文章编号:1673-0291(2019)05-0058-07 **DOI**:10.11860/j.issn.1673-0291.20190051

一种面向区块链的优化 PBFT 共识算法

方维维[1],王子岳[1],宋慧丽[1],王云鹏[1],丁 毅[2]
(1.北京交通大学 计算机与信息技术学院,北京 100044;2.北京物资学院 信息学院,北京 101149)

摘 要:区块链技术具有去中心化,数据不可篡改和数据透明等特点,使得该技术的应用领域不断扩展,但目前应用于区块链系统的共识算法存在着资源浪费和共识效率较低等问题,限制了区块链技术的发展.针对此问题,基于实用拜占庭容错算法(Practical Byzantine Fault Tolerance,PBFT),算法的基本思想,提出了一种优化的共识算法.该算法引入积分机制,根据节点积分挑选参与共识的节点,以降低网络中的通信开销;在不存在拜占庭节点的情况下,优化 PBFT 算法的一致性协议;引入升降级机制,动态更新参与共识的节点集合,以保证算法在大部分时间内都执行优化一致性协议.实验结果表明:与 PBFT 算法相比,本文提出的共识算法将共识过程的时间复杂度从 $O(N^2)$ 下降到 $O(N)$,有效降低了网络中的通信开销,平均时延从 55 ms 降到 37 ms,平均吞吐量从 342 TPS 提升到 677 TPS.

关键词:区块链;共识算法;PBFT;拜占庭错误
中图分类号:TP311.13 **文献标志码**:A

An optimized PBFT consensus algorithm for blockchain

FANG Weiwei[1], *WANG Ziyue*[1], *SONG Huili*[1], *WANG Yunpeng*[1], *DING Yi*[2]
(1.School of Computer and Information Technology, Beijing Jiaotong University, Beijing 100044, China;
2. School of Information, Beijing Wuzi University, Beijing 101149, China)

Abstract: Blockchain technology has the characteristics of decentralization, data tamper-resistantand data transparency, which makes the application field of blockchain technology expand continuously. However, the consensus algorithm applied to the blockchain system currently has problems of resource waste and low efficiency, which limits the development of blockchain technology. Aiming at this problem, this paper proposes an optimized consensus algorithm based on the basic idea of PBFT(Practical Byzantine Fault Tolerance) algorithm. This algorithm introduces an integration mechanism to select nodes participating in the consensus based on node points to reduce communication overhead in the network, and optimizes the consistency protocol of the PBFT algorithm in the absence of Byzantine nodes. It also introduces a lifting level mechanism to dynamically update the nodes participating in the consensus process to ensure that the algorithm performs the optimized coherence protocol most of the time. The experiment results show

收稿日期:2019-05-17
基金项目:北京市社会科学基金研究基地项目(18JDGLB026);北京市教育委员会科技计划一般项目(KM201910037003);北京市智能物流系统协同创新中心开放课题项目(BILSCIC-2019KF-10)
Foundation items: Research Base Project of Beijing Municipal Social Science Foundation (18JDGLB026); Science and Technique General Program of Beijing Municipal Commission of Education (KM201910037003); Beijing Intelligent Logistics System Collaborative Innovation Center Open Project (BILSCIC-2019KF-10)
第一作者:方维维(1981—),男,安徽芜湖人,副教授,博士.研究方向为计算机网络.**email**:fangww@bjtu.edu.cn.

引用格式: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 Weiwei, WANG Ziyue, SONG Huili, et al. An optimized PBFT consensus algorithm for blockchain[J]. Journal of Beijing Jiaotong University, 2019, 43(5):58-64. (in Chinese)

that the new consensus algorithm reduces the time complexity of the consensus process from $O(N^2)$ to $O(N)$, which effectively reduces the communication overhead in the network, and decreases average delay from 55 ms to 37 ms, increases average throughput from 342 TPS to 677 TPS, as compared to the original PBFT algorithm.

Keywords: blockchain; consensus algorithm; PBFT; Byzantine fault tolerance

区块链(Blockchain)是以比特币为主的数字加密货币的重要底层技术,最早是在 2008 年由化名为中本聪的学者提出.他提出了一种不需要任何第三方金融机构参与,由交易双方直接进行交互的点对点电子现金系统,即比特币[1].随着比特币等数字加密货币的关注度不断提升,越来越多的学者开始对区块链技术进行研究.区块链技术拥有去中心化、数据安全、数据不可篡改等特点[2],使得其在数字货币、数据存储、数据鉴定和金融交易等领域都有很好的应用前景[3].但目前区块链系统存在着资源消耗过大以及交易时延较低等问题[4],限制了区块链技术的发展.

共识算法是区块链最为核心的技术,影响着区块链系统的性能,目前共识算法按照区块链的类型可以分为公有链共识、私有链共识和联盟链共识三类[4].工作量证明[5](Proof of Work,PoW)是应用于公有链的共识算法,但其依靠计算机计算能力来完成共识的机制带来了资源浪费的问题,文献[6]提出了股份授权证明(Delegated Proof of Stake,DPoS)算法来解决这一问题,但该算法依赖数字货币才能完成共识操作,而实际的区块链技术的应用场景不存在数字货币,导致此算法的实用型较差;不同于公有链,私有链主要应用于企业的内部,网络中不存在拜占庭节点,拜占庭节点是指故意发送错误信息而导致整个网络不能达成共识的节点,因此私有链中大多采用传统的分布式一致性算法 Raft[7]来完成共识操作;联盟链介于公有链和私有链之间,其网络由经过授权的联盟成员共同维护,联盟链中主要使用 PBFT[8]算法,但此算法为解决拜占庭将军问题,需要通过时间复杂度为 $O(N^2)$ 的网络通信才能完成共识(N 为网络中节点的总数),给网络带宽带来了巨大的压力.针对这一问题,文献[9]提出将 DPoS 和 PBFT 算法相结合,根据节点的股份投票选出部分节点来参与共识,减少了共识过程中节点间的通信量,但此算法与 DPoS 算法有相同问题,即需要数字货币才能完成共识操作;文献[10]和[11]则提出在网络中不存在拜占庭节点的情况下,简化 PBFT 算法的一致性协议,使共识过程的时间复杂度从 $O(N^2)$ 降到了 $O(N)$,但在存在拜占庭节点的情况下,会执行和 PBFT 算法类似的过程,算法的性能也退化到和 PBFT 一样的地步.

目前联盟链更加适用于各种应用场景的需要,因此本文作者针对联盟链使用的 PBFT 算法存在的问题,基于 PBFT 算法的思想,提出了一种新的算法 SPBFT(Selection-based Practical Byzantine Fault Tolerance).SPBFT 算法对 PBFT 算法的一致性协议进行了优化,并引入积分机制和升降级机制,使得算法在大部分时间内都执行优化的一致性协议.最后通过对比实验证明了在相同状况下,SPBFT 算法的共识效率和吞吐量更高,网络中的通信开销更低.

1　共识算法分析

1.1　工作量证明算法

工作量证明算法(PoW)适合应用在区块链公有链系统中,工作量证明过程可以分成求解和验证两个阶段,求解阶段是进行大量复杂的数学计算,求得一个数学解;验证阶段则是进行简单的数学计算来验证求得的解是否正确,在极短的时间内即可完成.这是一种完全不对称的计算过程,其原因是算法设定求解成功的节点拥有记账的权利,但网络中存在着大量节点,系统希望每一笔交易都只有一个节点拥有记账权,以免造成系统记账混乱,因此增加求解的难度来实现此机制.PoW 共识算法需要通过大量计算来保障区块链数据的一致性和正确性,导致此算法需要消耗大量的计算资源和电力资源;并且为保证数据的安全性,PoW 算法的出块时间较长,导致 PoW 共识效率较低.

1.2　股份授权证明算法

DPoS 共识算法的共识过程类似于“董事会决策”,网络中拥有数字货币也就是拥有股份的节点可以进行投票来选择共识代表,获得票数最高并且希望成为代表的前 101 个节点加入“董事会”.“董事会”成员按照约定的记账时间轮流生成区块,如果某代表在规定的时间片内没有生成区块,则会被踢出“董事会”,“董事会”的成员按照维护周期(一般为 1 d)更新一次.DPoS 共识算法较 PoW 共识效率更高,也更加节约资源.但其依赖于数字货币才能完成共

识的机制，使得 DPoS 算法不适合应用于大部分区块链应用中.

1.3 实用拜占庭容错算法

PBFT 共识算法是适用于联盟链中，其设计的目的是解决拜占庭将军问题.拜占庭将军问题证明了在拜占庭节点数目为 f，网络中节点总数 N 大于 $3f$ 情况下，分布式系统能够达成共识，其算法的时间复杂度为 $O(N^{f+1})$. PBFT 共识算法网络中同样允许存在 f 个拜占庭节点，也要求 $f<(N-1)/3$，但 PBFT 将算法时间复杂度降为 $O(N^2)$，让其可以应用在实际的分布式系统中.PBFT 是一种基于状态机副本复制的分布式一致性算法，由一致性协议，视图更换协议和检查点协议组成.一致性协议用来保证全网所有的节点保存数据的一致性，其通过三阶段节点间的互相通信来实现；在一致性协议执行过程中，如果主节点出现故障时，会触发视图更换协议来更换主节点；检查点协议则是定时触发，用来清理一致性协议执行过程中各个节点存储的通信消息，并同步各个节点的状态.

一致性协议是 PBFT 算法能够完成共识的核心协议，主要分为预准备(pre-prepare)、准备(prepare)和提交(commit)3 个阶段，其执行过程见图 1.

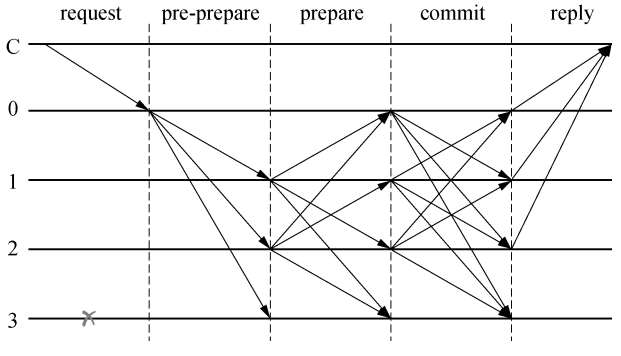


图 1 PBFT 算法一致性协议执行过程

Fig.1 Execution process of the consistency protocol in PBFT algorithm

1)预准备阶段：主节点会将接收到的客户端发送的请求生成预准备消息，并将预准备消息广播给所有从节点，消息格式为＜PRE-PREPARE，v，n，d，m＞，其中 v 为视图编号，m 为客户端发送的消息，d 为 m 进行 Hash 运算后的结果，n 为消息编号.

2)准备阶段：从节点收到主节点发送的预准备消息后，会生成准备消息，并将准备消息广播到其他节点，同时将预准备消息和准备消息写进日志文件中.其消息格式为＜PREPARE，v，n，d，i＞，i 为节点编号.在此阶段，每个节点收到所有其他节点广播的准备消息，节点会验证消息的真实性，将收到的准备消息和自己日志中的准备消息进行比较，主要比较 n、v、m 字段，如果超过 $(f+1)$ 个准备消息是正确的，则会进入确认阶段.

3)确认阶段：所有节点生成确认消息并广播到其他节点，消息格式为＜COMMIT，v，n，d，i＞，在此阶段会完成同准备阶段相同的验证工作，当验证通过后，此次请求的共识过程才能完成.

2 改进共识算法

通过图 1 可以看到，PBFT 算法在执行一致性协议时节点间会进行大量的通信，随着节点数量和交易数量的增多，网络通信量会快速增长，从而增加带宽的压力，进而影响算法的共识效率.因此，本文针对此问题，并结合联盟链的特点，在不存在拜占庭节点的情况下，优化了一致性协议，以降低节点间的通信量；同时引入了积分机制和升降机机制，使得算法能够在网络中出现拜占庭节点的情况下，快速地恢复到最优状态，使得算法在大部分时间内都能够执行优化的一致性协议.

2.1 改进思路

在 1.3 节对 PBFT 算法中的一致性协议进行了分析，在其准备阶段和确认阶段的验证过程中，节点接收到超过 $(f+1)$ 个正确的消息，就可完成验证进入下一阶段.通过分析，可以发现，在算法进行一致性协议的过程中，各个节点只需要接收正常节点发送的消息，并且在正确的消息的数量超过 $(f+1)$ 个情况下，系统就可以达成共识.

因此，SPBFT 算法引入积分机制，根据节点的积分将网络中的节点分成两类，一类为共识节点，其数量为 $(N-f)$，共识节点参与共识过程；一部分为候选节点，其数量为 f，候选节点不参与共识，但需要接受共识结果.SPBFT 算法会根据共识节点的状态来选择执行不同的一致性协议，主要分两种情况：

1)共识节点中不存在拜占庭节点，拜占庭节点全部在候选节点中，此时共识节点会执行优化一致性协议.

2)共识节点中存在拜占庭节点，这时网络中所有的节点会执行 PBFT 算法的一致性协议，保证算法的容错性.

SPBFT 共识算法是为联盟链设计的，在联盟链中，大部分节点都是诚实节点，因此，SPBFT 算法在绝大多数时间内都会执行优化一致性协议.同时在算法中引入升降级机制，当共识节点中出现了拜占庭节点，在执行完一次 PBFT 算法的一致性协议后，拜占庭节点会被剔除，并从候选节点中根据积分

选择一个节点加入共识节点,保证共识节点大概率都为诚实节点,SPBFT 算法将继续执行优化一致性协议完成接下来的共识操作.

2.2　优化一致性协议设计

PBFT 共识算法的一致性协议在运行过程中需要完成两次复杂度为 $O(N^2)$ 的节点通信,这样设计是为了在网络中存在拜占庭节点的情况下,算法仍能够完成共识.SPBFT 算法在不存在拜占庭节点的情况下,对一致性协议进行优化,使其在完成复杂度为 $O(N)$ 的节点通信后,算法就能够完成共识.

本文参考文献[12]中提出的一种简化一致性协议,并结合本文的改进思路,设计了如图 2 所示的优化一致性协议.

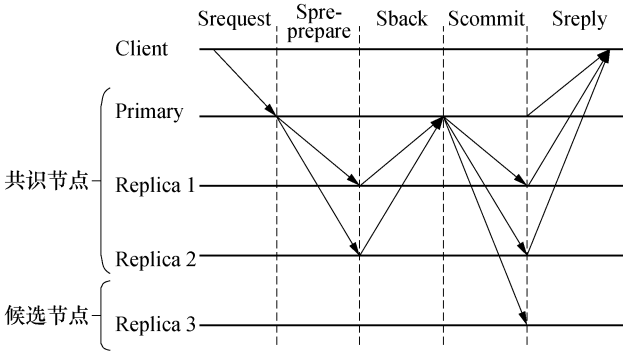


图 2　优化一致性协议执行过程

Fig.2　Execution process of the optimized consistency protocol

协议的具体执行过程如下:

1)优化一致性协议发送请求阶段(Srequest):同 PBFT 算法的请求阶段一样,客服端向主节点发送请求消息,消息格式为<SREQUEST, o, t, c >,其中 o 为请求执行状态机,t 为时间戳,c 表示客户端编号.

2)优化一致性协议预准备阶段(Spre-prepare):主节点接收客户端发送的请求后会生成预准备消息,并将预准备消息广播到所有共识节点,消息格式如下<<SPRE-PREPARE, v, n, d, g >, w, m >.其中 w 为节点积分信息,此信息用于对节点进行升降级处理,g 为 w 进行 Hash 计算的结果.

3)优化一致性协议反馈阶段(Sback):共识节点接收到主节点发送的预准备消息后,首先会判断预准备消息中的 g 值和本地的 g 值是否相同,如果不同,则更新本地的积分信息 s;之后会生成反馈消息,并将消息发送给主节点.消息格式为<SBACK, v, n, d, i >.

4)优化一致性协议确认阶段(Scommit):主节点会接收到所有共识节点发送的反馈信息,并且在所有反馈信息完全相同的情况下,主节点会生成确认消息并广播到网络中的所有节点,消息格式为<SCOMMIT, v, n, d, a >,其中 a 为确定添加信息,表示主节点已经确认添加.所有的节点接收到确认消息后,将次交易信息添加都本地内存中.

2.3　算法的实现

SPBFT 算法是在 PBFT 算法上进行改进,其也是通过网络中节点之间的互相通信来完成共识操作,其通信过程根据一致性协议来执行.本算法对 PBFT 算法的一致性协议进行改进,设计了一种优化一致性协议,以减少共识过程中节点间的通信量.SPBFT 算法的流程图如图 3 所示.

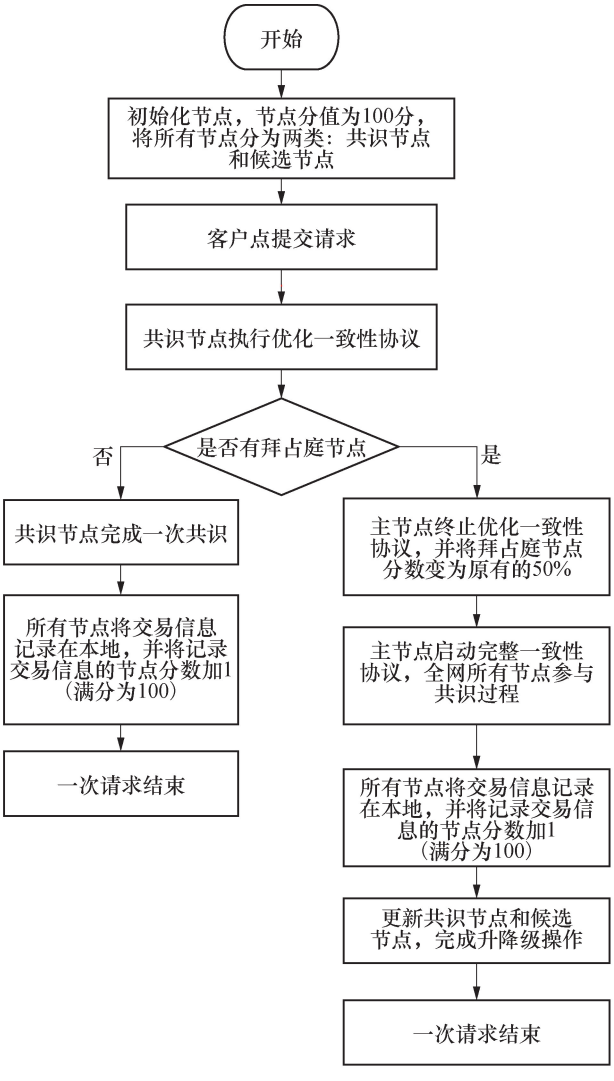


图 3　SPBFT 算法流程

Fig.3　Flow chart of the SPBFT algorithm

SPBFT 算法的具体执行过程:

1)进行节点的初始化工作.首先,用 $\{0,1,2,\cdots,|S|-1\}$ 共 N 个整数对网络中的节点进行编号,将节点的积分初始化为 100 分,设置 $f=(N-1)/3$;其次,初始化共识节点集合 CS 和候选节点集合 DS,$CS=\{0,1,2,\cdots,(N-f-1)\}$,$DS=\{(N-f),$

…,$(N-1)$},共识节点的数量为:$|CS|=(N-f)$,候选节点的数量为:$|DS|=f$.

2)客户端向主节点发送一笔交易请求,主节点接收请求后对请求消息进行编号,之后主节点执行优化一致性协议.

3)所有共识节点执行优化一致性协议,在优化一致性协议的确认阶段,会对共识节点的状态进行判断.在此阶段,主节点会接收所有共识节点的反馈消息,并判断反馈消息<SBACK, v,n,d,i >的正确性,将其与本地保存的 v,n 与之相同的预准备消息<<SPRE-PREPARE, v,n,d,g >,w,m >进行对比,判断字段 d 对应的值是否相同的,一旦交易信息被篡改,其 Hash 值一定会改变.因此,系统会出现两种比较结果,算法会根据不同的比较结果进行不同操作.

①若 d 值相同,说明此条反馈消息是正确的,当主节点判断接收到的所有反馈消息都是正确的后,共识节点会继续执行优化一致性协议,完成此次共识过程.

②若 d 值不同,说明交易的消息内容被篡改过,可以判定发送此条反馈消息的节点为拜占庭节点,此时主节点会执行 3 个步骤:首先,主节点会终止此次优化一致性协议的运行,因为优化一致性协议不支持拜占庭容错,此时系统不能达成共识;其次,主节点会执行 PBFT 算法的一致性协议来完成此次交易的共识过程,保证系统的容错性;最后,主节点会根据节点的积分情况,对节点进行升降级处理,更新共识节点集合和候选节点集合,来保证共识节点中大概率都是诚实节点,然后在下一次共识过程中继续执行优化一致性协议.

3 实验及结果分析

基于 Java 编程语言实现了一个小型的多节点区块链实验系统,在该系统中对原 PBFT 算法和本文提出的 SPBFT 算法进行了验证.在存在拜占庭节点和不存在拜占庭节点两种情况下,在交易时延、吞吐量和通信开销三个方面对两种算法进行比较分析.

3.1 交易时延

交易时延是指客户端向主节点发送一个交易请求到客户端确认完成共识的时间间隔.为不失一般性,交易时延取 200 次交易的平均值,并在不同节点个数的情况下对交易时延进行测试.图 4 所示为两种算法的交易时延对比.

图 4(a)中,SPBFT 算法在不存在拜占庭节点的情况下会执行优化一致性协议,可以看出,其交易时延明显优于 PBFT 算法.并且随着节点数目的增加,PBFT 算法的交易时延增长较快,而 SPBFT 算法的交易时延较为稳定,增长缓慢.因此,在节点较多的情况下,SPBFT 算法的优势更为明显,平均时延从 55 ms 下降到 37 ms.与图 4(a)相比,图 4(b)中 PBFT 算法的交易时延基本没有发生变化,而 SPBFT 算法因要进行一致性协议的切换以及共识节点的更新等操作,其交易时延明显增长.

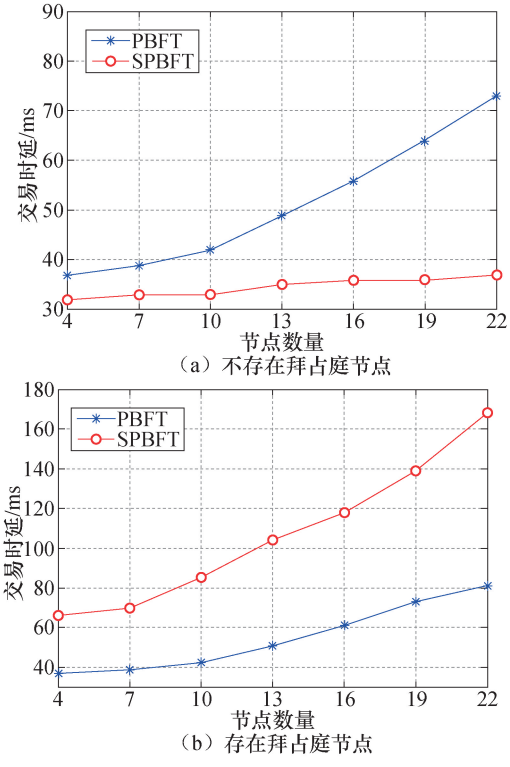
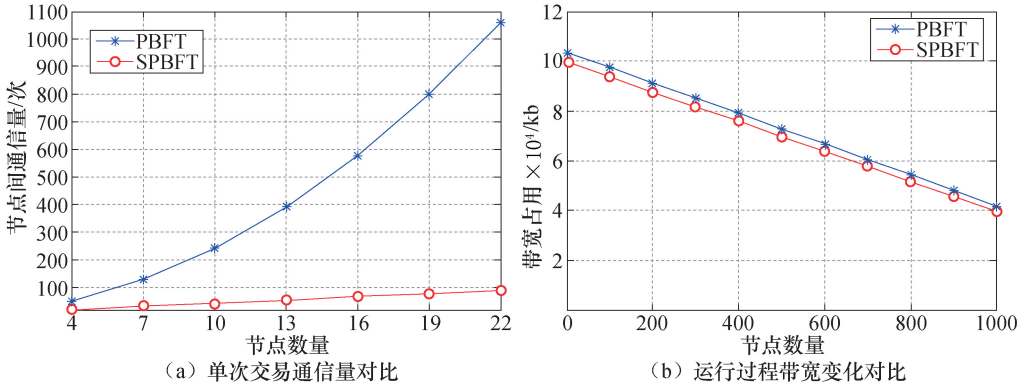


图 4 两种算法的交易时延比较

Fig.4 Comparisons on consensus completion time of the two algorithms

3.2 通信开销

通信开销指的是网络中节点在共识过程中产生的通信量.SPBFT 算法设计的主要目的就是减少通信量,降低通信开销,减轻带宽的压力,图 5 所示为两种算法的通信量对比.图 5(a)中,随着网络中在节点数目的不断增多,SPBFT 算法由于采用优化一致性协议,其通信量呈线性增长;而 PBFT 算法需要通过复杂度为 $O(N^2)$ 的节点通信来完成共识,其通信量增长快速.因此,可以看出,SPBFT 算法降低网络的通信开销,且效果显著.图 5(b)中设置网络中的节点个数为 4,客户端发送 1 000 条请求消息,每隔 100 ms 记录一次带宽信息,观察 1 s 内带宽的变化.假设每条消息的大小为 10 kb.可以看出,在算法执行的过程中,SPBFT 算法带宽占用要小于 PBFT 算法,在一定程度上减轻了共识过程中带宽的压力.



（a）单次交易通信量对比　　（b）运行过程带宽变化对比

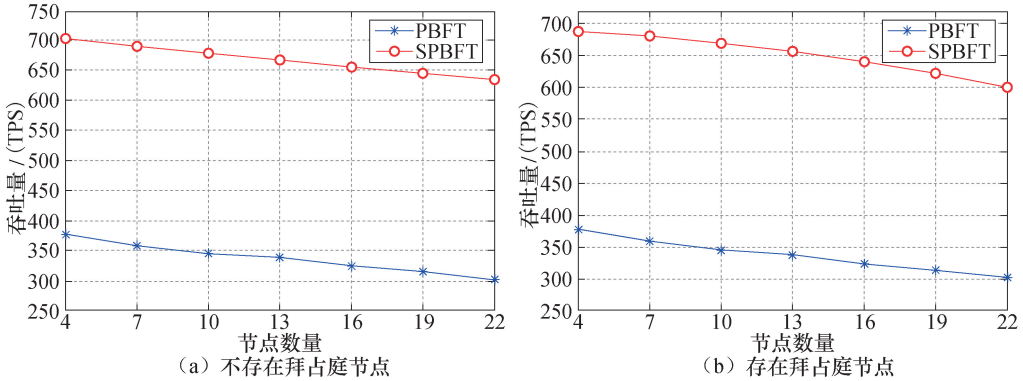
图 5　两种算法通信量对比

Fig.5　Comparisons on communication overhead of the two algorithms

3.3　吞吐量

吞吐量指的是在单位时间内完成的交易的数量,一般用 TPS(Transaction Per Second)来表示.实验中设置客户端发送 2 000 条请求,记录每秒能够完成共识的交易数量,并在不同节点个数的情况下进行测试.图 6 所示为两种算法的吞吐量对比图.

图 6(a)中,随着网络中节点数量的增多,两种算法的吞吐量都呈现下降的趋势,但从总体来看,SPBFT 算法的吞吐量要远高于 PBFT 算法的吞吐量,平均吞吐量从 342 TPS 提升到 677 TPS.图 6(b)中可以看出,随着网络中节点数量的增多,两种算法的吞吐量也都是呈现下降的趋势,但 SPBFT 算法的下降趋势较大,其主要原因是 SPBFT 算法在运行过程中,在出现拜占庭节点的情况下,算法会在一段时间内暂停共识过程,完成一致性协议的转换,执行 PBFT 算法的一致性协议,在这段时间内算法只会完成一笔交易的共识工作,影响了算法的吞吐量.但 SPBFT 算法吞吐量仍然大于 PBFT 算法的吞吐量.



（a）不存在拜占庭节点　　（b）存在拜占庭节点

图 6　两种算法的吞吐量比较

Fig.6　Comparisons on system throughput of the two algorithms

4　结论

1)基于 PBFT 算法的思想,提出一种优化的共识算法.在不存在拜占庭节点的情况下,优化了 PBFT 算法的一致性协议,以降低节点在共识过程中的通信量;同时引入积分机制和升降级机制,动态的更新参与共识的节点,使得算法能够快速恢复到最优状态,保证算法在大部分时间内都执行优化的一致性协议

2)实验结果表明,在相同的条件下与 PBFT 算法对比,本文提出的 SPBFT 算法将共识过程的时间复杂度从 $O(N^2)$ 下降到 $O(N)$,有效降低了网络中的通信开销,平均时延从 55 ms 降到 37 ms,平均吞吐量从 342 TPS 提升到 677 TPS.

在未来的工作中,可以对算法中积分机制和升降级机制进行改进,增加节点的动态加入和退出网络的功能,对算法完成进一步的优化,增强算法的适应性.

参考文献(References):

[1] NAKAMOTO S. Bitcoin: a peer-to-peer electronic cash system[J]. Consulted, 2008,39(1):53－67.

[2] JESSE Y H, DEOKYOON K, SUJIN C, et al. Where is current research on blockchain technology? —a systematic review[J]. PLOS ONE, 2016, 11(10).

[3] 邵奇峰,金澈清,张召,等.区块链技术:架构及进展[J]. 计算机学报,2018,41(5):3－22.
SHAO Qifeng, JIN Cheqing, ZHANG Zhao, et al. Blockchain: architecture and research progress [J]. Chinese Journal of Computers, 2018, 41(5):3－22.(in Chinese)

[4] 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 Yong, WANG Feiyue. Blockchain: the state of the art and future trends[J]. Acta Automatica Sinica, 2016,42(4): 481－494.(in Chinese)

[5] GERVAIS A, KARAME G O, KARL W, et al. On the security and performance of proof of work blockchains [C]//ACM SIGSAC Conference, 2016:3－16.

[6] LARIMER D. Delegated proof-of-stake white paper[J]. [2017-07-19].http://8btc.com/doc-view-151. html,2014.

[7] ONGARO D, OUSTERHOUT J. In search of an understandable consensus algorithm [C]//USENIX Annual Technical Conference, 2014: 305－319.

[8] CASTRO M, LISKOV B. Practical byzantine fault tolerance [C]// Symposium on Operating Systems Design & Implementation,1999: 33－36.

[9] CRAIN T, GRAMOLI V, LARREA M, et al. DBFT: efficient leaderless byzantine consensus and its application to blockchains[C]// IEEE 17th International Symposium on Network Computing and Applications, 2018.

[10] KOTL A R. Zyzzyva: speculative byzantine fault tolerance[C]// ACM Sigops Symposium on Operating Systems Principles, 2007:16－22.

[11] COWLING J, MYERS D, LISKOV B, et al. HQ replication: a hybrid quorum protocol for byzantine fault tolerance [C]// USENIX Symposium on Operating Systems Design & Implementation, 2006:13.

[12] 徐治理, 封化民, 刘飚. 一种基于信用的改进 PBFT 高效共识机制[J]. 计算机应用研究, 2019(9):2788－2791.
XU Zhili, FENG Huamin, LIU Biao. Study of highly efficient PBFT consensus mechanism based on credit [J]. Application Research of Computers, 2019(9): 2788－2791.(in Chinese)