计算机科学 COMPUTER SCIENCE
http://www.jsjkx.com
DOI:10.11896/jsjkx.191100055

BitXHub:基于侧链中继的异构区块链互操作平台

叶少杰[1] 汪小益[2] 徐才巢[2] 孙建伶[1]
1 浙江大学计算机科学与技术学院 杭州 310027
2 杭州趣链科技有限公司 杭州 310051
(yeshaojie@zju.edu.cn)

摘 要 为了使异构区块链间的信息得以交互,实现区块链的互操作性,提出了一种通用的链间消息传输协议IBTP,并基于该协议和侧链中继策略实现了同时支持同构及异构区块链间交易的跨链技术示范平台BitXHub,其允许异构的资产交换、信息互通及服务互补。BitXHub平台由中继链、应用链以及跨链网关(Pier)3种角色构成,具有通用跨链传输协议、异构交易验证引擎、多层级路由三大核心功能特性,保证了跨链交易的安全性、灵活性与可靠性。相较于Polkadot与Cosmos,BitXHub为同构和异构应用链提供了统一的跨链合约模板,中继链含有可动态升级的验证引擎,因此具备良好的异构区块链兼容性;BitXHub基于自组网的跨链网关实现了高可扩展性,且跨链网关能够无状态转发跨链消息。实验证明,BitXHub保证了异构区块链间的异步分布式事务,实现了高吞吐、低延迟、高可扩展性、低开销的高性能。

关键词:异构区块链;互操作性;通用跨链协议;无状态;可扩展性;分布式事务
中图法分类号 TP391

BitXHub:Side-relay Chain Based Heterogeneous Blockchain Interoperable Platform

YE Shao-jie[1], WANG Xiao-yi[2], XU Cai-chao[2] and SUN Jian-ling[1]
1 College of Computer Science and Technology, Zhejiang University, Hangzhou 310027, China
2 Hangzhou Qulian Technology Co., Ltd., Hangzhou 310051, China

Abstract In order to make the information between heterogeneous blockchains interact and realize the interoperability of blockchain, a general cross-chain message transfer protocol IBTP is proposed. Based on the protocol IBTP and side-relay chain strategy, this paper constructs a highly scalable, easily compatible, dynamically upgradeable, secure and highly available heterogeneous blockchain cross-chain system called BitXHub, which realizes heterogeneous asset exchange, information interoperability and service complementarity. BitXHub consists of three roles: relay chain, application chain and cross-chain gateway called Pier. It has three core technologies: universal cross-chain transmission protocol, heterogeneous transaction verification engine and multi-layer routing. It ensures the security and flexibility of cross-chain transactions. Compared to Polkadot and Cosmos, BitXHub provides a unified cross-chain contract template for homogeneous and heterogeneous application chains and relay chain contains a dynamically upgradeable verification engine, so it has good heterogeneous blockchain compatibility. Based on distributed hash table, cross-chain gateways form ad hoc network which let BitXHub achieve high scalability, and cross-chain gateways can forward cross-chain messages statelessly. It has been verified by experiments that BitXHub guarantees asynchronous distributed transactions between heterogeneous blockchains, achieving high throughput, low latency, high scalability, and low overhead.

Keywords Heterogeneous blockchain, Interoperability, Universal cross-chain protocol, Stateless, Scalability, Distributed transaction

1 引言

区块链有利于构建一个安全可信的价值传输网络。随着区块链在各个场景中的业务需求日益复杂,异构区块链间互联互通机制的缺失很大程度上限制了区块链的应用空间。目前来看,主流区块链平台中的每条链仍然是一个独立的、垂直的封闭体系。

跨链作为连接异构区块链的桥梁,可为区块链互联网络的形成与价值孤岛的互通提供可靠的底层支撑。按照异构区块链间交互内容的不同,跨链主要可以分为资产交换和信息交换。在资产交换方面,目前异构区块链间的价值交换主要依靠中心化的交易所来完成,链上的资产价值不能得到合理的评估。而信息交换由于涉及链与链之间的数据同步和相应的跨链调用,十分复杂,目前异构区块链间无法进行信息共享,业务互通的壁垒极高。

另一方面,现有区块链技术在单链架构下本身存在着性能不足的问题,共识机制在低信任网络中达成共识十分耗时,若考虑节点作恶则处理的复杂性会更高,因此很多情况下单节点的性能无法满足商业应用场景对交易吞吐量和交易延时的需求。同时,随着运行时间的增长,区块链的存储容量将线

到稿日期:2019-10-07 返修日期:2019-12-28 本文已加入开放科学计划(OSID),请扫描上方二维码获取补充信息。
通信作者:孙建伶(sunjl@zju.edu.cn)

性增长，而目前链式的本地数据存储无法进行平行扩展，因此单链架构无法满足应用的性能、容量及其他要求。

在业务发展与技术发展面临双重制约的背景下，异构区块链间的互操作性越来越被重视，跨链已成为必要性需求和必然的发展趋势。跨链技术作为连接各区块链的桥梁，主要是实现不同区块链之间的资产原子性交易、信息互通、服务互补等功能。跨链协议的严谨描述、规范实现和普通应用，将成为实现“价值互联网”的关键。

本文的主要贡献如下：

(1)定义了兼容异构区块链的通用跨链协议 IBTP(Inter Blockchain Transfer Protocol)。

(2)实现了基于侧链中继的异构区块链互操作系统 BitXHub，其具备以下特点。

1)无状态。BitXHub 采用去中心的区块链集群中继链作为跨链交易的合法性验证和交易路由的平台。中继链的区块即为世界状态，区块中包含由应用链跨链合约动态递增生成的跨链交易流水号，并且应用链在合约中持久化最新的交易流水号以备跨链系统宕机恢复，因此跨链异步分布式事务的消息转发是无状态的，即消息丢失后可以重发或重新获取。

2)高可扩展。BitXHub 内部通过 IBTP 协议不仅可以支持单条中继链的交易验证和路由，而且支持多层级全拓扑的异构区块链网络，即中继链之间亦可以通过跨链网关(Pier)进行消息通信。IBTP 协议不仅可以保证跨链交易在同层级之间路由，也支持其跨层级的路由，从而实现跨链中继的高度可扩展。

3)异构兼容。BitXHub 为应用链提供了跨链合约模板，其为需要进行跨链的区块链系统提供了便捷易用的跨链接口。该跨链合约的设计不会对现有应用链系统造成任何侵入性操作，同时便于区块链跨链的快速接入。应用链中相应的 SDK 升级时，只须通过插件的形式对跨链网关进行简便升级，无须对中继链进行改动。对于新加入异构应用链或应用链跨链消息验证规则的变化，中继链仅通过 Wasm 虚拟机动态升级即可。

4)鲁棒高可用。BitXHub 的鲁棒高可用性主要通过交易日志及中继链进行保障。在任意组件发生崩溃的情况下，系统均可以通过应用链或中继链日志进行动态失效恢复，以保证跨链系统的正常运行。

5)安全高可靠。BitXHub 设计了可验证数据结构，并将其融入 IBTP 协议的 Proof 字段中。跨链网关不具备应用链节点的私钥，仅做无状态的消息转发，因此作恶能力有限。中继链中含有可动态升级的验证引擎，可验证跨链交易的可靠性。

(3)对比了几种跨链方案的优点与不足。相较于 Polkadot 与 Cosmos，BitXHub 设计了符合 IBTP 协议的供应用链使用的跨链合约模板，中继链的验证引擎可对异构应用链验证逻辑动态升级，因此其对异构区块链的兼容性较强；BitXHub 实现了区块链间无状态的消息转发；BitXHub 中抽象出的跨链网关组件实现了跨链交易的多层级路由，进一步提升了区块链价值网络的可扩展性。

2 国内外研究现状

目前国内外有关跨链技术的研究主要演化出 4 种策略：中间人机制、侧链与中继机制、哈希时间锁定机制[1]以及分布式私钥控制技术。中间人机制即为一个团体或一组团体同意原子性[2]地执行链 A 上账户数字资产的减少以及链 B 上接收账户数字资产的增多，此团体是 A 与 B 信任的。侧链机制即某链可以验证和读取另一条链上所发生事务的事件。侧链中继机制为中继链，其不仅作为应用链的侧链，还负责跨链消息的验证与转发。哈希时间锁定机制的核心思想为在规定时间内提供与哈希值对应的原内容解冻资产。分布式私钥控制技术中的私钥由用户与去中心化网络共同管理，跨链时锁定原链资产，在目的链上发起智能合约，创建新的资产。

最初，Interledger[3]遵循公证人机制，提供具有公信力的 Escrow(第三方存管)及 Notary(公证人)，任何拥有两个及以上数字资产账户的人都可以成为 connector。后期，为了在确保实现跨链交易事务原子性的基础上脱离可信中间人的束缚，使用哈希时间锁定的方式跨链。前者的不足之处在于需要可信的第三方中介，会导致中心化的权力问题；后者对于联盟链来说，资产冻结的证明条件难以被外界访问，会造成交易对手难以获知跨链交易有效性的问题。

关注度最高的 Cosmos[4]与 Polkadot[5]都属于侧链中继模式。

Cosmos 使用 Tendermint[6-7]共识算法，是一个支持异构区块链跨链的网络。Cosmos 的优点在于通过 IBC 协议和 Tendermint 解决了跨链交易的安全与可信传输的问题；同时，跨链交易以 Merkle Tree 的形式记录在区块头，提供了可验证性。

Polkadot 的项目定位是实现一个可伸缩的异构多链系统。Polkadot 为了解决共识架构中一致性和有效性紧耦合导致共识时间过长而影响系统吞吐量的问题，提出了合并安全性的解决方案，即 Polkadot 中中继链会随机分配验证者到平行链中参与区块的确认，此时中继链与平行链共享安全性来保证跨链交易的有效性；为了解决跨链系统可扩展性[8]的问题，其通过互不信任节点间的经济激励机制进行解耦。Polkadot 系统尚处于开发状态，在技术可行性和社会治理方面都需要时间的考证。

另外，Layer-2 层面的闪电网络[9]的链下支付通道以及 Plasma[10]中的去中心化自治激励等技术亦值得关注。

3 异构区块链互操作性的挑战

在设计使得交易能够从 A 链安全可信地抵达 B 链的跨链系统时，针对公证人机制对中介的强依赖会降低安全性、哈希锁定机制现阶段难以证明资产锁定的有效性，以及分布式私钥技术可行性缺失的问题，本文引入侧链中继的策略。下面介绍该过程中遇到的挑战及针对性解决方案。

3.1 Polkadot 与 Cosmos 的缺点

Polkadot 和 Cosmos 是基于侧链中继策略的跨链系统。Polkadot 的缺点：不完善的代币经济治理策略，拥有一定数量代币的人可以控制跨链网络；中继链中的验证者角色节点对任一应用链的世界状态变更都有重要影响，尽管有钓鱼者的监控，但风险敞口依旧很大；共享安全，应用链与中继链区块同时生成，这要求大量更改应用链的源码，因此近期 Polkadot 推出 substrate，意图统一应用链的结构。Cosmos 的缺点：

Cosmos 跨链网络对异构应用链的兼容性不好，及时确认机制的链需要添加 IBC 协议，而最终确认链还需更加复杂的操作(PegZone 代理链)。

3.2 跨链系统分布式事务的 ACID 保证

ACID 指代 Atomicity(原子性)、Consistency(一致性)、Isolation(隔离性)和 Durability(持久性)。

两个账本之间的变动必须是原子性的，否则会出现双重支付或者价值丢失的问题。互联网信息传递是无状态的，网络包可丢失重发。但目前区块链之间的消息传递是有状态的，以在 A 链上某账户减 1 元钱而 B 链上某账户上增加 1 元为例，此交易在 A 链上发生，但若 B 链未接收到相应的执行命令，则不能保证交易的原子性，从而导致 A，B 链资产总量与之前不一致。

在进行信息交换，或者进行资产转移并且交易对手所在区块链的模型是基于账户的情况下，本文提出的观点是：让区块链之间的消息传递也如互联网的消息传递一样无状态化。首先，区块链系统自身会维护一致性；其次，跨链交易传输过程中引入中继链，中继链的区块持久化存储整个跨链系统的世界状态，利用日志恢复和系统的高可用设计保持跨链交易的最终一致性。任意跨链系统组件发生崩溃或宕机重启，均可以通过中继链上的状态日志进行恢复，并进行消息重发。因此，跨链消息是无状态的，最终一致性保证了原子性，即要么都执行，要么回滚反向执行已执行的操作。

在进行资产转移并且交易对手所在区块链的模型是基于 UTXO 的情况下，本文提出将分布式跨链事务分成多笔交易执行，即引入中间生成的、仅有目的链某账户可使用的 UTXO。

跨链交易的隔离性由区块链自身维护，即通过有向无环的跨链事务依赖图进行并发控制。跨链交易的持久化由中继链的区块维护。

3.3 异构区块链跨链系统的兼容性保证

为了兼容各异构区块链的接入，本文设计了 IBTP 通用跨链协议，为应用链提供符合 IBTP 跨链协议的跨链合约模板，应用链仅需实现模板中的函数接口，即可接入跨链系统。

本文遵循模块化松耦合设计，提出跨链网关组件，将与异构区块链打交道的逻辑从中继链抽离出来放入跨链网关中，以减轻中继链各节点冗余计算和冗余存储的开销；由跨链网关定制化地与各异构应用链对接，从而提升跨链系统的兼容性、可扩展性。

本文在中继链中引入高效可插拔的验证引擎，该引擎采用 WebAssembly[11] 实现中继链的易升级，升级内容为新加入的异构应用链抛出的跨链交易的验证逻辑，以及验证所需公钥。具体过程如下所述：向中继链发送一笔 WebAssembly 升级系统的智能合约交易，通过共识改变所有节点的代码逻辑。

3.4 异构区块链互操作的安全性保证

捆绑多个不同风险偏好、不同伸缩性需求、不同隐私需求的应用链会影响跨链系统的整体安全性。例如：比特币[12-13]出一个块需要 10 min，确认一个块内的交易需要 60 min，可能发生回滚；以太坊[14-15]曾因 the DAO 事件发生过硬分叉，跨链系统整体的安全性取决于最小安全的区块链。本文的观点是：中继链的区块持久化跨链系统的世界状态。已经在中继链落块的交易会被执行，若任一异构区块链单方面出现安全问题，中继链会检测出该应用链抛出事件中的基于跨链合约定序的跨链交易索引不一致，从而拒绝继续执行与该链有关的交易，由此中继链维护了跨链系统的整体安全性。

跨链网关作恶能力有限。跨链网关由各应用链维护，仅无状态地转发跨链消息，其转发成功到哪一条跨链消息的状态记录在中继链上。若跨链网关被黑客攻击或自身作恶，其仅可能不再转发消息，或转发有跨链网关自身签名但没有应用链节点背书签名的会被中继链验证引擎否决的消息。BitXHub 设计了可验证数据结构，并将其融入 IBTP 协议的 Proof 字段中，中继链里含有可动态升级的验证引擎，可验证跨链交易是否确实来自于某应用链。因此，无状态的跨链网关的作恶能力有限。

伪装成大量中继链和应用链节点的女巫攻击[16]不可能发生，因为中继链的物理部署架构是每一条应用链都需要承担一个或多个对等节点的维护工作。例如：部署时包含 4 条应用链，一条中继链中包含 4 个节点时，每条应用链管理一个对等节点，在默认情况下，中继链 4 个节点都会实时冗余计算和存储数据。

4 跨链系统 BitXHub 的设计与实现

根据上文描述，本文提出一个基于侧链中继策略的通用跨链技术平台 BitXHub，其中 Bit 代表区块链价值孤岛，Hub 代表区块链间价值传输的通道，X 代表高可用、可扩展、易升级运维等特性。

4.1 BitXHub 的整体架构

BitXHub 的整体架构如图 1 所示，其由中继链、应用链、跨链网关 3 种角色构成。

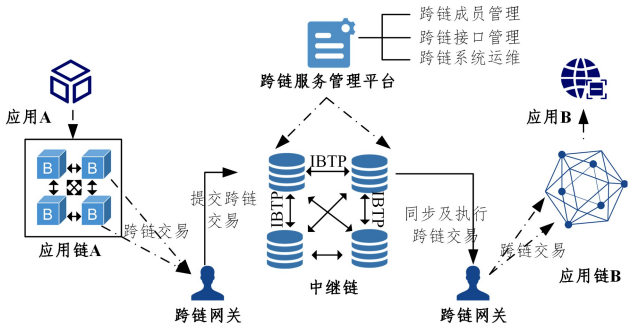

图 1 BitXHub 架构图

Fig. 1 Architecture of BitXHub

(1)应用链：负责具体的业务逻辑，分为两种。1)同构应用链：支持 BitXHub 跨链协议结构的区块链。同构应用链具有类似的区块、交易数据存储格式，且具有相同的共识算法和安全算法等。2)异构应用链：不一定满足 BitXHub 直接支持的区块数据存储结构、安全算法和共识逻辑的区块链，如 Fabric[17]，Hyperchain 等。

(2)中继链：BitXHub 系统的核心。中继链作为已接入 BitXHub 的异构区块链的侧链，主要负责应用链之间的跨链 IBTP 消息的验证、持久化以及路由。中继链和多个应用链组成联盟，因此中继链各节点由各应用链共同维护，是联盟链。中继链采用插件式共识，将 Hyperchain 的 RBFT，POS[18]，Polkadot 的 GRANDPA，AVA[19]，Algorand[20]，Cosmos 的 Tendermint 等共识算法作为插件接入，并根据应用链需求动态切换。

(3)跨链网关：位于应用链和中继链之间的独立系统，主要负责跨链交易的收集以及跨链交易的转发等任务。跨链网关不仅可用于应用链与中继链之间，亦通过 DHT(Distributed Hash Table)[21] P2P 自组网的方式实现跨链交易在中继链与中继链之间的路由。

(4)跨链服务管理平台。跨链服务管理平台主要提供以下功能：1)跨链成员管理——审核应用链；2)跨链接口管理——查询某一笔交易的状态、查询目标区块链是否在线、提供接入 BitXHub 的跨链合约的相关模板界面等；3)跨链系统运维——使用监控工具实时检测系统情况，并实时展现跨链交易热力图。

为进一步提高 BitXHub 的可扩展性，本文提出 BitXHub 多层级整体架构，如图 2 所示。中继链和多个应用链组成联盟，联盟间通过 IBTP 通用跨链协议进行跨链交易的路由与验证，中继链之间亦可通过 IBTP 协议，经 P2P 自组网的跨链网关转发交易，实现跨链中继的高度可扩展。应用链与中继链之间，以及中继链与中继链之间，通过跨链网关对符合 IBTP 协议的消息进行转发。

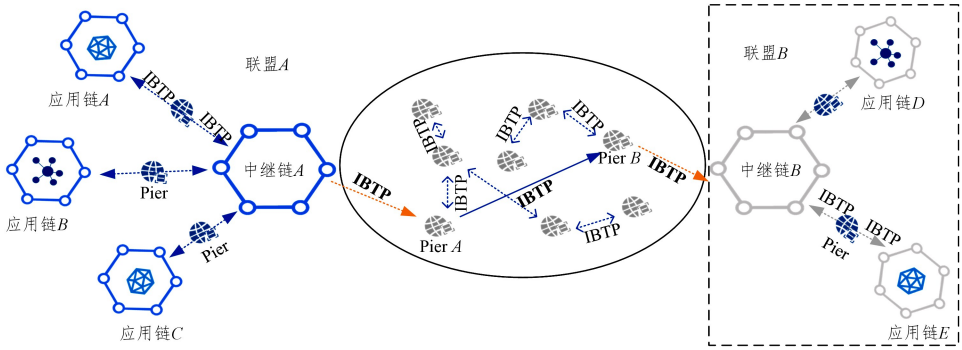

图 2 多层级全拓扑 BitXHub 架构图

Fig. 2 Multi-layer full-topology BitXHub architecture diagram

4.2 跨链协议 IBTP

IBTP (Inter Blockchain Transfer Protocol)是本文提出的一种通用的跨链交互的消息传输协议。鉴于异构应用链的共识算法与验签方式不同，为了兼容各异构应用链，本文提出类似于 TCP/IP 协议的 IBTP 协议用于应用链和中继链之间的交互以及多层级中继链间的交互。IBTP 的具体功能包括：提供跨链交易的验证证据；提供跨链交易的路由凭证。IBTP 的详细结构如图 3 所示。

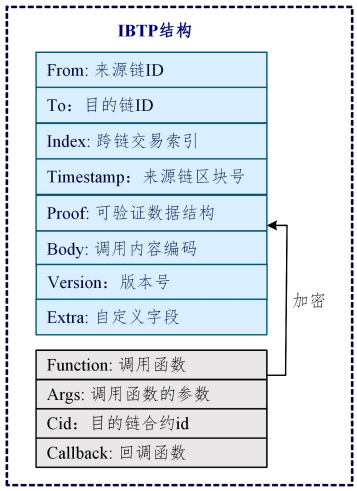

图 3 IBTP 的结构

Fig. 3 Structure of IBTP

图 3 中，From 和 To 字段代表来源链和目的链的唯一 ID 标识，此 ID 标识在应用链向中继链注册时由中继链生成。Index 是由应用链跨链合约动态逐一递增生成的跨链交易流水号，即索引。Timestamp 是来源链的区块号，代表时间戳。Proof 是一个相对灵活的可验证数据结构。Body 由需要目的链某合约调用函数以及函数参数、目的链合约 id 和 Callback 回调函数加密而形成。Version 是 IBTP 协议的版本号，便于后期升级与版本控制。Extra 是可由不同应用链自行定义的自定义字段。

基于同构应用链和增加跨链树的异构应用链的视角，Proof 字段由 Hash，Path，Merkle Root 和 Signature 经 json marshal 后得到。其中，Hash 是图 4 所示的叶子节点 Cross Tx 的 Hash 值，path 是类 SPV(Simplified Payment Verification)验证时 Hash 的 SPV 路径兄弟节点 Hash 值的集合，Merkle Root 是图 4 中的 Cross Tx Hash 节点，Signature 是对 Cross Tx Hash 的签名。根据 Merkle Tree 的构成原理，Hash 与路径上的兄弟节点哈希上行，会得到 Cross Tx Hash。因此，中继链的验证引擎会先验证签名是否正确，再验证该交易 Hash 是否真实存在。

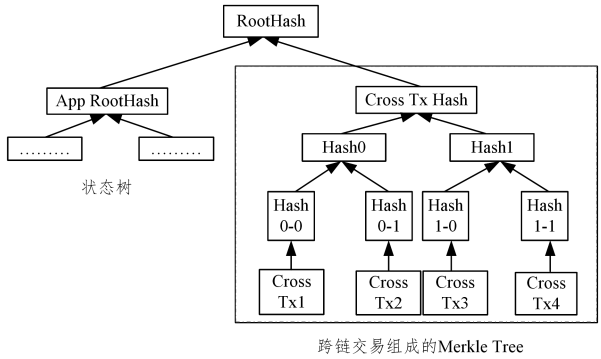

图 4 可验证数据结构：优化后的 Merkle Tree

Fig. 4 Verifiable data structure: optimized Merkle Tree

基于未增加跨链树的异构应用链的视角，如 Hyperledger Fabric，Proof 字段由单个跨链交易的 Hash 和 Endorsement 阶段后产生的背书签名序列 Marshal 而成。

以上涉及的同构应用链和异构应用链不包括概率确认共识策略和不具备节点签名功能的区块链，下面以以太坊为例进行分析。以太坊上是基于概率性确认机制的 PoW 共识[14,22]，Pier 中刚收到一个跨链交易时，并不能确认这个交易是否为有效交易，因此需要等待多个区块时间（默认是 20 个），使得该跨链交易在概率上确实存在之后，再将跨链交易转发到中继链上。

另外，考虑到单个 Pier 容易产生作弊行为，如谎称交易已经确认等情况，需要采用 Pier 集群的方式来增强可信度，每个 Pier 由权威机构进行背书；并且在中继链上引入对作弊节点的惩罚机制，通过经济激励的方式防治多个 Pier 联合起来进行作弊；同时，由于以太坊本身对跨链交易没有签名，因此需要多个 Pier 确认跨链交易存在，并对跨链交易进行签名后再转发到中继链上。

4.3 中继链的交易类型

中继链支持 3 种交易类型：系统交易、跨链交易和智能合约交易。

系统交易在中继链中使用系统内置合约进行执行。跨链交易是中继链的主要交易类型，所有从应用链抛出来的事件都会被转换成跨链交易。智能合约交易是需要在 BitXHub 中继链的特定虚拟机（WebAssembly，lua）执行的一种交易，其执行逻辑按照智能合约而定。目前，该智能合约服务于动态升级中继链的验证引擎。

4.3.1 系统交易

系统交易是通过调用中继链内置合约进行执行的一类交易。目前，中继链包括应用链管理等内置合约。应用链管理内置合约包括应用链注册、应用链审核和应用链跨链业务注册等几类操作。

（1）应用链注册。获得中继链的准入资格后，应用链才可以进行后续的跨链业务注册操作。如图 5 所示，应用链管理者生成一对公私钥，将应用链的验证者集合、应用链的共识策略、应用链管理者的公钥和应用链管理者的签名信息通过系统交易的方式发送给中继链，跨链服务管理平台从中继链同步信息，中继链管理者在平台上对应用链的注册进行审核，结果由平台以系统交易的形式发送给中继链，中继链再将其反馈给应用链管理者。

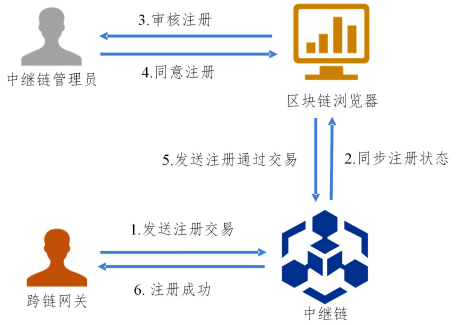

图 5 应用链注册的示意图

Fig. 5 Schematic diagram of application chain registration

（2）跨链业务注册。应用链注册完成后，需要进行跨链业务注册才可以实现应用链合约层面的跨链操作。中继链会根据注册信息生成一个唯一的 ID 返回给应用链管理者。

4.3.2 跨链交易

应用链需要跨链时，会抛出事件，然后由跨链网关封装成跨链交易并提交到中继链。跨链交易在中继链按以下顺序执行：交易池去重，中继链各节点共识落块，跨链交易验证。如图 6 所示，从交易池已验证交易中取出交易列表构建可验证数据结构并进行交易路由。中继链执行跨链交易时按照 DAG（Directed Acyclic Graph）[23-24] 分布式事务依赖图并行执行。中继链构建的可验证数据结构如图 4 所示。

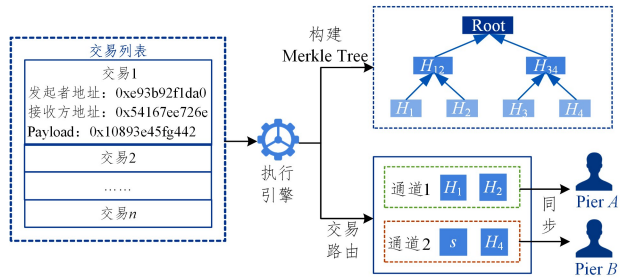

图 6 中继链构建可验证数据结构及交易路由示意图

Fig. 6 Relay chain constructs verifiable data structure and transaction routing

4.4 中继链

4.4.1 中继链验证引擎

中继链高效可插拔的验证引擎，支持应用链所提交的跨链交易可信验证和可靠路由，支持基于动态注入的验证规则对相应应用链提交的证明进行验证。

中继链需要对每一笔跨链交易进行校验，以防止出现交易被伪造或者篡改的情况。BitXHub 的验证引擎通过智能合约的方式管理多种复杂逻辑的验证规则，支持验证规则的在线升级和改造；能应对各种不同类型的区块链对对自身验证规则的迭代，支持对不同区块链验证规则的检验。

如图 7 所示，验证引擎分为 3 个模块。

（1）协议解析模块：验证引擎内部对 IBTP 协议的解析模块。协议解析模块会对传入中继链的跨链交易进行解析。由于所有跨链交易是遵循 IBTP 的，协议解析模块会提取来源链和验证证明信息作为后续验证引擎的输入。

（2）规则匹配模块：验证引擎根据解析模块解析出的来源链匹配对应的应用链的验证规则，应用链的规则是在应用链注册时注入的。每当有新的应用链注册到中继链时，都需要将链的信息和验证规则对应的脚本注入验证引擎中。

（3）规则执行模块：验证引擎的核心。规则执行模块的主要部分是 WASM[11] 的虚拟机，通过 WASM 虚拟机可以动态加载不同验证规则的脚本，实现了连入中继链的不同区块链验证规则的热更新。

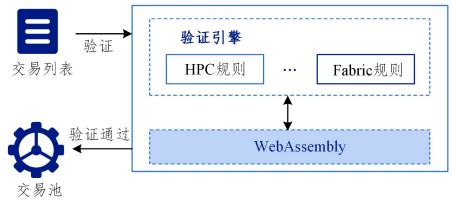

图 7 中继链验证引擎示意图

Fig. 7 Relay chain verification engine schematic

验证引擎的运作流程如下：

(1)当新的应用链注册到中继链时,需要通过合约的方式将应用链对应的验证规则上传。

(2)应用链产生一笔跨链交易,中继链接收之后,将跨链交易传入验证引擎,验证引擎解析 IBTP 协议以后,获知来源链的 ID,并根据 ID 检索出对应的验证规则脚本。

(3)检索到对应的验证规则脚本后,验证引擎会用虚拟机调用规则脚本,虚拟机会对 IBTP 解析出的 Proof 字段进行校验,从而确定交易的合法性。

基于上述结构,中继链的异构交易验证引擎具有诸多优势。1)高效:通过 WASM 虚拟机保证了验证规则执行的高效性。2)灵活:验证规则依据不同应用链规则的变化快速、低成本地更新。3)全面:满足各类型区块链的可信验证。4)便捷:应用链的业务人员可以直接管理验证脚本中的校验规则。5)安全:对 WASM 虚拟机设置安全限制,只能调用引擎自身所允许的函数和库。

4.4.2 中继链交易路由

路由模块可以路由跨链交易及其回执。跨链交易验证通过后,一个区块中所有要被路由的跨链交易会被构建成 Merkle Tree;Merkle Root 在经过验证节点的签名后被发送到目的链执行或者被发往下一跳即另一条中继链。当发往另一条中继链时,需要在 Proof 字段中添加本中继链的可验证数据结构及签名,以便在下一跳验证跨链交易的可靠性。

但是,中继链不负责具体的发送,而是将所有的跨链交易分门别类地放到对应的队列中供跨链网关获取和调用。中继链启动时,会为每个注册的应用链创建固定大小的跨链交易队列。跨链交易执行时,会按照目的链的 ID 插入不同的跨链交易队列,跨链网关取走的跨链交易会从跨链交易队列中删除。

跨链网关 Pier 可以并行执行不同来源链的交易。执行完的回执分为两种情况:1)没有回调的跨链交易对应的回执在返回中继链后,就完成了整个跨链流程;2)有回调的跨链交易对应的回执会经过中继链的排序打包,然后返回到来源链。

4.5 支持跨链交易的应用链

异构应用链接入跨链系统时,必须确保以下问题得到解决:如何构建跨链可验证数据结构;如何在跨链系统中添加由应用链共识算法决定的跨链交易可靠性验证条件;如何外部捕获应用链抛出的跨链交易。于是,符合 IBTP 协议的应用链合约模板、异构应用链验证引擎、事件监听的策略应运而生。

4.5.1 跨链合约协议

跨链合约是指应用链上有跨链行为的智能合约或一段脚本。为了兼容 IBTP 通用跨链协议,且方便开发者开发跨链合约,系统制定了一个跨链合约模板。该模板主要包括如下两个函数。

(1)跨链调用函数

function crosschainInvoke(string from, string to, string func,string args,string callback);

to 是目的跨链业务 ID,func 是跨链交易要调用的函数名,args 是跨链交易要调用的参数,callback 是回调函数名。

(2)获取跨链交易哈希

function getCrosschainTxHash(uint begin,uint end) returns(bytes32[])

begin 是获取跨链交易范围的开始索引,end 是获取跨链交易范围的结束索引。返回值是跨链交易的交易哈希数组。跨链交易的交易序号由 From_id-To_id-index 唯一标识,且该序号由跨链合约动态、逐一递增生成。应用链跨链合约持久化由交易序号索引的跨链交易哈希,以备 BitXHub 宕机恢复使用。Hyperchain 异构应用链上跨链合约的实现如下:

```
1. contract CrossChain {
2.     mapping(uint=>bytes32) crosschainM;
3.     uint index=0;
4.         TxHashAccessor hasher = TxHashAccessor (0x0000000000000
           0000000000000000000000000fa);
5.     event throwEvent(uint index,string func,string args,string callback);
6.      function crosschainInvoke(string to,string func,string args,string
        callback) {
7.      index++;
8.      crosschainM[index]=hasher.getHash();
9.      throwEvent(index,func,args,callback);
10.     }
11.     function getCrosschainEvent(uint begin,uint end) returns(bytes32[])
        {
12.       bytes32[] memory ret=new bytes32[](end-begin);
13.       for (uint i=0; i < end-begin; i++) {
14.         ret[i]=crosschainM[i+begin];
15.       }
16.       return ret;
17.     }
18. }
19. contract TxHashAccessor {
20.     function getHash() returns(bytes32);
21. }
```

4.5.2 Merkle 树的优化

为了方便支持跨链交易的验证,需要进一步对传统的区块链 Merkle 树进行优化,如图 4 所示:除了内部状态树之外,额外增加了跨链树。其中,跨链树用于外部进行跨链交易的类 SPV 验证。

其中,RootHash 的计算包括两个部分:一个是区块执行后的一个状态树根哈希,另一个是由跨链交易组成的一棵 Merkle 树计算出的跨链树哈希。RootHash 被包含在区块头里,然后进行共识。因此,随着区块的共识,跨链交易也被唯一确定下来,这就为中继链的跨链存在性证明提供了基础。应用链的 Merkle 树如果不修改,跨链交易的存在性证明将不够充分,但是依然可以通过签名的方式实现相应的效果,如 Hyperledger Fabric 在交易背书阶段会对 Proposal 的跨链交易进行签名,因此类似 Fabric 的异构应用链是不需要修改应用链源码的。

4.6 跨链网关

在复杂跨链的场景下,应用链如何便捷地接入跨链系统以达到跨链系统的良好扩展性对于激发跨链系统生态的活力至关重要。跨链网关 Pier 是配套设计的一个能够满足异构区块链接入跨链系统便捷性和多层级跨链系统扩展性的一个关键性接入口。

本质上来说,Pier 是一种在多区块链系统中用于收集和传播交易的中间模块,是连接不同区块链系统的交互组件。设计上,Pier 充当如下两个不同的角色:作为应用链与中继链的桥梁,或作为中继链与中继链之间的桥梁。

第一种角色是中间人,用于桥接应用链和中继链,处在单

个中继链的层级上。其核心设计原理是：将与应用链打交道的逻辑抽象到跨链网关中，以减轻中继链冗余计算和存储的开销。该逻辑包括调用应用链的 SDK 与应用链交互的逻辑，如事件监听、发送交易、取回丢失的跨链交易等。第二种角色中，Pier 在多个中继链系统中起到“路由器”的作用。Pier 与 Pier 之间形成 P2P 的路由网络，每个 Pier 包含了一部分跨链路由表，整个 Pier 网络通过分布式哈希表 Kademlia DHT 的方式实现多层级跨中继链的跨链交易的高效传播。

4.6.1 跨链网关的插件机制

现有的区块链系统在架构设计上存在差异，这些异构的区块链如何快速、便捷地接入跨链系统一直是比较棘手的问题。

Plugin 模式的一大特点是将 Pier 与应用链的交互模块和 Pier 核心模块进行解耦，从而方便更多的应用链加入跨链系统。如图 8 所示，Plugin 作为一个与 Pier 相对独立的模块，通过 Pier 运行时动态加载的方式使用。为了能实现与应用链的交互能力，Plugin 需要根据不同区块链的机制具体实现特定的接口。接口需要满足以下功能：监听相应区块链上的跨链事件并将其传递给 Pier 处理；执行来自于其他区块链的跨链请求；能够查询相应区块链上已收到和已执行的跨链请求的状态。

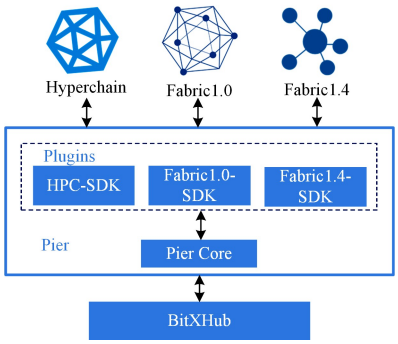

图 8 跨链网关的架构

Fig. 8 Architecture of Pier

采用 Plugin 机制有如下优势：方便更新，能够在不停止 Pier 的方式下更新 Plugin；方便适配不同的区块链，而不用改动 Pier。

4.6.2 跨链网关的功能

跨链网关在充当连接应用链和中继链以及连接不同的中继链两种不同角色时，主要抽象出以下 5 种功能。跨链网关的消息传递都是无状态的，可以重发或重新获取。

(1)收集来源链跨链交易。跨链网关需要监听应用链的事件抛出，如果出现丢失，还须到应用链获取丢失事件。Pier 在对接具体的应用链时，不仅仅要做到对交易的实时监听，更要保证跨链交易的有序性。

(2)提交交易至中继链。本文使用 GRPC[25] 通信方式在跨链网关上实现 Client，在中继链上实现 Server 端，以便通过短连接将交易提交至中继链。

(3)从中继链接收交易。跨链网关作为中继链的轻节点，不断同步中继链的区块。跨链网关仅需在启动时向中继链发起区块消息订阅，便可等待中继链通过 GRPC 长连接流向跨链网关推送跨链交易。

(4)转发交易至下一跳网关。通过 DHT 寻址下一跳网关，其中下一跳网关的地址由跨链网关组网中的分布式路由表确定。该分布式路由表记录跨链网关所管辖的中继链上的应用链 ID 集合。

(5)提交目的链交易执行。跨链网关在调用目的链合约时，如果出现失败的情况，则会采用最终一致性的方式不断尝试，直到成功为止。

4.6.3 跨链路由网络

在跨中继链进行跨链时，需要一个跨链路由网络进行跨链交易传递。跨链路由网络是一个由多个 Pier 组成的 P2P 网络。

每个 Pier 会维护一个跨链的路由表，路由表记录的是应用链和中继链的归属关系。Pier 在加入网络时，需要广播自己所属中继链的应用链信息，其他节点会根据这些信息进行路由表的更新。一个网络中的 Pier 需要传递跨链交易时，先通过分布式路由算法查找跨链交易的目的地址所属的 Pier 并将跨链交易发送给它；目的 Pier 根据跨链交易所属的中继链的验证者信息验证跨链交易的真实性，验证通过后再将跨链交易发送到相应的中继链中。跨链交易的执行结果也通过这种方式返回到来源中继链。

4.7 BitXHub 简易跨链交易的处理流程

本节以应用链 A、一条中继链、应用链 B 的跨链交易为例，来阐述 BitXHub 简易跨链交易的处理流程，如图 9 所示。

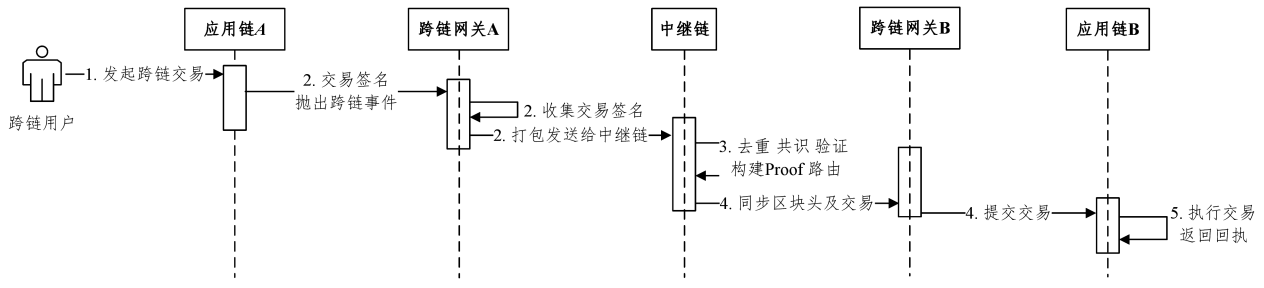

图 9 BitXHub 简易跨链交易的处理流程

Fig. 9 Simple cross-chain transaction processing flow in BitXHub

(1)应用链 A 上的跨链用户向应用链 A 发送交易指令(该指令的执行函数内包含跨链调用)；

(2)跨链网关 A 通过应用链 A 抛出的事件接收到跨链交易信息，并在收集到足够的交易签名及交易之后打包发送到中继链；

(3)中继链接收到跨链请求后进行交易的可靠性验证，验

证通过之后将其发送到共识模块参与中继链共识,最后将跨链交易从交易来源方队列存储到交易抵达方队列,即交易路由;

(4)跨链网关 B 同步来自中继链的区块头及跨链交易并将其提交至应用链 B;

(5)应用链 B 执行相应的跨链交易,如果跨链交易有回执,则反向执行步骤(2)—步骤(4)。

上述简易跨链交易案例仅考虑了一条中继链,倘若目的链不在该中继链上(设该中继链为 A 链),则需要中继链 A 将跨链交易转发至 P2P 自组织跨链网关中,再经 DHT 寻址;若目的链在某跨链网关所管辖的中继链 B 上,则可以进行跨链消息转发,此时中继链 B 须先验证中继链 A 转发跨链消息的可靠性,再经跨链网关将其提交给目的链执行。上述跨链交易的转发全程无状态,因为有来源链跨链合约上的跨链交易流水号和中继链最新的流水号做顺序的确认。上述跨链交易路由流程中,任何一步发现目的链不存在,就反向执行来源链的交易,即回滚。

5 BitXHub 测试

5.1 实验环境

实验在 20 个服务器节点上进行,每个节点的配置信息为:CPU 包含 4 个 Intel(R) Xeon(R) CPU E5-2660 2.20GHz 的核,32GB RAM,1TB 的机械硬盘,操作系统为 Ubuntu14.04。

部署 Hyperchain 集群、Hyperledger fabric 集群作为应用链。部署一条中继链集群,为模拟真实的跨链应用场景,将中继链集群的节点数与接入应用链数量的关系设置为:2~4 个应用链时,中继链节点为 4 个,此后每增加 1 条应用链,中继链节点增加 1 个。部署跨链网关节点,一条应用链部署一个跨链网关。

5.2 实验过程与结果

本节从宏观层面的系统吞吐量、交易延迟、可扩展性以及微观层面的内存、CPU、网络带宽、存储资源方面考量 BitXHub 的性能。实验中将每秒发多少笔交易作为自变量,将 BitXHub 每秒能处理多少笔交易(TPS)、交易平均延时(latency)作为因变量。实验结果均不考虑应用链上的开销。

5.2.1 吞吐量及延迟测试

(1)吞吐量测试。对于跨链交易,实验过程为通过逐步增加发送简易跨链交易的速率,测试 BitXHub 的峰值吞吐量。其中,发送的简易跨链交易为 Hyperchain 上某账户减少一个单位资产,Fabric 上某账户增加一个单位资产。实验结果如图 10 所示。

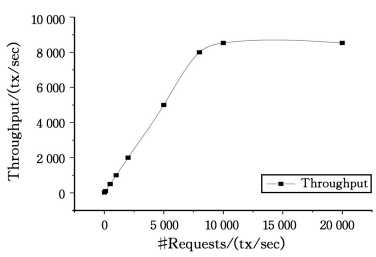

图 10 吞吐量随简易跨链交易请求速率的变化情况

Fig. 10 Throughput changes with simple cross-chain transaction request rates

(2)延迟测试。中继链处理一笔系统合约需要 30ms。BitXHub 处理一笔简易的从 Hyperchain 发向 Fabric 的跨链交易,除去 Hyperchain 以及 Fabric 处理交易的时间,平均延迟为 232ms,且最终一致性会影响到跨链交易的延迟;其次,现实世界中随着各个异构或同构应用链的接入,BitXHub 的网络拓扑结构会愈发复杂,分布式网络中各个节点之间的连通性以及连接的稳定性亦会影响到交易的延迟。

5.2.2 可扩展性测试

BitXHub 的可扩展性表现为随着接入应用链数量的增多,BitXHub 的吞吐量和延迟的变化情况。实验过程为仅用一条中继链集群,不断增加 Fabric 和 Hyperchain 应用链集群的数量。通过不断调高多链间发送简易跨链交易的速率,测试某数量应用链集群的峰值吞吐量,实验结果如图 11 所示。

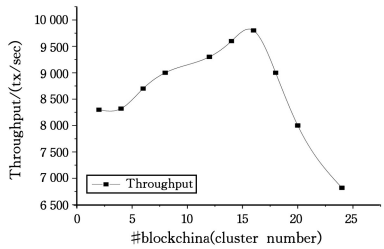

图 11 峰值吞吐量随接入应用链数量增多的变化情况

Fig. 11 Peak throughput changes as number of access application chains increases

可以看出,BitXHub 接入应用链的数量在 16 条以内时,性能峰值由 8300TPS 逐步上升至 9800TPS;峰值自 16 条开始逐渐下降,这缘于中继链里面 BFT[26] 的网络开销;尝试接入 49 条应用链时,BitXHub 出现不响应的现象,但是跨链交易的最终一致性在长时间的等待后依旧可以保持。

5.2.3 微观测试

本节使用 pprof 进行 BitXHub 微观测试,结果表明 BitXHub 在中继链共识时有一定的网络开销;在 CPU、内存、硬盘存储资源消耗层面,中继链以及跨链网关消耗很少,可以忽略。例如,跨链网关仅仅是无状态的消息转发,因此无须消耗硬盘存储资源。

结束语 BitXHub 为公链和联盟链提供了一套全新的、透明可信的跨链技术方案,基于通用跨链消息传输协议,IBTP 打造了一个异构区块链跨链示范平台。该平台秉承去中心、可扩展、高可用、易接入的设计理念,为链上的资产、数据、服务开拓价值互通的渠道,助力区块链技术从"链孤岛"到"链网络"的发展。

BitXHub 基于侧链中继策略,与 Polkadot 和 Cosmos 相比,定义了通用跨链协议 IBTP,实现了无状态的跨链消息转发,并基于跨链网关实现了多层级的跨链消息路由,由此构建了高可扩展的、兼容性良好的区块链价值网络。经实验佐证,BitXHub 能够保证异构区块链间的异步分布式事务,并保持高吞吐、低延迟、高可扩展性、低开销、稳定安全的性能。

未来计划对跨链交易的有效性检验做进一步研究,跨链交易有效性验证的难点在于如何在跨链系统中添加权威准确、不可篡改并可接受审计的外部可信数据源信息即预言机;另一方面,跨链交易在传输过程中可能会失效,资产冻结时间

在分布式多机间难以协同，未来可能需要结合类 UTXO 的数据结构做资产冻结。未来还需对 BFT 的网络开销进行进一步研究，改进 $O(n^2)$ 的网络拓扑，或通过分片的方式提高共识算法的可扩展性。

目前 BitXHub 已开源，本文希望携手相关研究人员共创区块链价值网络。

参 考 文 献

[1] BUTERIN V. Chain interoperability[EB/OL]. [2016-9-9]. https://allquantor.at/blockchainbib/pdf/vitalik2016chain.pdf.

[2] HERLIHY M. Atomic cross-chain swaps[C]// Proceedings of the 2018 ACM Symposium on Principles of Distributed Computing. ACM, 2018: 245-254.

[3] HOPE-BAILIE A, THOMAS S. Interledger: Creating a standard for payments[C]// Proceedings of the 25th International Conference Companion on World Wide Web. International World Wide Web Conferences Steering Committee, 2016: 281-282.

[4] KWON J, BUCHMAN E. A Network of Distributed Ledgers Cosmos [EB/OL]. https://static.coinpaper.io/files/whitepapers/atom-cosmos_whitepaper.pdf.

[5] WOOD G. Polkadot: Vision for a heterogeneous multi-chain framework[EB/OL]. https://polkadot.network/PolkaDotPaper.pdf.

[6] KWON J. Tendermint: Consensus without mining[EB/OL]. https://cdn.relayto.com/media/files/LPgoWO18TCeMIggJVakt_tendermint.pdf.

[7] BUCHMAN E, KWON J, MILOSEVIC Z. The latest gossip on BFT consensus[J]. arXiv: 1807.04938, 2018.

[8] TEUTSCH J, REITWIEßNER C. A scalable verification solution for blockchains[J]. arXiv: 1908.04756, 2019.

[9] POON J, DRYJA T. The bitcoin lightning network: Scalable off-chain instant payments[EB/OL]. https://www.bitcoinlightning.com/wp-content/uploads/2018/03/lightning-network-paper.pdf.

[10] POON J, BUTERIN V. Plasma: Scalable autonomous smart contracts[EB/OL]. https://plasma.io/plasma.pdf.

[11] HAAS A, ROSSBERG A, SCHUFF D L, et al. Bringing the web up to speed with WebAssembly[J]. ACM SIGPLAN Notices, 2017, 52(6): 185-200.

[12] GARAY J, KIAYIAS A, LEONARDOS N. The bitcoin backbone protocol: Analysis and applications[C]// Annual International Conference on the Theory and Applications of Cryptographic Techniques. Berlin: Springer, 2015: 281-310.

[13] NAKAMOTO S. Bitcoin: A Peer-to-Peer Electronic Cash System[EB/OL]. [2008-11-1]. https://bitcoin.org/bitcoin.pdf.

[14] WOOD G. Ethereum: A secure decentralised generalised transaction ledger[EB/OL]. https://ljk.imag.fr/membres/Jean-Guillaume.Dumas/Enseignements/ProjetsCrypto/Ethereum/ethereum-yellowpaper.pdf.

[15] BUTERIN V. A next-generation smart contract and decentralized application platform[EB/OL]. https://cryptorating.eu/whitepapers/Ethereum/Ethereum_white_paper.pdf.

[16] DOUCEUR J R. The sybil attack[C]// International workshop on peer-to-peer systems. Berlin: Springer, 2002: 251-260.

[17] ANDROULAKI E, BARGER A, BORTNIKOV V, et al. Hyperledger fabric: a distributed operating system for permissioned blockchains[C]// Proceedings of the Thirteenth EuroSys Conference. ACM, 2018: 30.

[18] KIAYIAS A, KONSTANTINOU I, RUSSELL A, et al. A Provably Secure Proof-of-Stake Blockchain Protocol[J]. IACR Cryptology ePrint Archive, 2016, 2016: 889.

[19] ROCKET T. Snowflake to avalanche: A novel metastable consensus protocol family for cryptocurrencies[EB/OL]. https://assets.ctfassets.net/xwo28v1qbyr0/CCMPhMqQM0kKMKyyiq0sE/d55ade6e3ea5294f3fdb913647630246/avalanche-consensus.pdf.

[20] GILAD Y, HEMO R, MICALI S, et al. Algorand: Scaling byzantine agreements for cryptocurrencies[C]// Proceedings of the 26th Symposium on Operating Systems Principles. ACM, 2017: 51-68.

[21] MAYMOUNKOV P, MAZIERES D. Kademlia: A peer-to-peer information system based on the xor metric[C]// International Workshop on Peer-to-Peer Systems. Berlin: Springer, 2002: 53-65.

[22] CHEN Z, ZHUO Y U, DUAN Z, et al. Inter-blockchain communication[J]. DEStech Transactions on Computer Science and Engineering, 2017(cst): 448-454.

[23] BAIRD L. The swirlds hashgraph consensus algorithm: Fair, fast, byzantine fault tolerance: Swirlds Tech Reports SWIRLDS-TR-2016-01[R]. 2016.

[24] BENTOV I, HUBÁCEK P, MORAN T, et al. Tortoise and Hares Consensus: the Meshcash Framework for Incentive-Compatible, Scalable Cryptocurrencies[J]. IACR Cryptology ePrint Archive, 2017, 2017: 300.

[25] WANG X, ZHAO H, ZHU J. GRPC: A communication cooperation mechanism in distributed systems[J]. ACM SIGOPS Operating Systems Review, 1993, 27(3): 75-86.

[26] VUKOL I. The quest for scalable blockchain fabric: Proof-of-work vs. BFT replication[C]// International workshop on open problems in network security. Cham: Springer, 2015: 112-125.

YE Shao-jie, born in 1995, Ph.D candidate, is a member of China Computer Federation. His main research interests include blockchain, distributed database and storage system.

SUN Jian-ling, born in 1964, Ph.D, professor, Ph. D supervisor, is a senior member of China Computer Federation. His main research interests include database system, machine learning, financial technology and software engineering.