# 面向区块链的广义状态通道技术研究

肖 毅[1], 蒿敬波[1], 何 速[1], 姜新文[1,2]

[1](湘江区块链研究院,湖南 长沙 410008)

[2](湘潭大学 信息工程学院,湖南 湘潭 411105)

摘 要: 状态通道技术是区块链可扩展性研究的热点之一.状态通道为基于区块链的去中心化应用 DApp 提供了便利,可以显著降低 DApp 的延迟,促进了广义状态通道概念的提出.对近年来国外学者在该研究领域取得的成果进行了系统总结.首先介绍了状态通道的技术演进,随后对广义状态通道的概念设计进行了详细说明,最后对未来值得关注的研究方向进行了展望.

关键词: 区块链;可扩展性;支付通道;广义状态通道

中图法分类号: TP309

## A Survey of Generalized State Channel Technology for Blockchains

XIAO Yi[1], HAO Jing-Bo[1], HE Su[1], JIANG Xin-Wen[1,2]

[1](Xiangjiang Blockchain Institute, Changsha 410008, China)

[2](College of Information Engineering, Xiangtan University, Xiangtan 411105, China)

**Abstract**: State channel technology is an active research topic in the domain of blockchain scalability. It can facilitate decentralized applications (DApps) on blockchains with obvious latency reduction. For that reason the idea of generalized state channels has been proposed. This survey offers a systematic survey of existing research achievements of foreign researchers in recent years. First, the technology evolution of state channels is introduced. Next, a conceptual design of generalized state channels is specified. Finally a perspective of the future work in this research area is discussed.

**Key words**: blockchain; scalability; payment channel; generalized state channel

可扩展性(scalability)是区块链生态建设面临的基本问题之一,为此以太坊创始人 Vitalik Buterin 提出分片(sharding)与状态通道(state channel)[1]两种扩容(scaling)方案.分片是一种基于数据库分片概念的扩容技术,将区块链网络分成多个由不同网络节点组成的分片,每个分片负责处理一小部分交易,通过与链上的其他分片平行处理并相互协作来增强区块链的吞吐能力,因此通常被称作第一层或链上(on-chain)扩容.状态通道则保留底层的区块链运作模式,但仅将其作为最终结算层, 而把主要的交易操作放到链下执行.通过在不同实体之间建立一个双向通道,为参与各方提供状态维护服务,等链下交易操作完成多方签名确认后才将最终结果上链,因此通常被称作第二层或链下(off-chain)扩容.

状态通道技术为去中心化应用(decentralized application, DApp)提供了便利,可以显著降低 DApp 的延迟而将响应时间控制在用户的容忍范围内.状态通道是两方之间的互动,适用于任何智能合约,可以在保证交互性能和隐私性的同时,降低交易成本.相较来说,使用分片技术可以在一定程度上提升可扩展性,但是对于依赖大量琐碎交易操作的 DApp 来说无法有效降低成本,而通过状态通道则可以有效避免这一问题.

状态通道技术发展至今,经历了支付通道(payment channel)、状态通道和广义状态通道(generalized state channel)三个阶段,功能类型不断扩充和泛化.支付通道仅支持代币转移,状态通道则支持面向特定应用的任意状态转移操作,而广义状态通道更进一步支持应用的安装扩展[2].广义状态通道技术具有光明的发展前景,对于 DApp 的应用推广意义重大,因此本文的研究重点就集中在广义状态通道上面.本文第 1 节介绍状态通道的技术演进.第 2 节对广义状态通道的概念设计进行详细说明.最后对未来值得关注的研究方向进行探讨.

# 1 状态通道的技术演进

## 1.1 微支付通道

支付通道是在区块链之外双方进行代币交易的无信任机制[3].比特币网络每秒钟最多处理 7 笔交易,并且每笔交易还需要手续费,如果交易双方有大量的小额、微额交易,那么大量的交易不仅会让比特币网络负担沉重,而且手续费也不划算.为此,有人提出在交易双方之间建立微支付通道,专门支持小额支付,不再全部经过比特币网络.微支付通道除了在建立和关闭时需要和比特币网络通信,其他时间都是双方链下交易,其工作机制如下所示:

(1) 甲方发起 1 笔保证金交易(funding transaction),把资金送入一个多重签名的资金池,这笔资金需要交易双方同时出具私钥才能取出.

(2) 甲方会同时发起 1 笔退款交易(refund transaction),甲方先将这笔交易发送给乙方,乙方签名后再返回给甲方,以便作为甲方的一个保底措施,保证资金不会拿不回来.随后甲方才会将保证金交易发送给乙方,经双方签名后广播到区块链上.

(3) 双方开始持续进行链下交易,每笔交易都需要双方签名,称为更新交易(updated transaction)或承诺交易(commitment transaction),交易仅在双方之间传递,不会广播到区块链上.

(4) 所有交易完成后,甲方发起 1 个结算交易(settlement transaction),完成最后的资金分配,经双方签名后广播到区块链上并立即生效,交易终止.

微支付通道的缺点在于通道是单向的,只能实现资金的单向流动.

## 1.2 闪电网络

闪电网络(Lightning Network)是一种路由支付通道网络,它以比特币区块链为后盾,在链下实现真正的点对点微支付交易,其核心概念由序列到期可撤销合约 RSMC(revocable sequence maturity contract)与哈希时间锁合约 HTLC(hashed timelock contract)组成.

RSMC 的出现解决了微支付通道资金的单向流动问题,使撤销前个交易成为可能,由此奠定了双向微支付通道的基础,其工作机制如下所示:

(1) 交易双方将资金按协议比例放入资金池中,并在互相签署合约后,将总资金广播计入主链.

(2) 交易双方在不超过资金池总金额的前提下,可在链下进行交易,交易次数无限制.

(3) 每一次交易都必须签署全新的合约,合约仅在双方的交易通道中流转,并未广播计入主链.

(4) 当最后双方协议不再进行新交易,准备取回各自资金时,将由其中一方发起广播请求.

(5) 若一方发现交易结果不正确,可以根据双方交易合约中的前置条件,在有效时间内提出验证请求.

(6) 虚假交易一经证实,作假方在资金池内所有的自持资金将会作为赔偿支付给另一方.

RSMC 实现了简单的无条件资金支付通道,HTLC 则在其基础上进一步实现了有条件资金支付路径,其工作机制如下所示:

(1) 甲乙、乙丙、丙丁双方之间已建立 RSMC 通道,但甲丁之间没有建立通道.

(2) 甲方要付款给丁方 n 比特币,丁方先将一个秘密 s 的哈希值 H(s)发送给甲方.

(3) 甲方和乙方签订一个 HTLC,合约内容是:甲方支付乙方 n 比特币,前置条件为“乙方需要在 72 小时内出示 s 给甲方,否则交易自动取消”.

(4) 乙方和丙方签订一个 HTLC,合约内容是:乙方支付丙方 n 比特币,前置条件为“丙方需要在 48 小时内出示 s 给乙方,否则交易自动取消”.

(5) 丙方和丁方签订一个 HTLC,合约内容是:丙方支付丁方 n 比特币,前置条件为“丁方需要在 24 小时内出示 s 给丙方,否则交易自动取消”.

(6) 在规定时间内,s 由丁方交给丙方,丙方交给乙方,乙方交给甲方,大家从而依次拿到 n 比特币,甲丁双方的交易至此完成.

通过整合 RSMC 与 HTLC 这两种合约,再加入一定的路由机制,即可构成完整的闪电网络.例如,每当有节点

对另一节点发起支付时,它首先通过连接具有足够容量的支付通道来构建支付路径.节点宣传路由信息,包括它们已经打开多少通道,通道拥有多少容量,以及收取多少路由费用,路由信息以某种方式共享.随着闪电网络技术的进步,不同的路由协议可能会出现.类似地,雷电网络(Raiden Network)是以太坊的链下扩容方案,以闪电网络技术理念为基础,关键技术与闪电网络相一致,包括了 RSMC 和 HTLC.

### 1.3 状态通道

状态通道是支付通道的泛化形式,将支付通道的思想应用于区块链上任何发生状态改变的操作[4].状态通道的基本机制如下所示:

(1) 通过多重签名或智能合约锁定部分区块链状态,使得特定一组参与者必须达成完全一致才能实现状态更新.

(2) 参与者们自行创建和签名交易来更新状态,但更新全部保持在链下,不会立即提交到区块链上.

(3) 最终参与者们提交结果状态到区块链上,关闭状态通道,解锁状态.

在决定某种应用是否适合使用状态通道时,需要注意以下几点[5]:

(1) 状态通道依赖于网络可靠性.

(2) 状态通道在需要长期进行大量状态更新的情况下非常有用.创建一个通道会产生初始成本,但是一旦创建完成,通道内每一次状态更新的成本都很低.

(3) 状态通道最适于有一组明确参与者的应用程序.

(4) 状态通道具有很强的隐私性.绝大部分交易都发生通道内,而不是广播到链上.

(5) 状态通道的权威性是即时生效的.只要签署了一个状态更新,即可被认为是最终状态,在必要情况下可将状态提交到链上.

### 1.4 广义状态通道

状态通道可显著提高区块链的交易速度并降低交易成本,但目前在以太坊中并没有得到充分的利用.每一个想要使用状态通道的项目都必须要搭建自定义实现框架,从而导致冗余和不必要的风险出现.此外,现存的状态通道实现还是会将很多操作放在链上,而且需要放弃一些隐私性.为使这项技术能够被更好地利用,有人提出了广义状态通道的发展目标[2]:

(1) 设计一个可以保护隐私的状态通道,使用模块化组件搭建,支持单个通道内的多个并行操作,而且允许参与者在不进行任何链上操作的情况下升级通道设计.

(2) 提供一个框架和标准化组件,构建安全、性能良好的应用,让开发者能够更简单地使用状态通道.

广义状态通道代表了状态通道技术的未来,具有很高的研究价值.

## 2 广义状态通道概念设计

### 2.1 设计目标

为增强支持状态通道的区块链的可扩展性,需要将链上状态转移置入链下状态通道中,状态通道可以在保持相同语义的同时显著提高可扩展性.状态通道允许互不信任的各方在链下就最新状态迅速达成一致,并通过链上债权合约保证其不可篡改.这一理念最初由闪电网络引入,以支持高吞吐量的链下比特币小额交易.自闪电网络概念提出以来已有一些研究工作在支付通道的背景下解决了不同的问题,例如路由算法、时间锁优化和隐私保护.不过状态通道还处于早期发展阶段,在模块化、灵活性和成本效益等方面仍面临挑战.现有支付通道方案的一个主要局限在于缺乏对广义状态转移(generalized state transition)的支持,随着以太坊等智能合约平台的兴起,对广义状态转移的需求也随之而来.

相关研究[2,6]认为,广义状态通道应实现以下设计目标:

(1) 实现快速、灵活且无需信任的链下交互.多数时候链下状态转移将保持直到任务完成,为此需要优化常用的链下模式,使其支持与内置的链上组件之间进行简单交互.

(2) 设计适用于不同区块链的数据结构和交互逻辑.可以构建一个底层区块链之上的平台,并支持智能合约在不同的区块链上运行,因此需要一个通用的数据结构和中间层.

(3) 基于通道状态机的形式规约,验证实现状态更新的通信协议的安全属性,并尽可能提供有效的链上解决机制.

### 2.2 总体规范

本节将以 Celer 网络[6]为例,来说明广义状态通道的核心组件,并描述适用于任何支持价值传递与合约逻辑的状态通道的通用状态通道接口.首先介绍一些重要的符号和术语.

- 状态. 以 $s$ 表示通道状态,对于双向支付通道,$s$ 代表双方可用余额;对于棋盘游戏,$s$ 代表棋盘状态.
- 状态证明. 状态证明 $sp$ 作为链上合约和链下通信协议之间的桥接数据结构,包含以下字段:

$$sp = \{\Delta s, seq, merkle\_root, sigs\}$$

其中,$\Delta s$ 表示到目前为止累积的状态更新.给定一个基础状态 $s_0$ 和一个状态更新 $\Delta s$,可以生成唯一的新状态 $s$. $seq$ 是状态证明的序列号,高序列号的证明会抵消掉低序列号的证明.$merkle\_root$ 是所有未决条件组的的默克尔根,对于创建状态之间的条件依赖至关重要.$sigs$ 表示所有参与方对此状态证明的签名,仅当各方签名都存在时状态证明才有效.

- 条件. 条件 $cond$ 是表示条件依赖基本单元的数据结构,也是条件依赖 DAG 被构造的地方.一个条件可以表示如下:

$$cond = \{timeout, * \text{ ISFINALIZED}(args), * \text{ QUERYRESULT}(args)\}$$

其中,$timeout$ 表示条件到期的时间限制.布尔函数指针 ISFINALIZED($args$)用于检查条件在超时之前是否已经解析并且成立,其参数与应用相关.QUERYRESULT($args$)是一个结果查询函数指针,返回任意字节长度的条件解析结果.条件解析过程是先执行 ISFINALIZED($args$),然后再执行 QUERYRESULT($args$).

- 条件组. 条件组 $cond\_group$ 是对表示广义状态依赖的一组条件的高级抽象.一个条件组可以表示如下:

$$cond\_group = \{\Lambda, \text{RESOLVEGROUP}(cond\_results)\}$$

其中,$\Lambda$ 表示该条件组包含的一组条件,每个条件$cond \in \Lambda$可解析为 $cond$.QUERYRESULT($args$)输出的字节数组,这些字节数组由组解析函数 RESOLVEGROUP($cond\_results$)处理,它将全部条件解析结果作为输入并返回状态更新 $\Delta s$.对于一个支付通道,每个条件组对应于一个条件支付.

在此基础上,一个状态通道 $C$ 可以表示为以下元组:

$$C = \{p, s_0, sp, s, F, \tau\}$$

其中,$p = \{p_1, p_2, \ldots, p_n\}$表示此通道的参与者集合, $s_0$ 表示链上基础状态,$sp$ 表示已知最新的状态证明,$s$ 表示状态证明 $sp$ 成立后的最新通道状态,$\tau$ 表示状态证明成立时限的增量,$F$ 包含了每个状态通道都应该实现的一组标准函数:

- RESOLVESTATEPROOF($sp$, $cond\_groups$). 该函数通过解析关联的条件组来更新当前状态证明.
- GETUPDATEDSTATE($sp$, $s_0$). 该函数基于链下的状态证明 $sp$ 和链上基础状态 $s_0$ 来得到最新状态.
- UPDATESTATE($s$). 该函数允许在链上更新状态通道当前已解析的状态 $s$.
- INTENDSETTLE($new\_sp$). 该函数在成立时限之前开启一个挑战期,在挑战期内该函数接收状态证明作为输入,若输入较新则更新当前状态证明.
- CONFIRMSETTLE($sp$). 该函数会验证并确认当前状态证明已成立,如果当前时间超过成立时限.
- ISFINALIZED($args$)与 QUERYRESULT($args$). 解析条件依赖的入口点.
- CLOSESTATECHANNEL($s$). 该函数结束状态通道,并根据最后确定的状态 $s$ 进行链上更新.

状态证明的成立时限取决于上次调用 RESOLVESTATEPROOF 花费的时间,其增量为$\tau$.另外,在状态通道之间创建依赖关系时,为保证依赖 DAG 的正确解析,需要施加一定强制约束.

### 2.3 基础功能

上述抽象对广义状态通道构建的通用模式进行了定义,不同区块链中的具体实现可能有所不同.回顾多种

区块链上状态转移虚拟机的实现,可以总结出在实践中广义状态通道必需的两个基础功能:

(1) 链下地址转换器(off-chain address translator, OAT).在上面的抽象中,条件和条件组与不同的函数相关联,这些函数理应是对链上合约函数的引用.但智能合约状态本质上并非区块链上的约束,那么就不存在必须要有链上存在的需求基础.使之完全脱链的唯一障碍是对诸如 ISFINALIZED 和 QUERYRESULT 等函数的引用可能存在歧义.为解决这种歧义,可以在链上定义一套规则集,来将链下引用映射到链上引用,OAT 便由此而来.对于不涉及价值的合约,可通过其合约代码、初始状态和特定 nonce 生成的唯一标识来引用,称之为链下地址.当解析链上条件时,被引用合约应当能被调用,并且相对应的函数(如 ISFINALIZED 和 QUERYRESULT)应能将链下地址转换为链上地址.为实现此功能,OAT 需要将链上合约的合约代码和初始状态部署至链上,并建立从链下地址到链上地址的映射.

(2) 哈希时间锁注册表(hash time lock registry, HTLR).哈希时间锁 HTL 通常应用于需要保持原子性且涉及多个状态通道的交易场景.HTL 可以完全在链下实现,但是这属于一种过度优化,实际上限制了链下的可扩展性.文献[7]中提出了一个所有时间锁都能引用的中心注册表机制,对其进行扩展即可得到 HTLR.HTLR 为充当锁的条件提供依赖端点(ISFINALIZED 和 QUERYRESULT).ISFINALIZED 接受一个哈希值和区块号,如果哈希值对应的原像在指定区块号之前已注册则返回 true;QUERYRESULT 接受一个哈希值,如果对应的原像已注册则返回 true.HTLR 以及相关的 ISFINALIZED 和 QUERYRESULT 始终位于链上.

## 3 未来工作

由于状态通道建立成本的原因,在每对节点间建立通道并不实际.因此,有必要建立一个由状态通道组成的网络,其中状态传输的中继应该以无需信任的方式进行.这其中路由的设计尤为关键,它决定了状态通道网络的可扩展性,然而现有方案都难以满足状态通道网络的独特属性带来的挑战.现有的路由机制都可归结为基于可用余额考虑的最短路径路由.在传统数据网络中,最短路径算法确实可以提供较好的吞吐量和延迟性能,但这是基于网络拓扑稳定且链接容量无状态的假设.状态通道网络属于有状态的链接模型,并且网络拓扑也会随时发生变化,因此上述假设不再成立.面对状态通道网络路由的这一挑战,未来还有很多研究工作要做.此外,还可以结合高速共识算法,以便进一步增强区块链的可扩展性.例如目前已公开的全球首个达到百万 TPS 量级的实用化区块链 4.0 项目 InterValue[8],在这方面就进行了有益的尝试,值得参考和借鉴.

**References**:


[1] Buterin V. Ethereum：Platform review - Opportunities and challenges for private and consortium blockchains. Technical Report, 2016. https://static1.squarespace.com/static/55f73743e4b051cfcc0b02cf/t/57506f387da24ff6bdecb3c1/1464889147417/Ethereum_Paper.pdf

[2] Coleman J, Horne L, Li XJ. Counterfactual: Generalized state channels. Technical Report, 2018. https://l4.ventures/papers/statechannels.pdf

[3] Antonopoulos AM. Mastering Bitcoin: Programming the Open Blockchain (2nd Ed.). Sebastopol: O'Reilly Media, 2017.

[4] Coleman J. State channels. Technical Report, 2015. https://www.jeffcoleman.ca/state-channels/

[5] Stark J. Making sense of Ethereum's layer 2 scaling solutions: State channels, Plasma, and Truebit. Technical Report, 2018. https://medium.com/l4-media/making-sense-of-ethereums-layer-2-scaling-solutions-state-channels-plasma-and-truebit-22cb40dcc2f4

[6] ScaleSphere Foundation. Celer network: Bring Internet scale to every blockchain. White Paper, 2018. https://www.celer.network/doc/CelerNetwork-Whitepaper.pdf

[7] Miller A, Bentov I, Kumaresan R, Cordi C, McCorry P. Sprites and state channels: Payment networks that go faster than Lightning network. Technical Report, 2017. https://arxiv.org/pdf/1702.05812.pdf

[8] InterValue Team. InterValue: Connect, transfer and exchange all digital assets over the world. White Paper, 2018. https://www.inve.one/InterValue_whitepaper_en.pdf