计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.200500112

基于信任度匹配的改进 PBFT 共识算法

季钰翔[1]　黄建华[1]　王　喆[1]　郑　红[1]　唐瑞琮[2]
1 华东理工大学信息科学与工程学院　上海 200237
2 香港 DAEX 区块链有限公司　上海 200120
(lygjyx@sina.com)

摘　要　共识算法是去中心化的区块链系统实现数据状态一致的关键。针对传统的实用拜占庭容错(Practical Byzantine Fault Tolerance,PBFT)共识算法在可扩展性和安全性方面存在的不足,提出一种基于信任度的匹配拜占庭共识算法(Trust-based Matching Byzantine Fault Tolerance,TMBFT)。首先,通过基于信任度的邻居匹配模型来选取部分节点进行投票共识,以降低区块链网络的通信量;其次,引入信任度评价机制来监督邻居节点的行为,确保有效检测出拜占庭节点,保证节点投票的安全性;最后,设计投票计数机制保证了共识结果的一致性,并提高了共识效率。与 PBFT 相比,TMBFT 将通信复杂度从 $O(N^2)$ 降到 $O(N\log_2 N)$,有效降低了网络中的通信开销。安全性分析表明,信任度评价机制可降低节点作恶的概率,并有效提高系统安全性。实验结果表明,TMBFT 较传统拜占庭算法具有更好的性能优势。

关键词:区块链;共识算法;拜占庭容错;信任度;邻居匹配;投票计数
中图法分类号　TP393

Improved PBFT Consensus Algorithm Based on Trust Matching

JI Yu-xiang[1],HUANG Jian-hua[1],WANG Zhe[1],ZHENG Hong[1] and TANG Rui-cong[2]
1 School of Information Science & Engineering,East China University of Science & Technology,Shanghai 200237,China
2 Hong Kong DAEX Blockchain Limited,Shanghai 200120,China

Abstract　Consensus algorithm is the key to realize data consistency in decentralized blockchain systems. Aiming at the scalability and security problems of Practical Byzantine Fault Tolerance (PBFT), a Trust-based Matching Byzantine Fault Tolerance (TMBFT) algorithm is proposed. Firstly, the trust-based neighbor matching model is used to select some nodes for voting consensus, so as to reduce the traffic of the blockchain network. Secondly, a trust evaluation mechanism is introduced to supervise the behavior of neighbor nodes, to ensure the effective detection of Byzantine nodes and the security of node voting. Finally, a vote counting mechanism is designed to ensure the consistency of consensus results and improve the efficiency of consensus. Compared with PBFT, TMBFT reduces the communication complexity from $O(N^2)$ to $O(N\log_2 N)$, and effectively reduces the communication overhead in the network. Security analysis shows that the trust evaluation mechanism reduces the probability of malicious voting and improves the system security effectively. Experimental results show that TMBFT has better performance than the traditional Byzantine algorithm.

Keywords　Blockchain, Consensus algorithm, Byzantine fault tolerance, Trust, Neighbor matching, Vote counting

1　引言

自比特币[1]诞生以来,加密货币蓬勃发展,其创造性的底层区块链技术也随之受到了广泛关注。区块链使用点对点通信、共识算法、密码学等多项技术来解决现有中心化机制存在的数据垄断和安全性问题,被认为是一项具有颠覆传统行业潜力的新兴技术[2],在金融、物联网[3]、医疗保健等领域有着广阔的应用前景。

区块链技术的关键是共识算法,共识的目标是使所有的诚实节点保持一致的区块链视图。目前的共识算法主要分为两大类型。第一种类型是基于证明(Proof-Based)的共识算法,其中较为经典的是比特币系统采用的工作量证明(Proof of Work,PoW)。PoW 的核心思想是根据节点的计算能力分配记账权限和记账奖励。然而,PoW 算法在计算过程中需要消耗大量的资源,且系统吞吐量较低。第二种类型是基于投票(Voting-Based)的共识算法,其要求网络中的节点交换新区块或者交易的验证结果,并基于多数的验证结果做出最终的决定,最典型的是实用拜占庭算法(PBFT)[4]。PBFT 解决

到稿日期:2020-05-22　返修日期:2020-10-05　本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
基金项目:国家自然科学基金(61472139)
This work was supported by the National Natural Science Foundation of China(61472139).
通信作者:黄建华(jhhuang@ecust.edu.cn)

了之前的拜占庭容错算法[5]效率不高的问题，可以容忍小于1/3个无效或恶意节点，具有共识终结性。其问题是节点之间的通信量会随着节点数量的增加而急剧增加，通信复杂度达到 $O(N^2)$，且系统可扩展性较差。为了解决这个问题，一些研究采用概率模型来达成共识。节点在共识过程中只接触少量的节点并依靠节点的投票意见来确定自己的投票意见，这类模型通常被称为选民模型(Voter Models)。一般情况下，选民模型的共识状态主要关注两个度量值，即投票意见全部为0或全部为1。文献[6]通过其他节点意见达成共识，并且通信复杂度较低。然而，算法中假设恶意节点不存在或者很少，这不符合区块链的正常工作环境。

为了解决以上问题，本文提出基于信任度匹配的改进PBFT共识算法(TMBFT)来对投票共识的过程进行优化。共识节点基于邻居匹配模型选择邻居节点，在共识过程中仅与邻居节点进行通信，以降低节点之间的通信量，提高网络的可扩展性。信任度评价机制可以有效地检测出作恶节点，而惩罚机制增加了节点的作恶成本，提高了邻居匹配的安全性。针对投票共识算法中恶意攻击的问题，本文算法通过信任度和邻居匹配机制有效限制了攻击发生的可能性，提高了网络的安全性。

2 相关工作

共识算法与区块链系统的性能、安全性和可扩展性有着很大的关系。PoW作为最早应用的区块链共识算法，要求加入网络中的节点计算难度值，以争夺出块权，从而保障了比特币的安全性和公平性。但PoW在计算过程中要浪费大量的电力。针对这一问题，King等[7]在PPCoin(PPC)中提出了权益证明(Proof of Stake，PoS)算法。在基于PoS的系统中，数字货币拥有的币龄等于硬币持有的数量与持有时间的乘积，货币持有者也会根据货币的年龄获得一定的收入。PoS区块链不完全依赖工作量证明，有效地解决了PoW资源浪费的问题。同时，攻击者需要长时间持有大量数字货币，攻击的成本大大增加。然而，PoS也更容易分叉，暴露出远程攻击和无利害关系攻击[8]的漏洞。如何改进PoS算法是许多研究者关注的焦点。DPoS[9]结合了PoW和PoS的特点，每个矿工可以根据自己的权益投票选出一名代表，参与选举且获得票数最多的节点获得对区块打包的权利。在Algorand[10]中，VRF[11]算法使符合条件的用户参与一个加密抽签过程。用户的帐户余额与成为区块生产者的概率成正比，共识组中的节点使用类BFT算法来确定最后生成的块。在Ouroboros[12]中，所有符合条件的节点都可能成为下一阶段的区块生产者。节点在特定阶段发布加密随机数，然后在验证阶段解密并发布随机数，最后使用VRF从这些节点中随机选取下一阶段的共识节点。

PBFT是一种经典的基于投票的共识算法，其将系统的节点划分为主节点和从节点，在拜占庭节点数量不超过1/3的条件下，通过三阶段共识流程完成投票共识，并采用视图切换协议保证系统的正常运行。PBFT可以保证共识终结性，从而避免交易的延迟确认，但其存在通信复杂度高和可扩展性差的问题。为了提升性能，很多以系统吞吐量和鲁棒性为核心的解决方案被先后提出，如DBFT[13]和Tendermint[14]。这类算法解决了原有的投票算法效率不高的问题，且具有强一致性，不易分叉，但仍然存在通信复杂度高、可扩展性较差等问题。Zyzzyva[15]提出了SBFT(Speculative Byzantine Fault Tolerance)，对PBFT在无拜占庭错误的场景下进行优化，其通信复杂度为 $O(N)$。但当恶意节点数量增加时，Zyzzyva的延迟显著提高。

为了解决PBFT通信复杂度的问题，一些研究通过改进选民模型来提高系统性能。例如，文献[16]提出了一种投票计数器机制，节点通过随机访问其他节点并根据投票计数器完成共识。IOTA中的Coordicide[17]计划使用FPC[18](Fast Probabilistic Consensus)进行区块链优化。FPC讨论的是一个二值问题，即所有共识节点要么选择0，要么选择1。算法中引入一个随机数 a 作为选择意见的阈值，当共识节点接收其他节点的1的意见数占总意见数的百分比大于 a 时，则将自己的意见数置为1，否则置为0。这样，恶意节点虽然可以控制部分节点得到的意见值(即0或者1)，但是无法控制阈值 a，这使得网络中大部分诚实节点更容易达成共识。但是，当持有相反意见的节点数量差距缩小时，共识节点由于随机数的波动性无法快速达成共识，在糟糕的情况下，这可能会导致共识的时延比预期的要长；其次，虽然系统的安全性有保障，但无法保证各个随机连接节点的安全性，因而降低了节点的共识效率。

在PBFT系统容错性和可扩展性方面，Trust-PBFT[19]和T-PBFT[20]尝试将信任度与PBFT算法相结合，通过信任度量的方式奖励诚实节点，惩罚恶意节点，但这类算法并没有解决PBFT的通信复杂度高的问题。MBFT[21]采用分层的网络结构，将网络中的节点分成了低层共识组(Low-level Consensus Group，LCG)和高层共识组(High-level Consensus Group，HCG)，LCG通过共识形成微区块(Mini block)，由高HCG负责整合形成最终的区块。MBFT通过扩展LCG来提高区块链系统的吞吐量，同时使用信任机制鼓励节点诚实投票。但其中没有解决PBFT的通信复杂度高的问题，在节点数量较多的情况下会出现跨组交易的问题，整个系统的吞吐量也会低于理想情况下线性增长的吞吐量。

3 基于信任度匹配的投票共识算法

3.1 基本定义

定义1(周期) 一组共识节点完成若干出块的时间间隔。

共识周期如图1所示，在TMBFT协议中将整个周期(epoch)划分成多个时期(slot)，每个slot产生一个区块。slot中包含多个共识轮(round)，在每一轮内，所有共识节点可以查询其匹配的 K 个节点在当前轮次的意见，当满足一致性条件时，节点将完成投票共识。假设在某一周期内有 N 个共识节点，则协议中 K 的值可以相对较大，即 $K \geqslant \log_2 N$，但仍假设 $K \ll N$。当开始下一个epoch时，系统将重新选择共识节点，并且共识节点之间的信任度将会重置。

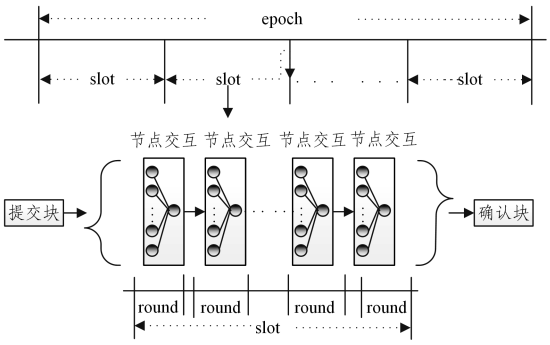

图 1 共识周期

Fig. 1 Consensus epoch

定义 2(邻居节点) 在某一个 slot 内与共识节点进行直接通信的节点。

在 TMBFT 中，节点通信模型如图 2 所示，其中以 A 节点为中心的虚线圈内的节点表示与 A 节点在本周期内匹配过的共识节点，与 A 节点在一个 slot 内成功匹配的节点称为 A 节点在该 slot 内的直接邻居节点，未与 A 节点匹配成功的节点称为间接邻居节点。与传统的协商一致性协议不同的是，TMBFT 中的共识节点使用邻居匹配模型选取部分共识节点作为自己的邻居节点，在共识过程中与匹配的邻居节点进行多轮通信。在每一轮的投票通信中，节点根据邻居节点的投票意见修正自己的意见。在 TMBFT 中，节点每轮通信也会获得其直接邻居节点的邻居节点投票意见集，意见集中的签名保证了各节点投票结果的真实性，这种策略保证了节点可以感知到全网大多数节点的意见。

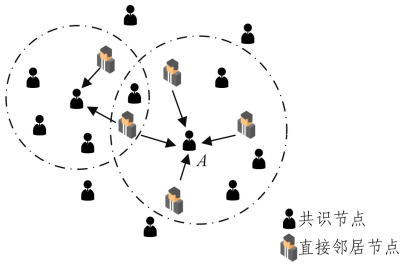

图 2 节点通信图

Fig. 2 Nodes communication diagram

定义 3(投票计数器) 统计邻居节点投票数量的计数器。

简单的选民模型无法在有恶意节点存在的情况下提供强大的安全保障。特别是在二值的共识问题上(非 0 即 1)，如果诚实的节点倾向于一种投票结果，那么恶意节点可以尝试采取相反的投票行为以阻碍共识的形成。基于这个原因，TMBFT 在节点上引入了状态存储，所有的共识节点内置一个投票计数器，节点根据计数器来改变自己的投票意见。节点计数器遵循如下规则：

1)每轮通信后，占比高的投票意见的计数器数值自增 1。

2)当节点计数器的当前投票意见的计数值低于新的投票意见时，节点会支持新的投票意见。

3)一致性条件。当节点计数器达到阈值 p 时，节点最终选择自己当前的投票意见作为投票结果。

定义 4(信任度期望 $E(OT)$) 节点在某个 slot 需要拥有的信任度值。

为了确保邻居节点连接的安全性，防止恶意节点进行恶意投票，每个共识节点在每个 slot 之初都会对其邻居节点的上一轮投票行为进行信任度评价，作为邻居匹配的依据。当恶意节点的信任度低于信任度期望时，共识节点将不与其进行匹配，使恶意节点无法参与下轮共识。

定义 5(主节点) 负责区块的打包和上链。

TMBFT 在每个 slot 采用 follow-the-satoshi(FTS)进行主节点的选择。在每一个 epoch 内都有一个不上链的初始区块，记录了本 epoch 中共识节点的公钥集、共识节点、上一个 epoch 结束时的信任度集合和初始种子 *seed*，其中 *seed* 由 Ouroboros 中的种子生成过程产生。

TMBFT 中对 follow-the-satoshi 算法的实现是将初始块中的 *seed*、共识节点集以及信任度集合作为输入执行 FTS 抽样函数，从共识节点集中随机抽样出主节点。主节点通过计算得知自己的主节点身份之后，会向全网广播一个空区块，其中包含了当前 slot 的编号、时间戳以及共识开始信息。所有共识节点在收到空区块后开始邻居匹配过程。在 TMBFT 的共识过程中，由主节点负责区块的打包和上链，但主节点不参与投票过程，其余共识节点则会进行一致性投票共识，最终确定此区块是否有上链资格。

3.2 系统设置

TMBFT 中有两种角色：普通节点和共识节点。普通节点可以随时加入或退出网络，在使用区块链系统服务的同时，也可以看到整个共识过程。在区块生成过程中，普通节点有提交交易的权力以及参与块转发的义务。共识节点负责整个区块链系统的运行，参与区块的投票以及上链过程；主节点负责打包交易形成区块并分发给各个节点进行共识。

假设全网总节点数为 M，恶意节点数为 f，网络正常运行需要满足 $M \geqslant 3f+1$，在此假设下可保证系统的运行不会因为受到恶意节点的影响而停止。为了成为共识节点，节点需要提交申请并缴纳押金成为候选人，由系统随机选择 N ($N \ll M$)个候选人节点成为共识节点；若共识节点多次作恶则会被淘汰，并扣除押金。为保证系统的一致性和安全性，在共识过程中要求所有诚实节点存储的区块数据和交易信息正确且一致；所有诚实节点都会接受已确认区块。

本文假设恶意节点可以随时查看每个诚实节点的状态，并可以立即改变自己的状态，但恶意节点不能调度或修改诚实节点之间的通信内容。此外，恶意节点可以根据自己的需要在共识过程中发动女巫攻击，但其算力是有限的。本文采用文献[22]中的网络模型，当共识节点的内部定时器超时或者接收到足够的消息时，将数据发送给所选的邻居节点。节点接收信息后更新其状态并将状态信息返回给发送节点。假设网络通信是部分同步的，即节点之间的信息通信存在延迟，但任何一组诚实节点之间的通信延迟不会超过一个本地轮(round)。同时共识节点在邻居匹配过程之后不会出现宕机、掉线等情况，能够正常完成投票过程。

3.3 TMBFT 一致性协议

在 TMBFT 中，每一个 slot 都会有一定数量的交易或账户状态变化被打包记录进区块中，共识节点通过一致性协议保证区块信息的正确性和一致性。在这个 slot 内的主节点将负责对接收到的交易进行验证并将完成验证的交易打包进区块。参与共识的节点会获得从 0 到 $N-1$ 的编号，主节点由前面介绍的算法选出。

图 3(a)为 PBFT 协议的共识过程，图 3(b)为 TMBFT 协议的共识过程，TMBFT 相比 PBFT 增加了一个 response 阶段，共分为如下 5 个阶段。

1)request 阶段：客户端将交易信息发送给主节点。

2)pre-prepare 阶段：主节点对正确的交易请求进行排序，分配编号 n 并生成 pre-prepare 消息，其中包括新区块、消息时间戳和主节点签名等。主节点将 pre-prepare 消息发送给共识节点，共识节点收到 pre-prepare 消息之后进入 prepare 阶段。

3)prepare 阶段：节点校验消息，包括区块内的交易正确性、区块头信息正确性、区块高度和签名等。之后节点向其邻居节点发送验证信息，并等待其他邻居节点的验证信息，共识节点进入 commit 阶段。

4)commit 阶段：当节点收到来自邻居节点的验证信息之后，启用计数器统计投票结果。节点重复步骤 3)直到达到一致性条件，进入 response 阶段。

5)response 阶段：共识节点将自己的投票结果反馈给主节点。

主节点在收到足够的反馈消息后将验证结果发送给客户端，反馈信息包含所有共识节点的签名。

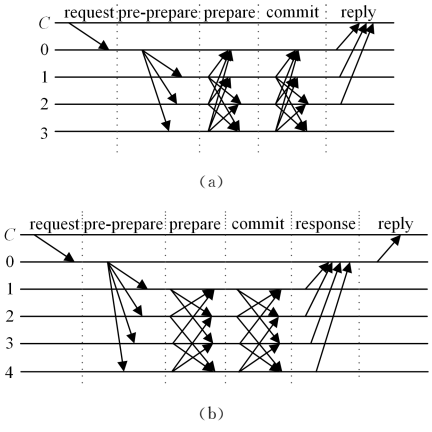

图 3 PBFT 与 TMBFT 协议的对比

Fig. 3 Comparison of PBFT and TMBFT

图 3(b)展示了 TMBFT 在投票计数器阈值 $p=2$ 时的多阶段协议过程。其中节点 1 和节点 2、节点 3 互为邻居节点，节点 4 和节点 2、节点 3 互为邻居节点。共识节点在共识过程中基于邻居节点通信，减少了节点通信的数量，降低了通信开销。

3.4 节点信任度模型

为了鼓励节点诚实投票，保证邻居节点之间的通信安全，在进行邻居匹配之前，共识节点需根据上一轮邻居节点的投票行为计算出节点的综合信任度。本文使用文献[23]的数据结构来存取评价数据。在新一轮匹配过程开始时，节点随机选择邻居表中的节点发送匹配请求。邻居节点在收到匹配请求消息后，根据计算出的节点综合信任度，并依据邻居匹配模型进行匹配。匹配成功后，节点向请求节点返回匹配成功信息。

定义 6(直接信任度 $DT_{i,j}$) 节点 i 针对直接邻居节点 j 的投票行为计算出的信任度称为直接信任度，计算公式如下：

$$DT_{i,j}=\frac{1-\mathrm{e}^{-\alpha\left(\sum_{x=1}^{n-1}t_{m,j}-\sum_{x=1}^{n-1}f_{m,j}\right)}}{1+\mathrm{e}^{-\alpha\left(\sum_{x=1}^{n-1}t_{m,j}-\sum_{x=1}^{n-1}f_{m,j}\right)}} \tag{1}$$

其中，n 表示节点达成共识所需的轮数；$t_{m,j}$ 表示节点 j 在当前 slot 内的第 m 轮是否诚实参与投票，诚实投票记为 1，反之为 0；$f_{m,j}$ 表示节点 j 在当前 slot 内的第 m 轮是否为恶意投票，恶意投票记为 1，反之为 0。共识节点在收到邻居节点的投票意见时，也会收到其邻居节点集的上一轮投票意见，若节点作恶被发现，则会被记为恶意投票。例如，节点 j 同时为节点 i 和 k 的公共邻居节点，当节点 j 发送给节点 i 的投票结果与其发送给节点 k 的投票结果相同时，记为诚实投票，否则记为恶意投票。参数 α 为信任度成长系数，当 epoch 周期过半时，系统将调整 α 值来控制信任度增长速度，实验表明这种机制有效降低了节点作恶的可能性。

在每个 epoch 初期，节点信任度较低，通过诚实投票可以使信任度加速增长。如果节点恶意投票，节点的信任度将快速下降；在每个 epoch 后期，$DT_{i,j}$ 随着 slot 数的增加逐渐收敛为 1，节点信任度的增长速度变缓。当全网诚实节点都达到高信任区时，由于信任度差距缩小，每个节点的匹配概率大致相同，此时节点的信任度匹配接近随机匹配，保证了整个系统的安全。

定义 7(间接信任度 $IT_{i,j}$) 除了评价主体节点 i 外，对被评估节点 j 进行过评价的节点 k 对节点 j 的直接信任度的加权和为被评价节点的间接信任度，计算公式如下：

$$IT_{i,j}=\frac{\sum_{k=1,k\neq i}^{n}|D_{k,j}|\times DT_{k,j}}{R} \tag{2}$$

其中，R 是除评价主体节点的评价外，被评价节点获得的直接评价总数；$|D_{k,j}|$ 是单个节点给出直接评价的次数，$|D_{k,j}|=1$ 表示节点 k 直接评价节点 j，$|D_{k,j}|=0$ 表示节点 k 没有直接评价节点 j。当节点与被评价节点匹配次数越多，给出的直接评价在计算间接信任度时的权重就会越高。

为了防止恶意节点给予不真实的评价，间接信任度需要有相应的指标对其进行控制。假设节点 j 的评价来源集合为 $R=\{n_1,n_2,\cdots,n_k\}$，节点 j 的评价次数集合为 $S=\{|D_{n1,j}|,|D_{n2,j}|,\cdots,|D_{nk,j}|\}$，直接信任度集合为 $DT=\{DT_{n1,j},DT_{n2,j},\cdots,DT_{nk,j}\}$。节点的直接信任度评价的离散程度影响间接信任度的可靠程度。评价节点集合中的节点对节点 j 的直接信任度评价的离散程度越大，越表明这些节点在评价节点 j 的投票行为上存在分歧，其间接信任度评价的可靠性越低，此时间接信任度的权重越小。间接信任度的权重可以表示为：

$$\mu=\frac{1}{\sigma+1} \tag{3}$$

其中，σ 是被评价节点信任度集合的标准差。

$$\sigma=\sqrt{\frac{\sum_{k=1,k\neq i}^{n}(DT_{k,j}-IT_{i,j})^2}{n}} \tag{4}$$

若各个评价节点对被评价节点的投票行为看法较为一致，表明节点计算出的间接信任度可靠性高。

定义 8（节点综合信任度 $OT_{i,j}$） 直接信任度与间接信任度的加权代数和，在匹配过程中根据综合信任度选择匹配节点。其计算公式如下：

$$OT_{i,j}^{(t)}=\begin{cases}(\beta\times T_{i,j}^{(t)})+((1-\beta)\times OT_{i,j}^{(t-1)}), & t>1\\ 0.5, & t=1\end{cases} \tag{5}$$

其中，t 为当前 epoch 的第 t 个 slot；$T_{i,j}^{(t)}$ 表示本轮计算出的直接信任度或间接信任度；$OT_{i,j}^{(t-1)}$ 表示在第 $t-1$ 个 slot 计算出的综合信任度；$\beta\in[0,1]$为信任度的权重，避免作恶节点在信任度惩罚之后快速获得信任度。当被评价节点 j 是节点 i 的间接邻居时，$T_{i,j}^{(t)}=IT_{i,j}$，$\beta=\mu$；当被评价节点 j 是节点 i 的直接邻居时，$T_{i,j}^{(t)}=DT_{i,j}$，β 为系统规定的系数。在每个 epoch 之初，系统期望共识节点在未来表现良好，并将每个共识节点综合信任度初始化为 0.5。

3.5 邻居匹配模型

为了防止拜占庭节点进行恶意投票，本文提出了一种基于信任度的邻居匹配抽签算法。假设每个共识节点需要匹配 K 个邻居。令 $\boldsymbol{\Omega}$ 为 $N\times K$ 分配矩阵，其中元素 $u_{i,j}=\{0,1\}$，$\forall i,j\in N$。$u_{i,j}=1$ 表示共识节点 i 与共识节点 j 成为邻居节点，$u_{i,j}=0$ 表示节点 i 与节点 j 之间没有匹配。因此，有如下约束：

$$\sum_{j=1}^{N}u_{i,j}\leqslant K_i \tag{6}$$

为了保证匹配和投票共识的效率，每个共识节点在共识过程中要维护一张邻居表和一张低信任度表。邻居表中存放备选的邻居节点及其信任度，低信任度表中存放的是低于信任度期望的节点。信任度期望 $E(OT)$ 与 t 有关，当 epoch 过半时，节点会假设请求节点为诚实节点，并计算出此节点在（$t-$epoch/2）时的综合信任度作为对请求节点的信任度期望。计算公式如下：

$$E(OT_{i,j}^{(t)})=\begin{cases}0, & t\leqslant\dfrac{epoch}{2}\\ OT_{i,j}^{(t-\frac{epoch}{2})}, & t>\dfrac{epoch}{2}\end{cases} \tag{7}$$

共识节点的匹配过程分为 4 个阶段，其流程如图 4 所示。

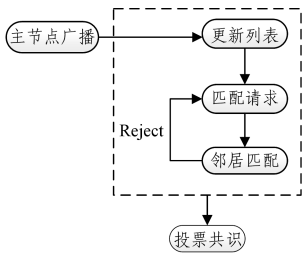

图 4 匹配流程

Fig. 4 Matching process

阶段 1 主节点广播预先生成的空区块，所有共识节点接收到空区块后更新自己的邻居列表，并将较低信任度的共识节点放入低信任度表。

阶段 2 共识节点随机选取其邻居表中的 $K-S$（S 为匹配成功节点，初始为 0）个节点发送匹配请求信息，并等待其他节点发送匹配请求。

阶段 3 节点在接收到其他共识节点的匹配请求后，根据算法 1 进行邻居选择，并向其发送匹配结果信息。

阶段 4 若共识节点收到少于 K 个匹配成功信息，节点将继续执行阶段 2 和阶段 3。当网络中的共识节点无法再进行匹配时开始共识。

节点收到匹配请求之后，首先拒绝在低信任度表中的节点请求，之后对于其他节点将采用匹配算法进行邻居选择。为了能使用户被选中的概率和其所拥有的信任度相对应，本文在匹配过程中借鉴了 Algorand 抽签算法和 VRF 算法的思想，并设计了使用基于信任度的匹配方式。如算法 1 所示，共识节点将邻居表中的节点信任度按单位信任度分割成若干份，并基于信任度份额进行选择。

算法 1 SelectByTrust

输入：(sk，seed，OT，blockNum，pkl，w，W)

$\langle \text{Hash},\pi\rangle\leftarrow \text{VRF}_{\text{sk}}(\text{seed}\parallel\text{blockNum}\parallel\text{pkl})$

$\text{p}\leftarrow\dfrac{\text{OT}}{\text{W}}$

$\text{i}\leftarrow 0$

while $\dfrac{\text{hash}}{2^{\text{hashlen}}}\notin[\sum_{\text{k}=0}^{\text{i}}\text{B}(\text{k};\text{w},\text{p}),\sum_{\text{k}=0}^{\text{i}+1}\text{B}(\text{k};\text{w},\text{p})]$ do

$\text{i}++$

输出：$\langle \text{hash},\pi_l,\text{i}\rangle$

其中，sk 为当前节点的私钥，$seed$ 是公共的种子，OT 为请求节点的信任度评价，$blockNum$ 是当前区块高度，pkl 为请求节点的公钥，w 是请求节点的信任度份额，W 是当前节点的邻居列表中所有的信任度份额之和。由算法 1 可以看出，当请求节点的信任度越高时，计算出的 i 越大，详细证明过程请参照 Algorand。该节点对所计算出的 i 按大小进行排序，选择排名靠前的 K 个节点作为自己的邻居节点，并向其发送匹配成功信息，信息中包含 π_l，可用于节点验证匹配信息的正确性。

3.6 视图更换协议

主节点完成区块的上链之后，共识节点将打包进区块的交易从待确认列表中移除，视图编号增加 1，进入到下一个 slot。但当主节点产生错误时会导致区块链系统的阻塞。视图更换协议用于主节点故障时变更主节点，以保证区块链系统正常运行。

当主节点没有完成区块生成时，由其余共识节点触发视图更换协议，具体执行过程如下：

1）新的主节点由公式 $p=v \bmod |N|$ 计算得出。其中 v 为视图的编号，$|N|$ 是共识节点的数量。新的主节点执行视图更换协议后进入视图 $v+1$，发送视图更换消息给所有节点，其中包括新的视图编号、新区块编号、摘要和签名。

2）若节点收到 $2f+1$ 个有效视图更换消息，则将确认消

息发送给视图 $v+1$ 的主节点，并等待主节点发起共识。

3)主节点生成新区块，开始新一轮的共识过程。

3.7 协议复杂度分析

本文提出的 TMBFT 共识算法通过选择一定数量的邻居节点进行投票共识，从而降低通信复杂度。若全网的共识节点数为 N，PBFT 完成一轮共识过程所需要的通信复杂度为 $O(N^2)$。在 TMBFT 算法中，节点每次通信都会获得直接邻居的邻居节点意见集，这保证了节点可以感知到全网大多数节点的意见。若共识节点选取固定 $K(K\geqslant\log_2 N)$ 个邻居进行共识，投票计数器的阈值为 $p(p\geqslant 1)$，由图 3(b)可知，完成一次共识所需的通信次数为 $Msg=1+(N-1)+N\cdot p\cdot\log_2 N+N+1=1+2N+pN\log_2 N$，即节点的通信复杂度为 $O(N\log_2 N)$。与 PBFT 相比，TMBFT 降低了通信复杂度，提高了系统的节点扩展性。

4 安全性分析

4.1 女巫攻击

在女巫(Sybil)攻击的场景中，大量的虚假节点通常是由一个恶意节点生成的。中本共识使用工作量证明来限制女巫攻击；PBFT 等共识算法则将女巫问题与共识分开处理。TMBFT 协议采用押金机制，普通节点成为共识节点需要缴纳押金，作恶则会有扣除押金和降低信任度的惩罚。节点信任度很低时，会被扣除押金且不允许参与共识，理性的节点通常会采取诚实行为。

4.2 静态攻击

静态攻击是指恶意节点始终向其余诚实节点提供虚假投票结果。在共识过程中，可以选择多轮恶意投票。若投票计数器的阈值 $p=5$，$\alpha=0.4$，由式(1)可知，恶意节点选择在一轮通信中进行恶意投票的直接信任度变化表示如下：

$$RT=\frac{DT1}{DT2}=\left(\frac{-e^{-0.4(3-1)}}{1+e^{-0.4(3-1)}}\right)\Big/\left(\frac{1-e^{-0.4(4-0)}}{1+e^{-0.4(4-0)}}\right)=0.57$$

通过计算可知，节点作恶一轮时，获得的直接信任度约为诚实投票时的一半，这对于节点的信任度成长是非常不利的。作恶节点在共识周期内会因为其信任度低于信任度期望而无法参与共识过程。

4.3 动态攻击

动态攻击是指恶意节点交替执行诚实投票行为和恶意投票行为，且恶意节点始终保持自身信任度在信任阈值之上。但这种攻击方式在本文的机制中是不可行的，这是因为邻居匹配的概率与其信任度有关。对于动态恶意节点而言，其恶意行为执行的次数越多，其信任度的波动就越大。当 epoch 周期过半时，诚实节点都会处于高信任度区域，少数作恶节点的信任度会低于诚实节点。随着 slot 的增加，对于节点的信任度期望会越来越高；而随着 α 的降低，作恶节点恢复信任度的速度放缓；当作恶节点的信任度低于信任度期望时该节点就会被淘汰。

4.4 共谋攻击

共谋攻击是指多个恶意节点给出高于或低于实际水平的信任度评价，其使得被评价节点的信任度与实际信任度产生偏差。根据攻击效果，可将共谋攻击分为贬低攻击和夸大攻击。

多个恶意节点合作对诚实节点实施贬低攻击，使诚实节点获得的信任度低于正常的信任度评价。由式(4)和式(5)可知，当共识节点收到被攻击节点的评价存在较大分歧时，节点将降低 μ 值使 β 减小，以降低贬低攻击对信任度计算造成的影响。

多个恶意节点合作对某个节点进行夸大攻击，使该节点获得的信任度高于正常的信任度评价，其目的是提高该节点的综合信任度。诚实节点在评价过程中会给予被评价节点较多负面评价，而进行夸大攻击的节点会给予虚假的正面评价，此时被评价节点的评价分歧较大。由式(4)和式(5)可知，这将导致 μ 值降低，从而 β 也会变小，因此降低了夸大攻击对信任度计算造成的影响。

5 实验分析

实验使用 Docker 技术模拟搭建多节点的区块链环境，Docker 的运行环境为 1 台 DELL R320 服务器，配置为 Intel I7-4702MQ CPU 2.20GHz 和 16GB 内存，操作系统为 Ubuntu server 14.04，软件版本为 Docker version 18.03.0-ce, build 0520e24。本文通过搭建原型验证模型，验证 TMBFT 算法的性能，并对节点信任度的增长与惩罚进行实验分析。

表 1 仿真参数

Table 1 Simulation parameters

名称	默认值
共识节点数 N	64
恶意节点比例/%	10
信任度系数 α	0.5
信任度权重 β	0.5
阈值 p	4

5.1 节点信任度增长

实验中以形成 20 个区块为一个周期，网络中共有 64 个共识节点。实验测试了 β 取不同值的情况下，系统中节点综合信任度的变化趋势。

如图 5 所示，随着 β 的增加，网络中节点的信任度增长速度加快。经过多个 slot 共识，网络中诚实节点的信任度会维持在一个恒定的高信任区域内，因此不会出现单个节点信任度过高而导致的中心化问题，同时由于此时诚实节点的信任区间大致相同，因此节点之间的邻居匹配近似于随机匹配。

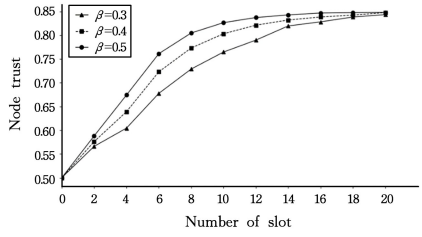

图 5 节点信任度增长

Fig. 5 Node trust growth

5.2 作恶节点信任度惩罚

(1)静态攻击

本实验测试了节点在发动静态攻击后的信任度惩罚情

况。静态攻击节点会选择多轮恶意投票以延长系统中节点达到一致性的时间。实验中采取的静态攻击策略是，恶意节点为了获得网络中的节点信息，在第一个 slot 正常投票，从第二个 slot 开始恶意投票。在每个 slot 内，恶意节点选择进行 1 个轮次($R=1$)、2 个轮次($R=2$)以及 3 个轮次($R=3$)的恶意投票，其信任度变化如图 6 所示，实验结果为 8 组数据的平均值。

由图 6 可知，由于采取的策略不同，节点信任度惩罚幅度也不相同。当节点在共识过程中选择 3 轮进行恶意投票时，其信任度出现显著下降，3 个 slot 后共识信任度就会降为 0，网络中的节点不会匹配信任度为 0 的节点，这表明 TMBFT 有效地限制了这种静态攻击方式；当节点在共识过程中选择 2 轮进行恶意投票时，其信任度下降较快，同样最后下降为 0。当节点在共识过程中选择 1 轮进行恶意投票时，信任度下降幅度较小，短时间内不会被淘汰。当进入 epoch 中后期时，该节点由于信任度期望过低而被淘汰，从而保证了网络安全。

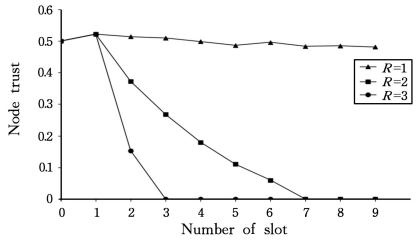

图 6 静态攻击节点的信任度惩罚

Fig. 6 Trust penalty for static attack nodes

(2)动态攻击

通过仿真恶意节点的动态攻击行为来分析恶意节点的综合信任度变化。在本实验中动态攻击的表现为，节点开始时诚实投票，之后发动动态攻击进行恶意投票；当节点的信任度较低时，节点开始诚实投票，直到恢复高信任度，重复之前的行为。

如图 7 所示，实验参数 $\alpha=0.5$，节点在高信任度区时发动动态攻击，导致其信任度下降速度非常快。节点在第 10 个 slot 共识后再次进行诚实投票，此时 α 会调整为 0.3，信任度期望开始上调，节点的信任度增长变缓，根据信任度期望规则在第 14 次共识之后此动态攻击节点将被淘汰。

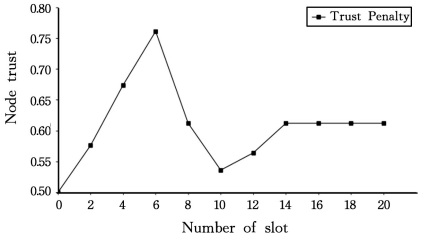

图 7 动态攻击节点的信任度惩罚

Fig. 7 Trust penalty for dynamic attack nodes

5.3 匹配时延

本实验测试了在有无恶意节点这两种情况下，匹配时延随着 slot 数增加的变化情况。如图 8 所示，随着 slot 数的增加，两种情况下的匹配时延都有所增加，原因是在 epoch 的前中期节点信任度变化幅度大，匹配过程相对复杂；当 epoch 进入中后期时，网络中的恶意节点将被淘汰；在后半个 epoch 内，匹配时延最终趋于稳定，这是因为诚实节点都处于高信任区，邻居匹配近似于随机匹配。

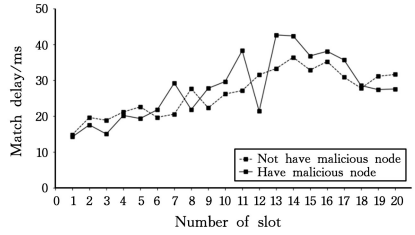

图 8 匹配时延

Fig. 8 Delay of matching

5.4 共识时延

本实验在相同环境下分别运行 PBFT，FPC，MBFT 和 TMBFT 协议。如图 9 所示，PBFT 协议的共识时延近似二次方增长，这是因为节点之间必须经过大量通信才能保证投票的一致性，通信复杂度为 $O(N^2)$。FPC 在通信上相较于 PBFT 有了极大的改进，通信复杂度为 $O(N\log_2 N)$。但其缺点是共识随机性较大，依赖网络中节点意见的统一性。例如，当实验中持有“1”意见的节点占全部节点的 90%以上时，网络中的共识节点会很快以“1”意见达成共识；当“1”意见占比不足 70%时，网络中的节点需要更多轮次才能完成共识，这导致其在共识时延方面显得不够稳定；同时频繁的随机邻居查询操作会增加节点的通信成本，从而降低共识效率。在本实验中 MBFT 采用了 2-LCG 的参数进行实验，且无跨组交易。TMBFT 的共识时延明显小于 PBFT，与 FPC 和 MBFT 的共识时延相比，提升了约 20%，且随着节点数量的增加，共识时延没有发生特别大的变化，这说明 TMBFT 具有良好的可扩展性。

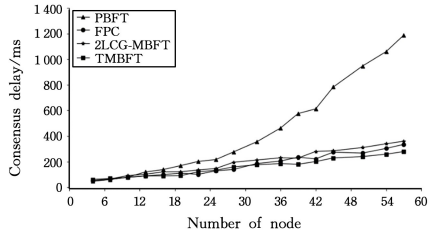

图 9 多节点共识时延

Fig. 9 Consensus delay of multi-node

结束语 针对传统 PBFT 协议通信开销大、节点扩展性较差等问题，本文提出了一种高效的共识算法 TMBFT。TMBFT 使用基于信任度的邻居匹配模型来选择网络中的部分节点成为自己的邻居节点，在投票共识过程中仅与邻居节点进行通信。这样的设计控制了节点交互的数量，从而减少了通信开销，提高了节点的可扩展性。每个共识节点使用投票计数器记录邻居节点的投票结果，保证了共识一致性。节点根据每次共识时邻居节点的投票行为进行信任度评价，有利于系统稳定运行。安全性分析表明，信任度惩罚机制和信任度期望机制有效地限制了针对投票共识过程的恶意攻击，

保证了系统安全。实验结果表明，与 PBFT 和 FPC 等算法相比，TMBFT 能保证共识节点更稳定有效地达成一致性，以提高共识效率。

参 考 文 献

[1] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash System[EB/OL]. http://bitcoin.org/bitcoin.pdf.

[2] CHEN W L, ZHENG Z B. Blockchain Data Analysis: A Review of Status, Trends and Challenges[J]. Journal of Computer Research and Development, 2018, 55(9): 1853-1870.

[3] NOVO O. Blockchain Meets IoT: An Architecture for Scalable Access Management in IoT[J]. IEEE Internet of Things Journal, 2018, 5(2): 1184-1195.

[4] CASTRO M, LISKOV B. Practical Byzantine Fault Tolerance [C]// Proceedings of the Third Symposium on Operating Systems Design and Implementation (OSDI). New Orleans, USA, 1999.

[5] FAN J, YI L T, SHU J W. Research on the Technologies of Byzantine System[J]. Journal of Software, 2013, 24(6): 1346-1360.

[6] GIULIA F, NINA H, YUVAL P, et al. Communication Cost of Consensus for Nodes with Limited Memory[EB/OL]. https://arxiv.org/pdf/1901.01665.pdf.

[7] KING S, NADAL S. PPCoin: Peer-to-Peer Crypto-Currency with Proof-of-Stake[EB/OL]. https://www.semanticscholar.org/paper/PPCoin%3A-Peer-to-Peer-Crypto-Currency-with-King-Nadal/0db38d32069f3341d34c35085dc009a85ba13c13.

[8] DEIRMENTZOGLOU E, PAPAKYRIAKOPOULOS G, PATSAKIS C. A Survey on Long-Range Attacks for Proof of Stake Protocols[J]. IEEE Access, 2019, 7: 28712-28725.

[9] LUO Y, CHEN Y, CHEN Q, et al. A New Election Algorithm for DPos Consensus Mechanism in Blockchain[C]// 2018 the 7th International Conference on DigitalHome (ICDH). IEEE, 2018: 116-120.

[10] GILAD Y, HEMO R, MICALI S, et al. Algorand: Scaling Byzantine Agreements for Cryptocurrencies[C]// Proceedings of the 26th Symposium on Operating Systems Principles. New York: ACM, 2017: 51-68.

[11] MICALI S, RABIN M, VADHAN S. Verifiable Random Functions[C]// 40th Annual Symposium on Foundations of Computer Science. New York: IEEE, 1999.

[12] KIAYIAS A, RUSSELL A, DAVID B, et al. Ouroboros: A Provably Secure Proof-of-Stake Blockchain Protocol[C]// Annual International Cryptology Conference. Springer, Cham, 2017: 357-388.

[13] CRAIN T, GRAMOLI V, LARREA M, et al. DBFT: Efficient Leaderless Byzantine Consensus and its Application to Blockchains[C]// 2018 IEEE 17th International Symposium on Network Computing and Applications (NCA). IEEE, 2018.

[14] KWON J. Tendermint: Consensus without Mining [EB/OL]. https://tendermint.com/static/docs/tendermint.pdf.

[15] KOTLA R. Zyzzyva: Speculative Byzantine Fault Tolerance [C]// ACM Sigops Symposium on Operating Systems Principles. ACM, 2007: 45-48.

[16] ROCKET T, YIN M, SEKNIQI K, et al. Scalable and Probabilistic Leaderless BFT Consensus through Metastability [EB/OL]. https://arxiv.org/pdf/1906.08936.pdf.

[17] COORDICIDE T, IOTA F. The Coordicide [EB/OL]. https://files.iota.org/papers/Coordicide_WP.pdf.

[18] POPOV S, BUCHANAN W J. FPC-BI: Fast Probabilistic Consensus within Byzantine Infrastructures[EB/OL]. https://arxiv.org/pdf/1905.10895.pdf.

[19] TONG W, DONG X, ZHENG J. Trust-PBFT: A PeerTrust-Based Practical Byzantine Consensus Algorithm[C]// 2019 International Conference on Networking and Network Applications (NaNA). Daegu: IEEE, 2019.

[20] GAO S. T-PBFT: An EigenTrust-Based Practical Byzantine Fault Tolerance Consensus Algorithm[J]. China Communications, 2019, 16(12): 111-123.

[21] DU M, CHEN Q, MA X. MBFT: A New Consensus Algorithm for Consortium Blockchain[J]. IEEE Access, 2020, 8: 87665-87675.

[22] OROSTICA B, NUNEZ F. Robust Gossiping for Distributed Average Consensus in IoT Environments[J]. IEEE Access, 2019, 7: 994-1005.

[23] WU D Y, LI Q, YU X, et al. Trust Model for P2P Based on Blockchain[J]. Computer Science, 2019, 46(12): 138-147.

JI Yu-xiang, born in 1996, postgraduate. His main research interests include blockchain and so on.

HUANG Jian-hua, born in 1963, Ph.D, professor, is a member of China Computer Federation. His main research interests include computer network, information security and blockchain.