Computer Science and Application 计算机科学与应用, 2021, 11(3), 643-653
Published Online March 2021 in Hans. http://www.hanspub.org/journal/csa
https://doi.org/10.12677/csa.2021.113066

Hans汉斯

基于PBFT共识算法的区块链改进方案

胡振宇，钟林峰，杨振国，刘文印

广东工业大学，计算机学院，广东 广州
Email: yuzhenhu98@outlook.com, linf.z@qq.com, liuwy@gdut.edu.cn

收稿日期：2021年2月23日；录用日期：2021年3月18日；发布日期：2021年3月25日

摘　要

针对应用于联盟链的实用拜占庭容错(PBFT)共识算法存在的主节点取余方法容易导致恶意节点当选、三阶段共识流程导致高时延等问题，提出了一种改进的实用拜占庭容错(PBFT+)共识算法。首先，基于信用机制来选取主节点，引入信用层次体系对系统节点进行信用评价，共识节点的信用层次随着其产生有效区块的数量进行动态变化，节点的信用层次划分是系统对节点的评判标准，系统将选取全网信用值最高的节点作为主节点，可以保证主节点的最优性。通过备用主节点机制避免频繁发生视图变更，提高系统可用性和安全性。其次，优化PBFT共识算法的三阶段共识流程，将确认阶段中全网节点广播确认消息改进为由主节点收集确认消息，减少了通信开销。实验部分将PBFT+算法与PBFT算法进行多组对比测试，通过仿真实验结果表明，PBFT+算法有效的减少了系统资源的消耗，降低了时延。

关键词

区块链，共识算法，PBFT，信用机制，主节点

Blockchain Improvement Scheme Based on PBFT Consensus Algorithm

Zhenyu Hu, Linfeng Zhong, Zhenguo Yang, Wenyin Liu

School of Computers, Guangdong University of Technology, Guangzhou Guangdong
Email: yuzhenhu98@outlook.com, linf.z@qq.com, liuwy@gdut.edu.cn

Received: Feb. 23rd, 2021; accepted: Mar. 18th, 2021; published: Mar. 25th, 2021

Abstract

Aiming at the problems of the master node surplus method of the Practical Byzantine Fault Tolerance Consensus algorithm applied to the alliance blockchain, which may easily lead to the election

文章引用: 胡振宇, 钟林峰, 杨振国, 刘文印. 基于 PBFT 共识算法的区块链改进方案[J]. 计算机科学与应用, 2021, 11(3): 643-653. DOI: 10.12677/csa.2021.113066

of malicious nodes and the high latency caused by the three-stage consensus process. In this paper, we extend PBFT with credit mechanism, denoted as PBFT+. More specifically, we introduce a credit mechanism to select the primary node instead of random selection, leading into a credit hierarchy system for credit evaluation of system nodes. The credit level of consensus node changes dynamically with the number of valid blocks that it generates. The credit level division of nodes is the system criterion for judging nodes, and the system will select the node with the highest credit value of the whole network as the master node to ensure the optimality of the master node. The availability and security of the system can be improved by using the standby primary node mechanism to avoid frequent view changes. Furthermore, the three-stage consensus process of the PBFT algorithm is optimized. In the confirmation phase, the primary node collects the confirmation information instead of broadcasting the message from the entire network node, which reduces the communication overhead. In the experimental part, the PBFT+ and the PBFT algorithm are tested in multiple groups. The experimental simulation results show that the PBFT+ algorithm can effectively reduce the consumption of system resources and reduce the delay.

Keywords

Blockchain, Consensus Algorithm, PBFT, Credit Mechanism, Primary Node

1. 引言

区块链作为一个分布式数据库技术方案[1] [2]，包含了分布式存储[3]技术、共识机制、加密算法、可编程的智能合约等技术[4]。由于区块链具有去中心化、不可篡改和匿名性的特点，区块链技术在数字货币、公共服务等领域快速发展[5] [6]。

虽然现阶段区块链得到快速发展，但其在存储、共识[7]等方面存在明显问题。基于工作量证明(POW) [8]的协议具有高延迟和低吞吐量，POW 协议的性能主要瓶颈在于需要解决密码难题，需要非常大的算力进行挖矿才能解决，造成大量的硬件资源浪费。比特币是基于 POW 协议的区块链系统，它现已被证明可以容纳数千个节点，但是它的交易吞吐量是每秒 6~10 个交易，平均需要 10 分钟来生成一个新的区块，效率非常低下，难以进行大范围应用。权益证明共识机制 POS [9]引入了“币龄”这一概念，币龄是用户持有的系统代币的数量和持有时间的乘积，系统根据矿工持有的币龄降低挖矿难度，矿工在生成区块后会消耗相应的币龄。与 POW 相比，POS 在性能有所提高。但是 POS 算法中权益高的节点对网络造成攻击，在一定程度上降低安全性。DPOS [10]共识机制是在 POW 及 POS 的基础上，用节点的权益作为选票选出一定数量的代表节点轮流进行区块的生成和验证。因此直接参与共识的节点减少，所以共识的速度大大提升。但是节点参与投票不积极，同时对于代币的依赖使得该机制不具有普适性。Paxos [11]是基于消息传递的，旨在解决分布式系统中内容达成一致的问题，在执行了相同的操作后，所有节点能够得到一致的结果。但是 Paxos 没有考虑系统中存在恶意节点的情况，一旦恶意节点发送假消息，那么整个系统将会存储假消息信息。实用的拜占庭容错(PBFT) [12]共识算法基于 BFT [13]算法进行改进，采用 C/S 架构[14]，将算法复杂度由指数级降为多项式级，已经被证明可以实现每秒处理数千个交易请求，具有更高的交易吞吐量。

然而现有的 PBFT 算法还存在一些不足。首先，PBFT 共识算法中的主节点的选取方式是按编号依次

轮流作主节点，选取方式较为随意，使得选取的主节点有可能是恶意节点。虽然恶意的主节点有可能被其他从节点识破并通过视图变更将其推翻，但是频繁的视图变更会极大的增加系统开销，导致共识效率降低。其次，PBFT 共识算法过程中采用两次全节点通信，需要比较大的网络传输和通信开销，需要进一步优化。

随着区块链技术的发展，新的共识算法层出不穷，其证明方式趋于多样化和混合化。FastBFT [15]算法试图解决性能问题，该算法将副本节点安排在一个树的结构当中，主节点作为树的根。在理想情况下，消息复杂度可以减少到 $O(nlogn)$，但是在失败的情况下消息复杂度为 $O(n^2)$。CBFT [16]算法通过区块缓存，节点变更来提高吞吐量，但是在交易处理的效率和达成共识的时延等方面需要进一步提升，并且在共识流程、区块同步和节点管理方面存在问题。Hot-stuff [17]是另一个基于 BFT 的协议，它将 PBFT 的视图更改消息复杂度减少到 $O(n)$，但在正常执行时，其消息复杂度与 PBFT 相当。SBFT [18]是一种使用称为 c 和 e 收集器的协议，它们是随机选择的，并在准备阶段和确认阶段负责收集签名。这些收集器有助于避免全网节点相互广播。在理想情况下，可以实现 $O(cn)$线性开销。但是如果收集器是恶意的，这将导致 SBFT 性能的下降，并导致 $O(n^2)$的通信开销。DDBFT [19]将 DPOS 算法应用于 PBFT 算法，使得 PBFT 算法具备动态性的特点，但是由于网络带宽有限，会造成网络阻塞、降低吞吐量。

本文基于现有的 PBFT 共识算法，提出了一种改进共识算法 PBFT+，其优势在于：1) 该算法通过信用机制来选取主节点，通过对节点进行信用层次划分，选取信用值最高的节点当选为主节点，保证了主节点的稳定性和最优性，同时通过备用主节点机制来减少视图切换所带来的系统开销，降低了时延；2) 通过在三阶段流程中的确认阶段中利用主节点来收集节点发送的确认消息，将二次消息复杂度减少到线性复杂度，降低了通信开销；3) 通过仿真实验进行验证，改进的 PBFT+共识算法能够有效地降低共识时延，减少系统资源的消耗。

2. 预备知识

2.1. PBFT 共识算法流程

PBFT 是一种状态机副本复制算法，状态机在任何一个副本节点上执行的结果都是相同的。同时，所有副本节点都必须从相同的状态开始执行。如图 1 所示，在 PBFT 算法中，一个主节点和其他副本节点的共识过程都处于一个视图 v 当中，视图 v 是连续递增的整数，在每个视图当中存在三种角色，分别是客户端、主节点、从节点。客户端发送请求。主节点接收客户端发送的请求并对其进行排序和编号分配，然后将请求向从节点广播。从节点负责对收到的请求进行验证，最后把验证结果反馈给客户端。

所有节点组成的集合用 R 进行表示，系统最多容忍恶意节点的数量为 f，存在如公式应该满足：

$$|R| = 3f + 1 \tag{1}$$

主节点的选取公式如下：

$$P = v \bmod |R| \tag{2}$$

其中：P 是副本编号，v 是视图编号，当主节点失效或者被从节点推翻时，发生视图变更，依照此公式选取新的主节点。

PBFT 算法的流程如图 1 所示。

请求阶段：当客户端 c 向主节点 p 发送请求消息 *message*。

预准备阶段：主节点收到消息后为消息分配编号，然后向从节点广播预准备消息。

准备阶段：从节点在接收到消息时会对其进行验证，如果同意该消息，则向其他节点广播准备消息。当每个节点收到 $2f+1$ 个从其他不同节点广播的准备消息，且与预准备消息一致，则准备阶段结束。

确认阶段。当节点完成准备阶段后，进入确认阶段，此时节点向其他节点发送确认消息。当每个节点已经接收了 2*f*+1 个确认消息且与预准备消息一致，则确认阶段结束。

回复阶段。当节点完成确认阶段后，向客户端发送回复消息。客户端在收到 *f*+1 个不同节点发送的相同回复消息时，请求执行成功。

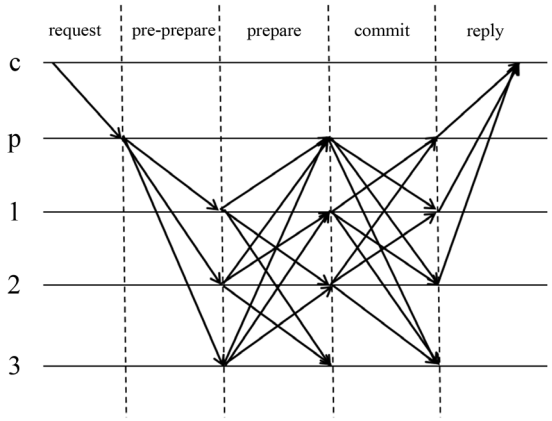

Figure 1. The algorithm flow of PBFT
图 1. PBFT 算法流程

2.2. PBFT 算法不足点分析

根据上一节对 PBFT 算法流程进行详细的分析，可以看出现有的 PBFT 共识算法具有以下不足：

1) PBFT 共识算法中的主节点通过取余法进行选取，主节点的选取方式较为随意，不能保证主节点的最优性，若连续选取出主节点为恶意节点，则将极大的影响系统性能，降低系统的可用性。

2) 在共识过程的三阶段广播协议中，有两次全网广播，严重占用系统带宽，当节点数目较多时，消息传递次数急剧增加，影响系统性能。

3. 改进方案

3.1. PBFT+算法思想

首先，对 PBFT 算法的确认阶段进行改进，这一阶段发送的确认消息不需要节点再次进行验证。消息在进入确认阶段后可以保证已经有 2*f*+1 的节点都完成了准备阶段，即该消息已经被合法数量的节点验证通过。PBFT+共识算法在确认阶段时，不再通过全网节点互相通信发送确认消息，而是全部发送到主节点上，由主节点负责收集确认消息，当主节点收集到 2*f*+1 个确认消息时，即完成确认阶段。一般情况下假设全网节点数为 *N*，那么完成一次共识过程需要消息传递次数为：

$$Z = N(N-1) + 2(N-1) \tag{3}$$

当节点较多时，可以节省对网络带宽的消耗，缩短共识过程。

其次，增加主节点选举过程，基于信用机制来选取最优主节点，保证主节点的最优性。通过备用主节点机制来避免频繁发生视图变更所带来的资源消耗，提高系统效率。

3.2. 主节点选取

3.2.1. 信用机制

定义 1 (信用层次) PBFT 算法中的区块产生和验证是由主节点来进行的，改进方案为了有效的监控主

节点行为，提出了信用机制，为每个节点设置信用层次，从而提高作恶节点成为主节点的难度。首先，通过每个共识节点的信用值来将节点分为三个层次，信任节点，待考察节点，故障节点。将三个层次的节点用一个信用值阈值来确定其范围：

信任节点：该层次的节点多次产生有效区块，其信用值范围为[0.7, 1]；

待考察节点：该层次的节点在区块产生的过程中没有无效区块产生，但是产生区块次数不多，其信用值范围为[0.3, 0.7)；

故障节点：该层次的节点产生过一次或者多次无效区块，其信用值范围为[0, 0.3)；

定义 2 (信用值)每个节点的信用值通过如下公式进行计算：

$$NC_i = NC_i^{init} + \left(t_{cur} - t_{pre} + 1\right)\sum_{i}^{j} T\left(i\right) * RP\left(i\right) \tag{4}$$

其中：NC_i 代表每个节点当前所获得的信用值，信用值范围为[0, 1]；i 代表每个节点的下标；NC_i^{init} 代表每个节点的初始信用值，信用值为 0.3；$T\left(i\right)$ 代表每个节点对应奖励或者惩罚的次数；$RP\left(i\right)$ 代表每个节点所获得的奖励或者受到的惩罚，当共识节点成功产生和验证区块后，系统会奖励 0.01 点信用值；当共识节点出现故障或有其他恶意行为，系统则惩罚 0.2 点信用值；t_{pre} 代表上一次产生区块的时间，t_{cur} 代表当前产生区块的时间，连续产生区块时，$t_{cur} - t_{pre}$ 为 0，否则不为 0。

定义 3 (信用层次动态变更)是指共识节点的信用层次会根据其产生的有效区块的数量和连续性动态变化。如图 2 所示，每个节点的初始信用值都为待考察层次，信用值为 0.3。当某一共识节点多次且连续产生有效区块后，其信用值上升到 0.7~1 之间，系统会根据该节点的信用层次升为信任节点。当该节点产生了无效区块后，但是次数较少，其信用值可能降到 0.3~0.7 之间，系统会将该节点降为待考察层次。当该节点多次产生无效区块后，其信用值下降到 0~0.3 之间，系统会将该节点降为故障节点层次。节点的信用层次划分是系统对节点是否为恶意节点的评判标准，信用层次低的节点在后续过程中处于劣势，而信用层次高的节点则更有优势。

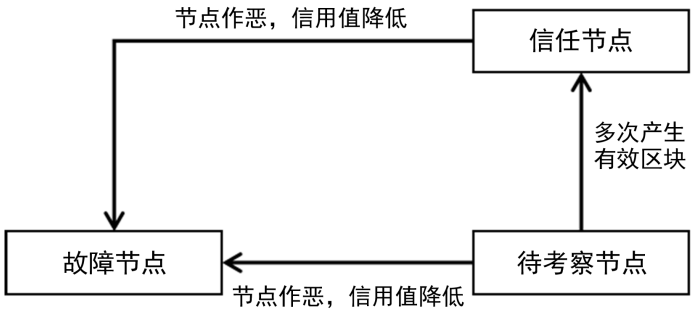

Figure 2. Credit level rise and fall
图 2. 信用层次升降

3.2.2. 奖惩机制

奖惩机制是针对共识节点经过信用值计算后的一种调节机制，用来对主节点和其他不同信用层次的从节点进行奖惩的措施。如图 3 所示，当某个主节点在该轮视图共识成功后，该主节点信用值加 0.01。当该主节点被从节点怀疑，并成功推翻该主节点时，该主节点的信用值减 0.2，同时该从节点的信用值加 0.01。奖惩机制作为一种调节机制，一方面保证了恶意节点受到了惩罚，诚实节点得到了奖励；另一方面，实现了快速剔除网络中的恶意主节点，避免恶意主节点影响系统的可靠性和安全性，有效地保证了共识节点的诚实性。

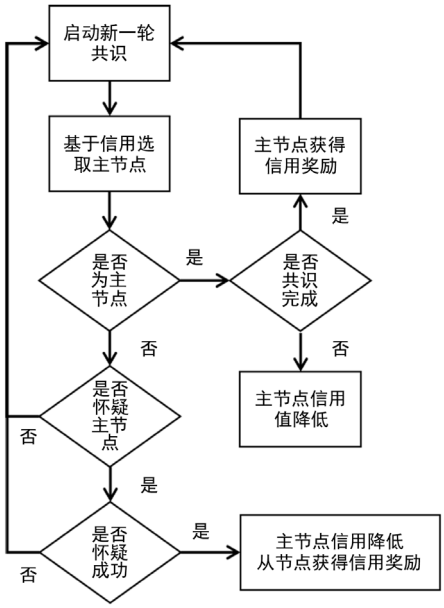

Figure 3. The process of reward and punishment
图 3. 奖惩机制流程

3.2.3. 备用主节点机制

针对主节点发生故障或者其他作恶行为，将会导致视图变更、浪费系统资源等情况，PBFT+利用备用主节点机制进行替换故障主节点。当故障主节点被其他副本节点推翻之后，不需要再进行视图变更，直接从备用主节点列表将信用值高的节点选为正式主节点，继续完成共识，保持系统的稳定性。通过上述机制确认了系统中信用值高的共识节点，改进方案将信用值高的前 n 个节点作为备用主节点，存储到备用主节点列表当中。如果有其他节点的信用值超过了备用主节点列表中的节点信用值，那么将信用值高的节点插入进来，将信用值低的节点剔除。添加备用主节点和备用主节点替换故障主节点的算法如算法 1 和算法 2 所示。

Algorithm 1. The adding standby master node algorithm
算法 1. 添加备用主节点

Algorithm 1 The adding standby master node algorithm

Input:
Blist：备用主节点列表
Node：节点
credit：节点信用值
round：共识阶段
Standby Master NodeValidation:
1. *Blist* ← *BackUpPrimaryNodeList*
2. for(*j=Blist[0]*:*Blist[n-1]*)
3. *credit* ← NC_i
4. *Node* ← *CurrentNode*
5. if (*credit*>*the least number of credit in Blist*)
6. *Add Node to Blist*
7. *Delete The low credit of Node*
8. end if
9. end for
Output:
当前备用主节点：*currentNode*
备用主节点列表：*Blist*

Algorithm 2. Replace the failed master node algorithm
算法 2. 替换故障主节点

Algorithm 2 Replace the failed master node algorithm

Input:
Blist：备用主节点列表
Node：节点
Pcurrent：节点信用值
credit：节点信用值
round：共识阶段
Master NodeValidation:
1. *Blist* ← *BackUpPrimaryNodeList*
2. *Pcurrent* ← *PrimaryNode*
3. while(*Blist is not empty*)
4. if (*the Pcurrent was the fault*)
5. *Select the highest credit of Node in Blist to PrimaryNode*
6. *Kick out the Pcurrent*
7. end if
8. Until the Blist is empty
Output:
当前备用主节点：*currentNode*
备用主节点列表：*Blist*

3.3. 改进共识算法

PBFT+共识算法引入信用机制，对全网各节点共识过程中的行为对节点进行信用评估，通过计算全网各节点的信用值来为节点划分不同的信用层次，诚实节点的信用值将会逐渐增加，而恶意节点的信用值将会逐渐递减，当选主节点的概率也将降低。PBFT+共识算法的工作流程如图 4 所示，具体步骤如下。

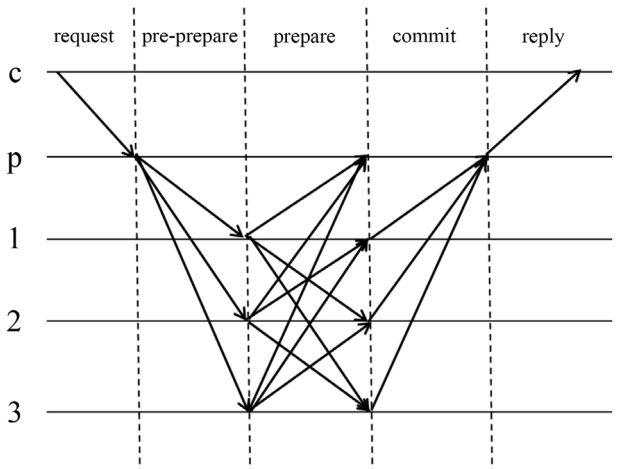

Figure 4. The algorithm flow of PBFT+
图 4. PBFT+算法流程

1) 客户端 *c* 发起交易 *tx*，并将交易发送给主节点 *P*。主节点收到交易后，验证交易是否有效。交易无效则直接删除，有效则打包到区块中。

2) 主节点发送预准备(pre-prepare)消息给全网其他共识节点，内容为<PRE-PREPARE, h, d, t, P>。其中 *h* 是区块的高度，*d* 是消息摘要，*t* 是时间戳，*P* 是当前主节点。

3) 当其他共识节点收到主节点发送的消息后，则进入准备(PREPARE)阶段，共识节点立即验证该消

息。如果验证通过，则节点向主节点及其他节点发送验证通过消息，当全网每个节点收到 2*f*+1 不同节点发送验证通过的消息，则完成准备阶段。如果有 *f*+1 个不同的节点验证不通过，则推翻当前主节点，进入下一个视图，然后从备用主节点列表中选取信用值最高的节点当选为主节点，继续进行共识。

4) 当节点完成准备阶段后，进入确认(COMMIT)阶段。该阶段由主节点负责收集确认消息，当主节点收集到 2*f*+1 个不同节点发送的确认消息且与准备消息一致，则确认阶段结束。

当节点完成确认阶段后，直接由主节点向客户端发送回复消息，当客户端收到主节点发送的回复消息后，回复阶段完成。

4. 实验与分析

本章详细叙述了仿真实验的细节和相关结果分析，从通信开销、带宽消耗、时延、吞吐量等方面对 PBFT+共识算法进行验证测试，同时与 PBFT、CBFT 共识算法进行对比，以此验证 PBFT+共识算法的有效性。本实验是在 PC 配置为 Intel Core i7-7700HQ 3.60GHz CPU 和 16GB 内存上进行的。通过开启不同的服务器端口在本地创建不同的共识节点，实验网络环境一共建立了 10 个共识节点。

4.1. 通信开销与带宽测试

假定系统中共识节点数目为 *n* (*n* > 4)，可以根据 PBFT、CBFT 与 PBFT+的共识流程分别进行对比分析。其中 PBFT、CBFT 和 PBFT+共识算法的三阶段通信次数如图 5 所示，在三阶段流程中 PBFT 算法的总通信开销是多项式级别的，随着节点的增加，通信开销急剧增长，影响系统性能，而改进的 PBFT+共识算法对三阶段中确认阶段进行优化，由主节点来收集确认信息，将通信开销降低了一半，可以有效的减少通信开销，提升共识效率。

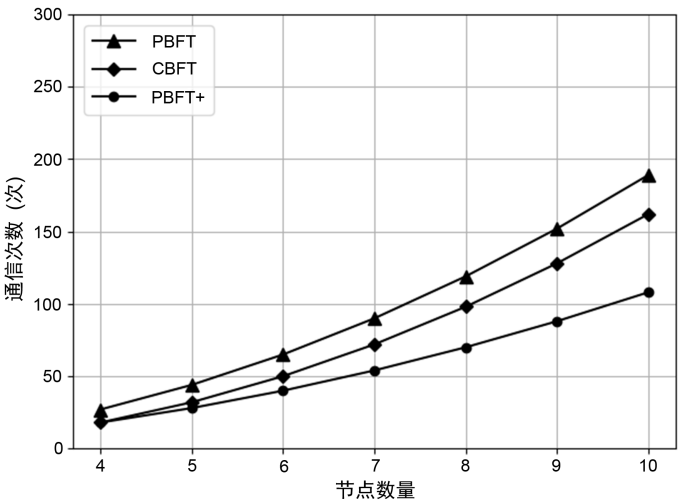

Figure 5. Comparison of consensus communication overhead
图 5. 共识通信开销对比

带宽也是衡量共识算法性能的重要指标。在区块链网络中，PBFT+、PBFT、CBFT 三种算法都需要进行数据的传输和消耗，带宽消耗越少共识算法性能越好。图 6 为改进前后的带宽消耗随节点数量 N 变化的对比图。从图 7 可以看出，在相同条件下，改进后的 PBFT+算法的带宽消耗要小于 PBFT 算法。改进后的 PBFT+算法在确认阶段中减少了一次对全节点广播，且通过信用机制来选取主节点，减少视图切换的次数，使得共识成功率提高，这都减少了带宽资源的消耗。

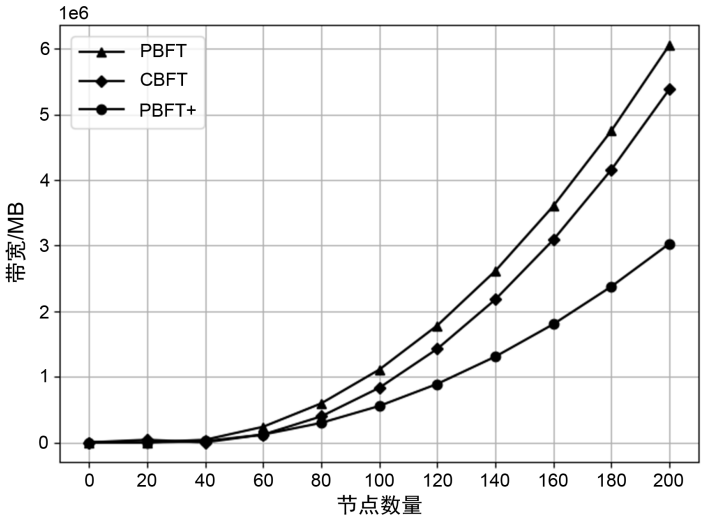

Figure 6. Comparison of consensus communication overhead
图 6. 共识通信开销对比

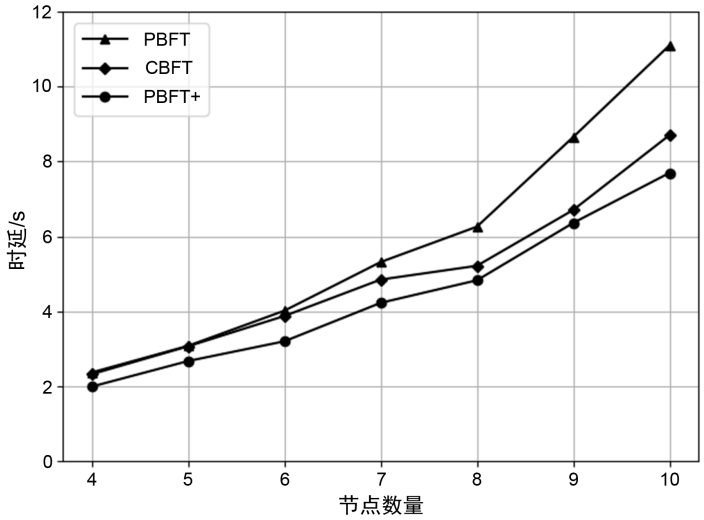

Figure 7. Comparison of consensus communication overhead
图 7. 共识通信开销对比

4.2. 时延测试

共识时延是指完成一次共识所需要的时间。它决定了交易和区块确认的速度，是衡量共识算法共识效率和运行速度的重要指标。较低的时延使系统更加安全可靠。一次共识所需要的时间是从客户端发出请求到请求执行成功所经历的时间。即：

$$T_{delay} = T_c - T_p \tag{5}$$

T_{delay} 为一次共识所需要的时间，T_c 为确认区块执行成功的时间，T_p 为交易产生的时间。为了方便对照，需要对条件和变量进行控制。一个区块的交易量设置为 50，在此条件下分别对 4 到 10 个节点进行多次实验，每组节点得到的数据取平均值。时延对比如图 7 所示，随着系统内共识节点的增加，三种算法的时延都有不同程度的增长，但是 PBFT+共识算法的时延明显低于 PBFT 与 CBFT 共识算法。CBFT 共识算法通过区块缓存，节点变更来降低时延，但是在交易处理的效率和达成共识的时延等需要进一步

提升。而 PBFT+共识算法通过信用机制来选取信用值最高的节点作为主节点，能够减少视图切换的次数，通过区块高度同步机制来快速达成共识，提升系统效率。

4.3. 吞吐量测试

吞吐量代表一个系统在单位时间内处理事务的能力，通常用 TPS(Transaction Per Second, 每秒交易数)来表示。在区块链中的 TPS 表示交易发出到交易确认并写入区块上的交易总数与消耗时间的比值。随着每个区块的容量增加，吞吐量也相应的增大，共识时延和网络负载也会增加，因此当区块容量大到一定程度时吞吐量有所下降。在区块链网络中，区块的大小也会影响 TPS，因此本实验将一个区块中包含的交易固定为 50 个，并分别测试 4 到 10 个节点时 PBFT 算法、CBFT 算法、PBFT+算法运行的平均 TPS。测试结果如图 8 所示，随着节点总数的增加，三种共识算法的 TPS 都有略微的下降，同时本文改进的 PBFT+共识算法在吞吐量上明显高于 PBFT 与 CBFT 共识算法。

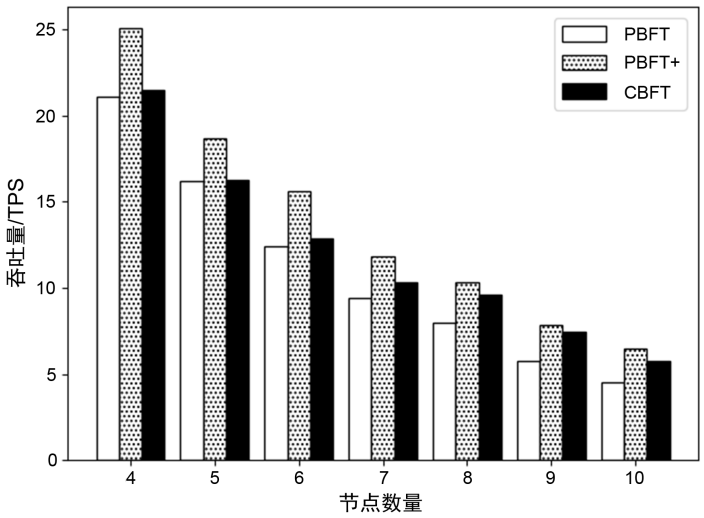

Figure 8. Comparison of throughput
图 8. 吞吐量对比

5. 总结与展望

本文针对 PBFT 共识算法中存在主节点选取方式随意、三阶段广播通信开销大等导致的频繁视图变更、通信开销大、共识无法快速达成等问题，基于 PBFT 共识算法提出了 PBFT+共识算法。首先该算法通过信用机制来选取信用值最高的节点作为主节点，可以保证主节点的最优性，通过信用层次动态变更有效的提升了信用值高的节点当选为主节点的概率，而降低了信用值低的节点当选的概率。其次，通过奖惩机制实现了快速剔除网络中的恶意主节点，避免恶意主节点影响系统的安全性，然后利用备用主节点机制来减少视图变更所带来的系统开销；通过优化三阶段共识过程，减少了通信开销。通过实验验证了 PBFT+共识算法的可行性，比 PBFT 算法更有效，减少了系统资源的消耗，降低了共识时延。

基金项目

国家自然科学基金资助项目(62076073,91748107)；广东省基础与应用基础研究基金(No.2020A1515010616)；广东省引进创新科研团队计划资助项目(2014ZT05G157)。

参考文献

[1] Wang, W., Hoang, D.T., Hu, P., *et al*. (2019) A Survey on Consensus Mechanisms and Mining Strategy Management in Blockchain Networks. *IEEE Access*, **7**, 22328-22370. https://doi.org/10.1109/ACCESS.2019.2896108

[2] 单进勇, 高胜. 区块链理论研究进展[J]. 密码学报, 2018, 5(5): 484-500.

[3] Ferrag, M.A., Derdour, M., Mukherjee, M., *et al*. (2018) Blockchain Technologies for the Internet of Things: Research Issues and Challenges. *IEEE Internet of Things Journal*, **6**, 2188-2204.

[4] 袁勇, 王飞跃. 基于区块链技术发展现状与展望[J]. 自动化学报, 2016, 42(4): 481-494.

[5] 何蒲, 于戈, 张岩峰. 区块链技术与应用前瞻综述[J]. 计算机科学, 2017, 44(4): 1-7+15.

[6] 姚前, 朱烨东. 中国区块链发展报告 2019[M]. 北京: 社会科学文献出版社, 2019.

[7] Al-Jaroodi, J. and Mohamed, N. (2019) Blockchain in Industries: A Survey. *IEEE Access*, **7**, 36500-36515. https://doi.org/10.1109/ACCESS.2019.2903554

[8] Li, J. and Wolf, T. (2016) A One-Way Proof-of-Work Protocol to Protect Controllers in Software-Defined Networks. *ACM/IEEE Symposium on Architectures for Networking and Communications Systems* (*ANCS*), Santa Clara, 17-18 March 2016. https://doi.org/10.1145/2881025.2889481

[9] Kiayias, A., Russell, A., David, B., *et al*. (2017) Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol. In: *Annual International Cryptology Conference*, Springer, Cham, 357-388. https://doi.org/10.1007/978-3-319-63688-7_12

[10] Luo, Y., Chen, Y., Chen, Q., *et al*. (2018) A New Election Algorithm for DPos Consensus Mechanism in Blockchain. 2018 7*th International Conference on Digital Home* (*ICDH*) *IEEE*, Guilin, 30 November-1 December 2018. https://doi.org/10.1109/ICDH.2018.00029

[11] Lamport, L. (1998) The Part-Time Parliament. *ACM Transactions on Computing Surveys*, **16**, 133-169. https://doi.org/10.1145/279227.279229

[12] Castro, M. and Liskov, B. (1999) Practical Byzantine Fault Tolerance. *OSDI*, Vol. 99, 173-186

[13] Lamport, L., Shostak, R. and Pease, M. (1982) The Byzantine Generals Problem. *ACM Transactions on Programming Languages and Systems* (*TOPLAS*), **4**, 382-401. https://doi.org/10.1145/357172.357176

[14] Androutsellis-Theotokis, S., Spinellis, D., *et al*. (2004) A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Transactions on Computing Surveys*, **36**, 335-371. https://doi.org/10.1145/1041680.1041681

[15] Liu, J., *et al*. (2019) Scalable Byzantine Consensus via Hardware-Assisted Secret Sharing. *IEEE Transactions on Computers*, **68**, 139-151. https://doi.org/10.1109/TC.2018.2860009

[16] 李剑锋. 基于拜占庭容错机制的区块链共识算法研究与应用[D]: [硕士学位论文]. 郑州: 郑州大学, 2018: 14-15, 31-56.

[17] Abraham, I., Gueta, G. and Malkhi, D. (2018) Hot-Stuff the Linear, Optimal Resilience, One-Message BFT Devil.

[18] Gueta, G.G., Abraham, I., Grossman, S., *et al*. (2018) SBFT: A Scalable Decentralized Trust Infrastructure for Blockchains.

[19] 刘肖飞. 基于动态授权的拜占庭容错共识算法的区块链性能改进研究[D]: [硕士学位论文]. 杭州: 浙江大学.