中继链环境中一种优化的跨链异步共识机制

张佩云[1),2)] 徐涪雅[1),3)] 陈子寒[1),2)]

[1)](南京信息工程大学数字取证教育部工程研究中心 南京 210044)

[2)](南京信息工程大学计算机学院、网络空间安全学院 南京 210044)

[3)](南京信息工程大学软件学院 南京 210044)

摘 要 中继链是一种旨在实现不同区块链之间互操作性的跨链技术。然而,同步和部分同步共识机制在中继链中的运行效率受到活跃性、网络延迟等问题的限制,难以满足中继链对高效和可靠的跨链操作的需求。异步共识机制的鲁棒性强,能够应对较差的网络环境并支持更多的应用链接入中继链。目前,异步共识机制由于存在大量重复交易、恶意节点攻击的问题而导致效率低下。因此,本文提出中继链环境中一种优化的跨链异步共识机制(Optimized Cross-chain Asynchronous Consensus Mechanism, OCAC),包括交易广播模块与多值拜占庭共识模块。首先,设计基于最优跨链交易匹配的交易广播模块,通过改变异步共识节点从各自交易池中随机选取批量交易作为输入的方式,综合考虑异步共识节点在交易广播模块的异构处理能力与共识成功率,实现最优跨链交易匹配,以避免交易重复广播并提高交易广播效率;其次,设计基于提议向量选择优化的多值拜占庭共识模块。该模块依据异步共识节点的不同性能参数的变化,迭代出性能高且稳定的节点集合,并从该集合中随机选择一个异步共识节点,对其提议向量进行异步二元共识,以减少需要运行的异步二元共识实例个数,加速多值拜占庭共识模块的运行,从而提高异步共识效率。实验结果表明本文所提出的OCAC在中继链环境中与典型的异步共识机制Dumbo、Chronos、TortoiseBFT、sDumbo、Dumbo-NG相比,具有较高的交易吞吐量与较低的交易时延。

关键词 区块链;跨链异步共识;最优跨链交易匹配;提议向量选择优化;多值拜占庭共识

中图法分类号 TP18 **DOI号** 10.11897/SP.J.1016.2026.00284

An Optimized Cross-Chain Asynchronous Consensus Mechanism in the Relay Chain Environment

ZHANG Pei-Yun[1),2)] XU Fu-Ya[1),3)] CHEN Zi-Han[1),2)]

[1)](Engineering Research Center of Digital Forensics of Ministry of Education, Nanjing University of Information Science & Technology, Nanjing 210044)

[2)](School of Computer Science, School of Cyberspace Security, Nanjing University of Information Science & Technology, Nanjing 210044)

[3)](School of Software, Nanjing University of Information Science & Technology, Nanjing 210044)

Abstract The relay chain is a cross-chain technology designed to achieve interoperability between heterogeneous blockchain systems. However, the operational efficiency of synchronous and partially synchronous consensus mechanisms in relay chain environments is constrained by challenges including liveness issues and network latency, making them inadequate to meet the stringent requirements for high-efficiency and reliable cross-chain operations. While asynchronous consensus mechanisms demonstrate superior robustness, enabling reliable operation in adverse network conditions and supporting connectivity with more application chains, their current implementations suffer from efficiency limitations due to redundant transaction propagation and

收稿日期:2024-12-20;在线发布日期:2025-08-22。本课题得到江苏省自然科学基金(No. BK20251888)和国家自然科学基金(No. 61872006)资助。**张佩云**,博士,教授,博士生导师,中国计算机学会(CCF)高级会员,主要研究领域为区块链、智能信息处理、云计算、可信计算。E-mail: zpy@nuist.edu.cn。**徐涪雅**,硕士,主要研究领域为区块链。**陈子寒**,硕士研究生,主要研究领域为区块链。

vulnerability to malicious node attacks. To address these challenges, this paper proposes an optimized cross-chain asynchronous consensus mechanism (OCAC) for relay chain environments, comprising the transaction broadcasting module and the multi-valued Byzantine consensus module. First, OCAC designs the transaction broadcasting module based on optimal cross-chain transaction matching. By replacing the conventional approach where asynchronous consensus nodes randomly select transactions from their buffers, our solution holistically considers the heterogeneous processing capabilities and consensus success rates of nodes during transaction propagation. This optimization achieves optimal cross-chain transaction matching to eliminate redundant broadcasting and enhance propagation efficiency. Second, OCAC designs a multi-valued Byzantine consensus module featuring optimized proposal vector selection. This module iteratively identifies high-performance node clusters with stable operational characteristics based on dynamic performance parameter variations. By randomly selecting nodes from this optimized cluster for asynchronous binary consensus on proposal vectors, OCAC effectively reduces the required number of asynchronous binary consensus instances, thereby accelerating the multi-valued Byzantine consensus process and improving asynchronous consensus efficiency. Experimental evaluations demonstrate that OCAC significantly outperforms state-of-the-art asynchronous consensus protocols including Dumbo, Chronos, TortoiseBFT, sDumbo, and Dumbo-NG, achieving superior transaction throughput and reduced transaction latency metrics in relay chain environments.

Keywords blockchain; cross-chain asynchronous consensus; optimal cross-chain transaction matching; proposal vector selection optimization; multi-valued Byzantine consensus

1 引 言

区块链技术作为一种实现去中心化、安全、可信数据存储与交易的分布式账本技术，在多年的发展中取得了显著成就，促进了网络空间中信任与安全问题的解决[1]。然而，由于不同区块链之间的高度异构性，区块链之间的互操作性问题面临着显著挑战[2]。为解决由单链所导致的价值孤岛问题，跨链技术应运而生。跨链技术[3]广义上是指实现多个独立区块链分布式账本之间的数据和资产互操作性的技术。目前主流的跨链技术包括公证人机制、哈希锁定机制、分布式私钥机制、中继链技术等。其中，中继链技术[4]是一种利用第三方区块链在不同区块链之间建立连接的技术，并通过跨链消息传递协议实现跨链交互。相较于其他跨链技术，中继链技术具备支持异构区块链、高效跨链能力、多样化跨链功能和场景优势。中继链主要功能在于连接多个应用链，为其提供跨链交易和验证的服务。为保障中继链的安全性和稳定性，并确保所连接子链之间的一致性和可靠性，中继链迫切需要合适的共识机制来实现上述目标。

目前，大多数中继链采用部分同步共识机制，该机制依赖于固定的消息传输延迟上限，能够提供较高的确定性与较低的确认时间，适用于网络环境较为稳定的场景。然而，其局限性在于对网络假设的强依赖性、难以适应不同节点分布、在跨链通信时容易受到不同链的延迟差异的影响。同时，随着区块链生态系统规模的扩展，不同区块链之间的共识负担不断增加，导致不同分布下的共识节点难以实现全局一致性，最终影响跨链系统的整体稳定性[5]。相比之下，异步共识机制不依赖于时间假设，能够容忍任意时延的消息传输，从而即使在网络条件较差的情况下也能保持协议活性。尤其在应对节点间传输延迟波动和网络带宽、稳定性等不确定性方面，异步共识机制有更高的灵活性和鲁棒性，适合节点分布广泛、对网络要求较高的中继链场景。

目前，异步共识机制在不断发展，如HB-BFT (Honey Badger BFT, HB-BFT)[6]、Dumbo[7]等，异步共识机制正处于持续发展阶段，在吞吐量和延迟方面逐步取得改进，逐步成为增强中继链共识协议健壮性和效率的一种可行解决方案。然而异步异构中继链网络环境下，异步共识机制仍面临着吞吐量低、延迟高的问题。因此，中继链亟需引入优化的异步共识机制，以更好地实现跨链交互。

1.1 研究问题和动机

跨链技术发展至今，仍主要适用于同步或部分同步网络模型。然而在实际应用中，随着中继链需求的不断增加，现有共识机制已难以充分应对于去中心化异步网络环境的挑战，尤其在跨链交易和信息传递方面。具体表现为：同步和部分同步网络模型对时延、消息丢失、消息乱序等问题的容忍度较低，尤其是在高延迟和不稳定的网络环境中，导致节点无法实时获取其他节点的状态信息，这增加了攻击者利用网络延迟或分区进行恶意操作的风险。因此，中继链需要异步共识机制的支持，然而其在实现过程中面临以下核心挑战：如何在缺乏全局同步的情况下，确保跨链交互中不同区块链之间的一致性与安全性。

现有的异步共识机制研究中，为在不依赖任何网络时序假设时保证系统安全，解决在异步拜占庭环境下的原子广播（Atomic Broadcast，ABC）问题，并确保各节点按相同顺序接收相同消息，Miller等人[6]提出HB-BFT异步共识机制，其将ABC构建为一个核心模块ACS（Asynchronous Common Subset，ACS），再将ACS分解成可靠广播（Reliable Broadcast，RBC）模块和异步二元共识（Asynchronous Binary Agreement，ABA）模块。在RBC模块中，HB-BFT利用纠删码技术对批量交易进行分割，以减轻异步共识节点的带宽压力。同时，该异步共识机制利用门限签名技术对交易进行签名保护，以防止审查攻击。但该机制存在以下问题：每个异步共识节点都会对其他节点的RBC成功与否执行1次ABA，即每轮异步共识中都要同时执行N个ABA实例。因此，随着网络规模增大，HB-BFT的延迟明显增加。

为解决上述问题，Dumbo[7]在HB-BFT基础上进行改进，提出两种新的异步原子广播协议，即Dumbo1和Dumbo2，通过显著减少HB-BFT中的并行ABA实例来降低异步共识时延。其中，Dumbo2通过在RBC模块整合门限签名技术，确保足够数量的诚实节点接收交易。同时，其提出多值拜占庭共识模块，该模块使用CBC（Consistent Broadcast，CBC）组件，能够在N个共识节点提供的N个子集中选出1个作为共识输出，有效地将需要运行的ABA实例数量降低为1。但Dumbo2存在以下问题：其RBC模块虽基于纠删码实现，具有一致性、全局性、可信性[8]，但需要保证所有诚实节点同时收到或不收到广播，消息数量高达$O(N^3)$。同时，由于其在实际广播中存在大量冗余消息，限制了异步共识机制的性能。为解决Dumbo2的该问题，快速小飞象（Speeding Dumbo，sDumbo）[9]采用了成本更低的可证明广播（Provable Broadcast，PB）组件，并增设交易恢复模块以确保异步共识的一致性。此外，sDumbo还实现了更为高效的多值拜占庭共识机制，与Dumbo相比，在延迟和吞吐量方面有数倍改进。Gao等人[10]提出了一种异步BFT原子广播协议Dumbo-NG，该协议引入了一种多值拜占庭共识机制，能够以至少1/2的概率确保输出源自诚实节点，同时支持交易广播与多值拜占庭共识的并行执行。

尽管Dumbo系列算法目前已经取得了较大的研究进展，但若要将其应用于实际中继链系统中，仍然存在以下问题，需要进一步优化。

问题1：在交易处理层面，Dumbo与sDumbo中每个异步共识节点需要在每轮共识中自主从各自的交易池中随机选取相同数量的批量交易进行共识。尽管该随机选取的策略能降低重复交易概率，但仍无法彻底规避交易重复问题。Dumbo-NG对输入缓冲区的理想化假设，导致其异步共识吞吐量受限。同时，由于多值拜占庭共识模块几乎不消耗带宽，在带宽无限的理想情况下，随着交易数量的增加，交易广播模块延迟将上升。

问题2：在节点协作层面，尽管Dumbo和sDumbo在减少需要执行的ABA实例数量方面取得了显著进展，但若提供提议向量的异步共识节点性能较差，其他异步共识节点接收其提议向量的速度将受到影响，进而导致ABA实例的执行时延增大，降低整个异步共识过程的效率。

与实用拜占庭容错算法等有领导者的共识协议不同，目前主流的异步共识机制均采用传统的分布式系统发送交易的方式。即：客户端需要将所有具有共识需求的交易发送至所有异步共识节点的交易池，并由其对批量交易进行随机提取与并发处理。同时，在异步共识机制中，由于每个节点需要广播大批量交易，异步共识节点的带宽容易成为瓶颈。由于中继链的异步共识节点在交易广播能力上存在差异，部分节点可能因带宽受限而影响异步共识机制的整体性能。此外，异步共识机制可能会受到交易丢失、延迟等影响。

鉴于上述问题，亟需一种优化的跨链异步共识机制。为此，本文提出中继链环境中一种优化的跨链异步共识机制（Optimized Cross-chain

Asynchronous Consensus Mechanism，OCAC）。该机制针对问题1，设计一种最优跨链交易匹配（Optimal Cross-chain Transaction Matching，OCTM）方法；针对问题2，设计一种提议向量选择优化（Optimal Proposal Vector Selection，OPVS）方法。该设计通过OCTM方法优化交易处理流程，并利用OPVS方法提升节点协作效率，从而降低中继链环境中的交易延迟和提高交易吞吐量，实现跨链异步共识性能提升。

1.2　贡　献

针对上述研究问题和动机，本文贡献如下：

（1）设计基于OCTM的交易广播模块，将中继链交易池的大批量交易构建成不重叠的批量跨链交易包，并综合考虑交易包的大小、所包含的交易数量、异步共识节点的异构广播能力与共识成功率，按照最优交易匹配方法将合适的交易包发送至对应的异步共识节点进行交易广播，以解决异步共识过程中存在大量重复交易的问题，提高每个共识节点的带宽利用率与跨链交易的共识成功率，从而有效提升异步共识的效率。

（2）设计基于OPVS的多值拜占庭共识模块，利用多目标优化技术在不同异步共识轮次中综合考虑所有异步共识节点的通信延迟、运行性能、共识准确率这些特征，迭代出优质的异步共识节点集合，每轮异步共识中从该集合中随机选择节点，并对其所提出的提议向量进行异步二元共识，以减少ABA实例的执行时延，进一步提高异步共识效率。

本文工作的其余部分组织如下：第2节介绍研究现状；第3节介绍提出的模型；第4节描述所设计的算法；第5节给出实验结果及其分析；第6节总结本文工作内容并给出展望。

2　研究现状

中继链跨链机制的优点是能高效地连接和协调多个区块链，实现跨链数据和资产的安全、快速传输。而异步共识机制的优点是无需依赖精确的时间同步，从而在不稳定网络条件下仍能实现高效且稳健的共识。国内外研究者在优化异步共识机制与跨链共识机制中提出了较多的理论和策略。本节将从单链环境下的异步共识机制、跨链环境下同步或部分同步共识机制、跨链环境下异步共识机制3方面介绍研究现状。

2.1　单链环境下的异步共识机制

当前区块链共识机制主要基于同步或部分同步的网络模型，并采用了消息传输的超时机制[11]。在这种设置下，异步环境可能削弱故障检测器的有效性，进而引起持续的视图更改。在实际工程应用中，即便底层网络遵循部分同步假设，调整定时器的策略也对共识协议性能产生重要影响。定时器设置过短可能导致频繁的视图变更，而设置过长则可能使网络在分区恢复后恢复速度减慢。

与同步或部分同步共识不同，异步共识机制不依赖于超时机制的设置。为确保协议的活跃性，研究者们采取了引入随机性元素以绕开FLP不可能性定理，或基于有向无环图结构设计异步共识机制，以确保最终一致性，同时弱化一致性的要求。Duan等人[12]在HB-BFT基础上提出BEAT，通过模块化设计集成的异步BFT协议，其延迟与吞吐量性能均优于HB-BFT。然而，BEAT的延迟仍然较高，吞吐量仍有待提升。Liu等人[13]引入新范式来桥接排序和协议组件，增强了异步共识机制的灵活性和适应能力。然而，当遭遇网络攻击或恶意行为时，该方法在鲁棒性和安全方面可能存在弱点。Duan等人[14]设计了不需要签名的常数时间的多值拜占庭共识协议，该协议在不增加消息传递和通信复杂性的前提下，提高了异步共识机制的效率和性能。尽管无阈值签名是一大优势，但在需要身份认证和数据完整性保障的场合，其适用性可能会受到一定限制。Zhang等人[15]设计了WaterBear，在不依赖门限加密的前提下，实现了无条件安全性和自适应安全性，但其假设系统存在认证信道，在开放或不可信网络中不容易完全满足该假设。尽管这些研究在异步共识机制中提出了更优异的多值拜占庭共识模块，但其未考虑实际异步共识应用中恶意共识节点对异步共识进程的影响。为优化HB-BFT效率，Zhang等人[16]设计了异步拜占庭共识协议Chronos，通过结合ACS的变体与RBC，但其ABA实例的数量并未减少，进而导致延迟较高。Liu等人[17]提出了适用于物联网的具有三阶段的高性能异步共识机制TortoiseBFT，通过引入交易恢复模块与节点信誉机制，减少了异步共识的通信开销，但其对节点的信誉评价不够全面。Antunes等人[18]提出了一种实用的异步拜占庭容错共识协议Alea-BFT，设计基于可验证的一致广播和基于ABA的协议。该机制可通过客户端请求定向提交、设置请求数量上限等方式缓解交易处理的冗余问题，但仍然难以胜任高并发高吞吐量的场景。

由上述分析可知，异步共识机制在单链环境中展现出其独特优势，尤其在弱化一致性要求和应对网络不稳定性方面表现突出，然而也面临吞吐量等性能瓶颈。目前跨链环境大多仍然采用同步或部分同步共识机制，如下述所示。

2.2 跨链环境下的同步或部分同步共识机制

在跨链交互场景中，共识机制不仅是跨链数据传输的桥梁，更是确保交易安全性和网络整体稳定性的关键。当区块链系统从单链扩展至多链时，共识机制的设计需要考虑更多因素。目前诸多学者已针对跨链共识机制展开研究。Shao等人[19]提出对跨链通信过程中的公证人进行身份加密，以构建安全高效的跨链共识机制，但其难以保证所有公证人是可靠的。Su等人[20]设计了基于交易的跨链交换模型，改变了区块链共识中的平衡过程，通过在交易中嵌入条件实现了资产转移的条件判定，但其跨链场景有限，局限于资产之间的交换。Yin等人[21]面向PoW/PoS共识机制的区块链系统，通过生成跨链证书并选择委员会来快速处理即时的跨链交易，但其在应对较为复杂的网络条件时性能较差。Nguyen等人[22]提出一种创新的基于PoS的共识机制，用于构建基于区块链的联邦学习系统，以提高对区块链特定攻击的防范能力并实现不同区块链系统的代币有效转移，但该机制无法适应异步网络环境。Jalalzai等人[23]提出了一种部分同步拜占庭容错共识协议Fast-HotStuff，其在经典HotStuff协议基础上，通过引入快速响应机制与高效视图切换策略，在部分同步网络模型下有较好的性能表现，但该机制同样难以适应异步网络环境。

中继链典型项目有Polkadot、Cosmos、BitXHub。Polkadot[24]采用GRANDPA与BABE相结合的混合共识机制。其中，GRANDPA是最终性协议，而BABE是区块生产协议，同时保证了跨链共识的最终性与活性。该机制虽然在中继链层面保障了网络安全性，但是在应用链和桥接链层面仍然存在安全风险，如应用链可能被恶意验证者攻击。Cosmos[25]采用BFT与PoS的混合共识机制，具备部分同步拜占庭容错特性，同时引入了分叉追责机制，但受限于较高的通信复杂度，在节点规模较大的跨链系统中适用性可能受限。BitXHub[26]支持多种共识机制，如Raft、Solo、Tendermint、LibP2P等，可以根据不同场景和需求选择共识机制进行灵活配置，然而其共识机制难以适应不稳定的网络环境。除此之外，Xie等人[27]针对物联网设备身份验证问题，提出一种基于中继链的跨链解决方案，验证设备节点的合法性。与传统中继链不同，该方案由仲裁委员会节点而不是所有节点来执行共识机制，但难以保证委员会节点的计算资源稳定性与网络通信可靠性。

由上述分析可知，跨链环境中的同步或部分同步共识机制设计难度相对较低，其主要关注点集中在安全性上，在部分同步场景中，活性通常由故障检测器辅助保障，但难以做到完全控制。因此，在跨链环境中，异步共识机制的支持显得尤为必要。

2.3 跨链环境下的异步共识机制

目前跨链异步共识机制方面的研究工作相对较少。为提高元宇宙中区块链的可扩展性，Xie等人[28]提出了一种基于中继链和异步共识的联盟区块链跨链模型。与传统的中继链模型（多采用同步或部分同步共识）相比，该模型虽同样利用中继链和跨链网关实现链间消息传递与跨链交易，但其核心差异在于引入了传统的HB-BFT异步共识机制。由于受限于HB-BFT的多轮通信逻辑，其共识效率较低，延迟较高。因此，尽管其为跨链异步共识机制的探索提供了一定思路，但仍存在优化空间，特别是在降低异步共识通信复杂度、提升跨链交易吞吐量等方面。

结合上述分析可知：(1)目前大部分跨链共识机制主要依赖同步或部分同步共识机制[19-26]。在实际的应用中，由于物理环境无法达到完全同步，跨链共识方案面临可靠性与稳定性的挑战。因此，中继链跨链机制亟需异步共识机制的支持。(2)目前异步共识机制的研究较少[28]，现有工作多专注于改进多值拜占庭共识协议[13-17]的容错性，较少关注跨链场景下的重复交易对共识吞吐量与延迟的影响，而重复交易会占用节点计算与通信资源，降低有效吞吐量。若将异步共识机制应用于中继链环境中，需通过机制设计避免对重复交易的无效共识，以提升性能。(3)在异步网络环境中，确保共识节点的可靠运作仍然是一个难题[17]，具备节点保障的异步共识机制开发目前还处于初步探索阶段。为此，本文提出中继链环境中一种优化的跨链异步共识机制。

3 中继链环境中一种优化的跨链异步共识机制

3.1 相关概念

(1)应用链：指有跨链交互需求的区块链，主要分为来源链与目标链。其中，来源链为发起跨链交易的区块链，目标链为接收跨链交易的区块链。不

同应用链均与中继链相互连接，通过中继链的服务进行跨链交易。应用链中的验证节点负责验证跨链交易，以确保跨链交易的合法性与正确性。

(2)中继链：在跨链交互中，中继链作为多个区块链之间的桥梁，支持各链数据和交易的跨链传递，并通过共识机制保障跨链交易的最终性与各链状态一致性。中继链接收来源链提交的跨链交易后，先验证合法性，再通过共识确认交易有效性，最终将其转发至目标链中的对应用户[4]。所有跨链交易最终都会记录在中继链的区块中。中继链中的节点划分为收集节点与异步共识节点两大类。其中，收集节点是应用链与中继链间跨链交易收集与路由的关键，主要负责监控中继链各个异步共识节点的运行状态，处理并转发跨链交易至异步共识委员会进行共识。而异步共识节点则负责对跨链交易进行验证并达成共识，将其打包成区块后在中继链网络中广播，确保全网节点同步账本，以保障跨链交互的一致性与安全性。

(3)异步共识委员会：本文采用异步共识机制解决中继链跨链系统中的全局一致性问题。在异步共识过程中，交易广播阶段需要消耗大量带宽，且随着节点数量增加，通信成本呈现上升趋势。为降低通信开销，本文设立由N个异步共识节点组成的异步共识委员会，确保系统的高效性和可靠性。

(4)跨链交易包：经来源链验证节点验证通过的跨链交易按照“先进先出”原则进入中继链交易池。为实现跨链交易的并发处理，收集节点集群将中继链交易池中的跨链交易随机划分为满足条件的交易包，其中交易包中的交易互不重叠，并将其分发给相应的异步共识节点进行共识。由于跨链交易的大小各不相同，经过收集节点集群处理后的各个交易包的大小与交易数量存在差异。来源链发起的跨链交易在中继链中完成初步共识，并且目标链中的相应用户执行完成后，目标链中的验证节点需要对来源链发送的跨链交易包进行确认，形成跨链确认交易包，以确保跨链交易在目标链上的正确执行和合法性。令R_{ix}表示交易包的结构，如式(1)所示：

$$R_{ix}=<i, x, Z, O> \tag{1}$$

其中，i为交易包的编号，用来区别不同的交易包。x为交易包的类别($x\in\{1, 2\}$，$x=1$表示跨链交易包，$x=2$表示跨链交易确认包)。Z表示交易包中跨链交易来自来源链或目标链中验证节点的签名集合，O表示交易包中的跨链交易集合。

(5)共识提案：异步共识节点从各自的交易池中依次提取中继链收集节点集群发送的对应轮次的交易包进行验证，生成共识提案以执行异步共识。令P_j表示共识节点j输入的共识提案的结构，如式(2)所示：

$$P_j=<R_{ix}, F_j> \tag{2}$$

其中，F_j表示异步共识节点j的签名。

3.2　优化的跨链异步共识机制系统架构

图1展示的是优化的跨链异步共识机制系统架构。假设来源链$\mathcal{A}$上的用户通过中继链向目标链$\mathcal{B}$上的用户发起跨链交易，其异步共识分为两个递进阶段。其中，第1阶段针对从来源链$\mathcal{A}$至中继链的跨链交易包进行异步共识，目的是确保来源链发起的跨链交易在中继链中得到验证和共识。在此基础上，第2阶段针对从目标链$\mathcal{B}$至中继链的跨链交易确认包进行异步共识，其目的是确认跨链交易在目标链上得到执行与验证。

图1中的第1阶段(从来源链$\mathcal{A}$至中继链)包括步骤1～7，第2阶段(从目标链$\mathcal{B}$至中继链)包括步骤8～12，具体步骤如下。

步骤1：来源链$\mathcal{A}$中的用户发起跨链交易。

步骤2：来源链$\mathcal{A}$中的验证节点验证来源链中用户发起的跨链交易的正确性与合法性，并将验证通过的跨链交易与对应的来源链验证节点签名按照“先进先出”的顺序添加至来源链交易队列。

步骤3：中继链收集节点集群依次从来源链$\mathcal{A}$的交易队列中提取跨链交易与其对应的验证签名至中继链交易池。来源链$\mathcal{A}$中的验证节点删除交易队列中的对应跨链交易。为实现跨链交易的并发处理，中继链收集节点集群将中继链交易池中的跨链交易随机划分为互不重叠的K个交易包R_{i1}，其中$K=l\times N$，$l\in N^{+}$，l表示K个交易包的预期共识执行轮数，$i\in\{1, 2, \cdots, K\}$。

步骤4：中继链收集节点集群实现跨链交易包与异步共识节点之间的匹配，并转发对应的交易包R_{i1}至对应的异步共识节点，进入第1阶段跨链异步共识。优化的跨链异步共识机制框架如图2所示。异步共识委员会中的异步共识节点提取对应的交易包将其作为共识提案，并对其进行门限加密。共识提案的内容只有在异步共识完成且收到超过$f+1$个份额的确认后才能解密，以防止审查攻击。其中，f表示异步共识委员会中恶意节点的数量。

步骤5：第1阶段跨链异步共识完成后，共识提案P的内容被解密，异步共识委员会将共识通过的跨链交易与对应结果发送给中继链收集节点集群。

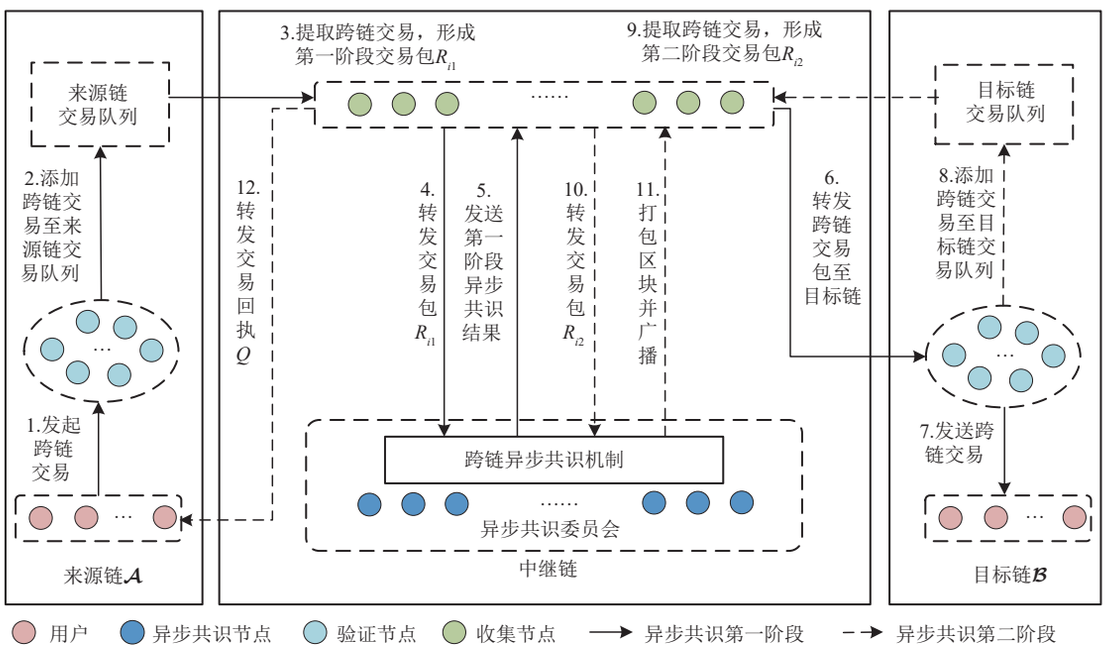


图1 优化的跨链异步共识机制系统架构图

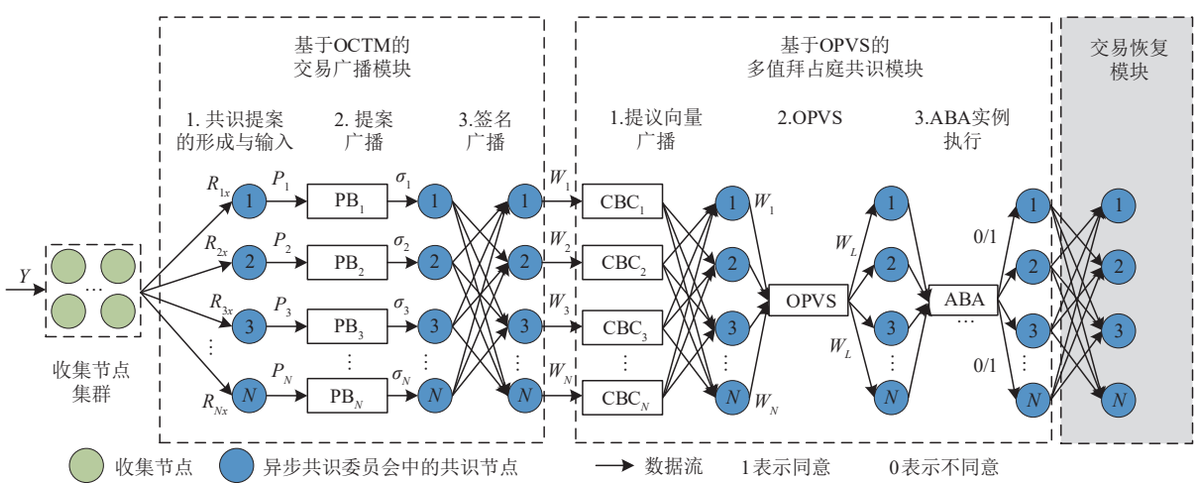


图2 OCAC框架图

（其中Y表示交易包集合，针对N个异步共识节点，有PB_1~PB_N相应组件，PB_i完成后输出生成的签名σ_i进行签名广播，节点i生成提议向量$\boldsymbol{W}_i$。每个节点广播自己的提议向量，有CBC_1~CBC_N相应组件，经过OPVS后，模块输出节点L的提议向量$\boldsymbol{W}_L$，以执行ABA实例）

步骤6：收集节点集群转发共识成功的跨链交易至目标链$\mathcal{B}$中的验证节点。

步骤7：目标链$\mathcal{B}$中的验证节点转发跨链交易至对应用户，目标用户执行相应跨链交易。

步骤8：目标用户执行完成后，目标链$\mathcal{B}$中的验证节点对第1阶段的跨链交易执行结果进行验证，并生成相应的交易收据Q。若验证成功，其将跨链交易及其对应的目标链验证节点签名添加到目标链$\mathcal{B}$的交易队列中。

步骤9：中继链收集节点集群依次从目标链$\mathcal{B}$的交易队列中提取执行完成的跨链交易与对应的目标链验证节点签名，以形成交易确认包R_{i2}，与第1阶段步骤3类似。目标链$\mathcal{B}$中的验证节点删除交易队列中的对应跨链交易。

步骤10：中继链收集节点集群将交易确认包R_{i2}转发至异步共识委员会，并进入第2阶段跨链异步共识（与第1阶段步骤4类似，核心是对目标链交易执行结果的合法性达成共识）。

步骤11：第2阶段跨链异步共识完成后，异步共识节点将完成两阶段共识的跨链交易打包成区块上链，并进行全网广播。收集节点集群在中继链交易池中删除完成两阶段共识的跨链交易，以释放资源。

步骤12：最后，为确保跨链交易的原子性（即来源链与目标链的状态变更一致），中继链收集节点集群将跨链交易所对应的回执Q转发至来源链$\mathcal{A}$中的对应发起用户，完成全流程闭环。

基于图1的跨链异步共识机制架构，本文假设中继链共识过程在异步网络环境中进行。异步共识节点之间的交易传输与处理不依赖于全局时钟信号，只保证跨链交易能到达各个节点而没有到达时间的限制。即便对手能够无限制地拖延时间，

诚实异步共识节点之间发送的值最终会被传递。基于图1，异步共识委员会需在中继链中执行两阶段异步共识：第1阶段针对步骤4中收集节点转发的交易包R_{i1}，第2阶段针对步骤10中收集节点转发的交易确认包R_{i2}。具体优化设计如3.3节所示。

3.3　OCAC设计

本文所提出的OCAC框架如图2所示，主要分成3个核心模块：交易广播模块、多值拜占庭共识模块、交易恢复模块。首先，本文在3.3.1节设计基于OCTM的交易广播模块，分为共识提案的形成与输入、提案广播、签名广播3个步骤。该模块针对共识提案的形成与输入进行改进，并使用PB组件进行广播；其次，在3.3.2节设计基于OPVS优化的多值拜占庭共识模块，分为提议向量广播、OPVS、ABA实例执行3个步骤；最后在有交易缺失时执行交易恢复模块。

3.3.1　基于OCTM的交易广播模块

（1）共识提案的形成与输入

令中继链异步共识委员会中共识节点集合为$X=\{1,2,\cdots,N\}$，中继链交易池中经收集节点集群处理过的所有待共识的交易包集合为$Y=\{1,2,\cdots,K\}$。为避免异步共识过程中交易的重复，1个交易包只能与唯一的异步共识节点进行匹配。异步共识过程中，收集节点集群依次分配共识轮次相同的N个交易包给异步共识委员会进行共识。

令矩阵$\boldsymbol{U}_K$表示K个跨链交易包在l轮异步共识中与N个异步共识节点之间的关系映射矩阵，如式(3)所示：

$$\boldsymbol{U}_K=\begin{bmatrix}u_{11} & u_{21} & \cdots & u_{1N}\\ u_{21} & u_{22} & \cdots & u_{2N}\\ \vdots & \vdots & \ddots & \vdots\\ u_{l1} & u_{l2} & \cdots & u_{lN}\end{bmatrix} \tag{3}$$

s. t.

$$u_{ij}\in\{1,2,\cdots,K\},$$
$$\forall i\in\{1,2,\cdots,l\},\ \forall j\in\{1,2,\cdots,N\}$$

其中，矩阵$\boldsymbol{U}_K$中元素u_{ij}表示异步共识节点j在第i轮异步共识中所对应的交易包编号，并且每个元素u_{ij}仅有对应唯一的交易包编号，且所有u_{ij}的取值互不重复。其中，$u_{ij}\in\{1,2,\cdots,K\}$（$\forall i\in\{1,2,\cdots,l\}$，$\forall j\in\{1,2,\cdots,N\}$）确保矩阵$\boldsymbol{U}_K$中所有元素都属于交易包编号的集合，避免无效编号导致的共识错误。令矩阵$\boldsymbol{G}_K$为K个交易包所对应的大小矩阵，如式(4)所示：

$$\boldsymbol{G}_K=[g_1,g_2,\cdots,g_z,\cdots,g_K]^{\mathrm{T}} \tag{4}$$

其中，g_z表示交易包z的大小。由于中继链异步共识节点广播交易的能力存在差异，异步共识的交易广播中会出现延迟不均衡情况。为加快整个中继链的异步共识过程，跨链交易匹配的过程中需要考虑异步共识节点在交易广播模块中广播跨链交易的速度。这一速度由收集节点集群对中继链的异步共识过程中各共识节点的状态进行监控并计算得出，以便将交易包优先分配给广播效率更高的节点，从而减少整体共识延迟。令v_j表示异步共识节点j在q轮异步共识过程中广播q个跨链交易包的速度，如式(5)所示：

$$v_j=\left(\sum_{z=1}^{q}w_{jz}\right)\Big/\left(\sum_{z=1}^{q}s_{jz}\right) \tag{5}$$

其中，w_{jz}表示异步共识节点j所广播的交易包z的大小，s_{jz}表示异步共识节点j广播交易包z所花费的时间。为便于计算每个异步共识节点广播各个交易包的预期时间，令矩阵$\boldsymbol{V}_N$表示N个异步共识节点的交易广播能力矩阵，如式(6)所示：

$$\boldsymbol{V}_N=[1/v_1,1/v_2,\cdots,1/v_j,\cdots,1/v_N]^{\mathrm{T}} \tag{6}$$

由式(4)与式(6)，令矩阵$\boldsymbol{H}_K$表示N个异步共识节点在交易广播模块中分别广播K个交易包的预期时间矩阵，如式(7)所示：

$$\boldsymbol{H}_K=\boldsymbol{V}_N(\boldsymbol{G}_K)^{\mathrm{T}} \tag{7}$$

结合式(3)中的矩阵$\boldsymbol{U}_K$与式(7)中的矩阵$\boldsymbol{H}_K$，可得K个跨链交易包在l轮异步共识中被N个异步共识节点广播所需的预期时间矩阵，记为$\boldsymbol{M}_K$，如式(8)所示：

$$\begin{aligned}\boldsymbol{M}_K=[\,&m_{ij}|m_{ij}=h_{j,u_{ij}},u_{ij}\in\boldsymbol{U}_K,h_{j,u_{ij}}\in\boldsymbol{H}_K,\\ &i\in\{1,2,\cdots,l\},j\in\{1,2,\cdots,N\}\,]\end{aligned} \tag{8}$$

式(8)中，m_{ij}表示矩阵$\boldsymbol{M}_K$中的元素，$h_{j,u_{ij}}$表示节点j在第i轮异步共识中广播编号为u_{ij}的交易包所需的预期时间。令异步共识节点j的交易池中待共识的交易包队列为$L_j=\{1,2,\cdots,m,\cdots,l\}$。令$t_m$表示中继链中轮次为$m$的异步共识过程中广播交易

的预期执行时间，由广播速度最慢的节点决定，如式(9)所示：

$$t_m=\max_{j\in\{1,2,\cdots,N\}}(b_j^m/v_j) \tag{9}$$

其中，b_j^m表示异步共识节点j的交易池中交易包队列中的第m个交易包的大小。由此，令矩阵$\boldsymbol{T}_K$表示K个交易包在l轮异步共识中交易广播的预期执行时间矩阵，如式(10)所示：

$$\boldsymbol{T}_K=[t_1,\ t_2,\ \cdots,\ t_l]^{\mathrm{T}} \tag{10}$$

为减少异步共识机制受到交易丢失、延迟等影响，本文将处理能力更强且成功率更高的节点与合适的交易包进行交易匹配。令$\mathcal{G}_j$表示异步共识节点j在q轮异步共识过程的共识成功率，如式(11)所示：

$$\mathcal{G}_j=1-\left(\sum_{z=1}^{q}\mathcal{T}_{jz}/\sum_{z=1}^{q}\mathcal{R}_{jz}\right) \tag{11}$$

其中，$\mathcal{T}_{jz}$表示异步共识节点j在第z轮异步共识中所提出的跨链交易未被包含在最终共识成功交易集合中的交易数量，$\mathcal{R}_{jz}$表示异步共识节点j在第z轮异步共识中输入的跨链交易总数量。令矩阵$\boldsymbol{\mathcal{G}}_N$表示$N$个异步共识节点的共识成功率矩阵，如式(12)所示：

$$\boldsymbol{\mathcal{G}}_N=[\mathcal{G}_1,\ \mathcal{G}_2,\ \cdots,\ \mathcal{G}_j,\ \cdots,\ \mathcal{G}_N]^{\mathrm{T}} \tag{12}$$

令矩阵$\boldsymbol{\mathcal{Q}}_K$为$K$个交易包所对应的交易数量矩阵，如式(13)所示：

$$\boldsymbol{\mathcal{Q}}_K=[\mathcal{Q}_1,\mathcal{Q}_2,\cdots,\ \mathcal{Q}_i,\cdots,\mathcal{Q}_K]^{\mathrm{T}} \tag{13}$$

其中，$\mathcal{Q}_i$表示第i个交易包所包含的跨链交易数量。因此，由式(12)与式(13)可得N个异步共识节点在交易广播模块中分别广播K个交易包的预期共识成功交易数量矩阵，记为$\boldsymbol{\mathcal{S}}_K$，如式(14)所示：

$$\boldsymbol{\mathcal{S}}_K=\boldsymbol{\mathcal{G}}_N(\boldsymbol{\mathcal{Q}}_K)^{\mathrm{T}} \tag{14}$$

结合式(3)中的矩阵$\boldsymbol{U}_K$与式(14)中的矩阵$\boldsymbol{\mathcal{S}}_K$，可得$K$个跨链交易包在$l$轮异步共识中被$N$个异步共识节点广播的预期共识成功交易数量矩阵为矩阵$\boldsymbol{\mathcal{O}}_K$，如式(15)所示：

$$\begin{aligned}\boldsymbol{\mathcal{O}}_K=[&n_{ij}|n_{ij}=s_{j,u_{ij}},u_{ij}\in\boldsymbol{U}_K,s_{j,u_{ij}}\in\boldsymbol{\mathcal{S}}_K,\\&i\in\{1,\ 2,\ \cdots,\ l\},\ j\in\{1,\ 2,\cdots,N\}]\end{aligned} \tag{15}$$

其中，n_{ij}表示矩阵$\boldsymbol{\mathcal{O}}_K$中的元素，$s_{j,u_{ij}}$表示节点$j$在第$i$轮异步共识中广播编号为$u_{ij}$的交易包的预期共识成功交易数量。

为最大程度提高中继链中异步共识的速度，本文综合考虑交易包的大小、所包含的交易数量、异步共识节点的性能，在避免节点成为性能瓶颈的同时，减少交易包处理时间过长而带来的异步共识时间长的问题。同时，本文考虑每个异步共识节点的共识成功率，使得每轮异步共识能够处理更多跨链交易。因此，本文依据式(8)、(10)、(15)，选择K个交易包的平均预期广播时间$\mathcal{U}_1$、平均每轮异步共识的预期交易广播执行时间$\mathcal{U}_2$、K个交易包的预期共识成功率$\mathcal{U}_3$作为优化目标，如式(16)～(18)所示：

$$\mathcal{U}_1=\frac{\sum_{a=1}^{l}\sum_{b=1}^{N}\{r_{ab}|r_{ab}\in\boldsymbol{M}_K\}}{K} \tag{16}$$

$$\mathcal{U}_2=\frac{\sum_{c=1}^{l}\{y_c|y_c\in\boldsymbol{T}_K\}}{l} \tag{17}$$

$$\mathcal{U}_3=\frac{\sum_{a=1}^{l}\sum_{b=1}^{N}\{g_{ab}|g_{ab}\in\boldsymbol{\mathcal{O}}_K\}}{\sum_{d=1}^{K}\{w_d|w_d\in\boldsymbol{\mathcal{Q}}_K\}} \tag{18}$$

式(16)中，r_{ab}表示矩阵$\boldsymbol{M}_K$中的元素。式(17)中，y_c表示矩阵T_K中的元素。式(18)中，g_{ab}表示矩阵$\boldsymbol{\mathcal{O}}_K$中的元素，$w_d$表示矩阵$\boldsymbol{\mathcal{Q}}_K$中的元素。矩阵$\boldsymbol{M}_K$、矩阵$T_K$、矩阵$\boldsymbol{\mathcal{O}}_K$都与矩阵$\boldsymbol{U}_K$有关，而矩阵$\boldsymbol{U}_K$是$K$个交易包与$N$个异步共识节点在$l$轮异步共识中的关系映射矩阵。因此，本文通过调整关系映射矩阵$\boldsymbol{U}_K$，通过最小化$\mathcal{U}_1$、$\mathcal{U}_2$、最大化$\mathcal{U}_3$来提高异步共识速度，如式(19)所示：

$$\min(\mathcal{U}_1,\ \mathcal{U}_2,\ -\mathcal{U}_3) \tag{19}$$

其中，最小化$\mathcal{U}_1$倾向于让广播速度快的节点处理更大的交易包，以缩短整体的异步共识时间；最小化$\mathcal{U}_2$倾向于让所有异步共识节点在每轮异步共识过程中广播交易包的时间更加均衡；而最大化$\mathcal{U}_3$则倾向于使得异步共识成功的跨链交易数量最大化。

收集节点集群依据得到的矩阵$\boldsymbol{U}_K$，依次将轮次相同的N个交易包转发至对应的异步共识节点。中继链每轮异步共识中，N个异步共识节点依次提取轮次相同的交易包进行验证签名，以形成共识提案进行输入。如果中继链完成异步共识，则需返回共识成功的信息至收集节点集群。否则，收集节点集群进行收集整理，并将交易包重新发送至其他异步共识节点。

(2)提案广播

为降低交易广播模块的消息复杂度，本文采用PB组件[25]进行共识提案的广播，只需保证$f+1$个诚实节点收到同1个共识提案，并加入交易恢复模块以保证异步共识机制的全局性。PB组件的具体过程分为两个步骤，以异步共识节点j作为发送者为例，如图3所示。首先异步共识节点j广播共识提案

P_j至共识委员会中其他共识节点。其次，共识节点i对P_j进行验证并生成签名ω_i，随后将其返回给发送者。若异步共识节点j收到$2f+1$个有效签名，则对其进行聚合并生成阈值签名σ_j进行输出。由于签名较小，不同异步共识节点执行PB组件的延迟差异主要取决于PB组件中的共识提案广播。

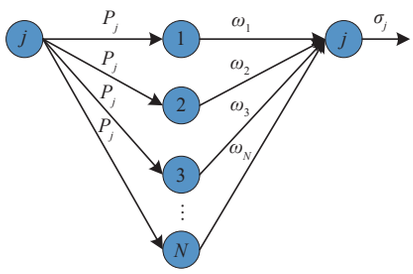


图3　PB组件的具体过程(以异步共识节点j作为发送者)

(3)签名广播

异步共识节点广播PB组件输出的阈值签名至其余共识节点，以证明有足够数量的诚实异步共识节点收到了共识提案。令$\boldsymbol{W}$表示签名广播完成后异步共识节点输出的提议向量，如式(20)所示：

$$\boldsymbol{W}=\left[(\tau_1,\sigma_1),\cdots,(\tau_j,\sigma_j),\cdots,(\tau_{f+1},\sigma_{f+1})\right] \tag{20}$$

其中，τ_j表示第j个PB组件的编号。若节点i收到来自第j个PB组件的阈值签名(记为σ_j)，则将σ_j与对应的编号τ_j放入对应的提议向量$\boldsymbol{W}$中。

3.3.2　基于OPVS的多值拜占庭共识模块

(1)提议向量广播

交易广播模块完成后，每个异步共识节点输出提议向量，并将其作为多值拜占庭共识模块的输入，随后使用CBC组件[7]广播至其他共识节点以便进行后续投票。与RBC相比，CBC速度较快，但不提供全局性。即在诚实节点中，若某个节点成功输出了某个值，CBC不要求所有其他诚实节点也一定会输出该值，仅保证不同诚实节点的输出结果一致。由于$\boldsymbol{W}$采用的阈值签名较小，能够显著提高多值拜占庭共识的处理速度。

(2)OPVS

在异步共识机制中，通常通过从N个异步共识节点提出的N个提议向量$\boldsymbol{W}$中随机选取1个，以执行ABA实例，从而达成共识。然而，若存在恶意节点攻击，其余节点将会针对错误的提议向量执行ABA实例，从而拖慢异步共识进程。同时，若选取的节点发送提议向量的速度较慢或网络状况较差，中继链异步共识委员会中其他共识节点收到该节点的提议向量的时间将会过长。这将导致ABA实例的输出值为0，同时ABA实例的执行时延增大。因此，为减少异步共识中ABA的执行时延，本文提出一种OPVS算法，通过综合考虑异步共识节点间的通信延迟、运行性能值、共识准确率，在异步共识委员会中迭代出最佳或最接近最佳节点集合S，其中$S\subset X$。中继链在每轮异步共识中从集合S中随机选取1个节点，并对其提议向量执行ABA，以提高多值拜占庭共识模块的执行速度。令d_{ij}表示异步共识节点i、j之间的通信延迟，如式(21)所示：

$$d_{ij}=\gamma_j-h_i \tag{21}$$

其中，h_i表示节点i发送消息的时刻，γ_j表示节点j接收到消息的时刻。令ξ_j表示n轮异步共识中节点j在多值拜占庭共识模块中的运行性能值，如式(22)所示。ξ_j值越大，节点j越稳定。

$$\xi_j=n/\sum_{i=1}^{n}\kappa_j^i \tag{22}$$

其中，κ_j^i表示节点j完成第i轮异步共识中多值拜占庭共识的时延。κ_j^i值越小，则表明节点j的运行性能越好。本文利用异步共识节点在n轮异步共识中输入ABA实例的共识准确率，以衡量其诚实程度。令φ_j表示节点j在n轮异步共识中的准确率，如式(23)所示：

$$\varphi_j=\eta_j/n \tag{23}$$

其中，η_j表示节点j在n轮异步共识中向ABA实例中输入正确值的次数。为确保中继链异步共识委员会中的所有节点都能快速收到提议向量，以对ABA实例进行输入，本文考虑节点之间的通信延迟。令$\bar{D}_S$表示最佳节点集合S中节点到其余所有节点的平均通信延迟，如式(24)所示：

$$\bar{D}_S=\frac{\sum_{j\in S,z\in X}d_{jz}}{k(N-1)} \tag{24}$$

其中，j表示集合S中的节点，z表示异步共识委员会X中的节点，k表示集合S中节点的数量。依据式(22)，令$\bar{J}_S$表示最佳节点集合S中节点的平均运行性能值，如式(25)所示：

$$\bar{J}_S=\frac{\sum_{j\in S}\xi_j}{k} \tag{25}$$

依据式(23)，令$\bar{\phi}_S$表示最佳节点集合S中节点的平均共识准确率，如式(26)所示：

$$\bar{\phi}_S=\frac{\sum_{j\in S}\varphi_j}{k} \tag{26}$$

构建最佳节点集合S之前，本文通过最小化集合S中节点到其余节点的平均通信延迟$\bar{D}_S$、最大化集合S中节点平均运行性能值$\bar{J}_S$、最大化最佳节点集合S节点的平均共识准确率$\bar{\phi}_S$，以优化该集合节点性能，如式(27)所示。

$$\max(-\bar{D}_S,\ \bar{J}_S,\ \bar{\phi}_S) \tag{27}$$
$$\text{s.t.}\begin{cases} f+1\leqslant k<N \\ S\in\Omega \end{cases}$$

其中，Ω表示解空间，表示所有满足条件的节点集合S的集合。由于三个目标相互制约，为最大程度上提高集合S中节点的整体性能，本文选择集合S中节点数量k作为约束条件，从异步共识委员会X中筛选出优异的节点，以构建最佳节点集合。在多值拜占庭共识中，OPVS算法随机从集合S中随机选取节点L，并将其提议向量$\boldsymbol{W}_L$作为OPVS算法的输出。

(3)ABA实例执行

异步共识委员会中的其余异步共识节点针对最佳节点集合S中随机选择的节点L所广播的提议向量$\boldsymbol{W}_L$进行投票。若其同意，则向ABA实例中输入1，否则，输入0。同时，当异步共识节点将ABA的输入设置为0，而其随后接收到来自超过$f+1$个其他节点的消息，指示ABA实例应被输入为1，则该节点将会调整其策略，将其输入更新为1。若ABA实例有超过$2N/3$的输入为0或1，则其对应的结果对应为0或1。由于异步网络的原因，诚实节点的输入会出现可能难以达到统一的情况，因此，ABA实例中引入随机数，以实现公平决策。当ABA实例的输出为1，则表明所有节点对节点L的提议向量$\boldsymbol{W}_L$达成了共识。若ABA实例的输出为0，则从集合S中重新随机选取节点，并对其提议向量执行ABA实例。若ABA实例输出为0的次数达到了$f+1$，则共识失败。

多值拜占庭共识模块结束后，当ABA实例的输出为1，所有异步共识节点得到一致的异步共识结果$\boldsymbol{W}_L$。由于PB组件只能保证至少有$f+1$个诚实共识节点收到相应共识提案，所有异步共识节点在多值拜占庭共识模块结束后需要对比异步共识的结果检查是否收到$\boldsymbol{W}_L$中所有的跨链交易，若全部收到则直接输出结果，否则需要广播请求交易消息，以寻求其他异步共识节点的帮助，恢复缺失的跨链交易。

4 算法设计

基于上述所提模型，本文设计OCTM算法(算法1)、异步共识优化算法(算法2)、OPVS算法(算法3)。算法流程图如图4所示。

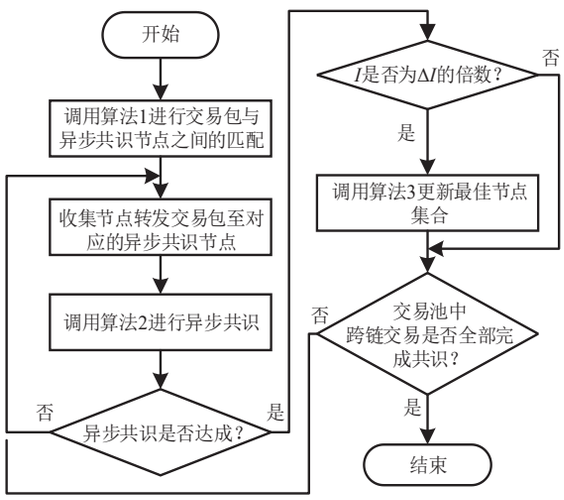


图4 算法流程图

(1)中继链系统初始运行阶段，系统首先调用算法1实现交易池中交易包与异步共识节点之间的最优匹配。收集节点转发交易包至所得到的最优交易匹配矩阵中对应的异步共识节点；

(2)系统调用算法2，使得异步共识节点从各自交易池中提取对应轮次的交易包进行异步共识；

(3)系统判断异步共识是否达成。若异步共识失败，则重新调用算法2对失败的交易包进行异步共识。若异步共识成功，系统则需判断当前异步共识轮次I是否为ΔI的倍数(ΔI表示评估周期)。若是，则调用算法3更新最佳节点集合，以提高异步共识效率。若不是，则判断中继链交易池中的跨链交易是否全部达成共识。若没有，则进行下一轮异步共识，否则，则结束中继链异步共识流程。

4.1 OCTM算法

本文设计OCTM算法，如算法1所示。相关解释如下。

(1)该算法的输入参数中，N表示异步共识节点总数，l表示K个交易包的预期共识执行轮数。其中，$K=lN$。q表示评估轮数；一维向量$\boldsymbol{Q}$、$\boldsymbol{G}$分别表示K个交易包的交易数量与对应大小；二维数组A、B分别表示异步共识节点在q轮异步共识中广播对应交易包的大小数组与所用时间数组；二维数组

C、D分别表示异步共识节点在q轮异步共识中所提出的跨链交易未被包含在最终共识成功交易集合中的对应交易数量数组与输入的对应跨链交易数量数组；n表示种群规模大小；t表示交叉率；a表示迭代次数。输出参数$\mathbb{C}$表示得到的最优跨链交易匹配数组。

(2)令数组$\boldsymbol{V}$、$\boldsymbol{W}$分别表示N个异步共识节点的交易广播能力数组与共识成功率数组，$\mathbb{Y}$表示交易匹配数组(行1～2)。c_1、c_2、c_3、c_4初始化为0，分别表示异步共识节点在第q轮异步共识中所广播的交易包的大小总和、所花费的时间总和、所提出的跨链交易未被包含在最终共识成功交易集合中的交易数量总和、输入的跨链交易数量总和。其中，数组$\boldsymbol{V}$与数组$\boldsymbol{W}$中元素的计算过程分别对应式(5)、(11)(行3～12)。

算法1： OCTM算法

输入参数：N, K, l, q, $\boldsymbol{Q}$, $\boldsymbol{G}$, A, B, C, D, n, t, a;

输出参数：$\mathbb{C}$;

1　double $\boldsymbol{V}[N]=$ NULL, $\boldsymbol{W}[N]=$ NULL;
2　int $\mathbb{Y}[l][N]=$ NULL; int $\mathbb{C}[l][N]=$ NULL;
3　/* 节点性能评估 */
4　For (int $j=1$; $j\leqslant N$; j++)
5　　double $c_1=0$, $c_2=0$; $c_3=0$, $c_4=0$;
6　　For (int $z=1$; $z\leqslant q$; z++)
7　　c_1 += $A[j][z]$; c_2 += $B[j][z]$;
8　　c_3 += $C[j][z]$; c_4 += $D[j][z]$;
9　　EndFor
10　　$\boldsymbol{V}[j]=c_2/c_1$; /* 对应式(5) */
11　　$\boldsymbol{W}[j]=1-(c_3/c_4)$; /* 对应式(11) */
12　EndFor
13　$\mathbb{Y}\leftarrow$生成$l\times N$的随机排列数组(整数1～K互不重复)
14　/* 多目标评估 */
15　double $S_1=0$, $S_2=0$, $S_3=0$;
16　double $U[l]=$ NULL; double $T=0$; double $F=0$;
17　For (int $i=1$; $i\leqslant l$; i++) /* 遍历每轮共识 */
18　　double r = 0; /* 记录每轮最大时间 */
19　　For (int $j=1$; $j\leqslant N$; j++)/* 遍历每个节点 */
20　　double $x=0$, $y=0$, $s=0$;
21　　$x=\mathbb{Y}[i][j]$; /* 获取交易包编号 */
22　$y=\boldsymbol{G}[x]\times\boldsymbol{V}[j]$; /* 计算第$j$个节点处理第$x$个交易包的预期时间 */
23　　$T=T+y$; /* 累加时间 */
24　$s=\boldsymbol{Q}[x]\times\boldsymbol{W}[j]$; /* 计算第$j$个节点处理第$x$个交易包的预期共识成功交易数量 */
25　　$F=F+s$; /* 累加共识成功交易数*/
26　　$r=\max(r, y)$; /* 更新本轮最大处理时间*/
27　　EndFor
28　　$U[i]=r$;
29　EndFor
30　$S_1\leftarrow T/K$; /*计算K个交易包平均预期广播时间*/
31　$S_2\leftarrow(U[1]+U[2]+\cdots+U[l])/l$; /* 计算平均每轮异步共识的交易广播预期执行时间 */
32　$S_3\leftarrow F/(\boldsymbol{Q}[1]+\boldsymbol{Q}[2]+\cdots+\boldsymbol{Q}[K])$; /* 计算跨链交易
33　预期共识成功率 */
34　/* 优化求解 */
35　$\mathbb{C}\leftarrow$ Call $NSGA-II$ ($\mathbb{Y}$, min(S_1), min(S_2), max(S_3), n, t, a) ; /* 基于式(19)调用NSGA-II算法进行求解 */
36　Return $\mathbb{C}$;

(3)算法1生成$l\times N$的随机排列数组，元素为1～K的唯一编号，表示交易包在不同共识轮次中与异步共识节点间的匹配关系。此过程确保每个交易包仅被分配至唯一节点(行13～14)。

(4)该算法计算K个交易包的平均预期广播时间S_1、平均每轮异步共识的预期交易广播执行时间S_2、K个交易包的预期共识成功率S_3，以便在后续的优化过程中进行调整(行15～37)。其中，S_1、S_2、S_3表示设置的目标变量。数组U用于存放K个交易包分别在l轮异步共识中交易广播的预期执行时间。变量T用于计算预期花费总时间，变量F用于计算预期共识成功总交易数。r用于记录每一轮的最大时间。x、y、s分别表示交易包索引、节点处理交易包的预期时间、预期共识成功交易数量。算法1调用NSGA-II算法，实现最小化S_1、S_2、最大化S_3，以求得最优跨链交易匹配数组$\mathbb{C}$，并返回该数组(行38～41)。

4.2　异步共识优化算法

本文基于第3节的模型，设计异步共识优化算法，如算法2所示。

算法2： 异步共识优化算法(以节点i为例)

输入参数：$\mathbb{C}$, r, S, X, N, f;

输出参数：T;

1　string $\boldsymbol{W}[N]=$ NULL, string $T[]=$ NULL;
2　int $x=0$, $z=0$;

```
3    P_i ← < C[r][i], F_i >; /* 节点i从其交易池中提取编号为C[r][i]的交易包并签名作为共识提案 */While 节点i获取共识提案P_i
4      Call PB(P_i); /* 将P_i输入到PB_i实例中 */
5      If PB(P_i)输出σ_i /* σ_i是来自PB_i的阈值签名 */
6        节点i广播σ_i;
7      EndIf
8    EndWhile
9    While 节点x_i收到节点j的σ_j
10     If σ_j验证有效
11       W[j] ← σ_j; /* 添加σ_j至节点i的提议向量 */
12       x++;
13     EndIf
14   EndWhile
15   int y = 0, d = 0;
16   If x == f+1
17     CBC(W); /* 节点i广播提议向量W */
18   EndIf
19   While y ≤ f+1 && d = 0
20     d = 1;
21     z ← 从集合S中随机选取1个节点;
22     V_z ← 获取节点z的提议向量;
23     Call ABA(V_z); /* 节点i针对提议向量V_z对
24       ABA实例进行输入 */
25     If ABA(V_z)实例输出1
26       T ← V_z; /* 提取向量V_z中的交易 */
27       If(W != T) /*节点i检查是否收到数组T中全
28         部交易 */
29         节点i请求其他节点进行交易恢复;
30     Else g ←重新从集合S中随机选取不包括z的其余节点;
31       V_g ← 获取节点g的提议向量;
32       Call ABA(V_g); /* 重新输入*/
33       y++;
34       If y > f+1
35         d = 0;
36       EndIf
37     EndIf
38   EndWhile
39   Return T;
```

算法2的相关解释如下：

(1)该算法以节点i为例。其中，输入参数中，$\mathbb{C}$表示算法1中输出的最优跨链交易匹配数组；r表示当前异步共识轮次；S表示最佳节点集合(依据算法3进行动态更新)；X表示异步共识节点集合；N表示集合X中节点的数量；f表示恶意节点数量。输出参数中的向量T表示达成共识的交易数组。一维数组$\boldsymbol{W}$表示提议向量，包括对应节点的阈值签名。变量x表示节点i收到其余节点签名的次数，变量z表示随机选取的节点编号(行1～2)。

(2)交易广播模块中，节点i形成共识提案并调用PB实例进行广播。交易广播模块完成后，节点i得到提议向量$\boldsymbol{W}$(行3～16)。

(3)多值拜占庭共识模块中，节点i调用CBC实例进行提议向量$\boldsymbol{W}$的广播，并基于随机选取的节点的提议向量对ABA实例进行输入(行17～26)。其中，变量y表示随机选择节点的次数，变量d表示ABA实例成功与否。

(4)若ABA实例输出1，算法则进入交易恢复模块，节点i对比交易数组T中的交易检查是否存在交易缺失。若存在，则需进行交易恢复(行27～31)。否则，算法2则从集合S中重新选择节点，以执行ABA实例。当重新选择节点的次数超过$f+1$，共识失败(行32～41)。

4.3 OPVS算法

本文设计OPVS算法(如算法3所示)，通过评估l轮异步共识中、集合X中各个异步共识节点的性能参数，获得最佳节点集合S，以优化异步共识流程中提议向量的选择并加速异步共识进程。算法3的相关解释如下。

(1)该算法输入参数中的X表示异步共识节点集合；N表示异步共识节点数量；l为评估的异步共识轮数；f为恶意节点数量；二维数组A为集合X中、节点完成l轮异步共识的对应多值拜占庭共识时延数组；一维数组B为集合X中、节点在l轮异步共识中向ABA实例中输入正确值次数的对应数组；k为集合S中元素的个数，满足$f+1 \leqslant k < N$；n为种群大小，t为交叉率，a为迭代次数。算法输出参数S为最佳节点集合。

(2)该算法首先声明并初始化二维通信延迟数组D与节点数组$\mathbb{Y}$(行1)。其次，该算法计算异步共识节点间通信延迟并赋值至数组D(行2～6)，接着计算每个节点的运行性能值与共识准确率(行7～14)。

(3)该算法在集合X中随机选取节点组成集合，以生成数组$\mathbb{Y}$(行15)。该算法设置3个目标变量(行16)，并调用NSGA-Ⅱ算法进行多目标优化求解，最终返回满足条件的集合S(行17～31)。其中，变量b表示集合S中节点到其余节点的平均通信延迟，变量d表示集合S中节点的平均运行性能值，变量w表示集合S节点的平均共识准确率。

算法3： OPVS算法

输入参数：X, N, l, f, A, B, k, n, t, a；

输出参数：S；

```
double D[][] = NULL; int 𝕐[k]= NULL;
For (int i = 1; i ≤ N; i ++)
  For (int j = 1; j ≤ N; j ++)
    D[i][j] ← d_ij; /* d_ij通过式(21)计算 */
  EndFor
EndFor
For (int i = 1; i ≤ N; i ++)
  double ξ_i = 0, φ_i = 0;
  For (int j = 1; j ≤ l; j ++)
    double κ_i^j = A[i][j];
  EndFor
  ξ_i ← l/(∑κ_i^j); /* ξ_i通过式(22)计算 */
  φ_i ← η_i/l; /* φ_i通过式(23)计算,η_i = B[i] */
EndFor
𝕐← 从集合X中随机选取k个元素;
double b = 0, d = 0, w = 0;
For (int i = 1; i ≤ k; i++ )
  int c = 𝕐[i];
  double J̄ = 0, φ̄ = 0, D̄ = 0;
  For (int j = 1; j ≤ N; j++)
    D̄ ← D̄ + D[c][j];
  EndFor
  J̄ ← J̄ + ξ_c;
  φ̄ ← φ̄ + φ_c;
EndFor
b ← D̄/(k(N − 1));
d ← J̄/k;
w ← φ̄/k;
S ← Call NSGA-II (𝕐, min(b), max(d), max(w), n, t, a); /* 基于式(27)调用NSGA-II
算法求解 */
Return S;
```

5 实　验

5.1 实验环境

本文在Ubuntu 20.04 LTS操作系统中，利用PyCharm平台开展中继链异步共识环境的原型仿真，并通过实验评估OCAC的性能。Python语言提供了强大的调试工具和测试框架，能够方便地对仿真代码进行调试和验证，以确保仿真结果的准确性。本文提出的OCAC利用gevent库处理大量并发交易，并结合基于配对的门限签名技术与混合门限加密技术，实现门限签名和门限加密功能。此外，本文使用开源进化算法工具包Pymoo中所提供的NSGA-Ⅱ算法解决多目标优化问题。NSGA-Ⅱ算法作为一种流行的多目标遗传算法，采用循环拥挤排序策略来改进传统的NSGA算法，简化了非劣排序遗传算法的复杂性，表现出快速运行和良好解集收敛性的优点，已成为其他多目标优化算法性能的标杆[29]。因此，本文通过调用NSGA-Ⅱ算法对多目标优化问题进行求解。Pymoo工具包提供了简洁易用的接口，使得利用NSGA-Ⅱ算法解决实际的多目标优化问题变得更加便捷。该工具包在大量实验中得到了广泛应用[30,31]。本文实验数据集采用XBlock-EOS中的区块链数据集[32]，其来自EOSIO平台的真实链上数据，是第一个提供最全面的链上经过良好处理的数据集。该数据集包含EOS区块链上各种活动的详细数据，包括约8983万条区块记录、11.28亿条代币转移记录、每年数千万到数亿条智能合约调用记录，以及超过164万条账户信息[33]。文献[34-36]均使用此数据集进行实验，由此可见该数据集具有良好的适用性。

5.2 对比方法与时间复杂度分析

5.2.1 对比方法

本文将所提出的OCAC在中继链环境中与Dumbo[7]、sDumbo[9]、Chronos[16]、TortoiseBFT[17]、Dumbo-NG[10]作对比，以有效评估OCAC，其中：

(1)Dumbo是第一个适用于区块链环境的异步共识机制，其在相关研究中广泛用作基准方法[9,10]。

(2)sDumbo的核心在于通过采用成本更低的广播组件替代RBC，并在此基础上实现了更高效的多值拜占庭共识协议。与Dumbo相比，该方法显著减少了将近一半的延迟，并实现吞吐量翻倍。

(3)Chronos提出了信号异步公共子集协议的ACS变体并基于RBC与信号异步公共子集协议构建拜占庭有序共识协议。

(4)TortoiseBFT实现了一个具有3阶段异步共识机制，通过确定交易顺序并请求丢失的交易，显著降低了通信开销。

(5)Dumbo-NG设计一种能够确保输出来自诚实节点概率至少为1/2的多值拜占庭共识协议，并支持交易广播与多值拜占庭共识协议的并行执行。

本文提出的OCAC方法在中继链环境中分别通过OCTM算法、OPVS算法优化交易广播模块、

多值拜占庭共识模块，显著提升了交易吞吐量并降低了交易时延。

5.2.2 时间复杂度分析

(1)Dumbo[7]通过优化异步BFT协议的核心组件ACS，显著降低了时间复杂度，其采用多值拜占庭共识协议将ABA实例数量降至常数，实现了$O(1)$的轮次复杂度以及$O(n^2)$的计算复杂度。Dumbo的通信复杂度为$O(n^2|m|+\lambda n^3\log n)$。其中，$n$表示节点数量，$|m|$表示每个节点输入的比特长度，$\lambda$为安全参数的大小。

(2)sDumbo[9]通过引入高效的PB组件和优化的多值拜占庭共识协议，显著降低了异步BFT共识的时间复杂度。具体而言，其通过精简协议流程等方法，降低了多值拜占庭共识轮数，使其轮次复杂度为$O(1)$。PB将每个广播实例的消息复杂度从$O(n^2)$降至$O(n)$。因此，其计算复杂度为$O(n^2)$、通信复杂度为$O(n^2|m|+\lambda n^3\log n)$。

(3)Chronos[16]通过引入信号异步公共子集优化了异步拜占庭有序共识的时间效率。由于其结构中仍存在n个ABA实例，其轮次复杂度为$O(\log n)$。其阈值签名和验证操作导致其总体计算复杂度趋近$O(n^3)$。Chronos通过验证谓词和信号机制减少冗余验证步骤，将通信复杂度控制在$O(n^2|m|+\lambda n^3)$。

(4)TortoiseBFT[17]先确定交易的顺序，然后请求缺失的交易。由于该方法的ABA实例只有1个，其轮次复杂度为$O(1)$。其阈值签名验证从传统协议的$O(n^3)$优化至$O(n^2)$。因此，TortoiseBFT总体计算复杂度为$O(n^2)$、通信复杂度为$O(n^2|m|+\lambda n^3\log n)$。

(5)Dumbo-NG[14]通过完全并行化交易广播和多值拜占庭共识协议，将轮次复杂度降至期望常数轮次$O(1)$。该协议由于采用了轻量级阈值签名，其计算复杂度为$O(n+|m|)$。该协议将多个并发的ABA实例替换为单一的多值拜占庭共识实例，其通信复杂度在渐近意义上为$O(n^2|m|+\lambda n^3\log n)$。

本文提出的OCAC方法在交易广播模块中使用PB组件代替RBC组件，降低了异步广播延迟和消息冗余；在多值拜占庭共识模块中，对经过筛选的高性能节点的提议向量执行ABA实例。其并发的PB实例只需常数轮数，且多值拜占庭共识模块的运行时间也是常数。因此，OCAC轮次复杂度为$O(1)$、计算复杂度为$O(n^2)$、通信复杂度为$O(n^2|m|+\lambda n^3\log n)$。

5.3 实验参数设置

实验参数设置如表1所示。为评估多目标优化算法的收敛性并确定合适的参数，本文将变异率设置为决策变量数量的倒数，并设置交叉率分别为0.3、0.5、0.7、0.9，种群规模分别为50、100、150、200。由于OCAC需要在异步网络环境下进行，异步共识节点之间的通信延迟设置为[100，1000] ms之间的随机值[17]。为评估异步共识机制在大规模交易下的性能，本文参照文献[9,16,17]，将中继链交易池中批量交易数量设置为1000、5000、10 000、50 000、100 000。其中，批量交易是指所有异步共识节点需要处理的跨链交易总数。由于本文使用的是联盟中继链，因此异步共识节点的总数不需要设置过多。同时，为便于比较各异步共识机制在中继链环境下的性能，本文参照文献[7,9,10,16,17]，将异步共识节点的数量分别设置为5、10、15、20、…、115、120，以进行性能评估。为模拟异步共识过程中不同性能的共识节点，本文将交易验证时延设置为在[2，500] ms之间的随机值[7]。同时，本文将中继链的交易大小设置为满足[100，250] B的均匀分布[29]。为评估不同恶意节点比例对异步共识机制的影响，本文将恶意节点比例分别设置为0.1、0.15、0.2、0.25、0.3。恶意节点数量为f，满足$N=3f+1$的条件。本文利用文献[37]对OCAC机制中的共识节点进行信誉评估，基于节点历史交易处理成功率、响应延迟等多维指标动态更新信誉值，实现对集合S中低信誉节点的自动识别与剔除。具体为：对于历史交易处理成功率高、响应延迟低的节点，其信誉值会相应提升；反之，若节点频繁出现交易处理失败或响应延迟过长的情况，信誉值则会降低。在每一轮交易处理完成后，系统会统计节点处理成功的交易数量与总交易数量，作为交易处理成功率的

表1 实验参数设置表

名称	含义	值
A	交叉率	0.3, 0.5, 0.7, 0.9
C	种群规模大小	50, 100, 150, 200
d	通信延迟	[100, 1000] ms
B	中继链交易池中批量交易的数量	1000, 5000, 10 000, 50 000, 100 000
N	中继链异步共识委员会节点数量	5, 10, 15, 20, …, 115, 120
v	交易验证时延	[2, 500] ms
q	交易大小	[100, 250] B
y	恶意节点比例	0.1, 0.15, 0.2, 0.25, 0.3
ΔI	评估周期	3
l	预期共识执行轮数	3

计算依据。同时，记录节点从接收交易请求到给出处理结果的时间间隔，作为响应延迟的衡量指标。根据这些指标对节点的信誉值进行更新，动态剔除集合S中信誉值低的节点，从而降低恶意节点或性能不佳节点对异步共识过程的干扰，以提升系统安全性和效率。为最大程度提高跨链异步共识机制的效率与可靠性，本文将评估周期为较小值3，即一个评估周期包含3轮异步共识。同时，为便于与其他异步共识机制进行对比，本文将OCTM算法中的预期共识执行轮数设置为3。为提高实验结果准确性，本文对100次实验所得数据采用算术平均处理。

5.4　实验指标

选取交易时延、交易吞吐量、超体积作为实验指标，如下。

(1)交易时延

令交易时延记为$\mathcal{C}_d$。该时延表示中继链中d笔跨链交易从开始到完成共识的时间间隔，如式(28)所示：

$$\mathcal{C}_d=\mathcal{M}_d-\mathcal{N}_d \tag{28}$$

其中，$\mathcal{M}_d$表示d笔跨链交易完成异步共识的时间，$\mathcal{N}_d$表示d笔跨链交易开始共识的时间。

(2)交易吞吐量

令交易吞吐量记为$\mathcal{D}$。该吞吐量表示异步共识委员会每秒钟完成共识的不重复的跨链交易数量，如式(29)所示：

$$\mathcal{D}=\frac{\mathcal{J}}{\mathcal{H}} \tag{29}$$

其中，$\mathcal{J}$表示中继链处理完成的批量跨链交易总数，$\mathcal{H}$表示批量跨链交易完成共识所花费的时间。

(3)超体积

超体积(Hypervolume，HV)是多目标优化领域广泛采用的性能评价指标，能够同时刻画解集的收敛性与分布性[37]。其定义为目标空间内解集中各个个体与预先设定的参考点所共同围成区域的面积或体积，能够综合反映解集的收敛性和多样性。给定解集$\mathbb{S}=\{z_1, z_2, \cdots, z_m\}$与参考点$r\in\mathbb{R}^n$，超体积定义为解集$\mathbb{S}$中所有点与$r$形成的非支配区域的体积，如式(30)所示：

$$\mathrm{HV}(\mathbb{S}; r)=\lambda(\bigcup_{z\in\mathbb{S}}[z, r]) \tag{30}$$

其中，$\lambda(\cdot)$表示n维空间的勒贝格测度(即体积)，$[z, r]$表示由解z与参考点r张成的超立方体，即目标空间中满足$\forall i: z_i\leqslant x_i\leqslant r_i$的区域且参考点$r$需要被所有解支配(即$\forall z\in\mathbb{S}, z\prec r$)。

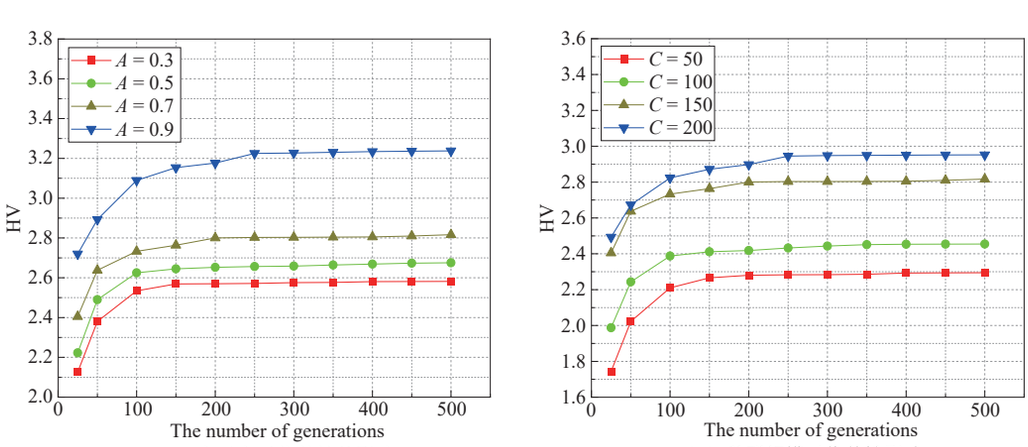
5.5　实验分析

5.5.1　基于不同实验参数的对比实验

(1)收敛性

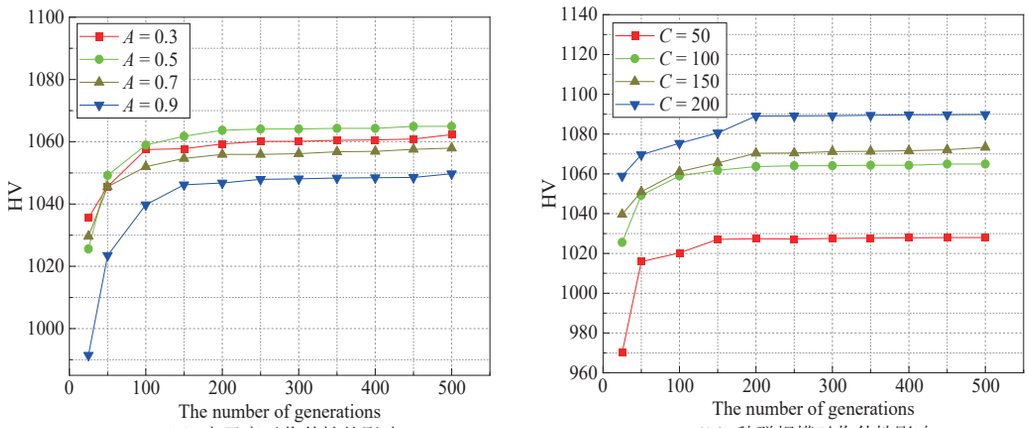
为评估NSGA-II算法的收敛性并确定合适的交叉率与种群规模，本文选择HV作为实验指标，以分析OCTM算法与OPVS算法中种群规模和交叉率对收敛性的影响，如图5~6所示。随着迭代次数的增加，图5~6中的HV值均逐渐趋于平稳，表明解集不再显著改进，具有良好的收敛性。

图5(a)中，当交叉率分别为0.3与0.5时，收敛速度较快，仅需要150次迭代。但由于HV值较小，解集覆盖的空间范围有限，解的分布可能过于集中或不均匀。而当交叉率为0.7与0.9时，HV值较高，收敛速度较慢，分别需要200次与250次迭代。因此，为平衡收敛速度与解的质量，OCTM算法选择0.7作为交叉率。图5(b)中，HV值随着种群规模的增大而增大。当种群规模为较大值150与200时，收敛速度较慢，分别需要200、250次迭代，但HV值较高。其原因在于种群规模扩大在提升解的质量



(a) 交叉率对收敛性的影响　　(b) 种群规模对收敛性影响

图5　OCTM算法中交叉率与种群大小对收敛性的影响



（a）交叉率对收敛性的影响 （b）种群规模对收敛性影响

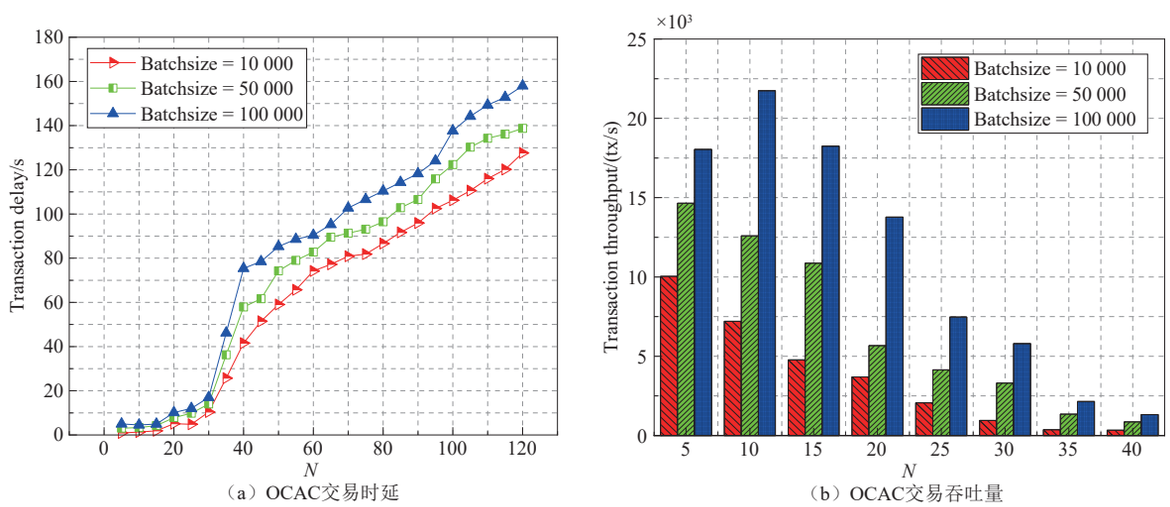
图6 OPVS算法中交叉率与种群大小对收敛性的影响

的同时，牺牲了收敛效率。因此，OCTM算法选择150作为种群规模大小。

图6(a)中，当交叉率为较高值0.9时，HV值最低，原因在于在该算法中高交叉率在增强探索性的同时，削弱了对优秀解的继承与精细改进，使得种群既难以积累高质量解，也难以快速逼近或覆盖帕累托前沿，HV值反而最低。而当交叉率分别为0.3、0.5、0.7时，HV值均较高，且收敛速度较快。其中，当交叉率为0.5时，HV值最高。因此，OPVS算法选择0.5作为交叉率。图6(b)中，当种群规模为150、200时，收敛速度较慢，均需要200次迭代，但HV值较高。因此，OPVS算法选择200作为种群规模大小。

(2)不同节点数量下的交易时延与吞吐量

OCAC在不同节点数量与批量交易数量下的交易时延与吞吐量如图7所示。中继链批量交易数量设置为10 000、50 000、100 000分别进行3组实验。实验从中继链交易池中的批量跨链交易中随机生成$l \times N$个交易包，并利用OCTM算法与OPVS算法执行优化的跨链共识机制。



（a）OCAC交易时延 （b）OCAC交易吞吐量

图7 OCAC的交易时延与交易吞吐量

图7(a)中，当异步共识节点数量分别设置为5、10、15、20、…、115、120，且中继链交易池中的批量交易数量设置为10 000、50 000、100 000时，交易时延低于160 s。然而，当N超过30时，随着节点数量的增加，不同批量交易数量下的异步共识时延均显著呈现上升趋势。其原因在于当异步共识节点数量越多时，需要等待更多节点完成任务，整体时延增加。图7(b)中，当节点数量超过10，交易吞吐量均随着节点数量的增加而减小。当批量交易数量为100 000，且节点数量为10时，交易吞吐量达到了最

大值2.17×10⁴ tx/s。

（3）OCTM与OPVS前后交易时延对比

为有效评估OCTM算法与OPVS算法在中继链中对异步共识机制的作用，本文在$N=10$、$f=3$、B分别为10 000、50 000、100 000的条件下对周期为1～9进行评估。其中，周期1～3的异步共识中、将批量交易平均分配给不同共识轮次中的各异步共识节点，并在多值拜占庭共识中随机选取提议向量。在周期4～9中，采用OCTM算法与OPVS算法，实现最优跨链交易匹配与提议向量选择优化。如图8所示，周期4～9的异步共识机制在交易时延方面均比前3个周期有所降低，由此可见OCAC的有效性。

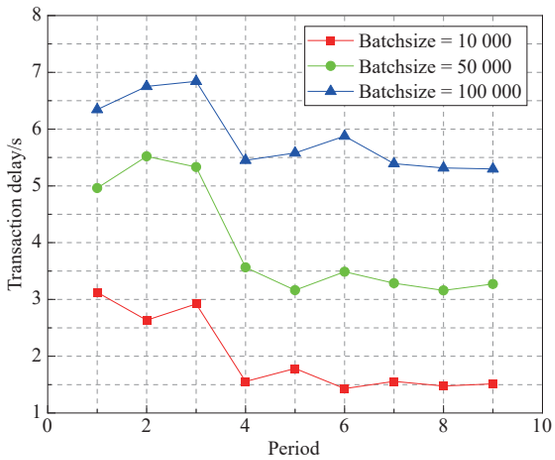


图8　OCTM与OPVS前后交易时延对比（$N=10$、$f=3$）

5.5.2　与其他方法的对比实验

（1）交易吞吐量与交易时延对比

图9～10展示了中继链中不同异步共识机制的交易吞吐量与时延随批量交易数量变化趋势。N取值分别为5、15、30、60；批量交易数量依次设置为1000、5000、10 000、50 000、100 000。

图9中，无论异步共识节点数量或批量交易数量如何变化，OCAC的交易吞吐量始终优于Chronos、Dumbo、TortoiseBFT、sDumbo、Dumbo-NG。尤其当批量交易数量为100 000时，OCAC优势更明显。当N为15、30且B为100 000时，OCAC的交易吞吐量是sDumbo的两倍多。这表明OCAC通过OCTM算法与OPVS算法能够在相同时间内

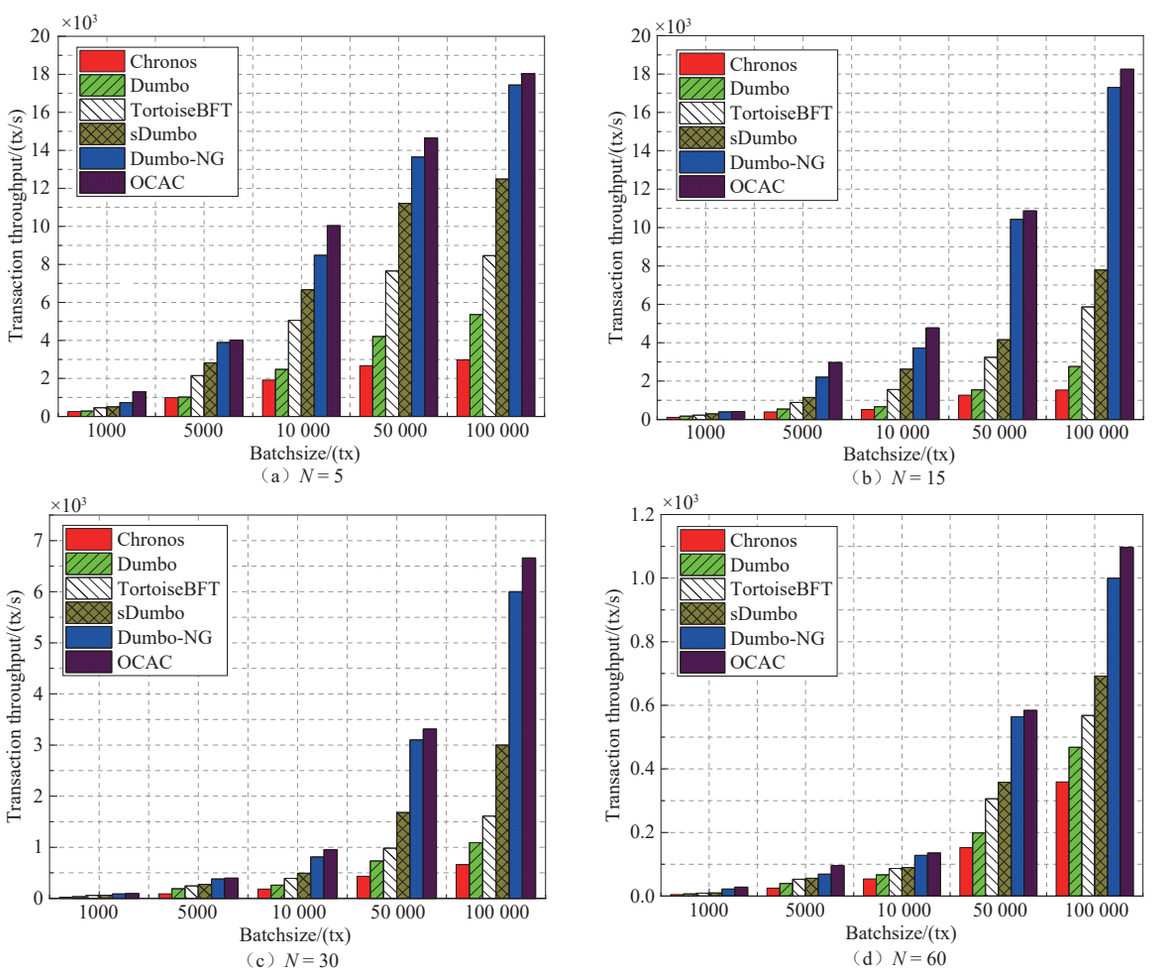


图9　不同异步共识机制在不同节点数量时交易吞吐量对比

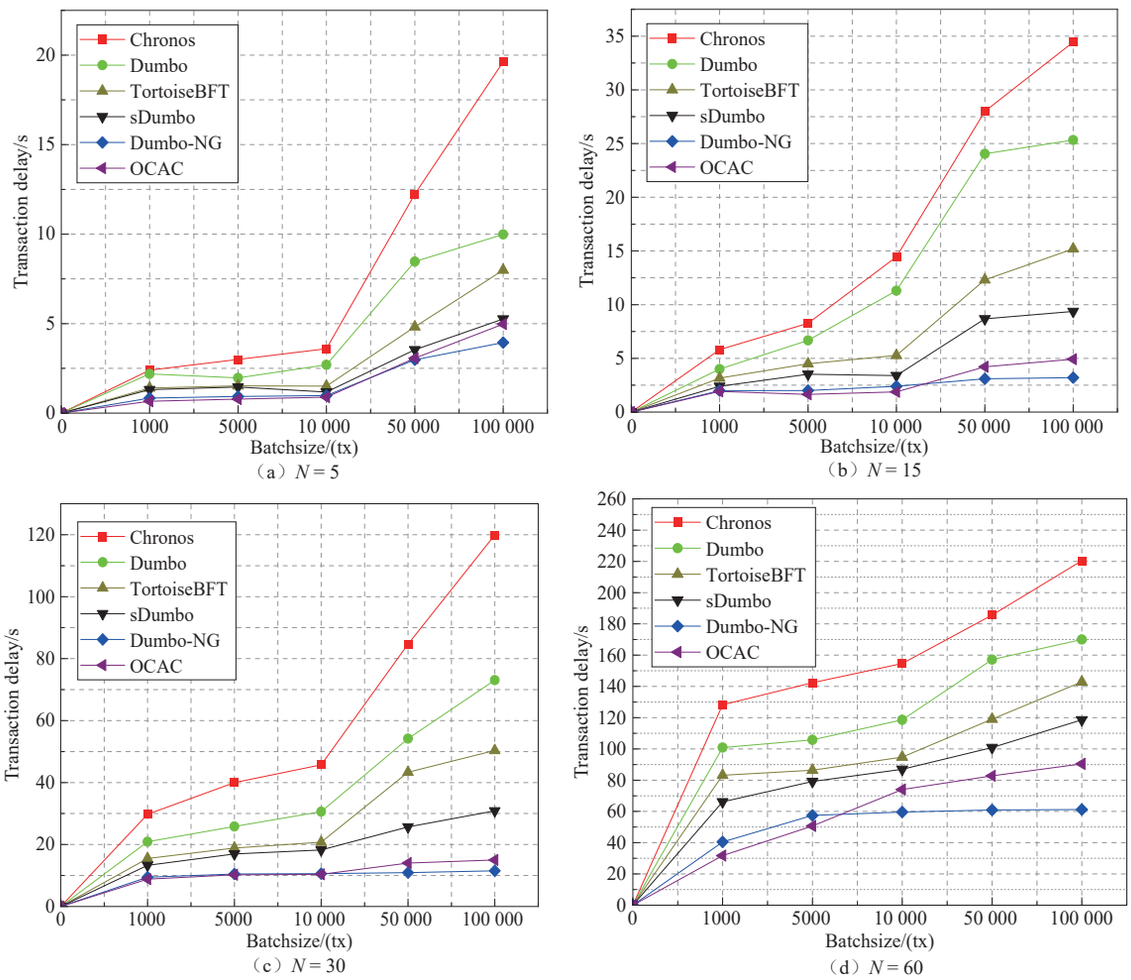


图10 不同异步共识机制在不同节点数量时交易时延对比

处理更多交易。其原因在于所对比的异步共识机制在实际运行中，交易全部转发至所有异步共识节点的交易池；在不同共识轮次中由各个异步共识节点随机从各自缓冲区中选取批量交易进行异步共识，存在不同节点选取相同交易进行共识的现象，同时忽略了节点间的性能差异性。由此可见，在实际中继链环境中，考虑节点的实际性能并避免诚实的异步共识节点对相同跨链交易进行共识，对于提升交易吞吐量至关重要。图10中，无论中继链中异步共识节点数量或批量交易数量如何变化，OCAC的交易时延均比Chronos、Dumbo、TortoiseBFT、sDumbo低。而在不同规模的批量交易数量下，OCAC与Dumbo-NG各展优势：前者在批量交易数量较小的场景中具备更低的交易时延，而后者则通过并行执行交易广播与多值拜占庭共识协议，降低了批量交易数量较大时的交易时延。

（2）多值拜占庭共识时延对比

由于多值拜占庭共识模块在异步共识机制中主要的作用是对较小的提议向量达成一致，本文在$N=3f+1$的条件下将异步共识机制中节点的输入设置为一笔交易，以对比多值拜占庭共识模块的时延[9]。表2展示了不同异步共识机制在不同节点数量下的多值拜占庭共识时延。

表2中不同异步共识机制的多值拜占庭共识时延均随节点数量的增加而增加。然而，Chronos的多值拜占庭共识时延与其余异步共识机制相比显著增加，原因在于其多值拜占庭共识模块仍然需要N个ABA实例，需要进行大量阈值签名验证，拖慢了异步共识进程。而Dumbo、TortoiseBFT、sDumbo、Dumbo-NG、OCAC都将ABA实例的数量减少到1。因此，与Chronos相比，Dumbo、TortoiseBFT、sDumbo、Dumbo-NG在不同节点数量下始终维持较低的多值拜占庭共识时延。其中，Dumbo-NG使用具有质量保证特性的多值拜占庭共识协议，使其输出来自诚实节点的概率至少提高到50%。因此，其多值拜占庭共识时延较低。而OCAC在多值拜

表2　不同异步共识机制在不同节点数量下的多值拜占庭共识时延　　(s)($N=3f+1$)

Node Count	Chronos	Dumbo	Tortoise BFT	sDumbo	Dumbo-NG	OCAC
5	1.21	0.72	0.58	0.56	0.27	**0.13**
10	3.65	2.16	1.87	1.26	1.14	**0.52**
15	5.25	3.88	3.02	1.82	1.53	**1.05**
20	8.25	5.03	3.96	2.47	2.04	**1.29**
25	11.25	6.62	5.42	2.51	2.27	**1.54**
30	15.13	8.64	7.21	2.96	2.46	**1.81**
35	18.32	10.81	8.64	3.51	3.14	**1.89**
40	22.55	13.54	10.81	4.09	3.78	**2.36**

占庭共识中通过OPVS算法优化提议向量的选择，其多值拜占庭共识时延在节点数量为5～40的范围内始终低于3 s。

(3)不同恶意节点比例、不同批量交易数量下异步共识机制的交易吞吐量对比

图11展示了不同异步共识机制在$N=30$条件下，不同批量交易数量下($B=50\,000$、$100\,000$)交易吞吐量随恶意节点比例变化(0.1、0.15、0.2、0.25、0.3)而产生的变化。由图11可知，OPVS能够有效降低恶意节点的影响，具有较好的鲁棒性，使得OCAC在不同恶意节点比例的情况下，整体吞吐量性能仍保持领先优势。

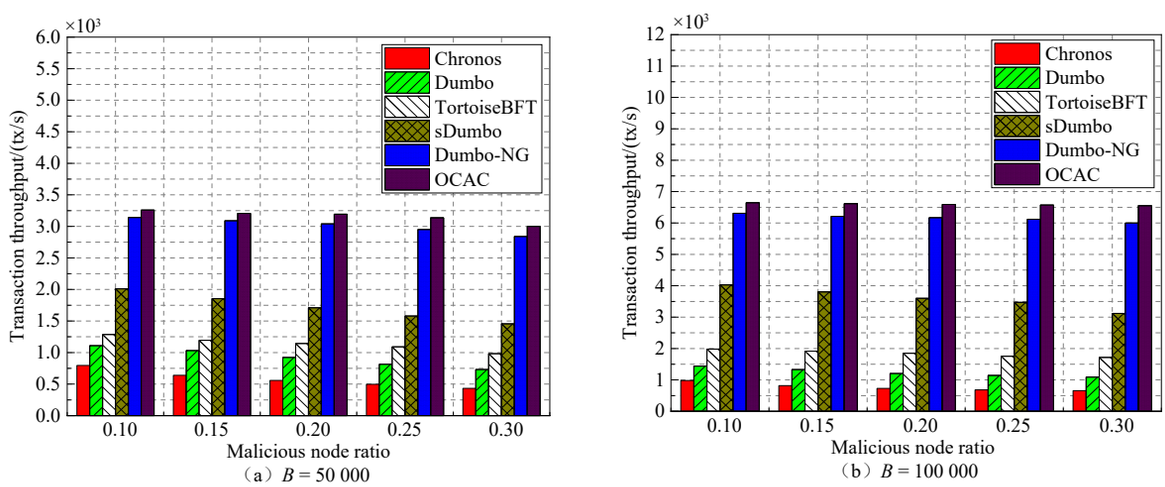


图11　不同恶意节点比例、不同批量交易数量下的异步共识机制交易吞吐量的对比

表3展示了基于图11的不同异步共识机制交易吞吐量统计结果，包括均值、方差、标准差。在不同恶意节点比例下，OCAC交易吞吐量的均值在$B=50\,000$、$100\,000$下均高于其余异步共识机制。同时，TortoiseBFT、Dumbo-NG、OCAC在$B=50\,000$、$100\,000$下的方差与标准差显著小于其余异步共识机制。其原因在于TortoiseBFT利用信誉机制对异步共识节点进行评估，能够有效降低恶意节点的影响；Dumbo-NG引入的多值拜占庭共识协议具有质量保证。而OCAC在交易广播模块中考虑异步共识节点的共识成功率，并在多值拜占庭共识模块利用OPVS算法实现了提议向量的选择优化，能够有效降低恶意节点的影响。因此，TortoiseBFT、Dumbo-NG、OCAC受恶意节点影响较小。相比之下，Chronos、Dumbo、sDumbo异步共识机制交易吞吐量的方差与标准差较大，容易因恶意节点比例的大小而波动。

实验分析总结如下。

(1)收敛性

本文通过HV指标以评估NSGA-II算法在本应用场景下的收敛性，由图5～6，可知本文所提出的OCTM算法与OPVS算法具有良好的收敛性。

(2)不同节点数量下的交易时延与交易吞吐量

由图7可知，随着共识节点数量的增大，OCAC的交易时延整体不断增大。当批量交易数量为100 000，且节点数量为10时，OCAC的交易吞吐量达到了最大值。由此可见，OCAC性能较高。

(3)OCTM与OPVS前后交易时延对比分析

表3 不同异步共识机制吞吐量统计结果(基于图11)

Name	基于图11(a)数据			基于图11(b)数据		
	均值 ($\times10^3$ tx/s)	方差 ($\times10^4$ tx/s)	标准差 ($\times10^2$ tx/s)	均值 ($\times10^3$ tx/s)	方差 ($\times10^4$ tx/s)	标准差 ($\times10^2$ tx/s)
Chronos	0.58	1.60	1.26	0.77	1.37	1.17
Dumbo	0.92	1.91	1.38	1.24	1.6	1.26
TortoiseBFT	1.14	1.03	1.01	1.84	0.96	0.98
sDumbo	1.72	3.86	1.96	3.6	9.43	3.07
Dumbo-NG	3.03	1.13	1.06	6.16	1.05	1.02
OCAC	**3.18**	**0.81**	**0.9**	**6.59**	**0.11**	**0.33**

由图8可知,本文所提出的OCTM算法与OPVS算法可以明显降低异步共识的交易时延,并使其趋于稳定。

(4)交易吞吐量与交易时延对比分析

由图9~10可知,当异步共识节点数量为5、15、30、60时,随着批量交易数量的增大,OCAC交易吞吐量不断增加,并且始终高于Chronos、Dumbo、TortoiseBFT、sDumbo、Dumbo-NG。同时,OCAC的交易时延始终低于sDumbo,在较小批量交易数量下低于Dumbo-NG。

(5)多值拜占庭共识时延对比分析

由表2可知,OCAC的多值拜占庭共识时延在不同节点数量下始终低于Chronos、Dumbo、TortoiseBFT、sDumbo、Dumbo-NG。

(6)不同恶意节点比例、不同批量交易数量下的异步共识机制交易吞吐量的对比分析

由图11与表3的统计分析可知,OCAC相较于Chronos、Dumbo、sDumbo、Dumbo-NG,受恶意节点影响较小。

综上,就收敛性、交易时延、交易吞吐量而言,本文所提出的OCAC在整体性能方面表现较优。

6 总 结

目前,单个区块链的可扩展性和互操作性限制了区块链生态系统的快速发展,而跨链技术的出现为研究区块链可扩展性开辟了新的途径[38]。本文通过设计基于最优跨链交易匹配的交易广播模块与基于提议向量选择优化的多值拜占庭共识模块,在中继链中实现了优化的跨链异步共识机制。同时,本文通过仿真实验有效评估了所提出跨链异步共识机制的基本性能,相比于传统异步共识机制,本文所提出的跨链异步共识机制能够在中继链中实现更低的交易时延与更高的交易吞吐量。未来研究可结合分片技术将中继链系统分为多个分片,每个分片独立运行异步共识机制,以进一步提升交易并发处理能力。

参 考 文 献

[1] Shen M, Che Z, Zhu L H, Xu G, Gao F, Yu C C, Wu Y. Anonymity in blockchain digital currency transactions: Protection and countermeasures. Chinese Journal of Computers, 2023, 46(1): 125-146 (in Chinese)
(沈蒙,车征,祝烈煌,徐恪,高峰,余聪聪,吴言.区块链数字货币交易的匿名性:保护与对抗.计算机学报,2023,46(1):125-146)

[2] Maesa D, Mori P. Blockchain 3.0 applications survey. Journal of Parallel and Distributed Computing, 2022, 138: 99-114

[3] Ou W, Huang S Y, Zheng J J, Zhang Q L, Zeng G, Han W B. An overview on cross-chain: mechanism, platforms, challenges and advances. Computer Networks, 2022, doi: 10.1016/j.comnet.2022.109378

[4] Lys L, Micoulet A, Potop-Butucaru M. Atomic cross chain swaps via relays and adapters//Proceedings of the 3rd Workshop on Cryptocurrencies and Blockchains for Distributed Systems. Toronto, Canada, 2020: 59-64

[5] Tong X, Zhang Z, Jin C Q, Zhou A Y. A survey of blockchain technology for edge-cloud collaborative architecture. Chinese Journal of Computers, 2021, 44(12):2345-2366 (in Chinese)
(佟兴,张召,金澈清,周傲英.面向端边云协同架构的区块链技术综述.计算机学报,2021,44(12):2345-2366)

[6] Miller A, Xia Y, Croman K, Shi E, Song D. The honey badger of BFT protocols//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 31-42

[7] Guo B Y, Lu Z L, Tang Q, Xu J, Zhang Z F. Dumbo: Faster asynchronous BFT protocols//Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security. Virtual, 2020: 803-818

[8] Tandon R, Verma A, Gupta P K. D-BLAC: A dual blockchain-based decentralized architecture for authentication and communication in VANET. Expert Systems with Applications, 2024, 237: doi: 10.1016/j.eswa.2023.121461

[9] Guo B Y, Lu Y, Lu Z L, Tang Q, Xu J, Zhang Z F. Speeding Dumbo: Pushing asynchronous BFT closer to practice//

Proceedings of the 2022 network and distributed system security. San Diego, USA, 2022: 1-15

[10] Gao Y Z, Lu Y L, Lu Z L, Tang Q, Xu J, Zhang Z F. Dumbo-NG: Fast asynchronous BFT consensus with throughput-oblivious latency//Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. Los Angeles, USA, 2022: 1187-1201

[11] Fu X, Wang H, Shi P C. A survey of blockchain consensus algorithms: mechanism, design and applications. Science China Information Sciences, 2020, 64: 1-15

[12] Duan S, Reiter M K, Zhang H. BEAT: Asynchronous BFT made practical//ACM SIGSAC Conference on Computer and Communications Security, 2018: 2028-2041

[13] Liu S Y, Xu W B, Shan C, Yan X F, Xu T J, Wang B, Fan L, Deng F X, Yan Y, Zhang H. Flexible advancement in asynchronous BFT consensus//Proceedings of the 29th Symposium on Operating Systems Principles. Koblenz, Germany, 2023: 264-280

[14] Duan S S, Wang X, Zhang H B. Fin: Practical signature-free asynchronous common subset in constant time//Proceedings of the 2023 ACM SIGSAC Conference on Computer and Communications Security. Copenhagen, Denmark, 2023: 815-829

[15] Zhang H B, Duan S S, Zhao B X, Zhu L H. WaterBear: Practical asynchronous BFT matching security guarantees of partially synchronous BFT//Proceedings of the 32nd USENIX Security Symposium (USENIX Security 23. ClaraSanta, USA, 2023: 5341-5357

[16] Zhang Z Y, Zhang L Y, Wang Z, Li Y X, Lu R X, Yu Y. Chronos: An efficient asynchronous byzantine ordered consensus. The Computer Journal, 2024, 67(3): 1153-1162

[17] Liu Y, Fu J H, Zhang M M, Shi S D, Chen J W, Peng S, Wang Y Q. TortoiseBFT: An asynchronous consensus algorithm for IoT system. Journal of King Saud University-Computer and Information Sciences, 2024, 36(6), doi: 10.1016/j.jksuci.2024.102104

[18] Antunes D, Oliveira A, Breda A, Franco M, Moniz H, Rodrigues R. Alea-BFT: practical asynchronous byzantine fault tolerance//Proceedings of the 21st USENIX Symposium on Networked Systems Design and Implementation. Seattle, USA, 2024: 313-328

[19] Shao S S, Chen F, Xiao X Y, Gu W H, Lu Y C, Wang S, Tang W, Liu S D, Wu F, He J, Zhang K X, Mei F. IBE-BCIOT: An IBE based cross-chain communication mechanism of blockchain in IoT. World Wide Web, 2021, 24(5): 1665-1690

[20] Su H, Guo B, Lu J Y, et al. Cross-chain exchange by transaction dependence with conditional transaction method. Soft Computing, 2022, doi: 10.1007/s00500-021-06577-5

[21] Yin L Y, Xu J, Tang Q. Sidechains with fast cross-chain transfers. IEEE Transactions on Dependable and Secure Computing, 2021, 19(6): 3925-3940

[22] Nguyen C, Hoang D, Nguyen D, Xiao Y, Pham H, Dutkiewicz E, Tuong N. Fedchain: Secure proof-of-stake-based framework for federated-blockchain systems. IEEE Transactions on Services Computing, 2021, 16: 2642-2656

[23] Jalalzai M M, Niu J, Feng C, Gai F. Fast-HotStuff: A fast and robust BFT protocol for blockchains. IEEE Transactions on Dependable and Secure Computing, 2024, 21(4): 2478-2493

[24] Abbas H, Caprolu M, Pietro R. Analysis of polkadot: Architecture, internals, and contradictions//Proceedings of the 2022 IEEE International Conference on Blockchain. Espoo, Finland, 2022: 61-70

[25] Liu Z T, Xiang Y X, Shi J, Gao P, Wang H Y, Xiao X S, Wen B, Li Q, Hu Y C. Make web 3.0 connected. IEEE Transactions on Dependable and Secure Computing, 2021, 19(5): 2965-2981

[26] Sun Y Y, Yi L Y, Duan L, Wang W. A decentralized cross-chain service protocol based on notary schemes and hash-locking//Proceedings of the 2022 IEEE International Conference on Services Computing (SCC). London, UK, 2022: 152-157

[27] Xie T X, Gai K K, Zhu L H, Guo Y M, Choo K. Cross-chain-based trustworthy node identity governance in internet of things. IEEE Internet of Things Journal, 2023, 10(24): 21580-21594

[28] Xie T X, Gai K K, Zhu L H, Wang S, Zhang Z J. RAC-Chain: An asynchronous consensus-based cross-chain approach to scalable blockchain for metaverse. ACM Transactions on Multimedia Computing, Communications and Applications, 2024, 20(7): 1-24

[29] Lakhan A, Mohammed M A, Abdulkareem K H, Deveci M, Marhoon H A, Nedoma J, Martinek R. A multi-objectives framework for secure blockchain in fog-cloud network of vehicle-to-infrastructure applications. Knowledge-Based Systems, 2024, 290, doi: 10.1016/j.knosys.2024.111576

[30] Zhou Y, Ren Y L, Xu M T, Feng G R. An improved NSGA-III algorithm based on deep Q-networks for cloud storage optimization of blockchain. IEEE Transactions on Parallel and Distributed Systems, 2023, 34(5): 1406-1419

[31] Xu M T, Feng G R, Ren Y L, Zhang X P. On cloud storage optimization of blockchain with a clustering-based genetic algorithm. IEEE Internet of Things Journal, 2020, 7(9): 8547-8558

[32] Zheng W L, Zheng Z N, Dai H N, Chen X, Zheng P L. XBlock-EOS: Extracting and exploring blockchain data from EOSIO. Information Processing & Management, 2021, 58(3), doi: 10.1016/j.ipm.2020.102477

[33] Zhang P Y, Guo W F, Liu Z J, Zhou M C, Huang B, Sedraoui K. Optimized blockchain sharding model based on node trust and allocation. IEEE Transactions on Network and Service Management, 2023, 20(3): 2804-2816

[34] Wu Z Y, Liu J L, Wu J J, Zheng Z B, Luo X P, Chen T. Know your transactions: Real-time and generic transaction semantic representation on blockchain & web3 ecosystem//Proceedings of the ACM Web Conference. Lisbon, Portugal, 2023: 1918-1927

[35] Hu T, Liu X L, Chen T, Zhang X S, Huang X M, Niu W, Lu J Z, Zhou K, Liu Y. Transaction-based classification and detection approach for Ethereum smart contract. Information Processing & Management, 2021, 58(2), doi: 10.1016/j.ipm.2020.102462

[36] Xu J, Ming Y L, Wu Z H, Wang C, Jia X H. X-shard: Optimistic cross-shard transaction processing for sharding-based blockchains. IEEE Transactions on Parallel and Distributed Systems, 2024, 35(4): 548-559

[37] Zhang P Y, Zhou M C, Zhao Q X, Abusorrah A, Bamasag O. A performance-optimized consensus mechanism for consortium blockchains consisting of trust-varying nodes. IEEE Transactions on Network Science and Engineering, 2021, 8(3): 2147-2159

[38] Liu A D, Du X H, Wang N, Wu X Y, Shan D B, Qiao R. Research progress on security protection techniques in blockchain systems. Chinese Journal of Computers, 2024, 47(3): 608-646 (in Chinese)
(刘敖迪,杜学绘,王娜,吴翔宇,单棣斌,乔蕊.区块链系统安全防护技术研究进展.计算机学报,2024,47(3):608-646.)

附录

符号表

参数	含 义
$\bar{D}_S$	最佳节点集合S中节点到其余所有节点平均通信延迟
F_j	异步共识节点j的签名
$\boldsymbol{G}_K$	K个交易包所对应的大小矩阵
$\boldsymbol{H}_K$	N个异步共识节点在交易广播模块中分别广播K个交易包的预期时间矩阵
$\bar{J}_S$	集合S中节点到其余所有节点平均通信延迟
K	中继链交易池中跨链交易包的数量
L_j	异步共识节点j交易池中的待共识交易包队列
$\boldsymbol{M}_K$	K个跨链交易包在l轮异步共识中被N个异步共识节点广播所需的预期时间矩阵
N	异步共识委员会中共识节点数量
O	交易包中跨链交易的集合
P_j	共识节点j输入的共识提案
Q	跨链交易收据
R	跨链交易包
S	最佳节点集合
$\boldsymbol{T}_K$	K个交易包在l轮异步共识中交易广播的预期执行时间矩阵
$\boldsymbol{U}_K$	K个跨链交易包在l轮异步共识中与N个异步共识节点之间的关系映射矩阵
$\boldsymbol{V}_N$	N个异步共识节点的交易广播能力矩阵
$\boldsymbol{W}$	提议向量
X	中继链异步共识委员会节点集合
Y	中继链交易池中待共识的交易包集合
Z	交易包中跨链交易来自验证节点的签名集合
$\mathbb{H}$	初始种群
$\mathbb{S}$	多目标优化给定解集
$\mathbb{Y}$	种群中的个体
$\mathcal{A}$	来源链
$\mathcal{B}$	目标链
$\mathcal{C}_d$	中继链中d笔跨链交易从开始到完成共识的时间间隔
$\mathcal{D}$	中继链的交易吞吐量
$\mathcal{G}_j$	异步共识节点j在q轮异步共识过程所提出交易的共识成功率
$\mathcal{H}$	提交批量跨链交易所花费的时间
$\mathcal{J}$	中继链提交的批量跨链交易总数
$\mathcal{M}_d$	d笔跨链交易完成异步共识的时间
$\mathcal{N}_d$	d笔交易开始共识的时间
$\boldsymbol{\mathcal{O}}_K$	K个跨链交易包在l轮异步共识中被N个异步共识节点广播的预期共识成功交易数量矩阵
$\boldsymbol{\mathcal{Q}}_K$	K个交易包所对应的交易数量矩阵
$\mathcal{R}_{jz}$	异步共识节点j在第z轮异步共识中输入的总跨链交易数量
$\boldsymbol{\mathcal{S}}_K$	N个异步共识节点在交易广播模块中分别广播K个交易包的预期共识成功交易数量矩阵
$\mathcal{T}_{jz}$	异步共识节点j在第z轮异步共识中,未被包含在最终共识成功交易集合中的跨链交易数量
g_{ab}	矩阵$\mathcal{O}_K$中的元素

续表

参数	含 义
h_i	节点i发送消息的时刻
m_{ij}	矩阵$\boldsymbol{M}_K$中的元素
n_{ij}	矩阵$\mathcal{O}_K$中的元素
r	参考点
$s_{j,u_{ij}}$	节点j在第i轮异步共识中广播编号为u_{ij}的交易包的预期共识成功交易数量
w_d	矩阵$\mathcal{Q}_K$中的元素
z	多目标优化解
b_j^m	异步共识节点j缓冲区中交易包队列中的第m个交易包的大小
d_{ij}	节点i与j之间的通信延迟
f	异步共识委员会中恶意节点的数量
g_z	交易包z的大小
h	矩阵$\boldsymbol{H}_k$中元素
l	K个交易包的预期共识执行轮数
r_{ab}	矩阵$\boldsymbol{M}_K$中的元素
s_{jz}	异步共识节点j广播交易包z所花费的时间
t_m	中继链中轮次为m的异步共识过程中广播交易的预期完成时间
u_{ij}	异步共识节点j在第i轮异步共识中所对应的交易包编号
v_j	异步共识节点j广播跨链交易包的速度
w_{jz}	异步共识节点j所广播的交易包z的大小
y_c	矩阵$\boldsymbol{T}_K$中的元素
τ_j	第j个PB组件的编号
σ_i	节点i的门限签名
γ_i	节点i发送收到消息的时刻
$\hat{\xi}_j$	节点j的运行性能值
κ_j^i	节点j完成第i轮异步共识的时延
φ_j	节点j在n轮异步共识中的准确率
η_j	节点j在n轮异步共识中向ABA实例中输入正确值的次数
Ω	所有满足条件的节点集合S的集合
$\bar{\phi}_S$	最佳节点集合S的节点平均共识准确率

ZHANG Pei-Yun, Ph. D., professor. Her research interests include blockchain, trusted computing, service computing, and intelligent information processing.

XU Fu-Ya, M. S. Her research interest is blockchain.

CHEN Zi-Han, M. S. candidate. His research interest is blockchain.

Background

The issue addressed in this work falls within the research domain of consensus mechanisms in blockchain cross-chain technology.

Blockchain technology, as a distributed ledger technology that enables decentralized, secure, and trustworthy data storage and transactions, has achieved significant progress over years of development. However, due to the high heterogeneity among different blockchains, interoperability issues between them present considerable challenges. To address the value island problem caused by single-chain systems, cross-chain technology has emerged. Cross-chain technology refers to the methods that enable data and asset interoperability between multiple

independent blockchain distributed ledgers. The relay chain technology is a solution that establishes connections between different blockchains using a third-party blockchain. It achieves cross-chain interaction through cross-chain messaging protocols and mainly serves to connect multiple application chains, providing services for cross-chain transactions and validation. To ensure the security and stability of the relay chain and maintain consistency and reliability among connected blockchains, an appropriate consensus mechanism is urgently needed.

Currently, most relay chain cross-chain mechanisms adopt partially synchronous consensus mechanisms. However, as the blockchain ecosystem expands, the consensus burden between different blockchains increases, which may make it difficult for consensus nodes to achieve global consistency. This can limit the number of application chains that the relay chain can connect, ultimately affecting the overall stability of the cross-chain system. In contrast, asynchronous consensus mechanisms offer advantages in handling arbitrary transmission delays and network uncertainties among nodes. They can more flexibly adapt to changes in network conditions, improving the fault tolerance and stability of the entire relay chain cross-chain system.

Asynchronous consensus mechanisms, such as HB-BFT and Dumbo, are continuously evolving and have made gradual improvements in throughput and latency. This suggests that asynchronous consensus mechanisms may provide a viable solution to enhance the robustness and efficiency of relay chain consensus protocols. However, these mechanisms still face challenges such as low throughput and high latency. Therefore, there is an urgent need to introduce optimized asynchronous consensus mechanisms in relay chains to improve cross-chain interaction. This work proposes an optimized relay chain cross-chain asynchronous consensus mechanism by designing a transaction broadcast module based on optimal cross-chain transaction matching and a multi-valued Byzantine consensus module based on proposal vector selection optimization.

This work was supported by the National Natural Science Foundation of China (No. 61872006), which focuses on key technologies for trusted blockchain, addressing user trust requirements and personalized transactions. The results have been published in 30 academic works in journals and international conferences, such as SCIENCE CHINA Information Sciences, IEEE Transactions on Automation Science and Engineering, IEEE Transactions on Systems, Man, and Cybernetics: Systems, IEEE Transactions on Network and Service Management, IEEE Transactions on Network Science and Engineering, and IEEE Transactions on Computational Social Systems. The research results of this paper can enhance the part of blockchain consensus mechanisms.