一种支持高并发的多人链下支付方案

葛钟慧[1] 张 奕[2),3)] 龙 宇[2)] 刘 振[2)] 刘志强[2)] 谷大武[1),2)]

[1)](上海交通大学网络空间安全学院 上海 200240)

[2)](上海交通大学计算机科学与工程系 上海 200240)

[3)](上海观源信息科技有限公司 上海 200241)

摘 要 随着区块链技术研究与应用的快速发展,可扩展性瓶颈对于其在大规模应用场景下的主要制约作用逐渐凸显.作为解决区块链可扩展性问题的关键技术之一,支付通道技术将交易清算从链上的全网矿工认证转移到链下通道内的支付双方认证,从而实现了支付近乎即时确认;结合路由算法构成的支付通道网络,实现了任意两点间的链下支付,极大地提升了区块链的可扩展性.然而,目前的研究大多针对双人通道,涉及到多方的链下支付无法在单个通道内或通过单条支付路径完成,同时对于具有频繁、相互交易需求的多个人,两两间建立通道或通过路由来完成链下支付的所需链上开销与复杂性较高.现有的多人通道方案效率低下,不适用于高并发的链下支付场景,且不能支持跨通道支付,限制了链下支付的范围.基于此,本文在原有多人通道框架内改进了通道内状态更新机制,将通道状态依据支付串行更新变为并行更新,并引入支付有效期来减轻网络时延与高并发支付场景对支付有效性的影响,从而实现通道内支付处理效率的提升和对链下高并发支付场景的支持,此外在不关闭通道的前提下允许节点退出通道以提高通道可持续性;将多人链下支付从通道内推广至跨通道,具体地,在多人通道内引入条件支付与赎回支付,以支持安全的跨通道支付,同时将原双人通道网络中的基于图嵌入的贪心路由算法应用至多人通道网络中,以实现网络任意两点间的支付路径寻找.分析表明,多人通道网络具备可行性,并可实现安全的链下支付;我们对链下支付的路由情况进行模拟,实验结果显示,该方案链下支付成功率为 88%,且在静态场景下的路由开销相对双人通道网络降低了 28%,面对网络环境与设置变化路由性能更加稳定.由此,此方案具有高效、支持高并发、稳定的特性,满足实际应用中对链下支付的需求,具有良好的应用前景.

关键词 区块链;多人通道;链下支付;高并发;路由

中图法分类号 TP309 **DOI 号** 10.11897/SP.J.1016.2021.00132

A High-Concurrency Multi-Party Off-Chain Payment Scheme

GE Zhong-Hui[1] ZHANG Yi[2),3)] LONG Yu[2)] LIU Zhen[2)] Liu Zhi-Qiang[2)] GU Da-Wu[1),2)]

[1)](*School of Cyber Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

[2)](*Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai* 200240)

[3)](*Shanghai Viewsource Information Science & Technology Co., Ltd, Shanghai* 200241)

Abstract Blockchain has received great attention in recent years. Due to its inherent attribute of decentralization, the low throughput impedes its large-scale applications. As one of the most promising solutions to the scalability issue, Payment Channel Networks (PCNs) move the on-chain payments into off-chain payment channels, taking the underlying blockchain as the arbitration platform. While an on-chain payment needs to be verified by all mines in the blockchain network, an off-chain payment needs only to be approved by both parties within the channel, leading to nearly instant payment confirmation. In combination with routing algorithms, counterparties without a

收稿日期:2019-12-01;在线发布日期:2020-05-14.本课题得到国家自然科学基金(61672347,61672339,61872142,61932014,61572318)、"十三五"国家密码发展基金(MMJJ20170111)、上海市 2019 年度"科技创新行动计划"(19511101403,19511103900)、上海市闵行区中小企业技术创新计划(2018MH110)资助.**葛钟慧**,博士研究生,主要研究方向为区块链.E-mail:zhonghui.ge@sjtu.edu.cn.**张 奕**,硕士研究生,主要研究方向为区块链.**龙 宇**,博士,副教授,主要研究方向为区块链、信息安全与密码学.**刘 振**,博士,副教授,主要研究方向为应用密码学、区块链安全与隐私保护.**刘志强**,博士,副教授,主要研究方向为区块链、信息安全与密码学.**谷大武**(通信作者),博士,教授,主要研究领域为密码学与计算机安全.E-mail:dwgu@sjtu.edu.cn.

direct payment channel can accomplish payments in the network via a path of intermediaries, which broadens the scope of off-chain payments. However, current researches mainly focus on two-party payment channels, payments with multiple payers or payees cannot be accomplished within one channel or via a single payment path. At the same time, for a group of people who make payments mutually and frequently, launching off-chain payments either by establishing two-party channels between each pair or through intermediaries leads to high on-chain cost and payment complexity. As for Gnocchi, the current multi-party off-chain payment scheme, there is a lack of efficiency and fails to support high-concurrency in-channel payments. Besides, there is no support for cross-channel payments, which limits the range of off-chain payments. In this paper, we improved the operation mechanism of Gnocchi to achieve the enhancement of in-channel payment processing efficiency and the support of high-concurrency off-chain payments. Similarly, we continued the introduction of supervisor within the multi-party channel as the proposer of updating channel state in each round. Differently, we transferred the serial channel state update into the parallel update and regard payments the supervisor receives in a certain time period as in one round. To reduce the impact of network latency and high-concurrency payment scenarios on payment effectiveness, the round number contained in one payment is regarded as valid if it is in a certain interval of the current round. Besides, the case of node departs and withdraw his value from the channel with no need to close the channel is considered to promote channel sustainability. Moreover, we extended the off-chain payments from in-channel to cross-channel. Specifically, the conditional payment format, whose value will be transferred to the payee when he fulfills its condition, is introduced within the channel to ensure the secure value transfer along the cross-channel path. Accordingly, the redemption payment format implements the condition fulfillment. The greedy embedding-based routing algorithm is deployed to achieve efficient path finding between any pair of nodes in the multi-party off-chain network. We analyzed the feasibility and security of the network, and evaluated the performance of the routing algorithm using real-world data. As the results show, it achieves the payments success ratio of 88% and reduces routing overhead by 28% compared to the two-party PCNs in the static scenario. Facing changes from network scenario and network settings, it achieves a comparably steadier routing performance. With the properties of efficiency, high-concurrency, and stability, it is concluded that the multi-party off-chain payment scheme is deployable in real-world applications.

Keywords　blockchain; multi-party payment channel; off-chain payment; high-concurrency; routing

1　引　言

自 2008 年比特币[1]问世以来,其底层区块链技术受到来自科研界与产业界的广泛关注.以太坊[2]所提出的智能合约进一步拓宽了区块链技术的应用边界,跳出了转账交易与简单业务脚本的框架.然而,由于区块链固有的分布式属性,即每笔交易都要经过全网矿工节点的确认,导致区块链的可扩展性较低.理论上比特币吞吐率为 7 TPS,以太坊为 15 TPS,远低于传统的中心化系统如 Visa,可支持每秒上千笔交易[3],这极大地限制了区块链在大规模场景下的应用,也使得区块链可扩展性成为亟待解决的问题.

作为区块链可扩展性问题[4]的主流解决方案[5-8]之一,支付通道网络(Payment Channel Networks, PCNs)[5,8]内有两种支付形式,通道内支付和跨通道支付.通道内支付指具有高频、小额交易需求的两个节点在链上预存金额至链下支付通道中,并将后续交易放在通道内执行,在通道关闭或双方对通道内金额分配产生纠纷时,将通道内最新状态提交至链上判决.跨通道支付针对链下支付通道网络中无直

接通道建立的交易方，通过路由算法寻找一条连通双方、资金充足的路径以完成支付，从而拓宽了链下支付的范围.通过基于密码学的交互协议与惩罚机制，并依靠区块链作为第三方仲裁平台，链下支付的安全与效率得以保障.由于交易在链下执行、只需通道内双方确认而无需上链，可实现即时确认.由于交易信息与在某一时刻通道内金额分配仅为通道双方所知，可实现交易隐私性保护.与其它方案相比，支付通道网络不涉及链本身的属性，从而具有更强的兼容性.

然而，目前支付通道网络研究大多关注双人支付通道，仅适用于两方，对于涉及多个参与方的链下支付，只能将此支付划分为多笔子支付，在多个通道内或通过多条支付路径完成，此时支付的复杂性提高.同时对于具有频繁交易需求的 l 个人，满足两两之间的交易需求需要建立 $O(l^2)$ 个通道或通过路由实现，提高了所需的链上开销与交易复杂性.已有的多人通道方案 Gnocchi[9] 将双人通道推广至多人通道，但是存在两点显著问题.首先，通道内通信开销大，且支付处理效率低，不能应用于通道内高并发支付的场景下.其次，此方案缺乏对跨通道支付应用场景的考虑，支付被局限在通道内部，限制了链下支付的范围.

针对以上问题，本文提出了一种支持高并发的多人链下支付方案，主要贡献有：

(1)在 Gnocchi 的基础上改进了多人通道内部运行机制，提高了通道内支付处理效率，实现了通道内状态一致性的正确、快速达成，满足对高并发支付的需求；

(2)将链下支付的范围从通道内扩展到跨通道.在多人通道内部引入条件支付与赎回支付，实现了安全的跨多人通道支付；设计了基于多人通道网络的路由算法，实现了分布式环境下网络中任意节点间的支付路径寻找；

(3)分析证明了多人通道网络的可行性与安全性，并通过实验验证网络路由性能.实验结果表明，多人通道网络实现了 88%的链下支付成功率，相比于双人通道网络实现了更低的路由开销与更稳定的路由性能，满足实际应用需求.

本文第 2 节对支付通道网络的相关工作，以及本文所需的预备知识进行介绍；第 3 节描述多人通道内支付方案；第 4 节阐释跨多人通道支付方案；第 5 节对多人通道网络的可行性与安全性进行分析；第 6 节对多人通道网络的路由性能进行实验验证，并对结果进行分析；第 7 节总结本文的工作，并展望下一步的研究方向.

2 相关工作与预备知识

2.1 相关工作

对支付通道网络的研究主要分为单通道和跨通道两个方面.在单个通道方面，针对通道中支付的可链接性以及通道建立、关闭过程中因信息上链而导致的隐私泄漏问题，Green 等人[10] 提出了具有隐私保护性质的支付通道方案.Zhang 等人[11] 基于 Zerocash 构建了链下支付通道.Pan 等人[9] 提出了一种链下多人通道方案 Gnocchi.多人建立一个通道并预存金额至其中，引入监管节点来处理通道内支付并广播更新的各节点余额.通道的安全性通过监管节点预存的保证金以及相应的惩罚机制来保证，具体细节将在第 3 节介绍.

在跨通道支付方面，高效的路由算法是其核心.根据节点掌握资源的不同，我们将路由算法划分为全局路由与局部路由两类.全局路由中，各节点维持链下支付网络的拓扑图.对此，Prihodko 等人[12] 提出了 Flare，通过将到相邻节点与信标节点的路径保存在路由表中，并结合交易双方的路由表找到公共节点来确定交易路径，通过问询路径上各节点的可用金额与支付费用来判断路径的可用性.Malavolta 等人[13] 提出的 SilentWhispers 采用 Landmark 路由算法[14]，选取网络中连通度高的节点作为 landmark，通过结合交易双方到 landmark 的路径生成总交易路径，通过安全多方计算确定路径上可转移金额，并且保护通道内金额与交易双方的隐私.局部路由中，节点仅掌握自身所在通道的信息.对此，Roos 等人[15] 提出了 SpeedyMurmurs，采用基于图嵌入的贪心路由算法[16] 实现分布式环境下的节点间路径寻找.

除路由算法外，Malavolta 等人[17] 针对支付通道网络的隐私性问题提出了协议 Fulgor，针对并发性问题提出了协议 Rayo，同时刻画了网络隐私性与并发性之间的权衡.针对通道内金额分布随着链下支付进行而失衡的问题，Khalil 等人[18] 提出了链下均衡协议，允许网络中节点依据其偏好来均衡通道内金额.Miller 等人[19] 改变了跨通道支付的完成确认方式，沿路各节点可通过问询全局合约以获取支付完成情况，从而降低所需时间成本.此外 Malavolta 等人[20] 还针对跨通道支付时存在的虫洞攻击，提出

了匿名多跳锁的密码学原语.

2.2　双人通道网络

我们以基于脚本的闪电网络[5]和基于合约的Sprites[19]为例介绍双人通道网络中的支付过程.

2.2.1　通道内支付

双人通道运行周期包含通道建立,通道状态更新和通道关闭三个过程.

在闪电网络中,两个节点通过预存金额至一个共同控制的多签名地址来建立链下支付通道.节点在通道内通过RSMC(Recoverable Sequence Maturity Contract)实现双向支付,通道内状态的更新需要来自双方的认证,同时作废之前的通道状态.当双方想关闭通道时,提交共同认可的最终通道内金额分配至链上.当一方提交通道状态至链上以取回通道内金额时,RSMC的时间锁使得通道关闭发起方延时取回其金额,保证另一方能在延时时间内发现可能存在的作恶行为并提交证据至链上,启动惩罚机制没收作恶方的金额,进而保证链下支付的安全性.

在Sprites中,两节点预存金额至智能合约来建立链下支付通道.通道内每笔支付带有当前通道轮数r,随着支付的进行轮数逐一递增.链上对于通道状态的判断基于所提交通道状态轮数的大小,即将轮数更高的通道状态视为通道最新状态.相对于闪电网络,Sprites中节点仅需存储最新通道状态,降低了存储开销.

2.2.2　跨通道内支付

将双人通道网络抽象为无向有权图$G=(V,E)$,V为网络中节点集,$E\subset V\times V$为支付通道集.用(u,v)来表示节点u与v之间的通道,$w_{u,v}$,$w_{v,u}$表示在通道(u,v)内节点u和v所拥有的金额,$N(u)$表示节点u的相邻节点,即与其建立支付通道的节点.网络路由算法即为发生在节点u和w间的支付寻找一条路径$p=\{n_i\mid i\in[1,k],(n_i,n_{i+1})\in E,n_1=u,n_{k+1}=w\}$.而路径$p$可实现的最大支付金额为$\min\{w_{n_i,n_{i+1}}\},i\in[1,k]$.

如图1所示,付款方Alice和收款方Emma没有直接通道,但是可通过路径p_1:Alice→Bob→Emma和路径p_2:Alice→Carl→David→Emma完成支付.通过路径p_1和p_2可实现的最大支付金额分别为$\min\{6,7\}=6$和$\min\{10,13,5\}=5$.

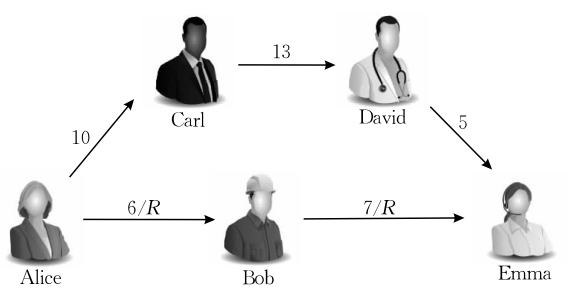


图1　双人通道网络路由示意图

为保证跨通道支付的安全性,闪电网络采用HTLC(Hashed TimeLock Contract)协议锁住支付金额,只有当收款方在规定时间内提供满足条件的哈希原像时,支付才会生效.如在路径p_1上进行跨通道支付时,Alice和Bob之间,Bob和Emma之间均建立支付,并以收款方Emma生成的哈希原像值R作为支付条件.支付建立后,Emma向Bob展示原像值以取回支付金额,Bob以此相同原像值从Alice处完成支付.跨通道支付的安全性通过哈希的抗原像攻击特性得以保证.而对于构建在图灵完备的区块链平台(如以太坊)上的支付通道,可以利用智能合约设置更为灵活的支付条件.

2.3　基于图嵌入的贪心路由算法

由于通道内支付在链下进行并仅需通道内双方的确认,即使在网络结构与各节点预存金额已知的情况下,金额在通道内的即时分配仅为通道双方所知,因此寻找一条资金满足条件的路径不是一个简单问题.

基于图嵌入的贪心路由算法是将网络结构图映射到特定的坐标空间中,基于节点的坐标表示以及距离计算公式,实现任意两节点间的路径寻找.此路由方法适用于点与点之间连接受限的网络,Speedy-Murmurs将其用于双人通道网络中的路由,在支付成功率、路径长度、路由开销等性能方面均有相对优越的性能.

节点坐标标记基于网络中的生成树结构,根节点赋坐标值为空矩阵,子节点的坐标赋值为父节点坐标与子节点标识的结合.对于坐标x_1和x_2,长度分别为$|x_1|$和$|x_2|$,相同前缀长度为$cpl(x_1,x_2)$,其间的距离计算如下:

$$\delta(x_1,x_2)=|x_1|+|x_2|-2cpl(x_1,x_2)\quad(1)$$

在分布式环境下,节点收到路由信息时,将其转发至符合支付条件的、距离目标节点更近的相邻节点,从而保证网络中任意两点间距离递减、资金充足的路径的寻找.

2.4　区块链平台

本文工作基于图灵完备的区块链平台.我们假定底层区块链为可信第三方平台,任意节点发送至区块链网络的有效信息,均可在Δ时间内上链,任何敌手不能阻碍节点对交易信息的发送与矿工对交

易信息的验证.同时,敌手不能对节点获取区块链上信息造成干扰.

3 多人通道内支付

一个多人通道的运行周期包括通道建立、通道状态更新与节点退出.本文延续了 Gnocchi 方案中在通道内部对监管节点的引入,监管节点通过处理通道内支付而获取手续费,费率与支付方式可由各方在通道建立时自行商定,我们在此将其忽略以介绍通道基本框架.为保证通道内各节点的资金安全,需设置作恶判决功能对监管节点的行为进行监督.多人通道运行机制如图 2 所示,每个通道均对应于链上的一个智能合约,通道建立、作恶判决与节点退出均由节点触发合约功能完成,通道状态更新由各节点在链下完成.

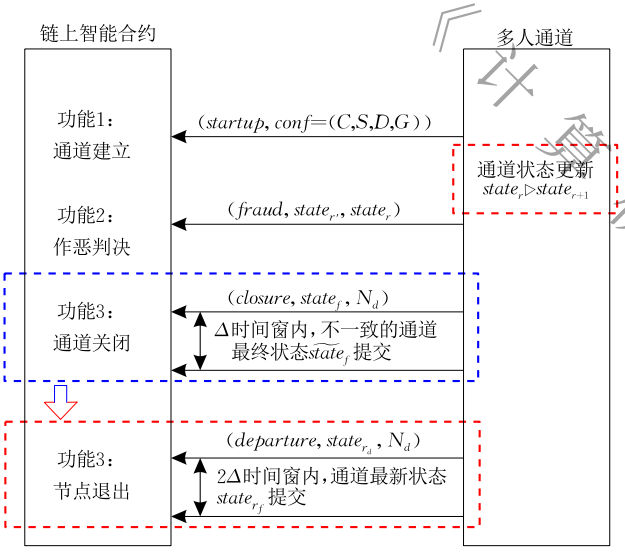


图 2 多人通道运行机制图

图 2 中的虚线框标示本文在 Gnocchi 方案基础上的改进部分,主要有两处.其一为通道状态更新过程.Gnocchi 中的状态更新过程描述与缺陷分析,以及具体改进方案将在第 3.2 节详细描述.其二为通道关闭过程.由于 Gnocchi 中通道内任一节点可发起通道关闭请求,降低了多人通道的可持续应用性.由此我们将其改进为节点退出过程,非监管节点可退出通道并提现通道内余额,监管节点退出时通道将关闭,此时需由链上合约对节点退出通道后的状态进行更新与认证,具体细节将在第 3.3 节描述.此外,通道建立与作恶判决的核心思想与原方案相同,我们分别在第 3.1 节与第 3.3 节做统一叙述.

3.1 通道建立

通道中的成员,以集合 C 表示,协商各方在此通道内的预存金额,并确定通道的监管节点 S.为保证正确履职,监管节点需额外缴纳保证金 G,并且满足关系式,

$$G \geq |C| \times \sum_{N_i \in C} D[N_i],$$

其中,$D[N_i]$表示节点 N_i的预存金额.

在监管节点被判决作恶时,G 会被罚没并分发给通道内的剩余节点.因此,监管节点作恶代价至少是其作恶收益的$|C|$倍,且其余各节点所收获的罚金一定不少于其在通道内由于监管节点作恶所带来的损失.由此激励节点积极监督这些监管节点,从而约束监管节点的行为,保护通道内交易的安全性.

通道建立过程如功能 1 所示,监管节点将各节点联合生成的通道建立请求与各节点对此请求的签名发往链上合约,由矿工对此请求的有效性进行验证.

功能 1. 通道建立.

输入:通道建立请求$(startup, conf=(C,S,D,G))$

输出:通道建立/null

1. 若 C 中节点对建立请求的签名错误,则抛弃此请求
2. 若 C 中节点在链上账户的金额小于其在 D 中的预存金额,则抛弃此请求
3. 计算保证金的最小值 $G_{\min} = |C| \times \sum_{N_i \in C} D[N_i]$.若 $G < G_{\min}$,则抛弃此请求
4. 若监管节点 S 在链上账户的金额小于其在 D 中预存金额与保证金之和,则抛弃此请求
5. 从节点的链上账户中扣除相应的预存金额,从监管节点的链上账户中扣除其预存金额与保证金
6. 初始化链上认证的通道最新状态 $state_{\text{best}}$ 为起始状态 $state_0$,其中的通道最新轮次 r_{best} 为 0;初始化通道总金额 $CV = \sum_{N_i \in C} D[N_i]$
7. 通知各节点通道建立

至此,一个可用的多人通道建立完毕,节点可在该通道内执行多次链下支付.

3.2 通道状态更新

在此节中我们介绍通道状态更新过程.我们假定通道内各节点间通过加密可认证信道通信,节点发送消息时均附带签名,且签名可由其他方验证.

3.2.1 Gnocchi 通道状态更新

在 Gnocchi 中,第 r 轮通道状态表示为

$$state_r = (r, D_r, P_r),$$

其中,D_r表示 r 轮后各节点通道内余额,P_r为第 r 轮通道内支付.自起始状态 $state_0=(0,D,\varnothing)$起,每轮通道状态更新由支付触发.以付款方 pr 与收款方 pe 发起一笔支付金额为 a 的支付为例,此笔支付被确认的过程如图 3 所示.

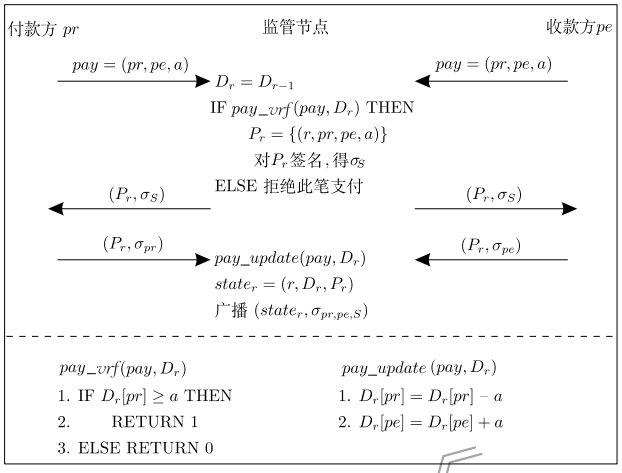


图 3　Gnocchi 通道状态更新图

支付被包含进通道状态时带有来自支付双方与监管节点三方的签名,作为通道状态更新的证明,以供其余节点验证.

3.2.2　Gnocchi 方案分析

Gnocchi 方案具有如下缺点:

(1)每笔支付需经历监管节点与支付双方共 6 次交互,且多次交互只为确定支付所处轮数,通信开销与冗余大;

(2)通道状态依据通道内支付串行更新,每轮更新仅包含一笔支付,每笔支付的确认均在通道内上一笔支付完成的条件下进行,支付处理效率低,限制了在高并发场景下的应用;

(3)缺乏对网络时延与支付方恶意延长支付确认情况的考量.在此种情况下,通道状态更新陷入停滞,造成通道堵塞.

3.2.3　通道状态更新方案改进

针对上述缺点,我们改进了通道状态更新方案,具体为

(1)支付方面,每笔支付所处的轮数由支付双方依据所收到的最新通道轮数确定,降低通信次数至 1 次;

(2)通道状态更新方面改进为并行更新.监管节点在每轮状态更新时开启 θ 的时间窗,将在此时间段内收到的支付视为此轮支付,并依据每笔有效支付变更通道状态.在此轮结束时打包通道状态、有效支付集与此时的时间戳 T,广播至通道内节点;

(3)考虑在网络时延与高并发场景下,一笔支付未能得到及时处理而导致支付轮数失效的情况,引入支付有效期 η,即当前轮数为 r 时,支付中包含的轮数 r_p 处于 $[r,r+\eta]$ 即视为有效;为抵御重放攻击,在通道状态中引入 I_r 记录各节点支付次数,同时每笔支付需包含付款方的更新支付次数 idx.各节点支付次数在 I_0 中初始化为 0.

通道状态更新自起始状态 $state_0=(0,D,\varnothing,I_0,T_0)$始,过程如算法 1 所示.

算法 1.　通道状态更新.

输入:第 $r-1$ 轮通道状态 $state_{r-1}$

输出:第 r 轮通道状态 $state_r$

1. $D_r=D_{r-1}, I_r=I_{r-1}, P_r=\varnothing$
　　//初始化第 r 轮通道状态信息
2. WHILE 在 θ 时间段内
3.　UPON 收到支付 $pay=(r_p,pr,pe,a,idx,\sigma_{pr,pe})$
4.　IF $pay\_vrf\_2(pay,D_r,I_r)$ THEN
5.　　$pay\_update\_2(pay,D_r,I_r,P_r)$
6. ENDWHILE
7. RETURN $state_r=(r,D_r,I_r,P_r,T_r)$

函数 1.　$pay\_vrf\_2(pay,D_r,I_r)$.

1. IF $(pay\_vrf(pay))\wedge(r_p\in[r,r+\eta])\wedge$
　　$(idx==I_r[pr]+1)$ THEN
2.　RETURN 1
3. ELSE RETURN 0

函数 2.　$pay\_update\_2(pay,D_r,I_r,P_r)$.

1. $pay\_update(pay,D_r,I_r)$
2. $I_r[pr]=I_r[pr]+1$
3. $P_r=P_r\cup\{pay\}$

通过以上改进,多人通道降低了通道内通信开销,提高了支付处理效率,可实现高并发场景下通道状态的快速更新.

3.2.4　通道状态验证

若两个通道状态处于同一轮次却不相同,我们称这两个状态是冲突的.对于相邻的通道状态 $state_{r-1}$ 与 $state_r$,若可依据 $state_{r-1}$ 中的节点余额与 $state_r$ 中包含的支付集得到 $state_r$ 中,则称状态是可达的.

收到监管节点广播的通道状态 $state_r$,通道内节点基于链上最新状态 $state_{best}$ 和状态 $state_{r-1}$ 验证其正确性,即验证 $state_r$ 是否与 $state_{best}$ 冲突,同时验证状态是否可达.若节点只存有状态 $state_{r'}$,$r'<r-1$,则可向监管节点问询 $r'+1$ 轮至 $r-1$ 轮的通道状态,并验证每对相邻状态.若验证通过,则

同样称其为可达的,将其记作

$$state_{r'} \triangleright state_r.$$

为方便后文描述,我们不再区分 r' 与 r 之间的大小关系,以此符号表示任意两个状态间的可达关系.若通道状态不可达,节点将不可达的通道状态提交至链上作恶判决功能,对监管节点进行惩罚.

3.3 通道监管节点作恶判决

监管节点对通道内状态更新行为受到通道内其它节点的监督.当节点接收到来自监管节点的最新通道状态并发现其作恶行为时,即提交相关状态信息至链上合约做判决,判决过程如功能 2 所示.

功能 2. 作恶判决.

输入:作恶判决请求($fraud$,$state_{r'}$,$state_r$)

输出:通道关闭/null

1. IF $\min\{r',r\}<r_{\text{best}}$ THEN 抛弃此请求
2. IF $state_{r'}$,$state_r$,$state_{\text{best}}$之间存在冲突 THEN
3. RETURN $fraud\_supervisor()$
4. IF r'与 r 不相邻,且状态中的时间戳与此功能调用的时间差不超过 Δ THEN
5. 通知监管节点提交处于之间轮次的通道状态
6. 开启 Δ 的时间窗,若在此期间内未有完整状态提交:
7. RETURN $fraud\_supervisor()$
8. IF !($state_{r'} \triangleright state_r$) THEN //通道状态不可达
9. RETURN $fraud\_supervisor()$
10. RETURN null

函数 3. $fraud\_supervisor()$.

1. $D[N_j]=(CV+G)/(|C|-1)$,$N_j \in C\backslash S$

//将通道内金额和保证金分配给除监管节点外的剩余节点

2. $D[S]=0$ //将监管节点金额置 0
3. 通道关闭,按照 D 退还各节点余额

若节点提交的状态未处在相邻轮次,即针对监管节点未能递增轮数 r 来更新通道状态的情况,此时链上开启 Δ 的时间窗,在此期间内监管节点需提交中间轮次的通道状态至链上,若未能完成提交则判断为监管节点作恶.为防止节点恶意发送已验证过的、不相邻的通道状态至链上,因此需验证通道状态中的时间戳与合约调用时间的时间差是否在 Δ 范围内,即此通道状态是否是在最近生成的,从而要求节点在发现监管节点作恶的情况后及时提交状态至链上.时间差的范围设定为 Δ,对应于最坏情况下此状态通过节点退出功能在链上获得,具体在下一节中介绍.

3.4 通道内节点退出

通道内节点可退出通道并提现通道内金额至链上账户,此时由链上来更新节点退出后的最新通道状态.监管节点退出时还将取回其在通道内的保证金,此外通道将被关闭,通道内金额返还至各节点.在金额提现之前,退出节点需接受通道内其它节点的监督.若被判决为作恶,则其金额被没收并分配至通道内剩余节点.

收到退出请求,链上节点退出功能向通道内各节点发送此请求,并开启时长为 Δ 的等待窗口收集通道内最新状态,以此来判断退出节点是否作恶,并在链上更新通道状态.时间窗的时长保证了节点在收到请求后,提交的状态 $state_{r_p}$ 可在窗口关闭前上链.此时节点若发现其中存在冲突或者不可达的通道状态,则可将其提交至链上作恶检测合约,节点退出功能设定 Δ 的等待窗口等待作恶判决的运行结果.在两个时间窗关闭后,合约依据收集到的状态进行通道状态更新.

对于监管节点而言,离开时需提交通道最新状态至链上.因此若收集到最新通道状态处于更高轮次,则可判定为作恶.同时,若收到非监管节点的退出请求,监管节点需将目前最新的、与退出请求不同的通道状态提交至链上,并暂停目前对于通道状态的更新,直至收到链上发送的最新状态,从而避免产生与链上最新状态相冲突的状态进而被判断为作恶.对于非监管节点而言,离开时需提交包含其最新余额的通道状态至链上.因此若此节点在最新通道状态与提交的通道状态中的余额不同,则可判定为作恶.验证过程如功能 3 所示.

功能 3. 节点退出.

输入:节点退出请求($departure$,$state_{r_d}$,N_d)

输出:通道关闭/通道最新状态 $state_{\text{best}}$

1. IF $r_d<r_{\text{best}}$ THEN 抛弃此请求
2. 将节点退出请求发送至通道内其它节点,在此请求完成前抛弃其余节点的退出请求
3. 开启 Δ 的时间窗,将此段时间内收到的状态 $state_{r_p}$ 加入列表 States 中
4. 开启 Δ 的时间窗,等待可能运行的作恶判决的结果
5. 将 $state'_{\text{best}}$ 赋值为 $States$ 中具有最高轮次的状态,并分解为 $\{r'_{\text{best}},D'_{\text{best}},I'_{\text{best}},P'_{\text{best}},T'_{\text{best}}\}$
6. IF ($N_d==S$) $\wedge$ ($r_d<r'_{\text{best}}$) THEN //监管节点提交过期状态
7. RETURN $fraud\_supervisor()$
8. IF ($N_d \neq S$) $\wedge$ ($D_{r_d}[N_d] \neq D_{r'_{\text{best}}}[N_d]$) THEN //非监管节点提交过期余额信息
9. RETURN $departure\_node(0)$

10. IF $N_d==S$ THEN　　//监管节点诚实
11.　按 $state_{r_d}$ 中集合 D_{r_d} 退还节点余额,退还保证金
12. ELSE RETURN $departure\_node(1)$
　　//非监管节点诚实

函数 4.　$departure\_node(b)$.

1. $C=C\backslash N_d$　　//移除退出节点
2. $D_{\text{best}}=state\_update(b,D'_{\text{best}},N_d)$
　　//依据退出节点诚实与否更新通道状态
3. $r_{\text{best}}=r'_{\text{best}}+1,P_{\text{best}}=\varnothing,T_{\text{best}}=T_{\text{now}}$
4. $I_{\text{best}}[N_j]=I'_{\text{best}}[N_j],N_j\in C$
5. $state_{\text{best}}=(r_{\text{best}},D_{\text{best}},I_{\text{best}},P_{\text{best}},T_{\text{best}})$ //生成通道状态
6. 向各节点发送最新通道状态 $state_{\text{best}}$

函数 5.　$state\_update(b,D,N_d)$.

1. IF $b==0$ THEN　　//退出节点作恶
2.　$amount=D[N_d]/|C|$
3.　$D'[N_i]=D[N_i]+amount,\ N_i\in C$
　　//没收其余额,并分配至剩余节点
4. ELSE $D'[N_i]=D[N_i],N_i\in C$
5.　$CV=CV-D[N_d]$
6.　退还 $D[N_d]$至节点 N_d
7. RETURN D'

非监管节点在退出前需要确保自己所有已签名的支付均已被包含在状态信息中或已失效,避免被判定为作恶.非监管节点退出后,监管节点在收到的链上最新状态的基础上继续链下通道更新.

4　跨多人通道支付

跨多人通道支付应用于付款方与收款方在不同通道内的支付场景.具体而言,通过同时存在于不同通道的共同节点作为支付中转点来完成支付.由此,链下网络中支付不再局限于通道内,支付范围得以扩展.

为保证支付金额在通道间安全转移,本文定义了通道内条件支付与赎回支付;为实现节点跨通道支付的路径寻找,本文提出了基于多人通道网络的高效、灵活路由算法.在支付路径确定后,沿路相邻节点可顺序建立条件支付并逆序发起赎回支付.

4.1　多人通道内支付

条件支付为支付金额被某种条件和规定时间限锁住的支付.只有当收款方在规定时间内满足所包含的条件时,才能赎回被锁住的支付金额.因此一个完整的支付过程包括条件支付与赎回支付.

4.1.1　条件支付验证与更新

条件支付的格式为

$$conpay=(r_p,pr,pe,a,idx,h,BH,\sigma_{pr},\sigma_{pe}),$$

其中,哈希值 h 作为支付条件,底层区块链的区块高度 BH 作为时间限,表示收款方需在此时限之前赎回金额.

监管节点对收到的条件支付进行验证,包括对支付的有效性验证,以及对支付有效时限的验证,如函数 6 所示,其中 BH_c 表示当前的区块链高度.

函数 6.　$conpay\_vrf(conpay,D_r,I_r)$.

1. IF $(pay\_vrf\_2(conpay))\wedge(BH\geq BH_c)$ THEN
2.　RETURN 1
3. ELSE RETURN 0

由于条件支付的赎回通常不在单轮内完成,因此在通道状态 $state_r$ 中增加 C_r 字段,记录到第 r 轮为止有效的、未被赎回的条件支付,用以验证后续赎回支付.针对有效条件支付的通道状态更新如函数 7 所示.

函数 7.　$conpay\_update(conpay,D_r,I_r,P_r,C_r)$.

1. $D_r[pr]=D_r[pr]-a$
2. $I_r[pr]=I_r[pr]+1$
3. $P_r=P_r\cup\{conpay\}$
4. $C_r=C_r\cup\{conpay\}$

在每轮通道状态更新结束时,监管节点需验证 C_r 中各条件支付的有效性,如函数 8 所示.

函数 8.　$conpay\_return(C_r)$.

1. FOR each $conpay$ in C_r
2.　IF $BH<BH_c$ THEN　　//条件支付失效
3.　　$D_r[pr]=D_r[pr]+a$　　//返还支付金额
4.　　$C_r=C_r\backslash\{conpay\}$　　//移除失效支付
5. ENDFOR

4.1.2　赎回支付验证与更新

赎回支付的格式为

$$redemp=(r_p,conpay,img,idx,\sigma_{pe}),$$

其中,$conpay$ 为此赎回支付所针对的条件支付,img 为支付条件的哈希原像.

监管节点验证收到的赎回支付,包括此笔支付是否有效,以及其针对的条件支付是否仍然有效且未被赎回,如函数 9 所示,其中 $hash$ 表示抗前像攻击的哈希函数.若赎回支付有效,则支付金额将转移至收款方,通道状态更新如函数 10 所示.

函数 9.　$redemp\_vrf(redemp,D_r,I_r,P_r,C_r)$.

1. IF $(r_p\in[r,r+\eta])\wedge(conpay\in C_r)\wedge(hash(img)==h)\wedge(idx==I_r[pe]+1)$ THEN
2.　RETURN 1
3. ELSE RETURN 0

函数 10. $redemp\_update(redemp, D_r, I_r, P_r, C_r)$.

1. $D_r[pe]=D_r[pe]+a$ //收款方赎回金额
2. $I_r[pe]=I_r[pe]+1$
3. $P_r=P_r\cup\{redemp\}$
4. $C_r=C_r\backslash\{conpay\}$ //移除完成的条件支付

4.1.3 通道内节点退出

由于条件支付的存在,为保护收款方利益免因节点退出而受损,链上节点退出功能需做如下检测:

(1)当监管节点离开时,在验证其诚实性的基础上,若其提交的通道状态信息包含有效的、尚未赎回的条件支付,即 $C_r\neq\varnothing$,退出请求被驳回;

(2)当非监管节点离开时,在验证其诚实性的基础上,若其作为收款方存在于有效的、未赎回的条件支付,且同时提交相应的赎回支付并验证通过,则将此条件支付金额退回至此节点账户;若其作为收款方存在,且未提交相应的赎回支付,或者其作为付款方存在,则将金额转移至此条件支付的另一方.若节点被判决为作恶,则转移此金额至条件支付的另一方.根据以上情形,合约生成对应的最新通道状态,并发送至各节点.

由于赎回支付需要得到监管节点的确认,若经过多次发送尝试赎回支付仍未被包含在通道状态中,节点可选择退出并提交赎回支付,从而免遭因监管节点失联或恶意操作而导致利益受损.

4.2 多人通道网络模型

多人通道网络由多人通道构成.跨通道支付的实现依赖于支付双方间支付路径的寻找.为最小化节点在路由时所需的资源,我们采用目前局部路由算法中基于图嵌入的方式,将多人通道映射到坐标空间中,再对通道中的节点进行坐标赋值,即将节点坐标的建立分为两个阶段,通道坐标建立与节点坐标建立.

4.2.1 通道坐标建立

多人通道网络图嵌入. 当两个多人通道具有共同节点时,通道内的其它节点可以通过此共同节点作为中转点完成支付.此时,两个通道是连通的,称为相邻通道.

将多人通道网络抽象为无向图 $G=(V,E)$,其中 V 代表多人通道集,$E\subset V\times V$ 代表相邻通道间的边集,则 $NPC(pc_1)=\{pc_2\mid pc_2\in V,(pc_1,pc_2)\in E\}$ 为通道 pc_1 的相邻通道集.与相邻通道间的最大可转移金额即为通道共同节点在此通道内的金额之和.

分布式环境下通道坐标建立. 由于多人通道在建立阶段上链,各通道内的成员列表及预存的初始金额在网络内是公开的.作为通道监管节点,需确定其相邻通道集.

在分布式环境下的通道坐标建立过程中,选取某个公认的通道作为根通道,并设定其坐标为空矩阵.从根通道开始,每个通道的监管节点生成通道坐标,并将其广播至相邻通道的监管节点.收到相邻通道的通道坐标信息,监管节点将其存储,并从中选取坐标长度最短的通道作为父通道,通过结合父通道坐标与自身独特标识生成通道坐标.

当因节点退出而导致与父通道不再连通,或者通道间可转移金额匮乏时,子通道可选择存储的其它相邻通道作为父通道并重新生成通道坐标,从而保证在动态网络环境下的路由准确性.

4.2.2 节点坐标建立

通道坐标确立后,监管节点将其广播至通道内,各节点通过结合通道坐标与自身在通道内的独特标识生成节点坐标.网络中节点坐标具有以下性质:

(1)节点 u 处于 l 个通道内,则具有 l 个坐标,用 $C(u)=\{uc_1,\cdots,uc_l\}$ 表示其坐标集,用 $NN(u)$ 表示在此 l 个通道内的剩余节点集,称为 u 的相邻节点.若与相邻节点 w 可分别通过坐标 u_c 与 w_c 进行直接支付,即两坐标在同一个通道中,则称 u_c 为与 w 或 w_c 的相邻坐标;

(2)处于同一个通道内的坐标,其长度与前缀相同,且前缀即为通道坐标;

(3)坐标的长度反映其与根通道的距离,即对于坐标 x,可通过一条长为 $|x|-1$ 的路径到达根通道内节点.

4.3 多人通道网络路由

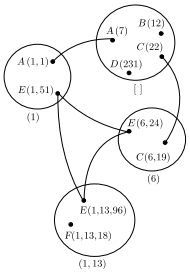
在分布式环境下,节点需将路由信息发送到距离目标地址更近的相邻节点,而与节点的距离判断基于坐标距离计算.因此,网络中每个节点需维持一个路由表,表中记录此节点的相邻节点与它们的坐标集.同时,节点与坐标之间的距离计算基于坐标间的距离计算.在此基础上,路由算法实现了在分布式环境下网络节点间一条资金充足的连通路径寻找.与双人通道网络中对路由算法的研究[14]相同,我们忽略跨通道支付过程中,中间节点对支付费用的收取.

4.3.1 路由表

各节点在确定坐标后将其坐标集广播至相邻节点,同时储存收到的节点坐标信息至路由表中,并依

据网络结构动态变化更新坐标以及路由表.

图 4 为多人通道网络与其中节点 A 维持的路由表示意图.在左图所示的网络结构与各通道节点坐标标识下,节点 A 处于通道 $pc_1=[\,]$与 $pc_2=(1)$ 中,相邻节点有 B,C,D,E,故在路由表中记录每个相邻节点的全部坐标.



相邻节点	坐标集
B	(12)
C	(22),(6,19)
D	(231)
E	(1,51),(6,24),(1,13,96)

图 4　左图为多人通道网络与节点坐标示意图;右表为网络示意图中节点 A 的路由表

4.3.2　节点与坐标距离计算

为寻找一条离目标坐标距离递减的路径,收到路由信息的节点需计算相邻节点与目标坐标之间的距离.对于节点 u 与坐标集 $C(u)=\{uc_1,\cdots,uc_l\}$,定义其与坐标 x 间的距离为

$$\tilde{\delta}(u,x)=\min_{uc_i\in C(u)}\delta(uc_i,x)\qquad(2)$$

其中,δ 为坐标间距离计算公式(1),节点和坐标间的距离定义为节点各坐标与坐标距离的最小值.

4.3.3　路由算法

定义路由信息的格式为(付款地址,目标地址,支付金额).在分布式网络下,对于节点 u 发起一笔支付或是收到路由信息,需进行如下四个阶段的路由,其中 $C(u)=\{uc_1,\cdots,uc_l\}$ 为其坐标集,同时初始化路由失败节点集合 FF 为空集.

阶段 1.　发起支付/选择收款坐标.

(1)若节点 u 为支付付款方,则与收款方协商支付金额 a 与收款地址 dst,进入阶段 2;

(2)若节点 u 从节点 z 处收到路由信息(z_s,dst,a),则验证 z 是否为相邻节点,同时坐标 z_s 的金额是否大于支付金额 a;

(3)若验证通过,则从与 z_s 的相邻坐标集中收款坐标 u_r:

① 若 $dst\in C(u)$,即 u 为收款方,则回复 z 收款坐标 u_r;

② 若 u 不是收款方,则进入阶段 2;

(4)若验证失败,回复节点 z 路由失败.

阶段 2.　选择下一跳路由节点.

(1)定义 FN 为距目标坐标更近的相邻节点集,即 $FN=\{w\mid w\in NN(u),\tilde{\delta}(w,dst)<\tilde{\delta}(u,dst)\}$;

(2)有效的下一跳节点集合为 $FW=FN\backslash FF$;

(3)若 FW 为空集,当 u 为付款方,则路由失败,否则回复节点 z 路由失败;

(4)否则,从中随机选取一节点作为下一跳路由节点 w,进入阶段 3.

阶段 3.　选择付款坐标.

(1)从与 w 的相邻坐标集中筛选出金额大于 a 的坐标集,定义为 FC;

(2)若 FC 为空,则 $FF=FF\cup\{w\}$,返回阶段 2;

(3)否则,从 FC 随机选取坐标 u_s 作为付款坐标,进入阶段 4.

阶段 4.　传递路由信息.

(1)向下一跳节点 w 发送路由信息(u_s,dst,a),并从坐标 u_s 处预留金额 a;

(2)若 w 在限定时间内回复路由信息 w_r,当 u 为付款方,路由成功,否则回复 z 收款坐标 u_r;

(3)若 w 未在限定时间内回复,或者回复路由失败,则恢复坐标 u_s 处的预留金额,$FF=FF\cup\{w\}$,返回阶段 2.

当付款方收到从下一跳节点返回的路由信息时,支付路径确立.

4.4　跨通道支付建立与完成

路径确定后,从跨通道支付的付款方开始,路径上每个节点通过选定的付款坐标与下一跳节点返回的收款坐标,以收款方生成的哈希值作为条件建立支付.

条件支付建立后,从收款方以哈希原像完成赎回支付开始,各节点以此哈希原像顺序完成支付.

至此,一笔跨通道支付完成.

5　多人通道网络分析

5.1　多人通道网络可行性分析

多人通道网络的可行性主要集中于通道内监管节点的引入,我们从其引入的必要性与可行性两方面进行分析.

5.1.1　监管节点引入的必要性

在多人通道内部实现通道状态一致性快速、准确达成,本质上是一个共识问题,主流解决方案[21]有经典共识机制如 Paxos 算法和 PBFT(Practical Byzantine Fault Tolerance)协议,区块链共识机制如 POW(Proof-of-Work)和 POS(Proof-of-Stake).然而在多人链下通道场景下,Paxos 算法不能容忍

恶意节点，PBFT 协议不支持节点动态退出且通信复杂度为 $O(l^2)$；区块链共识机制则与建立链下通道的初衷相悖. 因此，我们延续监管节点在多人通道中的引入，由其做每轮通道状态更新的提议者，以 $O(l)$ 的通信复杂度达成了多人通道内状态共识.

5.1.2 监管节点引入的可行性

对于通道内监管节点，其通过处理通道内支付而收取手续费获利. 对应地，其需要具有一定的资源与能力，包括抵押在通道内一定数额的保证金，处理通道内支付并存储每轮通道状态，获取链上信息并更新通道状态. 此外监管节点无须受各节点信任，因此选择方法较为灵活，可由通道内各节点协商选定.

对于通道内非监管节点，监管节点的引入在保证通道安全性的基础上，减轻了其计算负担与存储负担. 计算方面，在现有双人链下通道中，通道状态的更新需要双方共同完成，并且为防止通道内另一方提交过期通道状态至链上造成自身利益受损，节点需保持在线并检查链上信息. 在多人通道中，通道状态更新由监管节点完成，对通道状态的验证以及对链上通道信息的检测由通道内多参与方共同完成，缓解了对于单个节点计算资源的要求，同时即使在部分节点掉线的情况下依旧可以保证通道的正常运转. 存储方面，在双人通道网络中，节点需存储相邻节点的坐标以传递路由信息. 在多人通道网络中，非监管节点可仅保存自身坐标，并在收到路由信息时向监管节点问询，因此此方案对计算与存储资源受限的节点更为友好.

5.2 多人通道网络安全性分析

我们定义网络中的恶意节点具有概率多项式时间计算能力，即不能伪造消息签名与攻破哈希函数；并且可以加入任意通道，控制网络中部分节点或与其余节点合谋.

基于以上敌手模型，多人通道网络安全性定义为敌手不能使网络中任一诚实节点利益受损. 下面我们从多人通道内支付与跨通道支付方面分析多人通道网络的安全性.

(1) 在多人通道内，敌手可通过发动两种攻击获利：发起无效支付/生成不可达通道状态(敌手作为监管节点时)和退出通道时提交过期或错误通道状态. 此两项攻击在生效前均需接受通道内剩余节点监督与链上判决，一方面此攻击可使其余节点利益受损，另一方面作恶惩罚机制也为其监督提供了激励.

此外，敌手作为监管节点时可恶意拒绝对某笔有效支付的确认，此时节点可选择退出通道并取回在通道内的金额，敌手不能对其造成经济损失.

(2) 在跨通道支付时，敌手可通过恶意拒绝确认赎回支付却从中获取支付条件而获利，此时节点可提交赎回支付至链上完成支付.

综上所述，惩罚机制与链上判决的存在与对自身利益的维护促使各节点诚实行事，并监督其它节点的行为，从而保证多人通道的安全性与正常运转.

6 实验结果与分析

基于安全的多人通道链下支付，我们对多人通道网络的路由性能进行了实验验证，以此衡量跨通道支付的效率. 将结果与双人通道网络进行对比，以分析多人通道网络的实际可应用性.

6.1 实验设计

为与双人通道网络形成对比，实验采用相同的网络结构与支付数据集[15]. 首先我们将双人通道网络映射为多人通道网络，随后搭建相同的实验环境，并设定相应测试指标.

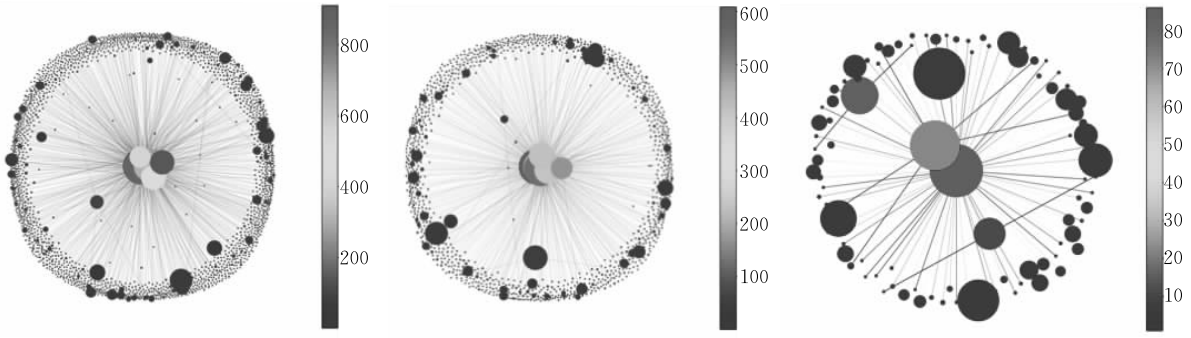
6.1.1 通道划分与网络建立

在双人通道网络结构 C_{Nov16}[15] 的基础上，为符合多人间高频支付的应用场景，与保持双人通道网络中各节点间连接关系，我们将可通过一跳到达的节点视为在同一个多人通道中，以此将双人通道网络中的节点映射至多人通道网络中.

对 C_{Nov16} 中包含的 67 149 个节点按照连通度进行统计排序，将度大于等于 d 的每个节点，与其相邻节点划分至一个通道内，并将此节点作为通道的监管节点. 依据具有相应度的节点所占比例，参数 d 取值为 5/6/29，对应生成 2068/1232/99 个通道. 对于通道划分后网络中存在的孤立节点，将其划分至距离最近的通道.

为保证多人通道网络的连通性，对于通道划分后网络中存在的一个孤立通道，设定其监管节点同时存在于离其最近的多人通道中. 为维持各节点在映射前后金额不变，我们将节点在原网络中的金额平均分配至所处的各多人通道中.

最终形成的多人通道网络结构如图 5 所示. 其中圆圈代表通道，圆圈的大小与通道中所包含节点个数的平方根成正比，圆圈的颜色深浅代表通道的度的大小. 从图中可以看出，通道之间关于所含节点数量的对比明显，少数通道包含了网络中的大部分节点，并且具有高连通度.



(a) $d=5$　　(b) $d=6$　　(c) $d=29$

图 5　以 d 为参数的多人通道网络结构图

在双人通道网络中,前 L 个连通度高的节点被选为根生成 L 棵树,支付金额被随机划分至 L 棵树上,在每棵树上寻找路径以完成支付.此时,支付金额的隐私性得以提升,即使敌手获取了一笔支付在某条或某几条路径上的支付金额,仍然无法确定此笔支付的总金额.类似地,在多人通道网络中,我们按照通道连通度选取前 L 个通道作为根生成 L 棵树.

6.1.2　实验环境搭建

类比于双人通道网络,我们假定各节点均为诚实节点,忽略支付费用以及支付过程的密码学实现,并将实验划分为静态实验与动态实验.

静态实验. 静态实验即在每笔支付完成后恢复网络至原始状态,因而每笔支付的路由环境相同,实现不同路由方法间的可比性,具体为

(1) 选取 $L=3$,设置一笔支付的尝试次数 $attempt=2$,若在第一次路由时失败,则在 $tl=2000$ 笔支付后重试;

(2) 每次运行与原实验中相同的 50 000 笔支付,运行 20 次并将结果取平均,结果记录在表 1 中.

动态实验. 动态实验即在每笔支付完成后不再恢复网络状态,相比静态实验更能反应出网络在实际环境中的可应用性.

由于双人通道网络中的动态实验涉及节点动态加入网络的过程,不适用于多人通道网络,因此我们在静态实验设定的基础上进行动态测试.此外,依据双人通道网络中反映出的随着 L 的增加,支付隐私性提升而成功率下降的情况,考虑到在现实链下支付的应用场景中节点对于成功率的更强需求,我们增加对 $L=1$ 设定下的网络性能的测试.为便于比较,不同设定下的动态实验结果均记录在表 2 中,便于比较.

6.1.3　实验测试指标

我们通过支付成功率、路径长度和路由开销量化多人通道网络路由性能.其中,支付成功率定义为成功完成的支付数量与总支付数量的比值,路径长度定义为支付平均在每棵树上所经历的节点跳数,路由开销定义为支付完成所需的路由信息个数.

此外我们测量网络稳定开销.双人通道网络实验中将网络稳定开销设定为因网络内支付而导致的各节点状态变化,进而导致的生成树结构变化,并以变动时所需改变的坐标数目来衡量.由于多人通道网络中各通道间连通度高,因某笔支付、或某个节点离开而导致通道不连通的概率很小.故我们定义多人通道网络的稳定开销为网络建立时的坐标广播数量,也称网络初始化开销.

6.2　实验结果与分析

6.2.1　静态实验结果分析与比较

如表 1 所示,多人通道网络的支付成功率稳定在 88%左右,并且随着 d 的增大而增大.d 增大代表网络内通道数减少,一方面,节点平均分配在各通道内的金额增加,在跨通道支付路由时可提供的金额增大,提高路由成功率.另一方面,通道数的减少使得同时存在于多个通道内的节点的个数减少,即潜在的中转节点数减少,从而降低了路由成功率.从结果来看,两种变化对成功率的影响相似且第一种变化的影响效果略大.

表 1　静态实验结果

	网络设置	成功率	路径长度/跳	路由开销/个
双人通道网络	SilentWhispers	0.65	5.30	82.0
	SpeedyMurmurs	0.906	1.87	18.3
多人通道网络	$d=5$	0.875	1.93	13.31
	$d=6$	0.877	1.93	13.25
	$d=29$	0.881	1.91	12.96

多人通道网络路径长度稳定在 1.9 跳左右，网络结构对其影响较小，原因在于网络中高连通度通道的存在使得任意两个节点所在的通道总是可以通过此类通道连通. 路由开销稳定在 13 左右，即每棵树上路由开销 4.3 左右，约为支付路径的两倍，即寻路总是可以一次完成，无需多次尝试.

与双人通道网络相比，多人通道网络的成功率高于 SilentWhispers，但是低于采用相同路由算法的 SpeedyMurmurs 方案的 90.6%的支付成功率. 究其原因，在双人通道网络映射至多人通道网络过程中，监管节点的各相邻节点间被添加了连接关系，除监管节点外剩余节点间的部分相连关系被舍弃. 添加连接关系的节点间原本即可通过监管节点相连，故对支付成功率的影响较小. 而舍弃的连接关系导致在原网络中连通的节点在映射后的网络中不再连通，从而导致支付成功率的下降. 故成功率的下降是网络映射所致，在实际应用场景下会有相应的提升.

在支付路径跳数方面，多人通道网络的性能近似于 SpeedyMurmurs，并优于 SilentWhispers. 路由开销方面，多人通道网络显著优于 SilentWhispers，相对同样路由算法的 SpeedyMurmurs，路由开销降低了约 28%，原因为多人通道网络中路由一次性完成的概率高，进而路由开销降低.

6.2.2 动态实验结果分析与比较

动态实验结果如表 2 所示. $L=3$ 时，相对于静态实验结果，多人通道网络在成功率上略有提升，在路径长度和路由开销上结果近似. 而双人通道网络在成功率上具有较大下降，SpeedyMurmurs 下降至 72%，相对于多人通道网络降低了 18%，路由开销略有降低但仍高于多人通道网络. 因此多人通道网络在动态环境下的性能更优，且面对网络环境变化性能更加稳定.

表 2 动态实验结果

	网络设置	成功率	路径长度/跳	路由开销/个
$L=3$	SilentWhispers	0.52	5.33	64.28
	SpeedyMurmurs	0.72	1.91	16.05
	$d=5$	0.881	1.93	13.43
	$d=6$	0.883	1.93	13.36
	$d=29$	0.887	1.91	13.10
$L=1$	SilentWhispers	0.57	4.57	10.55
	SpeedyMurmurs	0.94	1.79	5.46
	$d=5$	0.945	1.93	4.48
	$d=6$	0.946	1.93	4.46
	$d=29$	0.947	1.91	4.38

在 $L=1$ 时，各网络性能相对于 $L=3$ 时均有所提升，路由开销均下降至三分之一. SpeedyMurmurs 的支付成功率提升至 94%，相对提升了 23%，路径长度略有下降；多人通道网络的成功率提升至 95%，相对提升了 7%. 相比之下，网络设置对 SpeedyMurmurs 的影响更大，即多人通道网络的性能更加稳定.

6.2.3 网络稳定开销

在网络稳定开销方面，SilentWhispers 采用周期性更新方式，即每 $epoch=1000$ 笔交易后更新网络结构，此时开销为 598 722. SpeedyMurmurs 采用按需更新方式，当因支付导致节点间不连通时子节点重新选择父节点，此时的网络开销平均到每个周期中为 300，但是在网络变动剧烈时，网络稳定开销达 10^9.

多人通道网络在通道建立时需广播的坐标数量约为 6.4×10^8. 理论上，假设网络中共 N 个节点，m 个通道，第 i 个通道中节点数为 N_i，则需广播的坐标数量为

$$\sum_{i=1}^{m} N_i(N_i-1)=\sum_{i=1}^{m} N_i^2-N$$

$$\text{s.t.}\ \sum_{i=1}^{m} N_i=N,\ N_i\geq 2$$

可得网络建立时的通信复杂度为 $\Theta(N^2)$，与实验结果相符. 在实际应用中，网络稳定开销主要来源于新通道的加入. 由于通道间的相连关系主要基于连通度高的节点，因此对于新加入的低连通度的通道，其所需开销非常小. 若新加入的通道为高连通度通道，此时需对网络中的树结构有较大改变，但此种情况发生的概率较小.

7 总结与展望

本文提出了一种支持高并发的多人链下支付方案. 首先，在多人通道内部提高了支付处理效率，有效解决了实际应用中对于高并发链下支付处理的需求. 其次，实现了网络内跨通道支付与节点间支付路径的寻找，扩展了链下支付的范围. 最后，分析了多人通道网络的可行性与安全性，验证了网络路由性能. 实验结果显示，多人通道网络可实现 88%的支付成功率，在静态环境下相对于多人通道网络降低了约 28%的路由开销，同时面对网络环境与设置变化路由性能更加稳定，满足了实际应用场景中对于

多人链下支付的需求.

在隐私保护方面,跨通道支付的隐私保护特性类似于 SpeedyMurmurs,但尚缺乏对单个通道内节点的隐私保护.由于支付信息与各节点的余额信息在通道内公开,尽管在各频繁且相互交易的节点间存在一定的信任度,仍然不能免去隐私泄漏的风险.因此如何实现通道内的隐私保护是下一步研究中需要重点考虑的问题.

参 考 文 献

[1] Nakamoto S. Bitcoin: A peer-to-peer electronic cash system. White Paper, 2008

[2] Wood G. Ethereum: A secure decentralised generalised transaction ledger. Ethereum Project Yellow Paper, 2014, 151(2014): 1-32

[3] Croman K, Decker C, Eyal I, et al. On scaling decentralized blockchains//Proceedings of the International Conference on Financial Cryptography and Data Security. Berlin, Germany, 2016: 106-125

[4] Pan Chen, Liu Zhi-Qiang, Liu Zhen, et al. Research on scalability of blockchain technology: Problems and methods. Journal of Computer Research and Development, 2018, 55(10): 2099-2110(in Chinese)
(潘晨, 刘志强, 刘振等. 区块链可扩展性研究: 问题与方法. 计算机研究与发展, 2018, 55(10): 2099-2110)

[5] Poon J, Dryja T. The bitcoin lightning network: Scalable off-chain instant payments. White Paper, 2016

[6] Eyal I, Gencer A E, Sirer E G, et al. Bitcoin-NG: A scalable blockchain protocol//Proceedings of the 13th USENIX Symposium on Networked Systems Design and Implementation (NSDI'16). Santa Clara, USA, 2016: 45-59

[7] Luu L, Narayanan V, Zheng C, et al. A secure sharding protocol for open blockchains//Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security. Vienna, Austria, 2016: 17-30

[8] Decker C, Wattenhofer R. A fast and scalable payment network with bitcoin duplex micropayment channels//Proceedings of the Symposium on Self-Stabilizing Systems. Edmonton, Canada, 2015: 3-18

[9] Pan C, Tang S, Ge Z, et al. Gnocchi: Multiplexed payment channels for cryptocurrencies//Proceedings of the Network and System Security, 13th International Conference. Sapporo, Japan, 2019: 488-503

[10] Green M, Miers I. Bolt: Anonymous payment channels for decentralized currencies//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 473-489

[11] Zhang Y, Long Y, Liu Z, et al. Z-channel: Scalable and efficient scheme in zerocash. Computers & Security, 2019, 86: 112-131

[12] Prihodko P, Zhigulin S, Sahno M, et al. Flare: An approach to routing in lightning network. White Paper, 2016

[13] Malavolta G, Moreno-Sanchez P, Kate A, et al. SilentWhispers: Enforcing security and privacy in decentralized credit networks//Proceedings of the Network and Distributed System Security Symposium. San Diego, USA, 2017: 59-73

[14] Tsuchiya P F. The landmark hierarchy: A new hierarchy for routing in very large networks//Proceedings of the ACM SIGCOMM Computer Communication Review. Stanford, USA, 1988, 18(4): 35-42

[15] Roos S, Moreno-Sanchez P, Kate A, et al. Settling payments fast and private: Efficient decentralized routing for path-based transactions//Proceedings of the Network and Distributed System Security Symposium. San Diego, USA, 2018: 991-1005

[16] Roos S, Beck M, Strufe T. Anonymous addresses for efficient and resilient routing in F2F overlays//Proceedings of the IEEE INFOCOM 2016-the 35th Annual IEEE International Conference on Computer Communications. San Francisco, USA, 2016: 1-9

[17] Malavolta G, Moreno-Sanchez P, Kate A, et al. Concurrency and privacy with payment-channel networks//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 455-471

[18] Khalil R, Gervais A. Revive: Rebalancing off-blockchain payment networks//Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security. Dallas, USA, 2017: 439-453

[19] Miller A, Bentov I, Bakshi S, et al. Sprites and state channels: Payment networks that go faster than lightning//St. Kitts, Nevis eds. International Conference on Financial Cryptography and Data Security. Cham, USA: Springer, 2019: 508-526

[20] Malavolta G, Moreno-Sanchez P, Schneidewind C, et al. Anonymous multi-hop locks for blockchain scalability and interoperability//Proceedings of the 26th Annual Network and Distributed System Security Symposium. San Diego, USA, 2019: 1186-1200

[21] Gao Zheng-Feng, Zheng Ji-Lai, Tang Shu-Yang, et al. State-of-the-art survey of consensus mechanisms on DAG-based distributed ledger. Journal of Software, 2020, 31(4): 1124-1142(in Chinese)
(高政风, 郑继来, 汤舒扬等. 基于 DAG 的分布式账本共识机制研究. 软件学报, 2020, 31(4): 1124-1142)

GE Zhong-Hui, Ph. D. candidate. Her research interest is blockchain.

ZHANG Yi, M. S. candidate. His research interest is blockchain.

LONG Yu, Ph. D., associate professor. Her research interests include blockchain, information security and cryptography.

LIU Zhen, Ph. D., associate professor. His research interests include applied cryptography, blockchain security and privacy protection.

LIU Zhi-Qiang, Ph. D., associate professor. His research interests include blockchain, information security and cryptography.

GU Da-Wu, Ph. D., professor. His research interests include cryptography and computer security.

Background

Blockchain has witnessed rapid development in recent years since the release of Bitcoin. The emerging of Ethereum broadens the application of blockchain, based on which smart contracts can accomplish a complex logic. Nevertheless, the throughput of Bitcoin is 7 TPS and Ethereum is 15 TPS, which is far from real-world applications. Solutions such as Payment Channel Networks (PCNs), sharding and Bitcoin-NG have been proposed, among which PCNs move the on-chain payments into off-chain channels and support cross-channel payments in the network. Since PCNs don't involve properties of the underlying blockchain and regard it as an arbitration platform, it is of higher compatibility.

Various researches have been deployed in PCNs. In terms of one single payment channel, Green et al. proposed Bolt to construct privacy-preserving payment channels. Zhang et al. constructed payment channels in zero cash. As for routing algorithms in payment channel networks, Prihodko et al. proposed Flare in Lightning Network, which achieves fast path finding based on the storage of paths to neighbor and beacon nodes. Malavolta et al. presented SilentWhispers, which utilizes the landmarking routing and secure multi-party computation protocol in the distributed network to achieve the privacy-preserving off-chain payments. Roos et al. proposed SpeedyMurmurs on basis of the greedy embedding-based routing algorithm and compared performances of different routing algorithms in the distributed network. Moreover, Malavolta et al. studied the privacy and concurrency issues, and characterized their tradeoff. Khalil et al. presented the solution for nodes to rebalance their amounts in channels when it is depleted. Miller et al. proposed Sprites which utilizes a global contract to reduce the collateral cost of a multi-path payment. Malavolta et al. constructed anonymous multi-hop locks (AMHLs) facing the wormhole attack in PCNs.

However, all these works are based on two-party PCNs, where an off-chain payment with multiple payers or payees cannot be accomplished via a single channel or path. Besides, for multiple users who make payments mutually and frequently, establishing channels between each pair or making payments through intermediaries is complex and costly. Pan et al. extended the payment channel from two-party into multi-party. Nevertheless, it is of inefficient payments processing and payments are limited to single channels, which is far from real-world applications.

Our work in this paper enriches researches on multi-party off-chain payments. Firstly, we improved the efficiency of in-channel payment processing and achieved support for high concurrent off-chain payments. Secondly, we implemented cross-channel payments in the multi-party off-chain network. Finally, we proved the feasibility and security of the multi-party off-chain network. The performance of the routing algorithm under real-world data shows that it achieves a comparably good performance to PCNs in static scenarios and is more stable in dynamic scenarios. It is indicated that our work is deployable in practical applications.

This work is partially sponsored by the National Natural Science Foundation of China (Nos. 61672347, 61672339, 61872142, 61932014, 61572318), the "13th Five-Year" National Cryptography Development Fund (No. MMJJ20170111), the Shanghai Science and Technology Innovation Fund (Nos. 19511101403, 19511103900), and the Minhang Technology Innovation Program for SMEs (No. 2018MH110).