币比特币[1]的底层技术，随着比特币的发展而备受关注。区块链本质上是一种由哈希算法、数字签名、P2P网络、共识算法、智能合约等技术构成的分布式基础架构与计算范式[2]，具有透明可靠、防篡改可追溯、隐私安全保障、系统高可靠等特性[3-4]，广泛应用于金融、交通、隐私保护等领域[5-6]。共识机制[7]是区块链的必要元素和核心部分，是确保区块链系统高效合作的关键。共识机制是指分布式系统中全部节点（或大部分节点）就某个数据的真实性或者某条交易的价值达成一致并据此更新各节点记录的机制。根据不同场景和应用需求，需要设计不同的共识机制。典型的区块链共识机制大致可分为证明类共识机制（如PoW[8]、PoS、DPoS等）和拜占庭协议机制[9-10]，其中实用拜占庭容错（Practical Byzantine Fault Tolerance，PBFT）[11]算法因能够解决拜占庭问题而得到广泛应用。PBFT基于状态机复制原理，通过一致性、检查点、视图转换3个协议，当系统中约有1/3的节点为恶意节点时仍能确保系统正常运行，同时大幅降低了共识过程的通信复杂度，但PBFT存在主节点选取随意[12]、通信复杂度高、共识效率低[13]等问题[14]。

针对PBFT算法主节点选取随意导致恶意节点具有较大概率成为主节点的问题：WANG等[15]提出CPBFT算法，该算法根据信用等级划分节点，并将相应的信用系数分配给不同级别的节点；ZHU等[16]提出CDBFT算法，该算法建立一种特权分类机制，有效地防止预期节点被选中；ZHANG等[17]提出实用的基于量化角色的拜占庭共识算法（QPBFT），该算法基于层次分析法（Analytic Hierarchy Process，AHP）对节点的可靠性属性进行量化，通过引入量化角色，使可靠性评价得分较高的节点更有可能参与区块生产；ZHENG等[18]将C4.5决策树与PBFT结合，通过计算信息熵进行节点分类的同时引入投票机制确定领导节点；GAO等[19]将EigenTrust模型与共识算法结合，用节点间的交易来评估节点的信任程度，从而选择网络中质量较高的节点来构建共识层；TONG等[20]将PeerTrust模型与共识结合，取代了原来所有节点都参与的情况，从而使分布式网络的规模可任意扩展；WANG等[21]注意到现有基于信誉投票改进PBFT的共识算法普遍存在马太效应造成信用价值积累问题，分别采用不同的信任模型并使用不同的积分函数来缓解上述问题，但由于部分参数设置不够合理，信用价值累积的问题并没有得到根本的解决。针对通信复杂度较高的问题，现有研究方法主要分成两类。一类是基于上述方法，先将节点分类，再通过使更多的诚实节点参与共识过程，减少共识过程的通信复杂度。另一类是对PBFT一致性协议[22-23]本身进行改进。

但上述算法仍存在两方面的问题。一是系统中所有节点初始信誉值均由本地计算产生，缺少验证手段，可信程度难以保证，可能会出现部分节点为了获取利益而恶意篡改自身初始信誉值，进而发动对系统的恶意攻击的情况。二是过于复杂的节点分类以及选举机制造成了额外的通信开销。现有多数算法都通过设置信誉模型、奖惩函数以及投票机制减少共识过程中的恶意节点数量，降低共识过程的通信复杂度，但忽视了设置各种模型函数本身给系统带来了额外的通信复杂度。从整体上看，这些算法并没有真正降低系统的通信复杂度。

本文提出一种基于信誉值投票与随机数选举的RN-VPBFT共识算法。增设初始记账节点，降低选举过程的通信复杂度以及提升选举公平性。将所用时间证明（Proof of Elapsed Time，PoET）[24]的领导者选举思想与现有PBFT共识算法相结合，通过在所有共识节点内部随机选举的方式避免主节点总由某些节点担任的现象，降低系统的中心化趋势。基于贪心算法的概念，由不参与共识的节点担任监督节点。简化PBFT的共识流程，以降低共识过程的通信复杂度，提高共识效率。

## 1　实用拜占庭容错算法

PBFT算法是一种保证分布式系统和拜占庭故障节点一致性的通用解决方案，主要解决系统中恶意节点向其他节点发送错误信息扰乱系统正常运行的问题。PBFT算法在保证系统安全性和可靠性的前提下提供了$(n-1)/3$的容错性，即允许系统至多存在1/3的失效节点。

PBFT要求节点共同维护一个状态且所有节点保持一致，因此需要运行一致性、视图转换、检查点等3类基本协议。一致性协议通过三阶段共识保证所有节点数据存储的一致性，若在一致性协议中主节点被检测出故障或作恶，则触发视图转换协议以更换出现故障的主节点。检查点协议是一个周期性过程，系统会设置一个检查的时间点，实现定期处理日志、节约资源并及时纠正节点状态的功能。

### 1.1　PBFT算法的一致性协议

一致性协议又称三阶段协议，是PBFT算法的核心，主要包括预准备、准备、提交3个阶段。PBFT算法共识过程如图1所示，其中，Client表示客户端，Primary node表示主节点，Replica 1，2表示备份节点，Replica 3被认为是错误节点。

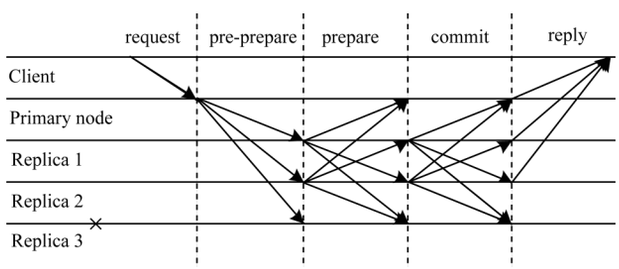


**图1　PBFT算法共识过程**

**Fig.1　Consensus process of PBFT algorithm**

PBFT算法共识的简要流程如下：

1)消息请求。客户端向主节点发送请求，如式(1)所示：

$$<\text{request}, o, t, c> \tag{1}$$

其中：request为消息名称；$o$为具体操作；$t$为时间戳；$c$为客户端标识。

2)预准备阶段。主节点将客户端的消息通过式(2)发送给其余节点：

$$<\text{pre-prepare}, V, n, d> \tag{2}$$

其中：$V$为视图编号；$n$为节点编号；$d$为信息摘要。若消息通过验证，则进入准备阶段。

3)准备阶段。备份节点之间发送如式(3)所示的消息：

$$<\text{prepare}, V, n, d> \tag{3}$$

当节点接收到超过$2f+1$个不同节点的pre-prepare和prepare信息并通过验证后进入确认阶段。

4)确认阶段。节点之间发送如式(4)所示的确认消息：

$$<\text{commit}, V, n, S(m)> \tag{4}$$

其中：$S(m)$为节点签名集合。

5)回复阶段。当节点收到$2f+1$个不同节点的确认消息后向客户端发送回复消息，如式(5)所示：

$$<\text{reply}, V, t, c, n, r> \tag{5}$$

当客户端收到$f+1$个消息时代表达成共识。

### 1.2 PBFT算法的视图转换协议

若某个备份节点检测出主节点出现问题时，触发视图转换协议，将视图编号$V$变更为$V+1$，同时不再接受除检查点、视图转换和新视图外的其他消息请求。PBFT算法视图转换过程如图2所示，其中，Replica 0表示出现问题的主节点，Replica 1表示新的主节点。

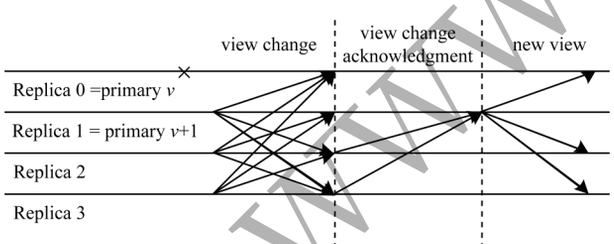


**图2 PBFT算法视图转换过程**

**Fig.2 View change process of PBFT algorithm**

PBFT算法视图转换的简要流程如下：

1)视图转换阶段。当系统中任一备份节点发现主节点出现问题时，将“视图转换”验证广播给所有节点。

2)确认视图转换。当一个节点收到$2f+1$个确认信息后(包括自己的信息)，将“确认视图转换”信息发送给视图$V+1$的主节点。新的主节点在接收到“视图转换”以及“确认视图转换”信息后进入新的视图。

3)新视图阶段。新节点确认系统状态后根据本地块链数据执行一致性协议。

PBFT算法虽然在一定程度上改善了传统共识算法通信复杂度高的问题，但由于恶意节点的存在，使得整个共识过程中节点间必须通过两两通信以确保消息的可靠。随着系统规模的不断扩大，PBFT通信复杂度增长迅速，因此本文针对该问题对算法进行改进，进一步降低算法复杂度。

## 2 RN-VPBFT共识算法

本文提出一种基于信誉值投票和随机数选举的RN-VPBFT共识算法，通过建立节点信誉模型、增设监督节点以及改进主节点选取方式，保证系统安全，降低共识过程的通信复杂度和系统的集中化趋势，提高共识效率。

### 2.1 RN-VPBFT共识算法流程

RN-VPBFT共识算法流程如图3所示，执行过程以轮为单位，每一轮执行过程分为准备、共识、结束3个阶段。在第一轮共识过程开始前，需要通过投票确定系统中所有节点的初始信誉值。

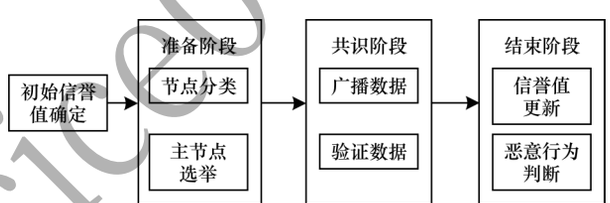


**图3 RN-VPBFT算法流程**

**Fig.3 Procedure of RN-VPBFT algorithm**

### 2.2 初始信誉值确定

所有新加入系统的节点需要通过相互投票的方式确定初始信誉值。现有算法多数通过节点间的相互通信确定得票数，并在本地通过计算得到自身的初始信誉值，可能会出现部分节点恶意篡改自身初始信誉值的现象。设置初始记账节点能够将所有投票记录集中于一个节点，这样既避免了节点的两两通信，又确保了投票结果的真实性。初始信誉值确定的具体步骤如下：

1)确定初始记账节点。所有节点首先按照进入系统的先后顺序分配各自的节点编号$1,2,\cdots,N$，然后系统随机产生1到$N$的一个随机数，节点编号与该随机数相同的节点即为初始记账节点。

2)投票。每个节点对系统中所有其他节点投赞成或反对票，并将投票结果复制两份，一份写入本地日志，另一份发送给记账节点，由记账节点计算初始信誉值。

3）确定初始信誉值。初始记账节点首先将所有节点的投票结果进行汇总并统计各个节点获得的赞成票 $S_i(i=1,2,\cdots,N)$ 与反对票 $A_i$，然后根据式（6）计算得到每个节点的初始信誉值，按信誉值的降序排序 $T_1>T_2>\cdots>T_n$。最后将各个节点的初始信誉值（$T_i$）及其对应排名发送给相应节点。

$$T_i=\frac{S_i-S_{\min}}{S_{\max}-S_{\min}} \tag{6}$$

所有节点在接收到初始记账节点发送的信息后均可以向初始记账节点提出质疑，通过访问初始记账节点的本地数据来确认接收信息的真实性。若初始记账节点在接受查询的过程中出现问题，则重新执行上述流程，同时该节点的信誉值清零且被系统记录为恶意节点，并无法参与共识协议，只能被动接收经过共识后的数据；若所有节点的初始信誉值均正确，则担任初始记账节点的节点信誉值将根据自身的初始信誉值的大小获得不同程度的奖励。

### 2.3　准备阶段

RN-VPBFT共识算法准备阶段主要完成节点分类以及主节点选举工作，如图4所示。

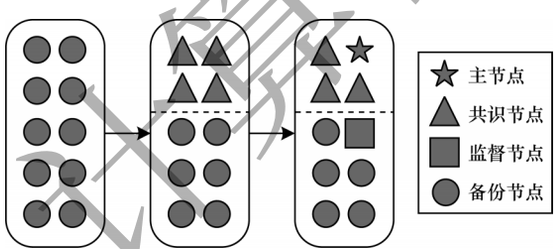


**图4　RN-VPBFT算法准备阶段示意图**

**Fig.4　Schematic diagram of the preparation stage of the RN-VPBFT algorithm**

#### 2.3.1　节点分类

RN-VPBFT算法节点模型如图5所示，节点被划分为主节点、共识节点、备份节点和监督节点4种节点。4种节点各司其职，相互监督，共同维护系统平衡：

1）备份节点。备份节点不参与共识过程，只能根据主节点传递的信息更新本地信息。同时，备份节点能够对监督节点进行监督，并有权弹劾监督节点。所有加入系统的网络节点都是备份节点。

2）共识节点。共识节点负责接收并验证主节点传递的信息，保证系统一致性。通过对系统中所有 $N$ 个节点信誉值进行比较，选取信誉值较高的前 $N_1$ 个节点为共识节点。

3）主节点。主节点负责接收用户需求、确认提交数据、打包并生成新区块。

4）监督节点。监督节点主要保证系统的安全。有权查询其他所有节点的本地日志，监督整个共识过程，查询备份节点的信息更新情况以及负责每轮共识结束后节点的信誉值更新。

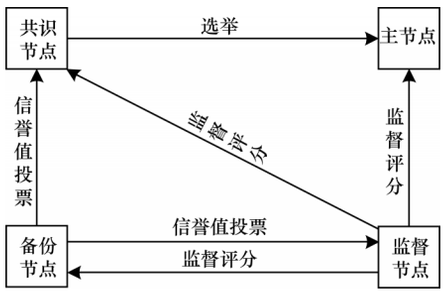


**图5　RN-VPBFT算法节点模型**

**Fig.5　Node model of RN-VPBFT algorithm**

#### 2.3.2　主节点和监督节点选举

PBFT的主节点选举方式如式（7）所示：

$$P=V \bmod |N| \tag{7}$$

其中：$P$ 为主节点。

由于算法中主节点编号与视图编号有很大的相关性，因此能够很容易被系统中的恶意节点预测，进而达到提前攻击的目的，不利于保证系统的安全性。现有PBFT改进算法多数基于节点的信誉值排序，选取信誉值最高（可信度最高）的节点担任主节点，这样虽然能够在很大程度上保证系统的安全，但每一轮共识结束后，主节点相较于其他节点往往会获得更多的报酬，随着时间累积，主节点往往仅会由某几个节点担任且节点之间的信誉值差值会越来越大，出现系统的集中化趋势。针对该问题，本文基于所用时间证明的领导者选举思想，所有共识节点在信誉值的基础上随机选举主节点。

1）所用时间证明

PoET概念是由英特尔于2016年初提出，提供了一个现成的高科技工具来解决随机领导者选举的计算问题，通常用于许可的区块链网络，以决定网络的采矿权或区块获胜者。PoET基于公平彩票系统的原则，使得网络参与者拥有公平的获胜机会。PoET算法工作流程如下：区块链网络中每个节点都会生成随机等待时间并在指定的持续时间内进入休眠状态。首先完成指定等待时间（具有最短等待时间）的节点被唤醒并向区块链提交新块，然后向整个对等网络广播必要的信息，最后重复相同过程以发现下一个新块。PoET整个共识过程需要具备2个重要因素：（1）参与节点真正地选择了随机的时间，而不是参与者为了获胜而故意选择的较短持续时间；（2）获胜者确实已经完成了指定等待时间。由于实际的区块链系统中所有的节点并不都是可信的，因此综合考虑上述2个因素以及在等待时间内节点处于休眠状态而造成的时间损耗问题，提出改进的所用时间证明。

2）改进的所用时间证明

利用监督节点的职能，在所有共识节点内部提出一种基于随机数的主节点选取方式，通过节点自身的日志记录以及监督节点对其他节点日志的访问，确保了当选主节点的公平性和真实性，避免了时间浪费的同时在一定程度上解决了PoET存在的问题。

依据式(8)和式(9)选举产生主节点($L$)和监督节点($S_u$):

$$L=i,\min(\mathrm{RN}(i)),\ i=1,2,\cdots,N_1 \tag{8}$$

$$S_u=j,\max(S_i),\ i=N_1+1,N_1+2,\cdots,N \tag{9}$$

改进的所用时间证明算法的工作流程如下:

(1)在所有共识节点内部生成同一范围内的RN($i$),并将<RN($i$),$i$>写入本地日志。

(2)共识节点广播<RN′($i$),$i$>给监督节点,监督节点通过访问共识节点的日志以验证收到的随机数无误后广播<min(RN′($i$)),$i$>。当存在多个节点同时拥有最小随机数时,广播<min (RN′($i$)),min($i$)>。

(3)若监督节点发现min(RN′($i$))≠min(RN($i$)),则监督节点会向整个系统广播该共识节点$i$,该节点将会立刻被驱逐出共识节点,并被系统记录为恶意节点。

(4)由于监督节点的高风险性,因此所有共识节点均可以在共识过程开始之前对结果提出质疑,通过节点间的相互监督,保证系统安全。

### 2.4 共识阶段

RN-VPBFT算法共识阶段主要执行一致性协议,如图6所示。在传统的PBFT算法共识过程中:当系统中存在$n$个节点时,达成共识所需的通信次数大致等于$2n^2$;当系统节点数不断增加时,达成共识所需要的通信次数将迅速增多,这不仅会带来传递消息的爆炸性增长,还会大大延迟达成共识所需要的时间,进而成为系统性能的瓶颈。

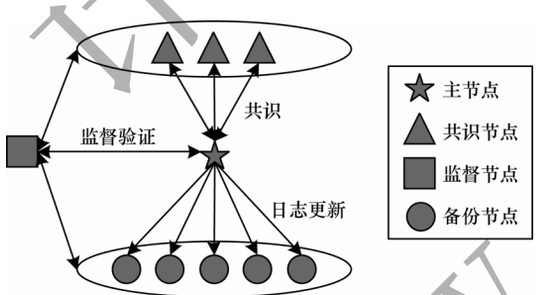


**图6 RN-VPBFT算法共识阶段示意图**

**Fig.6 Schematic diagram of the consensus stage of the RN-VPBFT algorithm**

为降低通信复杂度,本文提出的共识机制首先将复杂的消息验证工作交给监督节点完成。由于事先已按照信誉值对节点排序和分类,因此一致性协议的过程中参与共识的节点大概率为诚实节点。同时,在共识过程中,认为每个节点都能够做出自己的判断,主节点仅负责汇总所有的判断,然后做出最终决策。因此,可以用半数以上投票确认的方式验证数据信息的真实性,进而将原有的通信复杂度$2n^2$降至$2n$。基于简化的一致性协议,RN-VPBFT算法共识过程如图7所示,其中,Client表示客户端,Superior node表示监督节点,Primary node表示主节点,Replica 1,2,3表示共识节点,Replica 4表示备份节点。

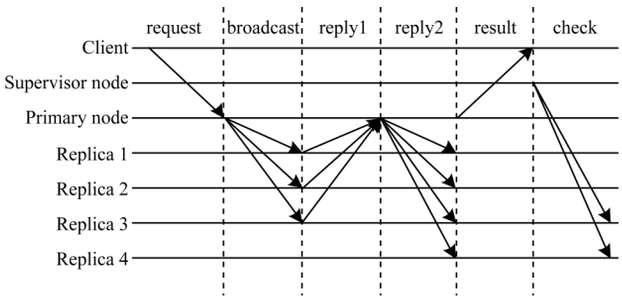


**图7 RN-VPBFT算法共识过程**

**Fig.7 Consensus process of RN-VPBFT algorithm**

主节点在对所有判断进行汇总并采用多数决定原则做出判断后,需要向所有备份节点广播每个节点对该信息的判断情况以及最终的决策情况。RN-VPBFT算法共识过程的简要流程如下:

1)当某一节点需要对数据库进行更新操作时,首先向主节点提出请求,主节点收到请求后可将待更新的数据通过式(10)发送给所有的共识节点进行验证:

$$K_{p_i}<\mathrm{timestamp},T> \tag{10}$$

其中:$K_{p_i}$为节点的公钥;timestamp为时间戳;$T$为发送时间。

2)每个共识节点收到主节点发送的消息后,对消息的真实性做出判断,并将自身对消息的判断通过式(11)发送回主节点。

$$K_{N_i}<\mathrm{timestamp},T,\mathrm{judgement}> \tag{11}$$

其中:$K_{N_i}$为节点的私钥;judgement为每个节点的判断结果。

3)当主节点收到至少$2f+1$个来自不同节点的信息时,对该数据信息进行最终判断,并将最终判断结果返回给系统中的所有节点。

需要注意的是,当主节点中存储的已验证的数据信息达到一定数量时,主节点必须将这些信息打包并广播给系统中的所有节点。

在整个共识过程中,节点间传递的信息量较传统PBFT算法减少了视图编号信息以及信息摘要,取而代之的是节点对获得消息做出判断。

### 2.5 结束阶段

RN-VPBFT结束阶段主要完成数据打包、节点信誉值更新、恶意节点记录等工作,如图8所示。信誉值更新的伪代码如算法1所示。

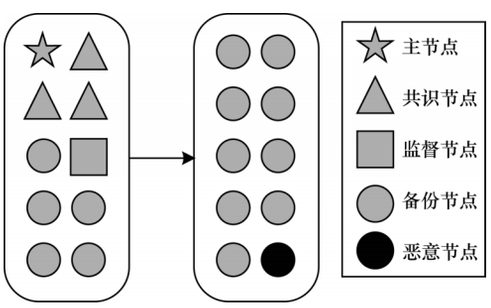


**图8 RN-VPBFT算法结束阶段示意图**

**Fig.8 Schematic diagram of the end stage of the RN-VPBFT algorithm**

**算法1**　信誉值更新

1.for i ≤ N do
2.calculate $\alpha_i$, $\beta_i$, $T_1$
3.$T_i'=\alpha\times T_i+\beta\times T_1$
4.$T_i\leftarrow T_i'$
5.end for
6.if $T_i\leqslant 0.5$ do
7.add i to E//E代表恶意节点的结合
8.return $T_i$, E
9.next round

**定义1**　共识轮数$R$指在一个共识阶段中达成共识次数$R_s$与未达成共识次数$R_f$的总和。根据主节点选举规则，为确保理论上每个共识阶段中所有共识节点都有机会担任主节点，$R$和$N_1$的关系一般满足$R=2N_1$。

**定义2**　每$R$个共识轮数称为一个共识阶段$S_t$。当系统进入一个新的共识阶段后，需要对系统内部节点的信誉值进行更新，然后重新确定各节点的角色。

在每个共识阶段结束后，监督节点根据系统中所有节点在当前共识阶段的表现并综合节点在前一轮共识阶段的表现，按照式(12)对每个节点的信誉值进行更新，并通过式(13)发送给各个节点。各个节点收到监督节点的消息后进行验证，待验证结束后监督节点根据其他节点的反馈，再对自身的信誉值进行更新。待所有节点完成信誉值更新后，系统会记录当前阶段出现的恶意节点编号，所有被记录的节点在之后的所有过程中只能充当备份节点，根据主节点发送的消息更新本地信息。

$$T_i'=\alpha\times T_i+\beta\times T_1 \tag{12}$$

$$<T_i, T_i', \text{behaviour}, \text{role}> \tag{13}$$

其中：$\alpha$、$\beta$为加权系数，满足$\alpha+\beta=1$；$T_1$表示信誉值更新规则；$T_i'$表示更新后的节点信誉值；behaviour表示节点在当前共识阶段中的表现；role表示节点在上一阶段系统中担任的角色。

对于不同的节点类型，$\alpha$、$\beta$、$T_1$的取值不同，同时考虑到节点更新后的信誉值在很大程度上应由节点在本轮共识阶段的表现决定，因此$\alpha\leqslant\beta$。根据上文对每个节点在系统中的职能，将各类节点按照在系统中的作用做如下排序：监督节点>主节点>共识节点>备份节点，并根据节点的作用大小，设置不同参数$\alpha$、$\beta$、$T_1$，如表1所示。

**表1　信誉值更新参数**

**Table 1　Reputation value update parameters**

| 节点行为 | 节点类型 | $\alpha$ | $\beta$ | $T_1$ |
|---|---|---|---|---|
| 诚实行为 | 主节点 | 0.30 | 0.70 | 一个$S_t$内成功达成共识的轮数除以一个$S_t$内的总轮数 |
| | 共识节点 | 0.35 | 0.65 | 一个$S_t$内成功达成共识的轮数除以一个$S_t$内的总轮数 |
| | 监督节点 | 0.20 | 0.80 | 1 |
| | 备份节点 | 0.50 | 0.50 | 一个$S_t$内成功达成共识的轮数除以一个$S_t$内的总轮数 |
| 恶意行为 | 主节点 | 0.30 | 0.70 | 0 |
| | 共识节点 | 0.35 | 0.65 | 0 |
| | 监督节点 | 0.20 | 0.80 | 0 |

为了贴近实际，假设备份节点能够在接收到主节点发送的信息后及时对本地数据进行更新而不会恶意篡改数据，不存在恶意行为。此外，所有诚实节点的信誉值范围为0~1，恶意节点的信誉值不超过0.5。设置节点信誉值动态更新机制既可以有效减少恶意节点对系统的不利影响，鼓励节点遵守系统规则，又可以保证节点的积极性，防止高信任度节点在共识过程中出现的恶意行为，激励节点在共识过程中做出诚实行为。与现有多数改进算法中设置的奖惩函数不同，本文设计的奖惩函数在此基础上还能很好地区分恶意节点与诚实节点，并且保证所有诚实节点的信誉值相近，降低了系统集中化的可能性。

## 3　实验与结果分析

实验通过对比多种共识算法在容错性、节点信誉值、通信复杂度以及系统集中化趋势4个方面的表现，测试RN-VPBFT共识算法的性能。实验环境为Windows 10操作系统，系统内存为16 GB，CPU为Intel Core i7处理器。实验基于Python对RN-VPBFT、PBFT、CPBFT、RC-VPBFT在内的多种共识算法进行模拟仿真，开发语言为Python 3.7。

### 3.1　容错性分析

假定节点总数是$N$，故障或者恶意节点数为$f$，剩余正确节点数为$N-f$，RN-VPBFT和传统PBFT共识算法本质上相同，只要收到$N-f$个消息且$N-f>f$就能做出决定，但$N-f$个消息中可能存在$f$个恶意节点冒充的消息(或因网络延迟导致$f$个恶意节点的消息先被收到)，则正确消息数为$N-f-f$。为达到多数一致，正确消息必须占多数，也就是$N-f-f>f$，因此$N$至少等于$3f+1$。

### 3.2　节点信誉值分析

利用Python搭建小型区块链系统，系统共设30个节点，包含9个拜占庭节点。每次实验经过4个共识阶段，每个阶段包含40轮共识过程，分别对每

个阶段结束后各节点的信誉值进行记录，如图9所示。每个共识阶段结束后，所有诚实节点之间信誉值的最大差值如表2所示。

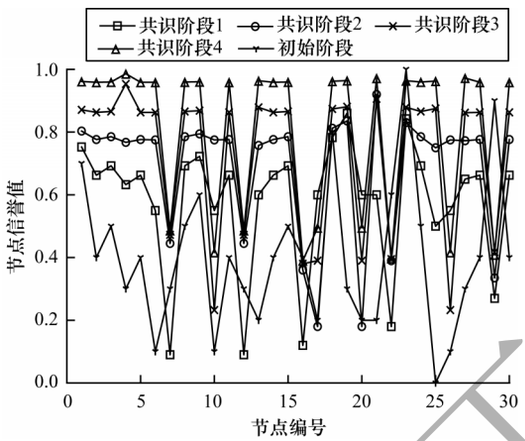


图9 各共识阶段不同节点的信誉值分布

Fig.9 Reputation value distribution of different nodes in each consensus stage

表2 共识阶段结束后诚实节点之间信誉值的最大差值

Table 2 Maximum difference in reputation value between honest nodes after the consensus phase ends

| 共识阶段 | 最大差值 |
|---|---|
| 1 | 0.36 |
| 2 | 0.17 |
| 3 | 0.20 |
| 4 | 0.02 |

结合图9和表2可分析得出：

1）系统中恶意节点的编号分别为7、10、12、16、17、20、22、26、29。

2）在共识阶段2结束后，系统中的诚实节点与恶意节点已基本能够通过信誉值的大小进行区分，进而保证了整个系统后续运行的安全性与稳定性。

3）随着共识过程的进行，各阶段在诚实节点处的曲折程度趋向平稳，诚实节点的信誉值差距逐渐减小。

4）节点编号为7、10、17、26的错误节点在共识阶段1后信誉值均超过了阈值0.5，主要原因是这4个节点相较于其他错误节点初始信誉值更低，在共识阶段1中没有能够获得担任主节点以及监督节点的权利，因此未被系统发现，但在共识阶段1后，其担任了对应角色，最终被系统记录。各共识阶段成功达成共识的轮数如表3所示。由表3中数据可知，共识阶段1的共识成功率较低，这是因为节点的初始信誉值主要依靠节点之间投票决定，具有一定的偶然性，可能会出现部分恶意节点获得较高的初始信誉度的情况，但随着共识过程的不断深入，系统将所有恶意节点一一记录，共识成功率达到100%。

表3 各共识阶段成功达成共识的轮数

Table 3 The number of rounds that successful reached consensus at each consensus stage

| 共识阶段 | 总轮数 | 达成共识的轮数 | 成功率/% |
|---|---|---|---|
| 1 | 40 | 23 | 57.5 |
| 2 | 40 | 36 | 90.0 |
| 3 | 40 | 40 | 100.0 |
| 4 | 40 | 40 | 100.0 |

### 3.3 通信复杂度分析

在PBFT算法中，广播消息需要进行预准备、准备和确认3个阶段的通信，对应的通信次数分别为$N$、$N^2$和$N^2$。在计算其与用户端的通信次数后，PBFT算法中一个完整的共识过程所需的通信次数如下：

$$T=1+N+N^2+N^2+N=2N^2+2N+1 \tag{14}$$

现有研究结果仅比较了传统算法与改进算法在共识过程中的通信复杂度，忽略了节点因分类、投票等过程给系统带来的额外通信复杂度。本文综合考虑了上述因素，将RN-VPBFT与PBFT、CPBFT、RC-VPBFT算法的通信复杂度进行分析比较，结果如图10所示。

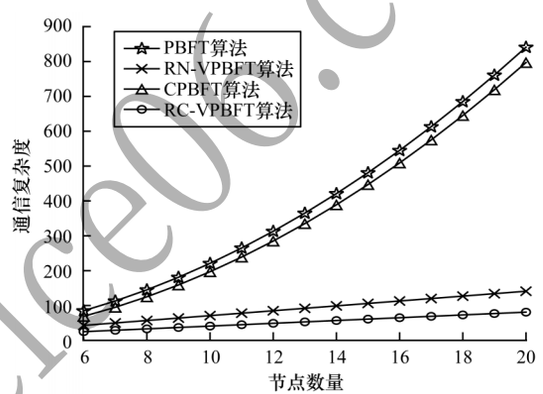


图10 不同算法的通信复杂度比较

Fig.10 Comparison of communication complexity among different algorithms

由图10可以看出：CPBFT算法虽然在共识阶段将通信复杂度降至$N^2-N-1$，但综合考虑在共识阶段之前的通信次数，总的通信次数为$T=N^2+N-1+N^2-N-2=2N^2-3$，相较于PBFT算法提升较小；RC-VPBFT算法虽然在整个共识过程中需要的通信次数仅为$4N+1$，但是在此之前需要进行聚类工作，这也在一定程度上增加了通信复杂度；RN-VPBFT算法在假设所有过程中的所有节点都进行了相关数据验证的情况下，除了初始值确定需要额外的$3N$次通信外，每轮共识主要包括领导者选举、共识、信誉值更新3个阶段，总的通信次数为$T=2N_1+2N_1+N+2N\leqslant 7N$，因此整个过程的通信复杂度由$O(N^2)$降至$O(N)$。

### 3.4 系统集中化趋势分析

区块链的核心优势是去中心化，因此需要考虑共识算法对系统去中心化程度的影响。通过对同一区块链系统分别使用RN-VPBFT与VPBFT[25]算法，统计在使用不同共识算法后系统中各个节点担任主

节点的次数并进行比较，结果如图11、图12所示。由图11、图12可以看出：VPBFT算法主要通过投票选出特定节点作为主节点，其他节点被选为主节点的机会很小，使得整个系统倾向于成为一个集中式系统；RN-VPBFT算法允许更多节点参与区块生产活动，所有满足要求的节点都有机会被选举成为主节点，能够更好地维护系统的去中心化特性。

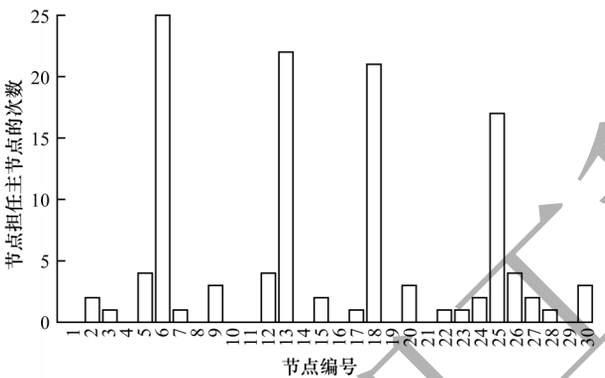


**图11　VPBFT算法中每个节点担任主节点的次数**

**Fig.11　The number of times each node becomes the primary node in the VPBFT algorithm**

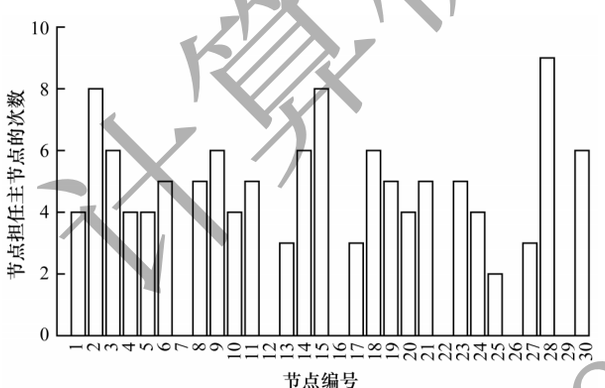


**图12　RN-VPBFT算法中每个节点担任主节点的次数**

**Fig.12　The number of times each node becomes the primary node in the RN-VPBFT algorithm**

## 4　结束语

针对传统PBFT共识算法容易出现集中化趋势、通信复杂度高等问题，本文提出一种基于随机数选举与投票机制的RN-VPBFT共识算法。引入兼具高风险和高收益的监督节点，避免了节点间因频繁通信带来的高通信复杂度。利用随机参数确保了选举的公平性，降低了系统的集中化趋势。此外，依据节点不同身份设置的信誉值更新策略不仅能够有效区分系统中的诚实节点与恶意节点，而且能够在一定程度上简化一致性协议并保证其安全与稳定运行。实验结果表明，RN-VPBFT算法相比传统PBFT共识算法具有更好的去中心化特性，并且有效降低了通信复杂度。后续将在不改变算法效率及通信复杂度的基础上增强拜占庭系统节点的容错性，并且分析与研究系统节点数量的实时变化对共识算法的影响，进一步提升系统容错性及动态性。

**参考文献**


[1] NAKAMOTO S. A peer-to-peer electronic cash system[EB/OL]. [2022-01-17]. http://bitcoins.info/bitcoin.pdf.

[2] 袁勇，王飞跃. 区块链技术发展现状与展望[J]. 自动化学报，2016，42(4)：481-494.
YUAN Y，WANG F Y. Blockchain：the state of the art and future trends[J]. Acta Automatica Sinica，2016，42(4)：481-494.（in Chinese）

[3] WANG H，QIN H，ZHAO M H，et al. Blockchain-based fair payment smart contract for public cloud storage auditing[J]. Information Sciences，2020，519：348-362.

[4] GAO F，ZHU L H，SHEN M，et al. A blockchain-based privacy-preserving payment mechanism for vehicle-to-grid networks[J]. IEEE Network，2018，32(6)：184-192.

[5] 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）

[6] 王志铧，柳平增，宋成宝，等. 基于区块链的农产品柔性可信溯源系统研究[J]. 计算机工程，2020，46(12)：313-320.
WANG Z H，LIU P Z，SONG C B，et al. Research on flexible and reliable blockchain-based traceability system for agricultural products[J]. Computer Engineering，2020，46(12)：313-320.（in Chinese）

[7] 刘懿中，刘建伟，张宗洋，等. 区块链共识机制研究综述[J]. 密码学报，2019，6(4)：395-432.
LIU Y Z，LIU J W，ZHANG Z Y，et al. Overview on blockchain consensus mechanisms[J]. Journal of Cryptologic Research，2019，6(4)：395-432.（in Chinese）

[8] JAKOBSSON M，JUELS A. Proofs of work and bread pudding protocols（extended abstract）[M]. Berlin，Germany：Springer，1999.

[9] 谭敏生，杨杰，丁琳，等. 区块链共识机制综述[J]. 计算机工程，2020，46(12)：1-11.
TAN M S，YANG J，DING L，et al. Review of consensus mechanism of blockchain[J]. Computer Engineering，2020，46(12)：1-11.（in Chinese）

[10] 范捷，易乐天，舒继武. 拜占庭系统技术研究综述[J]. 软件学报，2013，24(6)：1346-1360.
FAN J，YI L T，SHU J W. Research on the technologies of Byzantine system[J]. Journal of Software，2013，24(6)：1346-1360.（in Chinese）

[11] CASTRO M，LISKOV B. Practical Byzantine fault tolerance[EB/OL]. [2022-01-17]. https://www.researchgate.net/publication/2437947_Practical_Byzantine_Fault_Tolerance.

[12] 吴晓彤，柳平增. 基于备选投票机制的低时延PBFT改进研究[J]. 计算机工程，2021，47(7)：117-125，134.
WU X T，LIU P Z. Delay optimization for PBFT based on alternative voting mechanism[J]. Computer Engineering，2021，47(7)：117-125，134.（in Chinese）

[13] ZHENG X D，FENG W L. Research on practical Byzantine fault tolerant consensus algorithm based on blockchain[EB/OL]. [2022-01-17]. https://iopscience.iop.org/article/10.1088/1742-6596/1802/3/032022/pdf.

[14] LAMPORT L，SHOSTAK R，PEASE M. The Byzantine generals problem[J]. ACM Transactions on Programming Languages and Systems，1982，4(3)：382-401.


（下转第56页）

（上接第49页）

[15] WANG Y H, CAI S B, LIN C L, et al. Study of blockchain's consensus mechanism based on credit[J]. IEEE Access, 2019, 7: 10224-10231.

[16] ZHU S C, ZHANG Z Y, CHEN L Q, et al. A PBFT consensus scheme with reputation value voting based on dynamic clustering[M]. Berlin, Germany: Springer, 2020.

[17] ZHANG Z J, ZHU D L, FAN W. QPBFT: practical Byzantine fault tolerance consensus algorithm based on quantified-role[C]//Proceedings of IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications. Washington D. C., USA: IEEE Press, 2021: 991-997.

[18] ZHENG X D, FENG W L, HUANG M X, et al. Optimization of PBFT algorithm based on improved C4. 5[J]. Mathematical Problems in Engineering, 2021(2): 1-7.

[19] GAO S, YU T Y, ZHU J M, et al. T-PBFT: an EigenTrust-based practical Byzantine fault tolerance consensus algorithm[J]. China Communications, 2019, 16(12): 111-123.

[20] TONG W, DONG X W, ZHENG J W. Trust-PBFT: a PeerTrust-based practical Byzantine consensus algorithm[C]//Proceedings of International Conference on Networking and Network Applications. Washington D. C., USA: IEEE Press, 2019: 344-349.

[21] WANG H Y, GUO K X. Byzantine fault tolerant algorithm based on vote[C]//Proceedings of International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery. Berlin, Germany: Springer, 2019: 190-196.

[22] 方维维, 王子岳, 宋慧丽, 等. 一种面向区块链的优化PBFT共识算法[J]. 北京交通大学学报, 2019, 43(5): 58-64.
FANG W W, WANG Z Y, SONG H L, et al. An optimized PBFT consensus algorithm for blockchain[J]. Journal of Beijing Jiaotong University, 2019, 43(5): 58-64. (in Chinese)

[23] 王壹铭, 初剑峰, 王永军, 等. 基于有向无环图的高效区块链共识算法[J]. 吉林大学学报(理学版), 2020, 58(5): 1167-1172.
WANG Y M, CHU J F, WANG Y J, et al. Efficient blockchain consensus algorithm based on directed acyclic graph[J]. Journal of Jilin University (Science Edition), 2020, 58(5): 1167-1172. (in Chinese)

[24] CHEN L, XU L, SHAH N, et al. On security analysis of Proof-of-Elapsed-Time (PoET)[C]//Proceedings of International Symposium on Stabilization, Safety, and Security of Distributed Systems. Berlin, Germany: Springer, 2017: 282-297.

[25] 王海勇, 郭凯璇, 潘启青. 基于投票机制的拜占庭容错共识算法[J]. 计算机应用, 2019, 39(6): 1766-1771.
WANG H Y, GUO K X, PAN Q Q. Byzantine fault tolerance consensus algorithm based on voting mechanism[J]. Journal of Computer Applications, 2019, 39(6): 1766-1771. (in Chinese)

编辑　陆燕菲