面向异构区块链系统的中继链跨链方案

聂 鹏[1)] 王 玫[1)] 王 馨[2),5)] 赵 伟[3)] 段斯斯[4),5),6)] 贾珂婷[5),6),7)] 张国艳[2),3),5)]

[1)](山东大学网络空间安全学院 山东 青岛 266237)

[2)](山东大学密码科学与工程学院 济南 250101)

[3)](山东区块链研究院 济南 250001)

[4)](清华大学高等研究院 北京 100084)

[5)](密码与数字经济安全全国重点实验室 济南 250100)

[6)](中关村实验室 北京 100095)

[7)](清华大学网络科学与网络空间研究院 北京 100084)

摘 要 随着区块链技术的快速发展，数百种区块链系统陆续被提出并应用于不同服务中。面向一些数据跨域流通、链上资产转移等需求，跨链互操作是这些区块链系统的关键技术。跨链原子性是保障跨链操作安全的重要性质，旨在实现跨链交易在所有相关的区块链上都被执行或都不被执行的特性。针对跨链原子性，现有技术包含了哈希锁、原子交换、中继链等，其安全假设存在差异，适用于不同应用场景。其中基于中继链的技术路线是唯一具有普适性，且对交易方不做过多要求的技术路线。现有跨链技术通常将区块链当做黑盒来使用，缺乏对异构区块链，尤其是兼容其底层共识协议的处理方法，仅能够在理想情况下实现跨链原子性。此外，缺乏对多链的跨链互操作技术，现有跨链方案扩展性有待增强。本文提出了一种基于中继链的全新的跨链系统架构，通过异构链管理机制和基于向量时钟的多链协调机制两个模块，可以兼容异构区块链，尤其是兼容其底层共识协议的不同活性假设，并具有较好的可扩展性。具体来说，异构链管理机制将区块链共识的活性性质分成了两大类，针对每类的活性特性，进行了相应的跨链操作处理，去掉了传统二阶段确认(即 2PC)中对全局物理时钟计时器的强假设。同时，多链协调机制通过采用轻量级的向量时钟，针对不同跨链操作对原子性需求不同的特性，设计了灵活的跨链交易执行模式，如部分跨链操作仅需要在源链(即跨链交易发起的区块链)保证执行顺序，而无需在意目标链(即非发起方区块链)的执行顺序。通过在 FISCO BCOS、Fabric、CITA、Dyno 等不同的区块链上的大量实验，表明了本文方法的实用性。具体的跨链交易处理时间和相应的区块链系统的延迟、网络延迟、交易大小等因素相关，本文所提出的跨链方案由于去掉了全局物理时钟计时器的强假设，基本消除了跨链中的额外操作及等待时间，实现了跨链操作的高效性。

关键词 区块链互操作性；中继链；区块链最终性；向量时钟；异构区块链

中图法分类号 TP309 DOI 号 10.11897/SP.J.1016.2026.00638

收稿日期：2025-07-17；在线发布日期：2025-11-10。本课题得到国家重点研发计划项目(2022YFB2702800)资助。**聂 鹏**，博士研究生，主要研究领域为数据安全与隐私保护。E-mail: 202521420@mail.sdu.edu.cn。**王 玫**(共同第一作者)，硕士研究生，主要研究领域为分布式系统、区块链技术。E-mail: wangmay_w@163.com。**王 馨**(通信作者)，博士，助理研究员，主要研究领域为分布式系统安全、区块链技术。E-mail: wangxin87@sdu.edu.cn。**赵 伟**，硕士，主要研究领域为区块链技术。E-mail: 694469927@qq.com。**段斯斯**(通信作者)，博士，研究员，中国计算机学会(CCF)会员，主要研究领域为区块链、应用密码学。E-mail: duansisi@tsinghua.edu.cn。**贾珂婷**，博士，副研究员，主要研究领域为密码算法的分析设计、密码技术应用。E-mail: ktjia@tsinghua.edu.cn。**张国艳**，博士，教授，主要研究领域为密码算法的分析设计、区块链技术、密码创新应用。E-mail: guoyanzhang@sdu.edu.cn。

Towards a Relay Chain-Based Cross-Chain Solution for Heterogeneous Blockchains

NIE Peng[1)] WANG Mei[1)] WANG Xin[2),5)] ZHAO Wei[3)] DUAN Si-Si[4),5),6)]
JIA Ke-Ting[5),6),7)] ZHANG Guo-Yan[2),3),5)]

[1)](School of Cyber Science and Technology, Shandong University, Qingdao, Shandong 266237)

[2)](School of Cryptologic Science and Engineering, Shandong University, Jinan 250101)

[3)](Shandong Institute of Blockchain, Jinan 250001)

[4)](Institute for Advanced Study, BNRist, Tsinghua University, Beijing 100084)

[5)](State Key Laboratory of Cryptography and Digital Economy Security, Jinan 250100)

[6)](Zhongguancun Laboratory, Beijing 100095)

[7)](Institute for Network Sciences and Cyberspace, Tsinghua University, Beijing 100084)

Abstract With the rapid development of blockchain technology, hundreds of blockchain systems have been proposed and are serving different applications. As the data and digital assets are stored on different blockchains, the needs to transfer or exchange assets make blockchain interoperability an important technology. Cross-chain atomicity is a crucial security property of blockchain interoperability. Cross-chain atomicity ensures that a cross-chain transaction is executed by all the blockchains (that are related to the transaction), or none of the blockchains execute the transaction. To realize cross-chain atomicity, known approaches include hash time lock contract (HTLC), atomic swap, relay chains, etc. Each of the approaches has its own pros and cons. Among them, the relay chain is the only generic solution that does not pose strong requirements on the users (who are involved in the cross-chain transactions). Meanwhile, regardless of the approaches taken by current cross-chain techniques, almost all cross-chain solutions treat each blockchain as a black-box and assume that each blockchain achieves the ideal security properties, i.e., safety and liveness of the blockchain consensus. However, practical consensus mechanisms might have nuance in their security definitions. There is a lack of treatment for cross-chain transactions for heterogeneous blockchains, especially with a focus on the underlying consensus mechanisms. To conclude, existing approaches are not yet scalable in terms of handling cross-chain transactions where multiple blockchains are involved. This paper presents CrossWeave, a new relay chain-based solution for blockchain interoperability. CrossWeave consists of two main modules: heterogeneous blockchain management and vector clock-based multi-chain coordination. CrossWeave provides a lightweight and efficient solution for cross-chain transactions where multiple heterogeneous blockchains are involved. Specifically, the heterogeneous blockchain management module takes into consideration the liveness property of the Blockchain consensus mechanisms. It classifies the liveness properties of consensus mechanisms into two categories and takes different actions for different blockchains based on the category of the liveness property. In this way, the strong requirement on a global physical clock-based timer of the conventional 2-Phase Commit (2PC) approach is removed. Meanwhile, the multi-chain coordination module uses a lightweight vector clock and takes a fine-grained treatment of the atomicity requirements of different cross-chain transactions. For instance, some cross-chain transactions only care about the execution order on the source chain (i.e., the blockchain in which a cross-chain transaction is initiated) and do not have to be executed sequentially on the target chain (i.e., all non-source blockchains). Using our approach, cross-chain transactions are executed according to the nature of the transactions, making our approach more flexible. Via extensive experiments on FISCO BCOS, Hyperledger Fabric, CITA, and Dyno, we show that our solution is

efficient and highly practical. Specifically, besides the unavoidable factors such as latency of the blockchain systems themselves, network delay, and transactions, our approach almost eliminates the overhead caused by the global physical clock-based timer and is highly efficient.

Key words blockchain interoperability; relay chain; blockchain finality; vector clock; heterogeneous blockchain

1 引 言

1.1 研究背景与意义

区块链技术经过十余年的发展，已经产生了公链、联盟链、私有链等多种形态。这些区块链系统采用不同的共识机制、数据结构和编程模型，服务于不同的应用，形成了丰富多样的异构区块链环境。随着区块链应用的深入发展，区块链的跨链互操作需求日益突出[1-2]，其应用场景已经覆盖了侧链、区块链间的资产转移及交换等需求。在软件领域，“互操作性”通常是指两个或多个软件的协作，而这些软件可以用不同的语言编写，甚至运行在不同的平台上[3]。然而在区块链中，跨链不仅仅有软件层面的互通，还包含了数据层面、协议层面等多个复杂的逻辑。同时，也要保障系统的安全性。

跨链的原子性是跨链交易安全的重要属性，跨链原子性是指一笔在不同区块链之间的跨链交易被全部的区块链系统执行，或者全部的区块链系统都不执行，可以被理解为跨链交易防止“双花”的基本属性。当前，跨链具有多种技术路线，其安全假设、信任基础各不相同。主流的技术路线包含了哈希锁、原子交换、侧链、中继链等。其中基于中继链的方案是普适性最高、可扩展性较强的跨链技术路线，其安全通常基于中继链本身的安全性，并通过扩展中继链的功能，提供通用的跨链服务。工业界中有许多具有代表性的中继链方案，例如在海外针对加密数字货币平台的 Cosmos[4]、Polkadot[5] 以及 Chainlink 的 Cross-Chain Interoperability Protocol (CCIP) 方案[6]，以及国内的星火链网等平台。

尽管各类跨链方案陆续被提出，现有的跨链解决方案通常假设所有参与的区块链具有相同的安全属性，忽略了异构区块链系统的差异化处理，尤其是针对底层共识协议的异构化的处理。这种差异在跨链场景中尤为关键，如果不加区分地处理，可能导致交易状态不一致，严重影响跨链系统的可靠性。同时，现有的跨链协调机制往往只关注单一类型的原子性保证，缺乏灵活性，难以适应多样化的应用需求。

1.2 现有跨链范式的挑战及研究问题

以最常见的基于中继链的跨链原子性方案为例，几乎所有的现有跨链方案都遵循了两阶段提交 (即 Two-Phase Commit，2PC) 的范式，以两条链 A 、B 举例，中继链在此时起到 2PC 中协调者的作用。经典的 2PC 模式可以用来保障交易的原子性，该范式被广泛用于数据库等领域，也是区块链跨链原子性的基础之一。事实上非中继链的跨链方案也基本遵循了 2PC 的范式，只是表达形式略有出入。

2PC 模式同时也存在着许多细微的问题。例如，传统的 2PC 方案需要让协调者扮演“可信第三方”去管理可能出现的交易异常。然而在跨链环境中，中继链由分布式节点组成，并不属于一个真正意义上的“可信第三方”，同时由于不同区块链确认时间的差异、网络可能出现的暂时分区等原因，跨链交易的原子性在中继链方案中难以得到保障。此外，中继链作为系统的核心枢纽，其自身的处理性能 (如吞吐量和延迟) 也可能成为整个跨链系统的瓶颈[7]。另一个问题是链与链之间底层共识机制不同导致的跨链兼容难题。由于每个区块链的底层共识机制、系统架构、安全假设各不相同，导致各个链的交易状态不一致，异构链跨链的活性保障 (即交易是否能够被处理) 也无法兼容，甚至出现资产丢失或重复创建等严重问题。

我们可以看出，针对异构区块链的跨链处理方法，仍然需要进一步的研究。本文致力于解决以下核心问题：

如何设计一个能够针对异构区块链的通用中继链跨链方案，以确保异构环境下跨链交易的原子性与活性?

针对上述挑战，本文提出了一种全新的基于中继链的跨链方案 CrossWeave，构建了基于活性标记的异构链管理模块和基于向量时钟的多链协调模块，实现了对异构区块链的安全管理和可靠跨链。

1.3　本文贡献

针对上述挑战，本文的主要贡献包括：

(1) 提出了区块链跨链异构链管理模块，面向不同的异构区块链，通过对每条链进行活性标记，并通过在跨链协议中差异化的交易处理策略，实现了系统对异构链的兼容性，提升了跨链交易的可靠性。

(2) 设计了基于向量时钟的多链协调模块，通过对不同链本地与全局向量时钟的设计与更新机制，维护了系统对不同区块链跨链原子性要求，为不同应用场景提供了灵活的选择。

(3) 构建了基于中继链的跨链系统 CrossWeave，实现了兼容数据层、异构链管理模块、多链协调模块的跨链方案，通过统一的接口体系实现了跨链交易的全生命周期管理，支持资产跨链转移、消息跨链传递和智能合约跨链协作等多种应用场景。

(4) 通过在 FISCO BCOS[8]、Fabric[9]、CITA[10]以及 Dyno[11]等不同的区块链上的多组实验，证明了 CrossWeave 跨链系统处理跨链交易的低延迟与高效率。

1.4　论文组织结构

本文余下部分组织如下：第 2 节介绍相关工作；第 3 节展示协议的系统模型与安全假设；第 4 节介绍了 CrossWeave 的系统架构与协议的核心思路，并具体讨论了方案中的两大核心模块；第 5 节介绍了跨链协议的详细执行流程与系统针对强原子性进行的额外约束设计；第 6 节介绍了 CrossWeave 的工程实现；第 7 节给出性能测试结果；第 8 节总结全文并指出未来研究方向。

2　相关工作

2.1　跨链技术架构

根据系统架构的不同，跨链技术主要可以分为侧链架构、跨链桥架构、中继链+桥接混合架构、原子交换等。表 1 总结了各类技术路线的特点。

表 1　跨链架构核心区别对比

维度	侧链架构	跨链桥架构	中继链+桥接混合架构	原子交换
信任来源	信任主链与侧链的验证者网络	信任验证者网络或智能合约	信任中继链	无需信任第三方
安全性	主链+侧链安全性	依赖验证者网络安全性	共享中继链安全性	哈希时间锁保证
去中心化	较高	较高	中等（存在中心化风险）	最高
技术复杂度	中等	中等	高	低
互操作性	支持主链与多条同构侧链链接	任意两链链接	任意多条异构链链接	仅允许链间资产交换
典型项目	Polygon[12], Liquid[13]	Multichain[16], Wormhole[17]	Polkadot[5], Cosmos[4]	BTC-ETH 原子交换
主要优势	高安全性，高效率	灵活连接，部署简单	统一协调，可扩展性强，适用范围广	完全去中心化
主要劣势	限于同构链	限于两链互跨，依赖验证者	治理相对复杂，业务链安全性依赖于中继链安全性	功能有限，仅支持简单交换

侧链架构拥有自己的共识机制和验证者网络，是指通过双向锚定(Two-way Peg)机制，将主链资产锁定并在侧链上发行等价代币，典型项目包括以太坊的侧链 Polygon[12]、比特币的侧链 Liquid Network[13]和 Blockstream[14]等。其中性能较优异的 Polygon PoS 链上的交易速度极快(通常在几秒内确认)，但存款(以太坊到 Polygon 的跨链交易)通过官方的 PoS 桥，大约需要 10～30 min 完成确认①，而取款(Polygon 到以太坊的跨链交易)需要等待到达某个检查点才最终被提交到以太坊主网络，通常需要 2 h～3 h②。跨链桥架构实现多条链资产和数据的双向流动，由部署在各链上的智能合约组合，或由验证者网络维护的跨链通信机制实现。这种架构连接灵活，可以根据需要建立任意两条链之间的互通。典型项目包括跨链彩虹桥 (Rainbow Bridge)[15]、Multichain (原 Anyswap)[16]、Wormhole[17]、Celer cBridge[18]，以及陆羽跨链协议[19]等。普通跨链桥架构通常采用“锁定-铸造”(lock-and-mint)机制，跨链交易时间浮动较大，延迟从数分钟到数小时不等③。中继链+桥接混合架构则通过中继链来协调多链的互操作，中继链作为中心枢纽链接多条业务链，提供统一的共识和安全保障，桥接机制则负责与外部异构链的连接。典型项目包括 Polkadot[5]，基于 Hyperledger Fabric[9]框架构建的跨链互操作网络(如

① Polygon Bridge: A Secure Way to Transfer Assets, https://polygon-bridge.github.io/

② Polygon Bridge: The Fastest Way to Bridge to Polygon in 2025, https://across.to/blog/polygon-bridge-guide-2025

③ PortalBridge by Wormhole is a scam, https://www.reddit.com/r/CryptoCurrency/comments/1fva5p1/portalbridge_by_wormhole_is_a_scam/

Cacti)，Cosmos[4]以及其衍生协议 IRIS[20]等，其中 Cosmos 侧重 token 交换，而 IRIS 侧重数据交换。在 Polkadot 中，平行链的出块速度与中继链保持一致，即每 6 s 一个区块，跨链交易的最终确认时间约为 16.5 s 至 24 s①；Cosmos 中 IBC 消息的平均中继时间约为 55.4 s②。Polkadot 通过共享安全模型，实现了秒级的极低延迟，但代价是牺牲了对通用异构链的原生兼容性；Cosmos 的 IBC 协议虽然通用性强，但其点对点的模式决定了完成一次跨链交互至少需要三次独立的链上交易和两次网络往返，延迟相对较高。本文方案也采用了中继链+桥接混合架构，与 IBC 类似，通过中继链引入了额外的协调步骤，单笔交易的基础延迟高于 Polkadot 等高度集成的系统，而与 Cosmos 的延迟相仿，在与其他中继链+桥接混合架构性能相差不大的情况下提供了通用性和灵活性。

原子交换是一种不依赖跨链基础架构的点对点交换协议，通常基于哈希时间锁合约实现。这种方式完全去中心化，无需信任第三方，但功能相对有限，主要用于简单的资产交换，例如比特币和以太坊上的原子交换等。原子交换的总耗时由两笔链上交易在各自网络中的确认时间决定。一笔 BTC-ETH 的交换通常需要等待比特币网络的数个区块确认以太坊网络的区块确认，大约需要 20 min 到超过一个小时。

2.2 跨链技术信任机制

从信任机制的角度来看，跨链技术的信任来源可以分为三个主要类别：原生验证(Native Verification)、本地验证(Local Verification)和外部验证(External Verification)。不同跨链架构在结构上较为相似，但信任机制完全不同。

对于中继链架构来说，系统整体的跨链安全性完全依赖于中继链的安全性，而侧链架构可以通过轻客户端验证等方式验证主链状态，去中心化程度通常更高。原生验证模型提供了最高的安全性，因为它直接验证各条链的状态，不依赖任何第三方。但代价是为了实现任意 N 条链的互操作性，各节点都需要为其他 N - 1 个节点运行轻客户端，导致极高的通信和计算开销。除此以外，所有参与方需要持续在线，这导致用户体验不佳。外部验证通过引入一个可信实体，极大地简化了连接拓扑(所有链只需连接到公证人)，提供了最高的效率和可扩展性。但其代价是需引入额外的甚至是中心化的信任假设。附录 A(表 2)给出了跨链信任模型核心区别对比表，附录 B 对其进行了补充介绍。

本文采用结合了外部验证与原生验证的混合信任模型，实现了安全性、效率与可扩展性的平衡，为 CrossWeave 跨链协议奠定了必要且合理的架构与信任基础。

2.3 异构环境下的跨链协议挑战

区块链系统特性分析。异构区块链系统的不同特性直接影响跨链协议的设计和实现。Garay 等人[21]在论文中首次形式化定义了比特币区块链的三个核心属性：链质量、链增长和公共前缀。链质量确保区块链中诚实节点贡献的区块占一定比例，防止攻击者过度控制区块链内容。链增长保证在一定轮数内，区块链的长度会增长一定数量，这正是区块链活性的体现。公共前缀确保在一定深度后，不同诚实节点维护的区块链具有相同的前缀，为区块链的安全性提供保障。这些属性共同构成了分析比特币区块链的活性和安全性的基本框架。

共识机制的不同直接影响区块链系统的活性特性[22]。Kiayias 等[23]在分析 Ouroboros 系统的权益证明(PoS)共识时，证明了在特定条件下 PoS 共识可提供与工作量证明(PoW)类似的活性保证。Buchman 等人[24]使用基于拜占庭容错的共识提供强活性，即一旦达成共识，交易结果不会回滚，但在出现网络分区等故障时共识可能无法推进，一定程度上牺牲了部分活性。Gervais 等人[25]系统分析了基于 PoW 及其变体区块链的安全性和性能参数，给出了系统活性与安全性之间的复杂关系。这些研究为我们构建异构区块链活性分类体系提供了坚实基础。

跨链原子性与一致性机制。跨链原子性是确保多链环境中交易一致性的关键机制，在异步环境中尤为重要。Zakhary 等人[26]指出在基于哈希时间锁的系统中，由于服务崩溃或网络延迟往往会导致诚实参与者资产损失，在异步环境中难以保证原子性。Raynal 等人[27]提出了分布式系统中逻辑时钟系统的一般框架。Basten 等人[28]基于向量逻辑时钟，进一步讨论了弱优先关系与强优先关系。向量时钟启发了跨链系统对于不同依赖关系使用不同原子性模

① Parallel Computing, https://wiki.polkadot.com/learn/learn-elastic-scaling/

② A Deep Look Into Cosmos — the Internet of Blockchains, https://juliankoh.medium.com/a-deep-look-into-cosmos-the-internet-of-blockchains-af3aa1a97a5b

型的研究。尽管向量时钟为捕获分布式系统中的因果关系提供了强大的理论基础，但如何将其应用于一个能够灵活支持不同强度原子性保证并能与异构链活性模型相结合的统一跨链框架，仍然是一个有待深入研究的开放性问题。

3 系统模型与相关概念

本工作考虑一个通过第三方中继链 RC 完成的，针对区块链 A 和区块链 *B* 的跨链通信模型。假设中继链 RC 包含 n 个节点，链 A 和 B 分别包含一定数量的节点，例如$|A| = m_a$且$|B| = m_b$，在 A 中的节点可以表示成$\{p_1^A,\dots,p_{m_a}^A\}$，同理在 B 中的节点可以表示成$\{p_1^B,\dots,p_{m_b}^B\}$。本 p 工作假设一个半同步网络，即网络中消息延迟存在未知上界δ，同时假设一个静态拜占庭敌手模型，其中对手在系统启动并运行之前破坏了副本。

3.1 术语及定义

向 CrossWeave 跨链系统提交交易(请求)的客户以及所有其他客户账户都可以通过中继链 RC 进行验证，默认情况下，客户首先会提交跨链交易(开户和销户除外)给任一区块链节点，该区块链被称为该条跨链交易的源链 SC，源链节点会按照一定的顺序来处理这些交易。随后 SC 会向中继链 RC 发送跨链请求，待 RC 处理完成后转发跨链请求给另一区块链，则该区块链被称为该条交易的目标链 TC。待 TC 处理完成后，反馈结果给 RC，RC 再转发该反馈结果给 SC。本文将系统中所有区块链的本地共识看作一个黑盒模型，假设各链的内部执行原子广播协议[29]并满足其安全定义。

原子广播协议。在原子广播协议(Atomic broadcast, ABC)中，当系统中某个节点作为领导人节点发起消息/交易广播(a-broadcast 事件)，协议即被触发，最终系统中所有节点执行交付(a-deliver 事件)并完成消息/交易的共识，结束一个序列号的原子广播协议。其定义如下所示：

安全性：如果有一个正确节点在 a-deliver 区块 *m* 之前 a-deliver 了区块 m′，则没有正确节点在 a-deliver 区块 m′ 之前不先 a-deliver 区块 m；

活性：如果有一个正确节点 a-broadcast 了区块 m，则所有正确节点最终都会 a-deliver 该区块 m。

这里我们对原子广播协议的 API 进行了限制，即只有一个节点 a-broadcast 一条交易，在完全异步的情况下，所有节点也可同时 a-broadcast 多条交易。原子广播协议的 API 没有明确指定消息、交易等事务的顺序，在实践中，大多数协议将每个事务/区块与序列号(sequence)、高度(height)或轮次(epoch)关联起来。在本文中，我们使用序列号(gseq 和 lseq)来指定交易与区块的顺序，假设系统中所有节点都可验证已经 a-deliver 的上链区块以及区块中的交易序列。

本文系统假设半同步时间模型[30]，即存在未知的全局稳定时间，使得在该稳定时间之后，两个正确节点之间发送的交易可在固定的延迟内到达。在本文实验中，我们默认使用 Dyno 作为区块链的原子广播协议，另外分别使用 CITA、FISCO BCOS 作为对比项，该三种协议均假设半同步模型。

向量时钟。向量时钟是一种分布式系统中常用的指代事件逻辑顺序的方法，可以让每个节点根据本地的向量时钟值进行判断，而无需全局的事件顺序[31]。具体来说，向量时钟是一种用数字而非物理时间(有时也会同时使用，即混合向量时钟)代表了事件的先后顺序[32]。一个向量时钟是一个用数字构建的向量，向量的长度通常为节点的个数，在本文中，向量时钟的长度为需要参与跨链的区块链个数。如果我们用 G 来表示向量时钟，针对两个向量时钟 G_1, G_2所指代的事件 1 及事件 2，事件 1 发生在事件 2 之前的必要条件是，对于任意的 i, i 的区间为 G_1 及 G_2的长度(二者长度相同)，$G_1[i] \le G_2[i]$。如果不存在这种可以比对的关系，则事件 1 和事件 2 是并行的事件，即不存在先后序。

源链与目标链。我们假设区块链 A 为跨链请求的发起方，又称为源链 SC；假设区块链 B 为跨链请求的接收方，记为目标链 TC；在本文跨链进程中，源链和目标链可统一称为业务链 C。本文假设 C 包含 m_i 个节点，其中可最多容忍 t_i 个拜占庭错误，即 $m_i \ge 3t_i + 1$。我们假设业务链 C 中每个节点 p_i^C 在注册进入系统后均拥有一个唯一身份标识符，且每条业务链中节点的身份都是公开可查询的。

本文专注于交易的跨链处理，目标将 SC 已上链的交易通过中继链协调的方式与其他链进行互跨，不考虑 SC 在发起跨链交易之前的交易同步，假设 SC 节点在跨链交易发起之前已经通过本地共识协议(例如原子广播)完成交易排序并形成区块，且系统中所有节点均可验证该区块。本文系统假设区块链账户模型，即任意客户在系统中都拥有账户且可跟踪余额变化。系统使用序列号 lseq 来表示源

链和目标链本地的跨链交易序列号，gseq 来表示全局序列号，系统中所有业务链的全局的交易序列号则由向量时钟 G_i表示，G_i由中继链 RC 维护更新，业务链可根据 RC 中 G_i对应元素的值(即 gseq)，来更新自己本地交易的状态 lseq。

中继链。中继链 RC 作为跨链交易的第三方角色，运行多链跨链协议以维持各个异构业务链的活性与原子性。本文假设 RC 包含 n 个节点，其中可最多容忍 f 个拜占庭错误，且 RC 最优保持 $n \geq 3f+1$。

当业务链 C 注册进入系统时，RC 会根据各个异构链的活性假设对其标记活性 L，在执行跨链交易请求时，RC 会依据标记 L 对交易列队进行不同处理。为了协调多链间的交易原子性，RC 通过向量时钟 G_i维护系统全局状态，当业务链 C 注册进入系统时，RC 会在 G_i中初始化链标识符 C 的跨链序列号时钟，中继链 RC 在 C 完成跨链交易后更新各 C 的时钟，用于追踪系统中所有业务链的跨链序列号。

在 RC 共识完跨链交易 tx 后，该交易会被分成两笔子交易 tx_s和 tx_t，分别发送给源链与目标链。例如，若 tx 是将资产 1 从链 A 转移到链 B 的操作，那么这笔交易将会被拆分成 tx_s：删除资产 1 的所有权；tx_t：创建资产 1，并将所有权设置为相应拥有者。

可用性证书。可用性证书(Availability Certificate)[33]是一种可验证的密码学凭证，用于证明一笔交易(或包含该交易的区块)已在其所属的业务链上被特定数量的节点群体(Quorum)所接收、验证并达成共识。一个可用性证书 ac 通常由业务链 C 中共识节点的一个子集，来针对特定交易的哈希或其所在区块的哈希生成的数字签名聚合而成。在本文中，源链 SC 在本地完成对跨链交易的共识后，生成相应的可用性证书 ac。随后，ac 作为跨链交易请求 X-Req 的一部分，提交给中继链 RC。RC 在收到请求后，无需与 SC 节点通信，仅需验证该证书的有效性，即可确信相关交易已在源链上被确认，从而安全地启动后续的跨链交易流程。

3.2 跨链安全目标

CrossWeave 跨链系统主要实现了不同区块链间的互操作性，其安全目标也主要遵循传统区块链中拜占庭共识协议(即 Byzantine fault-tolerant state machine replication, BFT)。将向系统提交交易的角色定义为“客户端”，在本文跨链事务中，当源链中某个节点作为客户端向本链其他节点提交跨链交易请求(x-submit 事件)，CrossWeave 跨链协议被触发。在源链共识完成后，该客户端继续向中继链转发跨链交易广播(x-broadcast 事件)，最终系统中所有业务链节点发起跨链交易交付(x-deliver 事件)，结束一个序列号的跨链共识协议。为了实现系统的活性和原子性，本文在此部分对跨链协议的安全性质进行基本定义。

CrossWeave 跨链协议主要遵循如下安全性质：

(1) 强跨链活性(Strong Cross-Chain Liveness)：如果源链任意一个正确节点在系统中 x-broadcast 跨链交易 m，那么源链和目标链中所有正确的节点最终都会 x-deliver 该交易 m；

(2) 弱跨链活性(Weak Cross-Chain Liveness)：如果源链任意一个正确节点在系统中 x-broadcast 跨链交易 m，那么源链和目标链中至少有一正确节点最终会 x-deliver 该交易 m；

(3) 强跨链原子性(Strong Cross-Chain Atomicity)：对于任意两个跨链交易请求 m 和 m′，在所有源链和目标链中，如果有一个正确节点在 x-deliver 交易 m 之前 x-deliver 了 m′，则没有正确节点在 x-deliver 交易 m′之前不先 x- deliverm；

(4) 弱跨链原子性(Weak Cross-Chain Atomicity)：对于任意两个跨链交易请求 m 和 m′，在任一源链或目标链中，如果有一个正确节点在 x-deliver 交易 m 之前 x-deliver 了 m′，则没有正确节点在 x-deliver 交易 m'之前不先 x-deliver m。

强跨链活性是系统可靠性的保证，它确保任意一个被有效提交至系统的跨链交易请求 m，最终在时间上界 t 内会被确认。而弱跨链活性没有要求明确的时间上界，对于任意一个跨链交易请求 m，只要网络最终进入一段“足够长”的同步期(例如半同步网络)，且至少部分正确节点持续提议 m，则 m 最终会被确认。

强跨链原子性是较强的原子性保证，它要求在整个跨链系统中建立一个全局总排序。对于系统处理的任意两笔跨链交易，无论它们的来源或目标是什么，都必须存在一个所有参与方共同认可的、唯一的先后顺序。这使得整个异构多链系统在逻辑上如同一个单一、串行执行的处理机，为需要处理复杂跨链依赖关系的业务提供了最强的正确性保障。

弱跨链原子性是不需要实现全局的一致性，它放宽了全局总排序的要求，仅保证相同目标链的跨

链事务是按序交付的。该性质允许不涉及任何共同业务链的跨链交易并行处理。虽然弱跨链原子性在一致性保障上不如强跨链原子性，但其能够被用于一些特殊的交易中，例如针对转账类的交易并不需要在目标链中实现全局一致性。比如考虑两笔交易：tx_1为 Alice 向 Bob 转账 100；tx_2为 Alice 向 Carol 转账 100，Alice、Bob、Carol 分别的资产是在 A、B、C 三条链中。在实际跨链中，只需要保障 tx_1和 tx_2在 Alice所在 A 链中执行是按照顺序执行(否则会产生双花)，无需保障 B 链和 C 链两条链间的执行顺序，这是因为 B 和 C 两条链处理的交易内容没有关联。

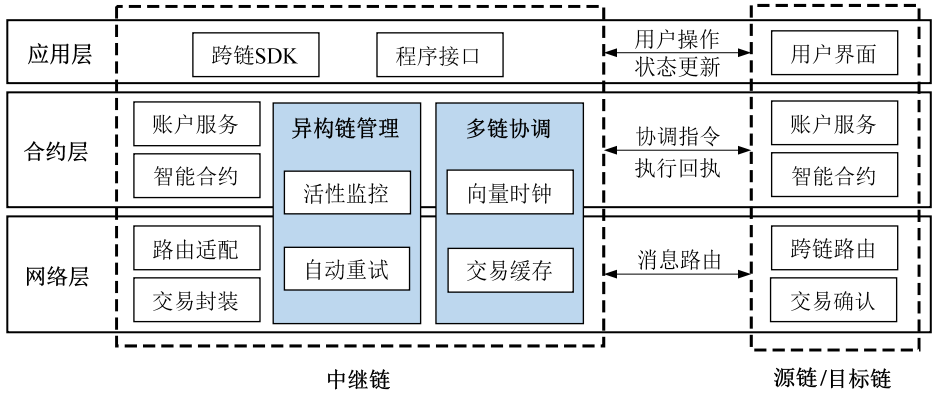
我们可以认为针对不同的跨链场景，只需要选择强跨链原子性或者弱跨链原子性进行实现即可。在 CrossWeave 系统中，本文提供对两种性质都支持的方案。

4 系统概述

为了完整地阐述 CrossWeave 跨链协议的工作方式，本章将系统性介绍跨链协议的整体分层架构，包括系统网络层、合约层、应用层的模块分布，并详细概括系统中两大核心模块，异构链管理模块与多链协调模块。

4.1 系统架构

CrossWeave 跨链协议的系统架构如图 1所示，系统架构主要分为网络层、合约层、应用层三个层级，各层级相互协同完成跨链交易事务。



注：蓝色部分为本文主要贡献

图 1 系统架构图

网络层是跨链架构中的底层关键部分，主要负责在各业务链的不同网络之间传输数据并确保数据包的正确路由，提供了跨链互操作性的网络通信功能，例如接收发送不同的跨链业务交易，支持不同区块链系统的交易格式转换与封装、待确认交易的缓存、交易共识过程中的账户哈希锁定等功能，网络层使用路由协议(如微众银行牵头提出的陆羽协议等)来确定数据的最佳传输路径，通过各节点在网络中的唯一标识符，确保高效的数据传输。

合约层也可被称为共识层，目的在于使高度分散的区块链节点针对交易数据的有效性达成共识，合约层针对不同区块链类型，通过部署智能合约或共识协议的方式，帮助节点在去中心化的环境中安全地完成跨链交易共识。智能合约包含跨链具体业务逻辑(如转账、投票、借贷等)，因此合约层通常结合应用层模块处理跨链业务逻辑，用户通过应用层的跨链 SDK 与不同合约交互，智能合约自动化执行预先定义内容，并将执行结果记录在中继链的分布式账本中。另外，合约层也包含了部分账户服务功能，例如节点注册及注销、节点密钥对分发、用户配置信息修改等，同时支持跨链交易查询服务，包含交易列表信息查看、共识进度读取、跨链结果查询等。

应用层以跨链 SDK 和程序接口的形式封装了跨链交易的各种应用场景和案例，例如金融场景的链分叉、区块链代币的跨区结算等，通过接口调用合约层的服务与共识。跨链 SDK 作为统一客户端接口，为用户提供单一入口管理多链资产和操作；数据分析模块实现多链协同场景下的联合决策和资源共享，例如根据跨链交易吞吐量调节链自身的跨链消息大小、提案频率等；账户服务模块实现用户身份和地址密钥数据的跨链复用，例如使用 W3C 标准存储可验证凭证，或生成可移植的跨链传递隐私保护的 ZK 证明；用户页面为用户提供可视化，实

时监控多链事件并触发跨链操作。

在本文设计中，由于第三方中继链的参与，CrossWeave 将跨链多链之间的活性保证和多链跨链协议的原子性维护难题分别通过两个模块进行解决。异构链管理模块通过中继链对各业务链的活性监控与交易的自动转发，维护了各异构业务链之间的跨链活性。多链协调模块则通过交易缓存与全局向量时钟的定义来维护多业务链之间的跨链原子性问题。下面本章将分别详细介绍这两个模块。

4.2 异构链管理模块

异构链管理模块的核心目标是为了解决当跨链场景中异构区块链拥有不同活性假设时，跨链系统仍能保持可靠性与最终一致性的难题。该模块由两个基本组件构成：一个用于活性监控的活性标记机制，以及根据活性标记进行差异化处理的状态追踪。

活性标记。活性标记代表协议的自适应能力，来源于系统对每一条业务链在注册时的精确分类。在 CrossWeave 系统中，当每条业务链注册接入时，中继链会根据其底层共识协议的活性保证，赋予该链一个活性标记 $L \in \{strong, weak\}$。

若源链为假设强活性保证的区块链(例如，基于 BFT 共识的联盟链)，CrossWeave 将其活性标记 L 设定为 strong，即该源链正确提案的所有交易，在确定性时间上限 t 内都能被全部交付。对于源链为假设为弱活性保证的区块链(例如，基于 PoW 的区块链)，CrossWeave 将其 L 设定为 weak 标记，这代表在该类型源链正确提案交易时，系统拥有持续出块能力，但没有一个确定性的时间上限来确保该交易的交付。各源链的活性标记 L 被记录在中继链账本中，是协议选择不同执行逻辑的关键依据。

状态追踪。当中继链收到来自源链的跨链请求 X-Req 并对其中的跨链交易 xtx 达成共识后，其将向所有目标链广播跨链准备 X-Ready 消息，表明 xtx 已在中继链上链。中继链为发送至每个目标链的 X-Ready 维护一个独立的交易队列 Q，并根据目标链的活性标记 L，对队列 Q 中的任务采取不同的生命周期管理策略：

(1) 对于 weak 标记的源链: 中继链在此模式下仅负责将 X-Ready 消息分发，而不追踪其后续状态。因此，当中继链成功发送 X-Ready 后，该消息会立即从队列 Q 中移除，即跨链交易最终会被目标链交付 0，但没有一个确定性的时间上限来确保该交易的交付。

(2) 对于 strong 标记的源链: 中继链必须采用一种状态追踪机制来追踪目标链对跨链交易的交付。当中继链发送 X-Ready 后，该消息会继续保留在队列 Q 中，并为此启动一个超时计时器Δ。若在Δ超时前收到跨链确认 X-Com 消息，则队列 Q 将该消息从队列中移除，并关闭计时器Δ；若在Δ超时前仍未收到确认，中继链将会重新发送该 X-Ready 消息，并重置计时器Δ，重复此步骤直至Δ被关闭。

通过上述设计，异构链管理模块能够以一种形式化且高效的方式，适配不同区块链的活性假设，从而显著提升了整个跨链系统的健壮性与可靠性。

4.3 多链协调模块

多链协调模块的核心目标是为了在异构区块链和异步网络环境下，通过设置本地与全局向量时钟为跨链事务提供可靠且灵活的原子性保证。本模块能够通过调整中继链对业务链时钟的更新规则，以及状态锁定，来实现系统对跨链交易的原子性保证。

中继链的两阶段提交。在该模块中，中继链的协调流程都遵循同一个改进的两阶段提交(2PC)模式：

(1) 第一阶段： 当中继链收到来自源链的跨链请求 X-Req 后，验证成功后更新所有业务链的锁定状态 Lock[C]，完成共识后更新所有目标链的 G_i，随后向所有目标链广播一条跨链消息 X-Ready，表明跨链交易 xtx 已在中继链上链。各目标链在收到并验证 X-Ready 后，执行交易在本链的共识，完成共识后返回确认消息 X-Com 给中继链。

(2) 第二阶段: 中继链异步地收集所有目标链的 X-Com 消息。当所有目标链都成功返回消息，中继链广播最终的 X-Done 消息给所有源链和目标链节点，并释放所有业务链的锁定状态 Lock[C]，即本交易已完成跨链且节点可以参与新一轮跨链共识，从而确保系统状态的一致性。

中继链只在所有业务链锁定状态 Lock[C] = false 时进行 xtx 共识；若中继链在 Lock[C] = true 时接收到 xtx，即中继链仍在共识其他交易，则 xtx 会被缓存至列队 atomQ 中，等待业务链被释放，即 Lock[C] = false。

本地与全局向量时钟。为了实现 CrossWeave 系统的跨链原子性，多链协调模块针对业务链和中继

链角色，分别定义了本地时钟 Iseq 和全局向量时钟 G_i：

(1) 本地时钟 Iseq：Iseq 由各个业务链本地维护，用于记录跨链交易与区块的顺序。当源链发起交易提案时，源链的本地时钟 $Iseq' = Iseq + 1$，并对 $Iseq'$ 的交易进行上链；当目标链收到跨链请求时，若对应 ac 认证成功，则目标链的本地时钟 $Iseq' = Iseq + 1$，并对 $Iseq'$ 交易进行上链。

(2) 全局向量时钟 G_i：G_i 由中继链维护，为所有系统业务链时钟的集合，从全局角度记录所有经过中继链跨链交易的序列号。当中继链收到来自源链的的跨链请求，若对应 ac 认证成功，则在 G_i 中更新该交易下所有目标链的时钟，表明目标链该序列号下的交易可以完成跨链；当所有目标链完成跨链交易上链并返回确认消息，中继链在 G_i 中更新该交易下源链的时钟，表明源链该序列号下的跨链交易已全部完成。

保证原子性的时钟更新机制。本协议的灵活性源于中继链在处理跨链请求时，对 G_i 所采用的可配置更新规则：

(1) 强原子性保证：在此保证下，中继链将源链与目标链全局时钟

(2) G_i 的更新分开。在中继链将 xtx 上链之后，它仅更新目标链的时钟 $G_i[TC]$，并将发送给目标链的交易序列号设置为该时钟；在收到所有目标链上链的 X-Com 消息后，再更新源链时钟 $G_i[SC]$。

(3) 此机制保证了跨链交易在每条业务链及中继链上的原子性，确保了每条链交易的确定性交付，但在业务链性能低下的情况下，系统性能也会随之受到影响。

(4) 弱原子性保证：在此保证下，中继链在处理请求时，会同步更新所有业务链的逻辑时钟。在中继链将 xtx 上链之后，同时更新目标链与源链的时钟。

(5) $G_i[TC]$ 和 $G_i[SC]$，视作当前序列号交易的结束，节点可以参与新一轮跨链共识。此机制仅保证了跨链交易在中继链上的原子性，没有一个确定性的时间上限来确保各业务链子交易的交付，系统仅需要关注中继链共识的性能，牺牲了部分原子性以提升系统效率。

多链协调模块通过在协议层面提供的两阶段提交和时钟更新机制，使得 CrossWeave 系统能够灵活地适应追求高效率的资产转移和要求强原子性的复杂多链业务流程等应用场景。

5 CrossWeave 跨链协议

为了完整地阐述 CrossWeave 跨链协议的工作方式，本章节将多链跨链事务分解为若干个阶段。我们首先描述了协议的核心执行流程，该流程旨在实现弱原子性，其次，介绍基于强原子性的跨链交易协议，以及系统为了实现该性质的额外讨论。

5.1 核心流程

为了更好地介绍 CrossWeave 跨链协议的核心流程，我们将系统核心流程的时序图于图 2 展示，核心流程的伪代码于图 3 展示。当源链完成本地交易共识且生成了交易的可用性证书 ac，该笔交易即可启动 CrossWeave 跨链协议。

当所有源链节点 p_i^{SC} 收到源链用户发起跨链交易 $xtx = \{Request, c, SC, TC, ts, o\}$ 时，p_i^{SC} 首先验证 xtx 内交易是否涉及 SC 与 TC 链的账户，若 xtx 符合跨链交易要求，SC 发起原子共识协议，对 xtx 达成共识得到可用性证书 ac(图 2 步骤 1)。SC 随后启动 CrossWeave 跨链协议，读取当前 SC 的本地时钟 Iseq，发送跨链请求{X-Req, Iseq, SC, TC, xtx, ac}给中继链所有节点(图 2 步骤 2)。

当中继链节点 p_i^{RC} 收到跨链请求{X- Req, Iseq, SC, TC, xtx, ac}消息时，验证 SC 与 TC 是否在系统中注册，ac 的有效性，以及是否满足同步条件 $Iseq = G_i[SC]$。验证通过后，中继链对 xtx 达成共识。随后，中继链将 xtx 分解为源链子交易 tx_s 和目标链子交易 tx_t，更新目标链的时钟 $G_i[TC] \leftarrow G_i[TC] + 1$，并记录交易凭证 $Ready_{TC}[G_i[TC] \leftarrow Hash(tx_t)]$。最后(图 2 步骤 3)，发送跨链子交易{X-Ready, $G_i[TC]$, tx_t, ac}给目标链所有节点(图 2 步骤 4)。

针对任一目标链节点 p_i^{TC}，当其收到来自 p_i^{RC} 的 $f+1$ 个匹配的{X-Ready, gseq, tx_t, ac}消息时，验证 ac 的有效性；若收到的序列号 gseq 等于当前 TC 的本地时钟 $Iseq+1$，则发起目标链原子共识协议，对子交易 tx_t 达成共识，更新计数器 $Iseq \leftarrow gseq$ (图 2 步骤 6)，并发送确认消息{X-Com, TC, Iseq, Hash (tx_t) }给中继链所有节点(图 2 步骤 7)。针对每一个目标链 TC，其活性保障机制取决于该链的活性标记 L[TC]。若标记为 strong，则中继链在发送 X-Ready 时会启动计时器 Δ(图 2 步骤 5)并将 X-Ready 存入交易队列 Q；若标记为 weak，则在中

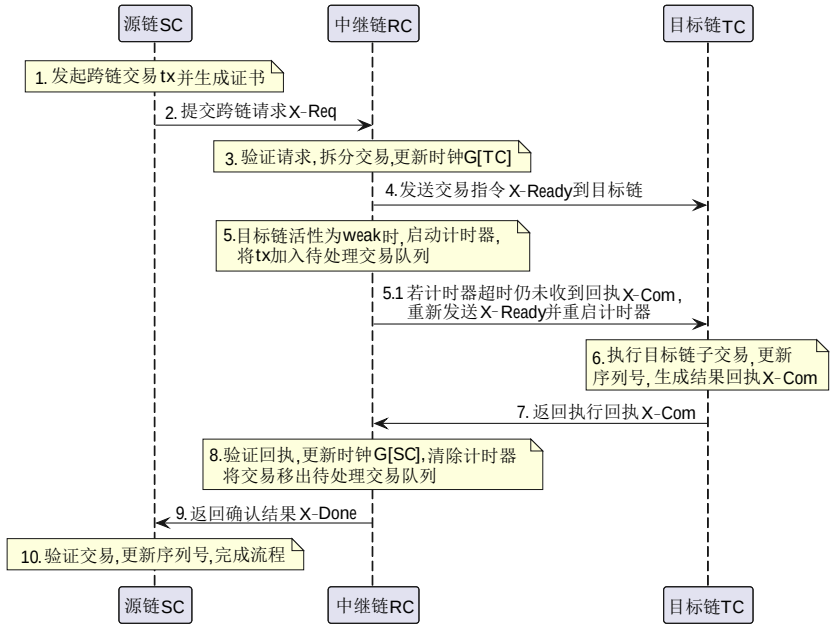


图 2 系统核心流程时序图

初始化

- 中继链节点 $p_i^{\mathcal{RC}}$: 向量时钟 $G_i \leftarrow \{\forall \mathcal{C} : G_i[\mathcal{C}] = 0\}$; 交易队列 $Q_i \leftarrow \varnothing$; 子交易哈希缓存 $Ready_{\mathcal{C}} \leftarrow \varnothing$, 活性配置表$L[\mathcal{C}] \leftarrow \varnothing$
- 业务链节点 $p_j^{\mathcal{C}}$: 本地跨链序列号 $lseq \leftarrow 0$

客户端c

- 发送跨链交易$(\text{Request}, c, \mathcal{SC}, \mathcal{TC}, ts, o)$给源链$\mathcal{SC}$，其中$ts$表示时间戳，$o$表示交易的具体内容

源链节点$p_i^{\mathcal{SC}}$

- 当收到来自客户端c的跨链交易xtx=$(\text{Request}, c, \mathcal{SC}, \mathcal{TC}, ts, o)$时，对$xtx$进行共识，等待$xtx$上链，得到可用性证书$ac \leftarrow ACGen(xtx)$，并发送$(X\text{-}Req, lseq, \mathcal{SC}, \mathcal{TC}, xtx, ac)$给$\mathcal{RC}$
- 当收到 $\mathcal{RC}$ 的$f+1$个匹配的 $(X\text{-}\text{Done}, gseq, tx_s)$ 消息时, 若 $gseq = lseq + 1$，则执行 tx_s 并更新 $lseq \leftarrow gseq$

目标链节点$p_i^{\mathcal{TC}}$

- 当收到$\mathcal{RC}$的$f+1$个匹配的$(X\text{-}\text{Ready}, gseq, tx_t, ac)$消息时, 若验证$ac$有效且$gseq = lseq + 1$，则执行$tx_t$
- 在tx_t上链后，更新$lseq \leftarrow gseq$，并发送$(X\text{-}\text{Com}, \mathcal{TC}, lseq, \text{Hash}(tx_t))$给$\mathcal{RC}$

中继链节点$p_i^{\mathcal{RC}}$

- 当收到来自源链节点p_j的跨链请求$(X\text{-}Req, lseq, \mathcal{SC}, \mathcal{TC}, xtx, ac)$时:
  - 确认$\mathcal{SC}$ 与 $\mathcal{TC}$ 是合法区块链，ac 是有效的，且 $lseq = G_i[\mathcal{SC}]$
  - 验证通过后，在中继链上对 xtx 请求进行共识，等待其上链
  - 将 xtx 分成$\mathcal{SC}$ 上的子交易 tx_s 和$\mathcal{TC}$ 上的子交易 tx_t
  - 更新目标链向量时钟$G_i[\mathcal{TC}] \leftarrow G_i[\mathcal{TC}] + 1$，$gseq \leftarrow G_i[\mathcal{TC}]$，更新子交易哈希缓存$Ready_{\mathcal{TC}}[gseq] \leftarrow \text{Hash}(tx_t)$
  - 发送交易 $tx \leftarrow (X\text{-}\text{Ready}, G_i[\mathcal{TC}], tx_t, ac)$给目标链$\mathcal{TC}$的所有节点
  - 若 $L[\mathcal{TC}] = \text{strong}$，将指令$tx$加入Liveness队列$Q_i$并为其开启计时器$\Delta$
- 当Δ超时且未收到对应的$(X\text{-}\text{Com})$时，从Q_i中获取tx重新发送给目标链$\mathcal{TC}$，并重置计时器Δ
- 当收到来自目标链节点p_j的跨链确认$(X\text{-}\text{Com}, \mathcal{TC}, lseq, h)$时:
  - 确认TC 是合法区块链，$lseq = G_i[TC]$，且$Ready_{\mathcal{TC}}[lseq] = h$，若该指令存在于$Q_i$中，则将其移除并停止计时器
  - 更新源链的向量时钟$G_i[\mathcal{SC}] \leftarrow G_i[\mathcal{SC}] + 1$，从分解的交易中获取 tx_s，发送跨链结果$(X\text{-}\text{Done}, G_i[\mathcal{SC}], tx_s, ac)$给源链$\mathcal{SC}$所有节点

图 3 跨链共识协议核心流程

继链成功发送 X-Ready 后，该消息会立即从队列 Q 中移除。当Δ超时且未收到 X-Com 时， p_i^{RC} 会重新转发 Q 中的 X-Ready(图 2 步骤 5.1)并重置计时器Δ，重复此步骤直至Δ被关闭。当 p_i^{RC} . 收到跨链确认 {X-Com, TC, lseq, h}消息时，验证 lseq = G_i[TC]，以及 h 与本地缓存 $Ready_{TC}$[lseq]是否匹配；若通过

验证，更新源链的向量时钟 $G_i[SC] \leftarrow G_i[SC]+1$，并生成最终的 X-Done 消息发送给源链，同时将事务从交易队列 Q 中移除(图 2 步骤 8, 9)并关闭Δ。

针对任一源链节点 p_i^{SC}，与目标链收到跨链子交易的流程相似，当其收到来自 p_i^{RC}. 的 f+1 个匹配的{X-done, tx_s, gseq, σ, ac}消息时，验证 ac 的有效性；若收到的序列号 gseq 等于当前本地时钟 lseq+1，则发起源链原子共识协议，对子交易 tx_s达成共识，并更新计数器 lseq ← gseq，完成整个跨链流程(图 2 步骤 10)。

正如上述流程所揭示的，本协议的异构链管理机制与原子性机制是分离的，即无论系统运行在何种原子性模式下，活性保障机制都独立地发挥作用。当中继链向目发送 X-Ready 时，它会检查该目标链的活性标记 L[TC]，只有当标记为 strong 时，才会启用由交易队列 Q_i和计时器Δ构成的状态追踪与超时重试机制。这种设计将核心的原子性保证与网络的容错保障解耦，使协议在保证灵活性的同时，具备了在复杂环境下的高度健壮性。

5.2　强原子性的额外约束

在此节，我们介绍 CrossWeave 跨链协议实现系统强原子性额外约束，即对所有跨链交易实现源链和目标链的原子性，具体流程的伪代码于图 4 中展示。具体来说，中继链需要强制让每条业务链跨链交易的全局时钟严格增长，即目标链的交易完成前不允许其作为源链提交新的跨链交易，反之亦然。

初始化

- 向量时钟$G_i \leftarrow \{\forall \mathcal{C} : G_i[\mathcal{C}] = 0\}$；子交易哈希缓存$Ready_{\mathcal{C}} \leftarrow \varnothing$；待确认的交易列队$Q \leftarrow \varnothing$；待接收的交易列队$atomQ \leftarrow \varnothing$
- $Lock[\mathcal{C}] \leftarrow \{\forall \mathcal{C} : Lock[\mathcal{C}] \leftarrow \text{false}\}$

中继链节点$p_i^{\mathcal{RC}}$

- 当收到跨链请求(X-Req, $lseq, \mathcal{SC}, \mathcal{TC}, xtx, ac$)时:
  - 确认$\mathcal{SC}$ 与$\mathcal{TC}$ 是合法区块链，ac是有效的，且$lseq = G_i[\mathcal{SC}]$
  - 当收到$f+1$个匹配的(X-Req)且($Lock[\mathcal{SC}]$ or $Lock[\mathcal{TC}]$)=true时，$atomQ \leftarrow xtx$
  - 当收到$f+1$个匹配的(X-Req)且$Lock[\mathcal{SC}] = Lock[\mathcal{TC}] = \text{false}$时:
    - $Lock[\mathcal{SC}] \leftarrow \text{true}$，$Lock[\mathcal{TC}] \leftarrow \text{true}$，$Q \leftarrow xtx \cup atomQ$，按照中继链原子广播对$xtx$进行共识，等待$xtx$上链
    - 将xtx分成$\mathcal{SC}$ 和$\mathcal{TC}$ 上的子交易tx_s和tx_t
    - $G_i[\mathcal{SC}] \leftarrow G_i[\mathcal{SC}]+1$，$G_i[\mathcal{TC}] \leftarrow G_i[\mathcal{TC}]+1$，$gseq \leftarrow G_i[\mathcal{TC}]$
    - $Ready_{\mathcal{SC}}[gseq] \leftarrow \text{Hash}(tx_s)$，$Ready_{\mathcal{TC}}[gseq] \leftarrow \text{Hash}(tx_t)$
    - 生成tx_t的签名σ_t，发送(X-Ready, $tx_t, gseq, \sigma_t, ac$)给目标链所有节点
    - 若$L[\mathcal{TC}] = \text{strong}$且$\Delta = \varnothing$，将 xtx 加入 Liveness队列 Q 并开启计时器Δ
  - 当Δ超时且未收到(X-Com)时，重新转发(X-Ready, $tx_t, gseq, \sigma_t, ac$)给目标链所有节点，重置Δ
  - 当收到跨链确认(X-Com, $\mathcal{TC}, lseq, h$)时:
    - 确认$\mathcal{TC}$ 是合法区块链，$lseq = G_i[\mathcal{TC}]$，且$Ready_{\mathcal{TC}}[lseq] = h$
    - 将xtx从Q中移除，生成tx_s的签名σ_s，发送跨链结果(X-Done, $tx_s, G_i[\mathcal{SC}], \sigma_s, ac$)给源链所有节点,
    - $Lock[\mathcal{SC}] \leftarrow \text{false}$，$Lock[\mathcal{TC}] \leftarrow \text{false}$
    - 从$atomQ$中取出下一个可处理的请求（如有）并开始处理

图 4　协议强原子性的额外约束

在 CrossWeave 跨链协议核心流程的基础上，当中继链 RC 收到跨链请求时，它必须检查此请求的 SC 和 TC 当前都未被锁定于其他任何未完成的跨链事务中，即当且仅当所有业务链处于空闲状态时，才能接收并处理该跨链请求，同时，RC 将这些链标记为锁定状态，直至该轮事务最终完成。若不满足该要求，RC 将跨链请求缓存于待确认原子性等待队列 atomQ 中，等待所有业务链再次处于空闲状态。

另外，当 RC 更新业务链向量时钟时，为保证强原子性，必须采用同步更新策略。即，在中继链对一个无冲突的跨链请求进行协调时，它会在向目标链分发 X-Ready 指令之前，原子地同步更新所有参与方(源链和目标链)的向量时钟。这个预先分配针对目标链的更新后时钟值，将作为序列号随 X-Ready 消息发送，为目标链提供一个严格的、全局一致的执行顺序。

通过上述两个额外约束，使得系统各业务链在任何时间点最多只参与一个跨链事务，使得每条业务链跨链交易的全局时钟严格增长，实现了跨链交易源链与目标链的强原子性。然而，这种强原子性的保证是以牺牲系统并发性和吞吐量为代价的，因此适用于对一致性要求极高而对性能要求相对宽松的特定场景，例如处理金融场景的链分叉、多条区块链代币之间的跨区结算等。

如上述伪代码所示，本协议的多链协调模块在强原子性模式下的核心是状态锁定与同步时钟更新机制。协议通过 Lock 状态来显式地追踪和管理每个业务链的状态。当中继链收到新请求时，会首先检查其所有参与方是否都处于空闲状态，并将冲突的请求放入 atomQ 等待队列。对于无冲突的请求，则在锁定其参与方后，同步更新所有相关方的向量时钟 G_i，并将更新后的时钟值作为序列号分发。

5.3 安全性分析

在本节，我们介绍 CrossWeave 跨链协议方案的信任假设，论证在设定的安全模型下方案如何保证活性与原子性。

定理 1(强跨链活性).如果一笔有效的跨链交易 xtx 已经由其源链 SC 中的诚实节点正确发起(即生成了有效的可用性证书 ac 并向中继链提交了 X-Req)，那么最终，所有参与该交易的诚实节点(在源链 SC 和所有目标链 TC 上)都会执行其对应的子交易。

协议通过 4 个关键阶段保证活性：(1)源链诚实节点提交 X-Req 至中继链；(2)中继链通过原子广播交付请求并发送 X-Ready；(3)目标链执行交易，其中强活性链依赖中继链超时重试机制，弱活性链依赖自身共识保证；(4)通过 X-Com 和 X-Done 完成全流程确认。详见附录 C。

定理 2(弱跨链活性).如果一笔有效的跨链交易 xtx 已经由其源链 SC 中的诚实节点正确发起，那么最终，源链 SC 和所有目标链 TC 上至少各有一个诚实节点会执行其对应的子交易。

由定理 1 可得出：若所有诚实节点都会执行交易，则至少有一个诚实节点会执行交易。因此弱跨链活性是强跨链活性的直接推论。详见附录 C。

定理 3(跨链弱原子性).对于任意两笔源自同一源链 *SC* 的跨链交易 xtx_1 和 xtx_2，如果 SC 上的一个诚实节点先执行 xtx_1 的源链子交易，再执行 xtx_2 的源链子交易，那么对于它们共有的任何目标链 TC，TC 上的诚实节点都不会在执行 xtx_1 的目标链子交易之前执行 xtx_2 的目标链子交易。

中继链的原子广播协议保证 X-Req 的全局顺序一致性，递增的全局时钟 G_i[TC]与目标链本地时钟 lseq 的同步验证机制(gseq = lseq + 1)确保目标链严格按序执行交易。详见附录 C。

定理 4(跨链强原子性).对于任意两笔共享至少一个业务链的跨链交易 xtx_1 和 xtx_2，如果某个参与链 *A* 上的一个诚实节点先执行其在 xtx_1 中的子交易再执行其在 xtx_2 中的子交易，那么在其他任何参与链 B 上，没有一个诚实节点会先执行其在 xtx_2 中的子交易，然后再执行其在 xtx_1 中的子交易。

强原子性模式通过 Lock 状态和 atomQ 等待队列实现全局互斥协调。中继链在处理 xtx_1 时锁定所有参与链(Lock[C] = true)，冲突交易 xtx_2 被缓存至 atomQ，仅在 xtx_1 完全结束并释放所有锁(Lock[C] = false)后才能执行。详见附录 C。

6 工程实现

本节旨在阐述 CrossWeave 跨链协议的核心工程实现。我们将详细介绍异构链管理与多链协调两大核心模块的具体实现方式，包括关键的软件设计、与现有区块链系统的接口对接方法，并最终展示这些模块如何协同工作以保障跨链交易的完整生命周期。

6.1 技术实现难点分析

在工程实践中，现有主流中继链跨链方案在处理异构链互操作时面临两个主要挑战：

在异构链兼容性方面，Polkadot 的共享安全模型要求平行链具备确定性的最终性，弱活性链必须通过特定的外部桥接才能接入生态。Cosmos 的 IBC 协议同样针对强活性链优化,对于弱活性链需部署独立的“锚定区”并等待大量区块确认(如上百个区块),这不仅引入额外的信任假设和基础设施成本,还导致显著的交易延迟。这些方案缺乏协议层面对不同活性特征的统一处理框架。

在原子性协调方面，Polkadot 的 XCMP 协议遵循“即发即忘(fire-and-forget)”模式，协议本身不提供确认或结果返回机制。Cosmos 的 IBC 本质上是异步的点对点消息传输层，仅保证数据的可靠传递，无法处理多链事务的原子性协调。两者均将多链事务的原子性协调完全交由应用层实现，增加了应用开发的负担和潜在风险。

本方案的异构链管理机制通过活性标记在协议层直接兼容不同的最终性假设，提供了一个更通用、高效且低信任依赖的接入框架。同时，本方案在协议层处理了多链协调问题，并提供了可选的原子性服务(强/弱原子性)，从而在工程层面降低了应用开发复杂度。

6.2 关键实现

针对前文介绍的异构链管理模块和多链协调模块，我们在该章节从工程角度介绍针对这两个模块的关键实现。

原型实现。在我们的实现中，中继链是基于Dyno共识协议构建的。同时，系统支持并对接了多种业务链，包括Dyno、FISCO BCOS和CITA。值得注意的是，作为业务链的Dyno与作为中继链的Dyno，其底层技术协议相同，但其在跨链架构中的角色与配置不同：中继链作为跨链协调的中枢，运行着本章所述的核心协调逻辑；而业务链则作为参与方，负责执行最终的业务指令。

异构链适配Relayer接口。为了兼容不同区块链，我们设计了Relayer软件接口。它作为连接中继链与各个业务链的“驱动程序”，要求每种被接入的区块链都提供其具体实现。Relayer接口定义了标准化的操作：

(1) CrossCall: 执行中继链两阶段提交中的第一阶段，即在目标链上预执行或锁定资源。

(2) CommitCall: 执行中继链两阶段提交中的第二阶段，最终确认或取消交易。

(3) IsSeqProcessed: 检查具有特定序列号的指令是否已被目标链处理。

该接口的实现，本质上是调用目标链的RPC(远程过程调用)服务。例如，CrossCall会调用业务链上特定的智能合约。中继链与业务链之间的通信则由陆羽跨链协议提供安全保障，确保消息的合法性与互信。

陆羽跨链协议主要聚焦于链与链之间的安全数据传递，并不涉及跨链事务的全局管理和一致性控制。该协议具备较强的通用性，通过标准化的接口定义解决了不同链类型在通信层面的差异，实现对多种区块链平台的兼容。CrossWeave协议在陆羽协议的通信与安全能力之上构建业务层逻辑，通过接入陆羽网关完成跨链通信，利用其建立的安全传输通道和统一接口标准来实现目标链的路由与交互，在此基础上增加了面向业务的事务处理规则。陆羽跨链协议为跨链通信提供了安全稳定的底层通道，CrossWeave协议在此基础上构建了面向业务的整体流程控制与事务一致性保障，实现了从安全通信到跨链业务逻辑的完整体系。

状态追踪。为了实现对不同活性链的差异化管理，我们设计了TransactionQueue模块。在Go语言的实现中，它是一个由互斥锁(sync.Mutex)保护的映射，为每条链维护一个独立的交易队列。

当与活性保证为strong的链交互时，TransactionQueue会为其启动一个计时器。若在超时前未收到确认，计时器会触发重试逻辑，该逻辑会首先调用Relayer的IsSeqProcessed方法进行检查，防止重复执行。若检查发现交易未被处理，则会重新调用CommitCall。此设计确保了即使在目标链确认消息丢失或网络不稳定的情况下，交易也能最终完成。

CrossWeave协议在设计上支持弱原子性和强原子性两种模式，这通过调整中继链的协调逻辑来实现。

强原子性保证。该保证的实现核心是引入了链状态锁定机制。如协议伪代码图4所示，中继链会维护一个全局的Lock状态表。在处理跨链请求时：

(1) 检查并加锁：检查事务所涉及的所有链是否处于Lock = false状态。若是，则将它们全部置为Lock = true并开始处理。

(2) 排队等待：若任一链已被锁定，则将新请求放入一个原子性等待队列(atomQ)，直至锁被释放。释放锁：当整笔跨链事务最终完成(X-Done)，中继链会释放相关链的锁，并从atomQ中取出下一个请求进行处理。

这种显式的锁定机制，将所有存在资源冲突的跨链事务强制串行化，从而实现了全局总排序。

6.3 完整流程

异构链管理模块和多链协调模块并非独立运行，而是在跨链交易的生命周期中紧密协同。下面以一次完整的跨链调用为例，从工程角度展示其协同流程：

(1) 请求接收与初始化：系统解析HTTP请求的参数并存证至中继链中。多链协调模块在此阶段检查Lock状态，若锁定则将交易缓存至atomQ队列。

(2) 序列号分配：为源链和所有目标链分配序列号，这是实现原子性排序的第一步。

(3) 入队管理：根据每个目标链的活性标记，将拆分后的子交易加入TransactionQueue中。

(4) 并发执行：系统启动多个Go线程(goroutine)，并行地通过各链对应的Relayer实例调用CrossCall方法。在此阶段异构链管理模块监控强活性链的响应状态，多链协调模块检查序列号条件。

(5) 结果收集与决策：主线程等待所有CrossCall完成。若出现任何失败，则全局决策为回滚；若全部成功，则决策为提交。多链协调模块在此阶段检查并更新序列号。

(6) 最终确认：根据决策，再次通过 Relayer 调用 CommitCall。对于标记为强活性的链，CommitCall 成功后任务才会从 TransactionQueue 中移除。若 CommitCall 失败或超时，TransactionQueue 的重试机制将自动触发，确保指令最终送达。在此阶段多链协调模块更新序列号，异构链管理模块处理交易缓存队列。

(7) 完成记录：中继链记录最终状态和延迟等性能指标，完成整个闭环。在此阶段多链协调模块释放参与链锁定状态。

6.4 中继链资源消耗分析

中继链作为跨链系统的协调枢纽，需要同时维护多条业务链的状态信息。随着接入链数的增长，中继链的资源消耗变化主要体现在存储、计算和网络通信三个方面。

存储开销方面，中继链需要为每条业务链维护的核心状态包括：全局向量时钟 G_i 中的序列号(每条链一个整数)、活性标记 L(每条链一个布尔值)以及针对强活性链的交易队列 TransactionQueue。向量时钟的存储空间复杂度为 O(N)，其中 N 为接入的业务链数量，额外存储开销较小。交易队列的存储开销与并发跨链交易数量相关，取决于具体的应用场景和业务负载。

计算开销方面，中继链的主要计算任务包括可用性证书 ac 的验证、向量时钟的更新以及交易路由决策。可用性证书验证的复杂度主要取决于签名算法类型(如 BLS 聚合签名验证复杂度为 O(1))。向量时钟更新操作本质上是整数读写，在弱原子性模式下仅需更新涉及的源链和目标链时钟(复杂度为 O(1+|TC|))，在强原子性模式下需额外检查所有链的锁定状态(复杂度为 O(|SC|+|TC|))，计算开销与跨链交易数量以及跨链交易涉及的业务链数量成正比，进而受到接入业务链数量 N 的间接影响。

网络通信开销方面，中继链需要与各业务链进行消息交互，通信量与跨链交易数量以及跨链交易涉及的业务链数量成正比。

综合来看，中继链各种资源消耗与接入链数量呈次线性或线性关系，在典型应用场景下资源压力可控。

7 测试与评估

本节对所设计的跨链机制进行实验测试与性能评估。实验在部署 8 个中继节点的多链环境中完成，所涉及链包括 FISCO、Dyno 和 CITA。在多链跨链场景中，跨链交易性能的稳定性与延迟瓶颈往往直接决定其可用性与扩展性。为验证本文提出的跨链机制在不同异构链组合、交易规模以及部署模式下的表现，本节设计了覆盖多场景的系统性实验。

实验结果表明：(1)中继链在多条件下的处理时间保持在 500 ms 稳定区间，而整体延迟则主要受制于目标链的出块间隔与确认策略；(2)交易体积的增大会对延迟产生一定的影响，但整体趋势保持稳定；(3)不同部署模式对性能的影响较小，多机部署相较于单机部署延迟仅略微增加。本节将按照实验场景依次给出方法与结果分析。

7.1 实验环境与配置

实验环境基于本地服务器初始部署，后续测试包括单机与多机场景。测试服务器的物理配置如下：

处理器为 Intel Xeon Gold 5218，主频 2.30 GHz，分配 4 核心，系统总内存 8 GB，可用约 3.2 GB，运行 Ubuntu 18.04 LTS 操作系统，Java 运行环境版本为 OpenJDK 1.8.0，Redis 版本为 4.0.9，Docker 版本为 20.10.21，JQ 工具版本为 1.5。

为验证所提出的异构链活性管理机制在不同活性假设下的适应性与鲁棒性，我们在多链环境中进行了系统性的实验评估。链节点初始部署在同一台本地服务器上，以保证环境可控与测试公平性，并在后续实验中扩展为多台物理机部署以分析其对性能的影响。

我们在部署的多链环境中进行了系统性的实验评估。系统涉及三条链：Dyno、FISCO 和 CITA，中继链为 Dyno。我们实现并部署了一个异构链跨链交易平台，所使用的区块链平台及节点部署如下：

Dyno：作为中继链与业务链各部署一条实例，各实例包含 8 个共识节点，为强活性链；

FISCO：部署 4 个共识节点，支持交易确认返回，为弱活性链；

CITA：部署 4 个共识节点，默认不支持交易确认返回，为强活性链。

其中，Dyno 作为跨链中继链，统一接收用户请求并根据目标链的活性假设进行跨链转发处理。

在上述三种类型的链中，FISCO 和 Dyno 均支持交易提交后立即返回交易执行结果，具备同步确认能力。CITA 则不返回交易确认反馈，仅返回交易哈希，系统需依赖外部轮询确认机制实现异步确认。针对该问题，我们设计并实现了基于定时器的

交易轮询机制。该机制在中继链将交易发送至目标链后，启动 200 ms 间隔的定时查询策略，直到交易确认或轮询超时为止。

7.2　跨链事件分析

为评估不同链组合下的跨链处理性能，我们设计并执行了一系列跨链延迟实验。实验场景覆盖了由 FISCO 作源链和 Dyno 作源链两大种路径。每个场景下，系统连续执行 100 次跨链交易，测量并记录其整体执行时间，取其平均值作为最终统计指标。为进一步测试系统对大体积交易的处理能力，我们分别构造了两种交易负载：原始交易大小约为 2^9 字节(512B)，扩展交易大小约为 2^{18} 字节(256KB)。其余配置保持不变，确保实验结果具备可比性。

为衡量完整跨链交易所需的时间，本节测试了从中继链接收到跨链交易请求起，到目标链成功执行该交易并返回响应为止的总时间。最终统计结果如图 5 所示。图中可以看出，Dyno 与 FISCO 之间的跨链平均延迟在 1.3 s ~ 2.5 s 范围内，不同交易大小造成的延迟增加值在 26 ms ~ 638 ms 之间，波动范围较小。这主要得益于其共识机制支持快速交易确认。相比之下，CITA 目标链的延迟均在 5.2 s ~ 5.9 s 区间。这是由于 CITA 缺乏交易确认反馈机制，需要中继链通过轮询方式确认交易上链状态，且

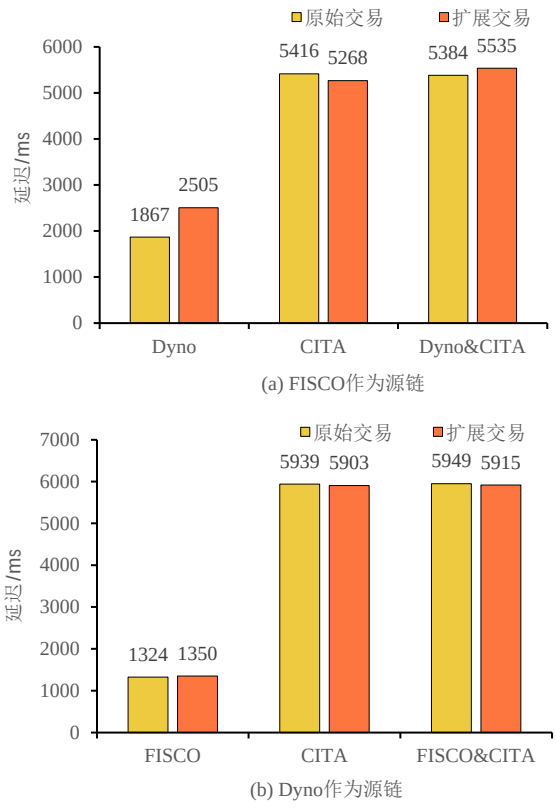


图 5　跨链交易执行时间对比

CITA 的出块时间为 2 s ~ 3 s[10]，其较长的出块时间导致了和 CITA 相关的跨链交易都产生了较高的延迟。

进一步比较发现，多链组合路径的总延迟与单一目标链为 CITA 的情形差距并不显著。这是因为总延迟往往由耗时最长的目标链阶段决定，当路径中存在 CITA 这类确认周期较长的链时，额外增加低延迟链对整体延迟影响有限。只有当增加的多条链延迟相近且均较高时，累积效应才会更明显。

进一步观察不同交易大小下的延迟变化，可以发现交易体积的增加对跨链性能具有一定影响。从整体趋势上看，交易体积的增大会对跨链时间产生一定影响，例如在 FISCO 到 Dyno 场景下，延迟提升了 638 ms。这是由于较大的交易体积增加了数据在源链、中继链与目标链之间的传播与处理开销。然而，在目标链是 CITA 的相关路径中，跨链延迟反而略有下降，如 FISCO 到 CITA 路径从 5416 ms 降低至 5268 ms，这是由于 CITA 使用异步返回机制，跨链成功主要依赖中继链主动轮询确认交易上链状态。在高负载下，由于 CITA 出块间隔仍为 2 s ~ 3 s，轮询机制的调度相对更趋均衡，出现一定程度的统计抖动，从而掩盖了延迟增长趋势。

7.3　中继链时间分析

在本实验中，我们进一步研究了交易体积扩展对中继链处理延迟的影响。本节设置中继链处理时间为从中继链接收到交易请求起，到将交易转发至目标链前的处理时间。实验设置原始交易体积约为 2^9 字节，为测试系统在大体积条件下的稳定性和处理性能，我们构造了含有 2^{18} 字节的扩展交易请求，并分别测试了原始交易与扩展交易在中继链的处理时间。结果如图 6 所示。

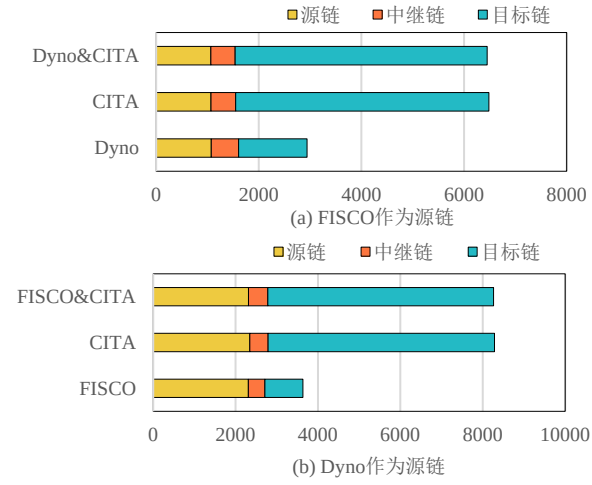


图 6　中继链执行时间对比

从图 6 中可以看出，随着交易体积的增大，大多数跨链路径中继链的处理时间呈现出轻微上升的趋势。FISCO 作源链时，原始交易的中继链延迟在 474 ms ~ 534 ms 之间，扩展交易在 554 ms ~ 580 ms 之间，增加幅度为 20 ms ~ 99 ms。Dyno 作源链时，中继链延迟在 401 ms ~ 470 ms 与 478 ms ~ 499 ms 之间，增加幅度为 77 ms ~ 88 ms。各场景扩展交易相较原始交易的中继链延迟增幅在 4.12%至 20.58%区间。这是由于中继链在处理跨链交易时，需要对每条目标链构造包含完整负载信息的交易请求结构。在该结构中，包含一段可调节体积的填充字段，该字段虽未参与实际网络传输，但在入队处理前已被作为有效负载的一部分封装并复制到中继链内部的数据结构中。由于中继链需为每条目标链分别构造完整的交易结构并执行入队操作，该过程中涉及大量内存分配与字符串复制操作。当填充字段体积较大时，这些操作会显著增加 CPU 和内存资源的消耗，从而延长整体处理耗时。

由于本文实验都是在本地服务器运行，因此扩大交易大小后，并不会带来额外的通信开销。可以预期的是如果跨链交易是在广域网进行，在交易较大时，延迟将会进一步增大。

7.4 延迟分解分析

为进一步明确异构链跨链交易过程中各阶段的时间开销，对跨链延迟进行了分解。实验基于前述部署环境，分别评估源链、中继链与目标链在整个跨链过程中的延迟时间。

源链时间定义为事件发生时刻到事件转发至中继链的时刻，但是源链时间的获取方式在两条链中存在一定差异。由于 FISCO 链采用基于订阅的事件推送模型，其事件日志中不直接包含高精度的事件触发时间。所以本文在监听到事件所在区块后，读取该区块的区块头时间戳，将事件时间近似视为前一个区块生成时刻，通过这种方式估算事件的实际发生时间，作为源链延迟计量的起始时间。Dyno 链通过 REST 接口轮询获取事件记录，返回数据中直接包含事件提交的时间戳，系统可据此精确提取事件发生时间，作为起始时间。

中继链时间为中继链在接收到源链事件数据后至将跨链交易提交给目标链之间的处理时间。起始时间为事件数据到达中继链的时间，终止时间为该节点向目标链共识节点发送跨链交易的时间。

目标链时间定义为跨链交易到达目标链后至交易最终确认之间的时间间隔。其统计方法为第 7.2 节得到的跨链总时间与中继链时间的差值，即为目标链时间。

基于上述定义，本文在不同跨链路径下分别测量了三类延迟，实验结果如图 7 所示。FISCO 作为源链时的延迟集中于 1.7 s 左右，而 Dyno 作为源链时，延迟明显增大至 2.31 s ~ 2.35 s 左右。主要原因是两条链的事件时间采集机制差异：FISCO 使用区块时间减去出块时间值估算事件时间，整体偏差小，采集近似及时，但不具备高精度；Dyno 使用实际事件触发时间戳，更接近真实提交时间，且因为 Dyno 系统中链上事件结果需从数据库接口轮询获取，从事件触发到事件被监听系统识别的整体链内处理时间更长，导致记录时间滞后。此外，目标链数量的增加并未显著影响源链延迟，这说明源链处理过程主要由事件触发及采集逻辑决定，与后续跨链路径相关性较小。

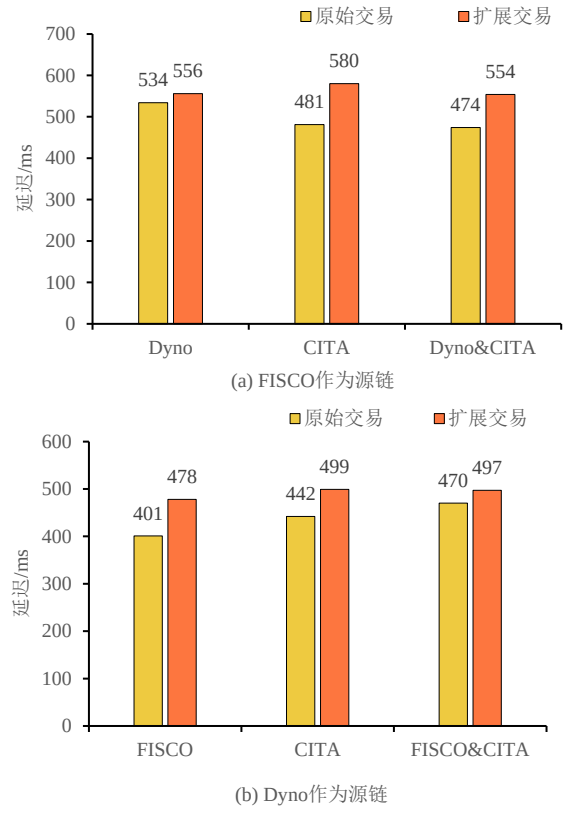


图 7 跨链延迟分解

中继链延迟在不同跨链路径中表现相对稳定，范围在 401 ms ~ 534 ms，说明在测试条件下中继链的消息调度与交易转发机制对整体性能的影响有限。

在多数跨链路径中，目标链阶段的延迟显著高于源链和中继链阶段，占比 47%～78%，在总耗时中占据主导地位。作为跨链事务的最终执行者，目标链必须完成业务合约调用、参数校验、状态更新、共识验证与区块写入等完整处理链路。高并发请求下，频繁的状态更新会导致磁盘 I/O 竞争，LevelDB 的持久化存储的写操作在高并发下会出现队列阻塞。且中继链会等待所有目标链处理完成后再统一提交确认或回滚，使得最慢的目标链会拖慢整个跨链事务，因此目标链延迟在三类阶段中普遍最高。

从图中进一步可以看出，当目标链为 CITA 时，延迟显著升高。在 FISCO 作为源链的场景中，CITA 目标链的延迟为 4934 ms，而 Dyno 目标链的延迟为 1333 ms，两者相差 3601 ms，相差约 2.7 倍。在 Dyno 作为源链的场景中，CITA 目标链延迟达到 5496 ms，FISCO 目标链延迟为 923 ms，差异为 4573 ms，相差约 5 倍。该结果与系统设计机制高度一致：由于 CITA 默认不支持交易确认返回，需依赖中继链以轮询方式确认交易是否成功上链。此外，CITA 区块链平台采用的共识机制具有较长的出块周期，进一步拉长了整体交易确认时间。

相比之下，目标链为 Dyno 或 FISCO 的路径表现出更低的延迟。这是由于两条链支持交易同步确认机制，使中继链在发送交易后可快速获知交易执行结果，从而缩短了响应时间。进一步对比目标链为 Dyno 和 FISCO 两种场景可发现，FISCO 作为源链时，目标链为 Dyno 的延迟为 1333 ms，而在 Dyno 作为源链时，目标链为 FISCO 的延迟仅为 923 ms，两者相差 410 ms。Dyno 作目标链的延迟明显高于 FISCO 作为目标链的延迟。这是因为 Dyno 在交易确认阶段引入了门限签名机制，在一个交易在确认之前，必须收集来自至少 $f+1$ 个共识节点的有效部分签名，并在代理节点处完成签名聚合。这一过程显著增加了网络通信轮数与消息处理复杂度。此外，Dyno 相较于 FISCO 多了一层代理节点，交易在到达目标链共识节点之前，需先经过代理节点的中转与预处理，从而增加了交易在路径中的传输时延与调度开销。而 FISCO 采用更为简化的共识流程，允许中继链直接与共识节点交互提交交易并接收确认响应，因此 FISCO 能更快速完成交易打包与确认。

由于系统架构设计的特性，异构链管理模块与多链协调模块在实际实现中存在较高的耦合度。这两个模块的核心逻辑在跨链请求处理的关键路径中连续执行。在跨链请求处理过程中，链类型识别与交易队列管理等操作是紧密衔接、连续执行的原子化流程，中间不存在可独立标记和测量的时间段。若拆分这些高度耦合环节，将会破坏原有流程结构，使测量结果失去准确性和可比性。本节的延迟分解以端到端延迟为核心，其数据本质上包含了上述两个模块的综合耗时，可客观反映模块协同工作的整体性能，当前测量方式更符合实际应用中用户对跨链延迟的感知。因此在当前实验设计下，端到端视角的延迟分析比单独拆分模块耗时更具参考价值。

为评估系统在不同部署模式下的跨链通信性能，我们设计了对比实验，分析在单机部署与多机部署场景中跨链交易延迟的差异。我们部署了一个具有 8 个中继节点的中继链网络。将中继节点分布在四台物理服务器上，网络连接为局域网。每轮实验从源链提交跨链交易，请求中继链转发至目标链，分别记录源链时间、中继链处理时间、完整跨链交易处理时间。

实验结果如图 8 所示，从实验结果来看，单机部署和多机部署在跨链交易总延迟上差异较小。FISCO 作源链时，总延迟增加范围在 109 ms～520 ms 之间，相对提升 2%～17.2%。Dyno 作源链时，延迟变化较小，仅增加 11 ms～14 ms，相对提升 0.18%～1.06%，整体呈现略微上升趋势。

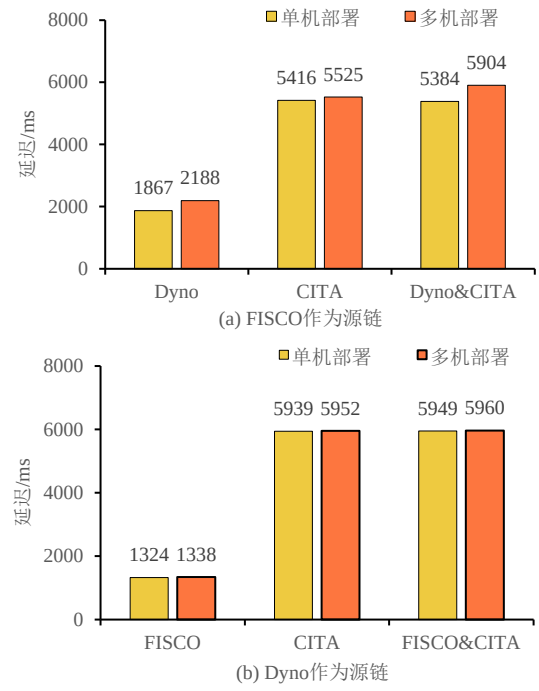


图 8　单机与多机部署跨链执行时间对比

7.5 单机与多机部署性能对比

从图 9 进一步分析可发现，中继链的处理时间在两种部署模式下有略微不同，平均维持在 500 ms 左右，但整体略微上升。FISCO作为源链时，中继链耗时在多机模式相较单机模式增加 11 ms ~ 59 ms；Dyno作为源链时，增加幅度在 45 ms ~ 78 ms之间。这表明中继链的内部处理逻辑具有一定的部署独立性和稳定性。在单机模式中，节点间通信通过本地回环网络完成，消息传输延迟极低；而在多机模式下，中继节点之间需通过局域网完成共识消息的广播与确认，尽管网络延迟较低，但由于涉及通信、系统调度等，其累计延迟不可忽略。部署于多台主机后，各节点之间的网络传输不可避免地引入了额外通信开销，特别是在共识阶段，由于涉及多个节点之间的消息广播与确认，延迟更敏感。

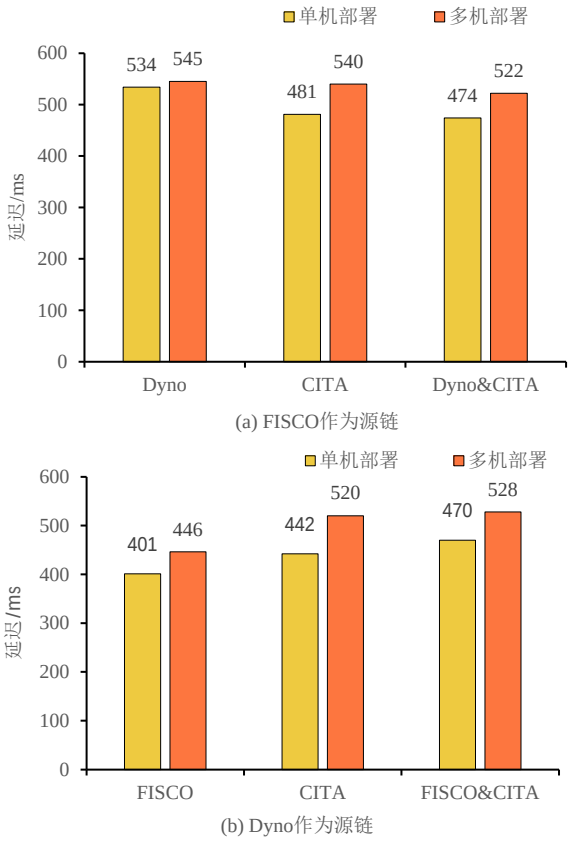


图 9 单机与多机部署中继链执行时间对比

图 10 展示了多机部署场景下不同跨链路径的延迟分解结果，整体与单机部署的趋势相同。从图中可见，中继链处理时间在不同路径中保持相对稳定，均在 500ms左右，说明系统中继层在多机环境下具备良好的稳定性和扩展性。相比之下，源链和目标链的确认延迟则呈现显著差异，成为整体性能的主要决定因素。在多数跨链路径中，目标链确认阶段占总延迟的 70% 以上，且不同目标链的确认时间差异较大，特别是涉及 CITA 的场景，其延迟显著高于其他链，成为整体性能的主要瓶颈。

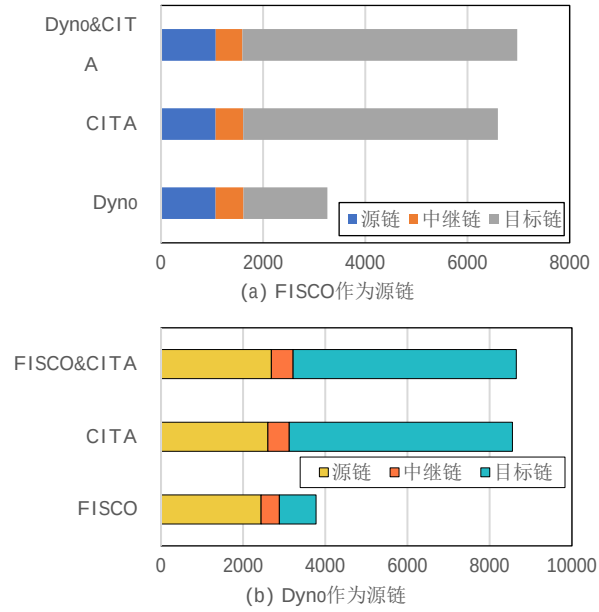


图 10 多机部署跨链延迟分解

8 结 论

本文提出了一种针对异构区块链、具有可扩展性的中继链跨链方案，通过针对区块链系统中共识协议的活性特性不同，提出了异构链管理机制，并通过多链的可扩展性，提出了基于向量时钟的多链协调机制。基于两个模块，搭建了一个跨链系统，通过在四个区块链系统上的大量实验，验证了本文方法的通用性及高效性。

参 考 文 献

[1] Duan Tiantian, Zhang Hanwen, Li Bo, et al. Survey on blockchain interoperability. Journal of Software, 2024, 35(2): 800-827(in Chinese)
(段田田, 张瀚文, 李博等. 区块链互操作技术综述. 软件学报, 2024, 35(2): 800-827)

[2] Deng Z, Tang C, Li T, et al. Enhancing blockchain cross chain interoperability: A comprehensive survey. 2025. arXiv: 2505.04934

[3] Wegner P. Interoperability. ACM Computing Surveys, 1996, 28(1): 285-287

[4] Kwon J, Buchman E. Cosmos whitepaper. 2019

[5] Wood G. Polkadot: Vision for a heterogeneous multi-chain framework. White Paper, 2016

[6] Breidenbach L, Cachin C, Chan B, et al. Chainlink 2.0: Next steps in the evolution of decentralized oracle networks. Chainlink Labs, 2021

[7] Li B, Duan T, Zhao Q, Guo Y, Song Z, Zhang H, Li Z, Sun Y. Performance modeling of relay chain. IEEE Transactions on Networking, 2025, 33(1): 194–209

[8] FISCO BCOS Open Source Team. Fisco bcos white paper v1.0: Financial blockchain infrastructure & practical examples. Shenzhen, China: WeBank and Financial Blockchain Shenzhen Consortium, Technical Report, 2017

[9] Androulaki E, Barger A, Bortnikov V, et al. Hyperledger fabric: A distributed operating system for permissioned blockchains// Proceedings of the thirteenth EuroSys conference. Porto, Portugal, 2018: 1–15

[10] Cryptape Team. CITA Technical Whitepaper. 2018(in Chinese) (Cryptape 团队. CITA 技术白皮书. 2018)

[11] Duan S, Zhang H. Foundations of dynamic bft//Proceedings of the IEEE Symposium on Security and Privacy. San Francisco, USA, 2022: 1317–1334

[12] Kanani J, Nailwal S, Arjun A. Matic whitepaper. Bengaluru, India: Polygon, Technical Report, 2021

[13] Nick J, Poelstra A, Sanders G. Liquid: A bitcoin sidechain. 2020

[14] Back A, Corallo M, Dashjr L, et al. Enabling blockchain innovations with pegged sidechains. 2014

[15] NEAR Protocol Team. Eth-near rainbow bridge. Technical deep dive documentation, 2020

[16] Multichain Team. Multichain: Cross-chain router protocol. Cross-chain infrastructure development documentation, 2023

[17] Wormhole Foundation. Wormhole protocol. 2020

[18] Dong M, Liang Q, Li X, Liu J. Celer network: Bring internet scale to every blockchain. 2018. arXiv: 1810.00037

[19] Luyu Cross Chain Protocol Team. Luyu cross chain protocol white paper. 2021(in Chinese)
(Luyu跨链协议团队. 陆羽跨链协议白皮书. 2021)

[20] IRIS Foundation. Iris network: Inter-chain service infrastructure and protocol for building trustworthy and distributed business applications. Shanghai, China: Bianjie and Tendermint, Technical Report, 2022

[21] Garay J, Kiayias A, Leonardos N. The bitcoin backbone protocol: Analysis and applications. Journal of the ACM, 2024, 71(4): 1–49

[22] Liu Yizhong, Liu Jianwei, Zhang Zongyang, et al. Overview on blockchain consensus mechanisms. Journal of Cryptologic Research, 2019, 6(4): 395–432(in Chinese)
(刘懿中, 刘建伟, 张宗洋等. 区块链共识机制研究综述. 密码学报, 2019, 6(4): 395–432)

[23] Kiayias A, Russell A, David B, Oliynykov R. Ouroboros: A provably secure proof-of-stake blockchain protocol//Annual international cryptology conference. Santa Barbara, USA, 2017: 357–388

[24] Buchman E, Kwon J, Milosevic Z. The latest gossip on bft consensus. 2018. arXiv: 1807.04938

[25] Gervais A, Karame GO, Wüst K, et al. On the security and performance of proof of work blockchains// Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 3–16

[26] Zakhary V, Agrawal D, El Abbadi A. Atomic commitment across blockchains. 2019. arXiv: 1905.02847

[27] Raynal M, Singhal M. Logical time: Capturing causality in distributed systems. Computer, 1996, 29(2): 49–56

[28] Basten T, Kunz T, Black JP, Coffin MH, Taylor DJ. Vector time and causality among abstract events in distributed computations. Distributed Computing, 1997, 11: 21–39

[29] Cachin C, Guerraoui R, Rodrigues L. Introduction to reliable and secure distributed programming. Berlin, Germany: Springer Science & Business Media, 2011

[30] Dwork C, Lynch N, Stockmeyer L. Consensus in the presence of partial synchrony. Journal of the ACM, 1988, 35(2): 288–323

[31] Fidge CJ. Timestamps in message-passing systems that preserve the partial ordering//Proceedings of the 11th Australian Computer Science Conference. Brisbane, Australia, 1988: 56–66

[32] Lamport L. Time, clocks, and the ordering of events in a distributed system. Communications of the ACM, 1978, 21(7): 558–565

[33] Wang X, Cao S, Jia K, et al. Lumi: Lightweight blockchain layer 2 protocol from on-chain coordination//Proceedings of the 2025 IEEE 45th International Conference on Distributed Computing Systems. Glasgow, UK, 2025: 538–548

[34] THORChain Team. Thorchain: A decentralised liquidity network. 2024

[35] ConsenSys. BTC relay. Technical documentation and implementation, 2016

[36] Matter Labs. Zksync era documentation: Bridging assets. 2023

[37] StarkWare. Starkgate architecture documentation. 2024

[38] Wanchain Foundation. Wanchain: Building super financial markets for the new digital economy. 2017

[39] FUSION Foundation. Distributed control rights management signature verification program. Technical implementation and documentation, 2018

[40] Raiden Network Team. Raiden network. Technical report, brainbot labs Est., 2023

[41] Liu Yizhong, Jiang Nan, Chen Ruonan, Liu Jianwei. Overview on blockchain off-chain channels. Journal of Cryptologic Research, 2024, 11(1): 45–66(in Chinese)
(刘懿中, 姜楠, 陈若楠, 刘建伟. 区块链链下通道研究综述. 密码学报, 2024, 11(1): 45–66)

附录A. 跨链信任模型核心区别对比表

附表 1 跨链信任模型核心区别对比表

维度	公证人机制	轻客户端验证	零知识证明	哈希时间锁	分布式私钥控制
信任来源分类	外部验证	原生验证	原生验证	本地验证	外部验证
信任假设	信任第三方公证人	信任源链共识	无需信任第三方	无需信任第三方	信任分布式网络
验证机制	无/多重签名或门限签名	区块头验证 + Merkle 证明	零知识证明(zk-SNARKs/zk-STARKs)	哈希锁+时间锁	门限签名或 MPC
性能特征	低延迟，无额外开销/中等延迟，存在协调开销	中等延迟，存在同步开销	快速验证，计算开销大	中等延迟，简单高效	高延迟，协调开销大
去中心化程度	低/中等	高	高	高	中等
技术复杂度	低	中等	高	低	高
可扩展性	高（适用于多条同构或异构链）	高（适用于多条同构或异构链）	中等（适用于预先支持 ZKP 验证的链或内置零知识证明验证模块的桥接方案）	低（仅适用于点对点交换）	高（适用于多条同构或异构链）
适用场景	快速跨链转账，适合安全性要求较低场景	广泛适用于跨链数据验证与资产交换	隐私保护及安全性要求较高的跨链应用	仅适用于点对点资产交换	对安全性有一定要求的跨链资产管理
典型项目	Multichain[16], ThorChain[34]	BTC Relay[35], Cosmos[4]	zkSync Bridge[36], StarkNet[37]	比特币与以太坊原子交换	Wanchain[38], Fusion DCRM[39]
主要优势	实现简单，响应迅速，成本低	完全去信任，继承源链安全性	强隐私保护，高效验证	完全去中心化，确保资产交换原子性	机制灵活，分散私钥控制，降低单点风险
主要劣势	中心化风险，依赖第三方公证人	需维护轻客户端状态，通信和同步开销较大	需要链或跨链桥支持零知识证明验证，计算与通信开销高	仅适用于点对点资产交换，功能单一	协议实现较为复杂，通信协调成本较高

附录B. 跨链信任模型介绍

跨链信任模型根据信任来源可以分为原生验证、本地验证、外部验证三类，我们在表 2 中总结了各种跨链信任模型的特点。

原生验证类信任模型。原生验证类信任模型通过直接验证源链的交易状态来建立信任，无需依赖外部第三方。这类模型的核心特征是能够独立验证跨链交易的有效性，直接继承源链的安全性保障。

轻客户端验证(Light Client Verification)通过在目标链上部署源链的轻客户端来验证跨链交易的有效性。轻客户端定期从全节点获取最新的区块头，并通过 Merkle 树根验证特定交易，无需下载完整区块即可确认交易，大幅降低了存储和计算开销。然而，轻客户端验证的可靠性高度依赖于源链的活性，即当源链是弱活性时，分叉风险使其在需要确定性交互的异构跨链场景下面临挑战。

本地验证类信任模型。本地验证类信任模型通过密码学协议在本地建立信任保证，不依赖外部验证或第三方机构，主要通过智能合约的原子性来确保交易的安全性。

哈希时间锁(Hash Time Locked Contract，HTLC) 是一种基于条件支付的智能合约机制，广泛应用于闪电网络、雷电网络[40]等区块链扩容技术中[41]。HTLC 巧妙地实现了“要么双方都成功交换，要么双方都安全退出”的原子性，并利用哈希函数的性质为资产交换提供安全性保障。典型应用包括比特币-以太坊原子交换——通过在两条链上分别部署 HTLC 合约实现 BTC 与 ETH 的直接交换；闪电网络支付路由——使用 HTLC 在支付通道网络中实现多跳支付的原子性保证。尽管 HTLC 设计精妙，但其功能较为单一，主要适用于点对点的资产交换，难以扩展至涉及多方或多阶段的跨链业务流程，也无法满足更细化的原子性需求。

外部验证类信任模型。外部验证类信任模型依赖于区块链网络之外的实体来提供信任保证，这些外部实体可以是中心化的第三方，也可以是去中心化的网络组织。

公证人机制(Notary)是该模型最直观的跨链解决方案，通过引入可信第三方作为中介，协调不同区块链之间的交互。根据公证人的组织形式，公证人机制又可分为中心化公证人方案和去中心化公证人方案。中心化公证人方案采用单一可信实体作为公证人，负责验证和转发跨链交易，具有响应速度快、实现简单的优势，但存在明显的单点故障风险。去中心化公证人方案采用多个公证人组成的联盟，通过多重签名[16,34]或门限签名[18]提供跨链服务，以提供更高的安全性。

附录C. 安全性证明

定理 1. 如果一笔有效的跨链交易 xtx 已经由其源链 SC 中的诚实节点正确发起(即生成了有效的可用性证书 ac 并向中继链提交了 X-Req，那么最终，所有参与该交易的诚实节点(在源链 SC 和所有目标链 TC 上)都会执行其对应的子交易。

证明. 我们下面论述任意交易 xtx 在协议流程中的执行过程，论证其最终必然完成。我们将流程分解为 4 个关键阶段，以说明活性保证。

(1) 提交请求：根据协议，当 xtx 在源链 SC 上达成共识后，所有诚实的 SC 节点都会向中继链 RC 发送 X-Req 消息。基于我们的部分同步网络假设，这些消息虽然可能延迟，但最终必然会被至少 2t+1 个诚实的中继链节点接收。

(2) 中继链协调：中继链 RC 运行一个安全的原子广播协议。根据该协议的活性属性，一旦有诚实节点广播了有效的 X-Req，该请求最终必将被所有诚实的中继链节点以确定的顺序交付和处理。处理后，所有诚实的 RC 节点会向目标链 TC 发送 X-Ready 消息，该消息同样会最终到达 TC 的诚实节点。

(3) 目标链执行：

① 若目标链为弱活性链：根据弱活性链的定义，任何被提交的有效交易最终都会被交付。因此，当 TC 的诚实节点收到有效的 X-Ready 消息后，其自身的共识机制保证了子交易 tx_t 最终会被执行。

② 若目标链为强活性链：根据强活性链的定义，TC 不保证单笔交易的交付。此时，中继链的超时与重试机制被激活。中继链上的诚实节点会启动计时器。若超时，则重新发送 X-Ready。由于网络是部分同步的，在一定时间后，重发的 X-Ready 消息最终被接收。同时弱活性链保证了其网络持续运行(持续处理新交易)。因此，经过有限次数的重试，X-Ready 最终会被 TC 成功执行。

(4) 交易确认与完成：目标链 TC 执行成功后会返回 X-Com，中继链完成本地处理后向源链 SC 发送 X-Done。类似地，这两个消息的传递和处理都由中继链与业务链，以及网络的活性所保证，最终保证源链上的子交易 tx_s 被成功执行，完成整个跨链流程。

在协议的各个阶段中，交易的执行都通过底层区块链协议的活性或本协议的超时重试机制得到保证，综上所述，正确发起的交易都会被执行。　　证毕.

定理 2. 如果一笔有效的跨链交易 xtx 已经由其源链 SC 中的诚实节点正确发起，那么最终，源链 SC 和所有目标链 TC 上至少各有一个诚实节点会执行其对应的子交易。

证明. 根据定理 1，所有参与该交易的诚实节点最终都会执行其对应的子交易。因此，源链 SC 和所有目标链 TC 上的所有诚实节点都会执行交易，显然满足至少各有一个诚实节点执行交易的要求。弱跨链活性得证。　　证毕.

定理 3. 对于任意两笔源自同一源链 SC 的跨链交易 xtx_1 和 xtx_2，如果 SC 上的一个诚实节点先执行 xtx_1 的源链子交易，再执行 xtx_2 的源链子交易，那么对于它们共有的任何目标链 TC，TC 上的诚实节点都不会在执行 xtx_1 的目标链子交易之前执行 xtx_2 的目标链子交易。

证明. 我们采用反证法。假设存在两笔来自同一源链 SC 的交易 xtx_1, xtx_2。SC 上的诚实节点先执行 xtx_1 再执行 xtx_2，但在某个共同的目标链 TC 上，存在诚实节点在执行 xtx_2 后再执行 xtx_1。

SC 先完成 xtx_1，即其向中继链提交 X-Req(xtx_1) 的时间点早于 X-Req(xtx_2)，根据中继链执行原子广播协议假设，所有诚实的中继链节点必然会以相同的顺序交付这两个请求，即先交付 X-Req (xtx_1)，后交付 X-Req (xtx_2)。在弱原子性模式下，当诚实的中继链节点处理 X-Req (xtx_1) 时，它会更新目标链的时钟 $G_i[TC] \leftarrow G_i[TC]+1$。假设此时. 从 k 变为 k+1。随后，中继链向 TC 发送携带有序列号 gseq = k+1 的 X-Ready 消息。在其后处理 X-Req(xtx_2)时，中继链会再次更新。其从 k+1 变为 k+2，并发送序列号为 gseq = k+2 的 X-Ready 消息。而目标链诚实节点只有在 gseq = lseq + 1 时，才会接受并执行序列号为 gseq 的 X-Ready 消息。因此，目标链节点必须先执行完序列号为 k+1 的 xtx_1(此时其 lseq 从 k 更新为 k+1)后才能满足序列号为 k+2 的 xtx_2 的执行条件，进而执行 xtx_2。因此，假设不成立，定理得证。　　证毕.

定理 4. 对于任意两笔共享至少一个业务链的跨链交易 xtx_1 和 xtx_2，如果某个参与链 A 上的一个诚实节点先执行其在 xtx_1 中的子交易再执行其在 xtx_2 中的子交易，那么在其他任何参与链 B 上，没有一个诚实节点会先执行其在 xtx_2 中的子交易，然后再执行其在 xtx_1 中的子交易。

证明. 我们采用反证法。假设存在两笔冲突的交易 xtx_1, xtx_2(共享业务链 A，B)，并且在链 A 上，xtx_1 先于 xtx_2 执行；但在链 B 上，xtx_2 先于 xtx_1 执行。

不失一般性，假设中继链的原子广播协议先交付了 X-Req(xtx_1)，后交付了 X-Req (xtx_2)。在强原子性模式下，当诚实的中继链节点处理 X-Req (xtx_1)时，它会检查 xtx_1 的所有参与链的 Lock 状态。在没有其他正在处理的冲突交易时，所有参与链 Lock 状态均为 false。协议会立即将这些锁设置为 true，然后开始协调 xtx_1 的执行。在其后处理 X-Req (xtx_2)时，协议检查 xtx_2 的参与链。由于 xtx_1 和 xtx_2 共享至少一条链 C，中继链检测到 Lock[C]的状态为 true。根据协议，当检测到参与链 Lock 状态为 true 时，X-Req (xtx_2)不能被执行，而是必须被放入等待队列 atomQ 中。只有在 xtx_1 的整个流程完全结束后(即所有 X-Com 均已收到，X-Done 已发送)，中继链才会还原 xtx_1 占有的所有 Lock 状态为 false。直到所有相关 Lock 状态都被还原为 false 后，等待队列 atomQ 中的 X-Req(xtx_2)才被提出并开始执行(其锁定和协调流程)。因此，假设不成立，定理得证。　　证毕.

NIE Peng, Ph.D. candidate. His research interests include data security and privacy protection.

WANG Mei, M.S. candidate. Her research interests include distributed systems and blockchain technology.

WANG Xin, Ph.D., assistant professor. Her research interests include distributed system security and blockchain technology.

ZHAO Wei, M.S. His research interest is blockchain technology.

DUAN Si-Si, Ph.D., professor. Her research interests include blockchain technology and applied cryptography.

JIA Ke-Ting, Ph.D., associate professor. Her research interests include analysis and design of cryptographic algorithms and application of cryptographic technology.

ZHANG Guo-Yan, Ph.D., professor. Her research interests include analysis and design of cryptographic algorithms, blockchain technology and innovative applications of cryptography.

Background

This paper studies the cross-chain of blockchains. Cross-chain atomicity is a crucial security property of blockchain interoperability. Cross-chain atomicity ensures that a cross-chain transaction is executed by all the blockchains (that are related to the transaction), or none of the blockchains execute the transaction. To realize cross-chain atomicity, known approaches include hash time lock contract (HTLC), atomic swap, relay chains, etc. Each of the approaches has its own pros and cons. Among them, the relay chain is the only generic solution that does not pose strong requirements on the users (who are involved in the cross-chain transactions). Meanwhile, regardless of the approaches taken by current cross-chain techniques, almost all cross-chain solutions treat each blockchain as a black-box and assume that each blockchain achieves the ideal security properties, i.e., safety and liveness of the blockchain consensus. However, practical consensus mechanisms might have nuance in their security definitions. There is a lack of treatment for cross-chain transactions for heterogeneous blockchains, especially with a focus on the underlying consensus mechanisms.

This paper presents CrossWeave, a new relay chain-based solution for blockchain interoperability. Compared to existing works, this work takes a formal treatment of the liveness property of the consensus mechanism of heterogeneous blockchain systems and provides a modular solution for cross-chain transactions.

The authors have more than a decade of research experience in the area of blockchain and distributed secure protocols. The team has published over 60 research papers in the area, including several noteworthy works that have been adopted by industrial systems. For instance, the BChain protocol (OPODIS 2014) has been adopted by the Hyperledger Iroha project and is now being used in the Central Bank Digital Currency (CBDC) of several countries, including Cambodia, Vietnam, etc. The Dashing protocol (Eurosys 2024) has been used in the cross-CBDC project called mBridge led by the Bank of International Settlement. Over 30 central banks are serving as observing members of the mBridge project.

This work is supported by the National Key R&D Program of China under project 2022YFB2702800. The goal of the project is to study trustworthy layer-1 and layer-2 interchange technologies of blockchain. This paper supports the task of providing a cross-chain solution for heterogeneous blockchain systems, which lays the foundation for the project. As part of the research project, this paper addresses the challenges of migrating or exchanging data across heterogeneous blockchains. Besides ensuring that the solution is secure, this approach is also lightweight and highly efficient. The solution aims to serve as the fundamental architecture of the project.