面向学位/学历证书可信管理的可扩展类PBFT算法

张学旺[1,2]，雷志滔[1]，林金朝[3]
（1.重庆邮电大学软件工程学院，重庆 400065；2.重庆大学微电子与通信工程学院，重庆 400004；
3.重庆邮电大学光电信息感测与传输技术重庆市重点实验室，重庆 400065）

摘　要：现有的学位/学历证书可信管理存在节点扩展的优化策略不足、未考虑节点差异性和吞吐量低等问题，“区块链+教育”为学位/学历证书的可信管理提供了一种解决方案。针对上述问题，提出一种面向学位/学历证书可信管理的可扩展类PBFT算法z-PBFT。该算法采用基于区域分组的分层设计，算法模型分为主区域和副区域节点簇，副区域内部采用局部共识机制；通过基于TOPSIS建模和熵值赋权法的加权随机选取分派策略评估节点性能，并选取共识委员节点集。实验结果表明，在大规模节点应用场景下，该算法在确保安全性的同时，具有更高的吞吐量和可扩展性。

关键词：z-PBFT共识算法；学位/学历证书；可扩展性；区块链
中图分类号：TP301.6
文献标志码：A
DOI：10.11959/j.issn.1000–436x.2024144

Scalable PBFT-like algorithm for trust management of degree/graduation certificates

ZHANG Xuewang[1,2], LEI Zhitao[1], LIN Jinzhao[3]
1. School of Software Engineering, Chongqing University of Posts and Telecommunications, Chongqing 400065, China
2. College of Microelectronic and Communication Engineering, Chongqing University, Chongqing 400004, China
3. Chongqing Key Laboratory of Photo Electronic Information Sensing and Transmitting Technology, Chongqing University of Posts and Telecommunications, Chongqing 400065, China

Abstract: “Blockchain+education” offers a solution for trust management of degree/graduation certificates. Existing methods for managing degree/graduation certificates suffer from issues such as lacked optimization strategies for node expansion, neglect of node heterogeneity, and low through put. A scalable PBFT-like algorithm for trust management of de-gree/graduation certificates was proposed to address the above issues, called z-PBFT. The algorithm was based on a hierarchical design using regional grouping, with a model consisting of main region and sub-region node clusters. Local consensus was employed within the sub-regions. A weighted random allocation strategy based on TOPSIS modeling and entropy weighting was used to evaluate node performance and select a consensus committee node set. Analysis and simulation experiments demonstrate that the proposed algorithm ensures security while achieving higher throughput and scalability in large-scale applications of degree/graduation certificate trust management.

Keywords: z-PBFT consensus algorithm, degree/graduation certificate, scalability, blockchain

0　引言

近年来，越来越多的公司将文凭证书作为重要的入职门槛，但当前的学位/学历证书管理仍面临信息“孤岛”、证书可信度低、学生信用体系不完善以及缺乏历史数据信息链等问题，这些问题导致高校学生无法便捷地出示可信的学位/

收稿日期：2024–02–29；**修回日期：**2024–06–07
基金项目：国家自然科学基金资助项目（No.U21A20447）
Foundation Item：The National Natural Science Foundation of China (No.U21A20447)

学历证书数据并享受相应服务。同时，由于缺乏有效的验证手段，导致用人单位和高校之间的信息不对称，使双方的信任度进一步降低[1]。此外，大量伪造学位/学历证书的现象已经威胁到学位/学历证书所有者和颁发机构的权益。根据《福布斯》的一项研究，伪造证书已形成了一个规模高达 70 亿美元的产业[2]。因此，实施可靠和安全的学位/学历证书可信管理已成为一项刻不容缓的任务。

随着区块链技术的兴起，其去中心化、防篡改、匿名性和透明性等特征[3]正在推动金融服务[4]、智慧能源[5]、数字存证[6]等多个行业的变革。同时，区块链技术也为教育领域的学位/学历可信管理提供了一个创新且可靠的解决方案。然而，现有基于区块链的学位/学历可信管理方案在可扩展性方面仍存在不足[7]。共识算法是区块链的核心技术之一[8]，高效的共识算法能够提升区块链系统的可扩展性。目前，主流的区块链共识算法包括工作量证明（PoW, proof of work）算法[9]、权益证明（PoS, proof of stake）算法[10]、代理权益证明（DPoS, delegate proof of stake）算法[11]和实用拜占庭容错（PBFT, practical Byzantine fault tolerance）算法[12]等。目前主流的“区块链+教育”的学位/学历可信管理使用的共识算法比较如表 1 所示。

尽管 PoS 共识算法对节点规模的影响较小，但存在吞吐量较低的问题。Kafka 集群[19]和 PBFT 都是 Hyperledger Fabric 区块链平台采用的共识算法。其中，Kafka 集群存在单点故障问题，无法解决拜占庭问题；而基于 PBFT 的区块链平台在大规模节点（超过 100 个节点）[20]的场景下难以实现扩展，从而严重影响整个平台的性能。虽然“区块链+教育”将区块链技术与学位/学历证书管理相结合，使学位/学历证书具备可视、可查、可追溯的特点，但区块链共识算法的局限性给基于区块链技术的学位/学历证书管理带来了挑战。在基于区块链技术的学位/学历证书管理中，存在以下主要问题：1) 不同高校之间的节点距离和服务器性能差异；2) 随着区块链节点（高校）数量的增加，学位/学历证书管理将出现频繁的证书操作，导致低吞吐量、高交易时延和高通信开销等问题。

本文提出一种面向学位/学历证书可信管理的可扩展类 PBFT 算法 z-PBFT（zone-practical Byzantine fault tolerance），以解决学位/学历证书可信管理系统所面临的可扩展性问题。该算法基于区域分组的分层设计对拜占庭容错共识算法进行改进，主要贡献如下。

1) 提出面向学位/学历证书可信管理的 z-PBFT 算法模型，将系统分为主区域和副区域节点簇。该模型将高校节点按省份划分为多个副区域，主区域由各副区域的主节点组成映射集合，实现副区域之间的数据同步。

2) z-PBFT 算法采用区域分组的分层共识机制，通过权重机制加权选取副区域的共识委员节点，使得每个副区域仅需达成局部共识并进行全网广播，即可完成区域内部共识。该方法显著减少了在大规模节点场景下参与共识流程的节点数量，降低了网络通信的复杂度，同时提高了节点的可扩展性。

3) 提出一种基于 TOPSIS（technique for order preference by similarity to an ideal solution）[21]建模和熵值赋权法的加权随机选取分派策略。该策略先对高校服务器节点进行评估和加权，随机选取共识委员节点集，再将其余节点分派给各共识委员节点集的成员。此方法能够有效减小服务器节点差异带来的影响，提高共识效率。

1　相关研究与技术分析

PBFT 算法为解决拜占庭故障问题提供了一种有效的方案[22]，在不超过 $\frac{n-1}{3}$ 个节点同时出现错误的情况下，能够确保 n 个节点系统的安全性和活

表 1　“区块链+教育”的学位/学历可信管理使用的共识算法比较

参考文献	区块链类型	平台	共识算法	吞吐量	受节点规模影响程度
文献[13-16]	公有链	Ethereum	PoS	10~20TPS	较小
文献[17]	联盟链	Hyperledger Fabric	PBFT	数千 TPS	较大
文献[18]	联盟链	Hyperledger Fabric	Kafka	数千 TPS	较大

性[23]。由于其性能好、交易确认时延低，并且基于密码学技术防范恶意行为，因此PBFT算法适用于联盟链场景。

PBFT的工作原理是通过状态机副本复制的方法，使分布式系统中的节点在存在故障或恶意节点的情况下达成对请求的一致性处理。在PBFT算法中，节点分为主服务节点和从属服务节点，主服务节点与从属服务节点之间通过多轮消息交换以及对消息的签名和验证，确保消息的正确性和完整性。当主服务节点p收到客户端的请求m后，执行一个3段协议将该请求广播到节点服务系统中。此3段协议包括预准备（pre-prepare）、准备（prepare）和提交（commit）3个阶段。其中，预准备和准备阶段用于对相同视图下的请求进行排序；准备和提交阶段则用于保证在不同视图中请求是完全有序的。服务节点按照顺序执行这些请求，从而保证系统的一致性。

图1展示了在4个服务节点下PBFT算法3段协议的运行示意，其中，C为客户端节点，节点1~节点3为从属服务节点，节点0为主服务节点，节点3为恶意节点。

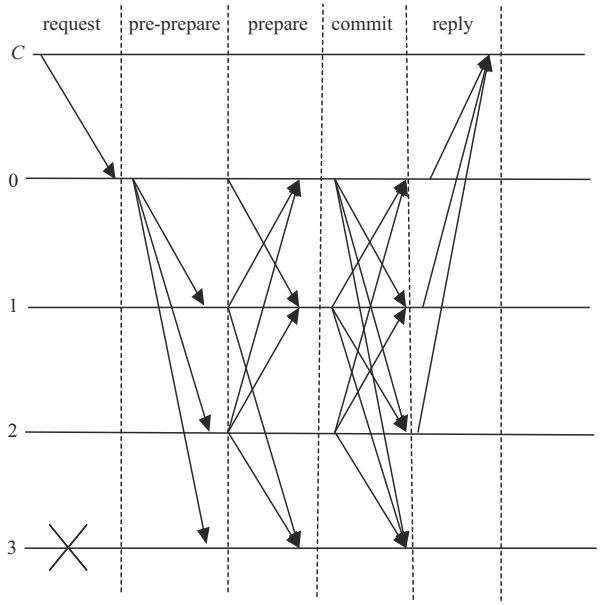

图1 PBFT协议运行示意

在请求（request）阶段，客户端节点C向系统的主服务节点0发出请求消息message，主服务节点0在收到请求消息message后，向节点1~节点3发送预准备消息<<pre-prepare,1,1,d>,message>，其中第一个1是视图编号，第二个1是视图的序列号，d是消息message的摘要。节点1和节点2在收到主服务节点0发来的预准备消息后，验证其合法性并接收该消息，然后分别向其他节点发送准备消息<prepare,1,1,d,1(2)>。此时，节点0~节点2满足准备就绪条件，完成准备阶段，开始向其他节点广播提交消息<commit,1,1,d,0(1/2),r>。最后在回复阶段（reply），客户端节点C收到来自3个节点的合法回复，r是其请求操作的执行结果。此时，系统从初始状态成功转移到执行message后的一致性状态。

PBFT算法具有两面性，一方面，它对区块链系统的数据一致性和安全验证至关重要；另一方面，在大规模节点场景下，其不可扩展性会对系统效率产生显著影响：1) 时延增加，需要更长的时间进行通信和完成共识，长时延会对应用造成负面影响；2) 网络开销增加，节点之间通信的网络开销增加，导致网络拥堵和资源受限问题。

针对PBFT算法的扩展性问题，已有多项研究进行了改进。文献[24]提出一种基于Raft集群的改进拜占庭共识算法，采用网络分片和双层级设计。首先，通过改进的Raft机制在节点内进行共识并选举领导者；然后，通过各组内选出的领导者组成网络委员会，委员会内部再采用PBFT机制进行共识，以提高可扩展性和共识效率。文献[25]提出一种DRBFT算法，通过引入随机选择RS（random selection）算法来减少参与共识过程的节点数量，从而提升区块链系统的扩展性。2019年，文献[26]提出HotStuff算法，该算法结合门限签名，将PBFT算法的共识消息互相广播的方式转变为由主节点处理、合并转发。相比于PBFT，HotStuff具有算法简单、网络复杂度与节点规模呈线性关系的优势。然而，HotStuff的复杂度仍与节点规模成正比，未能从根本上解决共识算法的可扩展性问题。此外，HotStuff在每个阶段都依赖于Leader的消息收集和广播，导致Leader成为每轮共识的瓶颈。文献[27]提出一种可扩展的多层PBFT机制，通过将节点进行分组并限制组内通信，降低了通信复杂度，从而提升PBFT区块链系统的可扩展性。文献[28]提出一种基于区块链的物联网系统的双层架构。该架构包括低层节点簇和高层虚拟簇，低

层节点簇通过虚拟叠加层相互连接，允许多个共识轮同时验证传入的数据块；高层虚拟簇负责对已验证的区块进行排序，并将其链接到复制的区块链总账中。该架构能够有效处理物联网中的大规模数据传输和验证操作，提升了系统的可扩展性和性能。文献[29]提出一种STBFT算法，该算法主要使用了树形拓扑网络、可验证随机函数（VRF, verifiable random function）[30]和反馈机制等技术。该算法将共识节点划分为不同的层和组，将全局共识转化为局部共识，以减少通信开销；通过树形拓扑网络结构设计系统模型，提高了系统的扩展性。文献[31]提出了P-PBFT算法，通过优化一致性协议，将网络节点分成不同的共识集，并实施分组共识，旨在解决现有药物可追溯性区块链技术的高时延、系统开销大和规模受限等问题。该算法侧重于提升基于区块链的药物可追溯性系统的效率和可扩展性。

针对PBFT算法节点可扩展性的研究可分为两大类：1) 通过选取部分节点作为共识节点，以减少参与共识的节点数量，从而降低通信量并提高算法效率；2) 改进共识结构，通过采用分组或分层的策略，将共识任务划分到不同的组或层中，以减少PBFT共识带来的通信压力。然而，在学位/学历证书可信管理中，由于节点间地理位置和服务器性能的差异，分组或分层策略未能充分考虑节点间传输网络的稳定性、节点间的距离以及服务器之间的性能差异。同时，当节点数量较多时，仅使用部分节点作为共识节点的改进方法容易导致中心化问题。z-PBFT算法由多个副区域和主区域（由每个副区域产生的主节点组成）构成一个双层网络架构。副区域和主区域分别进行共识，引入性能指标权重值作为选取节点类型的依据，高性能节点成为执行共识的主要节点，从而减小节点间距离和性能差异对共识效率的影响。此外，在多副区域结构中设计了随机选取共识委员节点的策略，避免了单一部分节点作为共识节点产生的中心化问题。

2　z-PBFT算法

z-PBFT算法将所有高校节点划分为主区域和副区域，不同区域根据节点规模采取不同的PBFT算法策略。

2.1　算法模型

z-PBFT算法将所有高校节点按区域划分为主区域和副区域，2种区域为相交关系，主区域的节点由各副区域选出，每个区域维护一条链。同时，将节点分为共识委员节点、验证节点和主节点。共识委员节点负责执行副区域的共识流程；验证节点仅进行区域内的区块验证而不参与副区域的共识流程；主节点从副区域中选出，负责主区域的共识流程。在收到证书操作请求后，副区域首先采用加权随机选取分派策略选出N_s个节点作为共识委员节点，其余节点作为验证节点并执行分组策略；然后，共识委员节点执行本轮PBFT共识算法，并在此过程中生成一个可验证的VRF随机数作为主节点选取的依据。共识委员节点完成共识后，执行多播策略以达成局部共识；最后，本轮周期选出的主节点将已确认的区块链数据上传至主区域。主区域节点再执行PBFT共识流程，达成区块一致并保存区块至本地，同时同步至所属副区域，从而达成全局共识。z-PBFT算法的网络结构和主要流程分别如图2和图3所示。其中，客户端表示向主服务节点发送证书操作请求的节点。客户端发送的证书操作请求格式为< REQUEST,*o*,*t*,*c* >，其中，*o*是客户端请求主服务节点执行的操作，*t*是时间戳，*c*是客户端编号。客户端发出的请求是有序的，即越早发出的请求其*t*值越小。

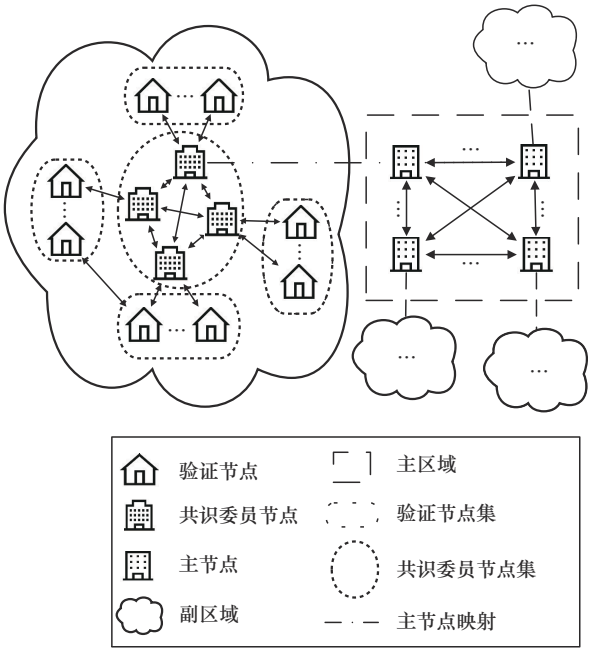

图2　z-PBFT算法的网络结构

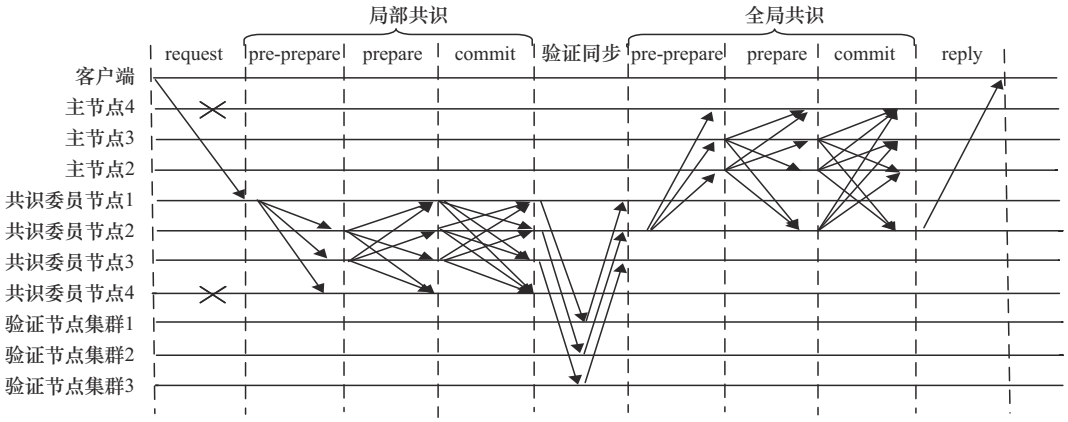

图3　z-PBFT算法的主要流程

2.2　相关术语

本节列出了相关的主要术语，如表2所示。

表2　主要术语

符号	符号说明
i	节点在副区域网络中的编号
N_i	副区域网络中第i个节点
W_i	副区域网络中第i个节点的总权重
T_{proci}	副区域网络中第i个节点的处理时延指标
M	传输数据量
d	数据大小，单位为bit
u_d	平均处理速度，单位为bit/s
T_{propi}	副区域网络中第i个节点的传输时延指标
u_t	平均传输速度，单位为bit/s
S_i	副区域网络中第i个节点的节点验证和共识率指标
num_c	共识完成数
num_v	验证完成数
num_f	共识和验证失败数
C_i	副区域网络中第i个节点的存储容量指标
V	节点权重集合
N_s	共识委员节点数
N_i	共识委员节点集中第i个节点
V_s	验证节点数
N_{vi}	验证节点集中第i个节点
m	副区域节点数
D_w	加权随机选取分派方法产生的阈值
G	验证节点集字典
L_{si}	共识委员节点N_{si}的组维护列表
U	视图编号
p_m	证书消息
p_n	证书消息编号
p_d	证书消息摘要
nbh	当前最高区块的哈希值
$\text{sealer}_{\text{si}}$	副区域执行共识流程编号为si的领导节点
g	洗牌算法产生的随机数

2.3　共识委员选取

2.3.1　加权随机选取分派策略

区块链节点的性能通常包括处理速度、带宽和稳定性等，节点性能越好，对整个区块链网络的贡献越大。考虑各高校在学生容量、信息化建设预算等方面的差异，区块链节点服务器的性能需求也会有所不同，在这种实际情况下，为了使参与证书提议共识流程的高校节点贡献最大化，需要基于节点差异性制定共识委员的选取策略，以达到正确协调整个区块链网络服务的目的，提高整个区块链网络的效率和稳定性。

本文提出一种基于TOPSIS建模和熵值赋权法的加权随机选取分派策略，通过评估各节点与理想解之间的距离，为每个节点分配一个理想贴近度，从而确定节点的最优性。将计算得到的理想贴近度作为最终权重，并发送给上一轮的主节点。主节点通过加权随机选取方法，选出目前最优的节点作为共识委员节点。

首先计算高校节点的各个指标权重W_i，利用节点N_i的处理时延T_{proci}、传输时延T_{propi}、存储容量C_i、节点验证和共识率S_i等指标进行计算。其中，T_{proci}由式(1)计算得到，T_{propi}由式(2)计算得到，S_i由式(3)计算得到，C_i是节点N_i的剩余存储容量（单位为TB）

$$T_{\text{proci}} = \sum_{j=1}^{M} \frac{d_i}{u_d} \tag{1}$$

$$T_{\text{propi}} = \sum_{j=1}^{M} \frac{d_i}{u_t} \tag{2}$$

$$S_i = \frac{\text{num}_c + \text{num}_v}{\text{num}_c + \text{num}_v + \text{num}_f} \tag{3}$$

为了解决缺少性能指标间的比较、单位不统一和主观性权重赋值等问题，需要对各个指标（T_{proci}、T_{propi}、C_i、S_i）进行全面评估，并计算出理想的最终权重，该过程有助于提高评价的准确性和可信度。本文基于TOPSIS建模和熵值赋权法，建立了节点评价方法。

1) 建立正向矩阵$\boldsymbol{X}$，x_{m4}为第m个节点的第4个指标

$$\boldsymbol{X}=\begin{bmatrix} x_{11} & x_{12} & x_{13} & x_{14} \\ x_{21} & x_{22} & x_{23} & x_{24} \\ \vdots & \vdots & \vdots & \vdots \\ x_{m1} & x_{m2} & x_{m3} & x_{m4} \end{bmatrix} \tag{4}$$

2) 矩阵标准化。记$\boldsymbol{X}$的标准化矩阵为$\boldsymbol{Z}$，计算$\boldsymbol{Z}$及其第i个节点的j指标的标准化值

$$\boldsymbol{Z}=\begin{bmatrix} z_{11} & z_{12} & z_{13} & z_{14} \\ z_{21} & z_{22} & z_{23} & z_{24} \\ \vdots & \vdots & \vdots & \vdots \\ z_{m1} & z_{m2} & z_{m3} & z_{m4} \end{bmatrix} \tag{5}$$

$$z_{ij}=\frac{x_{ij}}{\sqrt{\sum_{i=1}^{m}x_{ij}^2}} \tag{6}$$

其中，j=1,2,3,4。

节点指标异常可能使计算结果产生较大偏差，因此，通过标准化处理可以减少异常值的影响，提高系统的鲁棒性。

3) 计算第i个节点的j指标占总指标的比例p_{ij}

$$p_{ij}=\frac{z_{ij}}{\sum_{i=1}^{n}z_{ij}} \tag{7}$$

其中，j=1,2,3,4。

4) 计算指标j的熵值e_j

$$e_j=-\frac{1}{\ln m}\sum_{i=1}^{m}\ln\left(p_{ij}\right) \tag{8}$$

其中，$e_j>0$，j=1,2,3,4。

5) 计算第j项指标的差异系数df_j

$$\mathrm{df}_j=1-e_j \tag{9}$$

6) 计算第j项指标的初步权重w_j

$$w_j=\frac{\mathrm{df}_j}{\sum_{j=1}^{4}\mathrm{df}_j} \tag{10}$$

7) 计算正理想解Z^+，其表示在每个指标上都取得最大值的最优候选方案

$$\begin{aligned} Z^+=&(Z_1^+,Z_2^+,Z_3^+,Z_4^+)= \\ &\left(\max\left\{z_{11},z_{21},\cdots,z_{m1}\right\},\max\left\{z_{12},z_{22},\cdots,z_{m2}\right\},\right. \\ &\left.\max\left\{z_{13},z_{23},\cdots,z_{m3}\right\},\max\left\{z_{14},z_{24},\cdots,z_{m4}\right\}\right) \end{aligned} \tag{11}$$

8) 计算负理想解Z^-，其表示在每个指标上取得最小值的最差候选方案

$$\begin{aligned} Z^-=&(Z_1^-,Z_2^-,Z_3^-,Z_4^-)= \\ &\left(\min\left\{z_{11},z_{21},\cdots,z_{m1}\right\},\min\left\{z_{12},z_{22},\cdots,z_{m2}\right\},\right. \\ &\left.\min\left\{z_{13},z_{23},\cdots,z_{m3}\right\},\min\left\{z_{14},z_{24},\cdots,z_{m4}\right\}\right) \end{aligned} \tag{12}$$

9) 使用欧氏距离计算评估对象分别与正理想解和负理想解的距离

$$D_i^+=\sqrt{\sum_{j=1}^{4}w_j(Z_j^+-z_{ij})^2} \tag{13}$$

$$D_i^-=\sqrt{\sum_{j=1}^{4}w_j(Z_j^--z_{ij})^2} \tag{14}$$

10) 由欧氏距离计算第i个节点的理想贴近度，并将其作为权重

$$W_i=\frac{D_i^-}{D_i^++D_i^-} \tag{15}$$

根据式(1)~式(10)分别计算各评估指标的初步权重w_i，然后根据式(11)~式(14)分别求得评估对象与正理想解的距离D_i^+和负理想解的距离D_i^-。最后，根据式(15)计算节点i的最终权重W_i。在每个周期开始时，所有节点将权重值广播给所属副区域网络的上一轮周期主节点。

节点的权重值决定其被选为共识委员节点的概率。为了确保每个高校节点都能参与共识流程，并防止高性能权重的高校节点长时间抢占共识委员节点的位置，本文设计一种加权随机选取及分派方法，以保证节点的多样性和可靠性，防止某一类型的节点过于集中。另外，将验证节点按照权重平均分派给选取出来的共识委员节点，使得每个验证节点集的性能均衡，以便尽快实现副区域的全局数据一致性。由于生成高质量随机数的性能开销较大，因此本文采用了A-ExpJ算法[32]，将随机数的生成量从$O(n)$降低到$O\left(m\lg\left(\frac{n}{m}\right)\right)$，从而降低加权随机选取及分派方法的性能开销。高校节点权重集$V(\{N_1,W_1\},\{N_2,W_2\},\cdots,\{N_m,W_m\})$按性能权重不等概率随机选取$N_s$个共识委员节点的加权随机选取及分派方法，如图4所示。

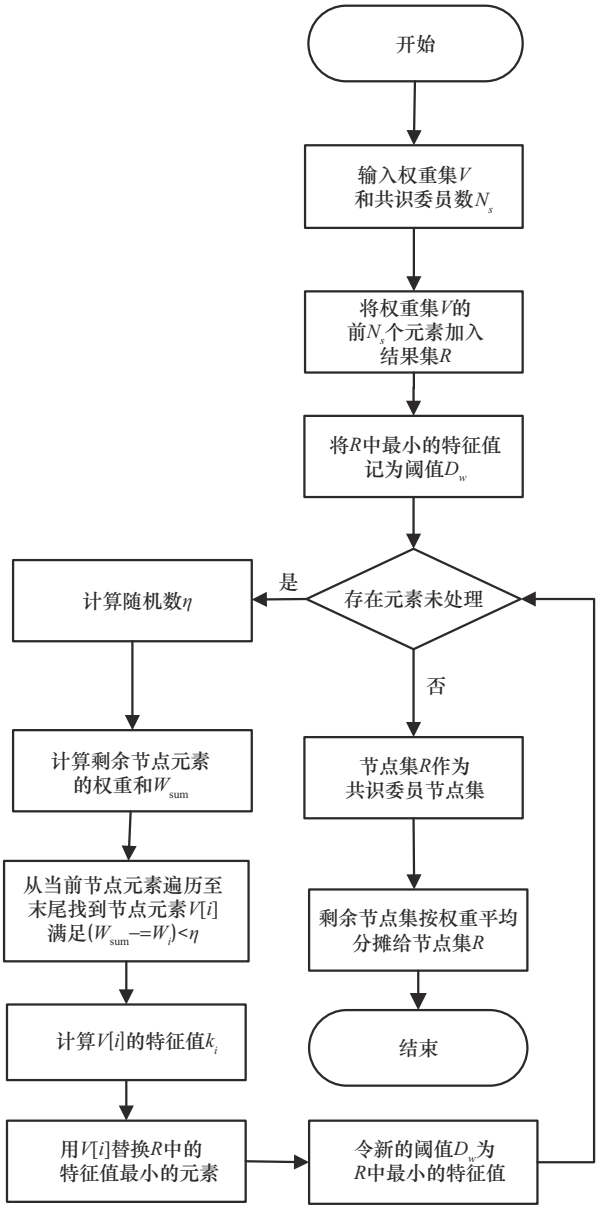

图4　共识委员节点选取流程

基于以上步骤，加权随机选取分派方法描述为算法1。

算法1　加权随机选取分派方法

输入　副区域共识委员节点的数量N_s，节点权重集V，节点编号N_i，权重W_i

输出　含共识委员节点的集合R，验证节点集字典G

/*选取共识委员节点集合R*/

1) 初始化数组 $R \leftarrow \{\}$

2) for i from 1 to N_s

3) u_i = random.uniform(0,1)

4) $k_i = u_i^{\frac{1}{w_i}}$

5) heapq.heappush(R,(k_i,$V[i]$))

6) end for

7) $D_w = R.\min()$

8) for j from N_s+1 to m

9) r = random.uniform(0,1)

10) $\eta = \frac{\text{math.log}(r)}{\text{math.log}(T_w)}$

11) W_{sum} = sum($W_m[j{:}m]$)

12) for i from j to m

13) W_{sum} −= W_i

14) if $W_{sum} < \eta$

15) filter$R$,node = heapq.heappop(R)

16) $t_w = D_w^{w_i}$

17) r_2 = random.uniform(t_w,1)

18) $k_i = r_2^{\frac{1}{w_i}}$

19) heapq.heappush(R,(k_i,$V[i]$))

20) $D_w = R.\min()$

21) break

22) end if

23) end for

24) end for

/*分派验证节点集合*/

25) consensusWeights = { node:0 for node in R }

26) G = { node:[ ] for node in R }

27) remaingNodes = sorted([nodefornodein$V$ if node[0] not in R],key=lambda,x:x[1], reverse = True)

28) for node in remainingNodes

29) minWeightNode=min(consensus,key=consensusWeights.get)

30) G[min WeightNode]

31) append(node[0])

32) consensus Weights minWeightNode+=node[1]]

33) end for

34) return R,G

算法1的步骤1)~步骤6)用于初始化集合R并计算第一个阈值D_w，其中random.uniform()函数生成0~1的随机浮点数，heapq.heappush()函数

用于将计算出的特征值存储在 R 中。步骤 7)~步骤 20)对剩余的元素执行 m~N_s 次基于最小特征值的元素替换更新操作，以得到最终的共识委员节点集 R。其中，heapq.heappop()函数用于从 R 中弹出最小特征值的元素。步骤 21)~步骤 23)用于分别初始化共识委员节点集、验证节点集和通过降序排序的剩余节点集，其中 sorted()函数用于对节点按权重进行降序排序。步骤 24)~步骤 29)逐个遍历剩余节点。在每次迭代中，找到当前权重最小的共识委员节点 minWeightNode。将剩余的节点分配给这个权重最小的共识委员节点，更新字典 G，将节点添加到对应的共识委员节点的节点列表中，并更新字典 consensusWeights 中对应共识委员节点的权重总和。

2.3.2　极限周期策略

节点权重计算分派策略直接影响共识效果，选择合适的周期 T 是关键。周期过短会导致频繁的权重计算和分派，从而增加资源消耗；而周期过长则无法实时更新权重，影响算法效果。

为了解决共识委员节点集选取周期的问题，本文提出基于权重变化率 $\triangle t$ 的极限周期策略。该策略对节点在周期 T 内的权重变化幅度与预设的权重变化率 $\triangle t$ 进行对比。当权重变化率大于 $\triangle t$ 时，节点将新的权重发送至主节点；当权重变化率小于 $\triangle t$ 时，节点仅在本地保存当前的权重信息。如果在一个周期 T 内没有收到其他节点提交的权重信息，则按照上一个周期 T 接收的权重执行加权随机选取分派方法。权重变化率 $\triangle t$ 为

$$\triangle t = \frac{W(i,t_2) - W(i,t_1)}{t_2 - t_1} \tag{16}$$

其中，$W(i,t_1)$ 和 $W(i,t_2)$ 分别表示节点 i 在 t_1 和 t_2 时刻的权重，且 $t_2 > t_1$。

2.4　副区域流程

2.4.1　准备阶段

按照基于TOPSIS建模和熵值赋权法建立的节点评价方法进行评价后，得到高校节点的权重集合 V。各个区域的节点总权重 $W_{\text{total}} = W_1 + W_2 + \cdots + W_m$，根据式(17)确定各副区域共识委员节点的数量 N_s。

$$N_s = \text{floor}\left(m\frac{p}{W_{\text{total}}}\right) \tag{17}$$

其中，p 为共识委员节点的总权重，需满足 PBFT 算法的条件 $n>3f+1$，即 $p \geqslant 4\max(W_1,W_1,\cdots W_m)$，其中 floor 为向零取舍函数，保证共识委员节点数为整数。随后，上一轮周期主节点将节点权重集合 V 和共识委员的节点数量 N_s 作为输入，执行加权随机选取分派方法，得到共识委员节点集 $R(N_{s1},N_{s2},\cdots,N_{sN_s})$ 及每个共识委员节点对应的验证节点集 $G[N_{\text{si}}](N_{v1},N_{v2},\cdots,N_{vV_s})$。在一轮周期开始时，执行一次分组策略。具体步骤如下。

1) 共识委员节点集 R 中的所有节点向其对应的验证节点集广播消息 $< N_{\text{si}},G[N_{\text{si}}],t_i > \partial_{\text{vi}}$，其中 t_i 为时间戳，∂_{vi} 为共识委员节点的签名。

2) 验证节点 N_{vi} 收到来自共识委员节点的消息后，会检查 $G[N_{\text{si}}]$ 中是否存在本节点的编号，并验证消息签名的正确性。如果确认无误，则向共识委员节点发送确认消息 $<\text{commit},t_i,N_{\text{vi}} > \partial_{\text{vi}}$。

3) 共识委员节点收到验证节点的确认消息后，将验证节点 N_{vi} 加入该共识委员节点的组维护列表 L_{si} 中。

4) 分组结束后，所有共识委员节点广播各自的组维护列表 L_{si}，完成分组确认。其中，L_{si} 包含共识委员节点在分组策略中所收到的验证节点信息。

2.4.2　副区域共识流程

z-PBFT 算法模型基于省级行政单位进行划分，每个省级行政单位内的区块链节点（即高校）数量通常超过 100，而 PBFT 算法在节点数量超过 100 时，其共识效率会显著下降。因此，副区域采用局部共识策略，有效降低了共识算法对区块链节点（高校）数量的依赖性。当客户端发出证书操作请求后，算法进入共识阶段，共分为以下 3 个阶段。

1) pre-prepare 阶段

与 PBFT 算法类似，首先在共识委员节点中选取一个节点 $U \bmod N_s$ 作为本轮共识的 proposer 节点。proposer 节点负责执行区块操作，并通过验证随机数生成算法产生 VRF 的随机数 y 及其证明 π。然后，proposer 节点生成签名包 <<pre-prepare, U,p_n,p_d,y,π,nbh > ,p_m >，并将其广播给所有共识委员节点。

2) prepare 阶段

共识委员节点在收集签名包并验证 VRF 证明

的有效性后，向其他共识委员节点发送准备消息 < prepare,U,p_n,p_d,sealer$_{si}$,bool >。其中bool是VRF随机数有效性的返回结果。各共识委员节点在收集到$2f$+1个签名包（包括自身的签名包）后，表明已经达到可以提交区块的状态，进入commit阶段。

3) commit阶段

在此阶段，任意共识委员节点验证通过后，会向其余共识委员节点发送 < commit,U,p_n,p_d, sealer$_{si}$ >。当某个共识委员节点收集到$2f$+1个commit包后，便已达到commit状态，此时该节点会将本地缓存的最新区块提交到数据库，并进入同步阶段。

2.4.3 同步阶段

如图5所示，当共识委员节点集完成共识后，共识委员节点执行组播策略，所有共识委员节点将已共识的区块广播到其对应的验证节点集中。各验证节点验证共识产生的区块：1) 校验区块的签名列表，每个区块必须至少包含三分之二共识委员节点的签名；2) 校验区块执行结果，本地执行结果需与共识委员节点产生的区块执行结果一致，验证成功后，验证节点同步更新区块，并定期向其他节点广播自身的最高区块高度。当节点收到其他节点广播的区块高度信息时，将其与自身的区块高度进行比较。若自身区块高度低于收到的区块高度，则启动区块下载流程。下载流程通过请求的方式进行，随机挑选满足条件的节点，并发送所需下载的区块范围请求。接收到下载请求的节点将根据请求内容回复相应的区块。

2.5 主区域流程

2.5.1 主节点选取机制

主节点从共识委员节点集$R(N_{s1},N_{s2},\cdots,N_{sNs})$中选取，以进一步保证选取的随机性。该选取过程分为2个阶段，分别是生成过程和验证选取过程。生成过程由该轮的proposer节点执行主节点选取方法，具体过程如下。

1) proposer节点计算VRF公钥，计算式为

$$\{\text{pk,sk}\}=\text{KeyGen}(g) \tag{18}$$

KeyGen是密钥生成函数，输入为密钥生成元g，pk和sk分别为生成的公钥和私钥。

2) 生成可验证随机数证明，计算式为

$$\{y,\pi\}=\text{Evaluate}(\text{sk,nbh}) \tag{19}$$

Evaluate是VRF随机数生成函数，输入为proposer节点的sk以及当前最高区块的哈希值nbh，输出伪随机数y和随机值证明π。

3) VRF证明打包，proposer节点将{ y,π,nbh }生成为一笔带有proposer签名的交易，并将其打包为区块中的最后一笔交易，最后将该区块组装到prepare包中。

在验证选取过程中，其他共识委员节点在收到prepare包后，会从prepare包中解码出区块并验证其有效性，随后选取主节点，具体流程如图6所示。具体执行过程如下。

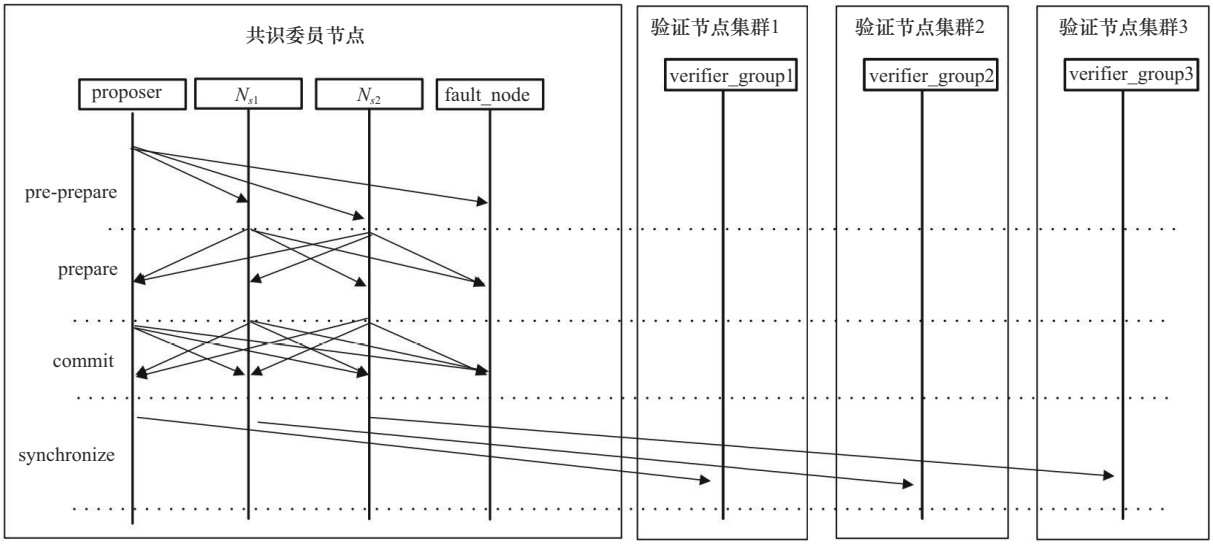

图5 副区域共识和验证流程

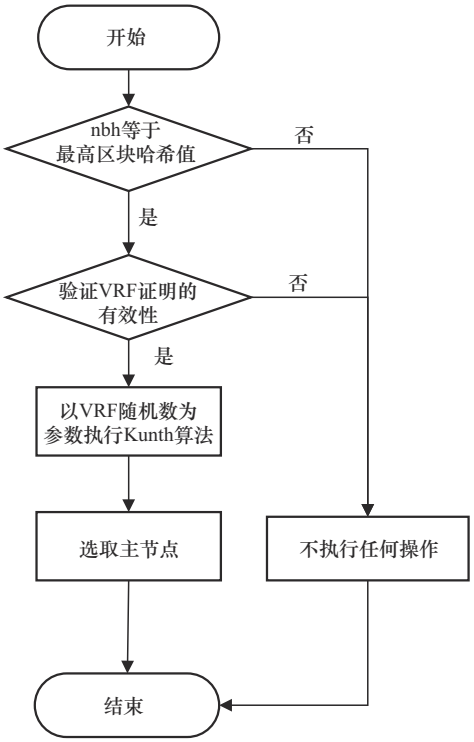

图6　验证并选取主节点流程

1) 验证VRF证明对应输入的nbh，即验证打包时的哈希值是否与最高区块的哈希值相等。如果相等，则验证过程正常进行。

2) 验证VRF证明的有效性，具体为

$$\text{bool} = \text{Verify}(\text{pk},\text{nbh},y,\pi) \tag{20}$$

Verify()为概率性验证算法，如果随机数被验证为有效，则该随机数被认定为Kunth洗牌算法的输入。

3) 以随机数y为输入，采用Kunth洗牌算法打乱副区域中所有共识委员节点的顺序，从打乱顺序的节点列表中，选取第一个节点作为主节点。具体如算法2所示。

算法2　基于Kunth的随机数种子输入选取节点算法

输入　随机数y，共识委员节点集R

输出　主节点C

1)　random.seed(y) //设置y为随机数种子

2)　temp = R.toArray() //将共识委员节点集转换为数组

3)　for i from 1 to N_s //遍历共识委员节点

4)　　g = random()%(i + 1)//执行节点随机交换

5)　　swap(temp, N_{si}, N_{sg}) //交换编号为i和g的共识委员节点的位置

6)　end for

7)　return C = temp.get(0) //选取数组中第一个节点作为主节点

在每个副区域的主节点选定后，该节点会生成一条广播消息，此消息包含节点ID、VRF随机数、签名和区块哈希值，并将该消息广播至主区域内的其他主节点。接收到广播消息的主节点将验证该消息的真实性，包括验证签名是否由声名的主节点生成以及VRF随机数是否与消息中提供的区块哈希值匹配。验证通过后，副区域主节点的ID和相关信息将被加入本地主节点列表中。所有主节点都维护一个本地主节点列表，该列表中的每项记录都包括主节点ID、VRF随机数、区块哈希值和验证状态。主节点之间定期更新列表，以确保主节点列表的实时性和准确性。每隔固定周期，主节点会相互同步本地的主节点列表，并合并更新信息。如果某个主节点发现列表中的某条记录有更新（如验证状态变化），则将该信息广播给其他主节点。若同一主节点ID对应多个VRF随机数或区块哈希值不一致，则选择最新的记录作为有效记录。

2.5.2　主区域共识流程

在副区域选出的主节点会将本轮周期生成的区块数据打包并上传至主区域网络。在上传之前，主节点会再次确认区块数据是否至少包含三分之二共识委员节点的签名，以确保数据的正确性和可靠性。当区块数据确认无误后，主节点会执行跨链交易的生成和提交操作。在主区域，所有主节点共同协作，采用PBFT共识算法完成主区域的共识过程。

2.5.3　数据跨链

在跨链共识中，每个副区域在固定周期内选出的主节点将作为代表，负责将该副区域本轮周期产生的区块跨链到主区域进行共识。跨链共识过程分为以下2个阶段：1) 主节点生成证明，主节点将跨链交易数据打包，并生成一份证明，作为该从链参与主链共识的证据；2) 主链共识验证，主区域在收到跨链交易的证明后，将其提交到主链共识过程中，验证证明的合法性和正确性，并将合法的交易写入主链中。在一轮共识完成后，所有主节点将区块保存至本地并进行同步，使得所在副区域的所有节点可以查询该数据，从而完

成数据的跨链操作。

3 实验与分析

本节将从交易时延、通信开销以及吞吐量3个方面对z-PBFT、经典PBFT算法和HotStuff的表现进行分析，并对z-PBFT算法的安全性进行验证，以论证z-PBFT的有效性和可靠性。

3.1 实验环境与参数配置

本文实验环境选择的主机系统为Windows10，处理器为Intel(R) Core(TM) i7-9700，内存为32 GB。本文使用基于Java的t-io工具搭建区块链网络环境进行仿真实验，通过虚拟机划分副区域。实验中通过控制台输入java命令java -jar zpbft.jar ip地址：端口映射的方式来模拟多节点环境，如表3所示，其中zpbft.jar是打包的算法实现文件。为了验证节点间通信距离对性能的影响，引入网络时延参数，以反映不同节点间的物理距离。每个副区域被模拟为一定的地理分布，时延根据节点间的虚拟距离动态计算。时延Δs以ms为单位，其值按式(21)计算

$$\Delta s = D_{\text{z-pbft}} \times R_{\text{z-pbft}} \tag{21}$$

其中，$D_{\text{z-pbft}}$是节点间的虚拟距离，虚拟距离指的是在本实验中为方便计算而定义的一种抽象距离度量，并不对应实际的物理距离；$R_{\text{z-pbft}}$是单位距离的时延。例如，每个虚拟距离单位的时延为20 ms。节点间的虚拟距离根据节点编号进行分配。例如，相邻节点（0和1，1和2）之间的距离设定为1个虚拟距离单位，而更远的节点（0和2）的距离设定为2个虚拟距离单位。实验中设置了不同数量的节点，并模拟在4个周期内不同节点不断发送请求的情况。实验模拟了4个副区域，每个副区域的硬盘容量为40 GB。首先，为每个副区域中节点索引0~3的节点分配2 GB虚拟存储容量资源，用于模拟该区域的高性能节点，其余10 GB容量资源由其他节点均分。每个副区域的起始节点数均为4个，此后依次在各副区域增加一个节点进行数据测试。

在第2节中，基于TOPSIS建模和熵值赋权法的加权随机选取分派策略需要确定一个未知参数——周期T。为了测试周期T的取值，实验设定从2 min开始，每隔1 min进行一次测试，直到10 min结束。同时，使用压力测试工具JMeter生成数据，设置2个周期内的不间断请求，并分别设置单个副区域节点数为20和40作为实验条件。通过测试其吞吐量TPS（transaction per second），找出吞吐量最佳的周期，即为最优周期T。

如图7所示，在2~5 min，由于周期较短，更多的时间被用于完成加权随机选取分派方法。随着周期T的增加，吞吐量逐渐上升，并在5 min时达到峰值。此后，由于周期过长，对节点权重集的分派不及时，低性能节点仍然留在共识委员节点集中，进而导致吞吐量逐渐下降。实验结果表明，最优周期T为5 min。

表3 实验节点配置

虚拟机名	节点索引	CPU	硬盘/GB	IP地址	端口
Ubuntu18.04-0	0	双核	40	169.254.91.41	4500
Ubuntu18.04-0	1	双核	40	169.254.91.41	4501
…	…	…	…	…	…
Ubuntu18.04-1	0	单核	40	169.254.91.42	4500
Ubuntu18.04-1	1	单核	40	169.254.91.42	4501
…	…	…	…	…	…
Ubuntu18.04-2	0	双核	40	169.254.91.43	4500
Ubuntu18.04-2	1	双核	40	169.254.91.43	4501
…	…	…	…	…	…
Ubuntu18.04-3	0	单核	40	169.254.91.44	4500
Ubuntu18.04-3	1	单核	40	169.254.91.44	4501
…	…	…	…	…	…

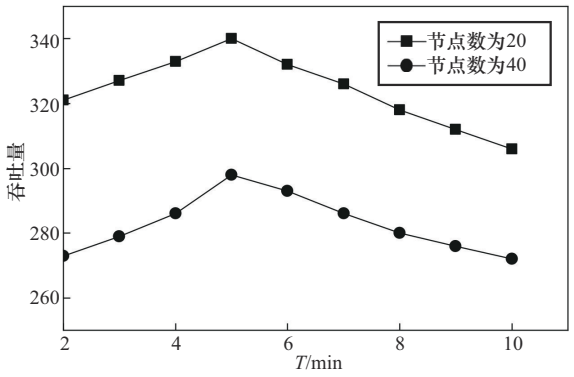

图7　吞吐量与T的关系

在选出最优周期T后，设定初始权重变化率$\triangle t$为0.10，并依次以0.05的步长增加至0.40进行测试，选取最佳的权重变化率$\triangle t$，测试条件与选取最优周期时相同。实验结果如图8所示，当$\triangle t$取值为0.20时为最佳变化率，因此本文实验选定$\triangle t$=0.20。

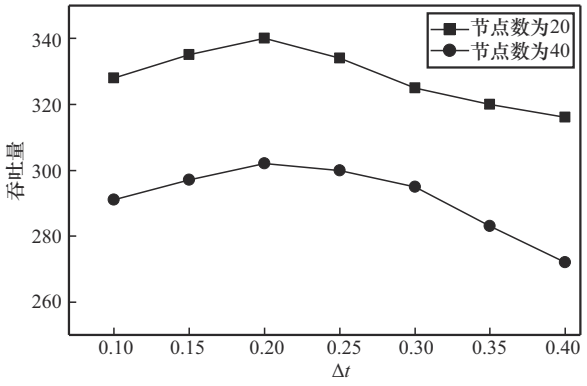

图8　吞吐量与Δt的关系

3.2　交易时延

交易时延是在分布式系统中，从交易请求发出到该请求被确认并执行所需的时间。在本文中，z-PBFT共识算法采用两阶段机制，包括副区域共识和主区域共识，因此交易时延可以被分为这2个阶段的时延之和，表示为

$$\mathrm{DTime}_{\text{z-pbft}}=t_s+t_m+t_t \tag{22}$$

其中，t_s是副区域时延，t_m是主区域时延，t_t是信息传输时间。如图9所示，相较于PBFT，z-PBFT在交易时延方面表现更优秀。在节点数量较少（小于24个节点）时，z-PBFT、PBFT和HotStuff的平均交易时延差异不大，但随着节点数量的增加，z-PBFT的交易时延明显低于PBFT，略低于Hot-Stuff。这是因为z-PBFT引入了加权随机选取方法，定量选取共识委员节点，从而减少参与共识的节点数量，缩短了交易在共识过程中的总传播时间和确认时间。

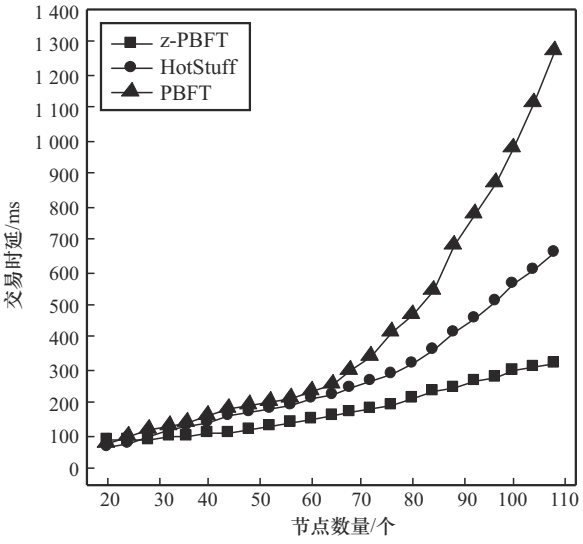

图9　交易时延对比

3.3　通信与计算开销

本文提出的模型包含n个副区域，每个副区域包含N_s个共识委员节点和V_s个验证节点，主区域由副区域的n个节点组成。在实验中假设每个副区域的节点数相同，因此z-PBFT算法模型中共有$n(N_s+V_s)$个节点。z-PBFT算法的核心在于通过双层PBFT设计来减少共识节点的数量，从而降低通信开销。传统PBFT算法的通信复杂度为所有节点数的平方，即$O((n(N_s+V_s))^2)$。而在z-PBFT算法中，副区域的共识仅涉及共识委员节点，验证过程呈线性增长，整体的共识通信复杂度为$O(n^2+n({N_s}^2+V_s))$。在大规模节点情况下，由于$V_s \gg N_s$，因此z-PBFT的通信复杂度明显低于全节点参与的情况。在共识委员节点选择方面，通过加权随机选取分派策略计算每个节点的权重W_i，涉及多个性能指标的评估和归一化，其计算复杂度为$O(p)$，其中p是指标数量。随机选取算法本身的复杂度为$O(N_s \log N_s)$，考虑该算法是周期性执行的，整体计算成本是可接受的。此外，VRF的计算开销主要集中在密钥生成、随机数生成和证明验证，其计算复杂度为$O(1)$。混洗操作的复杂度为$O(N_s)$，鉴于N_s的数量较少，该过程的计算开销也较低。在每轮共识中，仅需执行一次VRF计算和一次混洗操作。因此，加权随机选取分派策略、

VRF机制以及混洗算法的总计算开销相对较低。在仿真实验中，设置消息大小为128 B，对通信开销进行了详细分析。通信开销为

$$O_{\text{total}} = NMS \tag{23}$$

其中，O_{total}是总通信开销，N是节点数量，M是消息数量，S是消息大小。根据式(23)进行计算，总通信开销对比如图10所示。

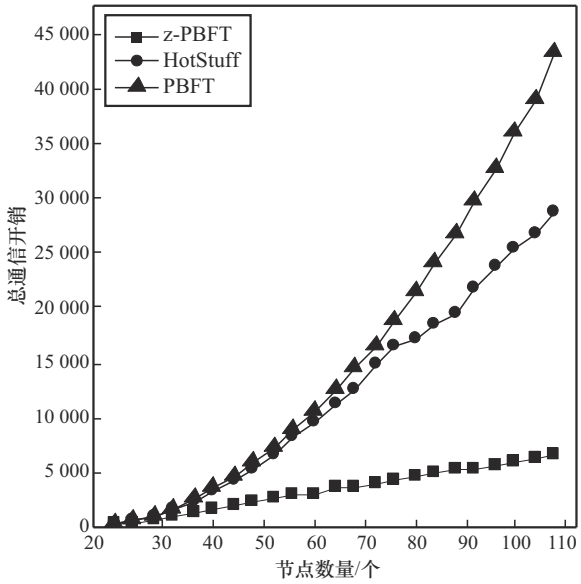

图10　总通信开销对比

由图10可以看出，PBFT和HotStuff算法的共识节点数量随着节点数量的增加而急剧增加。PBFT算法的总通信开销呈指数型增长，而HotStuff算法的复杂度虽然低于PBFT，但其复杂度仍然与节点规模成正比。相比之下，z-PBFT算法在共识流程中仅需一个固定数量的共识委员节点集合执行共识流程，并在交易落盘后进行一次广播，这大幅降低了通信开销，使其保持线性增长。

3.4　吞吐量

如图11所示，随着节点数量的增加，z-PBFT的吞吐量呈缓慢下降的趋势。相比之下，PBFT的吞吐量在节点数量增加时呈指数级下降，因为每个节点都需要与其他节点进行PBFT流程，其通信时间随着节点数量的增加呈指数级增长。HotStuff采用星形通信结构，当节点数量增多时，主节点的通信负担加重，导致整体吞吐量下降。z-PBFT算法在节点数量增加时，同步阶段的复杂度会略有增加。此外，由于采用了VRF技术和周期性加权随机分派策略，会产生一些额外的计算开销，因此算法性能会受到一定程度的影响。虽然z-PBFT的吞吐量随着节点数量的增加会有所下降，但总体上节点规模对吞吐量的影响明显小于PBFT算法。因此，z-PBFT算法展现出更好的可扩展性和性能。

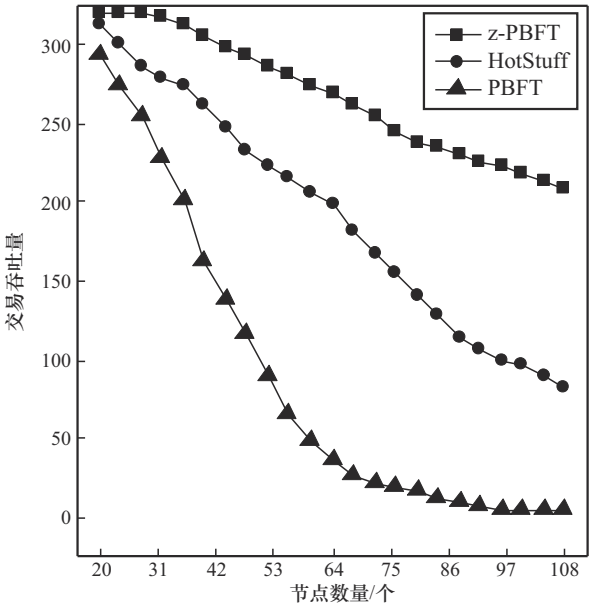

图11　交易吞吐量对比

3.5　安全性分析

在构成主区域网络的副区域选取主节点的体系中，共识机制的安全性由PBFT算法固有的抵御拜占庭节点的特性提供。同时，由于引入了VRF机制和洗牌算法，主节点的选取具有完全唯一性和完全抗冲突性。即使攻击者攻破了VRF密钥SK，也无法找到2个不同的输入alpha1和alpha2，使得2个随机数验证的结果一致（$\text{VRF}_{\text{hash}}(\text{SK,alpha1}) = = \text{VRF}_{\text{hash}}(\text{SK,alpha2})$），从而避免了主节点被恶意篡改。接下来，进一步对副区域内的拜占庭问题进行分析。

引理1　被选择的共识委员节点中，除了可以忽略的概率外，超过半数的成员节点是诚实的。

证明　假设在最不利的情况下，所有节点的权重相等，那么共识委员节点的选取可被视为一个独立的随机抽样问题，节点p_i被选取为共识委员节点的概率为

$$\Pr[p_i] = \frac{N_s W_i}{\sum_{j=1}^{m} W_j} = \frac{N_s}{m} \tag{24}$$

其中，N_s是共识委员节点的数量，m是副区域节点的数量，W_j是节点j的权重值。

设X是选出的节点是诚实节点的概率，Y是选出的节点是拜占庭节点的概率。随机变量A是共识委员会中诚实节点的数量，随机变量B是共识委员会中拜占庭节点的数量，随机变量A和B满足二项分布

$$\Pr[A=k]=\binom{m}{k}\left(\alpha\frac{N_s}{m}\right)^k\left(1-\alpha\frac{N_s}{m}\right)^{n-k} \tag{25}$$

$$\Pr[B=k]=\binom{m}{k}\left(\beta\frac{N_s}{m}\right)^k\left(1-\beta\frac{N_s}{m}\right)^{n-k} \tag{26}$$

X和Y表示为

$$X=\left\{A:0\leqslant A\leqslant\left\lfloor\frac{N_s}{3}\right\rfloor\right\} \tag{27}$$

$$Y=\left\{B:\left\lceil\frac{N_s}{3}\right\rceil\leqslant B\leqslant m\right\} \tag{28}$$

当X和Y都不成立时，才能得出共识委员会中诚实节点占多数的结论。$\Pr[X\cup Y]$表示X和Y中至少一个成立的概率，可以得到不等式

$$\Pr[X\cup Y]\leqslant\Pr[X]+\Pr[Y]=$$
$$\Pr\left[A\leqslant\left\lfloor\frac{N_s}{3}\right\rfloor\right]+\Pr\left[B\geqslant\left\lceil\frac{N_s}{3}\right\rceil\right]=$$
$$F\left(\left\lfloor\frac{N_s}{3}\right\rfloor,m,\alpha\frac{N_s}{m}\right)+F\left(m-\left\lceil\frac{N_s}{3}\right\rceil,m,1-\beta\frac{N_s}{m}\right) \tag{29}$$

其中，F是二项分布的累积分布函数。应用切尔诺夫不等式得到

$$F\left(\left\lfloor\frac{N_s}{3}\right\rfloor,m,\alpha\frac{N_s}{m}\right)+F\left(m-\left\lceil\frac{N_s}{3}\right\rceil,m,1-\beta\frac{N_s}{m}\right)\leqslant$$
$$2\exp\left(-2m\left(\frac{\left\lfloor\frac{N_s}{3}\right\rfloor-\beta N_s}{m}\right)^2\right)=\delta(N_s) \tag{30}$$

其中，β是拜占庭节点的比例。假设在最坏情况下，所有节点（无论是诚实节点还是拜占庭节点）都具有相同的权重。与之相反，在通常情况下，所有诚实节点和少数拜占庭节点的权重相等，而其他拜占庭节点的权重则较低。这表明节点权重越高，其被认为是诚实节点的概率也越高。根据每个节点被选为共识委员会成员的概率$\Pr[p_i]$，定义$\Pr[\text{Dishonest}]$为拜占庭节点占多数的概率。根据二项分布及其相关不等式得到

$$\Pr[\text{Dishonest}]_{\text{Normal}}<\Pr[\text{Dishonest}]_{\text{Worst}}=\delta(N_s) \tag{31}$$

其中，$\delta(N_s)$是在最坏情况下拜占庭节点占多数的最大概率，依据式(30)得出。因此，在共识委员节点选择过程中，正常情况下诚实节点占多数的概率明显高于最坏情况下的概率。即使在拜占庭节点试图影响系统的极端情况下，本文设计的z-PBFT算法仍能有效抵御攻击，确保共识流程的正常运行。证毕。

4　结束语

本文提出一种拜占庭容错共识算法z-PBFT，该算法能够解决基于区块链技术的学位/学历证书管理中区块链节点（高校）之间存在的差异性和不支持大规模节点的问题，有效缓解了当前“区块链+教育”证书应用系统在可扩展性方面的瓶颈。z-PBFT算法为各级教育机构和第三方机构提供了一种安全高效的证书存证服务。实验结果表明，z-PBFT算法在性能和可扩展性上具有显著优势，满足大规模区块链节点（高校）对证书处理的需求。

参考文献：

[1] LIU S G, BA L. Blockchain technology and its application prospect in higher education[C]//Proceedings of the 13th International Conference on Education Technology and Computers. New York: ACM Press, 2021: 237-242.

[2] EMMA W, JANET N. How thousands of nurses got licensed with fake degrees[R]. 2023.

[3] JIN H, XIAO J. Towards trustworthy blockchain systems in the era of “Internet of value”: development, challenges, and future trends[J]. Science China Information Sciences, 2021, 65(5): 153101.

[4] DU M X, CHEN Q J, XIAO J, et al. Supply chain finance innovation using blockchain[J]. IEEE Transactions on Engineering Management, 2020, 67(4): 1045-1058.

[5] GAI K K, WU Y L, ZHU L H, et al. Privacy-preserving energy trading using consortium blockchain in smart grid[J]. IEEE Transactions on Industrial Informatics, 2019, 15(6): 3548-3558.

[6] 谭海波, 周桐, 赵赫, 等. 基于区块链的档案数据保护与共享方法[J]. 软件学报, 2019, 30(9): 2620-2635.
TAN H B, ZHOU T, ZHAO H, et al. Archival data protection and sharing method based on blockchain[J]. Journal of Software, 2019, 30(9): 2620-2635.

[7] RUSTEMI A, DALIPI F, ATANASOVSKI V, et al. A systematic literature review on blockchain-based systems for academic certificate verification[J]. IEEE Access, 2023, 11: 64679-64696.

[8] 曾诗钦, 霍如, 黄韬, 等. 区块链技术研究综述: 原理、进展与应用[J]. 通信学报, 2020, 41(1): 134-151.
ZENG S Q, HUO R, HUANG T, et al. Survey of blockchain: principle, progress and application[J]. Journal on Communications, 2020, 41(1): 134-151.

[9] JAKOBSSON M, JUELS A. Proofs of work and bread pudding protocols(extended abstract)[C]//The International Federation for Information Processing. Berlin: Springer, 1999: 258-272.

[10] BAMAKAN S M H, MOTAVALI A, BABAEI B A. A survey of block-

chain consensus algorithms performance evaluation criteria[J]. Expert Systems with Applications, 2020, 154(9): 113385.

[11] LARIMER D: Delegated proof-of-stake consensus[R]. 2014.

[12] CASTRO M, LISKOV B. Practical Byzantine fault tolerance and proactive recovery[J]. ACM Transactions on Computer Systems, 2002, 20 (4): 398-461.

[13] GRATHER W, KOLVENBACH S, RULAND R, et al. Blockchain for education: Lifelong learning passport[C]//Proceedings of 1st ERCIM Blockchain Workshop 2018. Netherland: EUSSET, 2018: 2510-2591.

[14] AFRIANTO I, HERYANTO Y. Design and implementation of work training certificate verification based on public blockchain platform[C]// Proceedings of the 2020 Fifth International Conference on Informatics and Computing (ICIC). Piscataway: IEEE Press, 2020: 1-8.

[15] NIKOLIĆ S, MATIĆ S, ČAPKO D, et al. Development of a blockchain-based application for digital certificates in education[C]//Proceedings of the 2022 30th Telecommunications Forum (TELFOR). Piscataway: IEEE Press, 2022: 1-4.

[16] SHAWON S K, AHAMMAD H, SHETU S Z, et al. DIUcerts DApp: a blockchain-based solution for verification of educational certificates[C]// Proceedings of the 2021 12th International Conference on Computing Communication and Networking Technologies (ICCCNT). Piscataway: IEEE Press, 2021: 1-10.

[17] SHANG Y, LI Z X, LI Z, et al. Blockchain technology and its application in higher education[C]//Proceedings of the 14th International Conference on Education Technology and Computers. New York: ACM Press, 2022: 108-113.

[18] 刘东伟, 张学旺, 郭晓金. 基于区块链的学位证书存证系统设计与实现[J]. 计算机工程与设计, 2020, 41(2): 567-573.
LIU D W, ZHANG X W, GUO X J. Design and implementation of degree certificate storage and verification system based on blockchain[J]. Computer Engineering and Design, 2020, 41(2): 567-573.

[19] SUKHWANI H, WANG N, TRIVEDI K S, et al. Performance modeling of hyperledger fabric (permissioned blockchain network)[C]//Proceedings of the 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). Piscataway: IEEE Press, 2018: 1-8.

[20] SUKHWANI H, MARTÍNEZ J M, CHANG X L, et al. Performance modeling of PBFT consensus process for permissioned blockchain network (hyperledger fabric) [C]//Proceedings of the 2017 IEEE 36th Symposium on Reliable Distributed Systems (SRDS). Piscataway: IEEE Press, 2017: 253-255.

[21] HWANG C L, YOON K. Multiple attribute decision making[M]. Berlin: Springer, 1981.

[22] 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.

[23] GAN B, WU Q W, LI X, et al. Classification of blockchain consensus mechanisms based on PBFT algorithm[C]//Proceedings of the 2021 International Conference on Computer Engineering and Application (ICCEA). Piscataway: IEEE Press, 2021: 26-29.

[24] 黄冬艳, 李浪, 陈斌, 等. RBFT: 基于Raft集群的拜占庭容错共识机制[J]. 通信学报, 2021, 42(3): 209-219.
HUANG D Y, LI L, CHEN B, et al. RBFT: a new Byzantine fault-tolerant consensus mechanism based on Raft cluster[J]. Journal on Communications, 2021, 42(3): 209-219.

[25] ZHAN Y, WANG B C, LU R X, et al. DRBFT: Delegated randomization Byzantine fault tolerance consensus protocol for blockchains[J]. Information Sciences, 2021, 559: 8-21.

[26] YIN M F, MALKHI D, REITER M K, et al. HotStuff: BFT consensus with linearity and responsiveness[C]//Proceedings of the 2019 ACM Symposium on Principles of Distributed Computing. New York: ACM Press, 2019: 347-356.

[27] LI W Y, FENG C L, ZHANG L, et al. A scalable multi-layer PBFT consensus for blockchain[J]. IEEE Transactions on Parallel and Distributed Systems, 2021, 32(5): 1146-1160.

[28] QUSHTOM H, MIŠIĆ J, MIŠIĆ V B, et al. A high performance two-layer consensus architecture for blockchain-based IoT systems[J]. Peer-to-Peer Networking and Applications, 2022, 15(5): 2444-2456.

[29] JIANG W X, WU X X, SONG M Y, et al. A scalable Byzantine fault tolerance algorithm based on a tree topology network[J]. IEEE Access, 2023, 11: 33509-33519.

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

[31] LIU S N, ZHANG R H, LIU C Z, et al. P-PBFT: an improved blockchain algorithm to support large-scale pharmaceutical traceability[J]. Computers in Biology and Medicine, 2023, 154: 106590.

[32] ERNEST B, DANIEL G, KEVIN M, et al. Fast exponentiation with precomputation[C]//Proceedings of the 11th Annual International Conference on Theory and Application of Cryptographic Techniques. Berlin: Springer, 2001: 431-437.

[作者简介]

张学旺（1974– ），男，湖南祁东人，重庆大学博士生、重庆邮电大学副教授，主要研究方向为区块链、物联网、数据安全与隐私保护。

雷志滔（1998– ），男，四川内江人，重庆邮电大学硕士生，主要研究方向为区块链、互联网软件及安全。

林金朝（1966– ），男，四川蓬溪人，博士，重庆邮电大学教授、博士生导师，主要研究方向为无线通信传输、无线体域网与信息处理。