计算机研究与发展
Journal of Computer Research and Development
DOI: 10.7544/issn1000-1239.202230284
60(11): 2520−2533, 2023

PieBridge: 一种按需可扩展的跨链架构

段田田[1,2] 郭 仪[1,2] 李 博[1,2] 张瀚文[1,2] 宋兆雄[1] 李忠诚[1,2] 张 珺[3] 孙 毅[1,2]
[1](中国科学院计算技术研究所 北京 100190)
[2](中国科学院大学 北京 101408)
[3](内蒙古大学 内蒙古呼和浩特 010021)
(duantiantian@ict.ac.cn)

PieBridge: An On-Demand Scalable Cross-Chain Architecture

Duan Tiantian[1,2], Guo Yi[1,2], Li Bo[1,2], Zhang Hanwen[1,2], Song Zhaoxiong[1], Li Zhongcheng[1,2], Zhang Jun[3], and Sun Yi[1,2]
[1](*Institute of Computing Technology*, *Chinese Academy of Sciences*, *Beijing* 100190)
[2](*University of Chinese Academy of Sciences*, *Beijing* 101408)
[3](*Inner Mongolia University*, *Hohhot*, *Inner Mongolia* 010021)

Abstract Due to its decentralized and traceable nature, blockchain has been widely used in various fields such as digital currency, supply chain finance, and smart healthcare. As the demand for applications continues to expand, the need for independent blockchains to collaborate to build a broader value Internet is increasing, making it urgent to study cross-chain technology. However, as blockchain develops towards the application chain model, the number of blockchains has greatly increased, and the scale of cross-chain interaction has also increased. Existing cross-chain research cannot meet the challenges in terms of architectural scalability and cross-chain requirements diversity. Therefore, featured by the idea of "on-demand domain", we propose an on-demand and scalable cross-chain architecture called PieBridge and raise a four-layer cross-chain interaction protocol stack. By decoupling cross-chain transmission, verification, transactions and applications, our protocol stack meets the diverse requirements of cross-chain applications in terms of privacy, security, and performance. We implement the PieBridge prototype system and further verify PieBridge's scalability and flexible support for diverse cross-chain interactions from both theoretical and experimental perspectives.

Key words blockchain; cross-chain; atomicity; cross-chain transfer; queuing theory

摘 要 区块链由于其去中心、可溯源等特性,已被广泛应用于数字货币、供应链金融、智慧医疗等不同领域.随着应用需求的不断拓宽,各独立区块链协作以构建更广泛价值互联网的需求日益增强,因而迫切需要研究跨链技术.然而当前区块链生态规模不断扩大、丰富,异构/同构区块链间的互联互通需求也随之快速增长.而现有跨链研究无法应对架构可扩展性与跨链需求多样性方面的挑战.针对上述问题,基于"按需建域"的理念,提出一种按需可扩展的跨链架构 PieBridge,并提出了一套 4 层跨链交互协议栈,解耦跨链传输、验证、事务与应用,满足不同跨链应用在隐私、安全、性能等方面的差异化需求.同时实现了 PieBridge 原型系统,并通过建模分析与实验证明了 PieBridge 的可扩展性以及其对差异化跨链交互需求的灵活支持.

收稿日期: 2023-04-03; **修回日期**: 2023-06-08
基金项目: 国家重点研发计划 (2022YFB2702902); 国家自然科学基金项目 (61972382); 内蒙古自然科学基金项目 (2020MS06017)
This work was supported by the National Key Research and Development Program of China (2022YFB2702902), the National Natural Science Foundation of China (61972382) and the Natural Science Foundation of Inner Mongolia (2020MS06017).
通信作者: 张瀚文(hwzhang@ict.ac.cn)

关键词 区块链；跨链；原子性；跨链传输；排队论

中图法分类号 TP391

自比特币提出以来，区块链得到了学术界、工业界的极大关注，由于区块链的去中心、可溯源等特性，被广泛应用于数字货币、供应链金融、智慧医疗等不同领域，形成了多链、异构的区块链生态系统[1].

尽管区块链实现了区块链网络范围内的价值流通，但是各独立区块链之间呈现明显的孤岛效应. 随着应用需求的不断拓宽，各独立区块链相互联通、协作以构建更广泛价值互联网的需求日益增强. 如何融合同构、异构区块链，突破去中心化应用边界，构建更加开放、易于协作、多方共赢的区块链生态，成为当今的迫切需求. 因此，区块链的互操作性，即跨链研究，自 2016 年起得到了越来越多的关注[2].

跨链技术是跨越单一区块链的数据可信边界（共识机制作用范围）实现独立区块链间信息、价值流通的技术. 早期跨链研究通过在任意两条有跨链需求的区块链间建立连接实现直接的跨链交互[3-5]. 然而，区块链连接建立过程繁琐，需要完成区块链的适配，包括数据结构匹配、区块链验证规则加载以及必要数据同步①等工作，当彼此存在跨链交互需求的区块链较多时，两两连接的方式导致每条区块链的适配复杂度高（$O(n)$），该方案可扩展性较差.

因而，当前的跨链方案大多引入中继链（Relay Chain）连接所有交互区块链[6-8]. 存在跨链交互需求的区块链仅需与中继链进行适配、建立连接后，即可在中继链的桥接下实现与其它区块链的跨链交互，每条区块链的适配复杂度降至常数级（$O(1)$），但中继链的适配复杂度仍然维持在 $O(n)$.

随着区块链生态应用类型不断丰富，跨链互通的规模不断扩大，现有跨链方案面临着挑战，主要体现在 2 个方面：

1）跨链架构的可扩展性

无论是公链技术体系还是联盟链技术体系，异构区块链数量都在快速增长，异构/同构区块链间的互联互通需求也随之快速增长. 在公链技术体系下，由于性能、可定制性、价值捕捉等原因，开发者越来越多地转向构建独立的应用链而不是在公共区块链平台上部署智能合约[9]. Cosmos[6], Polkadot[7], zkSync[10] 等多种平台先后出现，用于帮助开发人员快速构建应用链，仅 Cosmos 主网（2021 年启动）就有 49 条活跃应用链，测试网有 247 条应用链[11]. 在联盟链技术体系下，更是涌现出了 Fabric、FiscoBicos、长安链、趣链等大量形态各异的区块链技术平台，基于这些异构区块链技术平台，面向金融、政务服务等不同领域的行业应用，构建了大量区块链服务平台[12]. 各区块链在既有用户和价值积累的基础上，产生了与其它区块链交互的外延需求，因此区块链间互联互通的需求也快速增长，给跨链解决方案的可扩展性带来新的挑战.

然而，现有跨链方案中，中继链承载着所有跨链消息的路由、转发和验证等工作，随着接入区块链和跨链交易的增多，中继链将会出现拥塞并成为跨链系统的性能瓶颈. 尽管 BitXHub[8] 等方案将中继链扩展成中继链网络以支持更多区块链的接入，但这使得大量跨链交易需要经过多跳中继链处理，导致跨链交互时延大幅度增加.

2）跨链需求的多样性

一方面，不是所有区块链间都存在相同强度的跨链需求，例如：医院区块链与康复养老区块链之间存在频繁的跨链交互，医院区块链却与供应链金融的区块链之间几乎不存在跨链交互. 而现有的“一通百通”的中继链跨链方案②难以满足不同区块链间的按需连通.

另一方面，现有跨链研究大多针对资产转移、代币互换等数字货币应用，然而区块链在各领域的广泛应用带来了更多类型的跨链应用，同时这些跨链应用对跨链交互在性能、安全、隐私等方面提出了差异化的要求. 例如，跨链代币互换为了避免双花问题更注重安全性；跨链医疗数据共享为了保护患者隐私更注重隐私性；跨链城市数据共享为了保证数据的实时性更注重性能. 现有跨链架构与单一、僵化的跨链交互机制难以满足不同区块链以及不同跨链应用在隐私、安全、性能等方面的差异化跨链需求.

针对上述 2 个挑战，本文提出了一种按需可扩展的跨链架构——PieBridge，并在此基础上提出了具有独立事务层的 4 层跨链交互协议栈，保证跨链架构可扩展的同时支持差异化的跨链交互.

① 例如，SPV 验证需要区块头链.

② “一通百通”是指区块链接入跨链系统后即可与该系统内所有区块链进行跨链交互.

具体而言，本文引入中继域，基于“按需建域”的理念，将有交互需求的区块链划分在一个中继域内，各中继域基于域内中继链实现域内区块链的跨链互联. 通过域的划分，一方面实现跨链系统负载的合理切分，保证跨链架构的可扩展性；另一方面满足域内区块链间的按需连通，以及对差异化跨链交互的支持. 在按需建域的基础上，针对单中继域内跨链交易数量增加造成的中继链拥塞问题，本文设计了中继域按需扩容机制，通过域内中继链复制与中继链负载分流，将跨链交易约束在单一中继链中进行处理，在实现中继域通量提升的同时，避免跨链交互时延的增加. 进一步，本文针对跨链交互需要解决的跨链信息传输、跨链信任传递与跨链事务处理 3 个基本问题，提出了具有独立事务层的 4 层跨链交互协议栈. 在跨链传输层、验证层、事务层实现多种基础协议，支持应用对各层协议的灵活选择，满足不同跨链应用在性能、安全、隐私等方面的差异化需求. 该协议栈首次解耦了跨链事务与应用，抽象出独立的跨链事务层，一方面简化跨链应用的设计与开发，支持跨链应用的灵活、快速构建；另一方面为跨链事务提供系统级保障，包括跨链事务的原子性与事务间的隔离性，避免跨链应用在事务保障方面的设计、实现缺陷为所在跨链系统引入安全漏洞.

本文的主要贡献包括 3 个方面：

1)提出了一种按需可扩展的跨链架构——PieBridge. 基于“按需建域”的理念，将需要跨链交互的区块链按需组建成中继域，通过域的管理与性能优化，保证跨链架构的可扩展性.

2)提出了一种具有独立事务层的 4 层跨链交互协议栈. 在对跨链交互进行功能分层的基础上，在各层实现多种基础协议，支持应用在交互过程中对各层协议的灵活选择，满足不同跨链应用在隐私、安全、性能等方面的差异化需求. 同时解耦跨链事务与应用，支持跨链应用简易、灵活、快速构建的同时，实现系统级跨链事务保障.

3)基于所提跨链架构实现了 PieBridge 原型系统，并通过理论分析与实验验证，证明了 PieBridge 的可扩展性，以及其对差异化跨链交互需求的灵活支持.

1 相关工作

跨链研究包括跨链架构和跨链交互机制 2 个层面的工作，其中跨链交互机制是在跨链架构的基础上实现跨链交互的具体方法.

1.1 跨链架构

现有跨链研究根据是否需要借助中继链可以分为直连跨链架构与中继跨链架构.

直连跨链架构的基本思想是有跨链需求的区块链直接实现跨链交互.

BTC Relay[3] 被认为是最早的跨链方案，其在以太坊上部署可以接收并处理比特币区块头与交易的智能合约，并借助 Relayer 在以太坊上实现基于最长链规则的比特币轻客户端，从而实现了比特币与以太坊的单向跨链操作.

RSK[4] 是锚定比特币的一个开源智能合约平台，其目标是将智能合约引入比特币.其通过多家有较高社会声誉的组织、公司组成的可信联邦与参与联合挖矿的比特币矿工作为代理，实现与比特币的双向跨链操作.

BTC Relay 和 RSK 这 2 种方案是针对特定 2 种区块链跨链的方案，针对性强、可扩展性差，难以直接移植至其它区块链跨链交互.

WeCross[13] 面向区块链互联互通问题，旨在构建一套未来区块链互联基础设施，虽然其支持多链跨链，但是本质上依旧是基于直连跨链架构的方案. WeCross 的核心组件是跨链路由，每条参与跨链交互的区块链都有一个由部署该区块链的机构搭建的跨链路由，区块链强信任其跨链路由. WeCross 的交互区块链通过跨链路由直接建立连接实现跨链交互. 与 BTC Relay 和 RSK 这 2 种方案相比，WeCross 面向更通用的区块链跨链交互，并将适配复杂度由链上卸载至跨链路由，但是跨链路由间依旧需要进行 ($O(n)$) 次适配，适配复杂度高.

基于中继链的跨链方案，其基本思想是有跨链需求的区块链借助其它区块链作为中继，通过一跳中继链处理实现跨链交互. 单个中继链的处理能力有限，当存在大规模跨链需求时，中继链可以进一步扩展成中继链网络，有跨链需求的区块链借助中继链网络，通过多跳中继链处理实现跨链交互.

Cosmos[6] 针对区块链存在的扩展性、可用性以及独立性问题，提出了构建区块链互联网的设想，为此设计了一种区块链网络架构，并提出了该架构下的跨链交互方案. Cosmos 的区块链网络架构分为分区和枢纽 2 部分，独立区块链被称作分区(Zone)，连接分区的特殊分区被称作枢纽(Hub)，分区借助枢纽实现跨链交互，在大量分区存在交互需求时，这种架构可以减少区块链之间的适配复杂度. 然而由于 Cosmos Hub 性能是有限的，其可连接 Zone 的数量也

是有限的，无法满足大规模跨链交互场景下跨链架构可扩展性的要求.

Polkadot[7]针对区块链在可扩展性、可伸缩性、安全性等方面普遍存在的问题，设计了一种多链架构. Polkadot由一或多条中继链以及多条平行链(parachain)构成. 其中，平行链负责具体业务的执行；中继链负责与其直接连接的所有平行链的最终性共识. 为了在保证安全性的前提下实现中继链上复杂的操作，Polkadot设计了4种角色共同维护网络：收集者、渔夫、提名者以及验证者. 与Cosmos类似，由于Polkadot需要在中继链上实现所有平行链的全局共识，这使得Polkadot的可扩展性极大地被中继链性能所约束，目前理论上仅100条平行链的接入.

1.2 跨链交互机制

跨链交互机制是在跨链架构的基础上实现交互功能的具体方法. 跨链交互机制研究需要解决跨链信息传输、跨链信任传递与跨链事务处理3个层面的基本问题.

1)跨链信息传输是指将一个区块链中的数据传送到另一个区块链. 与传统网络信息传输不同的是，区块链是一个封闭的去中心化系统，无法主动向其它外部用户或者区块链发送信息.

2)跨链信任传递是指目的区块链接收跨链信息后确认该信息来自于源区块链并已通过源区块链共识. 由于不同区块链采用不同的共识算法，其信任传递(即跨链信息验证)方式相应的也有所不同.

3)跨链事务是指为实现某个跨链交互逻辑而在2个(或多个)交互区块链内分别执行的一组链内交易(在不同链执行)，跨链事务处理需要保证原子性，即组成跨链事务的链内交易对所有链上状态的改变是原子的，所有涉及到的状态要么全部改变，要么全部不改变.

现有跨链研究面向跨链应用，提供一套完整的、但各层关键技术紧耦合的跨链交互机制. 哈希时间锁协议(Hashed Timelock Contract, HTLC)是典型的针对跨链资产交换应用的跨链交互机制[14-18]. 具体地，HTLC基于相同的哈希锁保证各个组成跨链事务的链内交易可以一致执行,基于引入差异化的时间锁,保证跨链操作最终一致回滚的同时,保证参与者有足够的时间提交交易. 在此过程中，HTLC并没有在区块链间传输跨链信息，而是通过用户在不同区块链上释放哈希锁密钥代替特定的跨链信息. 同时HTLC也没有直接实现跨链信任传递，而是根据预先设置的哈希锁验证哈希锁密钥的正确性，从而间接验证特定跨链信息有效. HTLC无法抵御审查攻击[19-21]、洪泛攻击[22]等，且仅适用于跨链资产交互应用，无法拓展至其他应用场景.

为了提供更通用的跨链交互功能，简化跨链应用的设计与实现，Cosmos, Polkadot, BitXHub等研究将跨链传输验证与应用进行了解耦.

Cosmos提出了区块链间通信协议(the Inter-Blockchain Communication Protocol, IBC协议)[23]. IBC协议是一种端到端、面向连接、有状态的协议，实现了独立分布式账本上模块之间的可靠、有序和认证通信. IBC协议具体由客户端、连接、通道与中继器组成. 其中，客户端负责验证跨链交易；连接与通道维护了跨链信息传输的状态，实现对端链与应用的认证以及传输有序性的支持；中继器实现跨链信息的转发.

Polkadot设计了平行链间跨链信息传输(the Cross-Consensus Message Passing, XCMP)[24]协议，XCMP协议通过允许双向通信的跨链消息传递通道在平行链间传递数据. 由于XCMP协议仍在设计实现阶段，目前平行链通信使用资源开销更大的、借助中继链通过多跳传输实现的水平中继路由信息传输(horizontal relay-routed message passing, HRMP). Polkadot通过中继链上的全局共识实现跨链信息验证.

BitXHub[8]提出的链间消息传输协议(Inter-Blockchain Transfer Protocol, IBTP协议)是一种类似TCP/IP的链间传输协议，其通过引入转发节点实现了记录跨链信息的IBTP数据包的传输，包括直接交互模式与中继链转发模式. BitXHub通过链上简单支付验证(simplified payment verification, SPV)实现对跨链信息的验证.

尽管Cosmos, Polkadot和BitXHub方案实现了最基本的传输验证与应用分离的2层结构，但是传输与验证紧耦合，事务与应用紧耦合. 在传输与验证方面，上述方案无法满足不同区块链差异化跨链数据验证的要求. 在事务与应用方面，各跨链应用的设计、实现需要保证事务的基础特性，设计、开发难度大且可能因为跨链应用在事务性基础特性保障方面的设计、实现缺陷为所在跨链系统引入安全漏洞.

2 按需可扩展跨链架构

本文提出了一种按需可扩展的跨链架构——PieBridge，并在此基础上提出了具有独立事务层的4

层跨链交互协议栈,在保证跨链架构的可扩展性的同时支持差异化的跨链交互,本节将介绍 PieBridge.

本节首先介绍 PieBridge 的整体设计思路,区分其与现有跨链架构;然后从中继域全生命周期管理与跨链交互的基本流程 2 个方面详细介绍了 PieBridge 的设计.

2.1 整体设计

本文引入中继域,并基于“按需建域”的跨链架构设计理念,设计了如图 1 所示的按需可扩展跨链架构 PieBridge. 与现有“一通百通”的基于中继链的跨链架构不同,“按需建域”考虑到区块链之间差异化的跨链需求强度,仅将有共同跨链交互需求的区块链连接在同一条中继链上,形成一个个独立的中继域. 各独立中继域在组建过程中可以根据跨链交互需求配置不同的规模、性能、安全等级的中继链. 通过“按需建域”,一方面将跨链交易隔离在中继域内,实现跨链系统负载的合理切分,保证跨链架构的可扩展性;另一方面,可以在中继域内按需配置中继链,从架构层面支持差异化的跨链交互需求.

在“按需建域”的基础上,本文针对单中继域内平行链数量增加、跨链交易比例升高等导致中继链拥塞的问题,设计了中继域动态扩容机制. 具体地,在中继域内复制一条新的中继链,并将跨链交易均衡分流至不同中继链进行处理. 通过中继域动态扩容,在保证实现中继域处理能力提升的同时,避免多跳中继链处理带来的跨链交互时延增加问题.

进一步地,考虑到中继域是随跨链交互需求动态生成、更新的,本文秉持去中心化的理念,引入了一条专门区块链对中继域进行去中心化的管理.

本文设计的 PieBridge 架构如图 1 所示,包括管理面与交互面 2 部分.

PieBridge 管理面由管理链组成,实现对交互面去中心、去信任、公开透明的管理. 管理链是为 PieBridge 提供去中心管理服务的区块链,其以智能合约的形式部署了具体的管理规则,通过管理决策下发机制可以将管理决策发送给交互面执行,具体实现方法将在 2.2 节中继域全生命周期管理中详细介绍. 此外,管理链还管理了一组用于构建中继链与网关节点的节点.

PieBridge 交互面实现具体的跨链应用,其由平行链、中继链、网关节点与按需组建的一系列中继域组成.

1)平行链. 平行链是存在跨链需求的独立区块链,其通过组建或者加入中继域实现跨链交互.

2)中继链. 中继链即按需组建的、在域内连接指定平行链的区块链,其通过与域内平行链适配、建立连接,为它们提供去中心的中继服务①.

3)网关节点. 网关节点即在域内平行链与中继链之间转发跨链数据,支撑跨链信息传输的节点. 网关节点既可以由平行链提供,又可以由跨链系统提

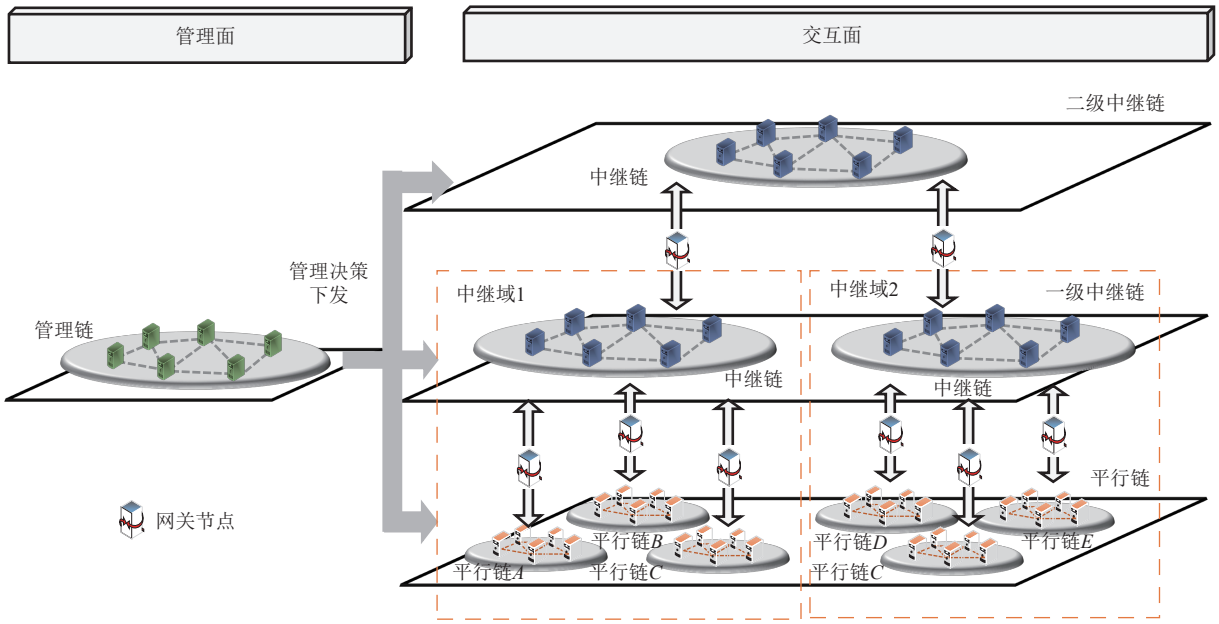

Fig. 1 The architecture of PieBridge

图 1 PieBridge 架构

① 为了应对域间不频繁、非时间敏感的跨链交互需求,可以引入连接中继链的二级中继链,以实现域间交互.

供，甚至可以通过设置激励机制引入，本文不讨论这部分内容.

4）中继域. 中继域是 PieBridge 中进行跨链交互的基本单元，区块链需要加入中继域才能进行跨链交互. 中继域由有共同跨链交互需求的平行链、连接这些平行链的中继链以及在区块链之间转发跨链信息的网关节点组成. 域内平行链大多为相同行业领域的区块链，而由于部分平行链可以应用于不同行业，与不同行业平行链存在跨链交互需求，例如供应链、保险等行业的跨链应用均需要引入银行区块链进行结算，因此平行链可以根据应用需求加入多个中继域.

2.2 中继域全生命周期管理

为了实现对中继域按需生成、按需更新的支持，本文设计、实现了对中继域的全生命周期管理，具体包括中继域的组建、中继域新增/删除平行链、多粒度的跨链资源访问控制与中继域动态扩容.

域的组建可以分为 3 个阶段，首先是请求发起阶段，各平行链代表向管理链提交建域请求，声明要组建一个域，并提出期望的共识算法、出块频率等中继链选型信息；随后进入决策阶段，管理链根据收到的建域请求生成中继链创世区块，实现了对中继链的按需配置与域内网关节点的初始化授权，并选择一组节点作为中继链初始共识节点；最后进入决策执行阶段，管理链基于网络将决策信息分发给中继链初始共识节点，这些节点验证决策已在管理链中达成一致后，依据该决策启动中继链，基于已授权的网关节点完成平行链与中继链的连接，实现域的构建.

域新增、删除平行链与域组建类似，但是在决策执行阶段管理链需要将决策信息分发给运行中的中继链，该操作无法直接基于网络传输实现. 为此，PieBridge 引入了中继链治理者，由治理者以中继链交易的形式实现决策分发. 而为了避免引入信任与单点故障问题，PieBridge 进一步设计了基于分布式私钥[25]的治理决策分发机制，由管理链共同担任中继链治理者，保证决策分发拥有与管理链相同的安全等级与去中心化程度. 具体地，管理链基于分布式私钥技术生成中继链治理节点公钥地址以及由管理链共识节点共同持有的分布式私钥，当管理链处理平行链组加域、退域请求时，管理链共识节点根据管理链决策分别使用其持有的分布式私钥签署中继链交易，当分布式私钥签名达到一定数量时该交易生效.

在多粒度的跨链资源访问控制方面，PieBridge 通过中继链对平行链网关节点权限的控制，保证只有授权网关节点所在平行链的跨链信息能被转发至中继链，从而实现区块链级别的资源访问控制；在平行链上部署基于区块链标识、跨链应用标识与用户全局标识的资源访问控制列表，并通过跨链请求与资源访问控制列表的匹配，实现应用与用户级别的资源访问控制.

由于中继链需要处理域内所有跨链交易，是中继域的性能瓶颈，因此随着接入平行链增多、跨链交易增多，中继链将率先出现拥塞，并导致域内跨链交互高时延问题. 为此，PieBridge 提出了如图 2 所示的动态扩容：管理链在域内复制一条与域内平行链连接的中继链，将域内原本由单一中继链处理的跨链交易分流至 2 条中继链上. 该方案通过适度增加平行链上的存储、计算开销，可以有效缓解单中继链拥塞问题，并保证域内跨链交易无需多跳中继链处理，从而降低跨链系统平均跨链交易时延.

域的扩容同样可以分为 3 个阶段：首先是请求发起阶段，平行链代表向管理链提交扩容请求，声明中继域达到性能瓶颈；随后进入决策阶段，管理链依据该域中继链生成新中继链创世区块、选择一组节点作为新中继链初始共识节点并生成跨链数据分流策略；最后进入决策执行阶段，管理链基于网络将创世区块分发给新中继链初始共识节点，由这些节点启

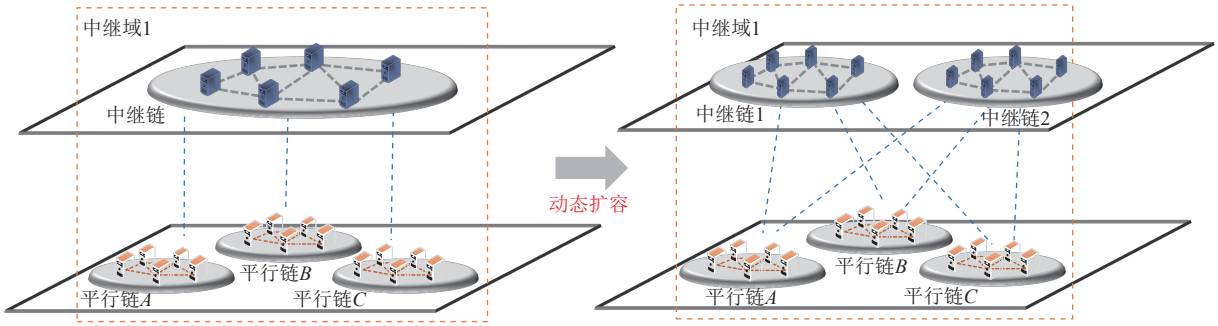

Fig. 2 The dynamic expansion of the relay domain

图 2 中继域动态扩容

动新中继链，基于网络将分流策略分发给已授权的网关节点，网关节点加载分流策略，同时平行链适配新增中继链，实现域的扩容.

2.3 跨链交互基本流程

PieBridge 中继域内，平行链借助中继链实现跨链交互，因此平行链与中继链均需要部署跨链交互功能. 跨链应用的具体功能需要基于跨链交互实现，例如，在资产类应用场景下，跨链转账应用通过资产区块链 *A* 对资产区块链 *B* 上用户资产状态的跨链更新实现；在医疗保险应用场景下，跨链核保应用通过保险区块链对医院区块链患者医疗数据的跨链读取实现. 而由于跨链交互无法协调交互区块链完成不存在的应用功能，因此交互区块链均需要部署相应的应用，在上述跨链转账应用中，资产区块链 *B* 需要本身支持对用户资产状态的管理.

本文抽象了 PieBridge 中跨链交互的流程，具体如图 3 所示，包括跨链请求发起阶段、跨链请求中继阶段、跨链请求处理阶段以及跨链回执反馈阶段. 为了便于描述，本文将发起跨链交互的平行链称为“源平行链”，接收并处理跨链请求的平行链称为“目的平行链”，在上述例子跨链核保应用中，保险区块链是源平行链，医院区块链是目的平行链.

1)跨链请求发起阶段. 用户在源平行链上以交易的形式调用跨链应用合约并触发跨链请求. 源平行链跨链协议栈合约将该跨链请求打包成标准的跨链数据包.

2)跨链请求中继阶段. 源网关节点通过监听源平行链获取跨链数据包，整理出证明数据包在源平行链中达成一致的证据，随后将该跨链数据包与证据以交易的形式发送到中继链中. 中继链跨链协议栈合约基于证据验证跨链数据包在源平行链达成一致，并将验证结果写入链中实现背书.

3)跨链请求处理阶段. 目的网关节点通过监听中继链获取跨链数据包与中继链背书的证据，并将该数据包以交易的形发送到目的平行链中. 目的平行链跨链协议栈合约解析跨链数据包，并通过验证中继链对跨链请求的背书间接验证该跨链请求在源平行链上达成一致，随后调用对应的应用合约，完成跨链请求响应.

4)跨链回执反馈阶段. 跨链请求处理回执以同样逐跳传输、验证的方式传回源平行链.

复杂的跨链应用需要多次跨链交互实现，PieBridge 跨链交互协议的具体设计与实现将在第 3 节中详细展开.

3 四层跨链交互协议栈

本节介绍具有独立事务层的 4 层跨链交互协议

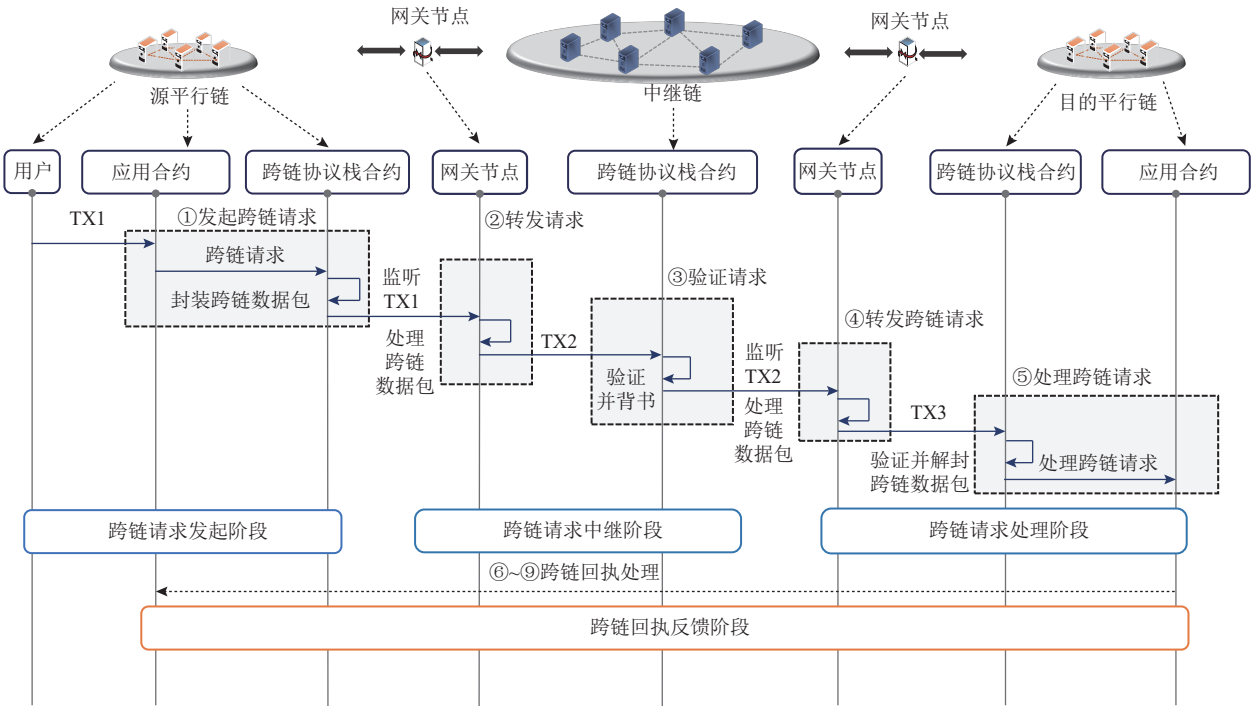

Fig. 3 The workflow of cross-chain interaction

图 3 跨链交互基本流程图

栈，首先概述了分层跨链交互协议栈的整体设计思路，随后详细介绍协议栈各层的内容.

3.1 协议栈整体设计

针对跨链交互需要解决的跨链信息传输、跨链信任传递与跨链事务处理3个层面的基本问题，本文通过抽象跨链元操作与跨链事务，首次将跨链事务与跨链应用解耦，提出了具有独立事务层的4层跨链交互协议栈，如图4所示. 独立事务层的抽象，一方面简化了跨链应用的设计与开发，支持跨链应用的灵活、快速构建；另一方面为跨链应用的事务性安全提供了系统级保障，避免跨链应用在事务处理方面的设计、实现缺陷为所在跨链系统引入安全漏洞，有利于跨链应用生态系统的发展. 在协议分层解耦的基础上，本文在协议栈各层均实现了多种基础协议，并采用适配器设计模式[26]，为协议栈各层引入聚合器，设计通用层间接口，将层间多协议的复杂交互转换为聚合器间的简单交互，降低了多种协议的管理复杂度，保证了层内协议的灵活性，满足不同跨链应用在性能、安全、隐私等方面的差异化需求.

4层跨链交互协议栈具体包括跨链应用层、跨链事务层、跨链验证层与跨链传输层.

1)跨链应用层. 跨链应用层部署在平行链上，包括多种跨链应用，各跨链应用设计、实现需要跨链交互的应用功能. 跨链应用需求将会被封装成跨链事务，交由跨链事务层处理.

2)跨链事务层. 跨链事务如图5所示，由在2个(或多个)交互区块链内分别执行的一组链内交易(在不同链执行)组成. 跨链事务层部署在平行链，提供了独立的事务协议，在保证跨链事务原子性的同时，保证跨链事务与跨链事务间、跨链事务与链内交易间的隔离性，即组成跨链事务的交易与组成其他跨链事务的交易及链内交易互不影响.

3)跨链验证层. 跨链验证层解决跨链信任传递的问题，与跨链传输层一起支撑跨链事务处理. 其部署在平行链、中继链以及网关节点上，提供了多种区块链验证方法与对应的证据生成机制，实现了平行链与中继链对基于跨链传输层所获取的跨链数据的可信性验证.

4)跨链传输层. 跨链传输层解决跨链信息传输的问题，与跨链验证层一起支撑跨链事务处理. 其部署在平行链、中继链以及网关节点上，提供了多种跨链传输协议，实现了跨链数据在交互区块链间的传输. 其中，由于区块链无法主动向外部发送信息[18]，各跨链传输协议均通过网关节点监听平行链与中继链获取跨链信息、转发跨链信息实现.

本文采用适配器设计模式，如图4所示，为协议栈各层引入聚合器，一方面实现层内协议管理，另一方面实现层间交互. 在层内协议管理方面，本文通过协议注册与协议地址映射表实现新协议的加载与协议的调用. 在层间交互方面，本文设计了如表1所示的各层控制信息. 在跨链请求发送阶段，上层聚合器将处理好的跨链数据以及控制信息发给下层聚合器，下层聚合器根据控制信息进行协议调度，数据经过层内协议处理后，聚合器再将本层数据打包，连同控

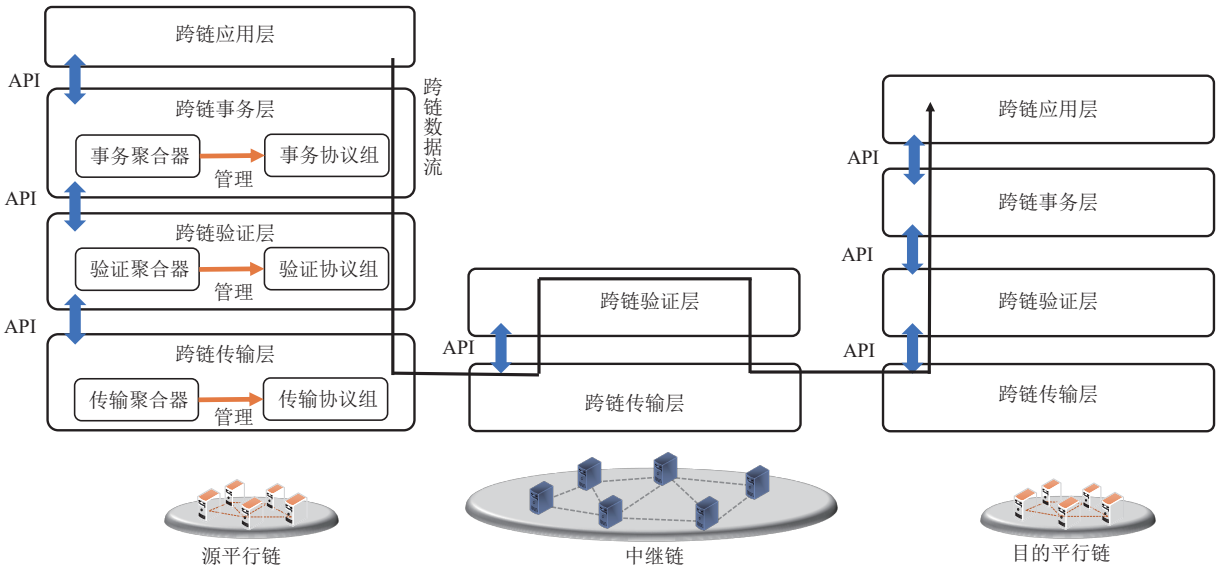

Fig. 4 The four-layer cross-chain interaction protocol stack

图4 4层跨链交互协议栈

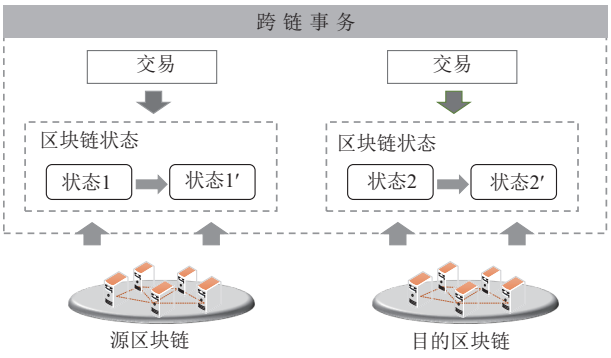

Fig. 5 The illustration of the cross-chain transaction

图 5 跨链事务示意图

Table 1 Control Information of the Cross-Chain Interaction Protocol Stack

表 1 跨链交互协议栈控制信息

协议层	控制信息	类型	说明
事务层	xid	String	事务 ID
	sourceApp	String	源应用
	destApp	String	目的应用
	tsctType	String	事务协议类型
	tsctstatus	Int	事务状态
	tsctSpec	Bytes	事务协议数据单元
验证层	verifyType	String	验证类型
传输层	tsptType	String	传输类型
	tsptStatus	String	传输状态

制信息一并发给下层聚合器；在跨链请求处理阶段，各层解析本层控制信息，并将本层控制信息外的其它数据发给上层聚合器. 各层的数据包拆封与解封方式与传统网络协议栈类似，每层的数据都是由本层的数据包头以及来自上层的数据包体组成，发送时层层封装，接收时层层解封.

3.2 分层协议栈设计

1)跨链事务层

跨链事务层解耦应用与事务，为跨链事务提供多种事务协议保障其原子性与隔离性.

本文针对事务协议的共性需求，抽象出资源锁服务、回滚日志服务、事务时钟服务 3 种基础服务.

资源锁服务是保证跨链事务隔离性的重要模块. 跨链事务隔离性本质上需要解决并行的多个跨链事务对同一公共资源的竞争问题. 资源锁服务通过在链上维护资源锁映射表，记录当前世界状态下指定对象的操作权限. 跨链事务在访问区块链资源前需要成功申请资源锁，在跨链事务完成后需要释放资源锁. 资源锁具体包括读锁、共享写锁和独占写锁.

回滚日志服务是保证跨链事务原子性的重要模块. 回滚日志服务通过预写日志和回滚日志的方法实现事务处理失败时所涉及的区块链状态的回滚. 具体而言，跨链事务在执行前需要将回滚操作加载至回滚日志服务中，若事务处理失败，则回滚日志服务会按照事务预先注册的回滚策略进行事务回滚.

事务时钟服务主要解决区块链缺少统一时钟带来的事务协议可终止性问题. 事务时钟服务通过链上记录事务有效时间与链下计时器监听、触发事务超时事件完成对跨链事务超时情况的监测.

基于上述 3 种基础服务，本文设计了 3 种基本的事务协议，包括基础事务协议(BTP)、自动事务协议(ATP)、多链事务协议(MTP). BTP 仅适用于有补偿操作的跨链应用，且需应用显示地给出事务处理失败时的补偿操作，其基于补偿保证原子性，基于资源锁服务保证隔离性. ATP 基于回滚日志服务保证原子性，基于资源锁服务保证隔离性，适用范围更广. MTP 适用于多链事务，其原子性和隔离性保障方式与 BTP 相同.

2)跨链验证层

跨链验证层提供多种跨链验证方法保证跨链交互过程中跨链信息的可信性，即跨链信息是经过源链共识的且存在于源链主链中的数据.

本文实现了基于 SPV 的跨链数据验证方法与基于公证人的跨链数据验证方法. 基于 SPV 的验证方法通过区块头链的同步保证区块头的可信性；通过状态树验证，保证指定状态数据在有效区块头中. 基于公证人的验证方法通过可信公证人背书实现跨链数据的验证.

跨链验证层部署在平行链、中继链以及网关节点上，其中平行链与中继链实现具体的验证方法；网关节点根据不同验证协议生成不同的证据.

3)跨链传输层

跨链传输层实现链到链的信息传输. 由于区块链需要借助链下网关节点监听转发实现跨链信息传输，在此过程中需要解决如何在源链上发现并获取待转发的跨链信息这一基本问题.

为此，本文设计了跨链事件抛出与跨链数据队列存储 2 种基础服务. 其中，跨链事件抛出服务是指利用大多数区块链自有的事件机制，将封装好的跨链数据包以事件形式抛出，以便网关节点监听. 跨链数据队列存储服务是指利用智能合约维护一个存储跨链数据的队列，新产生的跨链信息将存储在队列中等待传输，链下网关需要不断查询这个队列来判

断是否有新的数据包到来.

在此基础上，本文进一步设计了基础跨链传输协议与可靠跨链传输协议. 其中，基础跨链传输协议仅实现跨链信息监听、转发. 而可靠跨链传输协议为跨链数据包引入唯一的序号，实现重复跨链信息的过滤；基于源平行链、目的平行链上维护的下一笔跨链数据包发送序号、跨链数据包接受序号，实现跨链信息有序交付以及数据包丢失重传.

4 验证与分析

本节从理论分析与实验2个角度对PieBridge的可扩展性与差异化跨链交互进行了验证. 首先通过理论分析，对比中继链网络方案与PieBridge在扩展性方面的优劣. 接着，通过2组实验，对比了单中继链方案与PieBridge在可扩展性方面的优劣，并验证了PieBridge对差异化跨链交互的支持.

4.1 理论分析

本文选取跨链交易的时延作为指标，由于所有跨链交易都需要经过源平行链、目的平行链处理，因此此处时延仅指跨链交易在中继链中的时延. 若跨链系统在时延可接受、波动较小的情况下可以接入更多区块链，则表明该跨链系统扩展性良好；反之，若跨链系统为接入更多区块链，造成了严重的阻塞，甚至是无限制排队，导致时延很高，则跨链系统可扩展性不佳.

4.1.1 建 模

首先，本文对于一条中继链上跨链交易的到达和处理作出一些基本假设. 假设交易的到达、处理是服从参数为λ和μ的泊松过程，这种假设将每条中继链的交易处理过程简化为一个$M/M/1$模型[27]. 此处的λ和μ的单位并非区块链中常考量的tps，而是以中继链的区块为单位计数的频率，例如若中继链从一条区块链收到跨链交易的频率是5 tps，中继链处理交易的速度是100 tps，且区块能打包500笔交易，那么在本模型中对于中继链而言，此时每秒钟会从一条平行链收到$\lambda=\frac{5}{500}$个中继链区块，并处理$\mu=\frac{100}{500}$个中继链区块，即

$$\lambda=\frac{TPS_1\cdot\gamma}{B},\ \mu=\frac{TPS_2}{B},\tag{1}$$

其中，TPS_1是一条源平行链的交易处理速率，γ是其中的跨链交易比例，B是一个中继链区块所能容纳的交易数目，TPS_2是中继链处理交易速率.

接下来，讨论每种方案的理论时延.

1）中继链网络

记W_{relay}为中继链网络下一笔跨链交易从被发送至中继链到上链处理完毕的时延，表示为

$$W_{\text{relay}}=p_1\cdot W+p_2\cdot 2W.$$

因为中继链随机连接一组平行链，对于一笔跨链交易，其源平行链与目的平行链既可能连接同一条中继链，又可能连接不同的中继链，这2种情况对应的概率分别记做p_1和p_2. W为经历一次中继链的时延，则当源中继链、目的中继链在同一中继链群组时，跨链交易会经历一次中继链时延，源中继链、目的中继链分处于不同中继链群组时，跨链交易经历2次中继链时延. p_1和p_2的计算公式为：

$$p_1=\frac{1}{k},\ p_2=\frac{k-1}{k}.$$

k为中继链网络下中继链群组的数目，它是下面方程的解.

$$k-1+\frac{n}{k}=\theta\cdot m_{\max},$$

其中$k-1$是指当前中继链需要与所有其他中继链互联，n为整个网络中需要提供服务的平行链数量，则$\frac{n}{k}$为每个群组中随机分配到的平行链数量，$k-1+\frac{n}{k}$不得超过中继链的极限接入能力$m_{\max}$. θ是本文引入的参数，代表中继链接入链数目占其极限数目的百分比，$\theta<1$，记$m=\theta\cdot m_{\max}$. 根据前文抽象的排队论$M/M/1$模型，每条中继链的负载强度ρ不得大于1，否则中继链会发生阻塞，跨链交易无限制排队下去，由此可得$m_{\max}$：

$$\rho=\frac{m_{\max}\lambda}{\mu'}\leqslant 1,\ m_{\max}=\frac{\mu'}{\lambda}.$$

需要注意的是，式(1)中的μ表示中继链处理交易的能力，但是在中继链网络下，除跨链交易外，每个源平行链区块头也需要被中继链同步，这会降低中继链的处理能力，我们假设平均q笔跨链交易会有1个区块头需要同步，中继链对跨链交易的实际处理能力μ'为

$$\mu'=\frac{q}{q+1}\mu.$$

由$M/M/1$模型可知，单一中继链的时延W为

$$W=\frac{1}{\mu'-m\lambda}.$$

因此有

$$W_{\text{relay}}=\frac{2k-1}{k(\mu'-m\lambda)}.\tag{2}$$

2）PieBridge

本文分别为PieBridge不进行中继域扩容的基础

方案与进行中继域扩容的优化方案建模.

①基础方案

在“按需建域”的 PieBridge 中, 一笔跨链交易的源平行链、目的平行链总是存在于同一中继域中, 因此有,

$$W_{\text{PieBridge1}} = W = \frac{1}{\mu' - m\lambda}, \tag{3}$$

其中, $m = \theta \cdot m_{\max}$. 需要注意的是, PieBridge中继域依据跨链需求生成, 各中继域内平行链数量不一致, 本文建模对此进行了简化. 同时, 由于各中继域独立, 中继链不需要互联, 因此每条中继链所连接的所有区块链都是需要服务的平行链, 这将使得在全网区块链数目一定的情况下, 本文方案需要更少的中继链群组建立更少的中继链, 才达到更低的时延.

②优化方案

在支持中继域扩容的 PieBridge 中, 网关节点根据跨链交易分流策略, 实现中继链之间的负载均衡. 假设每个中继域有s条中继链, 此时整个中继链的模型不再是单链模型$M/M/1$的组合, 而是相当于一个多服务台的$M/M/s$系统[27].

根据图 6, 中继链时延$W_{\text{PieBridge2}}$可以表示为

$$W_{\text{PieBridge2}} = \frac{p_0\rho^s\rho_s + \rho s!(1-\rho_s)^2}{s!(1-\rho_s)^2 m\lambda}. \tag{4}$$

式(4)中p_0为中继链空载概率, 其计算公式为

$$p_0 = \left(\sum_{n=0}^{s-1}\frac{\rho^n}{n!} + \frac{\rho^s}{s!(1-\rho_s)}\right)^{-1},$$

其中ρ代表整体中继链负载强度, 其计算公式为

$$\rho = m\lambda/\mu''.$$

式(4)中ρ_s代表此时每条中继链负载强度, 其计算公式为

$$\rho_s = \frac{m\lambda}{s\mu''}.$$

此处μ''为该方案下每条中继链对跨链交易的实际处理能力, 需要注意的是, 此时即使一笔跨链交易没有经由某条中继链处理, 但是其区块头仍然需要s条中继链全部同步, 即

$$\mu'' = \frac{q}{q+s}\mu.$$

$m_{\max}$由$\rho_s = 1$取得, m仍然满足$m = \theta \cdot m_{\max}$. 值得注意的是, 此时$m_{\max}$扩展了$\frac{s(q+1)}{q+s}$倍. 尽管支持中继域扩容的 PieBridge 方案中, 由于域内中继链均需要适配所有平行链, 每条中继链上的平行链数据同步开销将随着平行链的接入而不断增加, 因此$m_{\max}$无法扩展为s倍. 但是, 仍然可以极大地提升跨链架构的可扩展性, 且提升效果在跨链交易密集的情况下更好.

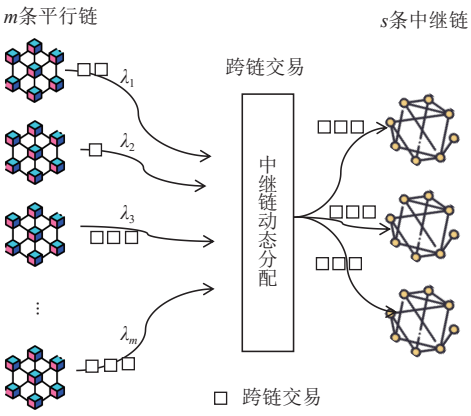

Fig. 6 The multi-server queuing model

图 6 多服务台排队模型

4.1.2 分 析

本文根据对 Cosmos 网络中跨链交易比例的测量以及现有相关测量工作[28], 进行了如表 2 所示的配置, 并讨论了表 2 配置下 PieBridge 与 Cosmos 的端到端跨链时延随接入平行链数量变化的模拟实验结果.

Table 2 Parameters When the Number of Connected Parachain Varies

表 2 接入平行链数量变化时的参数

n	TPS_1	γ	TPS_2	θ	q	s	B
变量	438	2%	438	95%	5	2	1 000

对比如图 7 所示的 3 种方案在不同接入区块链数目下的延时变化情况.

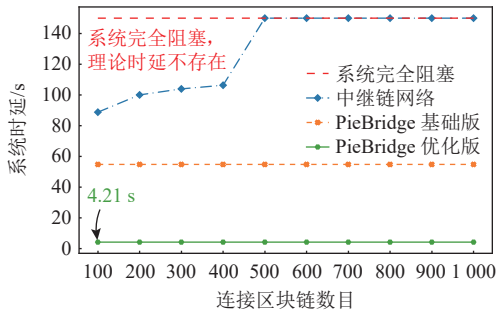

Fig. 7 Delay of transactions in the relay chain system when the number of connected parachain varies

图 7 随接入平行链数量变化的中继链交易时延

1)中继链网络的时延始终高于 PieBridge 的 2 种方法, 这是因为中继链网络下, 跨链交易有很大的概率需要跨越中继链群组, 随着区块链和中继链群组数目越多, 跨链交易需要涉及其他群组的概率也就越大, 因此其时延不断增长. 进一步地, 因为中继链

既需要与一定数目接入区块链相连，又需要负责连接数目越来越多的其他中继链，当接入区块链数目超过某一临界值后，所有中继链彻底拥塞，跨链交易开始无限制排队，此时如图 7 中虚线所示，理论时延不是虚线对应的数值 120 s，而是随着无限排队趋于无穷，因此中继链网络的可扩展性瓶颈可见一斑.

2）PieBridge 的 2 种方法因为不涉及跨链交易跨中继链群组传播，因此时延显著低于中继链网络方案，且因为中继链不需要连接其他中继链，系统不会因为中继链群组、接入区块链数目的增多而发生拥塞，展现了良好的可扩展性.

3）PieBridge 的优化方案展现出了极佳的性能，在每个中继链群组只额外引入 1 条区块链的情况下，系统的时延由 55 s 大幅度下降到了 4.21 s，该时延已经接近中继链本身的共识时延，基本可达到“跨链交易产生，即被中继链处理”的效果. 此外，如模型中讨论，PieBridge 优化方案的区块链可接入数量得到极大的提升，系统的可扩展性优异.

4.2 实验分析

本文选取中继链吞吐量（中继链单位时间内处理交易的数目）作为指标，进行了性能实验. 通过增加接入中继链的平行链数量，比较中继链吞吐量，衡量跨链系统性能. 若中继链吞吐量趋于平稳，则表明中继链已到达性能瓶颈，无法接入更多的平行链.

然后本文验证了 PieBridge 对差异化跨链交互的支持. 根据供应链数据存证场景与跨城市公共服务场景对跨链交互在可扩展性与时效性上的不同需求，针对性地构建了不同类型的中继域，并通过实验比较 2 个中继域在性能方面的差异性，验证了 PieBridge 对差异化跨链交互的支持.

考虑到实际跨链应用中的网络问题，本文选择部署在不同地区的云服务器进行实验. 服务器使用了 2 个 CPU,每个 CPU 配置为 2.10 GHz、20 核，服务器内存为 256 GB.

4.2.1 性能试验

本文将 Cosmos 跨链系统与按需建域的 PieBridge 进行比较，通过比较相同实验环境下 Cosmos 与 PieBridge 吞吐量随接入平行链数量的变化，验证 PieBridge 在性能方面的可扩展性.

在实验中，本文设置 Cosmos 跨链系统和 PieBridge 跨链系统的中继链配置相同，均采用 4 个共识节点，分别部署在 4 台服务器上，中继链每个区块可容纳的最大交易数量为 5 000；同时使用了进程模拟平行链出块过程，假设平行链出块间隔时间为 6 s，每个区块中的跨链交易比例为 10%，每当平行链有新的区块产生，区块内的跨链交易都会被逐个发送到中继链.

实验结果如图 8 所示. Cosmos 吞吐量首先随接入平行链数量的增多持续增大，并在接入 500 条平行链后趋于稳定，这意味着跨链系统到达性能瓶颈，无法接入更多的平行链、处理更多的跨链交易；而 PieBridge 由于将平行链接入了不同的中继域，并且当中继域到达性能瓶颈时会动态扩容，其吞吐量随接入平行链数量的增多持续增大. 由图 8 表明，PieBridge 可以通过域的扩容，按需提升跨链系统的吞吐量，保证跨链性能不随跨链交互的增加而降低.

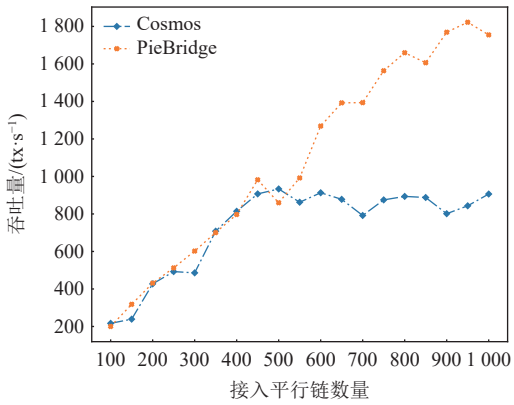

Fig. 8 Throughput of cross-chain system when the number of connected parachain varies

图 8 随接入平行链数量变化的跨链系统吞吐量

4.2.2 差异化交互实验

本文依据供应链数据存证场景与跨城市公共服务场景对跨链交互在可扩展性与时效性上的不同需求，构建 2 个中继域，并通过比较这 2 个中继域的吞吐量与跨链交互时延，验证 PieBridge 对差异化跨链交互的支持.

在该实验中，供应链数据存证场景更加注重中继域（Zone-1）的吞吐量，且需要保证存证数据完整可靠，中继链配置了较大的区块容量与权威证明（proof of authority, PoA）共识机制，跨链交互协议栈在验证层配置了基于 SPV 的跨链数据验证方法，传输层配置了可靠传输协议；面向跨城市公共服务场景的中继域（Zone-2）注重跨链交互的时效性，中继链配置了较小的出块间隔与 tendermint 共识机制，跨链交互协议栈在验证层配置了基于公证人的跨链数据验证方法，传输层配置了基础跨链传输协议.

实验结果如图 9 所示，Zone-1 拥有更高的平均吞吐量，Zone-2 拥有更低的跨链交互时延. 这是因为 Zone-1 配置了较大的区块容量，单区块包含的交易

数量增多;Zone-2 配置了较小的出块间隔,且采用基于公证人的跨链数据验证方法,缩短了交易等待时间与跨链数据验证时间. 该结果符合不同应用场景对跨链交互提出的需求,这意味着 PieBridge 可以通过中继链与跨链交互协议的配置实现差异化的跨链交互.

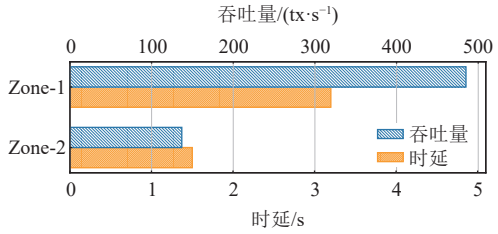

Fig. 9 Diversity of cross-chain interactions

图 9 差异性跨链交互

5 总 结

本文基于“按需建域”的理念提出了一种按需可扩展的跨链架构 PieBridge,并在此基础上提出了具有独立事务层的 4 层跨链交互协议栈,在保证跨链架构可扩展的同时支持差异化的跨链交互. 本文实现了 PieBridge 原型系统,并通过建模分析与实验验证了 PieBridge 的可扩展性,以及其对差异化跨链交互需求的灵活支持.

未来,我们将在本文工作的基础上,针对当前跨链交互性能低的问题,从 2 个层面开展工作:

1)架构层面. 针对中继链上交易类型单一、交易分区明显的特点,研究中继链并行化方法,通过提升单中继链性能,降低跨链交互时延.

2)交互协议层面. 针对跨链事务处理机制在多主体、长流程的场景下低效率的问题,研究单事务并行执行技术与多事务并发执行技术,通过提升跨链事务处理效率,提升跨链交互性能.

作者贡献声明: 段田田提出系统、实现系统并撰写论文;郭仪实现系统、参与实验;李博对系统进行理论分析;张瀚文、李忠诚、张珺、孙毅负责方案设计指导与论文修改;宋兆雄负责系统实现指导与论文修改.

参 考 文 献

[1] Xie Tiancheng, Zhang Jiaheng, Cheng Zerui, et al. ZkBridge: Trustless cross-chain bridges made practical[C] //Proc of the 28th Conf on Computer and Communications Security. New York: ACM, 2022: 3003–3017

[2] Belchior R, Vasconcelos A, Guerreiro S, et al. A survey on blockchain interoperability: Past, present, and future trends[J]. ACM Computing Surveys, 2021, 54(8): 1–41

[3] Ethereum Foundation and Consensys. BTC-relay: Ethereum contract for Bitcoin SPV[EB/OL]. [2023-06-06]. https://github.com/ethereum/btcrelay

[4] Lerner S D. RSK Whitepaper Overview[EB/OL]. [2022-12-05]. https://docs.rsk.co/RSK_White_Paper-Overview.pdf

[5] Frauenthaler P, Sigwart M, Spanring C, et al. ETH relay: A cost-efficient relay for ethereum-based blockchains[C] //Proc of the 2nd Int Conf on Blockchain. Piscataway, NJ: IEEE, 2020: 204–213

[6] Kwon J, Buchman E. Cosmos: A network of distributed ledgers[EB/OL]. [2023-06-06]. https://github.com/cosmos/cosmos/blob/master/WHI-TE-PAPER.md

[7] Wood G. Polkadot: Vision for a heterogeneous multi-chain framework[EB/OL]. [2023-06-06]. https://github.com/polkadot-io/polkadotpaper/raw/master/PolkaDotPaper.pdf

[8] Xu Caichao, Wang Xiaoyi, Xia Liwei, et al . BitXHub Whitepaper[EB/OL]. [2023-06-06]. https://uplo-ad.hyperchain.cn/BitXHub%20Whitepaper.pdf

[9] Berenzon D. Application-specific blockchains: The past, present, and future[EB/OL]. [2023-06-06]. https://medium.com/1kxnetwork/-application-specific-blockchains-9a36511c832

[10] Kirejczyk M, Szlachciak P, Jelski K, et al. Zero-Knowledge blockchain scalability[EB/OL]. [2023-06-06]. https://ethworks.io/assets-/download/zero-knowledge-blockchain-scali-ng-ethworks.pdf

[11] Map of Zones. Map of zones - Cosmos network explorer [EB/OL]. [2023-06-06]. https://mapofzones.com/home?columnKey=ibcVolume&period=24h

[12] China Academy of Information and Communications Technology. Blockchain Whitepaper (2022) [EB/OL]. [2023-06-06]. http://www.caict.ac.cn-/kxyj/qwfb/bps/202212/t20221229_413462.htm (in Chinese)
(中国信息通信研究院. 区块链白皮书(2022年)[EB/OL]. [2023-06-06].http://www.caict.ac.cn/kxyj/qwfb/bps/202212/t20221229_413462.htm)

[13] WeBank. WeCross technical documentation[EB/OL]. [2023-06-06]. https://wecross.readthedocs.io/zh_CN/latest/(in Chinese)
(微众银行. WeCross技术文档[EB/OL]. [2023-06-06]. https://wecross.readthe-docs.io/zh_CN/latest/)

[14] Nolan T. Alt chains and atomic transfers [EB/OL]. [2023-06-06]. https://bitcointalk.org/index.php?topic=193281.0

[15] Sean B, Daira H. HTLC implementation in the wallet[EB/OL]. [2023-06-06]. https://github.com/bitcoin/bips/blob/master/bip-0199.mediawiki

[16] Rickmar J. Decred-compatible cross-chain atomic swapping[EB/OL]. [2023-06-06]. https://github.com/decred/atomicswap

[17] Zyskind G, Kisagun C, Fromknecht C. Enigma Catalyst: A machine-based investing platform and infrastructure for crypto-assets[EB/OL]. [2023-06-06]. https://wikibitimg.fx994.com/attach/2021/06/155525475202/WBE-155525475202_15750.pdf

[18] Herlihy M. Atomic cross-chain swaps[C] //Proc of the 37th Symp on Principles of Distributed Computing. New York: ACM, 2018: 245−254

[19] Winzer F, Herd B, Faust S. Temporary censorship attacks in the presence of rational miners[C] //Proc of the 4th European Symp on Security and Privacy Workshops. Piscataway, NJ: IEEE, 2019: 357−366

[20] Tsabary I, Yechieli M, Manuskin A, et al. MAD-HTLC: Because HTLC is crazy-cheap to attack[C] //Proc of the 42nd Symp on Security and Privacy. Piscataway, NJ: IEEE, 2021: 1230−1248

[21] Nadahalli T, Khabbazian M, Wattenhofer R. Timelocked bribing[C] // Proc of the 25th Int Conf on Financial Cryptography and Data Security. Berlin: Springer, 2021: 53−72

[22] Harris J, Zohar A. Flood & loot: A systemic attack on the lightning network[C] //Proc of the 2nd Conf on Advances in Financial Technologies. New York: ACM, 2020: 202−213

[23] Goes C. The Interblockchain Communication Protocol: An overview[OL]. [2023-06-06]. https://arxiv.org/pdf/2006.15918.pdf

[24] Wood G. Polkadot Cross-Consensus Message (XCM) Format [EB/OL]. [2023-06-06]. https://github.com/paritytech/xcm-format/-blob/master/README.md

[25] Boneh D, Drijvers M, Neven G. Compact multi-signatures for smaller blockchains[C] //Proc of the 24th Int Conf on the Theory and Application of Cryptology and Information Security. Berlin: Springer, 2018: 435−464

[26] Martin R C. Design principles and patterns[EB/OL]. [2023-06-06]. http://staff.cs.utu.fi/staff/jouni.smed/doos_06/material/DesignPrinciples-AndPatterns.pdf

[27] Ross S M. Introduction to Probability Models[M]. Amsterdam, Netherlands: Elsevier, 2014

[28] Cason D, Fynn E, Milosevic N, et al. The design, architecture and performance of the tendermint blockchain network[C] //Proc of the 40th Int Symp on Reliable Distributed Systems. Piscataway, NJ: IEEE, 2021: 23−33

Duan Tiantian, born in 1996. PhD candidate. Student member of CCF. Her main research interests include blockchain technologies and distributed systems.

段田田, 1996 年生. 博士研究生. CCF 学生会员. 主要研究方向为区块链技术和分布式系统.

Guo Yi, born in 1997. PhD candidate. Student member of CCF. His main research interests include cross-chain technologies and consensus algorithms.

郭　仪, 1997 年生. 博士研究生. CCF 学生会员. 主要研究方向为跨链技术与共识算法.

Li Bo, born in 1996. PhD candidate. Student member of CCF. His main research interests include blockchain, data pricing, and performance analysis.

李　博, 1996 年生. 博士研究生. CCF 学生会员. 主要研究方向为区块链、数据定价与性能分析.

Zhang Hanwen, born in 1981. PhD. associate professor. Senior member of CCF. Her research interests include computer networks and blockchain technologies.

张瀚文, 1981 年生. 博士, 副研究员. CCF 高级会员. 主要研究方向为计算机网络和区块链技术.

Song Zhaoxiong, born in 1993. Master, engineer. Member of CCF. His main research interests include blockchain technologies and distributed systems.

宋兆雄, 1993 年生. 硕士, 工程师. CCF 会员. 主要研究方向为区块链技术和分布式系统.

Li Zhongcheng, born in 1962. PhD, professor, PhD supervisor. Senior member of CCF. His research interests include the computer networks and blockchain technologies.

李忠诚, 1962 年生. 博士, 研究员, 博士生导师. CCF 高级会员. 主要研究方向为计算机网络和区块链技术.

Zhang Jun, born in 1975. PhD, associate professor. Senior member of CCF. Her main research interests include blockchain, future Internet, and network security.

张　珺, 1975 年生. 博士, 副教授. CCF 高级会员. 主要研究方向为区块链、未来互联网和网络安全.

Sun Yi, born in 1979. PhD, professor, PhD supervisor. Distinguished member of CCF. His main research interests include blockchain, distributed systems, and network architecture.

孙　毅, 1979 年生. 博士, 研究员, 博士生导师. CCF 杰出会员. 主要研究方向为区块链、分布式系统和网络体系结构.