让支付快捷且私密：为基于路径的交易而来的高效分散的路由

Stefanie Roos

University of Waterloo sroos@uwaterloo.ca

ArXiv:1709.05748v1 [cs.CR] 18 Sep 2017

Pedro Moreno-Sanchez

Purdue University

pmorenos@purdue.edu

Aniket Kate

Purdue University

aniket@purdue.edu

Ian Goldberg

University of waterloo

iang@cs.uwateoo.ca

**摘要：基于路径的交易（PBT）网络是一个在研究中成长的领域，这些网络通过一个中间路径从一个用户向另一个用户提供支付。 他们克服了像比特币和以太坊这样的加密货币在可扩展性和隐私上的问题，通过廉价而快速的外链转移来代替昂贵且缓慢的链上区块链操作。以信用网络为形式的如Ripple和Stellar，它们还能够在不同的金融机构中实现低价实时总量结算。 例如，SilentWhispers是最近提出的完全分布式信用网络，依靠基于路径的交易进行安全的、特别是没有公共分类账的私人支付。**

**分布式PBT网络的核心是一种路由算法，可以发现付款人和收款人之间的交易路径。 在去年，许多路由算法已经被提出来，包括SilentWhispers的地标路由和比特币闪电网络的链路状态协议Flare。 然而，现有的临时性工作缺乏效率和隐私性，需要进行全面分析以确保PBT网络在实践中取得成功。**

**在这项工作中，我们首先鉴定了SilentWhispers中的几个效率问题。 有了这个知识，我们设计和评估了SpeedyMurmurs，一种用于分散式PBT网络的新型路由算法，使用高效灵活的基于嵌入的路径发现和按需高效的稳定来掌控PBT网络的动态。 我们基于当前部署的Ripple信用网络的实际数据的仿真研究，指出SpeedyMurmurs将稳定性开销减少了两个数量级，并将事务处理的开销减少了一倍以上。 此外，使用SpeedyMurmurs保持至少与分散的地标路由相同的成功率，同时提供较低的延迟。 最后，SpeedyMurmurs实现了PBT网络路由的关键隐私目标**。

一、介绍

自比特币[23]出现以来，许多其他基于区块链的支付系统已被提出并应用在实践中，以满足多种用途。 例如Ripple [3]，[5]或Stellar [4]等信用网络利用区块链技术实现两个不同货币和资产的最终用户之间的实时总额结算[2] 比目前的中央银行系统便宜。 以太坊[10]建立在区块链之上，构建一个平台来完整地运行表达的智能合约。

然而，越来越多的用户和交易的基础导致了区块链的可扩展性问题[6]，[25]。 此外，区块链的公共性质导致明显的敏感数据的隐私泄露，如交易伙伴的身份和交易价值[13]，[17]，[18]，[22]，[28]。 学术和行业方面的努力正在逐步引向对等（P2P）基于路径的交易（PBT）网络，例如用于比特币的Lightning Network [25]，用于Ethereum的Raiden网络[1]，用于信用网络的SilentWhispers [15] ，或用于区块链交易的InterLedger [33]和Atomic-swap [14]; 这些PBT网络有望通过无需昂贵的采矿工作的链式交易解决区块链和加密货币的可扩展性，效率和互操作性问题。 实际上，在最近的区块链事件中，InterLedger团队通过包括比特币，Ethereum和Ripple在内的七个分类帐证明了一笔交易[36]。

在PBT网络中，两个用户u和v在本地维护它们之间的加权链接（也称为支付通道，状态通道或信用链接，取决于应用程序）。 链接的权重特征化了一个用户可以转移到另一个用户的资金（或资产）的数量，链接的确切性质取决于应用程序。 例如，在信用网络中，权重定义了你愿意授予的信用额度v和已经欠你的金额v之间的差额。 然而，所有PBT网络的核心是通过在一个或多个路径之间转移资金来实现支付者与收款者之间的资金交易。

PBT网络建立在三个关键算法之上：路由，支付和问责。 路由算法负责从付款人到收款人寻找足够资金的路径。 支付算法将付款人与收款人之间的资金沿着连接它们的路径进行结算。 最后，问责算法允许在存在不当行为的用户的情况下解决争议。

虽然经常被忽视或误认为一个正交问题，路由算法的设计对于PBT网络的有效性来说是关键的，其特点是成功解决交易的一小部分;效率，其特点是在交易中经历的延迟以及由交易产生的开销;以及可扩展性，其特征在于PBT网络能够保持越来越多的用户和交易基础的有效性和效率。依靠最大流算法的PBT网络不太可能提高区块链以外的可扩展性，而其他方法可能无法发现大部分路径。效率，有效性和可扩展性被认为是有兴趣部署或投资PBT网络的公司的主要关注点。因此，隐私的重要性可能会被忽略，导致路由算法可能会泄露敏感信息，如交易价值，付款人和收款人的身份以及一个用户对另一个用户的债务。在本文中，我们强调所有的有效性，效率，可扩展性和隐私性对于路由算法的设计是重要的。隐私保护但效率不高的算法不太可能被部署。

到目前为止PBT网络提出的少数路由算法无法实现隐私，效率或可扩展性。例如，Canal [35]依靠单一服务器来存储完整的PBT网络，查找路径并解决用户之间的支付。因此，服务器对用户和他们的事务之间的所有链接都了如指掌。 PrivPay [21]利用可信任的硬件加密服务器上的支付网络数据，并使用不经意的算法来隐藏访问模式，从而增加了用户与其付款之间的链接的隐私。尽管如此，PrivPay仍然有单点故障和低可扩展性。 Flare [27]是比特币闪电网络的一种路由算法，它要求从付款人到收款人的路径上的每个用户，将他们的状态通道的当前资金数额发送给付款人，从而泄露敏感信息[27，3.6节]。关于隐私的最有前途的方法是SilentWhispers [15]，一个没有公共分类帐的完全分布的PBT网络。 SilentWhispers中的路由算法提供了正式的隐私保证，但是效率不高，正如我们在本文中所展示的那样。

在这项工作中，我们提出了一种PBT网络的路由算法SpeedyMurmurs，它在完全分布式的环境中提供了正式的隐私保护，并且在效率和效率方面优于最先进的路由算法。 SpeedyMurmurs扩展了VOUTE [30]，一种基于隐私保护的基于嵌入的路由算法，用于路由限制的P2P网络中的消息发送。 针对无向和无权网络中的消息传输，而不是支付，VOUTE没有处理加权链接，特别是以前的转移导致的这些权重的变化。 SpeedyMurmurs将VOUTE的基本思想与信用网络的特点相结合。 尤其是：

•SpeedyMurmurs在路由支付时考虑了可用资金和邻居到目的地的距离，从而得到了一个灵活的路径选择的高效算法。

•SpeedyMurmurs采用按需高效的稳定协议，在需要的时候对链路的变化做出反应，但是保持与这些变化相对应的开销低。

•SpeedyMurmurs通过允许节点主动精确分配交易所需的资金数量，而不是阻止并发交易完全使用链接或在随后的付款阶段冒失败风险，从而改进了并发交易处理。

•在我们的模拟研究中，SpeedyMurmurs模拟信用网络和基于2013年至2016年真实世界的Ripple数据集的交易，以大约两倍于SilentWhispers的速度执行交易，并将交易的通信开销减少至少二分之一，同时保持相似或更高的效果。

•SpeedyMurmurs减少管理链接变化的开销2-3个数量级，除了稀有Ripple数据集中的阶段（每年大约一个阶段）相应突然快速增长。

•SpeedyMurmurs实现了一些关键的隐私目标，特别是隐藏交易的价值以及付款人和收款人的身份。

总之，SpeedyMurmurs为信用网络中的隐私保护路由提供了一个有效的解决方案，从而成为即将部署这种网络的有希望的候选者。

二、现状和局限性

在本节中，我们描述SilentWhispers [15]，分布式隐私保护信用网络中最有前途的方法，以及VOUTE中基于嵌入的路由的概念[30]，我们在第四节中适应了PBT网络。

A.地标路由

地标路由技术[34]能够在网络图中计算付款人和收款人之间的路径子集，而不依赖于成本密集的最大流量方法。地标路由的关键思想是确定从付款人到收款人通过中间节点的路径，称为地标，通常是高连通性的着名节点。虽然地标路由没有发现所有可能的路径，并且可能导致成功支付的可能性较低，但过去的工作表明成功的减少与性能的提升相比是很小的[21,35]。

最初，每个地标启动宽度优先搜索（BFS）算法的两个实例。在第一个实例中，只考虑前沿，并且计算从地标到每个节点的最短路径。第二个实例仅考虑反向边缘并得出每个节点与地标之间的最短路径。随着PBT网络随着时间而改变，地标周期性地重复初始化过程。

付款人和收款人之间的路径发现使得从付款人到地标的路径（使用反向边缘）和从地标到收款人的路径（使用向前边缘）相关联。只要资金总额高达路径上每条链路的可用信用额，付款人就可以沿路径发送资金。

有两个版本的地标路由。第一个版本总是连接从源头到地标和从地标到目的地的路径。第二个版本，我们称之为树型路由，发现BFS树中的最短路径，它不一定包含地标。

1）SilentWhispers中的地标路由：SilentWhispers使用地标路由来发现多个路径，然后执行多方计算以确定沿着每个路径发送的资金量。地标路由的初始化遵循以上描述，使用执行周期性BFS的多个地标。

实际支付依赖于两个操作：探测操作和实际支付操作。 我们在这里描述探测操作，因为它执行路由并决定沿每条路径传输的信用。 然后支付操作仅以安全的方式执行由探测操作建议的转移。

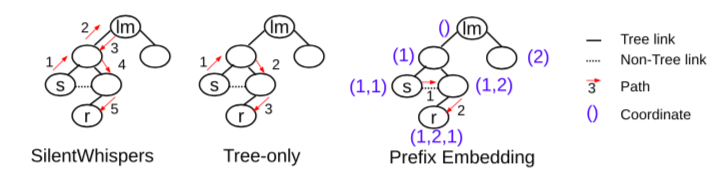


图1：不同树形路由方案的例子，地标lm，发送者s，接收者r。

探测操作的核心是基于秘密共享的多方计算，计算路径中的可用信用。在使用地标路由发现付款人和收款人之间的路径之后，路径中的每一对相邻用户将其链路值的一部分发送到每个地标。付款人和收款人必须建立额外的股分作为填充，以隐藏路径的实际长度，并有效地保留实际的付款人和收款人的身份。在密码签名的帮助下，依靠新鲜的密钥来隐藏路径上的节点的身份，并使用多方计算，地标确定支付者可以合并的份额以获得第i条路径的最小可用信用度zi 。如果zi 值的总和至少等于总支付金额，则付款人分配值c1,…, c | L |到ci≤zi的路径。探测操作的结果是这些值ci和对支付操作利用来执行实际转移的路径的处理。

2）SilentWhispers的弱点：基于以上描述，我们确定了在SilentWhispers中实现路由算法的四个问题。首先，定期树形成（BFS的执行）没有立即考虑到网络的变化，由于信息过时可能导致显著的失败率。而且，由于重构了可能没有改变的生成树的部分，周期性的树形成导致了不必要的开销。

第二，即使i）支付的发送者和接收者在同一分支中，或者ii）发送者和接收者之间存在短路径，但链路不是生成树的一部分，所有路径都包括地标。因此，用于支付的整个路径可能是不必要的漫长的，由于遇到至少一个没有足够资金的链路的机会增加，导致更长的延迟和更低的成功率。

第三，探测操作要求包含在交易路径中的节点将股份发送到所有地标。这意味着交易开销在地标数量上按比例缩放。

第四，SilentWhispers没有提供合适的并发解决方案。假设一个或多个探测操作旨在使用相同的链接。探测操作可以为两个链路提供相同数量的可用信用，或者在探测操作第一次发现包含链路的路径之后阻塞链路的使用一段时间。前者可能导致后来的失败作为可用的信用，虽然足够每次独自转账，可能不包括多次转账。而一个区块阻止这种复杂性，但它增加了失败的可能性，因为探测操作不能使用某些链接，这些链接可能有足够的信用来执行多个事务。因此，两种并发方法都有严重的缺点。

总之，地标路线和SilentWhispers有我们在这项工作中要克服的各种弱点。

B．基于嵌入的路由

路由对于将数据包从源转发到目标的任何网络都是必不可少的。 但是，与PBT网络相比，传统网络链路的容量不会因传输消息而变化。 在下文中，我们总结贪婪嵌入的概念[24]，一个高效的路由协议，需要很少的状态信息。 事实上，贪婪的嵌入保持很少的状态信息，因此揭示了有关网络拓扑结构的很少的信息，这促使它们用于隐私保护通信系统，如朋友对朋友叠加[12]，[30]。

嵌入依赖于将坐标分配给网络中的节点，并且节点根据节点已知的坐标与目标坐标之间的距离来转发分组。 贪婪嵌入类似于地标路由，因为它们根据节点在生成树中的位置来分配坐标。 然而，贪婪的嵌入在分配坐标之后忽略生成树，特别是使用不在生成树中的链接发现较短的路径。 我们引用树中没有包含但在路由期间用作快捷方式的链接。 当一个节点v转发一个消息到一个目的地坐标时，v选择邻居坐标最接近目标坐标转发消息。 因此，v可以使用生成树中的链接（转发给子节点或父节点）或快捷方式。

尽管路由可以包含快捷方式，但不能保证存在快捷方式的路由。 因此，树中的链路提供了路由工作和删除任何这样的链路的保证可能导致故障。 因此，在节点或链路改变时调整树是非常重要的。

前缀嵌入[12]是一个贪婪的嵌入，使消息路由在F2F中覆。 如图1所示，前缀嵌入以向量的形式分配坐标，从地标/根处的空向量开始。 生成树的每个内部节点枚举它的子节点和将子元素的枚举索引附加到其坐标以获得子坐标。 两个坐标之间的距离对应于它们之间生成树中最短路径的长度; 即用|id(v)| 表示两个坐标id(u)和id(v)的距离的坐标长度，cpl(id(u)，id(v))表示的通用前缀是

d(id(u),id(v))=|id(u)|+|id(v)|−2cpl(id(u),id(v)).(1)

根据公式图1，节点根据坐标的距离来确定哪个邻居离接收机最近并相应地转发消息。 图1展示了一个例子来说明各种基于树的路由方案之间的区别，并说明前缀嵌入中的坐标分配。

1）前缀嵌入VOUTE：VOUTE [30]是一个基于前缀嵌入的路由算法，目的是匿名和有效的消息传递动态路由限制覆盖;即不允许在任意节点之间建立链接的叠加。我们很快就描述了VOUTE如何解决匿名和动态的问题。

前缀嵌入显示目的地的唯一接收者地址。相反，VOUTE允许节点提供匿名回报地址，而不是原来的坐标。接收器通过将其坐标填充到固定长度并生成坐标元素的键控散列来生成返回地址。匿名返回地址然后对应于键控哈希值和密钥，允许转发节点确定公式中所需的公共前缀长度。根据接收者坐标id(r)和邻居坐标id(u)的公共前缀长度，转发节点可以计算d(id(u), id(r)) +Δ，Δ对应于填充的恒定长度。因此，他们可以沿着与使用明文坐标时相同的路径转发消息，同时保持接收者的真实坐标的私密性。

原始的前缀嵌入坐标反映了一个enu - meration，因此有小熵。 因此，VOUTE用随机的b位数字替换枚举索引; 例如，b = 128。以这种方式，猜测未知节点的坐标对于对手来说在计算上是不可行的。

VOUTE不是定期重建生成树，而是使用按需稳定协议来解决动态问题。 当构建树时，节点向所有邻居发送邀请，说明它们的坐标并提供成为父节点。 每个节点接受一个这样的邀请，但保持所有邻居的最近邀请，以快速响应网络动态。 如果节点建立新的链路，则生成树中已经包含的节点向其新邻居发送邀请。 如果一个节点还不是树的一部分，它接受邀请。 否则，将其存储以备将来考虑。 另一方面，如果生成树中的链接不再存在，则子节点及其所有后代将根据其剩余的邀请来选择新的父节点。 然后他们向所有的邻居传播新的协调。 以这种方式，生成树和嵌入具有按需修复机制而不是定期重新计算。

2）VOUTE的局限性：VOUTE尚未在PBT网络的环境中定义，因此在将其视为PBT网络中的路由算法之前必须克服一些限制。尤其是，VOUTE在链接和网络动态性质方面有不相容的假设。

首先，VOUTE会考虑用户对之间的无向链接和未加权链接。在一个PBT网络中，链接是加权和定向的，就像支付一样。虽然所有链接都允许在VOUTE中传输消息，但PBT网络中的链接可能没有足够的资金来执行传输。链接的定向性表明VOUTE的构建协议是不够的，因为不清楚如何处理单向链接。如果单向链路是生成树的一部分，那么一个节点（和它的后代）可能只能发送或接收资金，但不能同时发送或接收资金。链接的加权性质和不可能使用链接的所有付款都与VOUTE协议的一个关键假设相矛盾，即在没有链接失败的情况下，所有链接都可以传递消息。在信用网络中应用VOUTE，有必要设计处理加权链接和传输的协议。

其次，VOUTE考虑节点加入和离开网络的形式。但是，在PBT网络中，链路的权重是主要的变化来源。特别是，每个成功的交易可能会改变几个链接。如果对所有这些变化作出反应，PBT网络的VOUTE变体可能是低效的。确定何时以及如何适应链路变化对于这种变体的设计是重要的。

最后，VOUTE不必处理并发问题。虽然并行消息传输可能会增加延迟和拥塞，但它们不会改变链路容量，传输消息不会影响链路传输未来消息的能力。但是，像SilentWhispers中那样分开的探测和支付操作会产生并发问题。 SilentWhispers在这里提供的解决方案不足，所以我们需要一个新的并发协议。

总之，虽然VOUTE提供了一个有趣的地标路由替代方案，在SilentWhispers中为路由操作实现，但是它在PBT网络场景中的应用并不直接。

三、系统模型和目标

我们从分布式路由算法的通用系统模型开始，接下来是我们在加密游戏中表达的隐私目标以及我们的性能指标。

A.我们的模型

我们将分布式PBT网络(G,w)建模为有向图G =(V,E)和边集上的权函数w。节点集合V对应于PBT网络的参与者。如果u能够将资金转移到v，则存在从节点u到v的链接(边)。我们将节点v的输出邻居集定义为N*out*(v) = {u ∈ V : (v,u) ∈ E} 。相应地，我们定义节点v的入站邻居集合为N*in*(v) = {u ∈ V : (u,v) ∈ E} 。此外，路径p是链路e1 ...el 的序列，其中ei = (vi1,vi2), 并且 vi2 = v(i+1)1 ，1≤i≤l-1。此外，我们用L = {l1,...,l|L|} 表示一组高度连接的节点，称为地标，这是PBT网络中其他用户所熟知的。我们用|L|表示集L的大小。

函数w描述了两个共享一个边的节点之间可以转移的资金数量。由此我们从函数w的具体实现中抽象出来。例如，在比特币闪电网中，函数w：E→R被定义为w(u,v) = pot(u,v) ，其中pot定义了在一个支付通道他们。此外，每一条边都有一个上界(u,v)和一个下界，其值为0.通常0≤w(u,v)≤cred(u,v)。

我们将路径e1,...,el 中可用的资金定义为最小值(ei)。此外，我们定义了节点v的净平衡为

*1)* 操作：一个PBT网络由一个元组组成算法 (*setRoutes*, *setCred*, *routePay*) 的定义如下:

*setRoutes*(L): 给一组地标 l1 , . . . , l|L| , *setRoutes*初始化PBT网络中每个节点所需的路由信息。

*setCred*(c,u,v): 给定值 c 和节点 u 和 v, *setCred 设置* w(u,v) = c. 另外, *setCred* 可能会改变*setRoutes*最初生成的路由信息。

((*path*1, c1), . . . , (*path*|L|, c|L|)) ← *routePay*(c, u, v). 给一个定值 c, 一个发送者 u 和 一个接收者 v, *routePay* 返回一组元组 (*path*i , ci ), 表示 ci 资金通过*path*i := ei1 ,. . . , eili 描述的路径。

a）正确性：支付系统的关键属性是正确的。直观地说，正确性表明路由算法i）建议花费所需的资金c而不是更高的价值，并且ii）建议确实具有足够资金的路径。设(*setRoutes*,*setCred*,*routePay*) 是PBT网络的路由操作。 如果对于*routePay*(c, u, v)的所有结果((*path*1 , c1 ), . . . , (*path*|L| , c|L| ))有一下两种条件成立，我们说PBT网络是正确：

•

•每个*path*i := ei1,...,eili and each eij, ci ≤ w(eij).

我们注意到routePay操作可能会返回贡献i的路径，并且它仍然被认为是正确的。这说明了PBT网络在发送者和接收者之间没有提供足够流动性来执行交易的情况。

B.攻击者模型

我们考虑一个完全分布的网络。我们的主要攻击方案是对用户的财务状况感兴趣的公司和个人，而不是政府的安全机构。攻击者通过插入自己的节点或破坏现有节点来控制网络中的一部分节点。我们假设对手不能随意选择用户组，因为有些用户将很难被社交工程或恶意软件攻击所腐蚀。一般情况下，我们假设攻击者不知道网络中的所有链路和节点，特别是不能访问本地存储在非攻击节点上的路由信息​​。攻击者不知道来自多个地区和国家的参与者的大规模分布式系统的完整拓扑对于我们的攻击场景来说似乎是现实的。

我们的对手旨在破坏隐私，而不是进行大规模的拒绝服务攻击。我们认为，拒绝服务攻击的主要防御措施是检测和驱逐恶意节点。虽然涉及到路由，但实现检测和驱逐需要不同的操作，而且不在本文的讨论范围之内。

尽管我们的整体对手模型限制了对手的能力，但是我们仍然为具有全局拓扑结构的攻击者定义了我们的价值隐私目标，表明我们仍然可以获得一些隐私，以抵御更强大的攻击者。

C.隐私目标

密码学和分散化可能确保强大的隐私的希望是比特币和区块链早期成功的最强大驱动力之一。 我们期望雇用PBT网络的企业和客户有兴趣隐藏来自竞争对手甚至服务提供商的交易。 因此，确保基于路径的交易的隐私性非常重要。

像PrivPay [21]和SilentWhispers [15]一样，我们的目标是隐藏价值（价值隐私），以及基于路径的交易的付款人和收款人（付款人/收款人隐私）的身份。 我们使用术语交易隐私来指符合所有这三个概念。 接下来，我们非正式地描述PBT网络的这些隐私属性，并将读者引用到PrivPay论文[21]中，定义在信用网络环境中的形式化版本。

价值隐私：如果任何对手无法确定非受损用户之间的交易总价值，只要没有受到损害的中间节点，PBT网络就可以实现价值隐私。

让s和r成为两个不妥协的用户，并且让（(*path*1, c1), . . . , (*path*|L|, c|L|) 是*routePay*(c, s, r)操作的结果。如果对于每条路径路径，那条路径上的所有节点都不受影响，那么攻击者（即使是全局被动对手）也不会获得有关事务值c的信息。

请注意，正如第IV-G节所述，即使对手妥协了一些中间节点，只要所用路径中的至少一个路径上的所有节点都不受影响，我们也可以提供较弱的价值隐私形式。

付款人隐私：如果任何对手无法确定付款人在非受损用户之间的基于路径的交易中，则PBT网络实现付款人隐私。

特别是，对于两个非受损用户s和r，攻击者不应该能够确定任何路由操作*routePay*(z,s,r)的支付者，除非她完全了解s的输入链接，即她知道集合N*in*(s)虽然不一定是链接e ∈ N*in*(s)的资金。

请注意，尽管没有全局网络视图的本地攻击者可能知道（甚至控制）N*in*(s)中的所有节点，但是她可能并不知她控制了所有这样的节点。 因此，类似于P2P匿名系统[19]，[20]，[29]，控制所有邻居并不意味着s可以确定是否启动了路由。 因此，即使当攻击者控制付款人的所有节点N*in*(s)时，我们也期望付款人隐私保持不变，但是不知道她是否控制了整个集合。

收款人的隐私定义类似于付款人的隐私，对抗性假设也保持不变，除了付款人的邻居节点N*in*(s)，现在为收款人隐私，我们考虑收款人r的N*out*(r)。

D.性能指标

在本节中，我们描述了由路由协议实现的目标，我们用R来表示路由协议。下面用(Gt,wt)表示一个PBT网络在时间t的快照。请注意，虽然我们在这项工作中抽象了支付和问责协议，但PBT网络必须实施它们，因此PBT网络是动态的。 假设{(ti, ci, si, ri)}是从si 到 ri的对于时间t i的量ci一组支付请求。

路由协议R的性能由以下四个指标表现：

•成功率：((*path*1 , c1 ), . . . , (*path*|L| , c|L| )) 是由R中实现的routePay(c,s,r)返回的路径集合。只有当我ci = c时，交易才能成功。成功率说明成功交易的比例。**[[1]](#footnote-1)**

•（跳）延时：R关于交易(t,c,s,r)的延迟是终止时间与启动时间t之间的差值。在没有包括实际计算和通信延迟的具体实现的情况下，我们提供了如下的延迟的抽象测量。假设m1和m2是由R发送的消息。如果一个节点发送m2作为m1接收的结果，那么m2就是在m1之后。跳时延是R发送的后续消息的最长链的长度。

•交易开销：节点交换消息以执行交易（t，c，s，r）。事务开销表示交换位数。和以前一样，在没有具体的实现的情况下，我们通过假定每个实现的大小相同的消息来提取它，并且将消息的数量描述为开销。

•稳定性开销：类似于事务开销，稳定性开销对应于在特定时间间隔内发送的比特数以维护必要的状态信息，如R中的操作setRoutes（）的实现所要求的。同样，我们从说明消息的数量而不是比特数的具体实现。

前两项指标严重影响服务质量，而后两项指标与网络拥塞直接相关，从而影响延迟。 此外，开销决定了用户设备上的负载。

四、我们的解释

我们首先描述我们设计的关键思想，然后详细讨论三个操作中的每一个。 我们给出算法集中版本的伪代码，它允许以线性紧凑的方式呈现算法。 然后我们描述PBT网络中使用的分布式版本与集中版本的区别。

A.假设

PBT网络中的每个用户在本地维护与邻居的链接信息。 我们进一步假设共享链接的用户可以通过认证和保密的通信渠道相互发送消息。 此外，我们假设存在一组高度连接的节点，称为地标，是PBT网络中的其他用户所熟知的。 我们注意到这些假设与其他分布式PBT网络如SilentWhispers一致。

在本节中，我们指的是在两个方向上都有非零资金的链接，即pot(u, v) > 0与pot(v, u) > 0的链接(u, v)和(v, u)是双向的。 如果两条链路中的一条不存在或者资金不足，则u和v有一个单向链路。

B.概述和主要思想

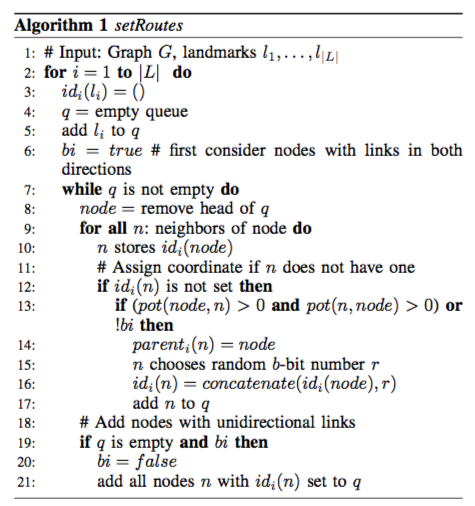
我们在这里描述SpeedyMurmurs关于setRoutes，setCred和routePay操作的关键思想。我们特别关注与现有协议的主要差异。

setRoutes：在这个算法中，我们构造了多个嵌入，每个标记一个嵌入。如第II-B部分所述，VOUTE为基于BFS的坐标分配提供了一个协议，该协议假设了未加权和无向链路。我们修改这个协议，把它分成两个阶段。首先，我们只添加双向链接。在协议的第二阶段中，尚未成为生成树一部分的节点通过向生成树添加单向链接进行加入。

setCred：我们利用SilentWhispers的协议来给c设置资金。主要挑战在于设计嵌入的适当变化。VOUTE没有提供关于如何对重量变化作出反应的指导。与setRoutes一致，如果两个节点i）建立了非零权重的新链接（或者将链接的值从0设置为非零值），则我们决定只发起改变，并且ii）零链接（或将其值设置为0）。如果他们建立了一个新的链接，其中一个节点可以选择另一个作为父母，如果它没有父母，或者其当前父母的链接只有一个方向的功劳。相比之下，如果链接被删除，其中一个节点必须选择一个新的父母（和坐标）如果其他节点以前是其父母。此外，受影响的节点的任何后代都必须改变坐标。

routePay：路由由三个步骤组成：i）收款人生成匿名返回地址并将它们发送给付款人; ii）付款人随机将交易值拆分为|l|路径，每个地标一个，iii）VOUTE的路由算法找到付款人和收款人之间的路径，限于有足够资金的链路。 我们的算法允许灵活的路线选择，更喜欢高资金路径。 事先确定每条路径上的资金转移，可以使节点在探测操作中阻止一定数量的信用，保证后续支付成功，不会阻塞所有的链路资金。

C. setRoutes

在算法1中描述的初始化阶段，我们导出嵌入。迭代所有地标，算法1将地标坐标分配为空向量（第3行），并将该地标附加到队列（第5行）。算法的主循环然后处理队列。在每个步骤中，算法从队列中删除节点（第8行）并考虑其所有邻居。如果邻居n还没有坐标并且有资格拥有一个坐标，算法将n合并到生成树中，通过连接父坐标和一个随机的b位数来分配坐标，并将其附加到队列中。 （第14-17行）。决定参与跨越的资格的标准取决于算法的阶段：最初(bi = true，第6行)，如果连接到和来自其潜在父母的链路上的可用信用为非零。在算法的第二阶段(bi = false)，由空队列（第18-21行）触发，所有节点都可以加入生成树。请注意，算法1不会阻止孩子选择父母，因此他们没有任何方向的资金。由于这样的链接在PBT网络中不起任何作用，我们假定网络不允许它们。或者，算法1可以在允许父子关系之前检查单向链路上的资金是否非零。一旦队列为空，算法终止，表明连接图中的所有节点都有坐标。

在分布式情况下，没有中央队列。相反，节点在加入生成树时将消息发送给它们的邻居。 每个消息包括树的索引i和潜在父节点的坐标idi（节点）。 在分布式情况下开始第二阶段是棘手的，并不会对所有节点同时进行。 相反，我们选择代表初始化时间上限的时间限制τ。 如果一个节点n从一个只有一个非零权重链路的邻居节点接收到形式为(i,idi(node))的消息，则n等待时间τ。 如果n个双向链接到n的邻居直到等待期结束前都没有表明他们是潜在的父母，n选择节点作为其父母节点。

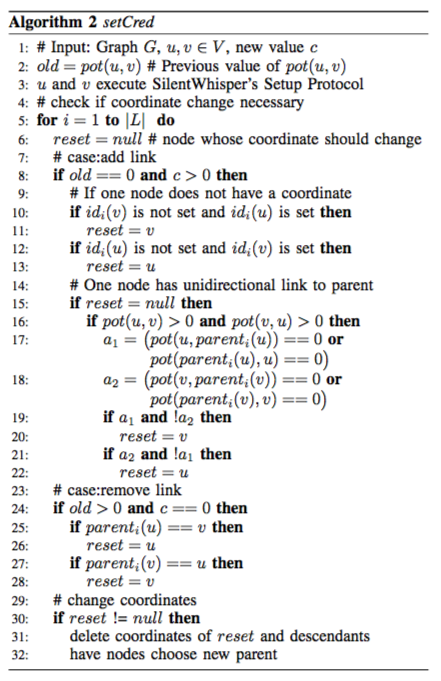
D. setCred

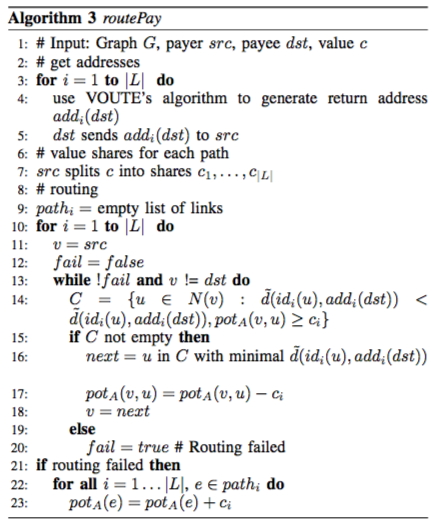
setCred对一对节点(u,v)作出反应，这些节点想要将其共享链接的值更改为c。在算法2中，u和v首先执行SilentWhispers的安全链路建立协议[15，Protocol 1]来改变链路的可用信用(u,v)。之后，该算法确定值的改变是否应该导致坐标变化。总共有三种情况表明需要进行坐标变换：

1）新的非零链接：其中一个节点还没有成为树的一部分，因此应该选择另一个作为它们的父节点（9-13行），以便能够参与路由。

2）新的非零链接：u和v共享一个双向的非零链接和（不失一般性）u只有一个单向链接到它的当前父节点。那么你应该改变它的父母v如果v有一个双向链接到它的父母（行14-22）。通过这种方式，双向连接取代了生成树中的单向链路，并增加了成功转移资金的可能性。

3）删除链接：u是v的孩子或v是u的孩子（25-28行）。子节点应该选择一个新的父节点来增加生成树中非零链接的数量和转移资金的可能性。

如果u或v中的一个改变了它的父母，那么所有的后代都会删除他们的坐标，并通知他们的邻居。之后，他们都选择一个新的父母和相应的坐标。与初始化setRoutes一致，节点首先只考虑在两个方向上都有非零链接的邻居。但是，如果一个节点没有这样的链接到任何邻居，它就会考虑一个方向的链接。如果他们有几个合适的父母，他们从坐标最短的那些候选人中随机选择他们的父母，因为通往地标的短路线减少了路径的长度[30]。选择一个新的坐标后，节点将新的坐标和树索引转发给它们的所有邻居。我们不提供伪代码，因为它与算法1非常相似。

算法2的分布式变体遵循相同的原理，但需要交换节点的消息来传递信息。u和v交换有关他们父母的链接的信息。然后他们每个单独决定是否要添加或删除另一个作为父母。从重置坐标的节点复位开始，所有后代首先通知邻居他们移除了他们对树i的旧坐标。一个节点的孩子依次移除自己的坐标并发送相应的消息。在第二阶段，节点选择新的坐标并通知邻居。由于这两个阶段可能在分布式设置中平行运行，所以节点必须确保在子孙选择新的坐标之前，他们不选

择先前的后代作为父代。然而，坐标的本质使得通过不允许节点v选择坐标包含v的前一个坐标的父母作为前缀来防止树中的这种周期变得容易。

E. routePay

routePay发现从付款人到付款人的一组路径收款人。它对应于SilentWhispers中的探测操作。算法3将该过程划分为三个步骤：i）生成接收者地址（第2-5行）; ii）将总交易价值c随机地分割在|L|路径，和 iii）找到可以传输所需值的所有嵌入的路径。

首先，收款人为所有嵌入生成匿名返回地址并将其发送给付款人（第2-5行）。其次，付款人在所有路径之间随机分配交易金额（第7行）。通过在路由之前定义每个路径的值，我们避免了SilentWhispers的昂贵的多方计算，以及ii）允许算法在多个可能的路线之间进行选择。避免最小化的多方计算也可以消除隐私泄露，因为知道每个链路上可用资金的最小值自然就会显示有关链路的信息。

第三，路由发现从v开始并且每个节点选择一个邻居成为路由上的下一个节点。在VOUTE中，每个节点将选择坐标最接近的邻居目的地，使用比较坐标的函数d与匿名回报地址。但是，这样的选择可能不适合路线资金的链接可能有可用信贷不足。结果，路由只考虑链接(v,u)有保证的可用信用的potA(v,u)至少ci （14行）。我们区分可用的信用罐(v,u)和保证可用的信贷处理并发。potA(v,u)是一个下界如果正在进行的探测操作成功，原来，potA等于实际可用的信用pot。我们不包括算法3中的初始化为多个并发routePay的exe- cutions可以影响potA(v,u)，算法可能以potA(v,u)<pot(v,u)开始。如果探测操作表明支付将通过链路(v,u)传输资金c，我们会主动通过ci（第17行）降低保证可用信用额度，以保持未来路由不会使用链路，除非最多需要保证可用信用。如果路由失败，我们再把ci加到potA(v,u)（21-23行）。如果节点v找不到比v的坐标更靠近目标坐标的邻居并且具有足够保证可用信用的链路，则路由失败。

算法3实现了正确性，正如第三节所定义的，因为i)= c 并且ii）节点总是在ith 路径上选择链接e和pot(e) ≥ potA(e) ≥ ci。

在算法3的分布式变体中，节点向路径上的下一个节点发送消息，其中包含地址addi 和部分值ci 。节点通过沿相反路径发送消息向收款人报告失败和成功。 为了解释消息丢失，如果支付操作在一定时间内不遵循探测操作，则节点也重置potA。

F.参数

上述路由算法的性能决定了几个参数。首先，数字|L|的地标决定了返回路径的数量。交易和稳定开销随|L|大致线性增加因为每个嵌入都需要路由和稳定。类似地，延迟对应于任何嵌入中最长的路由，因此可能随着|L|而增加。影响|L|在成功率上高度依赖于情景。第二个参数是a，事务尝试次数。发件人可以尝试执行一次交易。只有当所有的尝试失败时，才认为交易失败。 s在长度t1的区间内随机地均匀地选择两次连续的尝试之间的间隔。重复的事务尝试针对相同的发送者，接收者和值执行上述路由算法，但是使用不同的共享c1, . . . , c|L|。除了参数|L|，a和t1之外，地标的选择还会影响性能。通常情况下，地标是金融机构对应的节点，因此链路数量较多，可能导致生成树深度较低，性能较高。我们描述了这些参数在绩效评估中的影响。

G.隐私分析

接下来，我们认为SpeedyMurmurs实现了第III-C节提出的隐私目标。

价值隐私：非正式地，我们说PBT网络实现价值隐私，如果敌手不能确定两个非被盗用户之间的*routePay*(c,u,v)操作的值c，如果对手没有坐在任何一个涉及的路由路径。

SpeedyMurmurs是一个分布式的PBT网络，特别是routePay被定义为只涉及到付款人和收款人之间的路径中的用户。因此，如果攻击者不妥协任何这样的用户，她没有得到关于路由值的任何信息（因为点对点通信被加密），从而实现了隐私的价值。

当攻击者在付款人和收款人之间的某些路径上损坏用户时，会出现另一种情况，但不是全部。在这种情况下，我们不能阻止对手估计c。因为对于所有的i = 1...|L|我们有ci ≥ 0，知道这些值的一个子集自然揭示了关于总值c的信息，即c ≥ ci。此外，由于SpeedyMurmurs在路径中共享价值c并且仅使用正股，所以对手可以估计c为|L| ∗ ci观察ci。

付款人隐私：非正式地说，如果一个对手损害付款人和收款人之间的中间用户不能确定实际付款人之间的路线付费操作，那么PBT网络就可以实现付款人隐私。

一个坐在付款人和收款人之间的路径上的攻击者可能会收到一个匿名路由地址（即对手设法破坏了付款人的邻居）。无论如何，由于SpeedyMurmurs是一个分布式的PBT网络，对手无法确定实际的付款人是通过直接链接还是通过非受害用户路径连接到另一个用户。

我们注意到，在存在具有完整拓扑知识的对手（即，对手知道完整集合N*in*(s)）并且意识到这一点的情况下，实现付款人隐私的概念是不可能的。然而，在一个大规模的分布式系统中，攻击者似乎不太可能确定一个特定的非受害用户所建立的所有链路。因此，我们认为攻击者可能不知道特定节点的所有链路是否现实。

收款人隐私：非正式地说，如果一个在付款人和收款人之间使中介用户妥协的对手不能确定非被盗用户之间的一个路由支付的实际收款人，那么PBT网络就可以实现收款人隐私。

和以前一样，攻击者在收款人之前妥协的用户可能会转发匿名回报地址。不过，如VOUTE [31]的评估所示，匿名返回地址不会泄露网络中相应的用户。因此，攻击者不能完全确定r是否是实际的接收者，或者路由消息是针对另一个r′ 通过直接链接或者非受害用户路径连接到r的收款人。我们注意到收款人的隐私不能通过知道完整集合N*out*(r)的对手来实现，并意识到这一事实。如上所述，考虑到SpeedyMurmurs的分布式特性，我们认为这是不太可能的。

H.总结

在本节中，我们介绍了SpeedyMurmurs，它提出了一个PBT网络的隐私保护路由算法。我们将VOUTE修改为信用网络的主要贡献是：i）使用两阶段结构协议来解释单向链路的存在（算法1）; ii）确定什么时候应用的标准（算法2），iii）路径发现算法的设计，其可以基于可用信用和相邻节点的坐标自适应地选择链路，并且可以处理并发（算法3）。除了使用基于嵌入的路由之外，SpeedyMurmurs通过在路径发现之前拆分路径之间的信用来将自己与SilentWhispers区分开来。以这种方式，节点可以根据他们应该转发的信用量而不仅仅是他们的邻居到达目的地的距离作出转发决定。另一方面，在路径发现之前分配资金阻止算法考虑路径上的全部可用资金。在下一节中，我们将评估我们的设计决策对效率和效果的影响，特别是分析路由和资金分配顺序与成功率的关系。

五、表现评估

在这一节中，我们将评估Speedy-Murmurs的表现，特别是与相关工作的比较SilentWhispers的地标路由。为了简洁起见，我们编写Silent-Whispers来指示在SilentWhispers中应用的地标路由的版本。

更准确地说，我们的目标是回答以下研究问题：

•SpeedyMurmurs和SilentWhispers如何使用真实世界的数据集在成功率，延迟和开销方面执行？

•SpeedyMurmurs和SilentWhispers在三个主要领域有所不同——路由算法，随机信用分配和动态稳定。每个修改对上述性能标准有什么影响？

•这些结果与其他方法的表现相比如何？

•地标选择，树木数量和交易次数的影响是什么？

•信用网络的长期演变如何影响业绩？

我们从描述我们的仿真模型和数据集开始。之后，我们指定我们的模拟参数。我们介绍并讨论我们的结果。

通常，我们的模拟执行路由算法R并执行支付（如果成功的话）。我们把支付包含进去以实际评估由于链接变化引起的稳定性开销。然而，由于不影响路由算法及其性能，我们没有实施通常是支付的一部分的任何安全措施。特别是，我们不执行链路建立协议，以确保相邻节点就其链路的价值达成一致，之后可以通过提供签名声明来解决争端。

A.仿真模型

我们扩展了图形分析框架GTNA [32]，包括我们的信用交易机制。特别是，GTNA为路由算法和性能指标提供了模板。我们添加了PBT网络特有的功能，特别是动态更新链接权重的功能。

最初，我们的模拟根据提供的描述构建具有节点和链接的信用网络。之后，我们模拟信用网络中的一系列事件。交易清单，链路改变和生成树的周期性重新计算（只对无声 - 耳语需要），按其时间发生的顺序排列，确定了事件的顺序。在没有实际的延迟和带宽模型的情况下，我们没有在模拟中对并发进行建模。在开始下一个事件之前，模拟执行每个事件，包括生成树的结果更改。

我们实现了两种模拟模式。首先，我们考虑了一个静态信用网络。在每个步骤中，仿真执行一个事务，并随后修复生成树，如果动态稳定应用。之后，它将信用网络恢复到原来的状态。其次，我们考虑了一个随着时间的推移发展的动态网络交易，节点流失和扩展信用的修改改变了网络的结构和链路上的权重。虽然第二种模式更为现实，由于交易时的状态不同，第二种模式更难以直接比较个别交易的不同方法。

我们分别实现了SilentWhispers和SpeedyMurmurs的路由和稳定算法。但是，我们不考虑我们评估的密码细节，因为它们不会影响我们的性能指标。相反，发送者和接收者都只发送一条消息到由地标最短路径上的所有节点转发的每个地标。在我们实现SilentWhispers的过程中，每个标志都会发送一条消息到所有剩下的地标，这些地标会沿着最短的路径被转发，以解决多方计算的问题。当将基于嵌入的路由与多方计算相结合时，接收方将消息发送到所有地标。除了对每个单独的修改进行评估之外，两个设计的对齐也导致更公平地比较开销，因为原始SilentWhispers单独地发送签名链的所有元素，从而导致比发送更高的开销在一个消息。由于SilentWhispers的作者并未指定发件人如何决定分配给第i条路径的部分信用ci的数量，因此我们决定将总信用在各路径之间随机分配，并与可用的最小值相一致。换句话说，如果所有最小值的总和至少等于总交易值c，则我们首先在路径上随机地分割c。然后，我们随机将所有超过最小值的信用重新分配到剩余路径的路径上。我们重复了重新分配步骤，直到每条路径的部分信用最多等于路径上的最小信用。在模拟过程中，我们记录了导出第三部分描述的性能指标所需的所有信息。

对于SpeedyMurmurs和SilentWhispers，我们考虑以下参数：i）树数|L|，ii）节点在声明失败之前尝试执行一次事务的次数，iii）两次尝试之间的最大间隔t1相同的事务，以及iv）用于SilentWhispers的树的两个周期性重新计算之间的间隔时期。为了比较，我们表示了每个时期稳定消息中SpeedyMurmurs的稳定开销。除上述参数外，还提供了两种选择地标的方法：选择最大度的节点或选择随机节点。在这里，我们将节点的最大程度定义为在两个方向上具有可用积分的连接的数目。对于发展中的信用网络，我们选择了初始程度最高的节点。

我们实现了Ford-Fulkerson最大流算法的分布式版本[9]和仅用于比较的树。仅限树的路由只使用生成树中的链路，但选择最短路径而不是一直通过路标。对于Ford-Fulkerson，我们用分布式版本取代了集中式计算，该版本使用广度优先搜索来发现剩余流量。

B.数据集

我们从爬行PBT网络Ripple获得数据集[5]。特别是，我们从2016年11月起获得了完整网络的抓取以及自2013年1月创建以来的所有链接修改和交易。基于这些抓取，我们为我们的模拟模式（静态和演进网络）推导了数据集。在下面，我们首先描述我们的抓取方法，然后是抓取的数据的后处理。最后，我们介绍结果数据集的属性。

我们将评估限制在资金账户上：波动账户在拥有一定数量的XRP时被资助。[[2]](#footnote-2)2017年4月，用户需要20个XRP来为一个账户提供资金。在本文不考虑从一种货币转移信贷到另一个。因此，我们把所有的价值转换成美元并且删除非法定货币的所有链接和交易。后根据这三个规则清理数据集，我们导出2016年11月的信用网络C′Nov16 和交易和链接价值变化的清单，按时间顺序排序。根据所得到的交易和链接修改列表，我们在第一笔交易时产生了信用网络C0′，作为我们的第二种模式，即不断发展的网络的起点。由于我们的数据并没有显示节点何时加入和离开网络，我们在我们的初始信用网络中包括了所有的爬行节点和链接，但是将后来存在的链接(u, v)的权重设置为0。在生成树构建的过程中，这样的链接被忽略。

我们解决了我们的模型和模型之间的三个不一致之处数据集。在极少数情况下，涟漪表现出无效信用安排;即链接(u, v)，使得余额bal(v, u)超过可用信用额度cred(v,u)。通常，这种事件是由对扩展信用协议的变化而导致的。我们从数据集中删除了所有这些链接。此外，我们从数据集中删除了自我交易，因为他们不需要根据我们的模型的路由算法。最后，地标路由需要所有节点之间的路径地标，所以我们限制了对巨人的评价零件。这些处理步骤转换了初始快照C′Nov16和C′0放入我们的最终数据集CNov16和C0中。我们通过将以前的列表限制为仅涉及最终快照中的节点的条目来获得最终的事务和链接修改列表。

我们现在描述结果数据集的属性。 C0包含93,502个节点和331,096个链接，而CNov16包含67,149个节点和199,574个链接。造成差异的原因是C0包含了所有活动链路及其相邻节点超过3年的时间，而CNov16是在特定日期的网络快照。我们的最终交易清单分别为C0和CNov16分别有970,472和692,737个条目。我们记录了总共652,216个不断发展的网络C0的链接修改。数据集和代码是公开可用的。[[3]](#footnote-3)

C.模拟设置

我们的第一个仿真设置基于快照CNov16实现了静态仿真模式。我们重复模拟20次，每次运行使用一组不同的50,000次交易。我们从所有使用Ford-Fulkerson的交易中选择了这些交易，按照运行编号进行伪随机分配，总计331,642笔交易。然后，我们评估了路由算法（地标路由或基于嵌入的），路径的信用分配（多方计算或随机分配）和稳定算法（定期或按需）的所有8种可能的组合，|L| = 3和a = 2。我们选择epoch = 1000，这意味着我们重新计算生成1000个交易的树。我们选择重排队间隔为tl = 2·epoch。对于地标选择，我们考虑了两个选项：随机和最高度。注意随机选择在运行数中是确定性的，确保在相同情况下所有方法的可比性。为了与相关方法进行比较，我们评估了两个版本的纯树路由，使用SilentWhispers的多方计算和第一版本的周期性稳定和SpeedyMurmurs的随机信贷分配和按需稳定。然后我们评估了不同参数对SilentWhispers和SpeedyMurmurs的影响。我们改变地标|L|的数量在1到7之间，尝试次数在1到10之间。

我们的第二个模拟设置实现了不同算法下的网络演变：Ford-Fulkerson，Silent-Whispers和SpeedyMurmurs。从初始网络C0开始，仿真启动交易并根据数据集改变链接值。对于SilentWhispers和SpeedyMurmurs，我们设置|L| = 3, a = 2, epoch = 1000δAv, and tl = 2δAv ，δAv表示两次交易之间的平均时间。以这种方式，一个纪元大致相当于一天。我们选择了SpeedyMurmurs和SilentWhispers最高的地标。由于Ford-Fulkerson是一个确定性的算法，我们只执行一次，但平均我们的结果SpeedyMurmurs和SilentWhispers超过20次运行。

D.结果

我们从比较静态模拟设置的各种算法开始。表I显示了三种对SilentWhispers提出的修改的不同组合的结果，以及我们的树型路由和Ford-Fulkerson的实现。请注意，Ford-Fulkerson是一种确定性算法，但其延迟和开销会随着交易之间的交易集而变化。

正如所料，由于在树的不同分支之间找到捷径，贪婪嵌入导致更短的路径。因此，所有使用贪婪嵌入的设置都比相应的基于地标的协议表现出更低的延迟和事务开销。事实上，贪婪嵌入将路径长度和事务开销减少了近2倍。贪婪嵌入也增加了成功率，因为​​路径越短，遇到可用信用低的链接的可能性越小。

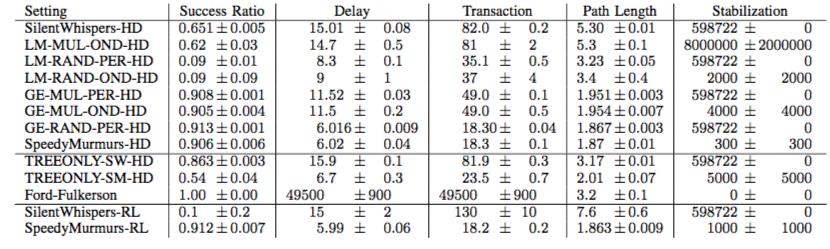
随机分配信贷的影响不那么明确：虽然消除将地标引入路由选择过程的需要降低了所有参数设置的延迟和交易开销，但对基于嵌入和地标的成功率的影响是不同的路由。与地标路线相结合，随机信贷分配导致成功率从60％以上下降到8％。成功率低的原因是遇到至少一个信用不足的链接以满足随机分配的高概率。相比之下，贪婪的嵌入表现出更短的路径和在几个邻居之间进行选择的灵活性。这两个性质抵消了随机信用分配的不利影响，因此贪婪嵌入与随机分配相结合，与多方计算相结合的成功率为91％。

表1：不同交易方案的性能，改变路由算法（T-Tree，GE-贪婪嵌入），稳定方法（PER-周期，OND-on-demand），路径上的信用分配（MUL- （RAND-随机）和地标选择（HD-最高度，RL-随机地标），其中五个度量：成功率：成功交易的比例（越高越好），延迟：最长的消息链（越低越好 ）事务：每个事务发送的消息（越低越好），路径长度：发送者和接收者之间发现路径的长度（越低越好），稳定性：用于稳定每个时期发送的树的消息（越低越好）。 SilentWhispers对应于LM-MUL-PER的设置，而SpeedyMurmurs是GE-RAND-OND。

按需稳定性大大降低了稳定性开销（表1中的稳定性缩写）：定期重建生成树导致每个时期超过50万条消息，按需稳定仅需要几千条消息，如表1中的最后一列比较使用“PER”和“OND”的算法。按需稳定化引起了高的变化，因为接近生成树根的链路的值很少下降到0，但是在这些罕见事件中产生了巨大的开销。仿真显示了按需稳定的明显优势。我们承认按需稳定的显着优势部分是由于缺乏链路价值的变化和静态仿真的实际动态。在本节的第二部分中，我们评估动态环境中的稳定开销。

其次，我们将SilentWhispers和SpeedyMurmurs与仅有树的路由和Ford-Fulkerson进行了比较，结果如表1所示。正如预期的那样，Ford-Fulkerson表现出令人望而却步的延迟和交易开销。 Ford-Fulkerson也导致了较长的平均路径长度，这一事实起初似乎是非常有效的。然而，结果是Ford-Fulkerson发现了其他途径未能发现的可用信用最大化的长路径的副作用。如图1所示，并在第2节详细说明。 只有树的路由在生成树中找到最短路由，可能没有经过地标，但不包括树中不包含的链路。因此，只有树的路由是SilentWhispers的路由算法和基于嵌入的路由之间的折衷。因此，仅使用树链接时的性能结果介于SilentWhispers和SpeedyMurmurs的性能之间。

接下来，我们评估不同配置参数对性能的影响。如表1的最后两行所示，选择随机地标不会对SpeedyMurmurs的性能产生影响，但会由于存在较长路径而无法连接，因此降低了SilentWhispers的性能。相反，增加树|L|的数量如图2a所示，积极影响SpeedyMurmurs和SilentWhispers的成功率。观察到的成功率下降的原因是至少有一条路径信贷不足的可能性增加。增加|L|进一步增加了延迟，如图2b所示。 SilentWhispers的影响更为明显，因为地标必须等到多方计算的所有消息到达。如图2c所示，尝试次数对成功率有轻微的正面影响。然而，由于事务开销在尝试次数上是线性的，所以略微增加可能不保证多次尝试。

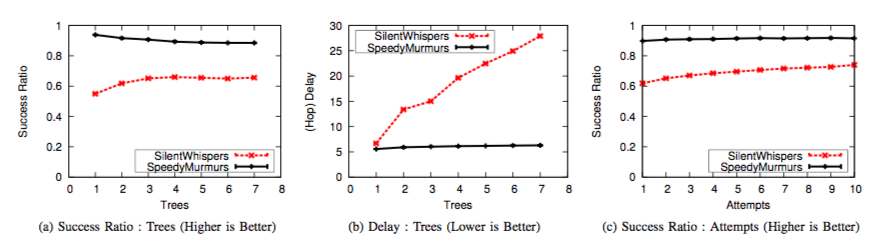
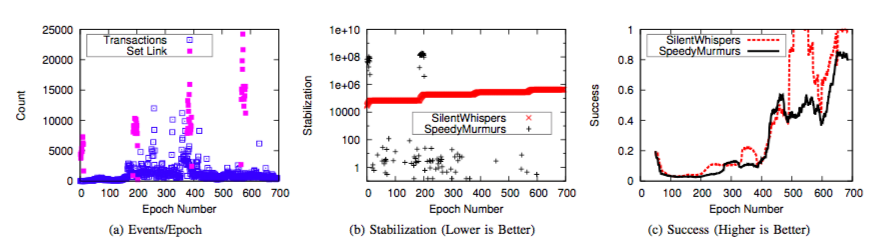
对于所有的算法，只有Ford-Fulkerson，成功率远远低于100％。可以肯定的是，很多用户可能不愿意接受10％或更高的失败率。请注意，路由失败不会减少任何用户的资金，所以与路由失败相关的资金不会丢失。而且，在非静态的环境下，用户可以在网络已经充分改变后再重试事务。如果既不失败也不等待，我们可以将Ford-Fulkerson应用于失败。通过将需要Ford-Fulkerson的交易减少到10％，相对于仅依靠Ford-Fulkerson的网络，由于之前使用SpeedyMurmurs而导致延迟略微增加，我们仍然大大提高了效率。另外，我们希望随着PBT网络的连通性和可用资金的增加，越来越受欢迎，这种情况的增加超出了目前Ripple网络的状态，这势必会有更高的成功率。

图3：根据波动交易和2013年至2016年每个时段的链接变化，动态设置SpeedyMurmurs和SilentWhispers的比较; 成功率以Ford-Fulkerson算法的实际成功率与成功率之比作为基准来计算; 对于c），我们提出了移动平均超过50个时期，以提高可读性

图2：SilentWhispers vs. SpeedyMurmurs：并行树的数量和尝试执行事务的影响

最后，我们评估动力学对SpeedyMurmurs和SilentWhispers的性能的影响。如上所述，动态的影响特别有助于决定按需稳定是否确实比周期性稳定更有效。为了更好地理解我们的结果背后的原因，图3a显示了Ripple数据集的每个历元的交易数量和链接变化。虽然交易次数在三年内没有太大的变化，但在短时间内连接创作和修改是频繁的，但在剩下的观察期间很少见。链路变化的频率直接与SpeedyMurmurs的稳定开销有关，如图3b所示。稳定开销通常低于每个时期100个消息，而在经常变化期间，开销增加到约109个消息。请注意，只有四批链接变化的前两个导致对稳定的需求急剧增加。在前两批之后，生成树已经形成并且新的链接添加主要创建了不需要改变树的捷径。相比之下，SilentWhispers的稳定开销只取决于网络中的边缘数量，因此随着时间的推移图形增加。在频繁变化的间隔期间，SilentWhispers的稳定开销远远低于SpeedyMurmurs的稳定开销。然而，在“正常”操作期间，SilentWhispers的稳定性开销超过了SpeedyMurmurs开销的2个数量级。我们评估了与Ford-Fulkerson相关的成功，并将Ford-Fulkerson在各个时代的实际成功率除以相应时期的成功率。从图3c可以看出，如果另一种路由算法显示出更高的成功率，则成功率可能会超过1。请注意，由于不同的路由算法导致不同的支付并因此导致不同的网络状态这一事实，所以成功率更高。不同的网络状态意味着一组不同的可能交易，所以Ford-Fulkerson的交易可能会失败，但对于SilentWhispers或SpeedyMurmurs来说却是成功的。相比之下，SpeedyMurmurs和Silent-Whispers在大部分时间都取得了相似的成功率。然而，在模拟间隔结束时，SilentWhispers胜过了SpeedyMurmurs。如图3a所示，成功的突然增加与许多环节的增加或改变相关。附加链接增加了图的密度，导致路径更短，因此成功概率更高。 SilentWhispers实现比SpeedyMurmurs更高的成功率的事实可能是由于树结构：SilentWhispers保持广度优先搜索树，而SpeedyMurmurs最初构造广度优先搜索树，但是如果新邻居提供了一个搜索树，则不会改变节点的父亲较短的路径到根。通往根源的较长路径可能会对成功的可能性产生负面影响。由于所有考虑的算法的实际成功率在后期较低（例如，经常低于5％），其结果可能是我们的数据集和后处理方法的块效应。

总之，我们回答了五个最初的研究问题如下：

•SpeedyMurmurs在静态方案的所有考虑指标方面都比SilentWhispers更高。

•按需稳定和基于嵌入的路由对所有5个性能指标都有正面影响。相反，使用随机信贷分配可能会略微降低成功率。但是，当与其他两种修改结合使用时，效果大多被否定。

•Ford-Fulkerson通常比SpeedyMurmurs和SilentWhispers取得更高的成功。然而，该算法导致了巨大的事务开销，超过了其他算法的开销2到3个数量级。

•增加树数或尝试进行交易并不能显着提高SpeedyMurmurs的成功率，但却增加了开销。

•PBT网络的演变会影响所提出的算法的性能。稳定的开销和成功率取决于交易和链接变化的频率。

动态评估建议从事于一个替代生成树维护算法的设计。具体而言，结果提出了按需定期和定期稳定之间动态切换的合适标准问题。事实上，由于SilentWhispers在频繁更换期间效率更高，但是导致更高的开销，所以这种交换机制可以进一步减少通信开销，从而提高可扩展性。

六、相关工作

在一个信用网络中最大化可能的交易集合是NP-hard [11]。相反，许多现有系统选择一次考虑一个事务，并将最大流方法[9]用作路由算法。然而，现有的算法[8]运行在O(V 3)或O(V 2 log(E))时间，因此不能扩展到越来越多的用户和事务[26,35]。

Prihodko等人最近提出了Flare [27]，这是比特币闪电网络的一种路由算法，比特币用户中的比特币支付渠道网络可以实现脱机交易[25]。在Flare中，所有节点都跟踪它们的k邻域;即至多k的跳距离处的节点以及它们之间的所有链路。另外，每个节点维护到一组附近的信标节点的路径。

该路由算法揭示了k邻域中所有链路的权重，通常k≥3。这导致隐私问题，因为两个用户之间的链路权重暴露给除这两个用户以外的用户。此外，节点将所有更新传播到k邻域，这意味着每个信用变化可能导致数百个消息，这对于频繁的交易是非常低效的，并因此改变可用信用。

Canal [35]提出了树型路由的第一个有效实现，用于寻找信用网络中的路径。可信中央方计算发送方和接收方之间生成树中的最短路径。如果这些路径提供足够的信用来解决交易，则路由成功终止。否则，它失败。面对网络动态，中央服务器不断重新计算生成树。

这种方法具有严重的隐私影响，因为中央服务器维护完整的PBT网络，并可以跟踪每次付款所采用的路线。而且，中央服务器构成了单点故障。

PrivPay [21]通过在中央服务器上使用可信硬件来增加Canal的隐私。然而，PrivPay依赖于类似的具有里程碑意义的技术，也是一个集中的解决方案，因此可扩展性仍然很低，单点故障的问题仍然没有解决。此外，PrivPay论文首次介绍了信用网络中的价值隐私和发送者/接收者隐私的概念。在这项工作中，我们定义了在PBT网络中路由的隐私概念，作为一个构建模块，不仅适用于信用网络，也适用于任何PBT网络。

SilentWhispers [15]在完全分布式的信用网络中使用地标路由。发送者和接收者都是在地标的方向发送消息，这些地标构成了集合节点。换句话说，SilentWhispers中的路径是发送者到地标的路径和从地标到接收者的路径的连接。即使发送者和接收者碰巧在同一分支中，所有路径都通过一个地标，可能导致性能问题。然而，正如我们在本文中讨论的那样，本文提出的路由算法SpeedyMurmurs比在SilentWhispers中提出的路由方法更胜一筹，同时实现了隐私的利益概念。

Malavolta等人[16]最近提出了Rayo和Fulgor，两个提供隐私和并发之间的必要权衡的支付通道网络（即PBT网络）。他们对并发性的研究可以用来扩展SpeedyMurmurs的并发性。不过，他们没有解决路径选择问题。因此，SpeedyMurmurs成为与Rayo和Fulgor互补的绝佳候选人。

总结：现有的路由方法经常忽视隐私。他们大多需要集中或共享公共信息; SilentWhispers是唯一专注于隐私的现有分布式系统。但是，它依赖于分布式地标路由技术，可能效率低下。我们深入的性能和隐私评估表明，与最先进的路由方法相比，SpeedyMurmurs提供了更高的整体性能，同时实现了隐私的利益概念。

七、结论和未来的工作

在这项工作中，我们设计了SpeedyMurmurs，一个高效的完全分布式支付系统的路由算法。 我们广泛的模拟研究和分析表明，SpeedyMurmurs是高效的，实现了高成功的可能性，同时仍然提供针对强大的网络攻击者的交易隐私。 由于交易隐私对于PBT应用来说至关重要，SpeedyMurmurs是信用网络和支付通道网络的理想路由算法，也是新兴的区块链间协议。

由于我们的研究结果表明按需和周期性稳定适合于PBT网络演进的不同阶段，未来的工作可以通过调查按需和周期性稳定之间的动态切换来扩展我们的结果。

致谢

滑铁卢大学的CrySP RIPPLE工具使这项工作受益匪浅。

参考文献

[1]  Raiden network. http://raiden.network/. Accessed July 2017.

[2]  Real-time gross settlement. https://en.wikipedia.org/wiki/Real- time gross settlement. Accessed July 2017.

[3]  Ripple website. https://ripple.com/. Accessed July 2017.

[4]  Stellar website. https://www.stellar.org/. Accessed July 2017.

[5]  Frederik Armknecht, Ghassan O Karame, Avikarsha Mandal, Franck Youssef, and Erik Zenner. Ripple: Overview and Outlook. In *Interna- tional Conference on Trust and Trustworthy Computing*, pages 163–180. Springer, 2015.

[6]  Kyle Croman, Christian Decker, Ittay Eyal, Adem Efe Gencer, Ari Juels, Ahmed Kosba, Andrew Miller, Prateek Saxena, Elaine Shi, Emin Gu ̈n Sirer, Dawn Song, and Roger Wattenhofer. *On Scaling Decentralized Blockchains*, pages 106–125. Springer Berlin Heidelberg, Berlin, Heidelberg, 2016.

[7]  Dimitri DeFigueiredo and Earl T. Barr. TrustDavis: A Non-Exploitable Online Reputation System. In *E-Commerce Technology*, 2005.

[8]  Yefim Dinitz. Dinitz’s Algorithm: The Original Version and Even’s Version. In *Theoretical Computer Science*. 2006.

[9]  Lester R Ford and Delbert R Fulkerson. Maximal Flow Through a Network. *Canadian journal of Mathematics*, 8(3), 1956.

[10]  Ethereum Foundation. Ethereum Project. https://www.ethereum.org/. Accessed July 2017.

[11]  Arpita Ghosh, Mohammad Mahdian, Daniel M. Reeves, David M. Pennock, and Ryan Fugger. Mechanism design on trust networks. In *Internet and Network Economics, Third International Workshop, WINE 2007, San Diego, CA, USA, December 12-14, 2007, Proceedings*, pages 257–268, 2007.

[12]  Andreas Hoefer, Stefanie Roos, and Thorsten Strufe. Greedy Embed- ding, Routing and Content Addressing for Darknets. In *NetSys*, 2013.

[13]  Amrit Kumar, Cle ́ment Fischer, Shruti Tople, and Prateek Saxena. A traceability analysis of Monero’s blockchain. Cryptology ePrint Archive, Report 2017/338, 2017. http://eprint.iacr.org/2017/338.

[14]  Lightning Network Team. Atomic cross-chain trading. https://en. bitcoin.it/wiki/Atomic cross-chain trading. Accessed July 2017.

[15]  Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, and Matteo Maffei. SilentWhispers: Enforcing Security and Privacy in Credit Networks. In *NDSS*, 2017.

[16]  Giulio Malavolta, Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Srivatsan Ravi. Concurrency and Privacy with Payment-Channel Networks. In *CCS*, 2017.

[17]  Sarah Meiklejohn, Marjori Pomarole, Grant Jordan, Kirill Levchenko, Damon McCoy, Geoffrey M. Voelker, and Stefan Savage. A fistful of bitcoins: Characterizing payments among men with no names. In *Pro- ceedings of the 2013 Conference on Internet Measurement Conference*, IMC ’13, pages 127–140, New York, NY, USA, 2013. ACM.

[18]  Andrew Miller, Malte Mo ̈ser, Kevin Lee, and Arvind Narayanan. An empirical analysis of linkability in the Monero blockchain. *CoRR*, abs/1704.04299, 2017.

[19]  Alan Mislove, Gaurav Oberoi, Ansley Post, Charles Reis, Peter Dr- uschel, and Dan S. Wallach. AP3: Cooperative, decentralized anony- mous communication. In *ACM SIGOPS European Workshop*, EW 11, 2004.

[20]  Prateek Mittal and Nikita Borisov. Shadowwalker: Peer-to-peer anony- mous communication using redundant structured topologies. In *ACM CCS*, pages 161–172, 2009.

[21]  Pedro Moreno-Sanchez, Aniket Kate, Matteo Maffei, and Kim Pecina. Privacy Preserving Payments in Credit Networks: Enabling trust with privacy in online marketplaces. In *NDSS*, 2015. [22]  Pedro Moreno-Sanchez, Muhammad Bilal Zafar, and Aniket Kate. Listening to Whispers of Ripple: Linking Wallets and Deanonymizing Transactions in the Ripple Network. *PoPETS*, 4/2016, 2016.

[23]  Satoshi Nakamoto. Bitcoin: A Peer-to-Peer Electronic Cash Sys- tem. http://fastbull.dl.sourceforge.net/project/bitcoin/Design%20Paper/ bitcoin.pdf/bitcoin.pdf, 2011.

[24]  Christos H Papadimitriou and David Ratajczak. On a Conjecture Related to Geometric Routing. In *Algorithmic Aspects of Wireless Sensor Networks*, 2004.

[25]  Joseph Poon and Thaddeus Dryja. The Bitcoin Lightning Net- work: Scalable Off-chain Instant Payments. https://lightning.network/ lightning-network-paper.pdf, 2015.

[26]  Ansley Post, Vijit Shah, and Alan Mislove. Bazaar: Strengthening user reputations in online marketplaces. In *NSDI*, 2011.

[27]  Pavel Prihodko, Slava Zhigulin, Mykola Sahno, Aleksei Ostrovskiy, and Olaoluwa Osuntokun. Flare: An Approach to Routing in Lightning Net- work. http://bitfury.com/content/5-white-papers-research/whitepaper flare an approach to routing in lightning network 7 7 2016.pdf, 2016.

[28]  Fergal Reid and Martin Harrigan. *An Analysis of Anonymity in the Bitcoin System*, pages 197–223. Springer New York, New York, NY, 2013.

[29]  Michael K. Reiter and Aviel D. Rubin. Crowds: Anonymity for web transactions. *ACM Trans. Inf. Syst. Secur.*, 1(1):66–92, November 1998.

[30]  Stefanie Roos, Martin Beck, and Thorsten Strufe. Anonymous Ad- dresses for Efficient and Resilient Routing in F2F Overlays. In *INFOCOM*, 2016.

[31]  Stefanie Roos, Martin Beck, and Thorsten Strufe. VOUTE-Virtual Overlays Using Tree Embeddings. *CoRR*, abs/1601.06119, 2016.

[32]  Benjamin Schiller and Thorsten Strufe. GTNA 2.0-A Framework for Rapid Prototyping and Evaluation of Routing Algorithms. In *Summer Computer Simulation Conference*, 2013.

[33]  Stefan Thomas and Evan Schwartz. A Protocol for Interledger Pay- ments. https://interledger.org/interledger.pdf, 2015.

[34]  P. F. Tsuchiya. The Landmark Hierarchy: A New Hierarchy for Routing in Very Large Networks. *SIGCOMM*, 18(4), 1988.

[35]  Bimal Viswanath, Mainack Mondal, Krishna P. Gummadi, Alan Mis- love, and Ansley Post. Canal: Scaling Social Network-based Sybil Tolerance Schemes. In *EuroSys*, 2012.

[36]  XRP Chat. Interledger payment through 7 ledgers. https://www. xrpchat.com/topic/5660-interledger-payment-through-7-ledgers/, 2017. Accessed July 2017.

1. 这固有地假定一个支付协议，在找到足够信用的路由之后总是成功的。 我们从而抽象得出支付协议的细节。 [↑](#footnote-ref-1)
2. XRP是Ripple货币的象征。 [↑](#footnote-ref-2)
3. 我们会根据要求通过计划主席匿名提供给审稿人。 [↑](#footnote-ref-3)