arXiv:2304.06317v1 [cs.DS] 13 Apr 2023

HYBRID 分布式模型中的普适最优确定性广播

Yi-JunChang Oren Hecht 新加坡国立大学
Technioncyijun@nus.edu.sg hecht.oren@campus.technion.ac.ilDean
Leitersdorf 新加坡国立大学dean.leitersdorf@gmail.com

摘要

在理论计算机科学中，为问题证明存在性下界是一种常见做法，即存在一族病态输入，使得任何算法都无法取得优于该下界的表现。然而，在许多情况下，大多数受关注的输入都可以更高效地求解，这催生了普适最优性这一概念。粗略地说，普适最优算法对于给定输入的运行速度，与专门针对该输入设计的最佳算法一样快。

关于分布式环境中普适最优算法是否存在的问题，Garay、Kutten 和 Peleg 首次在 FOCS '93 上提出。近期，一系列研究重新推动了这一方向的发展，其中包括 Haeupler、Wajc 和 Zuzic 在 STOC '21 上发表的开创性工作；该工作解决了 supported CONGEST 模型中的一些延续数十年的问题。

我们研究 Hybrid 分布式模型，该模型分析结合全局和局部通信的网络。近期，人们投入了大量精力来解决与距离相关的问题，例如 Hybrid 中的全点对最短路径（APSP），最终得到了一个用于精确全点对最短路径的 $\tilde{\Theta}(n^{1/2})$ 轮算法。然而，根据定义，Hybrid 中的每个问题都能在 D轮内求解，其中 D 是图的直径。这表明，尽管 $\tilde{\Theta}(n^{1/2})$ 轮对于 APSP 而言在存在性意义上是最优的，但远非普适最优。

我们通过提出一种基础工具，给出了 Hybrid 中首个 普适最优算法。该工具能够以普适最优的轮数确定性地解决任意 广播问题。具体而言，我们考虑 k-信息传播问题：给定一个 n 节点图G 以及任意分布在 G各处的一组 k 条消息 M，要求每个节点都获知所有 M。对于任意图 G、任意值 k以及 M 在 G上的任意分布，我们给出了一个普适下界和一个与之匹配的确定性上界。

这一广播工具为研究 Hybrid 中的普适最优算法开辟了令人振奋的新方向。举例来说，我们利用它设计了用于一般图中近似求解全点对最短路径问题，以及在稀疏图中精确求解全点对最短路径问题的算法；这些算法具有普适最优性，因为它们达到了仅需获知图中节点（可能随机生成的）标识符这一任务的下界，而这些标识符是输出最短路径距离所必需的。

目录

1 引言 21.1 我们的贡献.4 1.1.1 普遍最优广播.4 1.1.2 T_k 的分析.7 1.1.3 应用.7 1.1.4 割的近似.9 1.2 更多相关工作.9

2 预备知识 **10**

3 普适最优广播 11
3.1 T_k 的基本性质.11 3.2 下界.12 3.3 上界.14 3.4 普适最优聚合.18

4 应用 19
4.1 无权图上的 $(1+\epsilon)$-近似全点对最短路径.19 4.2 加权图中的全点对最短路径近似.20 4.3 通过学习谱稀疏化器近似割.23

5 不同图族上的 T_k 估计 23
5.1 路径图和环图.23 5.2 方格图.24

Hybrid 中的初步算法 30
A.1 确定性虚拟树构造.30 A.2 T_k 的基本性质.31

1 引言

我们研究分布式计算的关键模型 Hybrid，并着手解决在图上传播信息这一基础问题——以最佳可能的复杂度确定性地解决该问题最一般的变体。Hybrid 模型的分布式计算抽象了常见的实际分布式网络，从而为开展理论研究提供了一个框架，并且可以直接应用于现代数据中心和分布式网络 [16, 31, 42, 47]。在任何分布式网络中，广播信息都是一项基础任务；它既可以作为最终目标本身而具有意义（例如广播网络更新、故障通知等），也可以作为解决其他问题的基本构件（例如计算节点之间的路径和距离）。

我们解决了最一般形式的广播问题：有一些 k 消息，其值可以是任意 *</style}*值，并且最初以任意 *</style}*方式分布在图中（即，所有消息可以从一个节点开始，也可能分散开来，使每个节点各持有一条消息，等等）；目标是让图中的每个节点都获知所有 消息。我们以普遍最优的方式解决了这个问题，这意味着我们的算法与任何可能的最优算法一样快，即使后者事先知道图拓扑和消息的初始位置（但不知道消息内容）。本质上，我们设计了一种通用算法，适用于任意图和任意消息初始分布；而且，不可能找到更快的算法，包括针对特定图和消息初始分布量身定制的算法。最后，我们还使用广播工具来近似计算图中的距离和割。

Hybrid 网络。 Hybrid 模型 [38]研究这样一种分布式网络：物理距离较近的节点可以通过高带宽本地通信链路进行通信，同时还存在低带宽全局通信链路，用于在网络中物理距离较远的区域之间发送少量信息。这类混合网络出现在现实世界的应用中，包括无线通信受限的数据中心，以及高带宽、短距离的有线通信 [16, 31]。其他例子还包括蜂窝网络：设备可以通过高带宽链路在本地环境中进行通信（例如，附近智能手机之间使用蓝牙、WiFi Direct 或 LTE Direct 通信），同时也可以通过带宽较低的蜂窝基础设施进行全局通信 [34]。

迄今为止，通过 Hybrid 模型对混合网络进行的理论研究主要探索了距离计算任务，例如 k-源最短路径，尤其是单源最短路径（SSSP，$k = 1$）和全点对最短路径（APSP，$k = n$）、直径计算等 [3, 9, 10, 12, 13, 17, 39]。关键观察在于，高带宽本地网络与低带宽全局网络相结合，能够以远快于单独使用任一网络的速度解决问题——例如，APSP 在仅使用本地˜网络或全局网络时均需要 $\tilde{\Omega}(n)$ 轮[1] [6]，，而在 Hybrid 模型中同时使用这两个网络，则可在 $\Theta(\sqrt{n})$ 轮内解决 [6, 38]。最近，新的研究方向开始探讨路由方案和距离预言机 [12, 39]的应用。这些是高效数据包转发等应用的基础工具，而此类应用构成了现代互联网的骨干。

Hybrid 中的广播 Hybrid。 混合网络的一个基本用例是信息广播。信息广播本身就是一个有趣的最终目标，例如在数据中心宣布故障、策略变更或其他控制消息。此外，广播本身也可以成为解决其他问题的有用工具，这一点从围绕

[1]符号 $\tilde{O}(\cdot)$ $\tilde{\Omega}(\cdot)$ $\tilde{\Theta}(\cdot)$ 隐去了多对数因子。

广播拥塞团模型 (BCC) [15]– 在 BCC 中，在每一轮中每个节点都可以向其他每个节点广播一条消息，仅使用这一基本原语，已有研究表明许多问题都可以得到解决 [7, 8, 11, 32, 33, 41]。

我们研究最一般的广播变体：图中任意分布着一些 k 消息，并且希望图中的所有节点都能获知这些消息。这些消息最初可以以任意配置开始——即全部来自某个特定节点，或分散在图中的不同节点之间。我们针对任意值 k 以及任意初始消息分布解决了这一问题。作为推论，通过设置 $k = n$，我们展示了在 Hybrid 中对 BCC 的模拟。

普适最优性。 Hybrid 中迄今为止的所有研究都集中于存在性 下界，这意味着存在一个病态图族，使得任何算法都无法做到优于某个下界。例如˜，证明全点对最短路径具有 $\Theta(\sqrt{n})$ 的下界和上界，方法是展示一个图族˜，该图族要求至少 $\Omega(\sqrt{n})$ 轮次才能解决全点对最短路径，并给出一个相匹配的上界。然而，显然，Hybrid 中的任何问题都可以在 D 轮次内解决，其中 D 是网络直径，只需使用高带宽的本地网络即可。在实际例子中，许多网络的直径相对于节点数量 n而言都很小，这使得当前最先进的存在性最优算法在实践中并不实用。作为进一步的例子， [3, 17], 中已经证明，存在一些图，可以将全点对最短路径的求解速度提高几个数量级，仅需 $O(\log n)$ 轮次；与存在性下界相比，许多受关注的网络都能提供快得多的算法。

因此，一个值得追求的目标是普适最优算法；这一概念最早由 [18]在 FOCS '93 上针对分布式场景提出。粗略地说，普适最优算法在任意图上的运行速度都尽可能快，而不只是在最坏情况图上（下文很快会给出正式定义）。在 [20] 中，人们在广为人知的 CONGEST 分布式模型中迈出了探索非最坏情况算法的第一步，并提出了低拥塞快捷方式框架。此后，一系列颇具影响力的工作相继问世 [19, 21–23,25–29, 35, 44, 50, 51],，最终由 [30]的工作给出了普适最优性的定义。

我们展示了 Hybrid 中的首个普适最优算法。我们提出了一个参数 $T_k(G)$，适用于任意图 G，并证明它是将 k 令牌广播给所有 G 的普适下界。也就是说，˜在特定图 G 上，即使算法已知完整的 G 拓扑以及 k消息的初始位置（但不知道其内容），也无法在少于 $\Omega(T_k(G))$ 轮内解决广播 k 消息问题。我们用一个单一确定性算法补充了这一上界；该算法在任意图 G 上运行时，需要 $O(T_k(G))$ 轮即可解决广播问题。我们强调，我们算法的复杂度不依赖于原始令牌分布。

本质上，普适最优算法是分布式研究中的重要一步：它们既提出了理论挑战，也弥合了理论与实践之间的鸿沟，展示了适用于任何特定情形（包括现实世界图）的最优算法。我们相信，在 Hybrid 中为此类研究奠定基础，将为未来更多令人振奋的成果打开大门。

路线图。 接下来，我们概述本文的贡献，以及为展示这些贡献而开发的技术。在第 3 节中，我们给出广播的普适下界和与之匹配的上界，并讨论一种类似的、用于聚合 k 函数的算法。在第 4 节中，我们利用广播工具来近似全点对最短路径和各种割。最后，在第 5 节中，我们计算某些图族中 $T_k(G)$ 的界，以初步展示 $T_k(G)$ 与其他图参数（例如n 和 D）之间的关系。这也表明，在这类图族中，普适最优算法显著优于现有的最先进的、存在性最优算法。

1.1 我们的贡献

We n现描述我们的结果，并简要解释我们算法背后的技术 s.

在继续之前，我们先粗略定义 Hybrid（正式定义将在后文给出）。给定一个初始输入图 $G = (V, E)$，接下来按称为轮次的同步时间步进行。在每一轮中，V 中任意两个之间有 E 边的节点，都可以通过本地网络交换任意数量的比特。此外，每个节点 $v \in V$ 可以在V 中任意选择 $\log n$ 个节点，并通过全局网络分别向它们发送一条（可能各不相同的）$O(\log n)$ 比特消息。每轮中，V 中的每个节点通过全局网络最多只能接收 $\log n$ 条消息。

Hybrid 的一个限制更严格的变体是 Hybrid$_0$ （见 [4]中的 NCC$_0$ ），其中每个节点都有一个任意的 $O(\log n)$ 比特标识符；每个节点最初只知道自己和邻居的标识符，而不知道其他节点的标识符。Hybrid 与 Hybrid$_0$ 之间的关键区别在于，在 Hybrid$_0$ 中，节点必须先获知另一个节点的标识符，才能通过全局网络向该节点发送消息。

显然，如果某个算法在 Hybrid$_0$, 中有效，那么它在 Hybrid 中也有效；如果某个下界在 Hybrid 中成立，那么它在 Hybrid$_0$ 中也成立。

1.1.1 通用最优广播

我们的主要研究问题是以下广播问题。

定义 1 (k-信息传播)**：** 给定任意一组消息 M，其中 $k = |M|$，且每条 $m \in M$最初仅由图中的一个节点知晓，k-信息传播 问题要求所有消息 M 都被图中的每个节点知晓。

令牌传播的一个类似变体在定义该模型的论文 [6]中被作为最基本的 Hybrid 通信原语之一提出。在该论文中，他们将任意节点最初拥有的令牌数量限制为某个值 ℓ，并给出了一种在 $\tilde{O}(\sqrt{k+\ell})$ 轮内运行的随机化算法。最近，[3] 通过展示一种在$\tilde{O}(\sqrt{k})$ 轮内运行的确定性算法，移除了对$-\ell$ 的限制，但他们要求 $k \geq n$。在我们的研究中，我们解决了该问题最一般的变体，不对 k 或令牌在图中的初始分布作任何限制。

我们首先定义图的广播质量 。对于任意节点 v，将 $B_t(v)$ 称为以 v 为中心、半径为 t 的球——也就是说，所有能够通过至多 t 条边的路径到达 v 的节点。

定义 2 (广播质量**)。** 给定一个图 $G = (V, E)$、值 k 和节点 v，令

$$T_k(v) = \min\{\{t \mid |B_t(v)| \geq k/t\} \cup \{D\}\}$$

并令 $T_k(G)$ =最大$_{v \in V}$ $T_k(v)$。

当 G 从上下文中明确时，我们写作 T_k 而不是 T_k（G）。注意，T_k 实际上刻画了 G的幂图集合的一个性质。一个 G 的幂图 G^t 与 G具有相同的节点集，并且当 v 与 u 在 G 中之间存在一条至多包含 t 条边的路径时，具有一条边 $e = \{v, u\}$ 。 T_k 本质上是满足 G^t 中的最小度至少为 k/t的最小值 $t = T_k$ 。

我们现在继续展示 Hybrid 中解决 k-信息传播问题。这揭示了几个有趣的性质。首先，它表明在 Hybrid 中，k-信息传播的复杂度仅取决于图拓扑，而与令牌的原始分布无关。此外，它还表明，幂图集合的最小度数是 Hybrid 中的一项基本性质。

直观而言，这是合理的，因为在 Hybrid 中，t 轮次内，G 中的每个节点都能与其在 G^t中的每个邻居通信。同时，在这 t 轮次中，这些节点中的每个节点都可以接收

$O(t \cdot \log\ n)$ 条消息通过全局网络发送。因此， G^t 中节点的最小度决定了该节点在 t 轮次内结合全局网络和本地网络所能接收的信息量上界。相比之下，在 LOCAL 模型中，节点只能与其邻域通信，并且经过 t 轮后，每个节点只知道其在 G^t 中每个邻居最初存储的信息；而在 Hybrid 中，我们还可以选择哪些消息通过全局网络路由，然后让节点利用本地网络接收其附近节点通过全局网络看到的消息。

˜**下界**。 我们首先证明，在 Hybrid 中解决 k-信息传播需要 $\Omega(T_k)$ 轮。

˜**定理 1.1.** 在 Hybrid 中解决 k-信息传播问题，存在 $\Omega(T_k)$ 轮的普适下界。

证明的粗略思路如下。我们注意到，可以假设消息只能原样路由，也就是说，不使用任何编码技术来缩短消息，或计算一组 k 消息的较短表示。这一步的思路是：我们声称的是相对于图和消息位置的普适最优性，而不是相对于消息内容的普适最优性，因此在最坏情况下，消息可以是随机比特，所以任何压缩消息的方法，只要仍能以高概率成功，[2] 至多只能将所需轮数减少一个常数因子。因此，假设消息只能原样路由，意味着轮复杂度至多增加一个常数因子。

一旦确定消息只能以原样路由，就观察 v^*，即 G where$T_k(v^*) = T_k$ 中的节点。假设有一条消息 m，它最初位于节点 u，该节点距离 v^* 有 t 跳。为了让 v^* 在 $t' < t$ 轮次内获知这条消息，必然在某个时刻，m 被发送到一条全局边上，传至 $B_{t'}(v^*)$ 中的某个节点。反设此事并未发生——也就是说，在算法的前 t' 轮中， m 只通过本地边传递，或通过全局边发送到 $V \setminus B_{t'}(v^*)$ 中的节点。这意味着， m 必须仅通过本地边，从某个节点 $w \in (V \setminus B_{t'}(v^*))$ 传到 v^*，这显然无法在 t' 轮次内完成。

因此，我们考察 $B_{T_k-1}(v^*)$。根据定义 2， $B_{T_k}(v^*)$ 是以 v^* 为中心、满足 $|B_{T_k}(v^*)| \geq k/T_k$ 的最小半径球，因此有 $|B_{T_k-1}(v^*)| < k/(T_k - 1)$。如果至少有 $k/2$ 条消息最初位于 $B_{T_k-1}(v^*)$ 之外，那么我们将证明，节点 v^* 无法在 $T_k/(2 \log\ n)$ 轮内获知图中的所有k条消息。将最初位于 $B_{T_k-1}(v^*)$ 之外的 $k/2$ 条消息记为 M'。为了使 v^* 在 $(T_k - 1)/(2 \log\ n) < T_k - 1$ 轮内获知所有消息， M' 中的每条消息都必须在某个时刻通过一条全局边传送到 $B_{T_k-1}(v^*)$ 中的某个节点。然而，由于 $|B_{T_k-1}(v^*)| < k/(T_k - 1)$，在 $(T_k - 1)/(2 \log\ n)$ 轮内，这些节点最多只能接收 $(T_k - 1)/(2 \log\ n) \cdot |B_{T_k-1}(v^*)| \cdot \log\ n < k/2$ 条消息，因为根据 Hybrid 的定义，每个节点每轮最多只能接收 $\log\ n$ 条全局消息。因此，这就产生了矛盾。

反之，如果至少有 $k/2$ 条消息原本就在 $B_{T_k-1}(v^*)$ 内部，我们将证明 $B_{T_k-1}(v^*)$ 没有能力将这些消息发送出 $B_{T_k-1}(v^*)$，传递给图中的其余节点，即 $V \setminus B_{T_k-1}(v^*)$ 中的节点。

上界。 我们通过证明相同的轮复杂度足以确定性地解决 k-信息传播，来补充这个下界。

˜**定理 1.2.** 该 k-信息传播 问题可以确定性地 在 $O(T_k)$ 轮中在Hybrid中解决。该结果甚至在限制性更强的 Hybrid$_0$ 中也成立。

[2]以高概率（w.h.p.）表示，对于任意但固定的 $c > 0$，成功概率至少为 $1 - 1/n^c$。

˜为证明定理 1.2，我们将图划分为簇，每个簇的直径为 $O(T_k)$，包含$\Theta(k/T_k)$ 个节点。我们希望确保所有 k 消息都能到达每个簇，从而使每个节点最终都能通过本地边接收其所在簇掌握的所有信息，进而获知全部 k 消息。

为此，我们首先构建一棵包含所有簇的虚拟树。在 Hybrid 模型中，这项工作很简单；但在 Hybrid$_0$，中却颇具挑战，因为我们必须构建一棵覆盖整个图、˜深度为 $O(1)$ 且度数为常数的树，并确保树中的任意两个节点都知道彼此的标识符，即使它们在原始图中相距甚远。为此，我们借鉴了 [24]中的一些覆盖网络构建技术。

接着，对于二叉树中的任意一对父簇 P 和子簇 C，我们确保每个节点 node P 都恰好知道 C 中的一个节点的标识符，反之亦然，这样它们便能通过全局网络进行通信。要达到这种状态需要格外谨慎，因为将一个簇中所有节点的标识符发送给另一个簇中的所有节点是一项艰巨的任务。这必须通过全局网络完成（因为这些簇在物理上可能相距甚远），因此我们必须精心设计一种算法，以避免造成拥塞。

一旦簇 P 和 C 中的节点能够彼此通信，我们就沿着簇树向上传播所有 k 消息，使树根处的簇获知所有 k 消息。然后，我们再将这些消息向下传播，以确保每个簇都能收到 k 消息。在进行这些传播的同时，我们会在每个簇内执行负载均衡步骤，以确保其各个节点负责向其他簇发送的消息数量大致相同，从而避免全局网络发生拥塞。

推论。由上述结果可立即推出 Hybrid 中的 BCC 模拟，它需要在 G 中均匀分布地广播 $k = n$ 令牌。因此，我们使用 T_n，如下所示。

˜**推论 1.3**。 存在一个普适下界，为 $\Omega(T_n)$ 轮，用于模拟 BCC˜中的一轮，并且存在一个确定性算法，可在 $O(T_n)$ 轮内在 Hybrid 中完成该模拟（甚至在Hybrid$_0$中）。

作为我们关于 k-信息传播界的另一个推论，我们对 k-聚合问题得到了同样的普适最优刻画。

定义 *3*（k-聚合）。设 $F: \ X \times X \to X$ 是一个聚合函数（满足结合律和交换律）。假设每个节点 v 最初持有 k 个值 $f_1(v), \ldots, f_k(v)$。 k- 聚合问题要求每个节点了解所有值 $F(f_i(v_1), \ldots, f_i(v_n))$，对每个 $i \in [k]$均如此。假设 k 以及每个 $v \in V$对应的值 $f_1(v), \ldots, f_k(v)$ 都至多是关于 n 的多项式。

我们这一工作的核心思想是在 Hybrid 中证明 k-信息传播与 k-聚合之间存在双向归约，这意味着上述下界和上界都可以迁移。证明归约的两个方向都需要一些技术工作。证明 k-聚合能够解决 k-信息传播，需要一种在图中的所有节点之间进行协调的例程；而证明 k-信息传播能够解决 k-聚合，则需要对我们在定理 1.2 的算法中构造的树作出特定观察。因此，我们得到如下结论。

˜**定理 1.4.** 存在一个普适下界为$\Omega(T_k)$ 轮，在Hybrid 中，对于 k-聚合˜问题，有一个确定性算法可在 $O(T_k)$ 轮内解决该问题（甚至在Hybrid$_0$中）。

1.1.2 对 T_k 的分析

我们分析 T_k 的性质，以便将其与其他图参数进行比较，例如图中的节点数量及其直径。这使我们能够将我们的结果与已有的相关工作进行比较。

引理 1.5。若 $T_k \neq D$，则 $\sqrt{\frac{D}{3n}}k \leq T_k \leq$ 最小 $\{D, \sqrt{k}\}$。

回顾一下，此前用于解决较弱版本的 k-信息传播问题的工作都需要 $\tilde{O}(\sqrt{k})$ 轮 [3,6]，因此引理 1.5 表明，我们针对 $O(T_k)$ 轮内 k-信息传播问题的算法，其速度不会慢于此前的工作，并且适用于更多种情况。

反过来，我们分析满足 $T_k = o(\sqrt{k})$ 的某些图族中的图，以说明这类图有很多，而不仅仅是满足 $D = o(\sqrt{k})$ 的图。接下来，我们估算路径、环以及任意 d维方形网格图的 T_k 值（形式化定义将在后文给出）。例如，对于 d 维方形网格，我们得到如下结果。

Th定理 1.6。设 $G = (V, E)$ 为一个 d维方形网格，其中 $|V| = n = m^d$.

$$T_k = \begin{cases} \Theta(k^{1/(d+1)}) & k = O(n^{(d+1)/d}) \\ O(D) = O(dn^{1/d}) & k = \Omega(n^{(d+1)/d}) \end{cases}$$

这说明了几个有趣的结论。首先，当 k 不太大，即 $k = O(n^{1+1/d})$ 时，我们可以用远优于现有算法的轮复杂度广播 k 条消息，复杂度为 $\tilde{O}(T_k) = \tilde{O}(k^{1/(d+1)})$；而现有算法至少需要 $\tilde{O}(\sqrt{k})$。然而，一旦 k 超过 $\Omega(n^{1+1/d}) = \Omega(D^{d+1})$，我们就无法显著优于通过本地网络朴素地发送所有消息，这需要 $O(D)$ 轮。

具体来说，注意对于任意常数维度 d，网格图都有 $O(dn) = O(n)$ 条边，因此利用定理 1.2，在 $\tilde{O}(T_n) = O(n^{1/(d+1)})$ 轮内，所有节点都可以获知整个图，并在本地计算精确全点对最短路径。对于任意 $d \geq 2$，这在多项式时间上都快于 [3,6,38] 的存在性最优算法。

1.1.3 应用

我们展示了 k-信息传播和 k-聚合结果的多种应用。这些应用大多可以相当直接地从上述算法推出，因为广播和聚合是非常基础的构建模块。

我们首先介绍 Hybrid 中全点对最短路径（APSP）的若干应用。在以往的研究中，[6,38] 给出了精确加权全源最短路径问题的存在性下界和上界，均为 $O(\sqrt{n})$ 轮。其算法是随机化的，而目前已知最好的确定性算法 [3] 需要 $\Theta(\sqrt{n})$ 轮，但只能得到一个 $(\log n/\log\log n)$-近似。对于无权全源最短路径，[3] 还确定性地在 $O(\sqrt{n})$ 轮内给出了 $(1+\epsilon)$-近似。

我们给出了若干用于精确计算或近似计算全点对最短路径的算法。在介绍上界之前，我们强调，这些算法在 Hybrid$_0$ 中均适用。请注意，在此设置下，标识符可以是任意的 $O(\log n)$ 字符串。为了让节点生成全点对最短路径的输出，它必须知道 G 中所有节点的标识符，因此这相当于广播所有 n 标识符——本质上，它们是 n 任意消息。因此，根据定理 1.1，以下结论成立。[3]

[3]请注意，在 k-信息传播中，我们假设每条 k 消息最初仅由一个节点知晓。在这种情况下，我们必须广播节点在 G 中的 n 任意标识符；实际上，每个节点的标识符对其自身及其邻居都已知。这并不构成问题，因为不失一般性，可以假设任何用于 k-信息传播的算法都能免费执行一轮免费：最初持有消息的每个节点都通过本地网络将这些消息发送给所有邻居。这意味着定理 1.7 中的下界至多比定理 1.1 中的下界低一轮。

定理 1.7。 在 Hybrid$_0$，中，对于全点对最短路径的任何近似算法，都存在一个普遍最优的下界，即需要 $\Omega(T_n)$ 轮。

请注意，下界的普适最优性是相对于图拓扑而言的，而不是相对于标识符本身的选择而言的。

下面的大多数算法都在 $O(T_n)$轮内运行，因此是在 Hybrid$_0$中普适最优的 APSP 计算。此外，根据引理 1.5，$T_n \leq \sqrt{n}$ 对任意图都成立，这意味着我们的算法运行 $O(T_n) = O(\sqrt{n})$轮，因此在 Hybrid 中绝不会慢于 [6,38] 的 $\Theta(\sqrt{n})$轮算法，但当 $T_n = o(\sqrt{n})$时则更快。

作为热身，我们将展示，在稀疏图中，可以获知整个图，从而精确地计算 APSP。也就是说，对于包含 $O(n)$条边的图，我们应用定理 1.2，且 $k = n$ 至多 $O(1)$次，最终得到 $O(T_n)$轮。

推论 1.8. 给定一个稀疏加权图 $G = (V, E, \omega)$，其中 $|E| = O(n)$，存在一种算法，能够在 $O(T_n)$ 轮内解决 Hybrid$_0$，中的任意图问题，包括精确加权全源最短路径。

接下来，我们将讨论如何近似任意边数图中的全点对最短路径。

无权全源最短路径。 我们展示了一种无权全源最短路径的 $(1+\epsilon)$ 近似算法，该算法以高概率在$\tilde{O}(T_n/\epsilon^2)$ 轮内运行。已知在一般图上以几乎任意因子精确计算或近似全点对最短路径的最佳算法，需要 $O(\sqrt{n})$ 轮 [38]。由于 $T_n \leq \sqrt{n}$，我们的算法始终至少同样快，并且在 $T_n = o(\sqrt{n})$ 时更快。

定理 1.9. 对于任意 $\epsilon \in (0,1)$，存在一种随机化算法，能够在无权图中以 $(1+\epsilon)$ 近似计算全点对最短路径；该算法在 Hybrid$_0$，中以高概率在 $\widetilde{O}(T_n/\epsilon^2)$ 轮内运行。

基本上，我们将 [45] 的 $(1+\epsilon)$ 多对数级单源最短路径算法扩展为一种普遍最优的无权全源最短路径算法。该算法背后的技术如下。我们首先计算一个$O(T_n)$-弱直径聚类，最多包含 T_n 个簇。然后，我们从簇中心执行 $(1+\epsilon)$ 多对数级单源最短路径算法。接着，我们探索每个节点足够大的邻域，并让每个节点广播其最近的簇中心及其到该中心的距离。最后，我们利用 $(1+\epsilon)$ 对簇中心的近似以及广播的距离，便能够以足够高的精度近似这些距离。

加权 APSP. 现在，我们给出若干用于近似加权图中全点对最短路径的结果。

我们先介绍以下算法，它能计算出 $(\epsilon \cdot \log n)$ 近似；该算法基于 [43] 的一项已知结果来构造图伸展体（即一种能够保留距离近似值的稀疏子图），随后获取该伸展体的全部信息。

定理 1.10. 存在一个 Hybrid$_0$，确定性算法，对于给定的图 $G = (V, E, \omega)$，可在 $\widetilde{O}(2^{1/\epsilon}T_n)$ 轮内计算 *APSP* 的 $(\epsilon \cdot \log n)$-近似。

我们可以利用定理 1.10 得到一个与目前已知最佳确定性近似结果相当的结论：通过 [3], 确定性地实现相同的近似比 $\log n/\log\log n$，但所需轮数为 $O(T_n)$，而非 $O(\sqrt{n})$。

推论 1.11. 在 Hybrid$_0$ 中运行定理 *1.10*，并取 $\epsilon = 1/\log\log n$，即可在 $O(T_n)$ 轮内实现 $\log n/\log\log n$ 近似。

最后，我们给出了以下关于近似全点对最短路径的结果。它的运行速度略慢于$O(T_n)$ 轮，但近似比要好得多。例如，对于加权全点对最短路径的 3-近似，它实现的轮复杂度为 $\tilde{O}(n^{1/4}T_n^{1/2})$，这始终小于 $\tilde{O}(\sqrt{n})$，因为$T_n \leq \sqrt{n}$。该结果基于广为人知的骨架图技术，该技术最早由 [48] 观察到。

定理 1.12。对于任意整数 $\alpha \geq 1$，存在一种随机化算法，在加权图上于 $\tilde{O}(\alpha \cdot n^{1/(3\alpha+1)}(T_n)^{2/(3+1/\alpha)} + \alpha \cdot T_n)$ 轮内计算*APSP* 的 $(4\alpha-1)$-近似，并在Hybrid$_0$，以高概率完成。

1.1.4 割的近似

与 [3], 类似，我们利用割稀疏器 [46] 及其在 CONGEST [36] 中的高效实现来近似任意割并解决若干割问题。我们的算法在 Hybrid 中运行 $\tilde{O}(T_n/\epsilon+1/\epsilon^2)$ 轮 $_0$，其速度始终不慢于所有现有算法，并且在 $T_n = o(\sqrt{n})$ 时更快。我们结果背后的思路是，在 $\tilde{O}(1/\epsilon^2)$ 轮内执行 [36], 的 CONGEST 算法，从而构造一个包含 $\tilde{O}(n/\epsilon^2)$ 条边、可近似所有割的子图。然后，我们使用定理 1.2，在 $\sim$ $O(T_{n/\epsilon^2}) = O(T_n/\epsilon)$ 轮内广播该子图。

定理 1.13。对于任意 $\epsilon \in (0,1)$，存在一个 Hybrid$_0$ 算法，其运行 $\tilde{O}(T_n/\epsilon + 1/\epsilon^2)$轮次，以高概率，在此之后每个节点都可以在本地计算图中所有割的一个$(1+\epsilon^2)$-近似。这为许多问题提供了近似解，包括最小割、最小 s-t 割、最稀疏割和最大割。

1.2 更多相关工作

Hybrid. Hybrid 模型的当前形式最近在 [6]中被引入。此后，大多数研究集中于最短路径计算以及与之密切相关的问题，例如直径$^\sim$计算。在 [6] 中，对于全点对最短路径，即使是 $O(\sqrt{n})$-近似，也存在一个Ω($\sqrt{n}$)轮的存在性下界。 [38], 将这一结果推广为 k-源最短路径的Ω($\sqrt{k}$)轮下界。在 [38] 中，给出了一个存在性最优的随机化 $O(\sqrt{n})$算法，用于计算精确全点对最短路径。两篇论文都大量使用了令牌传播、令牌路由、骨架图，以及对 Congested Clique（一种不同的分布式模型）算法的模拟。

目前最先进的结果是由 [9] 实现的精确 $n^{1/3}$-单源最短路径（因此也适用于单源最短路径），需要 $\tilde{O}(n^{1/3})$ 轮次。同一作者还实现了 $\tilde{O}(n^{5/17})$ 轮次的（$1+\epsilon$）近似单源最短路径 [10]。最近， [45] 依托 [44] 建立的 minor 聚合框架，在多对数时间内实现了近乎最优的（$1+\epsilon$）单源最短路径近似。

值得注意的是，迄今为止，大多数 Hybrid 算法都是随机化的。例外的是 [3], 的算法，它们在 $O(\sqrt{n})$ 轮内实现了近似比为 $(\log n/\log\log n)$ 的全点对最短路径；此外，还对 k-信息传播进行了去随机化，适用于 $k \geq n$ 的情形，运行时间为 $\tilde{O}(\sqrt{k})$ 轮。

Hybrid 模型中的后续研究聚焦于直径计算和下界 [3,38],计算路由方案 [13,39], 以及更多与距离相关的问题 [9,10,17]。

普适最优性。分布式场景下的普适最优性 概念最早由 Garay、Kutten 和 Peleg 在 FOCS ’93 [18], 中提出。粗略地说，他们探讨了能否找出与各种基本网络问题的分布式复杂度相关的内在图参数，并为这些问题开发普适最优算法。

CONGEST 模型中一项在面向非最坏情况图的算法方面取得重大进展的研究方向是低拥塞快捷方式框架，该框架由 [20], 引入，并在许多后续工作中得到进一步发展 [19, 21–23, 25–29, 35, 44, 50, 51]。普适最优性的概念由 [30], 的工作正式化；该工作探讨了普适最优性的不同概念，并解决了许多重要问题。

杂项。 Hybrid 与其他已研究的混合型网络有关，例如 [2, 14, 24]。与 Hybrid 类似的另一个模型是 [1], 提出的云计算（CWC），该模型考虑由计算节点以及（通常一个）被动存储云节点组成的网络。他们探讨如何利用共享云存储，在不同容量限制下高效地执行联合计算。我们受到他们相关工作的启发，分析图中节点的邻域，以便为给定的图定义一个限制通信的最优参数。

2 预备知识

我们考虑无向连通图 $G = (V, E, \omega)$、 $n = |V|$、 $m = |E|$，以及权重函数ω，其中所有权重都是 n 的多项式。如果图是无权的，则 $\omega \equiv 1$。两个节点 $v, w \in V$ 之间的距离记为 $d(v, w)$。 $v, w \in V$ 之间的跳距离记为 $\mathsf{hop}(v, w)$，即两个节点之间的无权距离。图的直径记为D。记 $d^h(v, w)$ 为考虑所有长度至多为 h 的路径时， v 与 w 之间最短路径的权重。令 $N_i(v) = \{w \mid \mathsf{hop}(v, w) = i\}$ 表示与 v 的跳距离恰为 i 的节点集合；令 $B_t(v) = \{w \in V \mid \mathsf{hop}(v, w) \leq t\} = \bigcup_{0 \leq i \leq t} N_i(v)$ 表示以 v 为中心、半径为 t 的球。给定节点集合 $C \subseteq V$， C 的弱直径定义为 $\max_{u,v \in C} \mathsf{hop}(u, v)$，其中跳距离在原图 G中测量。 C 的强直径是 G 中由 C 诱导的子图的直径。对于任意正整数 x，令 $[x] = \{1, \ldots, x\}$。

模型定义。 我们正式定义 Hybrid 和 Hybrid$_0$。

定义 4 (Hybrid 模型 [6])。 我们考虑一个网络 $G = (V, E)$，其中 $|V| = n$，且节点的标识符取值范围为 $[n]$。通信以同步轮次进行。在每一轮中，节点可以执行任意本地计算，随后彼此通信。本地通信由 LOCAL 模型 *[40]*, 建模，其中对于任意 $e = \{v, u\} \in E$，节点 v 和 u 可以通过 e交换任意数量的比特。全局通信由 NCC 模型建模， *[5]*, 其中每个节点都可以与 G中最多任意 $\log n$ 个节点交换 $O(\log\ n)$ 比特的消息。要求每个节点在每轮中作为发送方和接收方的消息数均至多为 $\log n$ 。

定义 5 (Hybrid$_0$ 模型 [4, 6])。 Hybrid$_0$ 模型与 Hybrid 模型类似，区别在于标识符是任意的，并且对于某个常数 c，其取值范围为 $[n^c]$ 。这意味着节点可能不知道图中使用了哪些标识符，因此只能向其已知标识符的节点发送消息。也就是说，全局通信使用 NCC$_0$ *[4]*，而不是 NCC。假设在算法开始时，每个节点都知道自己的标识符以及其邻居的标识符。

可以通过以下参数化 Hybrid 模型：本地边上的最大消息大小为 λ ，每个节点每轮通过全局边总共可以交换 γ 比特。如前所述，我们考虑标准的 $\lambda = \infty$ 和 $\gamma = O(\log^2 n)$。标准分布式模型也是这种参数化的特例（相差常数因子），其中 LOCAL 为 $\lambda = \infty$， $\gamma = 0$，CONGEST 为 $\lambda = O(\log\ n)$ $\gamma = 0$，Congested Clique 为 $\lambda = 0$， $\gamma = n \log\ n$，而 NCC 为 $\lambda = 0$， $\gamma = O(\log^2 n)$。

通信原语。 作为基本工具，我们在Hybrid$_0$ 中证明以下结果。为此，我们使用 [5]，其针对一个与 Hybrid$_0$ 相似但略有不同的模型展示了类似结果。

引理 2.1。 在 Hybrid$_0$，中，可以构造一棵虚拟树 T，该树跨越 G，具有常数~最大度，并具有 O（1）深度。算法结束时，可以保证树中的每两个~相邻节点都知道彼此在 G中的标识符。这需要 O（1）轮。

引理 2.1 的证明推迟至附录 A.1。由于引理 2.1，我们仅针对 $k = 1$ 的特殊情况，得到以下 k-聚合解决方案。具体而言，请注意，在展示适用于一般 k 的 k-信息传播和 k-聚合算法时会利用这一结果。

˜引理 2.2。 </style}对于，可以在 Hybrid$_0$ 中确定性地用 O（1） 轮解决 k-聚合 问题。

问题定义。 我们为 Hybrid 和 Hybrid$_0$ 中尚未在上文定义的问题给出形式化定义。

定义 6 （全点对最短路径（APSP））。 每个节点 v 都必须针对每个节点 wˆ输出距离 d（v、w）。在 α-近似 *APSP* 中， v 对所有 $w \in V$ 输出 d（v、w），其中 d（v、w） $\leq$ˆd（v、w） $\leq \alpha \cdot d$（v、w）。注意，每个 $v \in V$ 都必须知道所有节点的标识符，才能写出输出。

普适最优性。我们遵循 [30], 的方法，并如下定义普适最优算法。给定问题 $P =$（S、I），将其输入拆分为固定设定 S 和参数化输入 I。例如，在 k-信息传播中，我们固定图 G 以及所有 k 消息的起始位置，但消息内容是任意的。对于求解 P 的给定算法 A，以及 S 的任意可能状态 s 和 I 的任意可能状态 i，用 t（a、s、i）表示 A 在 P 上运行且输入为 $S = s$ 和 $I = i$ 时的轮复杂度。如果对于 s 的任意选择，算法 A 相对于 P 是普适最优的，那么 A 的最坏情况˜轮复杂度至多为预先知道 s、用于求解 P 的最佳算法 A_s 的轮复杂度的 O（1）倍。形式化地，对于所有可能的 s 和任意算法 A_s，令 t（A、s） $=\max_i t$（A、s、i），且令 t（A_s） $=\max_i t$（A_s、s、i），则有 t（A、s） $= O$（t（A_s））。也就是说，必须固定一个适用于所有 s 的单一 A，但 A_s 可以针对每个 s 而不同。

其他事项。

定义 7 (方格网图**)。** A 个 d 维方格网图 $G = (V, E)$，其中 $n = m^d$，是由 d m 个 P_m 节点路径构成的笛卡尔积图。形式化地说， $G = P_m^1 \times \cdots \times P_m^d$。

3 普遍最优广播

现在我们展示普遍最优广播的结果。我们首先在第 3.1 节介绍T_k 的一些基本性质，接着在第 3.2 节给出下界，然后在第 3.3 节证明上界。最后，在第 3.4 节中，我们展示从 k-聚合到 Hybrid 中k-信息传播的双向归约，从而为 k-聚合实现普遍最优结果。

3.1 T_k的基本性质

我们给出关于 T_k的以下实用工具。其证明见附录 A.2。

˜引理 3.1。 可以在 $O(T_k)$ 轮内计算 T_k 并使其全局知晓，适用于Hybrid$_0$。

我们还得到以下推论，表明在上述算法中，节点获得了更多信息，而不仅仅是 T_k 的值。

推论 3.2. 使用引理 *3.1* 计算 T_k 时，每个节点都会获知图中 $T_k(v)$ 的分布，即对于任意 t，有多少个节点 v 具有 $T_k(v) = t$。

现在我们证明一个结论，它限制了 T_k 随着 k 增长的速度。该结论表明，对于 $k' = \alpha k$，$T_{k'}$ 的值相对于 T_k 最多只能大约大 $\sqrt{\alpha}$ 倍。证明思路是，对于任意图，所有半径为 T_k 的邻域都能在 T_k 轮内获知 k 条消息。因此，如果我们将 T_k 增大 α 倍，那么所有大小为 $T_k \cdot \alpha$ 的邻域都能在 $T_k \cdot \alpha$ 轮内获知 $k \cdot \alpha^2$ 条消息。完整证明见附录 A.2。

引理 3.3。对于 $\alpha \geq 1$，$T_{\alpha k} \leq 6\sqrt{\alpha} \cdot T_k$。

现在我们用 n，D，k 来界定 T_k，证明推迟到附录 A.2。

引理 1.5。如果 $T_k \neq D$，那么 $\sqrt{\frac{D}{3n}k} \leq T_k \leq$ 最小 $\{D, \sqrt{k}\}$。

3.2 下界

现在我们希望证明定理 1.1。

定理 1.1。对于在 Hybrid 模型中解决 k-信息传播问题，存在一个普适下界 $\Omega(T_k)$ 轮。

在此之前，我们先证明以下引理。该引理基本上指出，我们可以假设消息以原样通过全局网络发送，而不使用任何编码技术来压缩需要传输的比特数。请注意，以下结论适用于证明 Hybrid 中 k-信息传播的普适下界，因为普适下界是相对于图 G 和消息的初始位置而言的，而与消息内容无关。本质上，以下引理指出，如果消息是均匀随机选取的字符串，那么即使使用随机化算法，也不可能将消息压缩到原大小的常数倍以下。

引理 3.4。设有一个图 G，以及节点的一个划分 $V = A \cup B$、$A \cap B = \emptyset$、$A \neq \emptyset$、$B \neq \emptyset$，并给定某个值 k。假设 A 中的每个节点都可以自由选择任意数量的消息，且 A 中的所有节点总共选择 k 条任意的 $b = O(\log n)$ 比特消息。那么，为了使 B 中的节点能够以至少 $1/2$ 的成功概率重建全部 k 条消息，必须从 A 向 B 传递至少 $\Omega(kb)$ 比特的信息。即使每个节点都知道 G 的完整拓扑，以及 A 中的每个节点可以选择多少条消息，这一结论仍然成立。

证明。令 ALG 为最优算法（以在 A 与 B之间发送的比特数最少为准），该算法在 A 中的节点与 B 中的节点之间进行通信，使得执行结束时，B中的至少一个节点能够恢复 k 条消息，无论这些消息的内容是什么；假设 ALG 以至少 $1/2$的概率成功（即恢复所有消息）。记 r 为最坏情况下 ALG 从 A 传输到 B 的比特数。现在我们要证明 $r = \Omega(kb)$。

A 中能够选择消息的每个节点，都会为其每条消息独立均匀随机地选择随机比特。总计而言，A 中的节点会均匀随机地选择 kb 个随机比特。

注意，ALG 可能从 A 向 B 发送 2^r 个比特串——将这些比特串构成的集合记为 S。收到字符串 $s \in S$后，B 中的节点会执行某个算法，并在算法结束时声明，在 A 中选定的 k 条消息分别是字符串 $m = m_1$、……、m_k。对于每个 $s \in S$，B 中的节点对其认为 A 持有的 $m = m_1$、……、m_k 条消息有一个概率分布，将此分布记为 $m_s = P(m|s)$。

对于 A 选择的任意一组消息 $m = m_1$、……、m_k，用 $s_m = P(s|m)$ 表示在给定 m 的情况下，ALG 发送给 B 的 S 中字符串的分布。显然，由于 ALG 总能以至少二分之一的概率成功，对于任意特定的 m'，都有 $1/2$。记为

M 节点在 A 中可以选择的所有消息的集合——即所有长度为 kb比特的字符串的集合。对所有可能的 $m' \in M$求和，我们得到

$$\sum_{m' \in M} \sum_{s \in S} s_{m'} \cdot m'_s \geq |M|/2 = 2^{kb-1}.$$

请注意， $s_{m'} \leq 1$ 始终对应于某个事件的概率；对于任意给定的 $s \in S$，都有 $\sum_{m' \in M} m'_s = 1$，因为它对应于一组互斥事件的概率之和，而这些事件共同构成对事件空间的划分。将这两者代入上式可得

$$2^{kb-1} \leq \sum_{m' \in M} \sum_{s \in S} s_{m'} \cdot m'_s = \sum_{s \in S} \sum_{m' \in M} s_{m'} \cdot m'_s \leq \sum_{s \in S} \sum_{m' \in M} m'_s = \sum_{s \in S} 1 = |S| = 2^r.$$

因此， $r \geq kb - 1$ 所以 $r = \Omega(kb)$，如所要求。 □

现在，利用引理 3.4，我们可以证明定理 1.1。

证明。 我们用 A_M 表示将所有消息 M 广播给所有 G 所需的最优轮数。˜我们希望证明 $A_M = \Omega(T_k)$。此外，对于节点集合 $V' \subseteq V$，用 $M(V')$ 表示 M 中最初存储在 V' 任意节点上的消息。

设 $v^* = \arg\max_{v \in V} T_k(v)$。在整个证明中，我们使用以下观察：如果一条消息与 v^*的距离为 $\ell + 1$，那么为了让 v^* 在 ℓ 轮内收到它，至少必须通过全局网络将它发送到 $B_\ell(v^*)$ 中一次。根据引理 3.4，我们知道，无法将要发送的信息压缩超过一个常数因子，因此可以假设消息只是原样发送，不使用任何编码技术来缩短特定消息或消息集合。

现在我们分情况讨论。

情况 1： $T_k < D/2$**和** $|M(V \setminus B_{T_k}(v^*))| > k/2$。 我们对 $B_{T_k-1}(v^*)$ 的全局网络带宽容量进行了界定。由于 T_k 是满足 $|B_{T_k}(v^*)| \geq k/T_k$ 的最小半径，因此 $|B_{T_k-1}(v^*)| < k/T_k$。每个节点每轮可以使用全局网络接收 $\log n$ 条消息，因此在 $(T_k - 1)/(2 \log n)$ 轮内， $B_{T_k-1}(v^*)$ 至多可以接收

$$\frac{T_k - 1}{2 \log n} \log n |B_{T_k-1}(v^*)| \leq \frac{T_k - 1}{2} \frac{k}{T_k} \leq \frac{k}{2}$$

使用全局网络发送消息。

由于 $|M(V \setminus B_{T_k}(v^*))| > k/2$，因此也有 $P = M(V \setminus B_{T_k-1}(v^*))$ $|P| > k/2$。为了使 v^* 在至多 $T_k - 1$ 轮内接收消息集合 P，这些消息必须经由全局网络发送到 $B_{T_k-1}(v^*)$。然而，正如我们刚才所证明的，使用全局网络时， $B_{T_k-1}(v^*)$ 至少需要 $(T_k - 1)/(2 \log n)$ 轮才能接收 $k/2$ 条消息。因此， $A_M > \min\{(T_k - 1)/(2 \log n)\ T_k - 1\} = (T_k - 1)/(2 \log n)$。

情况 2： $T_k < D/2$**和** $|M(V \setminus B_{T_k}(v^*))| \leq k/2$。 现在， $B_{T_k}(v^*)$ 内至少有 $k/2$ 个令牌，我们再次分情况讨论。我们观察围绕 v^* 的两个 环 ，记为 $R_1 = B_{T_k/2}(v^*)$ 和$R_2 = B_{T_k}(v^*) \setminus R_1$。

情况 2.1： $|M(R_2)| \geq k/4$。 如果外环 R_2 至少有 $k/4$ 个令牌，那么我们将类似于上面的情况1进行处理。在 T_k/中， R_1 的全局容量在 $(4 \log n)$ 轮内至多为 $k/4$，因此 $A_M > T_k/(4 \log n)$。

情况 2.2： $|M(R_2)| < k/4$。由于 R_2 的令牌少于 $k/4$ 个，因此 R_1 至少有 $k/4$ 个令牌。现在我们将视角从接收容量转向发送容量：为了让节点 $w \notin B_{T_k}(v^*)$ 在少于 $T_k/4$ 轮内接收所有令牌，R_1 中的所有令牌都必须通过全局网络发送。经过 $T_k/(4\log\ n)$ 轮后，R_1 通过全局网络的发送容量与接收容量相同；根据与情况 2.1 相同的论证，该容量至多为 $k/4$ 。因此，至少需要 $T_k/(4\log\ n)$ 轮才能将所有令牌从 R_1 发送到 w，这意味着 $A_M > T_k/(4\log\ n)$。请注意，只有在存在节点 $w \notin B_{T_k}(v^*)$ 时，这一点才成立。否则，$B_{T_k}(v^*) = V$，但这意味着 $T_k \geq D/2$，与情况 2 中始终采用的假设矛盾。

情况 3： $T_k \geq D/2$。现在假设 $T_k \geq D/2$。这意味着，根据定义 2，$|B_{D/2-1}(v^*)| < k/T_k$，并且存在一个节点 $w \notin B_{D/2-1}(v^*)$，否则直径就会更小。我们可以使用与上述情况 1和2 中相同的论证，将 $D/2-1$ 替换为 T_k，并得到相同的结果。

在所有情况下，我们都证明了 $A_M = \tilde{\Omega}(T_k)$，因此我们就完成了。 □

3.3 上界

现在我们证明定理 1.2。

˜定理 1.2。该 k-信息传播 问题可以确定性地 在 $O(T_k)$ 轮中在Hybrid求解。即使在限制性更强的 Hybrid$_0$中，该结果也成立。

我们回顾一个关于在 CONGEST 中计算 (α, β)-支配集的已知结果。

定义 8。$G = (V, E)$ 的一个(α, β)-支配集是一个子集 $W \subseteq V$，使得对于每个 $v \in V$，都存在一个 $w \in W$，满足 $hop(v, w) \leq \beta$，且对于任意 $w_1, w_2 \in W$, $w_1 \neq w_2$，都有 $hop(w_1, w_2) \geq \alpha$。

定理3.5 (定理 1.1 中 [37])。令 μ 为正整数。一个$(\mu+1, \mu\lceil$对数 $n\rceil)$支配集可以在本地网络中通过确定性方式在 $O(\mu$对数 $n)$ 轮内于 CONGEST 中计算。

在整个证明过程中，我们使用以下术语，并在此给出其形式化定义。

定义 9 (泛洪**).** 通过本地网络泛洪信息，就是通过所有节点的所有相连本地边发送该信息。在后续轮中，节点汇总它们接收到的信息，并继续发送这些信息。经过 t 轮后，每个节点 v 都知道在泛洪开始前其 t邻域内任一节点持有的全部信息。

引理 3.6（均匀负载均衡）。给定一个弱直径为 d 的节点集合 C，以及一组消息 M，其中 $|M| = k$ 分布在 C 上，存在一种算法，使其终止时，每个 $v \in C$ 至多持有 $\lceil k/|C|\rceil$ 条消息。该算法运行 $2d = O(d)$ 轮。我们称 C 在自身内部进行均匀分布 M 。

证明。经过 d 轮，所有节点都会泛洪传播 C 的消息和标识符。随后，标识符最小的节点计算一种分配方案，使每个 $v \in C$ 至多负责 $\lceil k/|C|\rceil$ 条消息，并在接下来的 d 轮中泛洪传播该分配方案，从而使其到达所有 $v \in C$。 □

我们使用定理 3.5 证明以下引理：创建弱直径较小且节点数量大致相同的簇。

引理3.7。对于任意 k，都可以将节点集合划分为弱直径至多为 $4T_k\lceil\log n\rceil$ 的簇，使得每个簇包含介于 k/T_k 和 $2k/T_k$ 个节点之间。该引理返回一个 $R \subseteq V$ 簇领导者集合；对于每个 $r \in R$，记由其领导的簇为 $C(r)$；对于每个簇 C，记其领导者为 $r(C) \in R$。每个节点都知道自己是否属于 R，并且˜知道自己属于哪个簇。所有这些操作在 Hybrid$_0$ 中耗时 $O(T_k)$ 轮次。

˜证明。 我们通过引理 3.1在 T_k $O(T_k)$ 轮内完成计算。我们选择 $\mu = 2T_k$ 并使用定理 3.5，在 $O(T_k \log\ n)$ 轮内计算一个 $(2T_k + 1,\ 2T_k\lceil\log\ n\rceil)$-支配集[4]。我们将支配节点的集合记为 R。然后，在 $2T_k\lceil\log n\rceil$ 轮内，每个节点通过本地网络获知其 $2T_k\lceil\log n\rceil$ 邻域以及其中的支配节点。对于每个 $v \in V$，令 $r(v)$ 为按跳距离计算最近的支配节点；若存在平局，则选择标识符最小的节点。根据定义 8， $r(v)$ 必须位于其 $2T_k\lceil\log n\rceil$邻域中。通过探索该邻域，每个节点 v 找到 $r(v)$。

对于任意 $r \in R$，将 r 的簇定义为 $C(r)$。对于任意簇 C'，令 $r(C')$ 为满足 $C(r(C')) = C'$ 的 $r \in R$。每个节点 $v \in V \setminus R$ 都加入离它最近的统治节点 $r(v)$ 所在的簇。注意，任意簇 C 恰好包含一个统治节点 $r(C)$，并将 C 的簇标识符设为 $r(C)$ 的标识符。定义 8 保证每个这样的簇的弱直径至多为 $2\beta = 4T_k\lceil\log\ n\rceil$。因此，在 $4T_k\lceil\log n\rceil$ 轮内，每个节点 v 都通过本地网络泛洪 $r(v)$，从而对于每个簇 C，任意 $v \in C$ 都知道 C 中的所有节点。

设 C 为一个簇。由于对于每个 $r_1 \neq r_2 \in R$，$\mathsf{hop}(r_1, r_2) \geq \alpha = 2T_k + 1$，因此有 $B_{T_k}(r(C)) \subseteq C$——也就是说， $u \in B_{T_k}(r(C))$ 中的每个节点都加入 C，因为离 u 最近的统治节点是 $r(C)$。根据定义 2， $|C| \geq |B_{T_k}(r(C))| \geq k/T_k$。因此，每个簇的最小规模为 k/T_k。

现在，我们确保簇不会过大。每个包含 $|C| > 2k/T_k$ 的簇会以确定性方式拆分成更多簇，直到每个簇都包含 $k/T_k \leq |C| \leq 2k/T_k$。可以在每个簇内局部完成这一过程，例如，贪心地将簇内由 $2k/T_k$ 个节点标识符组成的组分配到新簇中，并选择标识符最小的节点作为领导者。完成这一过程后，我们至多得到 nT_k/k 个互不相交的簇，每个簇的弱直径至多为 $4T_k\lceil\log n\rceil$，大小为 $k/T_k \leq |C| \leq 2k/T_k$。我们将所有被拆分的新簇的领导者加入集合 R。

□

最后，我们还给出关于剪枝树的以下辅助引理。

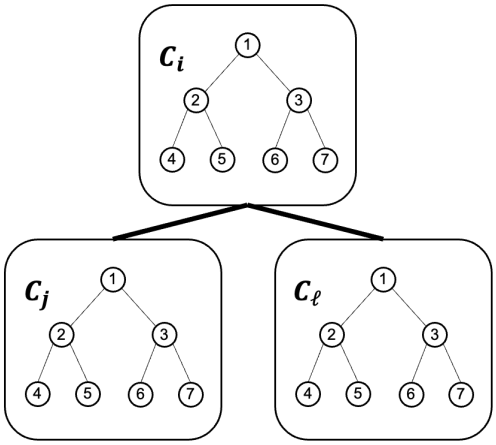
引理 3.8。 存在一棵树 $T = (V, E_T)$，其根节点为 r，最大度和深度为常数 d。给定某个函数 f: $V \to \{0, 1\}$，存在一种算法，可构造一棵树 $T' = (U, E_{T'})$,$U = \{v \mid v \in V, f(v) = 1\} \subseteq V$，其最大度和深度为常数 $d' \leq d$。这需要 $O(d^2)$轮，在 Hybrid$_0$ 中进行。假设对于每个 $v \in V$，值 $f(v)$ 在运行该算法之前已为 v 所知。

证明。 记 $U = \{v \mid v \in V, f(v) \text{ 为 } = 1\}$。注意，每个 $v \in V$都知道是否 $v \in U$。对于每个 $v \in V$，记 $T(v)$ 为以 v 为根的 T 子树。现在，每个 $v \in V$都会计算 $|U \cap T(v)|$。具体做法是，每个节点沿树向上发送其子树中属于 U 的节点数量。这需要 $O(d)$ 轮。

现在，根节点 r 观察自身。如果 $|U \cap T(r)| = 0$，算法就会停止，并返回一棵空树。否则，如果 $r \in U$，则不做任何操作。如果 $r \notin U$，它会找到某个任意节点 $v \in U$，并与 v 交换 位置——r 和 v 都会在 T 中通知各自的父节点和子节点它们交换了位置，也就是说， v 现在是树的根节点，而 r 现在占据了此前由 v 占据的位置。无论哪种情况，此时树的根节点都来自 U，我们接着递归处理根节点各个子节点的子树。

注意到，要找到节点 $v \in U$，节点 r 只需沿树向下遍历，每次选择前往其子树中包含 U 的某些节点的子节点。此外，还可以

[4]定理 3.5 在 CONGEST 中表述。显然，它可以在 Hybrid 中运行。一个潜在的有趣问题是，它是否可以在 Hybrid$_0$, 中执行，因为 CONGEST 可能假设图中的标识符来自某个特定的集合，例如 $[n]$。为了克服这一假设，我们执行引理 2.1，在所有节点˜上构造一棵虚拟树，并在 $O(1)$ 轮内利用它将节点重命名，使其标识符属于 CONGEST 中算法所假设的任意集合。节点仅在执行定理 3.5 期间使用这些新标识符，之后恢复使用其原始标识符。



˜图 1：定理 1.2 证明概览。我们首先创建弱直径为 $O(T_k)$、节点数大致相同的簇。然后，构造一棵由这些簇组成的逻辑树，其最大度为常数，深度为多对数级。在每个簇内，我们在该簇的节点上创建一棵逻辑二叉树。由于各簇的大小大致相同，我们可以确保簇内的树形状完全一致。我们确保簇树中任意两个相邻簇的内部树中的节点，都知道另一个簇的树中与自己对应的节点——也就是说，簇 C_i 中的节点 3 知道 C_j 中 C_ℓ 节点 3 的标识符，并且可以使用全局边直接与这些节点通信。构造完所有这些树后，我们将图中的所有 k 消息向上传递至簇树顶端，再向下传回所有簇，从而确保图中的每个节点都能收到所有消息。

并行执行所有递归步骤，因为递归操作是在互不相交的子树上进行的。最后，当某个子树只包含来自 $V \setminus U$的节点时，整个子树都会被移除，因为该子树的根节点将终止递归并返回一棵空树。因此，最终剩下的树是 T'，其中只包含来自 U 的节点，且深度为 $d' \leq d$。

总而言之，递归的每一步需要 $O(d)$ 轮，我们共进行 $O(d)$ 步递归，因此轮复杂度为 $O(d^2)$。□

现在我们可以证明定理 1.2 了。

证明。 该算法由几个阶段组成：聚类、集群链式连接、负载均衡和信息传播（见图 1）。聚类阶段确保将节点划分为大小相近且互不相交的簇，并使每个簇的弱直径较小。在集群链式连接阶段，我们将这些簇组织成一棵度为常数、深度为 polylog 的逻辑树，并让每个簇中的节点知晓其父子集群中的节点。在信息传播阶段，我们利用全局网络以及在集群链式连接阶段设计的链式连接，将所有令牌逐步向上传至根簇。随后，我们再沿树向下逐步传递令牌，使每个簇都能获得所有令牌。

我们首先使用引理 2.2，通过对每个节点持有的令牌数量求和来计算 k，耗时
˜ ˛$O(1)$ 轮。然后，根据引理 3.1，我们用 $O(T_k)$ 轮计算 T_k。

簇划分。 我们希望将图中的节点划分为大小大致相同且弱直径较小的簇。我们执行引理 3.7，并使用 k，得到 R，即˜簇领导者的集合。此过程需要 $O(T_k)$ 轮。

集群链式连接。 此阶段由两个子阶段组成。首先，我们为˜簇构建一棵逻辑树，记为 $\mathbb{T}_C$，其最大度为常数，深度至多为 $O(1)$。然后，在每个簇 C_i 内，我们按照一棵逻辑二叉树 $\mathbb{T}_i$对其节点排序。最后，我们使用内部树，将一个簇的节点与其父子集群中的节点关联起来。

˜**构建簇树**。 我们运行引理 2.1，在 $O(1)$ 轮内获得一棵虚拟树，该树˜覆盖了 G 最大度和深度均为常数的 $O(1)$。聚类阶段结束后，每个节点都知道自己是否为簇领导者。因此，我们定义函数 f，其中每个 $v \in V$ 将 $f(v)$ 设置为 $= 1$，若 $v \in R$，否则将 $f(v)$ 设置为 $= 0$。现在，我们使用引理 3.8 和 T，f计算一棵由簇领导者构成的树 $\mathbb{T}_C$，其˜ ˜最大度和深度均为常数 $O(1)$。这需要 $O(1)$ 轮。

匹配父节点和子节点簇节点。 观察簇 C_i。回想一下， C_i 中的每个节点都知道 C_i中的所有其他节点，因此它们各自计算出一棵由 C_i中的节点构成的逻辑二叉树 $\mathbb{T}_i$ ，其中 $r(C)$是 $\mathbb{T}_i$的根节点。我们希望 $\mathbb{T}_i$ 恰好有 $2k/T_k$ 个节点。由于$k/T_k \leq |C_i| \leq 2k/T_k$，那么我们只需将更多节点从 C_i 追加到 $\mathbb{T}_i$，必要时在 $\mathbb{T}_i$中将 C_i 中的每个节点重复两次。

令 C_i, C_j为两个簇，其领导者 $r(C_i)$ $r(C_j)$ 在簇树 $\mathbb{T}_C$– 中相邻。不失一般性，假设 $r(C_i)$ 是 $\mathbb{T}_C$ 中 $r(C_j)$ 的父节点。由于这些内部树具有相同数量的节点，且是通过虚拟构造而具有相同结构，因此 $\mathbb{T}_i, \mathbb{T}_j$具有相同的结构。令 $v_i \in \mathbb{T}_i, v_j \in \mathbb{T}_j$ 为其所在树中位置相同的两个节点（树的层级相同，且在该层中的索引相同）。现在，我们希望让 v_i 和 v_j 彼此知晓——也就是说，让它们获知对方的标识符，以便通过全局网络进行通信。

首先考虑分别位于 $\mathbb{T}_i$、$\mathbb{T}_j$ 根节点处的 $r(C_i)$ $r(C_j)$。根据 $\mathbb{T}_C$ 的构造，它们已知晓彼此的标识符。令 L_i, R_i 为 $r(C_i)$ 的子节点， L_j, R_j为 $r(C_j)$ 的子节点。节点 $r(C_i)$ 将标识符 L_i, R_i发送给 $r(C_j)$，而 $r(C_j)$ 将 L_j, R_j 发送给 $r(C_i)$。接着， $r(C_i)$ 将标识符 L_j, R_j 发送给 L_i, R_i ，同样地，$r(C_j)$ 与 L_j, R_j进行通信。通过全局网络完成所有这些操作需要 $O(1)$ 轮；对于 $\mathbb{T}_C$ 中任意一对相邻的 $r(C_i)$ $r(C_j)$，都可以并行执行。

注意到，现在 L_i, L_j 都知道彼此的标识符， R_i, R_j也是如此。因此，它们各自继续向下遍历相应的子树。由于这些树的深度为 $O(\log n)$，且处理树的每一层需要$O(1)$ 轮，因此总共需要 $O(\log n)$ 轮。

负载均衡。 每个簇 C_i 根据引理 3.6 在簇内均匀分配其节点的令牌。图中共有 k 个令牌，因此 C_i 中至多有 k 个令牌。此外， $|C_i| \geq k/T_k$，因此 C_i 可以对其令牌进行负载均衡，使每个节点至多有 T_k 个令牌。由于 C_i 的弱直径至多为 $4 \log nT_k$，这一阶段总共需要 $O(T_k)$ 轮˜。

信息传播。 我们现在要将所有令牌汇集到簇树˜$\mathbb{T}_C$的根簇 C_r 中。经过 $O(1)$ 次迭代，我们沿簇树向上传送每个簇中的令牌。每个节点最多持有 T_k 个令牌，并且与父簇中至多 2 个节点匹配，因此我们可以在 $2T_k$ 轮内通过全局通信网络发送这些令牌。每次迭代开始时，每个簇都会再次对上一轮收到的令牌进行负载均衡，这依据引理 3.6 完成。这样做是为了避免某个节点持有超过 T_k 个令牌，因为它可能从每个子簇接收多达 T_k 个令牌。随后，我们继续将令牌向上发送至根节点。根据簇树 $\mathbb{T}_C$ 的深度，这至多需要 $\log\ n$ 次迭代。考虑到每次迭代˜开始时进行的负载均衡，这一步需要 $\log n \cdot (T_k + 4T_k\lceil \log n \rceil) = O(T_k)$ 轮。

现在，根簇持有所有令牌。根簇再次在其节点之间对令牌进行负载均衡，使每个节点最多持有 T_k 个令牌。现在，我们以相同的˜方式沿簇树向下发送令牌。经过 $O(1)$ 次迭代，每个节点通过全局网络，将其持有的至多 T_k 个令牌发送给至多 2 个子簇中的匹配节点。我们在每次迭代中再次进行负载均衡，以防止每个节点积累超过 T_k 个令牌。这是必要的，因为匹配可能会将 C_i 中的 2 个节点匹配到 C_j中的一个节点。现在每个簇都持有所有令牌。每个节点通过本地网络泛洪所有令牌，持续 $4T_k\lceil\log n\rceil$（即一个簇的弱直径），使其簇中的所有节点都能获知所有令牌。此阶段需要˜$\log\ n \cdot (T_k + 4T_k\lceil\log n\rceil) + 4T_k\lceil\log n\rceil = O(T_k)$ 轮。

现在我们解决了 k-信息传播问题，因为 G 中的每个节点都知道所有令牌。将所有阶段的耗时相加，该算法需要 $O(T_k)$ 轮。 □

3.4 普适最优聚合

现在，我们通过从 k-信息传播问题进行双向归约，给出 k-聚合问题的普适最优解。

˜**定理 1.4.** 对于 k-聚合˜问题，在 Hybrid 中存在一个为 $\Omega(T_k)$ 轮的普适下界；并且存在一个确定性算法，可在 Hybrid 中用 $O(T_k)$ 轮解决该问题（甚至在 Hybrid$_0$ 中也可以）。

证明。 我们展示从 k-信息传播问题到 k-聚合问题的双向归约。给定一个图 $G = (V, E)$，如果存在一个能在 t 轮内解决 k-聚合问题的算法，我们就能在 $O(t)$ 轮内解决 k-信息传播问题，反之亦然。

首先，如果存在一个能在 t 轮内解决 k-聚合问题的算法，我们就可以用它在 $O(t)$ 轮内解决 k-˜信息传播问题。直观来说，由于我们有 k 个令牌需要传播，而且可以在 t 轮内让全局知晓 k 个聚合结果，因此我们希望将这些 k 个令牌放入各个值的不同索引中，并让其余节点在其他索引上发送单位元。唯一的问题是，所有持有令牌的节点需要协调各节点应将令牌放入哪些索引中，以便将 k 个令牌与 k 个索引匹配起来。这可以通过以下算法实现，该算法运行 $O(1)$ 轮。

首先，我们使用引理 2.1 和 3.8 构造一棵包含所有至少有一个令牌需要˜广播的节点的树。得到这棵树 T后，我们可以在 $O(1)$ 轮内计算出每个节点的子树（包括其自身）持有多少令牌。我们将其记为 $\ell(v)$。具体做法是从树的最后一层开始向上传递信息，在每个节点汇总令牌数量，然后将结果发送给父节点。接着，我们从根节点开始分配索引。

如果根节点持有 m 个令牌，就为自己保留前 m 个索引，并告知第一个子节点应从 m开始分配；如果存在第二个子节点，则告知它应从 $m + \ell$（第一个子节点）开始分配。根节点以此方式继续处理所有子节点。树中较低层的节点也以同样的方式继续处理。这样便在图中所有 k 个令牌与聚合的 k个索引之间建立了双射，并正确地为每个持有令牌的节点分配了所有索引。然后，我们可以在 t 轮内运行 k-聚合算法，所有节点都能获知 k 个令牌。总计，该˜˜算法需要 $t + O(1) = O(t)$ 轮。这表明 T_k 是k-聚合问题的普适下界。

˜反过来，我们证明确实可以确定性地在 $O(T_k)$ 轮内解决 k-聚合。我们注意到，一旦只有一个节点获知所有 k 聚合函数的结果，我们就可以根据定理 1.2 在 $O(T_k)$ 轮内将其˜传播出去。我们采用与定理 1.2 的证明类似的步骤。首先，我们使用相同的过程对节点进行聚类。然后，我们在每个互不相交的簇内计算k 中间聚合，并根据引理 3.6 在簇内对其进行负载均衡。这样，每个

节点至多持有 T_k 个聚合结果。然后，我们在证明中使用簇树和簇链式连接，将中间聚合结果沿簇树向上发送到根簇。在˜每个步骤中，我们再次进行负载均衡。由于所构造簇树的深度至多为 O(1)，这一˜过程在 $O(T_k)$轮内完成。一旦所有信息都存储在根簇中，我们就在其中泛洪传播这些信息˜，并在本地计算最终的 k 聚合结果。根据每个簇的弱直径，此步骤耗时 $O(T_k)$轮，该弱直径至多为 $4T_k \log n$。

最后，我们使用定理 1.2，在 $O(T_k)$ 轮内，将根簇中某个节点的 k 聚合结果传播到整个˜图。

□

4 应用

4.1 $(1+\epsilon)$-近似无权图中的全点对最短路径

现在我们证明定理 1.9。

定理 1.9。 对于任意 $\epsilon \in (0,1)$，存在一种随机化算法，能够在无权图中以 $(1+\epsilon)$-近似计算全点对最短路径，并在 Hybrid$_0$，中以 $\widetilde{O}(T_n/\epsilon^2)$ 轮高概率完成。

为此，我们需要采用 [45] 中的新颖的 $(1+\epsilon)$-近似单源最短路径结果。

定理 4.1 (来自 [45] 的定理 3.29)。 在$(1+\epsilon)$时间内，可以在Hybrid 模型中以高概率计算 *SSSP* 的近似值。

我们注意到， [45] 的算法高度依赖于 [44]中提出的新颖算法。现在我们继续证明定理 1.9。算法概览见算法 1。

˜证明。 我们首先使用引理 3.1 在 $O(T_n)$轮次内计算 T_n ，因此从现在起我们可以假设所有节点都知道该值。接下来，我们使用定理1.2 在˙$O(T_n)$ 轮次内广播所有节点的标识符。从现在起，我们可以假设我们处于Hybrid 中，而不是Hybrid$_0$，因此能够执行定理4.1等算法。我们在 $O(T_n)$ 轮次内使用引理 3.7和˙$k = n$，对节点进行聚类，使得我们知道一个由 R 簇领导者组成的集合，每个簇 C 的弱直径至多为 $4T_n\lceil \log n \rceil$，并且 $n/T_n \le |C| \le 2n/T_n$。

现在注意到，由于各个簇互不相交，且每个簇的大小至少为 n/T_n，因此簇的数量至多为 T_n ，从而有 $|R| \le T_n$。根据定理 4.1，可以在 $= \tilde{O}(T_n/\epsilon^2)$ 轮内，以高概率从 R 中的所有节点计算到 $\tilde{O}$ 中整个图的 $(1+\epsilon)$- 近似距离。将计算得到的近似距离记为 d。

每个节点 v 都能获知其 $x = (4T_n\lceil \log n \rceil)/\epsilon$ 邻域，记为 $B_x(v)$。这需要 $O(x)$ $=O(T_n/\epsilon)$ $= O(T_n/\epsilon^2)$ 轮。随后，每个节点 v 广播其在 R 中距离最近的节点，记为 $c_v \in R$，以及无权距离 $d(v, c_v)$。由于每个节点广播 $O(1)$ 条消息，根据定理 1.2，这需要 $O(T_n)$ 轮。

最后，每个节点 v 按如下方式近似其到每个节点 w 的距离。若 $w \in B_x(v)$，则v 知道到 w 的精确距离，因为图是无权图，因此设定 $\delta(v, w) = d(v, w)$。ˆ否则， v设定 $\delta(v, w) = d(v, c_w) + d(w, c_w)$，其中 c_w 是 R 中距离 w最近的节点。注意，v 知道 c_w 和 $d(w, c_w)$，因为 w 在上一步广播了这些值。

我们通过证明 δ 是 d 的 $(1+\epsilon)$ 近似来完成证明。

ˆ若 $w \in B_x(v)$，则 $\delta(v,w) = d(v,w)$。否则， $d(v,w) > x = 4T_n\lceil \log n \rceil/\epsilon$，且 $\delta = d(v,c_w) +$ˆ$d(w,c_w)$。我们首先证明 $\delta(v,w) \ge d(v,w)$。由于 $d(v,c_w)$ 是一个有效的 $(1+\epsilon)$-近似，ˆ$d(v,c_w) \ge d(v,c_w)$，因此 $\delta(v,w) \ge d(v,c_w) + d(w,c_w) \ge d(v,w)$，其中最后一个不等式是由三角不等式得出的。现在我们从上方界定 $\delta(v,w)$。有 $d(v,w) > 4T_n\lceil \log n \rceil/\epsilon$

且 $d(w,c_w) \le 4T_n\lceil\log n\rceil$，因为每个簇的弱直径至多为 $4T_n\lceil\log n\rceil$。因此，$d(w,c_w) \le 4T_n\lceil\log n\rceil < \epsilon \cdot d(v,w)$。由此可得以下结论。

$$\begin{aligned}\delta(v,w) &= \hat{d}(v,c_w) + d(c_w,w) \le (1+\epsilon)\cdot d(v,c_w) + d(w,c_w)\\ &\le (1+\epsilon)\cdot(d(v,w)+d(w,c_w)) + d(w,c_w)\\ &= (1+\epsilon)\cdot d(v,w) + (2+\epsilon)\cdot d(w,c_w)\\ &< (1+\epsilon)\cdot d(v,w) + (2\epsilon+\epsilon^2)\cdot d(v,w)\\ &= (1+3\epsilon+\epsilon^2)\cdot d(v,w) \overset{\epsilon'=3\epsilon+\epsilon^2}{=} (1+\epsilon')\cdot d(v,w)\end{aligned}$$

注意，我们实现了 $(1+\epsilon')$ 近似，其中 $\epsilon' = 3\epsilon+\epsilon^2$。由于 $\epsilon \in (0,1)$，因此 $\epsilon' < 4\epsilon$，所以可以使用 $\epsilon/4$ 运行上述所有步骤，以获得所需结果。 □

算法 1： $(1+\epsilon)$-近似无权全源最短路径

1 计算 T_n2 广播所有节点的标识符。3 运行引理 3.7，并使用 $k=n$ 获取簇领导者集合 R。
^4 从 R 中的每个节点运行 $(1+\epsilon)$- 近似单源最短路径。将计算出的距离记为 d。5 了解 $x = (4T_n\lceil\log n\rceil)/\epsilon$ 邻域。6 根据定理 1.2，每个节点 v 广播其最近的簇领导者 $c_v \in R$ 和 $d(v, c_v)$。7 节点 v 将其到任意 $w \in V$ 的距离近似为：

$$\delta(v,w) = \begin{cases} d(v,w), & w \in B_x(v)\\ \hat{d}(v,c_w) + d(c_w,w), & \text{otherwise}\end{cases}$$

4.2 加权图中的全点对最短路径近似

我们展示了若干用于计算加权图中全点对最短路径近似的算法。首先介绍定理 1.10。

定理 1.10。 Hybrid$_0$，中存在一种确定性算法，对于给定的图 $G=(V,E,\omega)$，能在 $\widetilde{O}(2^{1/\epsilon}T_n)$ 轮次内计算全点对最短路径的 $(\epsilon\cdot\log n)$ 近似。

我们采用一种广为人知的技术来计算稀疏生成器——一种边数更少、同时能保持原图中距离良好近似的子图。

定理4.2 (文献 [43] 中的推论 3.16)。 设 $G=(V,E,\omega)$ 为一个加权图。CONGEST 中存在一种确定性算法，可在 ~的 $O(1)$ 轮内计算一个大小为 $\widetilde{O}(kn^{1+1/k}\log n)$ 的$(2k-1)$-伸展稀疏生成器。

为证明定理 1.10，我们执行定理4.2，然后广播所得的稀疏生成器。

~证明。 我们首先使用定理 1.2，在 $O(T_n)$ 轮内广播所有节点在 $_0$，中的标识符。从现在起，我们可以假设处于 Hybrid 而非 Hybrid$_0$，，因此可以执行定理4.2等算法。我们使用 $k = \epsilon\log n/2$ 执行定理4.2，得到一个含有 $O(k\cdot n^{1+1/k}\cdot\log n) = \widetilde{O}(4^{1/\epsilon}\cdot n)$ 条边的稀疏生成器。根据引理 3.3，$T_{4^{1/\epsilon}n} = O(2^{1/\epsilon}T_n)$，并根据定理 1.2，我们在 $\widetilde{O}(2^{1/\epsilon}\cdot T_n)$ 轮内使稀疏生成器的边为全局已知，从而得到 APSP 的 $(2k-1) = \epsilon\log n - 1 < \epsilon\log n$ 近似。 □

我们现在要证明以下结论。

定理 1.12。 对于任意整数 $\alpha \geq 1$，存在一种随机化算法，可在加权图上于Hybrid$_0$，中以高概率在 $\tilde{O}(\alpha \cdot n^{1/(3\alpha+1)}(T_n)^{2/(3+1/\alpha)} + \alpha \cdot T_n)$ 轮内计算 *APSP* 的$(4\alpha-1)$-近似。

在此之前，我们必须介绍骨架图这一著名概念，该概念最早由 [48]提出。骨架图通过以概率 $1/x$对每个节点进行采样来构造，其中x为某个取值。其主要性质是：给定 u， $v \in V$，在 u 和 v之间存在某条最短路径$P_{u,\ v}$，且每隔 $O(x)$跳就会有一个被采样的节点。我们使用 $[6, 38]$ 关于 Hybrid 中骨架图的两个引理。

引理 4.3 ([6]中的引理 4.2**)**。设 $M \subseteq V$ 是 $G = (V, E)$ 的节点子集，通过以至少 $1/x$的概率独立采样每个节点得到。存在常数 $\xi > 0$，使得对于每个 $u, v \in V$，若 $hop(u, v) \geq \xi x \ln n$，则存在一条从 u 到 v的最短路径 P，使得 P中任意包含至少 $\xi x \ln n$ 个节点的子路径 Q 都包含 M中的一个节点，且该结论以高概率成立。

以下说明，骨架图中骨架节点之间的距离与原图中这些节点之间的距离相同，并且我们可以在 Hybrid 中构造骨架图。

引理 *4.4*（ [38] 中的引理 *C.2*）。设 G 是一个含有 n 个节点的连通图，通过以至少 $1/x$ 的概率对 G 中的每个节点进行采样至 $V_{\mathcal{S}}$，得到其骨架图 $\mathcal{S} = (V_{\mathcal{S}}, E_{\mathcal{S}})$。 $\mathcal{S}$ 的边为 $E_{\mathcal{S}} = \{\{u, v\} | u, v \in V_{\mathcal{S}},$ 跳 $(u, v) \leq h\}$（其中 $h := \xi x \ln n$ 是引理 *4.3* 中的参数），边权重为 $d^h(u, v)$，适用于 $\{u, v\} \in E_{\mathcal{S}}$。那么 $\mathcal{S}$ 是连通的，并且对于任意 $u, v \in V_{\mathcal{S}}$，我们以高概率有 $d_G(u, v) = d_{\mathcal{S}}(u, v)$。此外，在 *Hybrid* 中， $\mathcal{S}$ 可以在 $O(h) = O(1/x)$ 轮内计算得到。

算法 2： $(4\alpha-1)$-近似加权全点对最短路径

1 广播所有节点的标识符。2 计算 T_n。记作 $t = n^{1/(3\alpha+1)} \cdot (T_n)^{2/(3+1/\alpha)}$。3 以采样概率 $1/t$ 计算骨架图 $G_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}}, \omega_{\mathcal{S}})$。4 为 $G_{\mathcal{S}}$ 计算一个 $(2\alpha-1)$-伸展稀疏生成器，记作 K。5 广播 K 的边。在本地计算 $V_{\mathcal{S}}$ 中所有节点之间距离的一个 $(2\alpha-1)$ 近似，记作 d。6 学习 $h = \xi t \ln n$ 邻域，其中 ξ 是引理 4.3 中的常数。7 每个节点 $v \in V$ 将其 h-邻域中具有最小 $d^h(v, v_s)$ 的骨架节点记作 $v_s \in V_{\mathcal{S}}$，并广播 v_s 和 $d^h(v, v_s)$。8 节点 v 近似其到任意 $w \in V$ 的距离，方法如下：

$$\delta(v, w) = \min\{d^h(v, w), d^h(v, v_s) + \hat{d}(v_s, w_s) + d^h(w_s, w)\}$$

现在我们证明定理 1.12。我们的算法概述见算法 2。

˜证明。 我们首先使用定理 1.2，在 $O(T_n)$ 轮中广播所有节点的标识符。从现在起，我们可以假设处于Hybrid而不是Hybrid$_0$，中，因此能够执行˜算法，例如引理 4.4。接下来，我们使用引理 3.1，在 $O(T_n)$ 轮中计算 T_n，然后记为 $t = n^{1/(3\alpha+1)} \cdot (T_n)^{2/(3+1/\alpha)}$。注意，在接下来的算法中，我们致力于˜达到 $O(\alpha \cdot t + \alpha \cdot T_n)$ 的轮复杂度。

使用引理 4.4，我们计算骨架图 $G_{\mathcal{S}} = (V_{\mathcal{S}}, E_{\mathcal{S}}, \omega_{\mathcal{S}})$，其采样概率为 $1/t$，在 $O(t)$ 轮内完成。注意，$|V_{\mathcal{S}}| = \Theta(n/t)$ 以高概率成立。现在，我们使用定理4.2创建 $2\alpha-1$ 的 $G_{\mathcal{S}}$ 稀疏生成器，记为 K ——定理4.2的每一轮都通过 $G_{\mathcal{S}}$ 使用 G 的本地边进行模拟，因此需要 $O(t)$ 轮。由于定理4.2需要 $O(1)$ 轮，我们的整个模拟需要 $O(t)$ 轮。

根据定理4.2，K 有 $\tilde{O}(\alpha \cdot |V_{\mathcal{S}}|^{1+1/\alpha}) = \tilde{O}(\alpha \cdot (n/t)^{1+1/\alpha})$ 条边。令 $x = \max\{(n/t)^{1+1/\alpha}, n\}$，并使用引理 3.1 在 $O(T_x)$ 轮内计算 T_x。根据定理 1.2，我们可以在 $O(\alpha \cdot T_x)$ 轮内广播所有 K。

我们希望证明 $T_x = O(t + T_n)$。若 $x = n$，则显然有 $T_x = T_n$。否则，

$$\begin{aligned}
x &= (n/t)^{1+1/\alpha} = n^{1+1/\alpha} \cdot t^{-(1+1/\alpha)} \\
&= n^{1+1/\alpha} \cdot \left(n^{1/(3\alpha+1)} \cdot (T_n)^{2/(3+1/\alpha)}\right)^{-(1+1/\alpha)} \\
&= n^{1+1/\alpha-(1+1/\alpha)/(3\alpha+1)} \cdot (T_n)^{-(2+2/\alpha)/(3+1/\alpha)} \\
&= n^{((\alpha+1)/\alpha)\cdot(1-1/(3\alpha+1))} \cdot (T_n)^{-(2\alpha+2)/(3\alpha+1)} \\
&= n^{((\alpha+1)/\alpha)\cdot(3\alpha/(3\alpha+1))} \cdot (T_n)^{-(2\alpha+2)/(3\alpha+1)} \\
&= n^{3\cdot(\alpha+1)/(3\alpha+1)} \cdot (T_n)^{-(2\alpha+2)/(3\alpha+1)} \\
&= n^{1+2/(3\alpha+1)} \cdot (T_n)^{-(2\alpha+2)/(3\alpha+1)}.
\end{aligned}$$

根据引理 3.3，$T_x = O(\sqrt{x/n} \cdot T_n)$，因此

$$\begin{aligned}
T_x &= O(\sqrt{x/n} \cdot T_n) \\
&= O(\sqrt{n^{2/(3\alpha+1)} \cdot (T_n)^{-(2\alpha+2)/(3\alpha+1)}} \cdot T_n) \\
&= O(n^{1/(3\alpha+1)} \cdot (T_n)^{-(\alpha+1)/(3\alpha+1)} \cdot T_n) \\
&= O(n^{1/(3\alpha+1)} \cdot (T_n)^{2\alpha/(3\alpha+1)}) \\
&= O(n^{1/(3\alpha+1)} \cdot (T_n)^{2/(3+1/\alpha)}) \\
&= O(t).
\end{aligned}$$

因此，在两种情况下均有 $T_x = O(t + T_n)$。于是，根据定理 1.2，我们可以在 K 内用 $O(\alpha \cdot T_x) = O(\alpha \cdot (t + T_n))$ 轮广播所有 K 边。利用这些信息，每个节点在本地计算 $G_{\mathcal{S}}$ 中距离的一个 $(2\alpha-1)$ 近似。

接下来，每个节点 v 找到其 h-邻域，其中 ξ 是引理 4.3 中的常数。根据引理 4.3，每个节点 v 在其 h-邻域内至少能看到一个骨架节点。因此，v 将其 h-邻域中具有最小 $d^h(v, v_s)$ 的骨架节点记为 $v_s \in V_{\mathcal{S}}$，并广播 v_s 和 $d^h(v, v_s)$。根据定理 1.2，这需要 $\tilde{O}(T_n)$ 轮，因为每个节点广播 $O(1)$ 个值。

最后，节点 v 将其到任意节点 w 的距离近似为 $\delta(v, w) = \min\{d^h(v, w), d^h(v, v_s) + d(v_s, w_s) + d^h(w_s, w)\}$。还需证明这是一个 $(4\alpha-1)$ 近似。

设 $v, w \in V$。如果它们之间存在一条长度小于 h 跳的最短路径，那么 $\delta(v,w) = d^h(v,w) = d(v,w)$。否则，所有最短路径的长度都大于 h 跳；根据引理 4.3，其中一条路径上存在一个骨架节点 s。此外，不失一般性地，s 位于 v 的 h-邻域内。由于 s 位于从 v 到 w 的一条最短路径上，因此也有 $d^h(v,s) = d(v,s)$。最后，有 $d^h(w, w_s) \le d(w,s)$——这是因为，要么存在一条从 w 到 s、长度至多为 h 跳的最短路径，此时 $d^h(w, w_s) \le d^h(w,s) = d(w,s)$；要么存在一条从 w 到 s、经过骨架节点 s' 的路径，且该节点也位于 w 的 h-跳邻域内，此时 $d^h(w, w_s) \le d^h(w, s') \le d(w,s)$。

利用以上所有结果，我们证明如下结论。

$$\begin{aligned}
\delta(v,w) &= d^h(v,v_s) + \hat{d}(v_s,w_s) + d^h(w_s,w) \le d^h(v,v_s) + (2\alpha-1)d(v_s,w_s) + d^h(w_s,w) \\
&\le d^h(v,v_s) + (2\alpha-1)(d(v_s,v) + d(v,w) + d(w,w_s)) + d^h(w_s,w) \\
&\le d^h(v,v_s) + (2\alpha-1)(d^h(v_s,v) + d(v,w) + d^h(w,w_s)) + d^h(w_s,w) \\
&\le (2\alpha-1)d(v,w) + 2\alpha(d^h(v,v_s) + d^h(w_s,w)) \\
&\le (2\alpha-1)d(v,w) + 2\alpha(d^h(v,s) + d^h(w_s,w)) \\
&= (2\alpha-1)d(v,w) + 2\alpha(d(v,s) + d^h(w_s,w)) \\
&\le (2\alpha-1)d(v,w) + 2\alpha(d(v,s) + d(s,w)) \\
&= (2\alpha-1)d(v,w) + 2\alpha \cdot d(v,w) \\
&= (4\alpha-1)d(v,w)
\end{aligned}$$

注意，近似值从不低估，即 $\delta(v,w) \ge d(v,w)$，因为它对应于实际路径，因此我们完成了。

□

4.3 通过学习谱稀疏器近似割

我们将普适最优广播应用于不同的稀疏化工具，尤其是割稀疏化器。我们采用了 CONGEST 模型中的以下结果。

定理 *4.5*（将 [36]，中的定理 *5* 重新表述）。存在一种 *CONGEST* 算法，给定一个图 $G=(V,E,\omega)$ 和任意 $\epsilon > 0$，可计算出一个图 $H=(V,E,\tilde{\omega})$，使得对于任意割 $S \subset V$，都有 $(1-\epsilon)cut_H(S) \le cut_G(S) \le (1+\epsilon)cut_H(S)$，且 $|\tilde{E}| = \tilde{O}(n/\epsilon^2)$ 以高概率成立。该算法以 $\tilde{O}(1/\epsilon^2)$ 轮次运行，且以高概率完成。

注意到，我们可以在 Hybrid$_0$ 中使用未知标识符运行该算法，耗时 $\tilde{O}(T_n + 1/\epsilon^2)$ 轮，因为根据定理 1.2，我们可以在 $O(T_n)$ 轮内广播所有标识符，然后照常运行该算法。现在我们很容易得到定理 1.13。

定理 1.13. 对于任意 $\epsilon \in (0,1)$，存在一个 Hybrid$_0$ 算法，可在 $\tilde{O}(T_n/\epsilon + 1/\epsilon^2)$ 轮内运行，以高概率使每个节点都能在本地计算出图中所有割的一个 $(1+\epsilon^2)$ 近似。这为许多问题提供了近似解，包括最小割、最小 s-t 割、最稀疏割和最大割。

证明. 我们在 $\tilde{O}(T_n + 1/\epsilon^2)$ 中运行定理 4.5，以高概率得到一个含有 $\tilde{O}(n/\epsilon^2)$ 条边的割稀疏器。根据引理 3.3 和定理 1.2，我们可以在 $O(T_n/\epsilon)$ 轮内广播该稀疏器，并且每个节点都可以在本地计算任意割的近似。

□

5 不同图族上的 T_k 估计

5.1 路径和环图

可以说，最简单、最值得考虑并计算 T_k 的图，是由 n 个节点组成的路径图 P_n。根据定义，$T_k = \max_{v \in V}\{T_k(v)\}$，因此只需找出 $\arg\max_{v \in V} T_k(v)$，并计算其 $T_k(v)$ 值。$T_k(v)$ 值大致对应这样一个 v 的邻域大小：它具有足够的带宽，能够在 $T_k(v)$ 轮内接收并转发来自图中其余部分的 k 消息。邻域越稀疏，$T_k(v)$ 就越高。在路径图中，端点节点的 $T_k(v)$ 值最高。我们将其形式化如下。

引理 5.1。 令 $v, w \in V$。若对所有 $t \in \mathbb{N}^+$ $|B_t(v)| \leq |B_t(w)|$，均有 $T_k(v) \geq T_k(w)$。

证明。 这可直接由定义 2 得出。若对于任意 t, $|B_t(v)| \leq |B_t(w)|$，则满足定义中方程 $|B_t(v)| \geq k/t$ 的最小 t 对于 v更大，因此 $T_k(v) \geq T_k(w)$。 □

推论 5.2。 设 $v, w \in V$。如果 $B_{T_k(v)}(v)$ 与 $B_{T_k(v)}(w)$同构，那么 $T_k(w) = T_k(v)$。

引理 5.1 的条件显然适用于端点节点 $r, l \in P_n$ 相对于任何其他 $v \in P_n$，因为它们周围的球只会向一侧扩展。因此，只需计算更简单的值 $T_k(r)$ 或 $T_k(l)$。

现在我们计算 $T_k(r)$，其中 r 是路径图 P_n中最右侧的节点：

引理 5.3。 对于具有 n 个节点的路径图，$T_k = \begin{cases} \Theta(\sqrt{k}) \; k = O(n^2) \; D & k = \Omega(n^2) \end{cases}$。

证明。 $|B_t(r)| = t + 1$，根据定义 2，我们得到求最小 $t \leq D = n - 1$ 的方程，使得 $t + 1 \geq k/t \implies t^2 + t \geq k$；只有当 $k \leq D^2 + D = O(n^2)$ 时，该方程才成立。否则，渐近解为 $t = \Theta(\sqrt{k})$。 □

推论 5.4。 对于环图 C_n，其值 T_k 与 P_n相同，因为任意 $v \in V$ 周围的 $|B_t(v)|$ 的大小都是 $2t + 1$，因此渐近解保持不变。

5.2 方格图

现在我们来计算某个维度 d 下方格网格的 T_k，即 $n = m^d$。根据引理 5.1，只需考察端点节点即可计算 T_k，因为与其他节点相比，它们的邻域较小。作为热身，当 $d = 2$ 时，$|B_r(v)| = |B_{r-1}(v)| + r + 1$，因为每次扩展半径时都会增加一条副对角线。通过归纳可得 $|B_r(v)| = \sum_{i=0}^{r} i + 1 = \sum_{i=1}^{r+1} i = \frac{(r+1)(r+2}{2} = \frac{r^2+3r+2}{2})$。将其代入定义 2，可得 $T_k = \Theta(k^{1/3})$ 或$T_k = D$。我们将在下面的结论中推广这一论证。

引理 5.5。 设 $G = (V, E)$ 为一个 d维方形网格，其中 $|V| = n = m^d$。设 w 为一个角节点，并令 $N_r(w) = \{v \mid d(v, w) = r\}$，即距离角节点 w恰好为 r 的节点集合。若 $r < n^{1/d}$，则 $|N_r(w)| = \binom{r+d-1}{d-1}$。

证明。 集合 $N_r(w)$ 可以表示为 $N_r(w) = \{v \in \mathbb{N}^d | \sum_{i=1}^{d} v_i = r\}$，因为方形网格中的距离遵循 L_1 度量，并且不失一般性，角节点 w 对应于 $\mathbb{N}^d$ 中的 $\vec{0}$ 向量。因此，$N_r(w)$ 中的元素数量由 $\binom{r+d-1}{d-1}$ 给出，如 [49] 所示。我们要求 $r < n^{1/d}$，这样弱合成的数量才能正确描述环大小。这样，合成中的任何分量都不会超过边长 $m = n^{1/d}$，因此每种合成都对应于网格图中唯一且存在的节点。
</final>

□

引理 5.6。 若 $r < n^{1/d}$，则根据引理 *5.5* 的条件，$|B_r(v)| = |N_0(w) \cup \cdots \cup N_r(w)| = \binom{r+d}{d}$.

证明。 观察以下转移。

$$|B_r(v)| = |N_0(w) \cup \cdots \cup N_r(w)| = \sum_{i=0}^{r} |N_i(w)| = \sum_{i=0}^{r} \binom{i+d-1}{d-1} \overset{m=d-1}{=} \sum_{i=0}^{r} \binom{i+m}{m}$$

$$\overset{j=i+m}{=} \sum_{j=m}^{m+r} \binom{j}{m} \overset{\text{Pascal's Triangle}}{=} \sum_{j=m}^{m+r} [\binom{j+1}{m+1} - \binom{j}{m+1}] \overset{\text{Telescoping Sum}}{=} \binom{m+r+1}{m+1} = \binom{r+d}{d}$$

□

由此可得关于 d 维方格网格的以下定理。

Th定理 1.6。 令 $G = (V, E)$ 为一个 d 维方形网格，其中 $|V| = n = m^d$.

$$T_k = \begin{cases} \Theta(k^{1/(d+1)}) & k = O(n^{(d+1)/d}) \\ O(D) = O(dn^{1/d}) & k = \Omega(n^{(d+1)/d}) \end{cases}$$

证明。 根据引理 5.6，我们注意到，对于角节点 v， $|B_r(v)|$ 是关于 r 的 d 次多项式，因为 $\binom{r+d}{d} = (r+d)\cdot\cdots\cdot(r+1)\ d!$ 。

要求 $r < m = n^{1/d} = D/d$ 意味着近似解与 D 相差一个 d 因子，因此我们的计算在 d 因子范围内近似正确。

因此，定义 2 中的方程 $|B_r(v)| \geq k/r$ 变为 $r \cdot \binom{r+d}{d} \geq k$，这是一个 $d+1$ 次多项式，其常数项为 $a_0 = \Theta(k)$。 d 维网格的直径为 $dk = dn^{1/d}$，其中 k 是边长。我们要寻找小于 $D = dk = dn^{1/d}$ 的最小解 r 。因此，考虑到 r 也必须至多为 D，渐近解如下。

$$T_k = \begin{cases} \Theta(k^{1/(d+1)}) & k = O(D^{d+1}) = O(n^{(d+1)/d}) \\ O(D) = O(dn^{1/d}) & k = \Omega(D^{d+1}) = \Omega(n^{(d+1)/d}) \end{cases}$$

□

参考文献

[1] Yehuda Afek、GalGiladi 和 BoazPatt-Shamir。分布式计算与云。载于分布式系统的稳定性、安全性与安全保障：第*23*届国际研讨会，*SSS 2021*，线上活动，*2021*年*11*月*17–20*日，论文集 *23*，第1–20页。施普林格，2021年。

[2] Yehuda Afek、Gad M. Landau、Baruch Schieber 和 Moti Yung。多媒体的力量：结合点对点网络与多址网络。《信息与计算》，84(1):97– 118，1990年1月。[3] Ioannis Anagnostides 和 Themis Gouleakis。Hybrid 模型中的确定性分布式算法与下界。发表于第*35*届国际分布式计算研讨会（*DISC 2021*），第209卷，第5:1–5:19页，2021年。[4] John Augustine、Keerti Choudhary、Avi Cohen、David Peleg、Sumathi Sivasubramaniam 和 Suman Sourav。分布式图实现。《*IEEE*并行与分布式系统汇刊》，33(6):1321–1337，2021年。[5] John Augustine、Mohsen Ghaffari、Robert Gmyr、Kristian Hinnenthal、Christian Scheideler、Fabian Kuhn 和 Jason Li。节点容量受限网络中的分布式计算。发表于第*31*届 *ACM* 算法与体系结构并行性研讨会，第69–79页，2019年。[6] John Augustine、Kristian Hinnenthal、Fabian Kuhn、Christian Scheideler 和 Philipp Schnei- der。混合网络模型中的最短路径。发表于第十四届*ACM-SIAM*离散算法年度研讨会论文集，第1280–1299页。工业与应用数学学会，2020年。

[7] Florent Becker、Antonio Fernandez Anta、Ivan Rapaport 和 Eric Reémila。简短公告：拥塞团模型的层次结构，从广播到单播。载于*2015*年*ACM*分布式计算原理研讨会论文集，第 167–169 页，2015 年。

[8] Florent Becker、Pedro Montealegre、Ivan Rapaport 和 Ioan Todinca。广播拥塞团模型中局部性的影响。*SIAM* 离散数学杂志，34(1):682–700，2020。[9] Keren Censor-Hillel、Dean Leitersdorf 和 Volodymyr Polosukhin。通过预言机模拟在混合网络模型中进行距离计算。载于第*38*届国际计算机科学理论方面研讨会（*STACS 2021*），2021。[10] Keren Censor-Hillel、Dean Leitersdorf 和 Volodymyr Polosukhin。关于分布式计算中的稀疏性认知。载于第*33*届 *ACM* 算法与体系结构并行性研讨会论文集，第151–161页，2021。[11] Lijie Chen 和 Ofer Grossman。广播拥塞团：植入团与伪随机生成器。载于*2019*年 *ACM* 分布式计算原理研讨会论文集，第248–255页，2019。[12] Sam Coy、Artur Czumaj、Michael Feldmann、Kristian Hinnenthal、Fabian Kuhn、Christian Scheideler、Philipp Schneider 和 Martijn Struijs。混合通信网络中的近最短路径路由。载于第*25*届分布式系统原理国际会议（*OPODIS 2021*），第11:1–11:23页，2022。[13] Sam Coy、Artur Czumaj、Christian Scheideler、Philipp Schneider 和 Julian Werthmann。单位圆盘图混合通信网络的路由方案。*arXiv* 预印本 *arXiv:2210.05333*，2022。[14] Wenkai Dai、Michael Dinitz、Klaus-Tycho Foerster 和 Stefan Schmid。简短公告：最小化需求感知混合网络拓扑中的拥塞。载于第*36*届分布式计算国际研讨会（*DISC 2022*），第42:1–42:3页，2022。[15] Andrew Drucker、Fabian Kuhn 和 Rotem Oshman。关于拥塞团模型的能力。载于*2014*年 *ACM* 分布式计算原理研讨会论文集，第367–376页，2014。[16] Nathan Farrington、George Porter、Sivasankar Radhakrishnan、Hamid Hajabdolali Bazzaz、Vikram Subramanya、Yeshaiahu Fainman、George Papen 和 Amin Vahdat。Helios：模块化数据中心的混合电 / 光交换架构。载于 *ACM SIGCOMM 2010* 会议论文集，第339–350页，2010。[17] Michael Feldmann、Kristian Hinnenthal 和 Christian Scheideler。稀疏图最短路径的快速混合网络算法。载于第*24*届分布式系统原理国际会议（*OPODIS 2020*），第31:1–31:16页，2021。[18] Juan A. Garay、Shay Kutten 和 David Peleg。一种用于最小权生成树的亚线性时间分布式算法。*SIAM* 计算杂志，27(1):302–316，1998年2月。[19] Mohsen Ghaffari。重新审视分布式广播：迈向普适最优性。载于自动机、语言与编程，第638–649页，2015。[20] Mohsen Ghaffari 和 Bernhard Haeupler。平面网络的分布式算法 II：低拥塞捷径、最小生成树和最小割。载于第二十七届 *ACM-SIAM* 离散算法年度研讨会论文集，第202–219页。工业与应用数学学会，2016。

[21] Mohsen Ghaffari 和 Jason Li。通过并行算法转换，在近似混合时间内实现新的分布式算法。发表于第*32*届国际分布式计算研讨会（*DISC 2018*），第 31:1–31:16 页，2018 年。[22] Mohsen Ghaffari 和 Merav Parter。平面图中的近最优分布式 DFS。发表于第*31*届国际分布式计算研讨会（*DISC 2017*），2017 年。[23] Mohsen Ghaffari 和 Goran Zuzic。普适最优的分布式精确最小割。发表于*2022* 年 *ACM* 分布式计算原理研讨会论文集。ACM，2022 年 7 月。[24] Robert Gmyr、Kristian Hinnenthal、Christian Scheideler 和 Christian Sohler。网络属性的分布式监控：混合网络的力量。发表于第*44*届国际自动机、语言与编程研讨会（*ICALP 2017*），2017 年。[25] Bernhard Haeupler、Taisuke Izumi 和 Goran Zuzic。有界参数图上的近最优低拥塞捷径。发表于分布式计算：第*30*届国际研讨会，*DISC 2016*，法国巴黎，*2016*年*9*月*27–29*日。论文集，第 158–172 页，2016 年。[26] Bernhard Haeupler、Taisuke Izumi 和 Goran Zuzic。无需嵌入的低拥塞捷径。分布式计算，34(1):79–90，2021 年 2 月。[27] Bernhard Haeupler 和 Jason Li。通过捷径实现更快的分布式最短路径近似。发表于第*32*届国际分布式计算研讨会（*DISC 2018*），2018 年。[28] Bernhard Haeupler、Jason Li 和 Goran Zuzic。不含小图的网络族支持快速分布式算法。发表于*2018* 年 *ACM* 分布式计算原理研讨会论文集，第 465–474 页，2018 年。[29] Bernhard Haeupler、Harald Räcke 和 Mohsen Ghaffari。跳数约束的扩展子分解、无视路由与分布式普适最优性。发表于第*54*届 *ACM SIGACT* 计算理论年度研讨会论文集，2022 年。[30] Bernhard Haeupler、David Wajc 和 Goran Zuzic。适用于已知拓扑的普适最优分布式算法。发表于第*53*届 *ACM SIGACT* 计算理论年度研讨会论文集，第 1166–1179 页，2021 年。[31] Daniel Halperin、Srikanth Kandula、Jitendra Padhye、Paramvir Bahl 和 David Wetherall。通过多千兆无线链路增强数据中心网络。发表于*ACM SIGCOMM 2011* 会议论文集：*SIGCOMM - SIGCOMM '11*，2011 年。[32] Stephan Holzer 和 Nathan Pinsker。广播拥塞团中的距离与最短路径近似。发表于第*19*届分布式系统原理国际会议（*OPODIS 2015*），2015 年。[33] Tomasz Jurdziński 和 Krzysztof Nowicki。广播拥塞团中的连通性与最小割近似。发表于结构信息与通信复杂性：第*25*届国际研讨会，*SIROCCO 2018*，2018 年。[34] Udit Narayana Kar 和 Debarshi Kumar Sanyal。蜂窝网络中设备到设备通信概述。*ICT Express*，2018 年 12 月。

[35] Naoki Kitamura、Hirotaka Kitagawa、Yota Otachi 和 Taisuke Izumi。低拥塞捷径与图参数。分布式计算，2021年8月。[36] Ioannis Koutis 和 Shen Chen Xu。用于谱图稀疏化的简单并行与分布式算法。*ACM 并行计算汇刊 (TOPC)*，3(2):1–14，2016。[37] Fabian Kuhn、Yannic Maus 和 Simon Weidner。线图的确定性分布式支配集。载于结构信息与通信复杂性：第*25*届国际研讨会，*SIROCCO 2018*，2018。[38] Fabian Kuhn 和 Philipp Schneider。混合网络模型中的最短路径与直径计算。载于第*39*届分布式计算原理研讨会论文集，第109–118页，2020。[39] Fabian Kuhn 和 Philipp Schneider。混合模型中的路由方案与距离预言机。载于第*36*届分布式计算国际研讨会，2022。[40]Nathan Linial。分布式图算法中的局部性。*SIAM 计算杂志*，21(1):193–201，1992。[41] Pedro Montealegre 和 Ioan Todinca。简短公告：广播拥塞团中的确定性图连通性。载于*2016*年 *ACM* 分布式计算原理研讨会论文集，第245–247页，2016。[42] Michael Rossberg 和 Guenter Schaefer。虚拟专用网络自动配置综述。计算机网络，55(8):1684–1699，2011。[43] Václav Rozhoň 和 Mohsen Ghaffari。多对数时间确定性网络分解与分布式去随机化。载于第*52*届 *ACM SIGACT* 计算理论年度研讨会论文集，第350–363页，2020。[44] Václav Rozhoň、Christoph Grunau、Bernhard Haeupler、Goran Zuzic 和 Jason Li。通过次图聚合求无向 $(1+\epsilon)$ 最短路径：近最优的确定性并行与分布式算法。载于第*54*届 *ACM SIGACT* 计算理论年度研讨会论文集，第478–487页，2022。[45] Philipp Schneider。混合通信网络的能力与局限，2023。网址：https://freidok.uni-freiburg.de/data/232804。[46] Daniel A Spielman 和 Shang-Hua Teng。用于图划分、图稀疏化和求解线性方程组的近线性时间算法。载于第*36*届 *ACM* 计算理论年度研讨会论文集，第81–90页，2004。[47] Anis Tell、Wale Babalola、George Kalaba Kalebaila 和 Krishna C. Chinta。软件定义广域网：助力企业数字化转型的现代混合广域网。载于*IDC 白皮书*，2018。[48] JD Ullman 和 M Yannakakis。高概率并行传递闭包算法。*SIAM 计算杂志*（工业与应用数学学会），20(1)，1991。[49] 维基百科。组合（组合数学）——维基百科，自由的百科全书。http://en.wikipedia.org/w/index.php?title=Composition%20(combinatorics)&oldid=1132979545，2023。 [在线；访问日期：2023]年4月2日。

[50] Goran Žužic。《迈向分布式优化中的普适最优性》。博士论文，名古屋工业大学，2018 年。
[51]Goran Zuzic、Gramoz Goranci、Mingquan Ye、Bernhard Haeupler 和 Xiaorui Sun。通过基于图的 l1 无感路由实现普适最优的分布式最短路径与转运问题。载于*2022 年 ACM-SIAM 离散算法研讨会论文集（SODA）*，第 2549–2579 页。工业与应用数学学会，2022 年。

A Hybrid 中的初步算法

A.1 确定性虚拟树构造

我们展示了如何将 [24] 的确定性覆盖网络构造方法适配到 Hybrid$_0$。这使我们能够在多对数轮内，以确定性方式构造一棵最大度为常数、深度为多对数的虚拟树。

以下定理由 [24] 推出。

定理 A.1 (定理 2 的 [24], 改述)。给定一个具有 n 个节点且最大度为多对数级的连通图 G，存在一种算法，可以构造一棵常数度的虚拟树，其深度为 $O(\log n)$，包含 G 的所有节点，并以标识符最大的节点为根节点。该算法在 Hybrid$_0$ 中需要 $O(\log^2 n)$ 轮。

我们将说明如何去除图 G 最初具有多对数级最大度这一限制，从而证明以下结论。

引理 2.1。在 Hybrid$_0$，中，可以构造一棵覆盖 G 的虚拟树 T，其最大度为常数，且深度为 $O(1)$。可以保证，到算法结束时，树中任意两个相邻节点都知道对方在 G 中的标识符。这需要 $O(1)$ 轮。

证明。每个节点 $v \in V$ 一开始都将其标识符最高的邻居记为 $m(v)$。如果 v 的标识符高于其所有邻居的标识符，则将 $m(v) = v$。节点 v 向 $m(v)$发送一条消息，表示它希望加入该节点。

考虑有向图 $G_1 = (V, E_1)$，其中每个 v 到 $m(v)$都有一条边，当且仅当 $v \neq m(v)$。该图中的每个节点的出度至多为 1，但入度无界。设 $u \in V$ 是 G_1 中入度大于 1 的某个节点。对于某个 $x > 1$，将指向 u 的节点记为 v_1、.....、v_x，这些边都在 G中。对每个 v_i，节点 u 向 v_i 发送一条消息，以调整其边，使其不再指向 u，而改为指向 v_{i+1}。对于节点 v_x，节点 u 不发送消息（也就是说，该边仍指向 u）。将这个新图记为 G'_1。

注意，在 G'_1 中，每个节点的出度至多为 1，入度至多为 2。首先，观察到在 G_1 中，每个节点的出度至多为 1，而这条边在 G'_1 中可能只被一条边替换，因此每个节点的出度仍至多为 1。至于入度，任意节点 $u \in G'_1$ 至多有两条边指向它。将 v 记为 u 在 G_1 中指向的节点。在 G'_1 中，指向 u 的边至多只有两条：一条是在 G_1 中指向它的边，另一条是 v 告知某个其他节点 w 创建、并指向 u 的边。

现在，去掉 G'_1 中边的方向，得到一个最大度为常数的无向图。执行定理 A.1，使得 G'_1 中的每个连通分量都有一棵深度为多对数级、最大度为常数的树。这需要 $O(1)$ 轮。

观察 G'_1 中的连通分量。现在，每个连通分量都有一棵覆盖该分量的虚拟树，并且每个节点都知道其在虚拟树中包含的邻居的标识符~。在每个连通分量中，节点利用该树在 $O(1)$ 轮次内计算出该分量中标识符最大的节点。现在我们按如下方式创建一个新图 $G_2 = (V_1, E_2)$。G_2 中的每个节点对应于 G'_1 中的一个连通分量，并具有该分量中标识符最大的节点的标识符。对于 G_2 中的任意节点 $v \in V_1$，用 $c(v)$ 表示其在 G'_1 中对应的连通分量。$c(v)$ 中的节点计算具有最大标识符的节点 $u \in V_1$，使得存在节点 $x \in c(v)$ $y \in c(u)$，其中 x，y 是 G 中的邻居（如果 v 的标识符大于与其相邻的所有簇的标识符，则设为集合 $u = v$）。为计算 u，G 中的每个节点都将其现在所属簇的标识符告知其邻居。现在，

$c(v)$ 中的所有节点都知道其邻居所属簇的标识符，因此节点 $c(v)$ 可以通过在连接节点 $c(v)$ 的虚拟树上执行一次聚合来计算 u 。

至此，我们已经完成了 G_2 的构造。显然， G_2 的节点数至多是 G_1 的一半，因为在聚类阶段， G_1 中的每个节点都至少与另一个节点合并。因此，我们可以对 O 重复上述过程 n 的对数次迭代，每次都使用覆盖集群节点的虚拟树来模拟这些节点，并在合并簇时合并相应的树。每当我们创建一个新簇时，它都有一棵最大度和深度至多为多对数级的树，因此我们可以在簇内重新运行定理 A.1，以确保其最大度为常数，且深度至多为 $O(n)$。

□

A.2 T_k 的基本性质

我们给出第 3.1 节中各项陈述的证明。

引理 3.1。 可以在 Hybrid0 中用 $O(T_k)$ 轮次计算出 T_k ，并使其成为全局已知信息。

证明。 根据定义 2，如果一个节点 v 知道值 k 和 $B_{T_k}(v)$，就可以计算出 $T_k(v)$。我们利用这一事实逐步探索更大的邻域，并在 v 知道 $T_k(v)$ 时停止。首先，利用引理 2.2，所有节点都将使用求和函数汇总各自持有的令牌数量。此过程耗时 $O(1)$ 轮，并使 k 成为全局已知信息。

接下来，我们进行 T_k 次迭代。每次迭代开始时， v 利用本地网络将其邻域扩展一跳——也就是说，在第 t次迭代中，每个节点 v 都知道 $B_t(v)$。利用这些信息，并且由于 k 是全局已知的， v 可以确定 $T_k(v) = t$。然后，我们使用一次聚合，在 $O(1)$ 轮内计算图中有多少个节点满足 $T_k(u) \leq t$。如果我们知道所有 n 节点都满足 $T_k(u) \leq t$，那么我们就停止，并设置.....，进而可以计算 $T_k = t$。 □

引理 3.3. 对于 $\alpha \geq 1$， $T_{\alpha k} \leq 6\sqrt{\alpha} \cdot T_k$。

证明。 如果 $6\sqrt{\alpha} \cdot T_k \geq D$，那么根据定义， $T_{\alpha k} \leq D \leq 6\sqrt{\alpha} \cdot T_k$。因此，假设 $6\sqrt{\alpha} \cdot T_k < D$。令 v 为某个节点。必定存在 u ，使得 v 与 u 之间的跳距离至少为$D/2 > 3\sqrt{\alpha} \cdot T_k$；否则，图的直径小于 D。记 $P = (v_1 = v, \ldots\ldots, v_\ell = u)$ 为 v与 u之间长度为 $\ell > 3\sqrt{\alpha} \cdot T_k$ 的一条最短路径。

考察路径上任意两个相距 $3T_k$ 条边的节点，即节点 v_i、 v_{i+3T_k}。有 $B_{T_k}(v_i) \cap B_{T_k}(v_{i+3T_k}) = \emptyset$。为导出矛盾，假设 $B_{T_k}(v_i) \cap B_{T_k}(v_{i+3T_k}) \neq \emptyset$，并取 $w \in B_{T_k}(v_i) \cap B_{T_k}(v_{i+3T_k})$。有 $d(v_i, w) \leq T_k$，且 $d(w, v_{i+3T_k}) \leq T_k$，因此 $d(v_i, v_{i+3T_k}) \leq 2T_k$。然而，由于 P 是最短路径，因此 $d(v_i, v_{i+3T_k}) = 3T_k > 2T_k$；最后一个不等式成立是因为 $T_k \geq 1$，由此得到矛盾。

因此，我们得到

$$
\begin{aligned}
|B_{3\sqrt{\alpha} \cdot T_k}(v)| &\geq \sum_{i \in [\sqrt{\alpha}]} |B_{T_k}(v_{1+i \cdot 3T_k})| \\
&\geq \sqrt{\alpha} \cdot \frac{k}{T_k} = \frac{\alpha k}{\sqrt{\alpha} T_k} \geq \frac{\alpha k}{3\sqrt{\alpha} T_k}.
\end{aligned}
$$

因此，根据定义，$T_{\alpha k} \leq 3\sqrt{\alpha} \cdot T_k$，证毕。 □

[{"id": 0, "output": "引理 1.5。如果 $T_k \neq D$，则 $\sqrt{\frac{D}{3n}k} \leq T_k \leq$ 最小值 $\{D, \sqrt{k}\}$。"}]

证明。 对于第一个不等式，我们注意到，对所有 $v \in V$，都有 $|B_{T_k}(v)| \geq k/T_k$，因为 $T_k \neq D$，因此 $|B_{T_k}^{n}(v)| \leq \frac{nT_k}{k}$。因此，半径为 T_k 的互不相交的球最多有 $\binom{nT_k}{k}$ 个。给定一个

最短路径 $P_{u,\ v}$，它在每个这样的球中最多经过 $2T_k+1$ 个节点。否则，这个球的直径就会更大，因为第一个和最后一个 $s,\ t \in B_{T_k}(v)$ 之间的路径也是一条最短路径，而该球的直径为 $2T_k$。因此，由于 $T_k \geq 1$ 恒成立，我们得到$D \leq \binom{nT_k}{k} \cdot (2T_k+1) \leq 3^{\binom{nT_k^2}{k}}$，进而得到 $T_k \geq \sqrt{\frac{D}{3n}}k$。

对于第二个不等式，根据定义 2，$T_k \leq D$ is 成立。令 $v \in V$。如果 $\sqrt{k} \leq n$，那么$|B_{\sqrt{k}}(v)| \geq \sqrt{k} = k/\sqrt{k}$，因为我们假设 G 连通，并且每多探索一跳，至少会扩展一个节点。因此，根据定义 2，$T_k(v) \leq \sqrt{k}$。由此可得，对于所有 $v \in V$，$T_k(v) \leq \sqrt{k}$，从而 $T_k \leq \sqrt{k}$。如果 $\sqrt{k} > n$，那么 $B_{\sqrt{k}}(v) = V$，因此 $D \leq \sqrt{k}$。因此，由于 $T_k \leq D$，有 $T_k \leq \sqrt{k}$。

□