计算机科学
COMPUTER SCIENCE

基于贝叶斯理论的PBFT共识算法

潘彦炀, 杨槟豪, 纪庆革

引用本文

潘彦炀, 杨槟豪, 纪庆革. 基于贝叶斯理论的PBFT共识算法[J]. 计算机科学, 2026, 53(1): 331-340.

PAN Yanyang, YANG Binhao, JI Qingge. PBFT Consensus Algorithm Based on Bayesian Theory[J]. Computer Science, 2026, 53(1): 331-340.

计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.241100053

基于贝叶斯理论的 PBFT 共识算法

潘彦炀　杨槟豪　纪庆革
中山大学计算机学院　广州 510006
(panyy65@mail2.sysu.edu.cn)

摘　要　共识算法是一种用于确保区块链网络中所有节点达成一致的方法,常见的有工作量证明(Proof-of-Work,PoW)和权益证明(Proof of Stake,PoS)等,共识机制的优劣影响着区块链系统的性能。为了解决现有区块链共识算法存在的吞吐量较小、时延较长等问题,对区块链中实用拜占庭容错(PBFT)算法进行改进,引入基于 Bayes 理论的动态信任模型(Dynamic Trust Model),通过节点信任机制改变节点在共识轮中的信任度,并按照信任度进行分组等操作,在保证 PBFT 稳定性的同时提高了系统可扩展性,且完善了网络节点的加入退出机制,使得网络可拓展性得到提高。通过实验测试,相比传统 PBFT,改进后的算法在吞吐量上有 25%的提升,在节点数量达到 50 的情况下时延只有 PBFT 的一半,所提方法的这两项指标相比 HotStuff 算法和 Paxos 算法也有 20%～30%的提升。

关键词:区块链;共识算法;拜占庭容错;信任模型;贝叶斯理论
中图分类号　TP311

PBFT Consensus Algorithm Based on Bayesian Theory

PAN Yanyang,YANG Binhao and JI Qingge
School of Computer Science and Engineering,Sun Yat-sen University,Guangzhou 510006,China

Abstract　Consensus algorithm is a method to ensure that all nodes in the blockchain network reach a consensus,such as PoW and PoS. The consensus mechanism affects the performance of the blockchain system. In order to solve the problems of low throughput and long delay of existing blockchain consensus algorithms,this paper improves the PBFT algorithm in blockchain,introduces a dynamic trust model based on Bayes theory,changes the trust of nodes in the consensus round through the node trust mechanism,and conducts group operations according to the trust degree. In addition to ensuring the stability of PBFT,the system scalability is improved,and the joining and exiting mechanism of network nodes is perfected,so that the network scalability is improved. Through experimental tests,compared with traditional PBFT algorithms,the improved algorithm has a 25% improvement in throughput,and the delay is only half of that of PBFT when the number of nodes reaches 50. These two indicators also have a 20%～30% improvement compared with HotStuff algorithm and Paxos algorithm.

Keywords　Blockchain,Consensus algorithm,Byzantine fault-tolerant,Trust model,Bayesian theory

1　引言

1.1　研究背景与意义

区块链技术本质上是一种基于加密的分布式存储技术,具有去中心化、不可篡改等特点[1],交易数据通过哈希加密存储到区块上,一旦进行上链处理则任何人都无法修改。正是由于这种不可篡改的高安全性,区块链可以为许多高安全需求的问题提供思路。区块链也被认为是分布式数据库技术的一次重大改革与飞跃[2]。

区块链技术在提出之初,多应用于加密货币上[3]。2008 年由中本聪发表的 Bitcoin 论文正是最经典的代表[4],其使用 PoW 共识算法,奖励解决问题的矿工比特币。近十年来,比特币不断发展,其价格也在 2021 年初达到了惊人的 7 万美元一枚[5]。除了比特币的发展,其他各种加密货币以及以太坊的出现更是让区块链技术迈向新的高峰,可以说区块链技术是数字经济的基石[6]。

如今,共识机制的不断发展更是让区块链技术达到一个更高的台阶,其被应用于医疗[7]、食品[8]、交通[9]等多个领域。区块链凭借自身的去中心化以及不可篡改等特点,能够为各领域提供高安全性的服务。

区块链系统凭借自身的高安全性优势,在不同的领域展现出良好性能。而这种优势,正是由其背后的共识算法所建立的。共识算法[10]定义了区块链这个分布式系统中不互相信任节点之间完成可信任计算的规则,这种不信任节点之间完成可信任计算称为一次共识。

一个好的共识算法,可以在区块链中大放异彩,提升区块链的稳定性与性能。传统的 PBFT 虽然有许多优点,但其共识结构以及可扩展性仍有很大提升空间,如在共识结构上和

到稿日期:2024-11-07　返修日期:2025-02-26
通信作者:纪庆革(issjqg@mail.sysu.edu.cn)

节点加入退出协议上进行优化。针对上述问题，本文对PBFT进行分析，提出优化方案，并辅以实验进行验证。

1.2 研究内容与论文结构安排

本文的研究内容将从实用拜占庭容错算法PBFT的结构与可扩展性出发，针对PBFT现有的问题进行分析并提出相应的优化方案，具体如下：

1)在结构方面，针对PBFT在其内部节点变多的情况下性能急剧下降的问题，引入基于贝叶斯理论的信任模型对节点在共识过程中的行为进行信任度评估。同时，在共识过程中进行分组共识，将链上节点进行随机分组，分组后根据信任度进行主节点挑选，信任度高的节点在挑选中具有高优先级。

2)在可扩展性方面，在上文信任模型的基础上，设计节点动态加入退出的协议，完成节点的动态加入和退出。同时，引入黑名单机制，对于在共识过程中一直作恶的节点，系统会根据其过低的信任度将其拉入黑名单，在共识结束后广播节点的作恶情况，同时强制节点退出系统，保障系统的鲁棒性和安全性。

第2章介绍与本文研究方向相关的工作；第3章介绍基于贝叶斯理论的信任模型的概念与相关的机制；第4章介绍改进后的共识算法，使用基于贝叶斯理论的信任模型对原有的PBFT算法进行改进，并通过仿真实验从共识时延、吞吐量方面验证改进后的共识算法的性能；最后总结全文，并对存在问题以及后续研究内容进行展望。

2 相关工作

2.1 PBFT

实用拜占庭容错算法(Practical Byzantine Fault Tolerance，PBFT)由Castro等[11]于1999年提出。该算法以状态机副本复制为基础，旨在解决分布式系统中节点间的一致性问题。

PBFT算法可容忍的拜占庭节点最多为 f 个，其中，$f=\lfloor(N-1)/3\rfloor$，N 总节点个数。通过协议，PBFT算法能够在拜占庭故障的情况下仍然保持系统的一致性和可用性。

PBFT的流程分为5个阶段，分别是请求阶段、预准备阶段、准备阶段、提交阶段以及执行阶段。

2.2 分组共识架构与分片区块链

分组共识架构是一种在大规模分布式网络中实现共识的设计方式。其核心思想是将网络中的节点划分为多个小组或分区，让每个小组分别达成本地共识，然后通过特定的机制在各个小组之间协调，达成全局共识。

ResilientDB[12]是一个专为高性能和容错设计的分布式数据库系统，它支持拜占庭容错(BFT)机制，并且具有地理分布的特点。为了应对跨地理区域的分布式共识挑战，ResilientDB团队提出了GeoBFT协议。

OmniLedger[13]是一种高效、安全的分片区块链协议，旨在解决区块链在大规模交易环境下的扩展性问题。通过引入分片(Sharding)机制，OmniLedger能够实现高吞吐量和低延迟，同时确保数据的一致性和安全性。具体地，OmniLedger使用一种创新的分片方法，其将RandHound[14]无偏差随机数生成方案与基于VRF[15]的leader选举算法相结合，完成区块链的分片。

2.3 PBFT缺陷分析与改进工作

在算法复杂度方面，节点交换信息的复杂度为 $O(N^2)$，且节点越多，复杂度越高，吞吐量越低。对PBFT算法的改进大致分为改进共识结构和控制节点数量。

改进共识结构方面的研究大多致力于解决原先PBFT中节点变多使得性能下降的问题。Gueta等[16]提出了SBFT算法，其使用收集器收集节点发过来的消息，使用门限签名技术降低了通信开销。Yin等[17]提出了HotStuff算法，HotStuff在SBFT的基础上进行改进，保持门限签名不变的情况下使用流水线共识加快效率。Xu等[18]提出区块链共识算法SG-PBFT，通过对Commit阶段的改进，以及对节点组的划分来改善共识结构，从而降低通信复杂度，进一步提高共识效率。Feng等[19]提出了在每个代理节点上形成多个自治系统的可扩展动态多代理分层PBFT算法SDMA-PBFT，该自治系统能使原PBFT网络中的消息传播数量大量减少，对提高共识效率有显著帮助。Zheng等[20]使用C4.5算法进行信用评估和节点分组，选取高信用值节点组成主共识组，采用积分投票机制确定主节点，优化主节点选择。Liu等[21]则设计了一个信用和投票机制进行分组共识。

控制节点数量本质上也是为了解决当PBFT中节点过多导致性能急剧下降的问题。例如，DBFT[22]使用DPoS来减少参与PBFT共识的节点数量。具体来说，DBFT中的节点分为共识节点和普通节点。共识节点负责相互通信以生成新的区块，普通节点负责验证新块。在选举时，各节点投票选出最能代表自身利益的节点。由于短期内各节点的意愿不会轻易改变，因此投票热情不高，每次选举选择的节点大多相同，不具有代表性。因此，仅仅利用DPoS的投票思想来选择节点，很难保证系统的公平性和活跃性。

2.4 动机

本文在现有研究的基础上对PBFT算法进行改进，侧重于改进共识结构以及提高可拓展性，利用信任模型对网络节点信任度进行排序分组，将原先的单层共识结构改为双层，进一步提高共识效率；在可拓展性上，通过引入节点动态加入与退出的协议，解决了PBFT网络中节点数量固定以及作恶节点一直存在等问题。在PBFT优化方面做到双管齐下，在网络中节点变动较大的情况下，确保系统的安全性和鲁棒性。

3 基于贝叶斯理论的信任模型

3.1 贝叶斯理论

贝叶斯理论是18世纪英国数学家托马斯·贝叶斯(Thomas Bayes)提出的重要概率论理论[23]。

贝叶斯定理是关于随机事件 A 和 B 的条件概率：

$$P(A|B)=\frac{P(B|A)P(A)}{P(B)} \tag{1}$$

贝叶斯定理同时可表述为：

$$\text{后验概率}=\frac{\text{相似度}\times\text{先验概率}}{\text{标准化常量}} \tag{2}$$

式(1)中，$P(A)$ 是 A 的先验概率。$P(B|A)/P(B)$ 也被称为标准相似度，则贝叶斯定理可表述为：

$$\text{后验概率}=\text{标准相似度}\times\text{先验概率} \tag{3}$$

3.2 基于贝叶斯理论的信任模型

在信任模型中，通常通过被评估对象的历史行为来评估其信任度。贝叶斯理论的基本思想是利用先验概率结合新的评价数据，对后续行为进行预测，因此贝叶斯理论在信任评价中具有很高的适用性[24]。

3.2.1 模型概述

针对区块链中 PBFT 算法所缺乏的激励机制，本文提出了引入贝叶斯理论的信任模型，本文模型包含信任动态调整机制和信任等级机制两种。

在信任动态调整机制中，将贝叶斯理论引入信任模型评估中，收集当前节点在当前视图下共识的行为，以此为基础对节点的信任度进行评估。将得到的值作为节点信用值，评估最终结果的“先验概率”。同时，计算节点共识结果与网络共识结果二者之间的“标准相似度”。最后根据上述两个值计算得到最终的“后验概率”，即节点在本轮共识中的最终信任度评估值。

信任等级机制则在上述信任度变动之后，以节点本身信任度为基础对节点进行等级划分。高信任度的节点有着优先成为主节点的权利，而低信任度的节点自然也会面临禁止参与共识甚至被踢出网络的情况。信任等级机制通过等级划分以及对应的权利赋予，减少区块链中节点作恶的情况。

3.2.2 信任动态调整机制

信任动态调整机制的原理是根据共识过程中节点之间的相互通信，通过与目标节点进行通信的其他节点对目标节点的行为评价，以此对目标节点的信任度进行动态调整。在通信期间，如果节点正常参与共识，并且最终同步结果与大多数节点一致，与其交互的节点会给予正面评价，该节点的信任度将上升。相反，如果某节点因进行拜占庭行为，如篡改消息、冒用身份，导致其在通信过程中未能与网络中的大多数节点达成共识，则与其通信的节点会给予负面评价，该节点的信任度将下降。信任动态调整机制根据一轮共识的行为以及上一轮共识结束之后的信用度对节点信用进行动态调整。

节点 x 的信用度在某一轮的评估如下：

$$Credit(x)=Similarity(x)\times PriorCredit(x) \tag{4}$$

其中，$Credit(x)$为每一轮共识结束后，下一轮共识开始前，节点 x 最终的信用度；$PriorCredit(x)$为节点 x 在这轮共识中与其他节点交互所得到的基础信用度，$Similarity(x)$为节点 x 在这轮共识中的网络共识相似度。

而在信任评估中，节点 x 的“先验概率”，即基础信任度的计算式如下：

$$PriorCredit(x)=Rf\times Rise(x)+Mf\times Motive(x) \tag{5}$$

$$PriorCredit(x)=\frac{PriorCredit(x)-PriorCredit(x)_{\min}}{PriorCredit(x)_{\max}-PriorCredit(x)_{\min}} \tag{6}$$

式(5)为节点 x 的基础信用度计算式，其中 Rf 为正面因子，用于调节节点信用度增长趋势；$Rise(x)$为节点 x 的正面信用评价统计；$Motive(x)$为节点 x 的激励信用评价。

式(6)为基础信用度的归一化，将整个网络中节点的信用度归一化到[0,1]，以便信用等级区域的划分与计算，$PriorCredit(x)_{\max}$为网络中基础信任度最高的值，$PriorCredit(x)_{\min}$为网络中基础信任度最低的值，这部分的计算由网络主节点进行。

在一轮共识中，节点 x 的正面评价统计 $Rise(x)$ 的更新方式如下：

$$Rise(x)=\sum_{y-1}^{R_n}Rise(x)_y \tag{7}$$

其中，R_n为与 x 有过交互且对 x 为正面评价的节点数量，$Rise(x)_y$表示节点 y 对节点 x 的正面评价分数。在网络共识过程中，节点使用私钥进行签名，通过消息传递中摘要和签名对节点是否正常发送消息进行判别。根据节点的不同，$Rise(x)_y$的更新方式如下。

1)若节点 x 为主节点，在本轮共识中与节点 y 有过交互，即 y 为 x 的从节点。经节点 y 验证，x 在 PRE-PREPARE 之后的 PREPARE 和 COMMIT 两个阶段发送的消息内容与大多数节点一致通过且签名验证，即摘要和签名都是正确的，同时节点 x 最终领导从节点成功完成一轮共识。在这种情况下，认为节点 x 的行为在节点 y 的视角里是正确的，此时将 $Rise(x)_y$设置为 1，反之则设置为 0。

2)若节点 x 为从节点，在本轮共识中与节点 y 有过交互，即 y 为 x 发送消息的节点，y 可以为从节点也可以为主节点。经节点 y 验证，节点 x 发送的 PREPARE 消息与 COMMIT 消息与大多数节点一致，且签名验证通过。这种情况下认为节点 x 的行为在节点 y 的视角里是正确的，此时将 $Rise(x)_y$设置为 1，反之则设置为 0。

组成节点基础信任度的另一个部分是 $Motive(x)$，即节点的激励信用评价。在共识过程中，正确收发消息以及完成共识是最重要的一步，激励是为了给予某些节点重新回到高信任度区间的措施。例如，节点 x 在之前的共识中都正确工作，但在某一轮共识中由于网络波动等原因，无法参与该轮共识。在该轮共识结束之后的信任度评估环节中，由于节点在此轮共识中并未参与，其正面统计评价 $Rise(x)$ 为 0，意味着节点 x 可能掉落到低信任度区间甚至被踢出网络，这对节点 x 的打击无疑是巨大的且不公平的，因为它之前在网络中都正确地进行共识。节点的激励信用就是为了防止这种情况的发生，其计算方式如下：

$$Motive(x)=\left\lceil\frac{N}{k}\right\rceil\times Credit(x) \tag{8}$$

其中，$Credit(x)$ 为上一轮结束后计算得到的节点 x 的信任度，N 为节点数量，k 为分组数。

节点信任度计算的最后一部分是网络共识相似度 $Similarity(x)$。网络共识相似度表示节点所完成的结果与本轮整个网络共识得到的结果的相似程度，其计算式如下：

$$Similarity(x)=e^{-\Delta h} \tag{9}$$

计算相似度使用到了汉明距离 Δh。若节点 x 得到的结果与本轮共识得到的结果一致，则二者的汉明距离 Δh 为 0，此时其网络共识的相似度 $Similarity(x)$ 为 1；若结果之间存在差异，则汉明距离是一个大于 0 的正整数，说明节点 x 的结果与网络共识结果存在差异，其最终的信任度会在基础信任度的基础上打折扣。

在整个信任动态调整机制，即信任评估环节中，节点之间需要进行额外的通信。各个节点向交互过的节点发送对应评

价以供节点计算自身正面评价统计 $Rise(x)$ 值，计算结束后将该值与 $Motive(x)$ 一起计算得到自身基础信任度值 $PriorCredit(x)$，最后将 $PriorCredit(x)$ 发送给当前主节点以供归一化，主节点在归一化结束后将所有节点的信任度表发送给网络中所有节点。

3.2.3 信任等级机制

除了改变节点信任度的信任动态调整机制之外，本文模型还配备了信任等级机制。信任等级机制通过节点信任度的分级来为每个节点设置信任状态，赋予对应信任状态级别的节点对应的权限。节点 x 的信任状态 $C(x)$ 由信任度 $Credit(x)$ 决定，如表 1 所列，本文将信任状态级别设置为 3 类，其中 a 和 b 为信任状态变更的阈值大小。

表 1 节点信任等级

Table 1 Node trust level

Trust level	Interval
Low	$[0,a)$
Middle	$[a,b)$
High	$[b,1]$

在信任等级划分中，处于不同信任度区间的节点被分为 3 种状态。

处于$[0,a)$区间的节点的信任状态为 Low，此时他们被标记为黑名单节点，无法参与共识。节点需要多次作恶或者很长一段时间没有进行共识才会使得信任度掉落至此。设置黑名单以及禁止参与共识可以提高系统的稳定性。

区间$[a,b)$中的节点信任状态为 Middle。这类节点可以正常参与共识，也能成为主节点领导一轮共识，但并没有优先成为主节点的权利。

最后一个等级是$[b,1]$，此类节点的信任状态为 High，需要多次参与共识时表现正确才可以达到此等级。此类节点可以正常参与共识以及成为主节点领导一轮共识，相比 Middle 等级的节点，High 等级的节点有着在一轮共识，中优先成为主节点的权利。让高信任度的节点成为主节点领导共识有利于共识过程的稳定，一般情况下 a 值为 0.3，b 值为 0.7。

4 基于信任模型的拜占庭容错算法

本章将根据上述基于贝叶斯理论的信任模型提出基于该信任模型的拜占庭容错算法。首先概述区块链所面临的问题，同时对环境进行假设，然后对本文算法进行概述以及设计描述，最后通过设计实验，验证算法的有效性。

4.1 问题概述与假设

根据第 2 章中对 PBFT 算法的描述可知，PBFT 算法在执行的过程中进行主节点的轮换，即视图(View)轮换机制，每次进行共识时更换主节点；而从节点在共识过程中需要完成的任务则是计算从主节点处接收的消息，并与其他节点进行消息互通，即在 PREPARE 阶段以及 COMMIT 阶段节点需要进行消息的广播，此时节点之间会进行消息沟通，PBFT 算法的时间复杂度为 $O(N^2)$。

在节点数量较少时，节点之间进行广播的次数也相对较少，此时 PBFT 算法的效率很高。但随着节点数量的不断增加，PBFT 的效率下降趋势变快，每一次共识都需要花费很长时间在 PREPARE 阶段和 COMMIT 阶段中对节点的消息进行广播。同时因为 PBFT 的容错机制，区块链中存在最多 1/3 的拜占庭节点(即作恶节点)，在主节点选取时有概率会选到拜占庭节点使得共识失败，这增加了视图变换的次数，一定程度上降低了算法的稳定性。

与大多数 PBFT 算法一样，本文假定算法的运行环境是通过可靠的点对点双向通道连接节点间的部分同步网络环境，对于节点发出的消息，尽管会有延迟，但最终都会到达目标节点。除此之外，还有以下的假设存在于网络中：

1)保证网络的安全性，即拜占庭节点不能够破译其他节点的哈希函数以及签名，这一步保证了节点之间的信息传输安全可靠，是其他几个假设的基础。

2)系统中所有节点的公钥公开，这使得所有节点都可以查询其他节点发送的消息是否有过篡改。同时，节点维护一个表格，表格内存放其他节点的编号、公钥、信任度以及信任状态信息。

4.2 算法模型概述

基于上述假设，本文提出了基于信任模型的拜占庭容错算法，分为主节点的选取、节点分组、组内共识、组间共识、信任度调整 5 个部分。

1)初始化。在第一次共识开始之前，需要对节点进行初始化，在节点初始化中对节点进行密钥分配，同时为节点设置初始信任度并选取主节点。第一次共识结束后的每次共识，只需要对新加入的节点进行信任度设置，主节点的选取也变为跟节点信任度挂钩。

2)节点分组。在完成初始化和主节点选择后，把区段链上的节点分成 K 组。

3)组内共识。完成分组之后各组进行组内共识，即主节点发送消息到各个组的主节点，各组主节点领导组内节点进行 PBFT 共识，共识结束后将结果发送到组内主节点。

4)组间共识。在组内共识完成之后各个主节点再进行一次共识，称为组间共识，最后将共识结果 COMMIT 到客户端。

5)信任度评估。在组间共识结束之后，主节点根据动态信任模型更新节点信任度和信任状态，并重新选择主节点，执行视图切换进行下一次共识。

4.3 算法模型设计

本节按照 4.2 节中所述顺序，从各个部分对算法进行描述。

4.3.1 主节点选取

在主节点选取方面，算法使用优先级进行主节点的选取，而优先级与前文提到的节点信用度 Credit 挂钩，同时主节点的选取也与该节点上一次成为主节点到本次共识的时间间隔相关。主节点的选取优先级公式如下：

$$MN=Credit(x)\times\frac{\Delta M_x}{Mf} \tag{10}$$

其中，$Credit(x)$为节点 x 的信任度，由式(10)定义；ΔM_x为节点 x 上次担任主节点到本次共识的时间间隔；Mf 为系统设置的主节点选取参数值，用于控制时间间隔对主节点选取的影响。主节点选取前，区块链中的节点进行优先级更新，即在

上一轮共识结束之后根据动态信任模型进行信任度更新，随后根据优先级顺序选择主节点。

主节点选取算法如算法 1 所示，首先对主节点选取的优先级进行计算，再根据计算得到的优先级进行排序，最后选取排序后列表中第一的节点作为主节点。

算法 1 主节点选取算法

输入：List of Nodes：L

输出：MasterNode Number

1. Update Credit(x) of nodes
2. for x∈L do
3. (MN=Credit(x)×(ΔM_x)/Mf)
4. end for
5. List(x). Sort(M N,desc)
6. returnList[0]. number

4.3.2 节点分组

传统的 PBFT 受到节点数量的影响，在节点数量增多时，PREPARE 阶段与 COMMIT 阶段节点通信次数呈指数增长，效率下降，故本文使用节点分组进行改善。同时由于 PBFT 算法的效率与节点数量密切相关，为了防止在分组过程中分组不均匀导致节点较少的分组高速完成共识，而节点较多的分组此时还在共识过程中，组间共识需要等到所有组都共识完毕后才会执行，而分组不均会导致时间被浪费，因此在节点分组中需要保证各组节点的数量一致或者接近，本文规定分组时每组节点数最多为 $\left\lceil \frac{N}{k} \right\rceil$，其中 N 为区块链节点数量，k 为设定值，一般为分组数量。此外，分组时应该将节点完全随机地分到每个组中，避免组内拜占庭节点数量过多导致共识失败，进而影响网络共识效率。

因此，节点分组应当满足随机性和均匀性两点需求。为同时满足这两点需求，本文采用取模散列的方法对节点进行分组。将网络中信任状态为 Middle 和 High 状态的节点集合定义为 L，共分为 k 组，分组过程如算法 2 所示。

1）首先将 Middle 状态和 High 状态的节点随机分配到每组，分组过程需要无偏差随机数，该随机数由主节点通过具有强随机性、抗操控性和不可预测性的 VRF 算法生成[25]，利用时间戳 $Time$ 和节点私钥 SK 计算出 $RandNum$ 与证明 $Proof$，即：

$$RandNum=VRF(SK+Time) \tag{11}$$

每次分组时，$RandNum$ 都会进行更新。由于共识过程中时间戳在不断变化，因此满足分组所需要的随机性。

2）主节点将随机数以及证明进行广播，节点通过主节点的公钥进行验证，确保随机数的有效性。

3）使用随机数作为种子对网络节点进行混洗（Shuffle）和切片（Slice）得到随机排序。在已知分组数量 K 与节点集合 L 的情况下，节点通过编号便可确认自己在哪个分组里，即前 $\lceil L.length/k \rceil$ 为第一组，其 $index=1$。

算法 2 节点分组算法

输入：(block,L,a,k)

输出：(Index)

1. if MasterNode then
2. RandNum=VRF(SK+Time)
3. Broadcast RandNum,Proof
4. end if
5. for x∈L do
6. Verify RandNum,Proof
7. if RandNum,Proof are True then
8. Fill Slice with L
9. Use RandNum as seed and shuffle Slice
10. Confirm index by K and L. length
11. else
12. View Change
13. end if
14. end for

上述为正常节点，即信用度为 Middle 以及 High 节点分组的流程，而低信任度的节点则被禁止参与共识。通过 VRF 算法方式，可以使节点被均匀地分布在各个组中，增加随机性，使得拜占庭节点群体作恶的可能性降低。划分完成之后就会变成 k 个小组，其中通过主节点选取得到的主节点则自动成为当前小组的主节点，其他小组则进行一次主节点选取，挑选信任度排序第一的节点作为小组的主节点。为了方便区分，整个区块链的主节点依旧称为主节点，而小组内的主节点称为副主节点。组内共识由副主节点领导进行，完成共识之后由主节点领导副主节点进行组间共识。分组后的结构如图 1 所示，其中黑色线条链接的为上层集群，即主节点与各副主节点，蓝色线条链接的为下层集群，为各个小组。

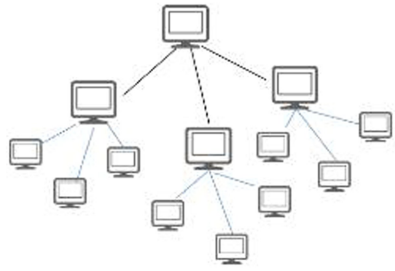

图 1 分组结构图（电子版为彩图）

Fig. 1 Group structure chart

4.3.3 共识流程

通过共识节点分组阶段，区块链中的节点被分为 k 个组，其中各个组的主节点又形成一个组，称为上层集群。在共识过程中，信任度最高的主节点为全区块链的主节点，由这个节点进行区块交易打包，并将消息传递给各个组的主节点，即副主节点，副主节点再向组内进行传递。整个共识过程分为 3 个部分：组内共识，组间共识以及更新同步信用度。

1）组内共识是由主节点发起的，其收集交易内容发送到副主节点，副主节点在自身组内进行传统 PBFT 共识，最后更新共识得到的数据到副主节点上。

2）组间共识则是在组内共识完成之后，由主节点领导各个组的副主节点进行 PBFT 共识，各个组的副主节点共识结果则是上一步组内共识的结果。

3）在完成组间共识之后，需要进行网络中信用度的更新同步，则节点发送评价消息到交互节点中以供节点进行信用度更新。

共识的具体流程如下：

1)在共识开始时,网络中客户端 C 向主节点发送请求消息 REQUEST,消息格式为$\langle\langle REQUEST,o,t,c\rangle\sigma_c\rangle$。其中 o 为客户端请求执行的操作,t 为时间戳,c 为客户端标识,σ_c 为客户端签名。

2)组间共识 PRE-PREPARE 阶段。主节点对客户端消息验证无误之后,由主节点 N 打包生成区块,同时将预准备消息发送到各组副主节点,消息格式为 $\langle\langle M\text{-}PRE\text{-}PREPARE,v,t,d\rangle\sigma_n,m\rangle$。其中 *M-PRE-PREPARE* 为主节点提出的消息标识,v 为视图编号,t 为时间戳,σ_n 为主节点 N 的签名,m 为客户端的请求消息,d 为 m 的摘要。

(1)组内共识 PRE-PREPARE 阶段。首先副主节点发出预准备阶段消息,消息格式为$\langle\langle PRE\text{-}PREPARE,v,t,d\rangle\sigma_{n1},m\rangle$。其中 *PRE-PREPARE* 是副主节点发出的消息标识,v 是当前视图编号,t 为时间戳,σ_{n1} 为副主节点 $N1$ 的签名,m 为客户端的请求消息,d 为 m 的摘要。

(2)组内共识 PREPARE 阶段。分组内的从节点接收到副主节点发出的预准备消息之后,对其进行验证,验证通过之后进行该阶段。此时各个节点进行组内广播消息,消息格式为$\langle\langle PREPARE,v,t,d\rangle\sigma_{ni}\rangle$。其中 *PREPARE* 为消息标识,$v$ 为当前视图编号,t 为时间戳,d 为 *PRE-PREPARE* 中接收到的消息的摘要,σ_{ni} 为发送该 *PREPARE* 消息节点的签名。当一个节点收到来自其他节点(包括自己)的 $2f_g+1$ 条正确的 *PREPARE* 消息之后,进入下一阶段,其中 f_g 为节点分组后各小组所能容纳的最大的拜占庭节点数量。

(3)组内共识 COMMIT 阶段。小组内的从节点编辑确认 *COMMIT* 消息,并将其广播至全组其他节点,消息格式为$\langle\langle COMMIT,v,t,d\rangle\sigma_{ni}\rangle$。其中 *COMMIT* 为消息标识,$v$ 为当前视图编号,t 为时间戳,d 为 *PRE-PREPARE* 中接收到的消息的摘要,σ_{ni} 为发送该 *COMMIT* 消息节点的签名。在广播的同时节点接收来自区块链中其他节点的消息,当节点收到来自其他节点(包括自己)的 $2f_g+1$ 条正确的 *COMMIT* 消息之后,完成组内共识,其中 f_g 为节点分组后各小组所能容纳的最大的拜占庭节点数量。

3)组间共识 PREPARE 阶段。各组在完成自身的组内共识之后,由副主节点向其他副主节点以及主节点进行广播,广播内容为$\langle\langle M\text{-}PREPARE,v,t,d\rangle\sigma_{ni}\rangle$。其中 *M-PREPARE* 为消息标识,$v$ 为当前视图编号,t 为时间戳,d 为消息的摘要,σ_{ni} 为发送该 *M-PREPARE* 消息节点的签名。当一个节点收到来自其他副主节点或者主节点(包括自己)的 $2f_g+1$ 条正确的 *PREPARE* 消息之后,进入下一阶段,其中 f_g 为节点分组后各小组所能容纳的最大拜占庭节点数量。

4)组间共识 COMMIT 阶段。在这个阶段内各组的副主节点进行 COMMIT 消息的广播,消息格式为$\langle\langle M\text{-}COMMIT,v,t,d,h\rangle\sigma_{ni}\rangle$。其中 *M-COMMIT* 为消息标识,$v$ 为当前视图编号,t 为时间戳,d 为消息的摘要,σ_{ni} 为发送该 *M-COMMIT* 消息节点的签名。在广播的同时,节点接收来自区块链中其他节点的消息,当节点收到来自其他节点(包括自己)的 $2f_g+1$ 条正确的 *COMMIT* 消息之后,完成组间共识,其中 f_g 为节点分组后各小组所能容纳的最大的拜占庭节点数量。

5)组间共识完成之后,网络中节点进行信用度的更新。节点把对应的评价消息发送到组内交互节点,广播内容为$\langle\langle CRE,v,t,d\rangle\sigma_n\rangle$。其中 *CRE* 为该信用度更新标识,$v$ 为当前视图编号,t 为时间戳,d 为对组内交互节点的评价,σ_{ni} 为发送该 *CRE* 消息的主节点的签名。当节点收到来自其他节点(包括自己)的 $2f_g+1$ 条正确的 *CRE* 消息之后,广播 *COMMIT* 消息$\langle\langle CRECOMMIT,v,t,d\rangle\sigma_n\rangle$。节点收到来自其他节点(包括自己)的 $2f_g+1$ 条正确的 *CRECOMMIT* 消息之后,通过 3.2.2 节中的信任动态调整机制公式计算组内节点的正面评价统计值 $Rise(x)$ 与激励信用值 $Motive(x)$,计算得到组内所有节点的基础信用度后将本组内的信用度表发送到主节点,由主节点进行归一化后广播。

4.3.4 视图切换

与 PBFT 一致,为了防止因主节点宕机或拜占庭行为而无法完成共识,网络中需要有视图切换协议,在从节点等待超时的情况下触发此协议,更新主节点,以保持系统的活性。虽然本文算法使用信任度模型减少了拜占庭节点成为主节点或者副主节点的概率,但仍然会出现拜占庭节点成为主节点的情况,所以本文考虑了视图切换协议,在主节点或者副主节点出现故障时启用,具体流程如下。

1)在网络中节点等待超时情况下,触发视图切换协议,此时各个正常节点进行广播,广播内容为$\langle\langle VIEW\text{-}CHANGE,v,t,m\rangle\sigma_{ni}\rangle$,其中 *VIEW-CHANGE* 为消息标识,$v$ 为当前视图编号,t 为时间戳,m 为新主节点的编号,σ_{ni} 为发送该视图切换消息节点的签名。新主节点的选取方式是在进行主节点选取时,主节点维护的优先级列表会广播出去,在主节点宕机之后选择优先级列表第二的节点成为主节点。

2)触发视图切换协议之后,更新视图为 $v+1$,当新视图的主节点接收到 $2f$ 个节点的视图切换消息之后,广播视图切换主节点切换成功的消息,消息格式为$\langle\langle VIEW\text{-}NEW,v,t\rangle\sigma_m\rangle$。其中 *VIEW-NEW* 为消息标识;$v$ 为切换后的视图编号,即原有视图+1;t 为时间戳;σ_m 为新视图主节点的签名。

3)网络内其他节点接收到新视图主节点的消息之后,进入新视图。

4.3.5 节点加入与退出

传统的 PBFT 并没有节点加入和退出阶段,其完全封闭的系统并不能实现在运行过程中进行节点的加入和退出,如果想要增加或者减少节点,整个系统需要暂停之后再对节点进行操作。这一特性使得传统的 PBFT 适合私有链以及联盟链而不是公有链。本文算法考虑到了这一点,相比传统的 PBFT 引入了节点的加入与退出协议,该协议分为 3 个部分,即节点的主动加入、节点的主动退出以及节点的被动退出。

1)节点的主动加入

当一个节点想要加入区块链网络时,它需要向网络中的某个节点发送请求加入消息,消息格式为$\langle\langle JOIN,t,IP,PK\rangle\sigma_n\rangle$,消息包含该节点的 IP、公钥、签名以及时间戳。当网络中的节点接收到该消息之后,将消息内容存储到自身消息池中,在一次共识结束之后,节点将加入的消息广播到网络中其他节点处,广播的消息包含该节点的各项消息以及本节点的签名。接收到这个消息的节点在验证完消息的正确性后在本地

保存新加入节点的数据消息,同时向该节点发送〈〈*JOIN-SUC*,*t*,*IP*,*PK*〉σ_n〉消息,该消息包含了网络中节点自身的公钥信息,使得新节点能够在本地进行节点信息保存,确保共识过程的顺利进行。具体流程如算法 3 所示。

算法 3 节点加入算法

输入:〈〈JOIN,t,IP,PK〉σ_n〉

输出:〈〈JOIN-SUC,t,IP,PK〉σ_n〉

1. Broadcast Joining Message 〈〈 JOIN,t,IP,PK 〉σ_n〉
2. for x∈L do
3. verify and save
4. end for
5. Broadcast Message of Joining Success
6. return〈〈JOIN-SUC,t,IP,PK〉σ_n〉

2)节点的主动退出

节点在网络中也可进行退出操作。当一个节点 n 需要退出网络时,它需要向网络内的所有节点进行广播,广播内容为〈〈*QUIT*,*t*,*IP*,*PK*〉σ_n〉其包含了该节点的 IP、公钥、签名以及时间戳消息。当其他节点接收到该退出消息之后,将消息保存到消息池中,在一次共识结束之后,节点对退出消息进行验证,验证通过后将发送确认退出消息给退出节点,消息格式为〈〈*QUIT-SUC*,*t*〉σ_n〉。当退出节点收到来自网络中 $2f$ 个节点的确认退出消息之后,方可完成退出。

节点完成退出之前,需要继续参与共识,即节点无法在共识过程中退出,必须在两次共识之间的时间点退出网络。当有节点退出之后,各发出确认该节点退出网络的节点在自身本地的信息存储池里将该节点的消息删去,避免在下一次共识中向该退出节点发送消息。具体流程如图 2 与算法 4 所示。

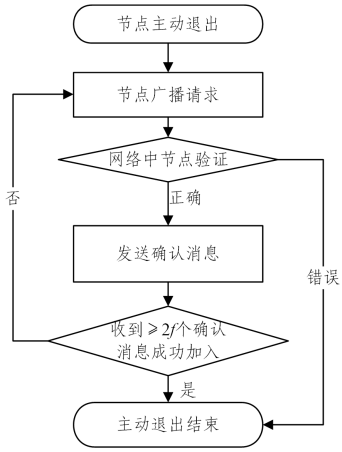

图 2 节点主动退出流程图

Fig. 2 Flowchart of nodes quit

算法 4 节点主动退出算法

输入:〈〈QUIT,t,IP,PK〉σ_n〉

输出:New Storage Pool

1. Broadcast Quit Message 〈〈 QUIT,t,IP,PK〉σ_n〉
2. for x∈L do
3. Broadcast 〈〈QUIT-SUC,t〉σ_n〉
4. end for
5. if x_n receive more than 2f Message then
6. Quit the Blockchain
7. end if
8. Return NewStoragePool

3)节点的被动退出

除了节点加入、退出的主动行为,网络还设计了节点的被动退出协议。该协议是针对作恶节点的,当一个节点宕机、拒绝共识或者进行拜占庭行为,其信任度会下降。节点信任度评估降低到 Low 之后,在新一轮共识开始之前执行该节点被动退出协议,此时网络所有节点筛选出信任度为 Low 的节点,广播消息〈〈*FIRE*,*t*,*F*〉σ_n〉给网络中的全部节点,其中 F 为信任度低节点的信息集合。当一个节点收到 2/3 信任度为 Middle 和 High 节点数量的 *FIRE* 消息之后,完成对节点的被动退出,此时网络中节点将集合 F 中的节点信息删去。具体流程如算法 5 所示。

算法 5 节点被动退出算法

输入:List of Nodes:L

输出:New Storage Pool

1. for x∈L do
2. Select Low Nodes
3. Broadcast 〈〈FIRE,t,F〉σ_n〉
4. if x_i receive more than 2/3 Messages from Middle and High Nodes
5. Update Storage Pool
6. end if
7. end for
8. Return New Storage Pool

4.4 实验分析

4.4.1 实验环境

为了评估算法的性能,需要对算法进行实验分析,本文使用 Go 语言对算法进行模拟仿真。在网络中,根据节点数量的变化以及分组的多少两个变量,对本文算法的吞吐量(TPS)和时延进行分析。对比的算法包括传统的 PBFT,HotStuff 算法以及 Paxos 算法。其中 HotStuff 算法为常用的事件驱动类型,Paxos 算法仅作为分布式共识算法与拜占庭容错算法进行对比。为了方便区分,本文制图时将算法缩写成 BC-PBFT,即基于贝叶斯理论的信任模型拜占庭容错算法。实验环境如表 2 所列,区块大小为 2 000 Bytes,交易注入速度为2 000 ms,总共注入的交易数量为 160 000,触发视图切换协议的超时间隔为 20 000 ms。

表 2 实验环境

Table 2 Experimental environment

Category	Version
Operating System	Windows 10
RAM	16 GB
CPU	Intel i5-12900k
Go	1.21.3
Simulation Platform	BlockEmulator

4.4.2 安全性分析

安全性分析,即如何抵抗拜占庭节点的攻击以及保证系统的稳定性。其中拜占庭节点的攻击分为对共识过程的攻击以及对信任度评估的攻击。

1)对共识过程的攻击

对于拜占庭节点对共识过程的攻击,在 4.3 节中提到了网络安全性的假设,这使得拜占庭节点对共识过程的攻击局

限为计算错误信息，阻挠网络完成共识。算法在组间共识与组内共识都采用 PBFT 算法，具有 $f=\lfloor(N-1)/3\rfloor$ 个拜占庭节点的容错性，在拜占庭节点数量较少的情况下仍能保证网络完成共识，但不排除某个组内拜占庭节点过多的情况，节点过多会阻挠网络达成组内共识。为应对这种问题，本文算法使用分层结构的设计，使得当下层某个组出现状况时，上层组间共识仍有极大概率保证网络成功达成共识，从而保证系统的稳定性。

2)对信任度评估的攻击

拜占庭节点对信任度评估的攻击方式是：作为恶意节点的主节点，通过恶意降低正常节点的可信度或提高其他恶意节点的可信度，使得正常节点被动退出网络，导致网络被恶意节点控制；恶意节点在网络信任度评估阶段进行阻挠。在信任度评估环节中，节点之间需要进行额外通信以做到对交互节点的信任度评估，可以理解成这也是一个由主节点发起的信任度共识阶段，节点之间进行额外通信以确保自身与交互节点之间的信任度评估计算是正确的。若拜占庭节点在此时加以干预，会使得该节点计算得到的信任度表与其他节点不一致，主节点可通过简单的比对得到拜占庭节点所提交的错误打分信息，因此得到对应的反应，排除干扰计算正确的信任度结果。而在主节点作为恶意节点的情况中，若主节点拒绝参与信任度共识，在超过视图切换协议时间间隔后正确的节点可以发起视图切换协议，以避免恶意节点带来的影响。若主节点在归一化时进行干扰，则会使得组内信任度错乱，但发生概率是可控的。在第二轮共识之后，领导节点是诚实节点的概率逐渐变高趋近于 1，上述情况发生的概率会逐步变小。

除上述拜占庭节点攻击外，系统的稳定性也在安全性分析范围中，其中系统的稳定性分析如下。

改进后的 PBFT 算法新增了信任度评估机制，进一步增强了领导节点的诚实概率。随着共识过程的进行，节点积分不断变化，信任度越高意味着节点越可靠，其成为领导节点的概率就越高，从而有效避免了领导节点出错的可能性。这是系统稳定性的重要指标。这里选取 Monte Carlo 方法进行实验分析。假设重复实验次数为 m，每次实验的共识次数为 n，领导者节点为诚实节点的频率为 f，每次重复实验中被选中的领导者节点为诚实节点的概率 $P(A)=\frac{f}{n}$，以 m 次重复实验概率的均值作为最终的期望概率，则所选领导节点为诚实节点的概率可作为领导节点诚实概率的近似。实验结果如图 3 所示，显然，$P(BC\text{-}PBFT)>P(PBFT)$，说明信任度评估机制可以增加领导者节点是诚实节点的概率，有效降低领导者节点出错的可能性，使得整个系统更加安全稳定。

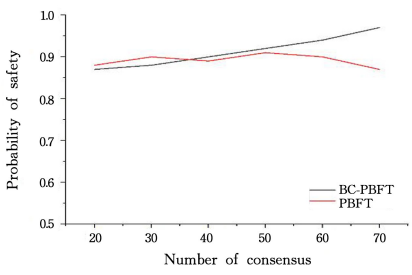

图 3 领导节点正确性概率

Fig. 3 Correctness probability of leader node

4.4.3 TPS 分析

吞吐量代表区块链系统在单位时间内处理事务的能力，一般用每秒交易数(Transaction Per Second，TPS)表示。区块链系统网络中交易吞吐量指，交易发出到交易确认并写入区块链中总交易数与消耗的时间的比值。区块链的吞吐量反映了区块链系统的事务处理速度，计算式如下：

$$TPS=\frac{Transaction_{\Delta t}}{\Delta t} \tag{12}$$

其中，$Transaction_{\Delta t}$ 表示在 Δt 时段之内的事务处理数量。

从图 4 中可以看到：在默认节点分组 $k=2$ 的情况下，随着节点数量增加，传统 PBFT 因节点间需要成对通信，且缺乏恶意节点退出机制，吞吐量下降幅度最大；HotStuff 算法采用星型拓扑通信来抵御恶意节点，受硬件资源和设备的限制较大，且缺乏针对恶意节点的退出机制，导致实验中吞吐量较低且下降趋势明显；Paxos 算法依靠单个主节点收集和分发消息，随着系统节点数增加，主节点的负载压力增大，导致吞吐量不断下降。PBFT，Paxos 和 HotStuff 算法都表现出明显的吞吐量下降趋势，而本文算法在节点数量增加的情况下，吞吐量也有下降趋势，但其因进行了分组，同时利用信任机制减少了恶意节点作为主节点的情况，一定程度上减少了恶意主节点导致的视图变换，下降速度较为缓慢。

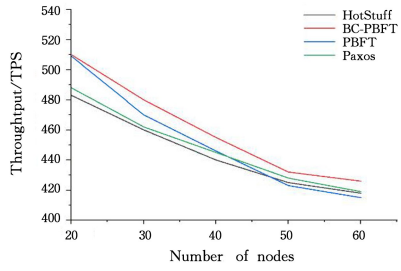

图 4 不同节点数量下的算法吞吐量对比

Fig. 4 Comparison of algorithm throughput under different number of nodes

本文同时考虑了不同分组情况下算法吞吐量的对比，从图 5 中可以看到：节点数量为 20 左右时，分组越多，其吞吐量越高，但此时不同分组下吞吐量之间的差距不明显；随着节点增多，不同分组数量情况下的区别越来越明显，三者之间的差异越来越大，分组越多，其吞吐量随着节点增多下降越慢，即在相同的节点数量情况下，分组越多其吞吐量越高，且不同分组之间的差距愈发明显。正是由于在节点数量增多的情况下，分组多使得各个小组之内的节点数量较少，因此减少了节点之间相互通信的开销。

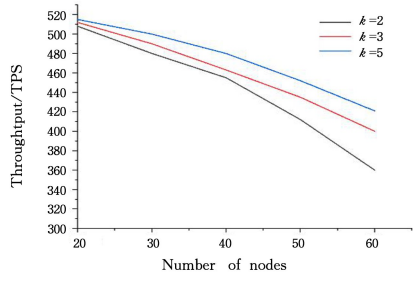

图 5 不同节点分组下的算法吞吐量对比

Fig. 5 Comparison of algorithm throughput under different grouping of nodes

4.4.4 时延分析

区块链系统中，时延是指从交易提交到交易完成之间所消耗的时间，也是完成一次共识的时间。它是衡量网络性能和共识算法运行时间的标准，时延越低，则可更快确认交易。

PBFT 以信息交换为基础，需要消耗通信资源。因此，通信开销是与共识效率相关的关键指标。

假设 N 表示参与共识的节点数。在 PBFT 的共识过程中，所有节点在准备阶段和提交阶段都需要在整个网络中广播。在这种情况下，每个节点所需的通信次数为 $N-1$。假设完成一轮 PBFT 共识所需的通信次数为 X，可得：

$$X=2N(N-1) \tag{13}$$

其中，预准备阶段为 $N-1$，准备阶段为 $(N-1)^2$，提交阶段为 $N(N-1)^2$。

本文算法将节点分为 k 组，则组内共识所需的通信次数 W 为：

$$W=2\frac{N}{k}\left(\frac{N}{k}-1\right) \tag{14}$$

组间共识所需的通信次数则为 $2k(k-1)$，可得 BC-PBFT 的通信开销 Y 为：

$$Y=2\frac{N}{k}\left(\frac{N}{k}-1\right)+2k(k-1) \tag{15}$$

Z 用于表示 PBFT 和 BC-PBFT 在通信次数上的比值，计算式如下：

$$Z=\frac{N(N-1)}{\frac{N}{k}\left(\frac{N}{k}-1\right)+k(k-1)} \tag{16}$$

随着 N 和 k 的不断增大，Z 值会越来越大，直至一个最高点。原因是分组数量的增加，导致每个分组中的节点数量减少。BC-PBFT 的分层有效减少了在小范围内达成共识所需的通信次数。当 k 增大到极值点时，Z 达到最大值，然后开始减小。此时，k 值过大会导致分组过多，共识所需的组间通信次数显著增加。PBFT 和 BC-PBFT 的通信开销对比如表 3 所列。

表 3 通信开销对比

Table 3 Comparison of communication cost

	PBFT	BC-PBFT Intra-group	BC-PBFT Inter-group
Pre-prepare	$N-1$	$N/k-1$	$k-1$
Prepare	$(N-1)^2$	$(N/k-1)^2$	$(k-1)^2$
Commit	$N(N-1)$	$N/k(N/k-1)$	$k(k-1)$
Total	$2N(N-1)$	$2N/k(N/k-1)+2k(k-1)$	

使用 BlockEmulator 仿真平台可以得到对应算法的时延数据，通过计算共识出块中时延的总时间以及其中注入交易的数量得到交易确认时延 $totLatency=latencySum/totTxNum$。从图 6 中可以看到，时延随着节点数量的增加而增加，4 种算法在开始时的时延相差不大，但当节点数达到 30 时，由于节间通信量的增加，PBFT，Paxos 和 HotStuff 算法的时延开始迅速增加，PBFT 与 Paxos 的时延远大于其他两种算法。其中，Paxos 算法的通信复杂度低于 PBFT 算法的通信复杂度，因此其时延低于 PBFT 算法，而 HotStuff 算法由于采用了链式共识方法和多主节点共识，延迟低于 Paxos 算法。本文改进的算法由于使用了分组通信的方式，在节点变多的情况下仍然保持着较低的通信开销，其时延虽然随着节点增多而增加，但总体变化较为稳定。

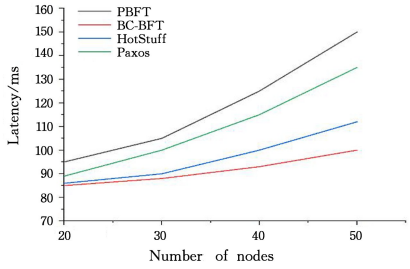

图 6 不同算法的时延对比

Fig. 6 Comparison of time delay between different algorithms

4.4.5 节点加入与退出分析

本文算法在 PBFT 的基础上新增节点加入与退出机制，使得共识网络更具可拓展性。网络需要进行额外通信来执行节点的加入以及退出协议。把节点数量固定在 40，每一次共识结束后有 5 个节点退出，5 个节点加入，共进行 10 次实验，得到 PBFT、BC-PBFT 与删去节点加入与退出机制的 BC-PBFT(BC-PBFT-2)3 种算法之间的时延对比，结果如图 7 所示。可以看出，新增节点加入与退出算法机制后，算法的时延比没有加入该机制的情况更高，虽牺牲了部分时延来获得更高的拓展性，但仍在可以接受的范围之内。

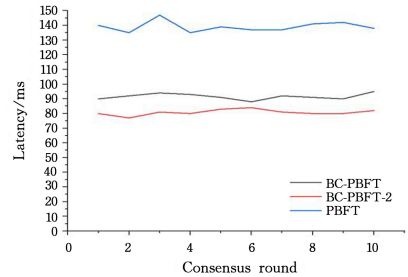

图 7 节点加入与退出的影响

Fig. 7 Effect of node joining and leaving

结束语 共识算法的优劣影响着区块链网络的性能，可以说，共识算法是区块链网络的核心机制。但目前区块链共识算法存在诸多问题，例如吞吐量低、时延高等问题。本文对区块链中的实用拜占庭容错算法(PBFT)进行改进，引入基于贝叶斯理论的动态信任模型，解决了 PBFT 不能进行分组共识以及控制节点动态加入退出网络的问题，提高共识效率的同时增强了系统的鲁棒性。

本文提出基于贝叶斯理论的动态信任拜占庭容错算法。首先引入贝叶斯理论，设计了基于贝叶斯理论的信任模型，提出信任模型中的信任动态调整机制，完善了节点在网络中信任度的变化计算公式，将节点在网络中进行共识的行为正确与否作为信任度变化的重要标准；同时设计信任度等级机制，每次共识均由中高信任度的节点进行，将低信任度的节点排除，禁止其参与共识，降低了系统在共识过程中被拜占庭节点攻击的概率，提高了系统总体的鲁棒性。此外，本文依靠提出的基于贝叶斯理论的信任模型，提出了基于贝叶斯理论的动态信任拜占庭容错算法，在算法模型设计方面，以共识过程的不同时段进行划分，辅以算法分点描述了包括主节点选取、

节点分组、共识过程、视图切换以及节点加入退出5个过程，每个过程都与提出的信任模型息息相关。最后通过实验验证了所提算法的有效性，改进后的算法不仅继承了传统PBFT的优点，也在节点的加入退出以及共识分组方法上有所创新，即依托基于贝叶斯理论的信任模型实现对网络中主节点选取和分组共识的调控，增强系统稳定性。

本文算法仍存在改进的空间。在分组方面，分组数量的值为实验前设定的，当节点数量过少时，分组的值若设置太大，则会使得某些分组不安全，即内部拜占庭节点过多，影响整个共识系统的安全性；而分组的值设置较小也存在弊端，在网络中节点数量越来越多的情况下，过小的分组值会使得分组后的效果不佳。未来的研究将改进为根据网络中节点的数量设置动态分组值。在实验设计方面，后续将使用模拟跨地理区域网络来进行模拟实验，更好地体现系统的可拓展性。

参 考 文 献

[1] SHAO Q F, JIN C Q, ZHANG Z, et al. Blockchain technology: Architecture and progress [J]. Journal of Computer Science, 2018, 41(5): 969-988.

[2] BELOTTI M, BOZIC N, PUJOLLE G, et al. A vademecum on blockchain technologies: When, which and how[J]. IEEE Communications Surveys Tutorials, 2019, 21(4): 3796-3838.

[3] CHEN J, HE K, LI K, et al. Block chain extension technology present situation and prospect [J]. Journal of Software, 2024, 35(2): 828-851.

[4] WRIGHT C S. Bitcoin: A peer-to-peer electronic cash system [EB/OL]. https://nakamotoinstitute.org/library/bitcoin/.

[5] ZHAO Y, ZHANG M, PEI Z, et al. The effects of quantitative easing on bitcoin prices[J]. Finance Research Letters, 2023, 57: 104232.

[6] SI X M, CHEN W G. Chain blocks and digital currency technology project introduction [J]. Journal of Software, 2019, 30(6): 1575-1576.

[7] AGBO C C, MAHMOUD Q H, EKLUND J M. Blockchain technology in healthcare: A systematic review[J]. Healthcare, 2019, 7(2): 56.

[8] TSE D, ZHANG B, YANG Y, et al. Blockchain application in food supply information security[C]// 2017 IEEE International Conference on Industrial Engineering and Engineering Management(IEEM). IEEE, 2017: 1357-1361.

[9] XU Z, LIANG W, LI K C, et al. A blockchain-based roadside unit-assisted authentication and key agreement protocol for internet of vehicles[J]. Journal of Parallel and Distributed Computing, 2021, 149: 29-39.

[10] BOURAGA S. A taxonomy of blockchain consensus protocols: A survey and classification framework[J]. Expert Systems with Applications, 2021, 168: 114384.

[11] CASTRO M, LISKOV B. Practical byzantine fault tolerance [C]// OSDI. 1999: 173-186.

[12] GUPTA S, RAHNAMA S, HELLINGS J, et al. ResilientDB: Global scale resilient blockchain fabric[C]// Proceedings of the VLDB Endowment. 2020: 868-883.

[13] KOKORIS-KOGIAS E, JOVANOVIC P, GASSER L, et al. Omniledger: A secure, scale-out, decentralized ledger via sharding [C]// 2018 IEEE Symposium on Security and Privacy (SP). IEEE, 2018: 583-598.

[14] SYTA E, JOVANOVIC P, KOGIAS E K, et al. Scalable bias-resistant distributed randomness [C]// 2017 IEEE Symposium on Security and Privacy(SP). IEEE, 2017: 444-460.

[15] MICALI S, RABIN M, VADHAN S. Verifiable random functions[C]// 40th Annual Symposium on Foundations of Computer Science. IEEE, 1999: 120-130.

[16] GUETA G G, ABRAHAM I, GROSSMAN S, et al. Sbft: A scalable and decentralized trust infrastructure[C]// 2019 49th Annual IEEE/IFIP International Conference on Dependable Systems and Networks(DSN). IEEE, 2019: 568-580.

[17] YIN M, MALKHI D, REITER M K, et al. HotStuff: BFT Consensus in the Lens of Blockchain[J]. arXiv: 1803.05069, 2018.

[18] XU G, BAI H, XING J, et al. SG-PBFT: A secure and highly efficient distributed blockchain pbft consensus algorithm for intelligent internet of vehicles[J]. Journal of Parallel and Distributed Computing, 2022, 164: 1-11.

[19] FENG L, ZHANG H, CHEN Y, et al. Scalable dynamic multi-agent practical byzantine fault-tolerant consensus in permissioned blockchain[J]. Applied Sciences, 2018, 8(10): 1919.

[20] ZHENG X, FENG W, HUANG M, et al. Optimization of pbft algorithm based on improved c4.5[J]. Mathematical Problems in Engineering, 2021, 2021(1): 5542078.

[21] LIU S, ZHANG R, LIU C, et al. Improvement of the PBFT algorithm based on grouping and reputation value voting [J]. International Journal of Digital Crime and Forensics, 2022, 14(3): 1-15.

[22] NEO white paper[EB/OL]. http://docs.neo.org/en-us/.

[23] BERNARDO J M, SMITH A F. Bayesian theory: Vol. 405[M]. John Wiley & Sons, 2009.

[24] WANG Y, VASSILEVA J. Bayesian network-based trust model [C]// Proceedings IEEE/WIC International Conference on Web Intelligence(WI 2003). IEEE, 2003: 372-378.

[25] MICALI S, RABIN M, VADHAN S. Verifiable random functions[C]// 40th Annual Symposium on Foundations of Computer Science. IEEE, 1999: 120-130.

PAN Yanyang, born in 2000, postgraduate. His main research interests include blockchain and distributed system consensus algorithm optimization.

JI Qingge, born in 1966, Ph.D, associate professor, master supervisor, is a senior member of CCF(No. 07014S). His main research interests include computer graphics, virtual reality, computer vision, computer simulation and blockchain.

（责任编辑：何杨）