**Celer Network: 将每个区块链扩展到互联网规模**

ScaleSphere Foundation Ltd. (“Foundation”)

2018年6月15日

草稿版本，会有修改

**摘要.** 就像90年代56kbps拨号互联网不可能支持4K视频流一样，今天区块链的可扩展性问题是限制其使用落地的关键因素。当前区块链的吞吐量较低，因为每个操作都需要经过绝大多数节点的处理才能达到链上共识，这正是“如何构建一个超慢速的分布式系统”。而且，链上共识方案也会导致隐私问题，因为任何节点都可以看到其他人的全部历史交易。虽然新的共识算法不断被提出和发展，但链上共识天生具有其局限性。

链下扩展技术允许相互不信任的各方在本地而不是在全球区块链网络上执行合约。参与交易的各方维护多签名的链下复制状态机，并且只有在绝对必要时（例如，当双方对某个状态有分歧时），才诉诸链上裁决。链下扩展是支持完全扩展分布式应用程序（“dApps”）的唯一方法，它具有更好的隐私性，完全可信任，保证去中心化。它是区块链大规模应用的拐点，将成为所有可扩展dApps的引擎。

Celer Network是一个互联网规模、无信任和隐私保护的平台，每个人都可以快速构建、操作和使用可扩展的dApps。它不是一个独立的区块链，而是一个运行在现有和未来区块链之上的网络系统，它通过创新的链下扩展技术和公平激励的加密经济机制提供了前所未有的性能和灵活性。

Celer Network是分层架构，具有清晰的抽象，能够快速进化每个独立的组件，包括支持快速和通用的链下状态转换的广义状态通道和侧链套件；一个可证明的最优值传输路由机制，通过相比于最先进的解决方案；为链下应用程序提供强大的开发框架和运行环境；以及为链下生态系统提供一个新的加密经济模型，用户网络规模、稳定流动性和高可用性的保障。

**重要提示：继续阅读之前您必须阅读以下免责声明**

CELR（“代币”）即为本白皮书中详述的Celer Network使用的加密代币，CELR的转卖（“代币转卖”），仅针对特定人员。本白皮书不是任何形式的招股说明书或要约文件，不构成任何形式的证券要约、商业信托中的单位、集体投资计划中的单位或任何其他形式的投资要约，也不构成任何司法管辖区内任何形式的投资请求。监管机构未审查或批准本白皮书中的任何信息。本白皮书尚未在任何司法管辖区的任何监管机构注册。

访问和/或接受本白皮书或其任何部分（视情况而定）中的任何信息，您将向cSpeed有限公司（BVI商业公司编号1976167）（“代币供应商”）声明并保证：

1. 你不是被排除在外的人（定义如下）；
2. 您不在禁止、限制或未经授权以任何形式或方式进行代币销售的司法管辖区内，无论其法律、监管要求或规则是全部还是部分禁止、限制或未经授权；
3. 您已阅读本白皮书的全部内容，并了解您购买代币所带来的风险；
4. 贵方同意受本协议所述限制和限制的约束；
5. 您承认，本白皮书已准备好交付给您，以帮助您决定是否购买代币，仅供参考，不打算在您和代币供应商之间建立法律关系，或对代币供应商具有法律约束力或强制执行效力。

如果您不同意上述任何一项，请不要进一步阅读本白皮书的内容。请参阅标题为“重要通知”的章节和标题为“免责声明”、“无陈述和保证”、“您的陈述和保证”、“前瞻性声明的注意事项”、“第三方信息和其他人的不同意”、“使用的条款”、“无建议”的章节。在继续本白皮书之前，请仔细阅读“无进一步信息或更新”、“分配和传播限制”、“无投资或注册要约”、“风险和不确定性”。

目录

[**1 简介** **4**](#_Toc99270)

[1.1 Celer技术栈 5](#_Toc99271)

[1.2 Celer密码学 8](#_Toc99272)

[**2 cChannel: 链下扩展的功能** **10**](#_Toc99273)

[2.1 广义状态通道 10](#_Toc99274)

[2.1.1 主要思想和简单例子 10](#_Toc99275)

[2.1.2 设计目标 12](#_Toc99276)

[2.1.3 通用规范 12](#_Toc99277)

[2.1.4 通用基础组件 16](#_Toc99278)

[2.1.5 开箱即用 17](#_Toc99279)

[2.2 备选侧链通道模型 19](#_Toc99280)

[**3 cRoute: 最优证明的传输路由** **21**](#_Toc99281)

[3.1 状态通道网络路由的挑战 21](#_Toc99282)

[3.2 分布式平衡路由 (DBR) 23](#_Toc99283)

[3.2.1 系统模型 24](#_Toc99284)

[3.2.2 协议描述 25](#_Toc99285)

[3.2.3 DBR的吞吐性能 27](#_Toc99286)

[3.3 DBR分析 33](#_Toc99287)

[3.3.1 故障恢复能力 33](#_Toc99288)

[3.3.2 隐私 33](#_Toc99289)

[3.4 模拟结果 33](#_Toc99290)

[**4 cOS: 链下分布式应用程序操作系统** **35**](#_Toc99291)

[4.1 有条件依赖的状态的有向无环图 35](#_Toc99292)

[4.2 链下应用程序开发框架 36](#_Toc99293)

[4.3 链下应用程序运行 38](#_Toc99294)

[**5 cEconomy: 链下加密经济机制设计** **40**](#_Toc99295)

[5.1 链下生态系统的取舍 41](#_Toc99296)

[5.1.1 链下的可扩展性和流动性 41](#_Toc99297)

[5.1.2 链下的可扩展性和可用性 42](#_Toc99298)

[5.2 cEconomy 设计 43](#_Toc99299)

[5.2.1 流动性保证证明（PoLC）挖矿 44](#_Toc99300)

[5.2.2 流动性抵押拍卖 (LiBA) 45](#_Toc99301)

[5.2.3 状态守护网络 49](#_Toc99302)

[5.2.4 总结 52](#_Toc99303)

[**6 结论** **52**](#_Toc99304)

[**7 创始团队** **53**](#_Toc99305)

# 简介

许多现代经济活动本质上是信息和价值的流动和交换。在过去的两个世纪里，信息的传递已经从通过鸽子网络发送简单信息发展到通过光速互联网传输大量信息。然而，价值传输部分离光速还很远，仍然是由隔离的金融体系控制。这种不匹配造成了经济发展的毁灭性瓶颈：无论信息流动有多快，昂贵而缓慢的价值传输都限制了两者的生产产出。

区块链技术本质上是对不信任方之间的信任的革命性抽象，它有公平激励的分布式共识，是废除隔离的金融体系的基础，并显著扩大了全球价值流动的范围和自由度。然而，在实践中，与传统的价值传输工具相比，区块链的处理能力较低，它的价值传输还是比较缓慢，可扩展性是区块链技术大规模应用面临的一个根本挑战。

我们设想一个分布式的生态系统的未来，人、计算机、移动工具和物联网（“IOT”）设备可以大规模执行安全、隐私和无信任的信息价值传输。为了实现这一点，区块链应该与互联网的规模相匹配，并且每秒支持数亿或数十亿个交易。然而，考虑到现有区块链的处理速度（即每秒几次或数十次交易），是否真的有可能将互联网的规模扩展到区块链？答案是肯定的，但只有链下扩展技术。

而链上共识是区块链技术的基础，其局限性也是显而易见的。在某种意义上，共识与可扩展性是相驳的。对于任何分布式系统，如果所有节点都需要对每一个交易达成共识，那么它整体的性能（实际上，由于通信开销的原因）将不会比处理每个事务的单个节点的集中式系统要好，这意味着整体系统性能最终会受到最慢节点的处理能力的限制。链上共识对隐私也有严重危害，因为所有交易都是永久公开的。目前已经提出了一些链上共识的改进，包括分片和各种Proof-of-X共识机制。它们使区块链在性能、去中心化、安全性和终局性上的进行不同取舍获得某些更优特性，但不能改变链上共识天生的局限性。

为了使互联网规模的区块链系统具有更好的隐私性，并且在信任和去中心化方面没有妥协，我们不能局限于链上共识的改进。设计可扩展的分布式系统的核心原则是使不同节点上的操作基本上独立，一个简单的直觉表明，完全扩展去中心化应用程序的唯一方法是将大部分交易引入到链下，尽可能避免链上共识，并作为最后的选择。相关技术包括状态通道、侧链和链下计算Oracle。尽管链下扩展技术具有很高的潜力，但仍处于起步阶段，许多技术和经济挑战尚未解决。

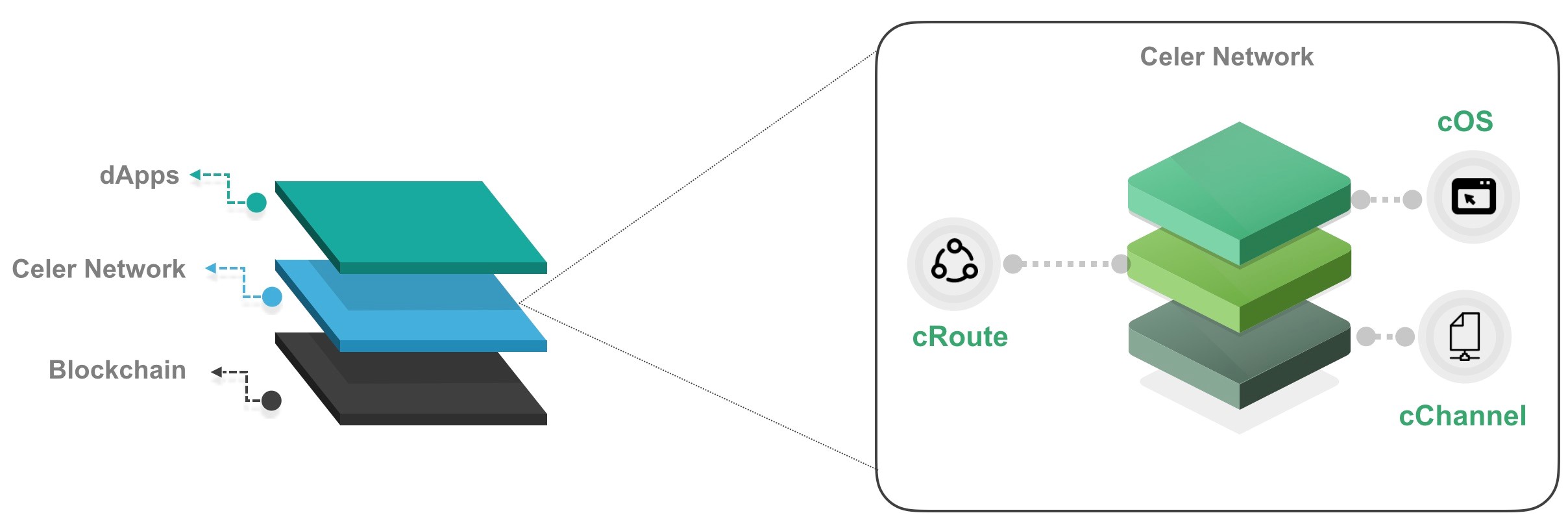
为了使链下扩展能够使用，我们提出了Celer Network，一种将互联网规模扩展到现有和未来区块链的一致架构。Celer Network的组成包括一个精心设计的链下扩展技术栈，该技术栈具有高度可扩展性和灵活性，具有强大的安全性和隐私保护，一个基于博弈理论的加密经济模型。

## Celer 技术栈

作为一个可以构建在现有和未来区块链基础上的全栈平台，Celer Network包含一个清晰的分层架构，将复杂的链下平台分离为分层模块。这种体系结构极大地简化了系统设计、开发和维护，从而使每个单独的组件都可以轻松地修改和升级。

一个设计良好的分层体系结构应该具有开放的接口，只要它们支持相同的跨层接口，就可以在每一层上启用不同的实现。每一层只需要专注于实现自己的功能。受互联网成功分层设计的启发，Celer Network采用了一个链下技术栈，可以构建在不同的区块链上，称为cStack，由以下层自下而上组成：

* **cChannel**: 广义状态通道和侧链套件。
* **cRoute**: 可证明的最优价值传输路由。
* **cOS**: 链下应用程序开发框架和运行环境



**Figure 1.** Celer Network分层

Celer架构是所有层可创新的解决方案，下面我们将重点介绍cChannel、cRoute和cOS的技术挑战和功能特性。

**cChannel.** 这是Celer Network的底层，它与不同的底层区块链交互，并为上层提供最新状态和有界时间的终局性的通用抽象。cChannel使用状态通道和侧链技术，这两种技术都是链下扩展平台的基础。

状态通道允许相互不信任的各方在链下执行程序，并快速确认最新一致的状态，其安全性和终局性由链上债券合约保证。它最初由Lightning Network[9]引入，以支持高吞吐要求的比特币小额支付。自从引入Lightning Network以来，已经有几项研究工作解决了支付通道网络中的各种问题，例如路由[10、13、16]、时间锁[5]和隐私[6]。然而，链下网络还处于起步阶段，在模块化、灵活性和成本效率方面面临着一些重大挑战。cChannel通过提供一系列新功能来满足当前的挑战。

* **广义状态通道.** 链下交易是DAG依赖的随意状态转换。这使得Celer Network能够支持复杂的高性能链下dApps，如游戏、在线拍卖、保险、预测市场和去中心化交易所。
* **灵活高效的价值传输.** 提供了多状态通道和侧链，在效率和终局性上有不同的权衡，以支持条件依赖、最小链上交互和最小资金锁定的快速价值传输。
* **纯链下合约.** 任何与链上抵押没有直接关系的合约不需要任何链上操作或初始化，除非触发了争议。每个纯链下合约或对象都有一个唯一可识别的链下地址，并且只需要在必要时在区块链上部署，链上地址由内置的链下地址转换器分配。

**cRoute.** Celer Network是一个高度可扩展的dApps平台，旨在支持高吞吐量的价值传输。链下价值传输是许多链下应用的基本要求。Celer Network能够支持除支付解决方案以外的dApps，它也对链下支付网络路由进行了创新性的改进，这些直接决定了价值在生态系统中的传输量和传输速度。

现有的链下支付网络路由方案[3、4、10、12、13、16]都归结为传统的“最短路径路由”算法，由于链路模型中的基本差异，现有的链下支付网络路由的性能可能较差。计算机网络的链路容量是无状态和稳定的（即，不受过去传输的影响）。然而，链下支付网络的链路容量是有状态的（即由链上抵押和过去支付决定的），这导致了一个高度动态的网络，其中拓扑和链路状态不断变化，这使得传统的最短路径路由算法难以收敛，从而产生低吞吐量、长延迟甚至中断。

为了应对这一基本挑战，Celer Network的支付网络路由模块cRoute引入了分布式均衡路由（DBR），它使用分布式拥塞梯度来路由支付。我们重点介绍了DBR的一些独特特性（详见第3.2.2节）。

* **可证明的最优吞吐量.** 我们可以证明，对于任何一个全局支付到达率，如果存在一个支持该比率的路由算法，那么DBR就可以满足。我们的评估表明，与最先进的解决方案相比，DBR的吞吐量提高了15倍，通道利用率提高了20倍。
* **彻底的通道平衡.** “保持通道平衡”是雷电网络提出以来要实现的目标。但是，现有的通道平衡尝试包括启发式方法，这些方法需要大量的链上或链下协调，但没有结果可靠的保证。DBR将通道平衡过程和路由结合，并在不需要任何额外协调的情况下持续地平衡网络。
* **完全去中心化**. DBR算法是一种完全去中心化的算法，其中每个节点只需要在状态通道网络拓扑中与其邻居“对话”。DBR的协议还降低了消息传递成本。
* **故障恢复能力**. DBR算法具有很强的抗故障能力：它可以快速检测和适应不响应的节点，可以支持剩余可用节点最大可能吞吐量。
* **隐私保护**. DBR算法可以与洋葱路由[11]无缝集成，以保持源/目的地的匿名性。由于DBR算法具有多路径特性，因此自然地保留了传输价值数量的隐私，而不使用任何额外的隐私保护技术（例如，zksnark）。

**cOS.** 一个链上dApp只是一个连接到区块链的前端。链下dApps虽然具有很高的可扩展性潜力，但与传统的链上dApps相比，不易构建和使用。Celer Network引入了cOS，这是一个开发框架和运行时环境，每个人都可以轻松地开发、操作和交互可扩展的链下dApps，而不会被链下扩展带来的额外复杂性所困扰。Celer Network允许开发人员集中精力于应用程序逻辑并创建最佳的用户体验，cOS处理繁重的工作，包括以下任务。

* 找出任意链下和链上状态之间的依赖关系.
* 处理链下状态的跟踪、存储和争议.
* 容忍中间节点的故障.
* 支持多并发链下dApps.
* 不同的链上和链下模块编译的统一实现.

## Celer 密码学

Celer Network的加密经济机制cEconomy是基于一个基本原则设计的：一个好的加密经济（token）模型应该提供额外的价值，并使新的博弈理论成为可能。在获得可扩展性的同时，链下平台也在进行网络流动性和状态可用性方面的权衡。

**新的权衡.** 链下平台通过以下权衡获得可扩展性.

* **可扩展性和流动性.** 链下价值传输要求链上抵押来提供网络流动性。这对潜在的链下服务提供商尤其具有挑战性，因为需要大量的流动性来为全球区块链用户提供高效的链下服务，无论是作为状态通道中的抵押还是侧链中的欺诈处罚债券。然而，大额加密资产的所有者（鲸鱼）可能不具备商业利益或技术能力去运行链下基础设施服务，而具有运行可靠且可扩展链下服务的技术能力的人往往没有足够的资金用于通道抵押或防欺诈债券。这种不匹配为链下网络的大规模应用和技术发展创造了巨大障碍。
* **可扩展性和可用性.** 虽然链下扩展不会对区块链的无信任属性造成任何妥协，但它确实牺牲了可用性保证。每个状态通道或链下合约都有了一个争议期，当不在线时间超过争议期或本地状态丢失时，相关方将面临风险。

因此，我们需要一种激励机制，为能够运行可靠且可扩展的链下基础设施服务的实体提供充足的流动性，并确保链下状态始终可用于可能的链上争议。

**新加密经济.** 为了完成链下扩展的解决方案，我们引入了一套名为cEconomy的加密经济机制，它通过Celer Network的协议代币（“CELR”）和三个紧耦合组件带来了不可或缺的价值、保证了网络效应，稳定流动性和高可用性。

* **流动性承诺证明(PoLC).** POLC是一个虚拟的挖矿过程，为链下生态系统提供了丰富稳定的流动性。要参与，只需将其闲置流动性（以数字资产的形式，包括但不限于加密货币和CELR）承诺（锁定）到链下平台一段时间，CELR将奖励给这些用户作为激励。
* **流动性支持拍卖 (LiBA).** LIBA使链下服务提供商能够通过协商利率的“群体贷款”来吸引流动性。放款人根据其“幸福分数”排名，幸福分数由利率、提供的流动性金额和抵押的CELR金额决定。特别是，持有更多CELR股份（作为其对生态系统的过去贡献的指标）的贷款人有更高的优先选择权，以向链下服务提供商提供流动性。
* **状态监护网络 (SGN).** SGN是一种特殊的紧凑型侧链，当用户处于不在线状态时，它监护状态，以便用户的状态始终可供争议。监护人需要将其CELR投资于SGN，以从用户那里获得监护机会和服务费。

第5节详细介绍了cEconomy机制，分析了CELR价值和激励模型。

# cChannel: 链下扩展的基础

Celer Network的cChannel旨在提供一个框架，使状态通道和侧链网络具有最大的灵活性和效率。本节从广义通道构造开始，概述了支持链上可验证状态之间任意有条件依赖的关键元素。然后，我们将扩展经典状态通道、研究如何将侧链封装，并向上层暴露统一的接口。

## 广义状态通道

### 关键思想和一个简单例子

现有支付网络解决方案的一个主要局限是缺乏对广义状态转换的支持。随着诸如以太坊之类的智能合约平台的兴起，需要进行广义的状态转换。智能合约支持基于任意合约逻辑的异步价值传输。为了使用链下状态通道理念来提高此类区块链的可扩展性，应将链上状态转换放入链下状态通道中，并使状态转换应用到相应的价值传输。

我们使用一个简单的条件支付示例来说明将链上状态转换变换为链下状态转换的关键思想。假设Alice和Carl想玩一个棋盘游戏，同时以一种不受信任的方式对这类游戏的结果下注：如果Carl赢了，Alice将支付Carl1美元，反之亦然。

这有一个在链上实现的简单逻辑。我们可以建立一个智能合约，在游戏开始前持有Alice和Carl的押金。Alice和Carl将通过调用链上智能合约的功能来玩这个游戏。当他们中的一个输掉比赛、投降或超时，胜利者将得到输家的押金。智能合约中的押金也是双方的存款，可视为有条件发行的付款（即交易对手获胜的条件）。不幸的是，链上智能合约操作非常缓慢和昂贵，因为每个操作都涉及链上交易。

链下状态通道可以在保持相同语义的同时，显著提高可扩展性。假设Alice和Carl之间有支付通道。为了实现上述语义，我们需要扩展支付通道的状态证明功能，以包含一个依赖于游戏胜利者状态的条件锁。然后，Alice可以给Carl发送一个链下有条件的支付，并说：“如果游戏合约的who\_is\_winner函数说Carl赢了，我将支付Carl1美元。”，游戏状态转换也可以移动到链下。最直接的方法是仍然有一个规定了棋盘游戏规则的链上合约，并且该合约的地址在有条件付款中被引用。所有状态转换都是在链下发生的，相互签署的游戏状态在必要时可以被发送到链上合约中。

但事实上，由于程序状态不需要任何形式的价值保证，因此只要参与方是协作的，整个游戏合约和相关状态就可以始终保持在链下。唯一的要求是，当需要时，相关的博弈状态在链上是可验证的。链上可验证状态意味着其他合约或对象可以引用它而不产生歧义。要实现这一点，我们需要有一个引用转换器合约，它将链下引用（例如合约代码的hash、构造函数参数和nonce）映射到链上引用（合约地址）。有了这些构造，Alice和Carl之间的游戏只涉及一个不限于游戏逻辑的长期链上合约，并且没有链上操作或游戏初始化。

上面的示例是链下设计模式的一个典型而简单的实例，它可以更复杂。有条件支付可能比简单的布尔条件更复杂，并且可以设计为基于任意合约逻辑重新分配锁定的流动性。事实上，有条件支付只是更普遍的有条件状态转换的一个特例。为了实现多跳状态中继的通用模式，信道依赖也可能比一对一依赖更复杂。我们在以下章节中详细说明了技术规范。

### 设计目标

我们的首要目标是实现快速、灵活和无信任的链下交互。我们预计在大多数情况下，链下状态转换将保持在链下直到最终结束。因此，我们的目标是将常用的链下模式优化，与内置的链上组件进行简洁的交互。

我们的第二个目标是设计适用于不同区块链的数据结构和对象交互逻辑。Celer Network旨在构建一个区块链不可知论平台，并在支持智能合约的不同区块链上运行。因此，需要一个通用的数据结构语义和一定的中间层。

除了这两个突出的目标之外，我们计划使用通道状态机的正式规范，并验证安全属性以及更改这些状态的通信协议。我们还应尽可能提供一个有效的链上裁决机制。

### 通用规范

在本节中，我们用自顶向下的方法提供了cChannel广义状态信道核心组件的规范，并描述了适用于具有任何价值传递和任意合约逻辑的状态通道的公共状态通道接口。对于不同的具体用例，可以有扩展的规范和优化，但是原理是相同的。

在详细说明广义状态通道之前，我们首先介绍几个重要的符号和术语，这些符号和术语将在本节中使用。

* (**State**). 用s表示通道的状态。对于双方支付通道，s代表双方的可用余额；对于棋盘游戏，s代表棋盘状态。
* (**State Proof**). 状态证明是链上合约和链下通信协议之间的接口数据结构。状态证明sp包含以下字段

*sp ={Δs,seq,merkle root,sigs}* *,* (1)

其中Δs表示到目前为止的累计状态更新。请注意，给定基态和状态更新Δs，我们可以唯一地生成一个新的通道状态s。例如，在双方支付通道中，基态对应于双方的抵押，状态更新Δs是一个映射，指示从一个参与者转移到另一个参与者的代币量。seq是状态证明的序列号。序列号较高的状态证明将禁用序列号较低的状态证明。merkle\_root是所有pending条件组的merkle tree的根，对于在cChannel中创建状态之间的条件依赖关系至关重要。最后，sigs代表此状态证明上所有各方的签名。只有当所有当事方都有签名时，状态证明才有效。

* (**条件**). cond是表示条件依赖的数据结构，条件依赖DAG建立在其上。一个条件可以指定如下。

*cond ={timeout,\*IsFinalized(args),\*QueryResult(args)}* (2)

这里，timeout是条件过期的超时时间。例如，对于取决于棋盘游戏结果的条件，超时可能对应于棋盘游戏的最大持续时间（例如，十分钟）。布尔函数指针IsFinalized(args)用于检查条件是否在条件超时之前已解决。此函数调用的参数是特定于应用程序的。例如，在棋盘游戏中，参数可以简单到args=[blocknumber]查询是否在blocknumber之前确定了游戏胜利者。另外，QueryResult(args)是一个结果查询函数指针，返回任意字节用于条件解析的结果。例如，在棋盘游戏中，参数可以是args=[player1]查询player1是否是赢家（布尔条件）；在二次价格拍卖中，参数可以是args=[participant1，participant2，····，participant]查询谁是赢家以及每个参与者应该支付的金额（一般条件）。条件的解决程序是首先执行IsFinalized(args)，然后执行结果查询QueryResult(args)。

* (**条件组**). 条件组 *cond\_group* 是表示广义状态依赖性的一组条件,条件组可以指定如下。

*cond group = {Λ,ResolveGroup(cond\_results)}* *,* (3)

其中Λ表示此条件组中包含的一组条件。每个条件解析为一个任意字节数组（即cond.QueryResult(args)的输出）。这些字节数组由组解析函数ResolveGroup(cond\_results)处理，将所有条件的解析结果作为输入并返回状态更新Δs。对于支付通道，每个条件组对应于有条件付款。例如，一个条件付款表示“A支付B1美元如果B赢了Gomoku游戏”，对应于包含两个条件的条件组：哈希时间锁定条件（用于多跳中继）和Gomoku游戏条件（“B赢得游戏”）。ResolveGroup函数只返回从A到B的转账1美元，如果两个条件都是真实的。

现在我们已经准备好为状态通道指定接口了。状态通道C可以指定为以下元组：

*,* (4)

是此通道的参与者集合， 是该通道的链上基态（例如，支付通道中每个参与者的初始抵押额）,sp表示该通道的最新已知状态证明。S是状态证明sp完全解决后更新的通状态，是稍后将指定的状态证明的计算超时增量。 包含一组应由每个状态通道实现的标准函数:

* ResolveStateProof(*sp, cond\_groups*). 此函数通过解析附加条件组来更新当前状态证明。
* GetUpdatedState(*sp, s*0). 此函数用于根据链下状态证明sp和链上基态获取最新状态。
* UpdateState(*s*). 此功能允许链上更新状态通道的当前裁决状态s.
* IntendSettle(*new sp*). 此函数在结算之前打开一个质询期。在质询期间，此函数将状态证明作为输入，如果输入更新，则更新当前状态证明。
* ConfirmSettle(*sp*). 如果当前时间过了结算质询期，此函数将验证并确认当前状态证明已完全结算。
* IsFinalized(*args*) and QueryResult(*args*) 是解析条件依赖项的入口点。它接受带有必要参数的外部查询，以便查询合约进行相应的解释。实际上，一些模式被频繁使用，在cChannel的实现中，我们将它们划分为预先定义的函数接口。
* CloseStateChannel(*s*). 此函数终止状态通道的生命周期，并根据最新的已结算状态s分配必要的状态。

结算质询期根据上一次调用ResolveStateProof或SettleStateProof的时间确定，结算质询期时间增量为。

**依赖性约束.** 当我们在不同的状态通道之间创建依赖关系时，为了保证依赖关系DAG的正确解析，需要强制执行一些约束。假设状态通道C1依赖于状态通道C2。然后，要求C1的参与者是C2参与者的一个子集，以便C1的参与者具有解决其对C1依赖性的必要信息。.

### 通用组件

上述抽象定义了广义状态通道构造的通用模式。在不同的区块链中，实际实现可能不同。例如，在以太坊中，交叉合约调用包含返回值，但在Dfinity中，交叉合约调用只触发注册的回调。回顾了多个区块链在状态转移虚拟机上的实现，我们确定了在实践中两个对广义状态通道操作至关重要的通用组件，如下所示:

* **链下地址翻译 (OAT).** 在上述抽象中，条件和条件组与不同的函数相关联。这些功能应该是链上合约功能的参考，但由于程序（智能合约）状态本身不受区块链约束，因此不应该有链上存在的基本要求。将它们完全移到链上的唯一障碍是对函数（如IsFinalized和QueryResult）的引用可能不明确。

为了解决这种歧义，我们可以定义一个链上规则集来将链下引用映射到链上引用。链下地址转换器是为它而构建的。对于不涉及值的合约，可以通过其合约代码、初始状态和某个nonce生成唯一标识符来引用它。我们称这种唯一标识符为链下地址。当在链上解决条件，需要部署引用的合约，相应的功能（如：IsFinalized和QueryResult）应能将链下地址转换为链上地址。为了实现这一功能，OAT需要能够部署合约代码和初始化链上合约，并建立从链下地址到链上地址的映射。.

* **哈希时间锁注册表(HTLR).** 哈希时间锁通常用于涉及多个状态通道的交易需要原子地发生的情况。例如，多跳中继支付（无条件或有条件）、不同token之间的原子交换、跨链桥接等。HTL可以完全在链下实现，但正如Sprite[5]指出的，这是一种过度优化，实际上限制了链下的可扩展性。因此，Sprite[5]提出了一个中央注册表，所有锁都可以引用该注册表。我们扩展和修改Sprite以适应cChannel的公共模型。实际上，HTLR为锁的条件提供依赖终结点（IsFinalized，QueryResult）。IsFinalized获取哈希和区块号，如果在区块号之前注册了相应的预映像，则返回true。如果哈希的预映像已注册，则QueryResult将获得哈希并返回true。这两个函数可以进一步简化为一个函数，但出于一般性的考虑，我们可以简单地将它们保留为两个独立的函数。请注意，HTLR及其关联的IsFinalized和QueryResult始终是链上的。

### 开箱即用

此外，我们还需要研究通常使用的模式，并使用开箱即用的功能来增强某些链上组件，以简化相应的链下交互。广义支付通道（GPC）就是一个很好的例子。广义支付通道是符合广义状态通道规范的支付通道，因此可以支持基于未来链上或链下对象的各种有条件支付。

我们首先在GPC环境中使抽象模型更具体。代表p中每一方的抵押，s代表每一方拥有的最终状态。SubmitStateProof是在调用SettleStateProof并确认状态证明之前提交状态证明并触发挑战期的函数。IsFinalized和QueryResult是检查该支付通道状态是否已完成以及查询当前余额的功能。人们可能会想，为什么支付通道有外部查询的接口。这是因为某些其他付款或状态可能依赖于或存在某些锁定在sp中的有条件付款。ResolveStateProof是最有趣的部分，因为这里会有大量的优化，并大大降低链下交互的复杂性。GetUpdatedState是一个简单的函数，可以根据初始抵押和完全裁决的sp计算各方的净出付款。CloseStateChannel只需关闭通道并将净出的最终余额分配给各方。

使用这个基本模型，我们将讨论如何进一步优化GPC结构，以启用有用的开箱即用特性。

* **合作解决** 在大多数情况下，状体通道应用程序的参与方是协作的。因此，通过挑战期并解决问题会增加复杂性和费用。因此，cChannel能够被协商为参与方不仅签署最新的状态证明，而且签署最终结果，以表明该状态证明中描述的状态更新确实是最终状态。这样，结算状态证明的交易数可以从2减少到1。
* **单交易来打开通道** cChannel带来的另一个优化是将打开通道的链上操作数从3减少到1。这是通过使用一个依赖合约为通道参与者存储抵押来实现的。所有参与者只需在链下签名，一个参与者可以把这些提交到链上来完成通道打开流程。
* **直接最终状态声明** 在构建广义状态通道应用程序时，通常使用条件状态依赖。在完成GPC时，一方可能希望避免遍历条件依赖关系图。这是为了限制对方的欺诈，有时对方会不在线，会拒绝合作将某些条件组转换为无条件状态更新。为了限制争议方所需的工作，我们介绍了直接最终状态声明的方法。它允许在线方直接声明最终状态，而不实际执行依赖关系图的任何附加遍历。无需交易对手签名。为避免滥用，索赔方还需要提供防欺诈保证金。经过一段挑战期后，状态将成为最终状态，无需执行任何附加操作。
* **动态抵押和提现.** GPC的一个共同要求是，当参与方未连接到网络时，实现无缝链上交易。对于资金的提取，为了满足这一要求，我们引入了两种功能：IntendWithdraw和ConfirmWithdraw。IntendToWithdraw在挑战期内修改基态。参与方可以提交冲突的sp进行裁决。如果在定义的挑战期内没有发生争议，则调用ConfirmToWithdraw确认并进行提现。这两个功能的工作非常类似于IntendSettle和ConfirmSettle。抵押很简单，因为它只修改基态*s*0.
* **布尔条件组.** 我们期望GPC最常见的用例是基于布尔的条件支付。例如，“如果函数X或函数Y返回真，则A支付B”。为了优化这种支付方式，我们调整了条件组和条件的接口。特别是，如果条件解析结果（或条件结果的布尔逻辑）中的任何一个为真，我们可以实现函数ResolveGroup来执行预定义的条件付款。这样，我们就省去了为ResolveGroup创建额外对象的麻烦，并节省了相应的多方通信开销。我们还将条件指定为布尔条件，这样我们就需要依赖对象具有一个与“isSatisfied”相同效果的接口，该接口根据查询的状态返回true或false。
* **资金分配条件组.** GPC的另一个更通用的用例是通用状态分配。我们通过引入另一种不同类型的条件组来实现这一点，其中只有一个条件。QueryResult将直接返回一个状态分配映射，指示一个Δs的更新。这将为GPC启用一个更通用的插件。可以插入一个链下合约，该合同是用一定的锁定流动性初始化的。这个合约不仅可以检查谁赢了一个游戏（布尔值），还可以检查胜利者为赢得游戏采取了多少步骤，然后通过执行一定的计算来分配流动性。参与方可以生成一个条件组，引用双方约定的链下合约的检查功能。

可以为不同的模式定义更多的通用模式，但是上面的示例说明了这种优化的设计原则。

## 备选的侧链通道模型

除了上述广义状态通道模型外，cChannel还引入了一个由侧链简化的替代状态通道模型[1]。例如，考虑多个用户需要相互支付的场景。用户可以将其抵押集中到一个中心合约中，该合约就像一个侧链合约，链下服务提供商扮演区块提议者的角色（与链下服务提供商形成“多方中心”，链下服务提供商是“中心运营商”），从而在一个中心内实现一对多的支付关系。通过参与者可接受的一定级别的防欺诈债券，确保链下服务提供商的诚实性。具体来说，在Celer Network中，每个链下服务提供商都可以运行一个侧链辅助状态通道:

*,* (5)

其中s为侧链状态，为块提议者（链下服务提供商），b为欺诈保证金，为最终超时。每个节点i都可以向通道中的其他节点发送侧链交易，以更新最新状态。与任何侧链交易一样，节点i不仅要签署此交易，还要签署另一个交易，以证明它已经看到该交易包含在创建的块中。第二个已签署的交易可以用作节点i的证明。只要参与者具有完全可用的块数据，此侧链交易的终局性也可以很快确认.

与前面提到的通道模型相比，这种侧链辅助通道模型具有以下预期好处[1]。

• **接收方不需要链上交易和保持在线.** 这是从侧链属性继承的自然好处。原因是，接收者可以在不进行任何侧链抵押的情况下，通过侧链辅助通道赎回其收到的资金。.

* **各方没有资金锁定.** 这种好处是在支付通道的背景下实现的。当采用侧链通道进行多方支付时，各方在相互支付前无需提前锁定其抵押（需要存入防欺诈保证金的区块提案人除外）.

但是，生态系统应该清楚地意识到这个通道模型的缺点，如下所示:

* **防欺诈债券仍然需要.** 对于基于侧链的通道，仍然需要直接由区块提议人或提供审计和保险服务的任何人提供防欺诈债券。应该清楚地理解，对于区块提议者（即链下服务提供商）来说，最坏情况下的流动性要求实际上是无限的。原因是，有了足够的共谋，恶意的政党可以创造无限的重复支付。
* **数据可用性问题会复杂化终局性.** 即使不涉及恶意方，侧链模型固有的终局性延迟仍然困扰着该通道模型，尤其是当数据可用性成为一个问题时。当区块数据在相关方之间并不总是可用时，侧链面临着不可避免的重新组织，因此终局最好会延迟，最坏的情况下会放弃整个侧链。

这些基于侧链的通道可以通过公共状态通道进一步相互连接。

# cRoute: Provably-Optimal Value Transfer Routing

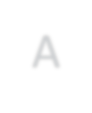
## Challenges in State Channel Network Routing

The need for state routing (or “payment routing” in the case of payment channel networks) is apparent: it is impractical to establish direct state channels between every pair of nodes due to channel opening costs and deposit liquidity lockup. Therefore, it is necessary to build a network consisting of state channels, where state transitions should be relayed in a trust-free manner. The design of state routing is crucial for the level of scalability that a state channel network can provide, i.e., how fast and how many transactions can flow on a given network. However, existing proposals all fall short to meet the fundamental challenges imposed by the unique properties of state channel networks.

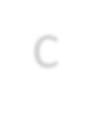
Landmark routing [16] has been proposed as one option for decentralized payment routing in several payment channel networks. For example, Lightning Network [9] adopts a landmark routing protocol called Flare [10]. A similar algorithm is also used in the decentralized IOU credit network SilentWhispers [4]. The key idea of landmark routing is to determine the shortest path from sender to receiver through an intermediate node, called a landmark, usually a well-known node with high connectivity.

Raiden Network [2] (a payment channel network) mentioned a few implementation alternatives for payment routing, such as *A*⇤ tree search which is a distributed implementation of shortest path routing. In addition, since route discovery is hard but crucial, nodes can provide pathfinding services for other nodes for some convenience fees in Raiden Network.

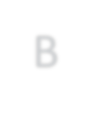
Recently proposed SpeedyMurmurs [13] enhances the previous shortest path routing algorithms (as used in Lightning Network and Raiden Network) by accounting for the available balances in each payment channel. Specifically, SpeedyMurmurs is based on



A



C



B

100

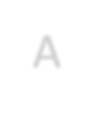
100

100

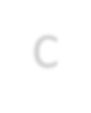
100

100

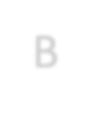
100



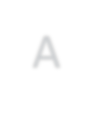
A



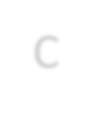
C



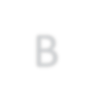
B



A



C



B

200

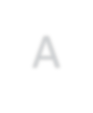
0

0

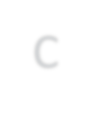
200

200

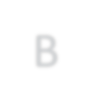
0



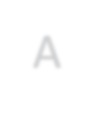
A



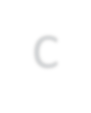
C



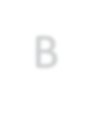
B



A



C



B

0

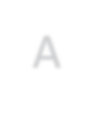
200

200

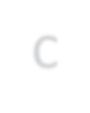
0

0

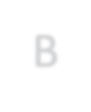
200



A



C



B

time slot 1 time slot 2 time slot 3

**Figure 2.** Shortest path routing leads to frequent topology changes due to channel imbalance.

the embedding-based routing algorithms that are commonly used in P2P networks [12], which first constructs a prefix tree and then assigns a coordinate for each node. The forwarding of each payment is based on the distances between the coordinate known to that node and the coordinate of the destination. The prefix tree and the coordinate of each node will be adjusted if there is any link that needs to be removed (i.e., when a link runs out of balance) or added (i.e., when a depleted link receives new funding).

It can be observed that all of the existing routing mechanisms boil down to “*shortest path routing with available balance consideration*”. In traditional data networks, shortest path routing does provide reasonably good throughput and delay performance, based on the assumption that network topology remains relatively stable and link capacity is “stateless” (i.e., the capacity of each link is not a↵ected by past transmissions). Unfortunately, such an assumption no longer holds for an o↵-chain state channel network due to its “stateful” link model, i.e., the capacity (available balance) of each directed link keeps changing as payments go through that link. Note that shortest path routing does not account for channel balancing, and thus each link may quickly run out of its capacity, which further leads to frequent changes in network topology. Figure 2 illustrates a scenario in which shortest path routing leads to topology changes every time slot. Suppose that at the beginning of each time slot, node *A*, node *B* and node *C* each initiates a payment of 100 tokens to node *B*, node *C* and node *A*, respectively. Under the initial channel balance distribution (time slot 1), every pair of nodes are connected by a bi-directional link, and each node selects a direct route to its destination under shortest path routing. However, this results in a uni-directional transfer over each channel, and thus the distribution of channel balances becomes highly skewed in time slot 2, where the underlying topology is a counter-clockwise cycle. In this new topology, shortest path routing continues to make uni-directional transfers (e.g., selects route *A* ! *C* ! *B* for payment *A* ! *B*), and channel balances are pushed to the other extreme, where the underlying topology is completely reversed to a clockwise cycle (time slot 3). The same pattern will repeat indefinitely. In contrast, if node *C* takes a longer route *C* ! *B* ! *A*, every channel will remain balanced all the time, and the network topology never changes. For any decentralized implementation of shortest path routing, such frequent topology changes could lead to poor performance since it takes time for the algorithm to converge on the new topology (e.g., to reconstruct the prefix tree as in SpeedyMurmurs [13]), during which sub-optimal routes may be taken. What’s even worse is that the network topology may change again before the algorithm converges, and thus the algorithm may *never converge* and achieve continually poor throughput performance.

Note that the recent project Revive [3] proposes an explicit channel rebalancing scheme. However, Revive does not account for state routing, which means that its channel rebalancing procedure is not transparent to the underlying routing process and requires extra out-of-band coordination. Moreover, Revive only works in a restricted class of network topologies that contain cyclic structures, and it does not provide any guarantee that its channel rebalancing procedure is feasible in a general topology. In comparison, we propose a routing algorithm that achieves transparent and optimal channel balancing during the routing process.

## Distributed Balanced Routing (DBR)

We propose *Distributed Balanced Routing* (DBR) as an e cient routing protocol for value transfers in an o↵-chain state channel network. The DBR algorithm is inspired by the BackPressure routing algorithm [7, 15] that was originally used in wireless networks. It is based on a completely di↵erent design philosophy from the traditional shortest path routing. In particular, DBR does not perform any explicit path computation from source to destination. Instead, the routing direction is guided by the current network’s *congestion gradients*. Think of water flowing from the top of a hill to a destination at the foot of the hill. The water does not need to know the route to its destination; all it needs to do is to follow the direction of gravity.

The DBR algorithm uses a similar design philosophy but also accounts for the stateful link model in state channel networks. In particular, the DBR algorithm is augmented with a state channel balancing ability that transparently maintains balanced transfer flows for each state channel. Compared with existing routing algorithms, the proposed DBR algorithm has the following advantages:

* **Provably optimal throughput**. In other words, for a given arrival rate of value transfer requests, if there exists any routing algorithm that “supports” the rate, DBR is also able to do that. The meaning of “support” will be specified in Section 3.2.3.
* **Transparent channel balancing**. In DBR, the channel rebalancing process is naturally embedded in the routing process without any additional coordination. It automatically rebalances each state channel to maintain balanced value transfers in the long term.
* **Fully decentralized**. The DBR algorithm is a fully decentralized algorithm where each node only needs to talk to its neighbors in the state channel network topology. DBR also has low messaging cost in the protocol.
* **Failure resilience**. The DBR algorithm is highly robust against failures: it can quickly detect and adapt to unresponsive nodes, supporting the maximum possible throughput over the remaining available nodes.
* **Privacy preserving**. Due to its multi-path nature, the DBR algorithm naturally preserves the privacy regarding the amount of transferred values, without using any additional privacy-preserving techniques (e.g., ZKSNARK). More importantly, the DBR algorithm can be seamlessly integrated with onion routing [11] to preserve anonymity for sources/destinations.

In the following, we first introduce the state channel network model, then describe the DBR algorithm, and finally prove the performance of DBR. Note that for the ease of exposition, we restrict our attention to bi-party payment channels in this section, but the same ideas apply to any state channel network that has value transfer requirements.

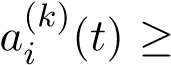
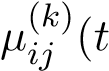
### System Model

In our model, time is discretized into slots of a fixed length, where the length of each slot usually corresponds to the physical transmission delay over one hop. Suppose that there are *N* nodes in the network. For each pair of nodes *i* and *j*, a pair of directed links *i* ! *j* and *j* ! *i* exit if there is a bi-directional payment channel *i* $ *j* between node *i* and node *j*. Let *cij*(*t*) be the capacity of link *i* ! *j* in slot *t*, which corresponds to the remaining balance in the payment channel that can be transferred from node *i* to node *j* at the beginning of that slot. There is a total deposit constraint for each bi-directional payment channel between node *i* and node *j*:

*cij*(*t*) + *cji*(*t*) = *Bi*$*j*(*t*)*,*

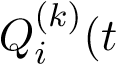
where *Bi*$*j*(*t*) is the total deposit of bi-directional payment channel *i* $ *j* at the beginning of slot *t*. Note that the total deposit *Bi*$*j*(*t*) may change over time due to dynamic 链上 fund deposit/withdrawal.

During each slot *t*, each node *i* receives new payment requests from outside the network, where the total amount of tokens that should be delivered to node *k* is

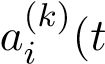
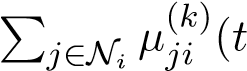
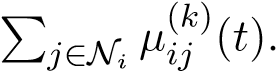
0. Also denote by) the amount of tokens (required to be delivered to node *k*) sent over link *i* ! *j* in slot *t*, which is referred to as a *routing variable*.

### Protocol Description

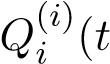
Before the description of DBR, we first introduce several important notions: debt queue, channel imbalance and congestion-plus-imbalance (CPI) weight.

**(Debt Queue).** In the operation of DBR, each node *i* needs to maintain a “debt queue” for payments destined to each node *k*, whose queue length) corresponds to the amount of tokens (with destination *k*) that should be relayed by node *i* to the next hop but have not been relayed yet at the beginning of slot *t*. Intuitively, the length of the debt queue is an indicator of congestion over each link. The queue length evolution is as follows:

*Q*(*ik*)(*t* + 1) = h*Q*(*ik*)(*t*) + *ai*(*k*)(*t*) + X*i µji*(*k*)(*t*) X *µ*(*ijk*)(*t*)i+*,* (6) *j*2N *j*2N*i*

where [*x*]+ = max{0*,x*} (since queue length cannot be negative) and N*i* is the set of neighbor nodes of node *i*. The above equation simply means that in slot *t*, the change of queue length is caused by three components: (1) new token transfer requests from outside the network (i.e., )), (2) tokens routed from neighbors to node *i*, i.e., ), and (3) tokens routed from node *i* to its neighbors, i.e.,

It should be noted that the queue length at the destination node is always zero, i.e.,

) = 0 for each node *i*, which guarantees that every packet can be eventually delivered to its destination under the DBR algorithm.

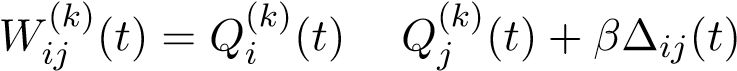
**(Channel Imbalance).** For each link *i* ! *j*, we define the channel imbalance as

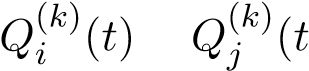
*ij*(*t*) = XX⇣*µ*(*jik*)(*⌧*) *µ*(*ijk*)(*⌧*)⌘*.* (7)

*⌧<t k*

Intuitively, *ij*(*t*) is the di↵erence between the total amount of tokens received by node *i* from node *j* and the total amount of tokens sent from *i* to *j* over their payment channel up to the beginning of slot *t*. Note that if *ij*(*t*) *<* 0 then it means that node *i* sent more tokens to node *j* than what was received from node *j*. Clearly, *ij*(*t*) is a natural measure of channel imbalance as perceived by node *i*. Our DBR algorithm tries to balance the payment channel such that lim*t*!1 *ij*(*t*)*/t* = 0 for each payment channel *i* $ *j*, which implies that the long-term sending rate from *i* to *j* equals the sending rate from *j* to *i*.

**(Congestion-Plus-Imbalance (CPI) Weight).** Define the Congestion-PlusImbalance (CPI) weight for link *i* ! *j* and destination *k* as

*,* (8)

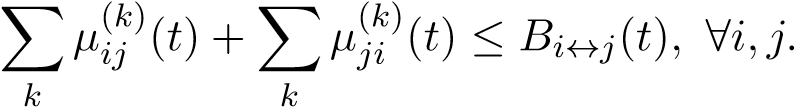
where *>* 0 is a parameter adjusting the importance of channel balancing. Intuitively, the above weight is the sum of the di↵erential backlog) for payments destined to node *k* between node *i* and node *j* (i.e., congestion gradient) and the channel imbalance *ij*(*t*) between node *i* and node *j*. The former is used to reduce network congestion and improve network throughput while the latter is used to balance payment channels.

|  |
| --- |
| **Distributed Balanced Routing (DBR)**  The following protocol is locally executed by each node *i*.  In each time slot *t*, node *i* first exchanges the queue length information with |
| its neighbors and calculates the CPI weights. Then for each link *i* ! *j*, node *i* calculates the best payment flow to transmit over that link:  *k*⇤ = argmax*.* (9) *k*  If 0, then ) otherwise ) = 0. For any *k* =6 *k*⇤, set |

**Remark.** In each slot *t*, DBR essentially tries to solve the following weighted-sum optimization problem:

max XX*µ*(*ijk*)(*t*)*Wij*(*k*)(*t*)

*ij k*

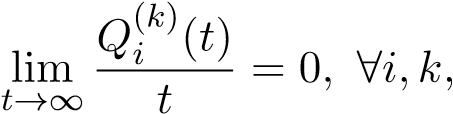
(10) s.t. 

The above optimization problem is also called MaxWeight and derived from our theoretical analysis of DBR (see the next section). The aforementioned algorithm description gives an approximate solution to MaxWeight.

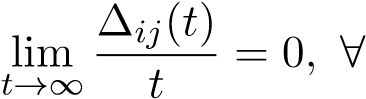
### Throughput Performance of DBR

To analyze the throughput performance of DBR, we first introduce a few definitions.

* A state channel network is said to be **stable** if

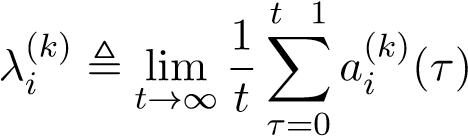


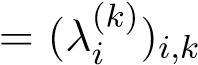
which implies that the long-term arrival rate to each debt queue equals the long-term departure rate from that queue. • A state channel network is said to **balanced** if

 channel *i* $ *j.*

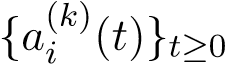
In other words, for each payment channel *i* $ *j* the long-term sending rate from node *i* to node *j* equals the sending rate from node *j* to node *i*.

* Define



as the long-term average arrival rate to node *i* for payments with destination *k*. An arrival rate vector is said to be **supportable** if there exists a routing algorithm that can keep the network stable and balanced under this arrival rate vector.

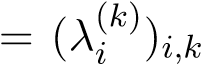
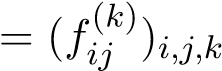
* The **throughput region** of a state channel network is the set of supportable arrival rate vectors.
* A routing algorithm is **throughput-optimal** if it can support any payment arrival rate vector inside the throughput region.

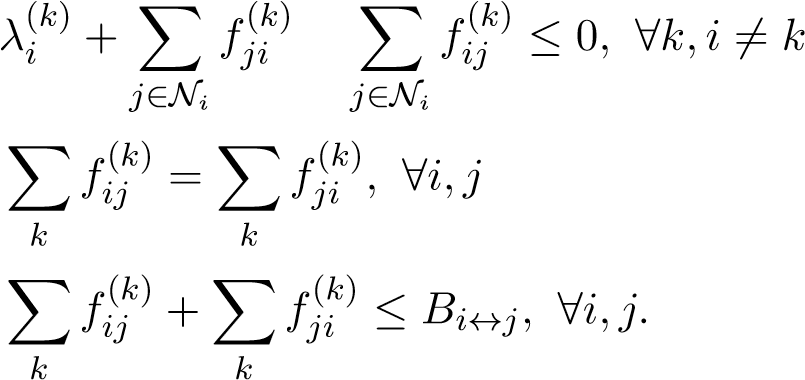
For the ease of exposition, we assume that the external payment arrival process  is stationary and has a steady-state distribution, and that the total de-

posit for each payment channel remains fixed, i.e., *Bi*$*j*(*t*) = *Bi*$*j* for any *t* 0. Our analysis can be extended to the case where the arrival process is non-stationary and channel deposits are time-varying (e.g., dynamic 链上 deposit/withdraw), at the expense of unwieldy notations. The following theorem shows the throughput performance of DBR.

**Theorem 3.1.** *The DBR algorithm is throughput-optimal.*

In other words, as long as there exists a routing algorithm that can keep the payment network stable and balanced, the DBR algorithm can also achieve that. The rest of §3.2.3 in below is the proof of Theorem 3.1. We first introduce a lemma which characterizes the throughput region for a state channel network.

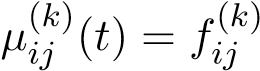
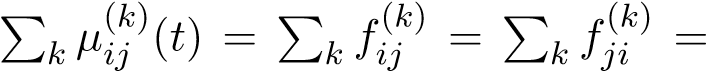
**Lemma 3.2.** *An arrival rate vector*  *is supportable if and only if there exist flow variables f* *that satisfy the following conditions:*

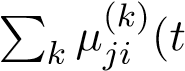
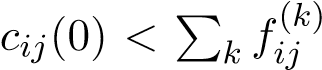
(11)

(12)

(13)

*Proof.* The necessity of the above conditions is trivial. Inequality (11) corresponds to the flow conservation requirement. If it is violated, then the arrival rate to node *i* is larger than the departure rate, and the state channel network is unstable. Equation (12) corresponds to the channel balance requirement. If it is violated, then channel *i* $ *j* is imbalanced. Inequality (13) corresponds to the channel capacity constraint, since the sum of tokens transferred over each channel *i* $ *j* cannot exceed the total channel deposit *Bi*$*j*.

In order to prove the su ciency of the above conditions, we construct an algorithm that can stabilize and balance the state channel network when the arrival rate vector satisfies (11)-(13). The algorithm is straightforward: in each slot *t*, set the routing variable for any *i,j,k*. Clearly, under this routing algorithm every channel *i* $ *j* remains balanced in each slot *t* since

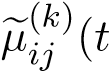
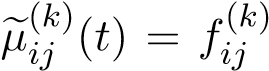
). Moreover, the network is stable under the algorithm since the flow conservation requirement is satisfied for every node. Note that each link *i* ! *j* may have insu cient fund initially (i.e., ) such that the routing decision is infeasible. In this case, we can let node *j* transfer some tokens to node *i* at the beginning in order to equalize the fund at both ends of the state channel. Such an adjustment process incurs at most *Bi*$*j* sub-optimal transfers for each channel *i* $ *j* and does not influence network stability and channel balance in the long term.

Therefore, equations (11)-(13) are a necessary and su cient condition for an arrival rate vector to be supportable.

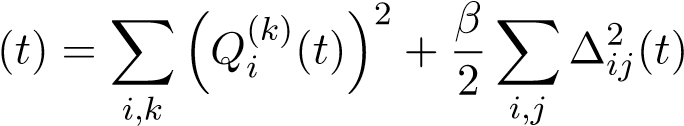
It should be noted that the routing algorithm mentioned in the proof of Lemma 3.2 cannot be implemented in practice since we do not know the external arrival rate vector in advance. In the following, we prove that DBR can achieve the same throughput performance without knowing any payment tra c statistics in advance.

By Lemma 3.2, if an arrival rate vector belongs to the throughput region, it must satisfy (11)-(13) and can be supported by the algorithm mentioned in the proof of

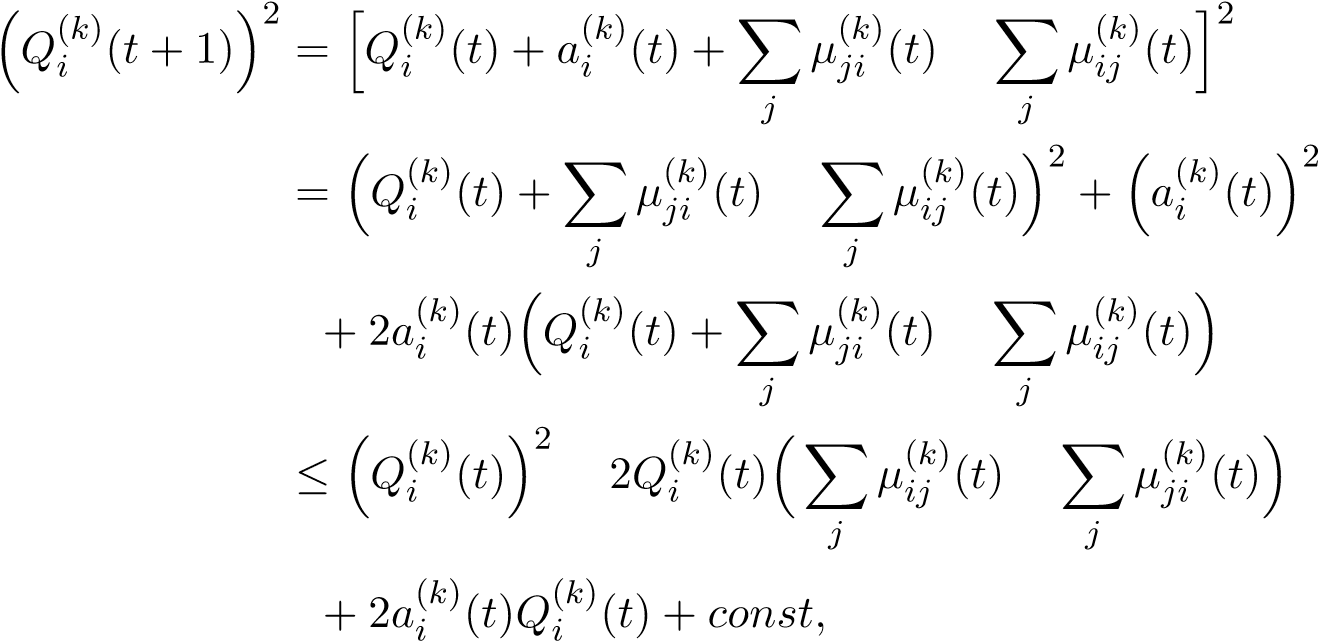
Lemma 3.2 (referred to as the *optimal oracle algorithm*). In the following, denote by

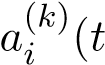
) the routing decision made by the optimal oracle algorithm in slot *t*. By the nature of the optimal oracle algorithm, we have for any *t* (if ignoring the initial fund adjustment process).

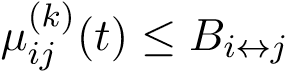
Define the *Lyapunov function* as follows:

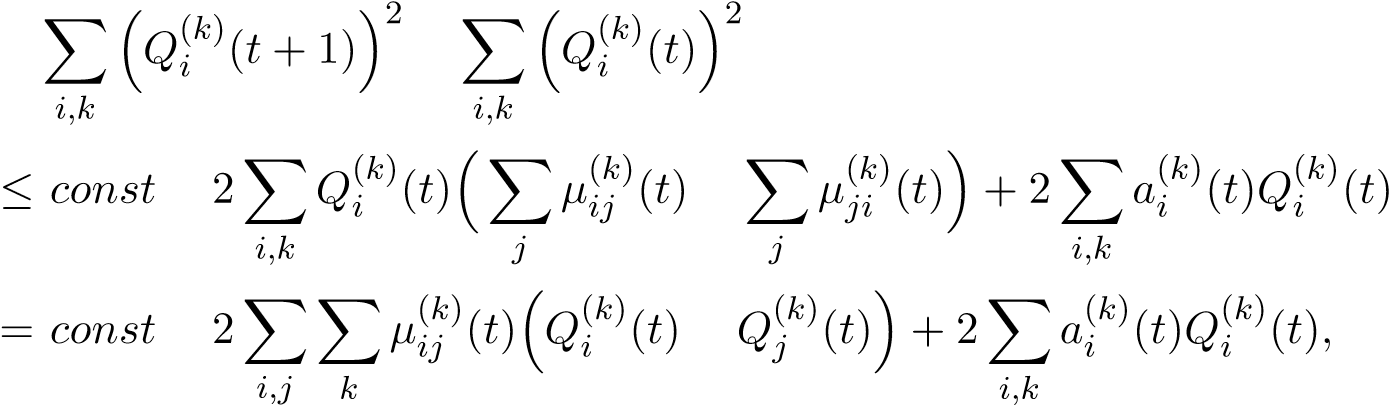
*.* (14)

Also define the *conditional Lyapunov drift* as *D*(*t*) ,E[ (*t* + 1) (*t*)|**Q**(*t*)*,* (*t*)], where the expectation is with respect to the randomness of arrivals. To facilitate the analysis, we assume that the amount of new payments arrivals to the network in each slot is bounded by some constant. By equation (6), we have

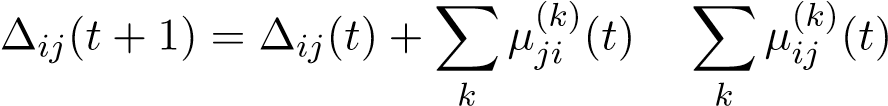
 (15)

where the inequality is due to the assumption that the arrival) in each slot *t* is bounded by some constant and the fact that the number of transferred tokens in each

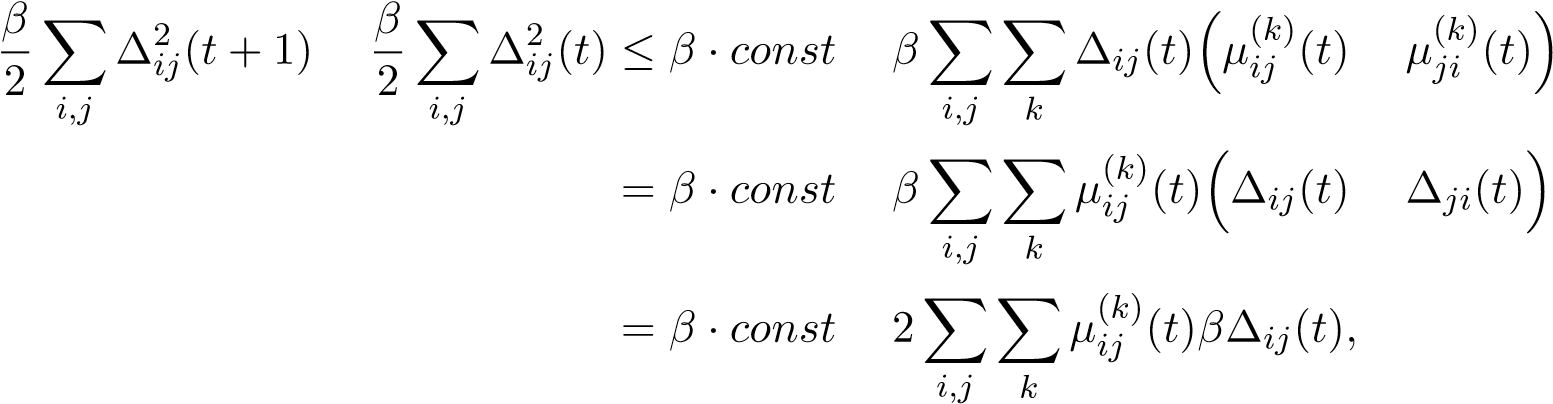
slot is also bounded (since). Now we have



where we rearrange the sum in the above equality. Similarly, noticing that

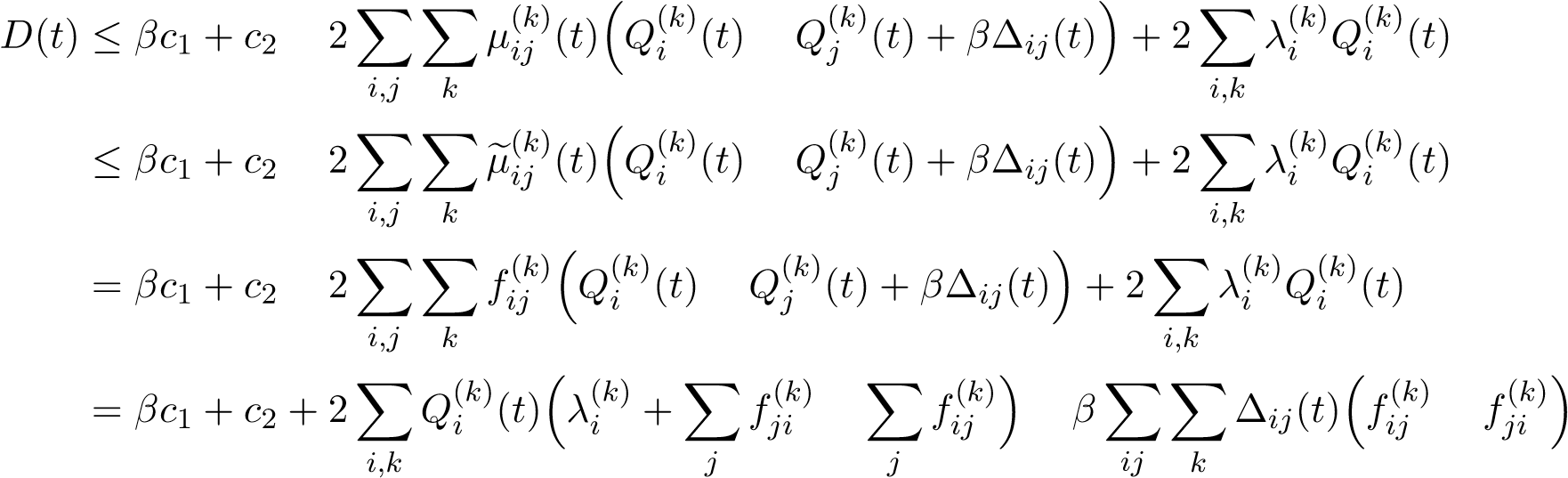
*,*

we can prove

*,*

(16)

where we use the fact that *ij*(*t*) = *ji*(*t*). As a result, by combining (15) and (16), the conditional Lyapunov drift can be bounded by



 *c*1 + *c*2*,*

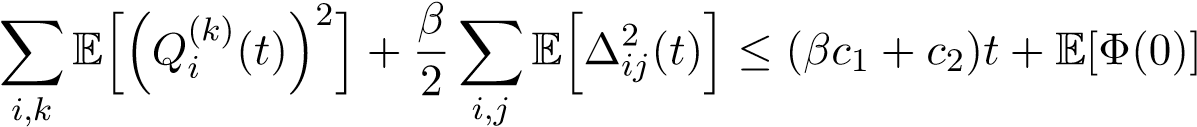
where *c*1 and *c*2 are some constants, the second inequality is due to the operation of DBR (see (10)), and the last inequality is due to (11) and (12). Using the law of iterated expectations yields:

E[ (*⌧* + 1)] E[ (*⌧*)]  *c*1 + *c*2*.*

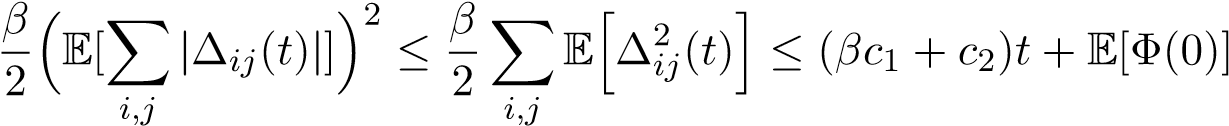
Summing over *⌧* = 0*,*··· *,t* 1, we have

E[ (*t*)] E[ (0)]  ( *c*1 + *c*2) · *t.*

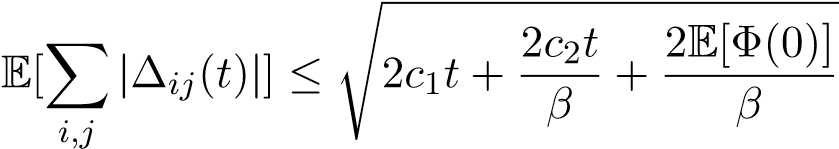
Then we have

*.* (17)

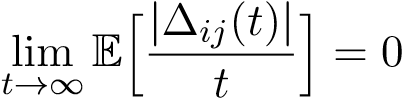
To show that DBR achieves channel balance, we note from (17) that

*,*

where the first inequality holds because the variance of2 | (*t*)| cannot be negative, i.e., Var(| (*t*)|) = EhP*i,j* 2*ij*(*t*)i ⇣E[P*i,j* | *ij*(*t*)|]⌘ 0. Thus we have

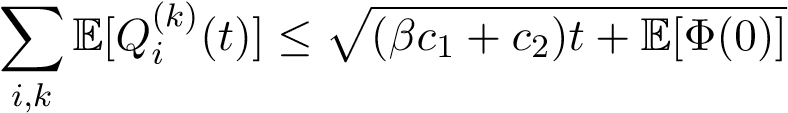
*.*

Since E[ (0)] *<* 1, we have that for any payment channel *i* $ *j*

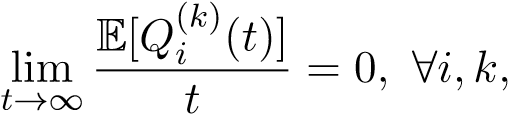
*,*

i.e., the network maintains channel balance under the DBR algorithm.

Similarly, we can show that

*,*

which implies that



i.e., the network is stable under the DBR algorithm.

## Discussions of DBR

### Failure Resilience

Due to its adaptive and multi-path nature, the DBR algorithm is inherently robust against network failures. For example, when there are unresponsive nodes, DBR can quickly adapt and support the maximum possible throughput over the remaining available nodes.

### Privacy

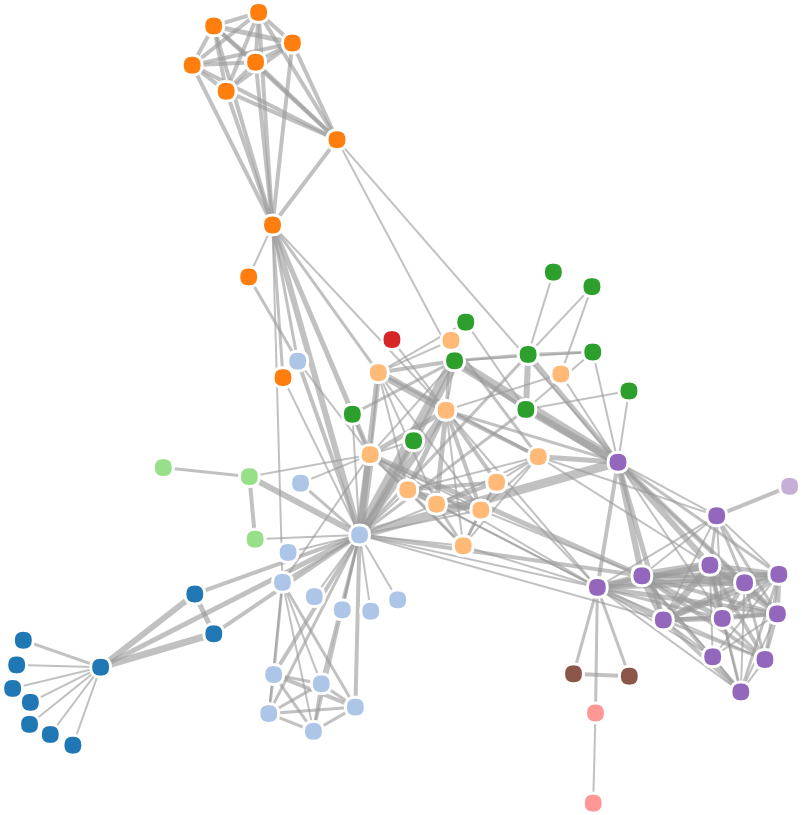
Due to the multi-path nature of DBR, any intermediate node can only access the information for a small fraction of each value transfer request. As a result, the DBR algorithm naturally provides good privacy protection in terms of the amount of transferred values. However, in DBR, each node does need to know the destination of each value transfer request in order to place it in a proper debt queue. If we also need to hide the payment destination, onion routing [11] can be used in conjunction with DBR. In onion routing, messages are encapsulated in layers of encryption. The encrypted data is transmitted through a series of network nodes called *onion nodes*, each of which “peels” away a single layer, uncovering the data’s next destination. When the final layer is decrypted, the message arrives at its destination. We can direct payments through an overlay network consisting of onion nodes and apply DBR to optimally route payments among onion nodes.

## Simulation Results

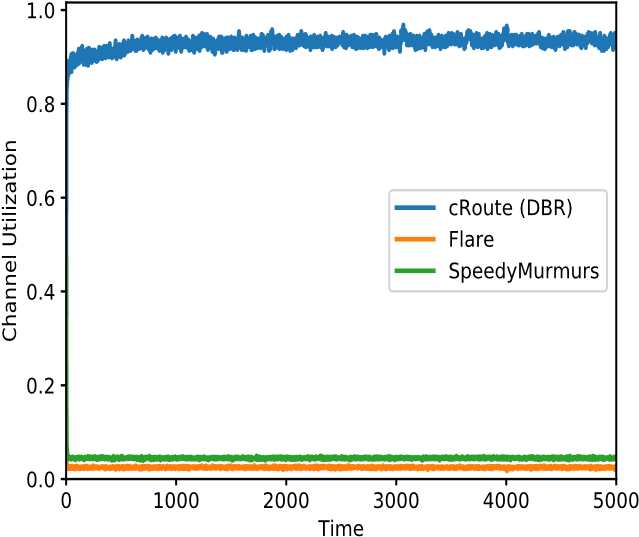
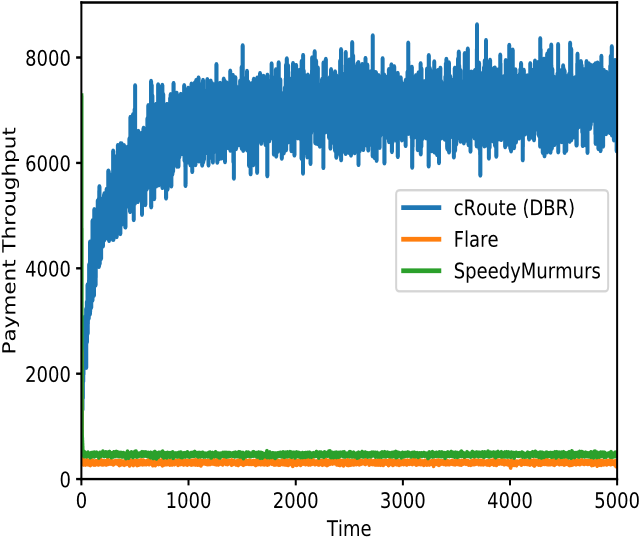
We numerically compare the performance of DBR with two existing routing algorithms in payment state channel networks: SpeedyMurmurs [13] and Flare [10] (used in Lightning Network). The simulation is conducted on the topology given in Figure

3.

Figure 4 shows the throughput performance comparison. It is observed that DBR achieves 15x improvement in average payment throughput as compared to the existing routing algorithms. The exceptional throughput performance of DBR is due to its channel-balancing and congestion-aware nature. In particular, Figure 5 illustrates the channel utilization[[1]](#footnote-1) under DBR, SpeedyMurmurs and Flare, where we can observe



**Figure 3.** Payment channel network topology used in simulations (77 nodes, 254 bi-directional payment channels). The payment channel network is running with 40 payment flows with randomly chosen source-destination pairs. The initial deposit for each channel is uniformly distributed within [100,200] tokens. Payment arrivals follow a Poisson process and the size of each payment follows a geometric distribution with the mean of 3 tokens.



**Figure 4.** Instant Payment throughput **Figure 5.** Channel utilization comparison comparison among DBR (avg: 6748 pay- among DBR, SpeedyMurmurs and Flare. ments/slot), SpeedyMurmurs (avg: 467 Higher channel utilization implies a higher payments/slot) and Flare (avg: 316 pay- level of channel balancing. ments/slot).

that DBR consistently achieves high (nearly 100%) channel utilization while the other routing algorithms only achieve less than 5% channel utilization due to the lack of channel balancing.

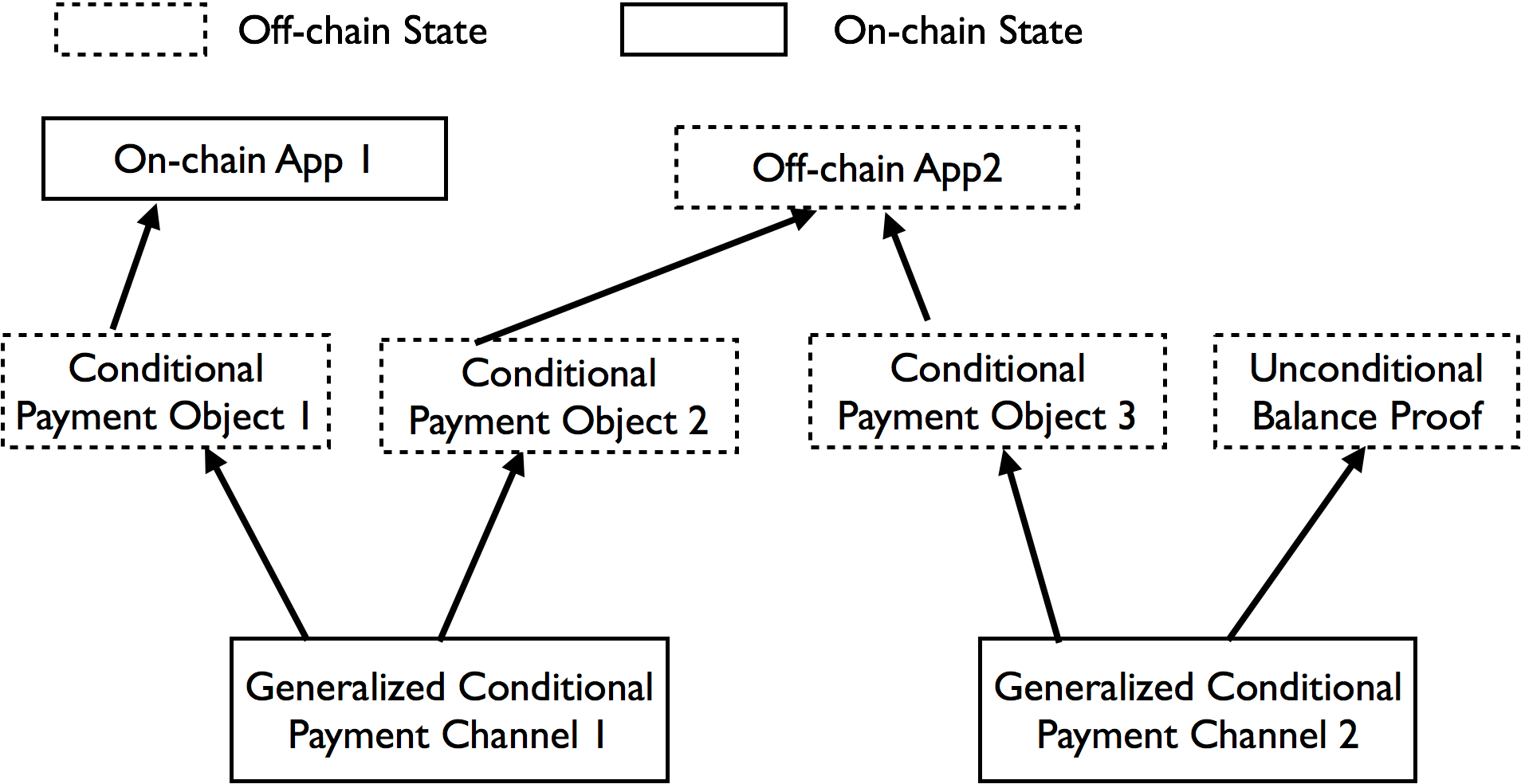
the total token deposits of all channels. For example, if the total amount of deposits is 100 and only 50 of them are moved in slot *t*, then the overall channel utilization in that slot is 50%.

# cOS: O↵-chain Decentralized Application Operating System

To help everyone quickly build, operate, and use scalable o↵-chain decentralized applications without being hassled by the additional complexities introduced by o↵-chain scaling, Celer Network innovates on a higher level abstraction: cOS, a combination of application development framework (SDK) and runtime system. This section provides the high-level vision, design objectives and illustrations on cOS.

## Directed Acyclic Graph of Conditionally Dependent States

In this section, we provide a view of our abstraction model on the construct of o↵-chain applications, and describe how the model is integrated with state channel networks. In order to support use cases beyond simple P2P payments, we model a system of o↵-chain applications as a directed acyclic graph (DAG) of conditionally dependent states, where the edges represent the dependencies among them.



**Figure 6.** DAG of Conditionally Dependent States

Figure 6 illustrates the system model, where the Generalized Conditional Payment Channel in the payment networks are the only contracts with 链上 states. The settlement of these 链上 states depends on one or more conditional payment objects (e.g. Conditional Payment Object 3), which are completely o↵-chain but 链上 enforceable. We want to highlight that these conditional payment objects are not only simple time hash locked transactions, but can be conditioned on o↵-chain application contract states, such as “O↵-chain App X” in Figure 6.

The conditional payment objects can be relayed through multiple hops just like simple unconditional payments objects. For example, Payment Channel 1 can be a channel connecting Alice and Bob and Payment Channel 2 can be a channel connecting Bob and Carl. Let “O↵-chain App 2” be an o↵-chain chess game Alice is playing with Carl, and suppose Alice wants to express the semantic of “Alice will pay Carl 10 ETH if Carl wins the game”. Even without a direct channel between Alice and Carl, Alice can send a conditional payment to Carl through Bob with two layers of conditional locks. The first layer is a simple time hashed lock to make sure that Bob relays and resolves the payment in a reasonable amount of time. The second layer locks the payment conditioning on the result of the chess game. With this two-hop relay, the conditional payment between Alice and Carl can be settled via Bob even though Bob did not involve in the chess game. This is a minimized example of how a dependency graph formed by generalized state channels, conditional payment objects and o↵-chain applications can support arbitrarily complex multi-party interactions.

Note that o↵-chain objects do not have to depend only on o↵-chain objects. For example, Alice can pay Carl when the latter successfully transfers a certain ENS name to the former. In other words, the payment depends on the o↵-chain condition that the owner of the ENS name changes from Carl to Alice.

Also, o↵-chain payment objects do not always have to be conditional: a conditional payment object can “degenerate” into an unconditional balance proof as the application runs. More generally speaking, conditional dependencies are transient by nature: application state updates are done via a pair of two topological traversals of the underlying state graph. The first traversal goes in the forward direction and the second one in the reverse direction. The forward traversal, starting from the 链上 state-channel contracts, creates additional transient conditional dependency edges and modifies existing ones. The reverse traversal may remove existing transient conditional dependency edges, because some conditions evaluate to constant true when traversing backward.

## O↵-chain Application Development Framework

Much like how modern high-level languages and operating systems abstract away the details about the underlying hardware, the complexity of interacting with conditional state dependency graphs necessitates a dedicated development framework. With the principle of “ease of use” on our mind, Celer Network presents the cOS SDK, a complete toolchain solution for the creation, tracking, and resolution of states in o↵-chain applications. We hope that the SDK will accelerate the adoption of the o↵-chain scaling solution and the payment network provided by Celer Network, fostering a strong ecosystem.

cOS API

Smart contract

Platform-specific code

State dependency graph

cOS state compiler

cApp

**Figure 7.** Structure of a decentralized application on Celer Network (cApp)

In general, we categorize decentralized applications into two classes: simple payper-use applications and more complex multi-party applications. The pay-per-use applications include examples like the Orchid Protocol, where the user keeps receiving microservices (e.g. data relay) from a real-world entity and streams payments through the payment network. Since there is no need for conditional dependency on other o↵chain states, a lean transport layer API on top of the routing layer, both of which are provided by Celer Network, su ce for such cases.

The class of multi-party applications, the general structure of which is illustrated in Figure 7, is where the idea of conditional state dependency graphs really shines. The SDK defines a set of design patterns and a common framework for developers to express the conditional dependencies. We plan to extend the existing smart contract languages with modern software construction techniques such as metaprogramming, annotation processing, and dependency injection so that the dependency information can be written out explicitly without being too intrusive. A compiler then processes the application code, extracts the declared o↵-chain objects, and generates the conditional dependency graph. The compiler detects invalid or unfulfillable dependency information and generates human-readable errors to assist the developer in debugging. To help developers reason about the dependencies even further, the SDK will be able to serialize the graphs into common formats such as Graphviz, with which they can be easily visualized and presented.

The SDK also provides a code generator that generates a set of “bridge methods” for interacting with smart contracts whose code is available at compile time. The code generator parses the application binary interface (ABI), which specifies the signature of all callable functions in a smart contract, and generates the corresponding bridge methods in platform-specific languages such as Java. The main advantage of this approach is type safety: the glue methods replicate the method signatures of the functions in the smart contract faithfully, providing a static and robust compile-time check before dispatching the method to the cOS runtime for execution.

## O↵-chain Application Runtime

The cOS runtime serves as the interface between cApps[[2]](#footnote-2) and the Celer Network transport layer. It supports cApps in terms of both network communication and local o↵chain state management. The overall architecture is illustrated in Figure 8.

Smart contract VM

VM-native bridge

Local storage

cApp

State guardian network

Blockchain

cApp

cApp

cChannel

Communicate

Persist

Dispute

Handoff

cOS

**Figure 8.** cOS Runtime Architecture

On the network front, the runtime handles multi-party communication during the lifecycle of a cApp. It also provides a set of primitives for secure multi-party computation capable of supporting complex use cases such as gaming. In the case of counter-party failure, whether fail-stop or Byzantine, the runtime relays disputes to the 链上 state. In the case of the client going o✏ine, the runtime handles availability o✏oading to the State Guardian Network. When the client comes back online, the runtime synchronizes the local states with the State Guardian Network.

For local o↵-chain state management, the conditional state graphs synthesized by the cOS SDK is bundled within the cApp and passed on to the runtime for o↵-chain execution. The runtime serves as the infrastructure to create, update, store and monitor o↵-chain states locally on Celer Network clients. It tracks the internal logic of the applications running on top of it and performs the DAG traversal of state updates as outlined in § 4.1. It also gracefully handles payment reliability issues such as insu cient capacity for routing the payment.

At its core, the cOS runtime bundles a native virtual machine (VM) for running smart contracts. While we intend to deploy cOS to many platforms including desktop, web, mobile and IoT devices, we have adopted the ambitious design principle of “write once, run anywhere”. In other words, we enable developers to write the common business logic once and run the exact same 链上 smart contract code in every environment as opposed to having to implement multiple variants of the same logic. By adopting this principle, we aim to eliminate code duplication and ensure high degree of consistency across various platforms.

The platform-specific part of cApps, such as user interface (UI), can be built in languages most suitable to each platform (eg. Kotlin for Android and Swift for iOS). The UI code is also free to use platform-specific utilities and libraries, so that the look and feel of cApps match the respective design guidelines on each platform.

The cOS runtime provides VM-native bridge implementations in di↵erent languages for the platform-specific code to interact with the underlying business logic. For example, consider a cApp representing a chess game running on iOS with the UI written in Swift and business logic written in Solidity. Naturally, the UI layer will need to query the cOS VM for the state of the game board, and it will be able to do so via the Solidity-Swift bridge. Because the code for the contract is available at compile time, the code generator cOS SDK would have generated a bridge method named *chess.getBoardState*, which is dispatched to the VM for the actual query. Whenever possible, we make use of the language’s foreign function interface (eg. JNI) to reduce the overhead of calling back-and-forth between smart contracts and the native code. The developer will also be able to use the same debugging and profiling tools for 链上 smart contracts in the o↵-chain development scenario.

In order to genuinely replicate the state changes that would have happened onchain in the o↵-chain environment, the VM progresses through the same bytecode as if they were executed 链上, with the caveat of a few di↵erences. The first major di↵erence is that the VM needs to update and store the states locally instead of on the blockchain. To achieve seamless and transparent inter-operation between the VM and the rest of cOS, we will implement a set of APIs that bridge platform-specific storage backends with the VM. The second major di↵erence is that as opposed to being always online, a local VM can shut down unexpectedly at any time due to software bugs, hardware failure or simply loss of power. To avoid corruption of local states, we need to implement a robust logging, checkpointing and committing protocol. A third minor di↵erence is that the logic for gas metering can be omitted, because the execution happens locally and it does not make sense to charge gas fees.

The bundled VM needs to be lightweight and performant so that it can run well on mobile and IoT devices, which tend to operate under tight processor power, memory capacity and battery life constraints. While we currently embed a lightweight Ethereum VM in cOS, we are researching into adopting more common bytecode formats (eg. WebAssembly) with the goal of supporting more contract languages and other blockchains.

In our ultimate vision of the cOS VM, we will apply modern VM techniques such as ahead-of-time compilation (AOT) and just-in-time compilation (JIT) to achieve nearnative performance of o↵-chain smart contract execution. Instead of interpreting the smart contract bytecodes like what most Ethereum VMs currently do, we compile the bytecodes to lower level intermediate representations that are closer to native code. If the code for a certain contract is available at compile time (eg. a contract that is already deployed 链上), we perform the compilation ahead of time and statically link the binary with the rest of the application. For the contracts that are dynamically loaded at runtime, we profile them for frequently-called functions (i.e. “hot” code) and perform just-in-time compilation. We believe that the combination of these two techniques will bring a great balance between performance and energy consumption, which are both crucial for mobile and IoT devices.

# cEconomy: O↵-chain Cryptoeconomics Mechanism Design

The native digital cryptographically-secured protocol token of the Celer Network, (CELR) is a major component of the ecosystem on the Celer Network, and is designed to be used solely on the network. CELR is a non-refundable functional utility token which will be used as the platform currency in the ecosystem on the Celer Network. CELR does not in any way represent any shareholding, participation, right, title, or interest in the Token Vendor, the Foundation, their a liates, or any other company, enterprise or undertaking, nor will CELR entitle token holders to any promise of fees, revenue, profits or investment returns, and are not intended to constitute securities in Singapore or any relevant jurisdiction. CELR may only be utilized on the Celer Network, and ownership of CELR carries no rights, express or implied, other than the right to use CELR as a means to enable usage of and interaction with the Celer Network.

In the following, we introduce Celer Network’s cryptoeconomics mechanisms, cEconomy, whose design is based on the principle that a good cryptoeconomics model (token model) should provide additional values and enable new game-theoretical dynamics that are otherwise impossible. In the following, we first elaborate the fundamental tradeo↵s in o↵-chain ecosystems (Section 5.1) and then demonstrate how cEconomy can bring value and enable new dynamics to “balance out” those tradeo↵s (Section

5.2).

## Tradeo↵s in O↵-chain Ecosystems

Any o↵-chain solution, while gaining scalability, is also making tradeo↵s. In the following, we describe two fundamental tradeo↵s in o↵-chain ecosystems: scalability-liquidity tradeo↵s and scalability-availability tradeo↵s.

### O↵-Chain Scalability vs. Liquidity

O↵-chain platform gains scalability by first trading o↵ network liquidity. For example, in a bi-party payment state channel, the two involved parties can safely send each other payments at high speeds without hitting the underlying blockchain because they have deposited liquidity to the 链上 bond contract at the beginning. Liquidity-locking of this nature works is fine for the end users because the end users can simply deposit their own liquidity to the open channels and enjoy the scalable dApps. However, it poses a significant challenge for those who want to operate as an O↵-chain Service Providers (OSPs). Using state channels as an example, OSPs need to make deposits in each channel with outgoing payment possibility. Those deposits can easily aggregate to an astronomical amount. Even though Celer Networks sidechain channels can significantly reduce the level of liquidity requirement, each block proposer still needs to deposit fraud-proof bonds proportional to the level of value transfer “at stake”.

All in all, significant amount of liquidity is needed to provide e↵ective o↵-chain services for global blockchain users. However, whales may not have the business interest or technical capability to run an o↵-chain service infrastructure, while people who have the technical capability of running a reliable and scalable o↵-chain service often do not have enough capital for channel deposits or fraud-proof bonds. Such a mismatch creates a huge hurdle for the mass adoption and technical evolution of o↵-chain platforms. If not mitigated, eventually only the rich can serve as OSPs. This high capital barrier of becoming an OSP will result in a centralized network that providing undermines the entire premise of blockchain’s decentralization vision. From a more practical view, censorship, poor service quality and privacy breach will hurt users as today’s centralized services do.

### O↵-Chain Scalability vs. Availability

While an o↵-chain platform improves scalability by bringing application states o↵chain, it imposes an impractical “always online” responsibility on the users, because the o↵-chain states should always be available for 链上 disputes. For example, in a biparty payment state channel, if one party goes o✏ine, the counterparty may get hacked or act maliciously, and try to settle an old but more favorable state for himself. The data availability issue is even more critical in a sidechain channel where block proposers need to be independently monitored and validated while the participants are o✏ine; this is a matter of security and should be scrutinized carefully. This challenge is even more critical in machine to machine communication scenarios where IoT devices are not likely to be online all the time. Therefore, it is crucial to design proper mechanisms that guarantee data availability in an o↵-chain platform. Solving this challenge requires systematic thinking of the entire o↵-chain ecosystem and existing solutions all fail to provide the important properties of decentralization, e ciency, simplicity, flexibility, and security as we will discuss more in the following section.



**Figure 9.** Relationship among cEconomy components.

## cEconomy Design

To balance the above-mentioned tradeo↵s, we propose a suite of cryptoeconomics mechanisms called cEconomy that includes three tightly interconnected components: Proof of Liquidity Commitment (PoLC) mining, Liquidity Backing Auction (LiBA) and State Guardian Network (SGN). The relationship among the three components is illustrated in Figure 9.

Before moving on to the details of these components, we first introduce several terms that will be used throughout this section. Specifically, a user in our cEconomy system may play any of the three roles: O↵-chain Service Provider (OSP), End Users (EU), Network Liquidity Backer (NLB) and State Guardians (SG). O↵-chain Service Providers (OSP) are entities who have the technical capability to run highly redundant, scalable and secure o↵-chain infrastructures. End Users (EU) can access the o↵-chain services provided by OSP (e.g., pay and receive cryptocurrency). They can be common consumers or they can be IoT devices, VPN providers, live video streaming providers and CDN providers, counterparties in Machine to Machine (M2M) systems, or even an o↵-chain/链上 smart contract. Network Liquidity Backers (NLB) are entities that lock up their liquidity in the system to support the operations of o↵-chain infrastructure. State Guardians are those who provide EUs decentralized, secure, flexible and e cient state guarding service through State Guardian Network.

### Proof of Liquidity Commitment (PoLC) Mining

Our first goal is to balance out the scalability-liquidity tradeo↵ by lowering the liquidity barrier for technically capable parties to become o↵-chain service providers and thus creating an e cient and competitive market for good and reliable o↵-chain services. The gist of the idea is to enable service providers to tap into large amounts of liquidity whenever they need to. The first part to realize this idea is to provision an abundant and stable liquidity pool that can smooth out short-term liquidity supply fluctuation. To that end, we propose the Proof of Liquidity Commitment (PoLC) virtual mining process.

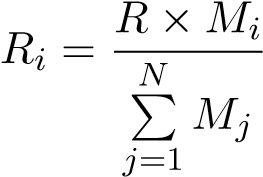
From a high level, the PoLC mining process is to incentivize Network Liquidity Backers (NLB) to lock in their liquidity (which can be in the form of digital assets, including but not limited to cryptocurrencies and CELR) in Celer Network for a long time by rewarding them with CELR tokens and therefore establishing a stable and abundant liquidity pool.

More specifically, the mining process involves NLBs to commit (lock) their idle liquidity (for example, ETH) to a “dumb box”, called Collateral Commitment Contract (CCC), for a certain period of time. During this period of time when the digital assets are locked, the NLB’s assets can only be used in the liquidity backing process and nothing else. More formally, the PoLC mining process can be defined as the following.

**Definition 5.1** (**PoLC Power**)**.** If NLB *i* locks *Si* amount of local cryptocurrency in a blockchain (e.g. ETH) for *Ti* time, its PoLC power *Mi* is computed as

*Mi* = *Si* ⇥ *Ti.* (18)

**Definition 5.2** (**PoLC Incentive Mechanism**)**.** For a limited period of time, Celer Network intends to provide incentives in the form of CELR to NLBs who lock their CCC as a show of support for the system. Incentives will be distributed proportional to each NLB’s PoLC power. Let *Ri* denote the incentives of *i*, one has:

*,* (19)

where *R* is the total reward for the current block.

Note that locking liquidity in CCC does not carry any inherent counterparty risk as it simply shows a liquidity commitment to Celer Network. Also, note that early unlocking of CCC is not allowed. One may try to create a “spoofed liquidation” with an appearance of one’s CCC getting liquefied due to “hacking” of a faked OSP. To prevent this spoofing, the newly mined CELR is not available for withdrawal and usage until CCC unlocks. Any early liquidation will cause the already mined CCC to be forfeited and redistributed to other miners. The construct of a common denominator of liquidity in PoLC is also an important question. For the initial launch of the platform, we will use the native currency of the target blockchain and later use more heterogeneous crypto assets through external price oracles.

With these mechanisms in place, the PoLC mining process ensures that the PoLC power in the system will grow as the system and utility of the CELR grows, forming a positive loop.

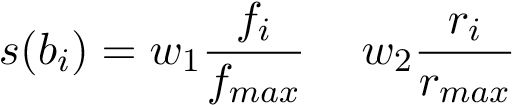
At this point, one may wonder why is CELR valuable so that it can act as such an incentive? We explain that in the following sections describing Liquidity Backing Auction and State Guardian Networks.

### Liquidity Backing Auction (LiBA)

The second part for solving the liquidity puzzle is to enable a way for o↵-chain service providers to access to liquidity pool globally, which is achieved via the Liquidity Backing Auction (LiBA). LiBA enables o↵-chain service providers to solicit liquidity through “crowd lending”. In essence, an o↵-chain service provider starts a LiBA on Celer Network to “borrow” a certain amount of liquidity for a certain amount of time. An interested liquidity backer can submit a bid that contains the interest rate to be o↵ered, amount of liquidity and the amount of CELR that she is willing to stake for the said period of time. The amount of liquidity can be submitted via a CCC. That is, CCC has the functionality to act as a liquidity backing asset. The borrowed liquidity will be used as a fraud-proof bond or outgoing channel deposit.

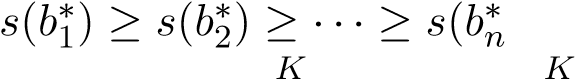
LiBA is a generalized multi-attribute Vickrey-Clarke-Groves (sealed-bid secondscore) auction. To start an auction process, an OSP creates a standard LiBA contract through the Celer Network’s central LiBA registry with information regarding the total amount of requested liquidity (*q*), duration of the request (*d*) and the highest interest rate (*rmax*) that it can accept. NLBs who watch the registry will notice this new LiBA contract and can start the bidding process. Celer Network requires all crypto assets to be locked in CCC for the bidding process. Note that CCC can be “lock-free” and simply used as a backing asset without the functionality of PoLC mining. CCC acts as a container for crypto assets and provides a unified verifiable value of heterogeneous crypto assets. Moreover, the use of CCC makes it easier for NLBs to participate in LiBA without moving crypto assets around every time they bid and thus simplifies the backing process and improve security. NLB *i* submits the bid in the form of a tuple *bi* = (*ri,ti,ci*), where *ri* is interest rate, *ti* is the total amount of CELR it is willing to lock up during the contract time and *ci* is the aggregate currency value contained in the set of CCCs bonded with this bid. Once the bid is submitted, the corresponding CCCs are temporarily frozen. After sealed bidding, the LiBA contract uses reverse second-score auction [8] to determine winning bids with the following three steps.

* (**Scoring Rule**). For each bid *bi* = (*ri,ti,ci*) in the bid set B = {*b*1*,b*2*,...,bn*} with , its score *si* is calculated as the following:

*,* (20)

where *fmax* = max{*f*1*,f*2*,...,fn*} and *rmax* = max{*r*1*,r*2*,...,rn*}. *w*1 and *w*2 are weights for the two components and are initially to ensure we take into account an interest rate with higher weight and then take into account the amount of staked CELR. [[3]](#footnote-3).

* (**Winner Determination**). To determine who has the opportunity to become the network liquidity backer, the LiBA contract sorts the bids in B in descending order by their scores. The sorted bid set is denoted by , where

) (ties are broken randomly). Winners are the first *K* bids P P1

B

in⇤, where *ti q* and *ti < q*. *i*=1 *i*=1

* (**Second-Score CELR Staking/Consumption**). After winners are determined, their CCCs will be locked in the LiBA contract for time *d* (the duration of the request), their interest requests are accepted and interests are prepaid by the OSP initiating the liquidity request. However, it is important to note that not all of their committed CELR are locked/consumed in this contract. Each winner only needs to lock up/consume enough CELR so its score matches the score of the first loser in this auction. Whether the token will be locked or consumed depends on the stage of the platform. In the first five years, new tokens will be generated through PoLC mining and LiBA only requires token staking. When the PoLC mining concludes, LiBA will start to consume token and the consumed tokens will be injected into the system as continuous PoLC mining rewards. Note that under the second-score CELR staking/consumption mechanism, the participants are projected to submit bids matching their true valuation (truthfulness [17]) of the good (in this case, the opportunity to back the network liquidity).

**Example:** Assume that an OSP initiated a LiBA with the following parameters (600 ETH, 30 days, 1%) and there are three potential bidders (let’s say A, B, and

C) for this LiBA. The three bidders’ bids are *bA* = (1%, 800 CELR, 400 ETH); *bB* = (0.5%, 800 CELR, 200 ETH); *bC* = (1%, 100 CELR, 400 ETH). According to the scoring rule, we have *sB > sA > sC*. Since A and B can fill the entire request, they are selected as winners. It should be noted that even though A and C have the same interest rate (1%) and provide the same amount of liquidity (400 ETH), bidder A is selected as a winner while bidder C loses; this is due to the fact that their committed CELR tokens, as a symbol of their contributions to this platform, are significantly di↵erent. Finally, according to the second-score staking rule, A and B lock (or consume) their CELR tokens to match the score of C for 30 days.

After the auction process finishes, the OSP who initiated the liquidity request pays the interests to the wining liquidity backers by depositing into the LiBA contract. Upon receiving the payment of interests, the LiBA contract then gives the interests to the corresponding liquidity backers and issues 1:1 backed cETHs (using ETH as an example) that match the liquidity request amount. Although cETH is essentially an IOU, it brings no risk to the user as these IOUs are 100% insured by the network liquidity backers in the LiBA contract.

In normal cases, the LiBA contract is resolved before the timeout when the OSP sends back all the cETH tokens. Basically, before the timeout, the OSP will settle all paid cETHs to EUs with real ETHs by withdrawing from upstream channels collectively.

In the case where the OSP may get hacked, Celer Network’s trust model can vary. The simplest trust model without any protocol-level overhead is reputation-driven, where NLBs choose a reputable OSP without any history of default. In this simple model, NLBs are exposed to the risk of losing funds and assets as their CCCs are insurances for the EUs if the OSP defaults. However, it is arguable that even in this simple trust model, operating a highly reliable and reputable OSP is possible; it is very unlikely that all backings will be lost. There are additional security features which may be added around LiBA to further alleviate the potential risk. For example, newly issued cETHs are only allowed to be deposited to a whitelist of state channel contracts; cETHs are only allowed to be used incrementally with an upper bound spending speed. There are also a lot of things an OSP can do to maintain a secure infrastructure such as compartmentalized multi-node deployment, formal verification of security access rule of network infrastructure and more.

In addition, we enable an enhanced security model where a randomly selected quorum of NLB will need to co-sign an OSP’s operations (e.g. payment). These NLBs will only allow an outgoing transfer if and only if they see an incoming transaction with matching amount. These NLBs are also tethered to the incoming payments of OSP. If OSP fails to make the repayment eventually, NLBs will have the first-priority right to claim the incoming funds to OSP from other channels. However, we do note that this operation model will inevitably tradeo↵ some e ciency of the network.

Having said these, we believe the ultimate balance in the trust model should be defined by the market demand. We open both trust model for the market to organically evolve. We envision that the trust-free model will be more favorable in the early days of network launch and then it will become more trust-based.

Regardless of the LiBA’s trust model, we want to highlight that the LiBA process ensures that **end users never take any security risk** as the required liquidity is 100% “insured” by the LiBA contract. In Celer Network’s system, we strive to make sure that the benevolent end users do not need to worry about the security of their received fund and LiBA achieves that. PoLC and LiBA together incentivize an abundant liquidity pool, lower the barrier of becoming an o↵-chain service provider, reduce centralization risk, and accelerate network adoption.

### State Guardian Network

Another usage of CELR is to provide o↵-chain data availability with novel insurance model and simple interactions, which balances out the scalability-availability tradeo↵s as mentioned in Section 5.1.2.

From the surface, the availability problem seems to be an easy one to solve. One possible answer to that question might be: let’s build some monitoring services in the future and people will pay for these monitoring services when themselves are not online. It feels like a reasonable solution at first look, but we drive this train of thought just a little bit forward, we will immediately see track-wrecking flaws.

Let’s start with this question: are these monitoring services trust-based? If the answer is yes, then it creates another centralized choking point, single point of failure and is just not secure. Malicious counterparty can easily bribe these monitoring services to hurt benevolent end users.

Can we construct a monitoring service that is trust-free? For example, we may punish the monitoring service providers if they fail to defend the states for the users. However, when delving into this idea, we immediately see some caveats that render this approach impractical. How much penalty should monitoring service providers pay? Ignoring the frictions, the total penalty bond for monitoring service providers should be equal to the largest potential loss incurred to the party that went o✏ine.

This e↵ectively doubles the liquidity requirement for an o↵-chain network because whenever someone goes o✏ine, in addition to the existing locked liquidity in channels or fraud-proof bond in sidechains, monitoring service providers also need to lock up the same amount of liquidity as penalty deposits.

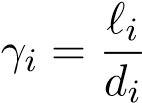
Worse, the monitoring service providers need to retain di↵erent assets for di↵erent monitoring tasks and things can get really complicated when the involved states are complex and multiple assets classes are in play. Sometimes, there is not even a straightforward translation from state to the underlying value, given all the complex state dependency for generalized state channels.

Even if there is enough liquidity, the “insurance” model here is really rigid: it is basically saying that you get *X*% back at once if the monitoring service providers fail to defend your states. If you choose a large value of *X*, it can become really expensive due to the additional liquidity locking, but if you choose a small value of *X*, it can become really insecure.

On top of these disadvantages, it is unclear how the price of state monitoring services should be determined as market information is still segregated with low e ciency. This low e ciency and the per-party bond on heterogeneous assets will further cause complicated 链上 and o↵-chain interactions with monitoring services and smash the usability of any o↵-chain platform. There are more issues, but above are already bad enough.

To solve these issues, we propose State Guardian Network (SGN). State Guardian Network is a special compact side chain to guard o↵-chain states when users are o✏ine. CELR token holders can stake their CELR into SGN and become state guardians. Before a user goes o✏ine, she can submit her state to SGN with a certain fee and ask the guardians to guard her state for a certain period of time. A number of guardians are then randomly selected to be responsible for this state based on state hash and the “responsibility score”. The detailed rules for selecting the guardians are as follows.

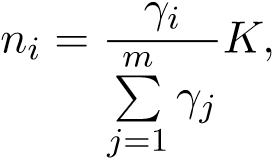
* (**State guarding request**). A state guarding request is a tuple *⌘i* = (*si,`i,di*) where *si* is the state that should be guarded, *`i* is the amount of service fee paid to guardians and *di* is the duration for which this state should be guarded.
* (**Responsibility Score**). The responsibility score of a state guarding request *⌘i* is calculated as:

*.*

A user’s Responsibility Score is essentially the income flow generated by this user to the SGN.

* (**Number of guardian stakes**). Given a set of outstanding state guarding request R = {*⌘*1*,*··· *,⌘m*}, the number of CELR at stake for each request *⌘i* 2 R

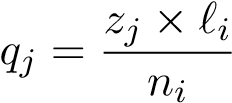
is



where *K* the total number of CELR stakes that guardians stake in the SGN. In other words, the amount of responsible CELR staked is proportional to the ratio between this requests responsibility score to the sum of all outstanding states responsibility scores.

* (**Assignment of guardian stakes**). Given a state guarding request *⌘i*, let *hi* be the hash value for the corresponding state *si* (e.g., Keccak256 hash). Each CELR stake *k* is associated with an ID *pk* (which is also a hash value). Let (*g*1*,g*2) be the distance between two hash values *g*1 and *g*2 (e.g., the distance measure used in Chord DHT [14]). Then CELR stakes are sorted in ascending order by their distance to the hash value *hi*. Suppose that (*p*1*,hi*)  (*p*2*,hi*)  ···  (*pK,hi*) (ties are broken randomly). The first *ni* CELR stakes that have the smallest distance are selected, and the corresponding stake owner will become the state guardian for this request.

(**State Guarding Service Fee Distribution**). For each state guarding request *⌘i* = (*si,`i,di*), the attached service fee *`i* is distributed to state guardians according to the following rule. For each state guardian *j*, let *zj* be the amount of his/her staked CELR that were selected for this state guarding request. Then the service fee that guardian *j* gets from state guarding request *⌘i* is

*.*

Note that each staked CELR has the same probability of being selected for a state guarding request. As a result, from the view of an SG, the more CELR staked in SGN, the more of such SG’s stakes will be selected in expectation (i.e., the value of *zj* will be larger), thus the amount of service fees that he will receive will increase. That a↵ords CELR significant value as a membership to the SGN.

(**Security and Collusion Resistance**). Each guardian is assigned a dispute slot based on the settlement timeout. If the guardian fails to dispute its slot when it ought to, subsequent guardians can report the event and get the failed guardian’s CELR stake. As a result, as long as at least one of the selected guardians are not corrupted and fulfills the job, an end user’s state is always safe and available for dispute.

The SGN mechanism also brings in the following additional values.

* **It does not require significant liquidity lock-up for guardians.** Guardians are only staking their CELR which can be used to guard arbitrary states regardless of the type/amount of the underlying value/tokens.
* **It provides a unified interface for arbitrary state monitoring.** Regardless of whether the state is related to ETH, any ERC20 tokens or complicated states, the users would just attach a fee and send it to SGN. SGN does not care about the underlying states and involved value, and simply allocates the amount of CELR proportional to the fee paid to be responsible for the state.
* **It enables simple interactions.** Users of Celer Network do not need to contact individual guardians and they only need to submit states to this sidechain.
* **Most importantly, it enables an entirely new and flexible state guarding economic dynamics.** Instead of forcing the rigid and opaque “get *X*% back” model, SGN brings users a novel mechanism to “get my money back in *X* period of time” and an e cient pricing mechanism for that fluid insurance model. If all guardians at stake fail to dispute for a user, she will get the CELR stakes from these guardians as compensation. In steady state, CELR tokens that are staked in the SGN represent an incoming flow (e.g., earning *x* Dai/second). Ignoring the cost of state monitoring and other frictions, when a user submits the state to SGN, she can choose explicitly how much CELR is “covering” for her state by choosing fees paid per second (i.e., the responsibility score).

### Summary

Thinking systematically, cEconomy covers the full life-cycle of an o↵-chain platform. LiBA and PoLC mining are about bringing intermediary transactions o↵-chain in a low-barrier fashion. SGN is about securing the capability to bring most up-to-date states back 链上 when needed. As such, we believe cEconomy is the first comprehensive o↵-chain platform cryptoeconomics that brings new value and enables otherwise impossible dynamics.

# Conclusion

Celer Network is a coherent technology and economic architecture that brings Internetlevel scalability to existing and future blockchains. It is horizontally scalable, trustfree, decentralized and privacy-preserving. It encompasses a layered architecture with significant technical innovations on each layer. In addition, Celer Network proposes a principled o↵-chain cryptoeconomics design to balance tradeo↵s made to achieve scalability. Celer Network is on a mission to fully unleash the power of blockchain and revolutionize how decentralized applications are built and used.

# Founding Team

**Dr. Mo Dong** received his Ph.D. from UIUC. His research focuses on learning based networking protocol design, distributed systems, formal verification and Game Theory. Dr. Dong led project revolutionizing Internet TCP and improved cross-continental data transfer speed by 10X to 100X with non-regret learning algorithms. His work was published in top conferences, won Internet2 Innovative Application Award and being adopted by major Internet content and service providers. Dr. Dong was a founding engineer and product manager at Veriflow, a startup specializes in network formal verification. The formal verification algorithms he developed is protecting networking security for fortune 50 companies. Dr. Dong is also experienced in applying Algorithmic Game Theory, especially auction theory, to computer system protocol designs. He has been teaching full-stack smart contract courses. He produces technical blogs and videos on blockchain with over 7000 subscribers.

**Dr. Junda Liu** received his Ph.D. from UC Berkeley, advised by Prof. Scott Shenker. He was the first to propose and develop DAG based routing to achieve nanosecond network recovery (1000x improvement over state of art). Dr. Liu joined Google in 2011 to apply his pioneer research to Googles global infrastructure. As the tech lead, he developed a dynamic datacenter topology capable of 1000 terabit/s bisection bandwidth and interconnecting more than 1 million nodes. In 2014, Dr. Liu became a founding member of Project Fi (Googles innovative mobile service). He was the tech lead for seamless carrier switching, and oversaw Fi from a concept to a $100M+/year business within 2 years. He was also the Android Tech Lead for carrier services, which run on more than 1.5B devices. Dr. Liu holds 6 US patents and published numerous papers in top conferences. He received BS and MS from Tsinghua University.

**Dr. Xiaozhou Li** received his Ph.D. from Princeton University and is broadly interested in distributed systems, networking, storage, and data management research. He publishes at top venues including SOSP, NSDI, FAST, SIGMOD, EuroSys, CoNEXT, and won the NSDI’18 best paper award for building a distributed coordination service with multi-billion QPS throughput and ten microseconds latency. Xiaozhou specializes in developing scalable algorithms and protocols that achieve high performance at low cost, some of which have become core components of widely deployed systems such as Google TensorFlow machine learning platform and Intel DPDK packet processing framework. Xiaozhou worked at Barefoot Networks, a startup company designing the worlds fastest and most programmable networks, where he led several groundbreaking projects, drove technical engagement with key customers, and filed six U.S. patents.

**Dr. Qingkai Liang** received his Ph.D. degree from MIT in the field of distributed systems, specializing in optimal network control algorithms in adversarial environments. He first-authored over 15 top-tier papers and invented 5 high-performance and highly-robust adversarial resistant routing algorithms that have been successfully applied in the industry such as in Raytheon BBN Technologies and Bell Labs. He was the recipient of Best Paper Nominee at IEEE MASCOTS 2017 and Best-in-Session Presentation Award at IEEE INFOCOM 2016 and 2018.

**References**

1. *Plasma: https://plasma.io/plasma.pdf*, .
2. Raiden Network Documentation: https://raiden-network.readthedocs.io. Accessed January 2018.
3. R. Khalil and A. Gervais, *Revive: Rebalancing O↵-Blockchain Payment Networks*, in *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*. ACM, 2017, pp. 439–453.
4. G. Malavolta, P. Moreno-Sanchez, A. Kate, and M. Ma↵ei, *SilentWhispers: Enforcing security and privacy in credit networks*, NDSS, 2017.
5. A. Miller, I. Bentov, R. Kumaresan, and P. McCorry, *Sprites: Payment channels that go faster than lightning*, CoRR abs/1702.05812 (2017). Available at http://arxiv.org/abs/1702.05812.
6. P. Moreno-Sanchez, A. Kate, M. Ma↵ei, and K. Pecina, *Privacy preserving payments in credit networks*, in *Network and Distributed Security Symposium*. 2015.
7. M.J. Neely, E. Modiano, and C.E. Rohrs, *Dynamic power allocation and routing for timevarying wireless networks*, IEEE Journal on Selected Areas in Communications 23 (2005), pp. 89–103.
8. L. Pham, J. Teich, H. Wallenius, and J. Wallenius, *Multi-attribute online reverse auctions: Recent research trends*, European Journal of Operational Research 242 (2015), pp. 1–9.
9. J. Poon and T. Dryja, *The bitcoin lightning network: Scalable o↵-chain instant payments*, Technical Report (draft) (2015).
10. P. Prihodko, S. Zhigulin, M. Sahno, A. Ostrovskiy, and O. Osuntokun, *Flare: An approach to routing in lightning network* (2016).
11. M.G. Reed, P.F. Syverson, and D.M. Goldschlag, *Anonymous connections and onion routing*, IEEE Journal on Selected areas in Communications 16 (1998), pp. 482–494.
12. S. Roos, M. Beck, and T. Strufe, *Anonymous addresses for e cient and resilient routing in F2F overlays*, in *Computer Communications, IEEE INFOCOM 2016-The 35th Annual IEEE International Conference on*. IEEE, 2016, pp. 1–9.
13. S. Roos, P. Moreno-Sanchez, A. Kate, and I. Goldberg, *Settling payments fast and private: E cient decentralized routing for path-based transactions*, arXiv preprint arXiv:1709.05748 (2017).
14. I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, *Chord: a scalable peer-to-peer lookup protocol for internet applications*, IEEE/ACM Transactions on Networking (TON) 11 (2003), pp. 17–32.
15. L. Tassiulas and A. Ephremides, *Stability properties of constrained queueing systems and scheduling policies for maximum throughput in multihop radio networks*, IEEE transactions on automatic control 37 (1992), pp. 1936–1948.
16. P.F. Tsuchiya, *The Landmark Hierarchy: A new hierarchy for routing in very large networks*, in *ACM SIGCOMM Computer Communication Review*, Vol. 18. ACM, 1988, pp.

35–42.

1. H.R. Varian and C. Harris, *The vcg auction in theory and practice*, American Economic Review 104 (2014), pp. 442–45.

**IMPORTANT NOTICE**

This Whitepaper in its current form is being circulated for general information and to invite investor feedback only on the Celer Network as presently conceived, and is subject to review and revision by the directors, the advisors, and/or the legal advisors of the Token Vendor. Please do not replicate or distribute any part of this Whitepaper without this note in accompaniment. No part of this Whitepaper is intended to create legal relations between a recipient of this Whitepaper and the Token Vendor, or to be legally binding or enforceable by such recipient against the Token Vendor. An updated version of this Whitepaper may be published on a date to be determined and announced by the Token Vendor in due course.

**PLEASE READ THIS SECTION AND THE FOLLOWING SECTIONS ENTITLED “DISCLAIMER OF LIABILITY”, “NO REPRESENTATIONS AND WARRANTIES”, “REPRESENTATIONS AND WARRANTIES BY YOU”, “CAUTIONARY NOTE ON**

**FORWARD-LOOKING STATEMENTS”, “THIRD PARTY INFORMATION AND NO CONSENT OF OTHER PERSONS”, “TERMS USED”, “NO ADVICE”, “NO FURTHER INFORMATION OR UPDATE”, “RESTRICTIONS ON DISTRIBUTION AND DISSEMINATION”, “NO OFFER OF INVESTMENT OR REGISTRATION” AND “RISKS AND UNCERTAINTIES” CAREFULLY.**

**IF YOU ARE IN ANY DOUBT AS TO THE ACTION YOU SHOULD TAKE, YOU SHOULD CONSULT YOUR LEGAL, FINANCIAL, TAX OR OTHER**

**PROFESSIONAL ADVISOR(S).**

While we make every effort to ensure that any material in this Whitepaper is accurate and up to date, such material in no way constitutes the provision of professional advice. The Token Vendor and its affiliates do not guarantee, and accepts no legal liability whatsoever arising from or connected to, the accuracy, reliability, currency, or completeness of any material contained in this Whitepaper. Participants and potential holders of Tokens should seek appropriate independent professional advice prior to relying on, or entering into any commitment or transaction based on, material published in this Whitepaper, which material is purely published for reference purposes alone. For the purposes of this Whitepaper, “**affiliates**” of the Token Vendor mean (i) any other person directly or indirectly controlling, controlled by, or under common control with, the Token Vendor, and (ii) the Foundation and any other entity developing and operating the Celer Network.

The Tokens will be issued as a cryptographic token. The Tokens are not intended to constitute, and should not be construed to constitute, securities of any form, units in a business trust, units in a collective investment scheme or any other form of investment in any jurisdiction. This Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities of any form, units in a business trust, units in a collective investment scheme or any other form of investment, or a solicitation for any form of investment in any jurisdiction.

This Whitepaper does not constitute or form part of any opinion or any advice to acquire, sell, or any solicitation of any offer by the Token Vendor to acquire any Tokens, nor shall it or any part of it, or the fact of its presentation, form the basis of, or be relied upon in connection with, any contract or investment decision.

No person is bound to enter into any contract or binding legal commitment in relation to the acquisition of Tokens and no cryptocurrency or other form of payment is to be accepted on the basis of this Whitepaper.

Any agreement between the Token Vendor and you as a participant in the Token Sale, and in relation to any purchase of Tokens, is to be governed by a separate document setting out the terms and conditions (the “**Token Sale Terms**”) of such agreement. In the event of any inconsistencies between the Token Sale Terms and this Whitepaper, the former shall prevail.

**PLEASE NOTE THAT THE TOKEN VENDOR WILL NOT OFFER OR SELL TO YOU, AND YOU ARE NOT ELIGIBLE AND YOU ARE NOT TO PURCHASE ANY TOKENS IN THE TOKEN SALE IF:**

1. **YOU ARE A CITIZEN, DOMICILED IN, OR A RESIDENT OF, OR LOCATED IN AN EXCLUDED JURSIDICTION;**

1. **YOU ARE INCORPORATED IN, OR OPERATE OUT OF, AN EXCLUDED JURSIDICTION; AND/OR**

1. **YOU ARE OTHERWISE PROHIBITED OR INELIGIBLE IN ANY WAY, WHETHER IN FULL OR IN PART, UNDER ANY LAW APPLICABLE TO YOU FROM PARTICIPATING IN ANY PART OF THE TOKEN SALE**

**(COLLECTIVELY, “EXCLUDED PERSONS”)**.

For the purposes hereof:

“**Excluded Jurisdiction**” means (i) jurisdictions with strategic anti-money laundering / counter-financing of terrorism deficiencies most recently identified by the Financial Action Task Force at <http://www.fatf-gafi.org/countries/#high-risk> (last accessed on [4 July] 2018); (ii) jurisdictions in which designated individuals and entities are identified by the Monetary Authority of Singapore for the purposes of regulations promulgated under the Monetary Authority of Singapore Act (Chapter 186) of Singapore, the United Nations Act (Chapter 339) of Singapore or the Terrorism (Suppression of Financing) Act (Chapter 325) of Singapore; (iii) Canada; (iv) New Zealand; (v) the People’s Republic of China; (vi) the United States of America; and (vii) any jurisdiction in which the Token Sale is prohibited, restricted or unauthorised in any form or manner whether in full or in part under the laws, requirements or rules in such jurisdiction.

No regulatory authority has examined or approved of any of the information set out in this Whitepaper. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements or rules have been complied with.

There are risks and uncertainties associated with the Token Vendor and its affiliates and their respective business and operations, Tokens, the Token Sale, and the Celer Network. Please refer to the section entitled “Risks and Disclosures” set out in this Whitepaper.

This Whitepaper, any part thereof and any copy thereof must not be taken or transmitted to any country where distribution or dissemination of this Whitepaper is prohibited or restricted.

No part of this Whitepaper is to be reproduced, distributed or disseminated without including this section and the following sections entitled “Disclaimer of Liability”, “No Representations and Warranties”, “Representations and Warranties By You”, “Cautionary Note On Forward-Looking Statements”, “Third Party Information and No Consent of Other Persons”, “Terms Used”, “No Advice”, “No Further Information or Update”, “Restrictions On Distribution and Dissemination”, “No Offer of Investment Or Registration” and “Risks and Uncertainties”.

**DISCLAIMER OF LIABILITY**

To the maximum extent permitted by all applicable laws, regulations and rules, the Token Vendor or any of its affiliates shall not be liable for any indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits, and loss of use or data), arising out of or in connection with any acceptance of or reliance on this Whitepaper or any part thereof by you or any person to whom you transmit any part of the Whitepaper to (whether authorised or unauthorised by any of the Token Vendor or any of its affiliates).

**NO REPRESENTATIONS AND WARRANTIES**

None of the Token Vendor or its affiliates makes or purports to make, and hereby disclaims, any representation, warranty or undertaking in any form whatsoever to any entity or person, including any representation, warranty or undertaking in relation to the truth, accuracy and completeness of any of the information set out in this Whitepaper.

**REPRESENTATIONS AND WARRANTIES BY YOU**

By accessing and/or accepting possession of any information in this Whitepaper or such part thereof (as the case may be), you represent and warrant to the Token Vendor as follows:

1. you agree and acknowledge that Tokens do not constitute securities of any form, units in a business trust, units in a collective investment scheme or any other form of investment in any jurisdiction;

1. you are not an Excluded Person;

1. you are not a citizen or resident of any jurisdiction in which either the purchase of, receipt, or holding of Tokens is prohibited, restricted, curtailed, hindered, impaired or otherwise adversely affected by any applicable law, regulation or rule;

1. none of you or (in the case of a corporation) any of your subsidiaries (if any), any of your directors or officers nor, any of your employees, agents or any other person

acting on behalf of you or any of your subsidiaries is an individual or entity that, or is owned or controlled by an individual or entity that:

* 1. is listed by the Monetary Authority of Singapore (“**MAS**”) as designated individuals or entities defined in the respective regulations promulgated under the Monetary Authority of Singapore Act (Chapter 186) of Singapore, the United Nations Act (Chapter 339) of Singapore or the Terrorism (Suppression of Financing) Act (Chapter 325) of Singapore or such other law, regulation or rule as may be prescribed by the MAS from time to time;

* 1. is the subject of sanctions administered or enforced by Singapore, the United States of America (including without limitation the U.S. Department of the Treasury’s Office of Foreign Asset Control), the United Kingdom of Great Britain and Northern Ireland, the European Union or any other Governmental

Authority (collectively, “**Sanctions**”);

* 1. is located, organised or resident in a country or territory that is the subject of country-wide or territory-wide Sanctions (including, without limitation, the Democratic People’s Republic of Korea, the Democratic Republic of Congo,

Eritea, Iran, Libya, Somalia, South Sudan, Sudan, and Yemen);

* 1. has engaged in and is not now engaged in any dealings or transactions with any government, person, entity or project targeted by, or located in any country or territory, that at the time of the dealing or transaction, is or was the subject of any Sanctions; or

* 1. is otherwise a party with which the Token Vendor is prohibited from dealing under laws applicable to you;

1. none of: (i) you; (ii) any person controlling or controlled by you; (iii) if you are a privately-held entity, any person having a beneficial interest in you; or (iv) any person for whom you are acting as agent or nominee in connection with your participation in the Token Sale is a senior foreign political figure, or any immediate family member or close associate of a senior foreign political figure, as such terms are defined below.

A “**senior foreign political figure**” is defined as a senior official in the executive, legislative, administrative, military or judicial branch of a government (whether elected or not), a senior official of a major political party, or a senior executive of a foreign government-owned corporation, and includes any corporation, business or other entity that has been formed by, or for the benefit of, a senior foreign political figure.

“**Immediate family**” of a senior foreign political figure typically includes such figure’s parents, siblings, spouse, children and in-laws. A “close associate” of a senior foreign political figure is a person who is widely and publicly known to maintain an unusually close relationship with such senior foreign political figure, and includes a person who is in a position to conduct substantial domestic and international financial transactions on behalf of such senior foreign political figure.

1. if you are affiliated with a non-U.S. banking institution (“**Foreign Bank**”), or if you receive deposits from, make payments on behalf of, or handle other financial transactions related to a Foreign Bank, you represent and warrant to the Token Vendor that:

* 1. the Foreign Bank has a fixed address, and not solely an electronic address, in a country in which the Foreign Bank is authorised to conduct banking activities;

* 1. the Foreign Bank maintains operating records related to its banking

activities;

* 1. the Foreign Bank is subject to inspection by the banking authority that licensed the Foreign Bank to conduct its banking activities; and

* 1. the Foreign Bank does not provide banking services to any other Foreign Bank that does not have a physical presence in any country and that is not a regulated affiliate;

1. you agree and acknowledge that this Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities of any form, units in a business trust, units in a collective investment scheme or any other form of investment in any jurisdiction, or a solicitation for any form of investment, and you are not bound to enter into any contract or binding legal commitment, and no cryptocurrency or other form of payment is to be accepted, on the basis of this Whitepaper;

1. you acknowledge and understand that no Tokens should be construed, interpreted, classified or treated as enabling, or according any opportunity to, token holders to participate in or receive profits, income, or other payments or returns arising from or in connection with Tokens or the proceeds of the Token Sale, or to receive sums paid out of such profits, income, or other payments or returns;

1. you agree and acknowledge that no regulatory authority has examined or approved of the information set out in this Whitepaper, no action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction and the publication, distribution or dissemination of this Whitepaper to you does not imply that the applicable laws, regulatory requirements or rules have been complied with;

1. you agree and acknowledge that this Whitepaper, the undertaking and/or the completion of the Token Sale, or future trading of Tokens on any cryptocurrency exchange, shall not be construed, interpreted or deemed by you as an indication of the merits of the Token Vendor and its affiliates, the Tokens, the Token Sale, and/or the Celer Network;

1. the distribution or dissemination of this Whitepaper, any part thereof or any copy thereof, or acceptance of the same by you, is not prohibited or restricted by the applicable laws, regulations or rules in your jurisdiction, and where any restrictions in relation to possession are applicable, you have observed and complied with all such restrictions at your own expense and without liability to the Token Vendor and/or its affiliates;

1. you agree and acknowledge that in the case where you wish to acquire any Tokens, Tokens are not to be construed, interpreted, classified or treated as:

* 1. any kind of currency other than cryptocurrency;

* 1. debentures, stocks or shares issued by any person or entity;

* 1. rights, options or derivatives in respect of such debentures, stocks or shares;

* 1. rights under a contract for differences or under any other contract the purpose or pretended purpose of which is to secure a profit or avoid a loss;

* 1. units in a collective investment scheme;

* 1. units in a business trust;

* 1. derivatives of units in a business trust; or

* 1. any form of investment;

1. you are legally permitted to participate in the Token Sale and all actions contemplated or associated with such participation, including the holding and use of Tokens;

1. the amounts that you use to acquire Tokens were not and are not directly or indirectly derived from any activities that contravene the laws and regulations of any jurisdiction, including anti-money laundering laws and regulations;

1. if you are a natural person, you are of sufficient age and capacity under the applicable laws of the jurisdiction in which you reside and the jurisdiction of which you are a citizen to participate in the Token Sale;

1. you are not obtaining or using Tokens for any illegal purpose;

1. you have a basic degree of understanding of the operation, functionality, usage, storage, transmission mechanisms and other material characteristics of cryptocurrencies, blockchain-based software systems, cryptocurrency wallets or other related token storage mechanisms, blockchain technology, and smart contract technology;

1. you are fully aware and understand that in the case where you wish to purchase any Tokens, there are risks associated with the Token Vendor and its affiliates and their respective business and operations, Tokens, the Token Sale, and Celer Network;

1. you bear the sole responsibility to determine what tax implications a purchase of Tokens may have for you and agree not to hold the Token Vendor, its affiliates and/or any other person involved in the Token Sale liable for any tax liability associated with or arising therefrom;

1. you agree and acknowledge that neither the Token Vendor nor its affiliates are liable for any direct, indirect, special, incidental, consequential or other losses of any kind, in tort, contract or otherwise (including but not limited to loss of revenue, income or profits, and loss of use or data), arising out of or in connection with any acceptance of or reliance on this Whitepaper or any part thereof by you;

1. you waive the right to participate in a class action lawsuit or a class wide arbitration against the Token Vendor, its affiliates and/or any person involved in the Token Sale and/or with the creation and distribution of Tokens; and

1. all of the above representations and warranties are true, complete, accurate and nonmisleading from the time of your access to and/or acceptance of possession this Whitepaper or such part thereof (as the case may be).

**CAUTIONARY NOTE ON FORWARD-LOOKING STATEMENTS**

All statements contained in this Whitepaper, statements made in press releases or in any place accessible by the public and oral statements that may be made by the Token Vendor or its directors, executive officers or employees acting on behalf of the Token Vendor (as the case may be), that are not statements of historical fact, constitute “forward-looking statements”. Some of these statements can be identified by forward-looking terms such as “aim”, “target”, “anticipate”, “believe”, “could”, “estimate”, “expect”, “if”, “intend”, “may”, “plan”, “possible”, “probable”, “project”, “should”, “would”, “will” or other similar terms. However, these terms are not the exclusive means of identifying forward-looking statements. All statements regarding the Token Vendor and/or its affiliates’ business strategies, plans and prospects and the future prospects of the industry which the Token Vendor and/or its affiliates are in are forward-looking statements. These forward-looking statements, including but not limited to statements as to the Token Vendor and/or its affiliates’ prospects, future plans, other expected industry trends and other matters discussed in this Whitepaper regarding the Token Vendor and/or its affiliates are matters that are not historic facts, but only predictions.

These forward-looking statements involve known and unknown risks, uncertainties and other factors that may cause the actual future results, performance or achievements of the Token Vendor and/or its affiliates to be materially different from any future results, performance or achievements expected, expressed or implied by such forward-looking statements. These factors include, amongst others:

1. changes in political, social, economic and stock or cryptocurrency market conditions, and the regulatory environment in the countries in which the Token Vendor and/or its affiliates conduct their respective businesses and operations;

1. the risk that the Token Vendor and/or its affiliates may be unable to execute or implement its business strategies and future plans;

1. changes in interest rates and exchange rates of fiat currencies and cryptocurrencies;

1. changes in the anticipated growth strategies and expected internal growth of the Token Vendor, its affiliates and/or the Celer Network;

1. changes in the availability and fees payable to the Token Vendor and/or its affiliates in connection with their respective businesses and operations or in the Celer Network;

1. changes in the availability and salaries of employees who are required by the Token

Vendor and/or its affiliates to operate their respective businesses and operations;

1. changes in preferences of users of the Celer Network;

1. changes in competitive conditions under which the Token Vendor and/or its affiliates operate, and the ability of the Token Vendor and/or its affiliates to compete under such conditions;

1. changes in the future capital needs of the Token Vendor and/or its affiliates and the availability of financing and capital to fund such needs;

1. war or acts of international or domestic terrorism;

1. occurrences of catastrophic events, natural disasters and acts of God that affect the businesses and/or operations of the Token Vendor and/or its affiliates;

1. other factors beyond the control of the Token Vendor and/or its affiliates; and

1. any risk and uncertainties associated with the Token Vendor and/or its affiliates and their respective business and operations, Tokens, the Token Sale, and the Celer Network.

All forward-looking statements made by or attributable to the Token Vendor and/or its affiliates and/or persons acting on behalf of the Token Vendor and/or its affiliates are expressly qualified in their entirety by such factors. Given that risks and uncertainties that may cause the actual future results, performance or achievements of the Token Vendor and/or its affiliates to be materially different from that expected, expressed or implied by the forward-looking statements in this Whitepaper, undue reliance must not be placed on these statements. These forward-looking statements are applicable only as of the date of this Whitepaper.

Neither the Token Vendor and/or its affiliates nor any other person represents, warrants, and/or undertakes that the actual future results, performance or achievements of the Token Vendor and/or its affiliates will be as discussed in those forward-looking statements. The actual results, performance or achievements of the Token Vendor and/or its affiliates may differ materially from those anticipated in these forward-looking statements.

Nothing contained in this Whitepaper is or may be relied upon as a promise, representation or undertaking as to the future performance or policies of the Token Vendor and/or its affiliates.

Further, the Token Vendor and/or its affiliates disclaim any responsibility to update any of those forward-looking statements or publicly announce any revisions to those forwardlooking statements to reflect future developments, events or circumstances, even if new information becomes available or other events occur in the future.

**THIRD PARTY INFORMATION AND NO CONSENT OF OTHER PERSONS**

This Whitepaper includes information obtained from various third party sources (“**Third Party Information**”). None of the publishers of Third Party Information has consented to the inclusion of Third Party Information in this Whitepaper and is therefore not liable for Third Party Information. While reasonable action has been taken to ensure that Third Party Information has been included in their proper form and context, neither the Token Vendor nor its directors, executive officers, and employees acting on their behalf, has independently verified the accuracy, reliability, completeness of the contents, or ascertained any applicable underlying assumption, of the relevant Third Party Information.

Consequently, neither the Token Vendor nor their directors, executive officers and employees acting on its behalf makes any representation or warranty as to the accuracy, reliability or completeness of such information and shall not be obliged to provide any updates on the same.

**TERMS USED**

To facilitate a better understanding of Tokens being the subject of the sale conducted by the Token Vendor, and the business and operations of the Token Vendor and/or its affiliates, certain technical terms and abbreviations, as well as, in certain instances, their descriptions, have been used in this Whitepaper. These descriptions and assigned meanings should not be treated as being definitive of their meanings and may not correspond to standard industry meanings or usage.

Words importing the singular shall, where applicable, include the plural and vice versa and words importing the masculine gender shall, where applicable, include the feminine and neuter genders and vice versa. References to persons shall include corporations.

**NO ADVICE**

No information in this Whitepaper should be considered to be business, legal, financial or tax advice regarding the Token Vendor and/or its affiliates, Tokens, the Token Sale, and/or the Celer Network. You should consult your own legal, financial, tax or other professional adviser regarding the Token Vendor and/or its affiliates and their respective business and operations, Tokens, the Token Sale, and the Celer Network. You should be aware that you may be required to bear the financial risk of any purchase of Tokens for an indefinite period of time.

None of the advisors engaged by us has made or purports to make any statement in the Whitepaper or any statement upon which a statement in the Whitepaper is based and each of them makes no representation regarding any statement in the Whitepaper and to the maximum extent permitted by law, expressly disclaims and takes no responsibility for any liability to any person which is based on, or arises out of, any statement, information or opinions in, or omission from, the Whitepaper.

**NO FURTHER INFORMATION OR UPDATE**

No person has been or is authorised to give any information or representation not contained in this Whitepaper in connection with the Token Vendor and/or its affiliates and their respective business and operations, Tokens, the Token Sale, or the Celer Network. If given, such information or representation must not be relied upon as having been authorised by or on behalf of the Token Vendor and/or its affiliates. The Token Sale shall not, under any circumstances, constitute a continuing representation or create any suggestion or implication that there has been no change, or development reasonably likely to involve a material change in the affairs, conditions and prospects of the Token Vendor and/or its affiliates or in any statement of fact or information contained in this Whitepaper since the date hereof.

**RESTRICTIONS ON DISTRIBUTION AND DISSEMINATION**

The distribution or dissemination of this Whitepaper or any part thereof may be prohibited or restricted by the laws, regulatory requirements, and rules of any jurisdiction. In the case where any restriction applies, you are to inform yourself about, and to observe, any restrictions which are applicable to your possession of this Whitepaper or such part thereof (as the case may be) at your own expense and without liability to the Token Vendor and/or its affiliates.

Persons to whom a copy of this Whitepaper has been distributed or disseminated, provided access to or who otherwise have the Whitepaper in their possession shall not circulate it to any other persons, reproduce or otherwise distribute this Whitepaper or any information contained herein for any purpose whatsoever nor permit or cause the same to occur.

**NO OFFER OF INVESTMENT OR REGISTRATION**

This Whitepaper does not constitute a prospectus or offer document of any sort and is not intended to constitute an offer of securities of any form, units in a business trust, units in a collective investment scheme or any other form of investment, or a solicitation for any form of investment in any jurisdiction. No person is bound to enter into any contract or binding legal commitment and no cryptocurrency or other form of payment is to be accepted on the basis of this Whitepaper.

No regulatory authority has examined or approved of any of the information set out in this Whitepaper. No such action has been or will be taken under the laws, regulatory requirements or rules of any jurisdiction. The publication, distribution or dissemination of this Whitepaper does not imply that the applicable laws, regulatory requirements or rules have been complied with.

**PREVAILING LANGUAGE**

The English language version of this Whitepaper is the only official version in force. If there is any inconsistency between this Whitepaper and other translations of this Whitepaper, the English version of this Whitepaper shall prevail. You acknowledge and agree that any translation you may have reviewed or which may have been made available to you is for your reference only and are not certified by the Token Vendor or its affiliates. Names of any laws and regulations, governmental authorities, institutions, natural persons or other entities which have been translated into English and included in this Whitepaper and for which no official English translation exists are unofficial translations for your reference only.

**RISKS AND UNCERTAINTIES**

Prospective purchasers of Tokens should carefully consider and evaluate all risks and uncertainties associated with the Token Vendor and/or its affiliates and their respective business and operations, Tokens, the Token Sale, and the Celer Network, all information set out in this Whitepaper and the Token Sale Terms prior to any purchase of Tokens. Further details of the risk factors relating to participating in the Token Sale and the Token Vendor will be set out in the Token Sale Terms. If any of such risks and uncertainties develops into actual events, the business, financial condition, results of operations and prospects of the Token Vendor and/or its affiliates could be materially and adversely affected. In such cases, you may lose all or part of the value of Tokens.

1. Channel utilization corresponds to the ratio between the amount of transferred tokens in each time slot and [↑](#footnote-ref-1)
2. We name decentralized applications running on Celer Network as cApps. [↑](#footnote-ref-2)
3. These weights are subject to future decentralized governance adjustment. [↑](#footnote-ref-3)