Ara — 全球去中心化基础架构
用于数字文件和内容的付款、用户身份、托管、
流媒体播放和所有权 (草案)

Eric Jiang, Charles Kelly, Joseph Werle, Tony Mugavero, Vanessa Kincaid

最后更新 2018 年 11 月 14 日 (部分 **)

摘要

现今，互联网的情况已大不如从前，成为几家主导公司支配下的笼中囚。这些大公司能够控制信息如何在全世界流动、信息的价格、消费信息的方式和时间，从而对一切信息施行全权掌控。*Ara* 是一个去中心化平台与协议组，旨在解决这一问题。通过借助新型所有权证明系统以授予数字文件与内容的许可并销售此类数字文件与内容，*Ara* 对全球数据传输进行处理，并支持使用本地的 *Ara* 代币购买这些资产。在这一过程中，*Ara* 平台也利用普遍存在的且经分配的用户 *ID* 与钱包系统，以允许用户保留其个人信息的所有权。*Ara* 可谓一种新的现代心智模式，其中涉及如何托管与传输互联网上的信息以及消费者如何使用信息并支付费用；它针对企业和消费者提供了一种全新的模式，使得身处这一系统下的企业与消费者能够通过托管与加入网络来获得奖励。点对点 *(P2P)* 文件共享、所有权与许可证发放的区块链技术以及分布式计算均在单个高效的去中心化系统内融合。

参与其中的许多受众均能受益于 *Ara*。消费者能够使用其自身的闲置存储、带宽以及处理能力来获取 *Ara* 代币（可以类比成计算机世界中的爱彼迎服务），并可使用这些代币以购买内容。企业通过 *P2P* 技术节省了向人们传输信息的费用，从而降低了消费者与其他企业的费用。任何人均能够作为数据中心加入网络并获取奖励。数字内容创作者、游戏与软件开发者、电影与电视工作室与发行商使用 *Ara* 代币在网络内发布已获得许可的内容，从而通过更多的收入分成在其作品上获取更多奖励，并向其粉丝提供剩余的收入分成作为托管费用。这将是一个三赢的局面。消费者获得回报、发行商赚取更多利润，企业也提高了收益。所有这些均采用去中心化以及网络中立的方式，使得中介公司无法再妨害向您传输信息与内容的任何企业。

** 这是自 2018 年 6 月起对白皮书作出的部分更新。在接下来的数月内将发布更为全面的更新版本。

注： Ara 目前正在积极研究与开发中。本文内容随时可能发生变动。请访问 https://ara.one 查看最新版本。若有任何意见与建议，请发送至 hello@ara.one。

1. 简介

1.1 背景

当今，超媒体景观已经过时。各种资讯汇集公司与应用商店正在加强其对内容创作者的控制；传统的内容传输网络效率不高且费用高昂；云计算被集中在一起，筛选出一些看守者；数据并非由其所有人进行存储，而是由从这些数据中获益的人进行存储。内容发布者与创作者被迫提高价格，从而把这个效率不高且昂贵的系统的费用转嫁给消费者，也同样使得发行商、消费者与创作者遭受了价值损失。

随着视频将在 2021[3] 年占据 80% 的网络流量，内容的成本将继续上涨，原因在于文件大小与传输该内容的费用不断上涨。4K、VR 与 AAA 级大作游戏均促成了这一趋势的形成。消费者不仅仅要就交易性内容与订阅内容支付更多费用，而且必须要应付复杂且被滥用的广告系统才能查看免费内容。这些因素加剧了盗版问题，导致内容所有人损失了数十亿美元 [16][12][4]。且人们还因此采用广告拦截器等工具跳过或删除广告。由于消费者采取措施以避免广告的出现以及内容所有人试图重新获取所损失的收入，这反而催生了反广告拦截器的技术以及价格更为高昂的订阅服务，变成了一个恶性循环。

为了抗击这种低效现象，点对点 (P2P) 文件分发架构开始涌现。这一架构从融合了如 Napster 这种集中式服务器和 Gnutella 甚至 BitTorrent 等完全去中心化系统的混合式解决方案发展而来。今天，P2P 文件传输的成本效益已经高到了连微软这样的公司也弃用其自家的 Azure 基础架构，转而使用 P2P 文件传输，从而节省 Windows 10 的分发费用。

但是，虽然 P2P 文件共享网络具有成本效益，但是从历史观点上说，在其被用在公共环境内时，就已面临着不担风险取利、盗版、破解与黑市等问题。没有确切的证据可以表明上传内容的个人确实拥有如此权利，也不存在任何方式能验证其所共享的内容是否按照内容所有人所希望的方式传播。存储与共享内容也不会有任何奖励，因此用户也没有动力主动在共享网络中做种。下载者可能在下载完成自己想要的内容后就会离开共享网络，而不会做种供其他下载者下载。为了防止产生这种问题，P2P 架构开始采用物物交换策略、声誉系统与专有货币等激励机制。但是，这些机制自身也存在问题，并会受到 Sybil 与洗白攻击。

1.2 综述

在本白皮书中，我们将介绍 Ara：一个社区主导的去中心化的分布式计算与内容传输平台。Ara 能够利用世界上任何设备的空闲处理能力、存储与带宽容量，使其成为全球超级计算机、数据库与传输网络的一部分。同时，这些设备形成了 Ara 网络，即一个任何人均可加入并从中获益的生态系统。

从根本上来讲，该网络由一群相互关联的消费者、服务请求者、服务提供者与软件开发者构成，所有人都会受到某种激励机制的影响，因而积极地参与到其中。通过 Ara，服务请求者在指尖就能够拥有一个巨大的计算资源储备以及不断扩大的分布式服务库。无论是智能手机、笔记本电脑或是游戏机，若服务提供者已支付了其设备的费用，就可以出租其未使用的资源来赚取回报。赚取收益的唯一要求只是提交小额押金，作为账户的保留金。支付此保留金是赚取和兑换任何奖励的必要步骤，但它随时可退还。软件开发者可利用 Ara 生态系统空前的规模来完成繁重的计算任务，并为加入其中的请求者与提供者构建新颖的分布式服务。同时，消费者通过欣赏其喜欢的节目和音乐也能获得奖励，而且不会影响他们的日常生活。

因此，任何拥有闲置计算资源的人都能够立即扮演服务履行者的角色，帮助分发内容而获得奖励；而任何寻求远程资源的人只需支付比传统云计算服务少得多的费用，就能够得到 Ara 去中心化服务，享受到优异的安全性、文件可用性与传输速度。Ara 摆脱了购买与管理基础结构的负担，各种各样的内容创作者也能因此获益：独立音乐人可以自行出版其最新专辑，而不再需要依赖唱片公司；大型传媒集团也不再需要新闻汇集公司就能直接将内容带给其受众。Ara 依赖其网络成员提供的资源；网络规模越大，就越强大高效。

1.3 平台服务

Ara 平台包含三 (3) 种核心服务与系统：

1. **AraID:** AraID 为 Ara 平台上的所有代理与内容建立安全、去中心化且可验证的全球身份识别信息，使数据的合法所有人能够对其数据进行控制。
2. **去中心化内容传输网络 (DCDN):** DCDN是 Ara 网络的基础，提供安全的点对点分布式文件系统与存储网络 (AFS)，支持内容完整性验证、激励机制、版本管理与去中心化身份验证。
3. **协议组：** Ara 通过安全的协议组进行连接，能够在DCDN、AraID 与以太坊区块链之间进行非信任交互。

2. 平台综述

无冲突文件系统网络 (CFSNet) 是 Ara 的点对点分布式文件系统、AraID 与DCDN的支柱。以默克尔树结构和精确事件可同步总帐协议 (SLEEP [1]) 文件格式为基础，CFSNet能够解决传统文件传输（客户端-服务器与 P2P）中存在的许多问题，还能提供密码保护的内容完整性验证、版本管理与修订历史，从而超越星际文件系统 (IPFS) 等现有技术。该网络由众多相互隔离的、称为CFS的文件系统构成。此外，每个CFS实例都会执行文件系统层次结构标准 (FHS [8]) 的一个子集，从而支持分区，并允许各个目录能够作为独立的 CFS 档案而存在，且可以各自设定其访问级别。在这些分区中，AFS使用`/home` 与`/etc` 分区以分别存储CFS内容与元数据。各个CFS分区可在整个网络内使用在创建时生成的唯一的 `Ed2551932` 字节公开密钥作为其公开身份。CFS的公开密钥仅能提供对文件系统的只读访问权限，只有在持私有密钥的情况下才能更新和发布其中包含的内容。

2.1 AraID

AraID 负责为 Ara 平台上的所有用户与内容创建并解析安全、可验证的去中心化的身份识别信息。AraID 完全遵守 W3C 去中心化标识 (DID spec [15]) 规范，使用DID Descriptor Object说明符对象，或简称DDO来表示用户与内容（参见图 *1*）。DDO是简单的 JSON-LD 文档 [15]，能够为认证与授权以及其他身份属性（包括服务端点和由所有者控制的私用通信频道）定义方法。由于DDO不会存储个人可验证信息 (PII [15])，所以这些服务端点与通信频道能够确定用以获取 PII 的安全手段，并允许实体自主控制其私人数据与在线身份。

2.1.1 去中心化的身份

对于 Ara 平台上的所有用户与内容，AraID 均为以下形式：

- `did:ara:ee93189c629cdaf94 9fd57bac5b005b916936d2a5c6806 40fd1aedc8315730a0`

AraID 采用了通用解析器方法 `method`，由 DID 的第二个字段表示 (上例中的 `ara`)，作为去中心化身份验证基础系统的一部分。[11]。该 `method`，也被称为驱动程序，确定了在 Ara 平台内解析 DID和DDO的方法。与互联网 URI 不同的是，DID不要求使用中央授权系统来进行注册或控制，且形成与DDO相一致的双射，而非 TCP/IP 与DNS的非单射且非满射的关系。

AraID 安全性的核心通过去中心化公钥基础架构DPKI[13] 来加密维持。其中，`Ed25519` 公开密钥同时被用作DID的 `id` 部分 (上例中的 `ee9318...`) 和存放相应DDO的CFS的公开密钥。这些文档包含一个 `publicKey` 属性，可保存用于数字签名、加密和其它密码相关操作的各种密钥。创建身份时，会将所拥有的身份密钥以及相应的以太坊账户的公共密钥填充到这一系列信息中。

对于 AFS AraID，包含相关内容元数据的`/etc` 分区的公共密钥也会保存在其中。由于该密钥存储于AFS DDO中，它可被拥有此AFS DID的任何请求者解析。

创建新身份时，会使用助记符词组创建密钥对。助记符会返回给所有者保管，便于维护私人密钥，让实体无需私人密钥也能轻松验证DID的所有权并还原账户。

身份存档与解析

创建身份时，最初会写入本地，使任何本地解析设置能够在连接网络之前检查缓存。但是，在远程解析身份之前，首先必须对其进行存档。Ara 会运行存档节点，其旨在存储这些身份，以备将来进行解析。

与存档节点类似，Ara 也运行解析器节点，其旨在为所请求的DDO向存档者进行查询。解析器请求会先在本地查找可能存储在磁盘上的解析身份，然后才会在网络中查找对所涉及的 AraID 进行了存档的远程存档者。

```
{
    'ddo': {
        '@context': 'https://w3id.org/did/v1',
        'id': 'did:ara:ee9318...',
        'authentication;: [{
           'type': 'Ed25519SignatureAuthentication2018',
           'publicKey': 'did:ara:ee9318...#owner'
        }],
        'publicKey': [{
           'id': 'did:ara:ee9318...#eth',
           'type': 'Secp256k1VerificationKey2018',
           'owner': 'did:ara:ee9318...,
           'publicKeyBase58': 'H3C2AVvLMv6gmMNam...'
        }],
        'service': {
           'ens': 'https://etherscan.io/enslookup',
        }
        ...
    }
}
```

图例 1: *DDO* 示例

以太坊账户

每个身份在创建时都会附带一个以太坊账户和关联的以太坊钱包，且可使用在身份创建期间所生成的随机助记符找回身份。由于以太坊账户与身份本身是使用该助记符进行创建的，因此用户可仅使用该助记符找回其完整的身份，包括以太坊账户与钱包。

AraID 可支持任何基于公开密钥密码术的账户。因此，它不会关心所支持的加密货币账户的类型，而且可以轻松地与任何类型的加密货币钱包关联。

2.2 去中心化内容传输网络 (DCDN)

DCDN是 Ara 可拓展、去中心化的超媒体与数字资产分发解决方案。说到底，DCDN是一个由 Ara 文件系统 (AFS)、存放内容的CFS实例以及与其相关的元数据构成的网络。

2.2.1 Ara 文件系统 (AFS)

AFS是一种经调整以满足 Ara 的特殊需求与目标的CFS。AFS利用当前CFS实现的两个分区，即`/home` 与`/etc` 分区。这些分区作为文件系统层次结构标准FHS[8] 的一个子集被实现，分别负责原始二进制数据与内容元数据。`/home` 分区仅可在用户购买了AFS的内容或获得许可访问AFS的内容后进行访问，而对于包含元数据的`/etc`，则可以无视内容所有权随意访问。AFS所有者可定义元数据模式，以供请求者解析。该协议并未对元数据的结构定义严格的标准，但是我们推荐使用Schema.org作为现有范例的参考，从而为去中心化服务的互操作性提供最佳支持。

在最初创建AFS时，会使用 `BIP39` [5] 助记符词组（包含 12 个随机单词）创建一个 AraID。所生成的 DID 被用作 AFS 的公开密钥，并对相应的DDO的 `authentication` 属性进行修改以涵括所有者的DID。这样，通过解析其DID，就能确定AFS的所有者。

每向系统纳入一份内容，就会创建一个AFS。其可为单个文件（如电影），也可以是多个文件的集合（如游戏）。为了通过密码验证内容的所有权，会将两组字节缓冲区写入以太坊区块链内。第一个为 `metadata.tree` 条目，其代表了数据存储层中所含的数据的序列化的默克尔树。第二个为 `metadata.signatures` 文件，包含序列化树根节点的签名。

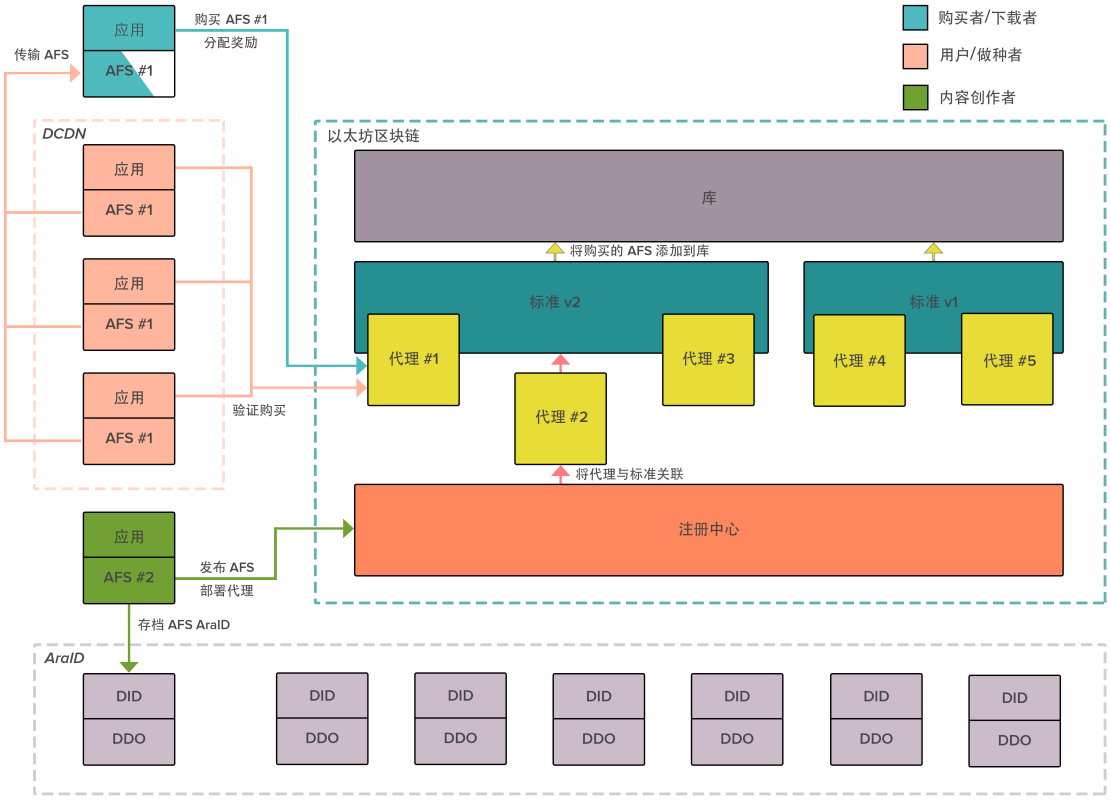
2.2.2 代币使用

最初，拥有关于 DCDN 的 Ara 代币后，就可以：

1. 购买并下载网络内的内容。
2. 提交 Ara 押金作为保留金，以加入 P2P 文件传输并通过内容的传播获取奖励。

2.3 Ara 协议组

以下内容定义了该平台的核心协议，详细描述了该系统的各个部分，并说明了各个部分之间的互操作性。



图例 2: *Ara* 协议示意图

2.3.1 奖励与激励

BitTorrent 等传统点对点文件共享系统过于依赖利他行为，缺乏有效的激励手段 [7] 促使用户上传与之下载量相当的内容，造成了不平衡状态。在这种情况下，吸血虫（只下载不上传文件的用户）能够轻易占据群集（所有下载或上传文件的用户的集合）中的主导地位。这种不平衡是“搭便车问题”[6] 的一种形式，不应再出现在健康的网络中。因为吸血虫往往只消耗他人的资源从中获利，却不提供任何回报作为补偿。Ara 实施奖励系统作为激励机制，向每次下载收取小额费用，将此费用返还给每位为下载内容做种的用户作为奖励，从而缓解网络的低效率情况，并提高文件的可用性。依据付费模式，该费用可整合为内容的总费用，作为奖励分配；对“免费”的内容而言，该费用可替代传统的广告或数据收集（现代网络中用户为“免费”内容付费的方式）。由于每次下载成为了奖励的来源，随着时间的流逝，网络参与者从奖励中获取的金额要比为下载而支付的费用多得多。

2.3.2 文件传输

文件传输是支撑 Ara 的内容传输网络以及参与者获取奖励的主要机制。Ara 的文件传输协议始于一组四步握手机制。

1. 内容请求者 Alice 通过网络发现协议，广

播对某个内容的下载请求（CFS 网络实施了多个策略，包括 mDNS 与 BitTorrent）。
2. 许可验证者与内容传输者 Bob 收到该请求并回复了其文件的可用性。
3. Alice 从响应池（群集）中挑选对等点，使用她的中间公开密钥以及她的 DID 发送消息。
4. Bob 使用密码验证了 Alice 的消息，以及其是否购买了基础 AFS 的许可证。

在握手过程完成后，将开始文件传输。

2.3.3 智能合同

Ara 最初将在以太坊主网上运营，因此智能合同将成为 Ara 协议集的一个重要部分。这些智能合同调整并促进了 DCDN、AraID 与应用层之间的互操作性，确保了该平台上的所有非暂时性财产与实体均在以太坊区块链上进行了注册，包括：

- 发布的内容
- 购买
- 奖励
- Ara 余额

Ara 的智能合同架构的设计考虑了安全性与可修改性。为了对不断演进的观念，即如何出售与购买 AFS、如何处理与分配奖励，以及如何在系统内处理付款与制定付款路线提供支持，Ara 为各个已发布的 AFS 部署 `Proxy Contract`。通过 `Registry Contract` 部署 `Proxies`，其中该等代理与特定版本的 `AFS Standard` 相关，而该 AFS 标准为 AFS 确定了商业逻辑。鉴于为各个 AFS 部署全部的 `AFS Standard` 面临着费用高昂且更新困难的问题 – AFS 相当于被牢牢限定在特定的标准中 – 代理架构允许在 AFS 的使用期内部署单个 `Proxy`，并通过更改其所参考的 `AFS Standard` 版本进行更新。代理架构也能够确保只有经注册的 `Proxy` 地址（即有效的 AFS 内容）才能够添加入 `Library Contract` 内的用户库内。

AFS 标准

`AFS Standard` 使得 AFS 能够以经确定、结构化且独立的形式存在于以太坊区块链上。其包括购买与奖励的方法，以及存储 `metadata` SLEEP 寄存器内的 `tree` 与 `signatures` 文件的方法。`metadata` SLEEP 寄存器将存储 AFS 内的内容相关的元数据，包括文件名、大小与许可，而 `content` SLEEP 寄存器则存储该文件的原始二进制内容。在 `metadata` 寄存器内，`tree` 文件代表了构成 `content` 寄存器内的数据的序列化默克尔树，而 `signatures` 文件内存储了经签名的序列化树根。对于 CFS，这些文件将在磁盘上进行读写，而对于 AFS，它们将在以太坊区块链上进行读写。由于 AFS 通过其自身的 `Proxy` 与该标准进行通信，因此多个不同的 `AFS Standards` 可以共存，使得内容创作者能够选择与其需求最匹配的标准。

`Proxies` 的使用独立于存储系统的逻辑，因此 `AFS Standard` 将成为采用了该版本标准的任何 AFS 的逻辑层，而各个 `Proxy` 则成为单个 AFS 的存储层。至少，`AFS Standards` 必须实现一个 `AFS Standard` 的抽象类，该抽象类会强制实现定价、购买、奖励与存储功能。

在最基本的（默认的）`AFS Standard` 中，价格仅可由 AFS 的 `owner` 进行修改。在购买后，费用将从购买者的 Ara 钱包转入所有者的 Ara 钱包。基础的 `AFS Standard` 会为奖励预算提供支持，而该预算必须在下载前提交。下载完成后，在参与的用户之间分配预算，只要这些用户没有收回参与赢利所必需的保留金，就可以兑换其奖励。

经内容创作者自行决定，`AFS Standards` 也支持众多的个性化定制的商业管制措施：

1. **版税**：可对购买行为进行自定义，以根据所占比重在多个 Ara 账户之间分配收入。
2. **批量购买**：可根据所采购的数量制定各级价格。
3. **转售条件**：所购买的内容可以内容创作者规定的最低转售价格多次转售。
4. **所有权转让**：AFS 的所有人可向另一以太坊地址转让所有权。
5. **预订**：若要下载内容，必须先行购买。购买者可提前提交奖励预算，以此他们能够在 AFS 开放下载后尽快开始下载。
6. **短缺**：内容创作者可为 AFS 确定最大销售量。若超过这一销售量，AFS 将从销售列表中 `unlisted` 且不可购买。

这些商业管制措施向各种各样的内容创作者提供了确定与其需求相匹配的业务与收入模式的权力。可以整合这些管制措施，形成难以通过

传统手段实施的、极具吸引力的新模式。例如，某人喜欢对音乐进行重新混音，则可以从原创者处购买曲目（曲目的转售条件与最低转售价格均已确定）、重新混音、出售新版本，并仍旧能向原创者支付费用。在过去，整合这些管制措施会消耗大量的时间和资金，并涉及法律事务，因而成为了小内容创作者加入的重大阻碍。而今，这些管制措施可向每个人免费提供。

注册中心

作为代理架构的一部分，`Registry Contract` 有两个主要功能：

1. 用作 `Proxy` 工厂
2. 追踪所有 `AFS Standard` 版本

若第一次发布AFS，`Registry` 会为该AFS部署 `Proxy`，并在其与指定的 `AFS Standard` 之间建立关系。如果某 `Proxy` 被调用并将调用委托给其自身的 `AFS Standard` 地址，该 `Proxy` 会就该地址咨询 `Registry`（调用会在该地址处处理并返回给 `Proxy`）。

库

Ara 网络利用 `Library Contract` 创建用户可访问（不论是经购买或其他）的AFS权威事实来源。当购买内容时，`AFS Standard` 内的 `purchase` 功能将自动将AFS DID添加到购买者库内的 `Library Contract` 中。这使得要求提供用户库信息的任何服务能够就该信息查询合同。存储在区块链内的AFS DID允许任何服务解析出内含的内容。`Library` 强制规定只有经注册的 `Proxies` 才可将其相应的 AFS 添加入用户库内，确保任何人均无法在未获得同意的情况下篡改另一用户库。

3. 未来的发展

3.1 模块

Ara 平台的基本原理决定了它并不关心能在其上运行的分布式服务的具体类型。模块是指分布式和/或去中心化服务，这些服务可实现模块API，并在平台内互换使用。ERC-20 代币标准允许以太坊上的任何代币被其他应用再次使用的原理类似 [14]，模块 API 也允许实现该接口的所有分布式服务在整个平台内使用。这促进了致力于建立分布式服务生态系统的开发者社区的形成。社区的参与者们能够利用 Ara 的奖励、购买与支付系统，在本质上形成一个具有高回报性的分布式服务经济。试图利用奖励系统的模块预计会实现额外的智能合同 API，并在此 API 中定义自己的奖励机制与方法论。各个模块智能合同将存储其通过使用分布式服务而累积的奖励分配并对其作出相应分配。

3.2 Ara 名称系统 (ANS)

ANS，与域名系统 (DNS [9]) 十分相似，是在 Ara 网络内注册、查询、调用或撤销证书的一种去中心化方式。DNS属于互联网的应用层，作用是将对人具有可读性的 URI 解析为里层的 IP 地址，使得网页浏览器等用户代理能够获得并呈现所要求的内容。类似地，ANS将对人具有可读性的名称解析为DID，并最终解析为DDO。ANS在后台使用身份存档器与解析器进行第二阶段的解析，从DID中提取出DDO。可以将ANS视为针对具备可读性的 UR 的存档器和解析器。如果安装了基于 Ara 构建的网络浏览器，ANS就可以通过主机名解析出DDO。这是它的应用示例之一。

为了辨别各种各样的记录类型，各个记录内会保存一个 `TYPE` 资源记录，类似DNS对其自身的记录作出的分类 [17]。`TYPE` 字段采用数值进行表示，允许在未来将其他类型的记录存储于ANS内。下表对记录的 `TYPE` 字段进行了说明：

TYPE	Value	Description
USR	00	User
PCT	01	Published Content

每个由ANS构成的超节点中心运行一个 HyperDB 实例 [2]。类似 HyperDB 的分布式且可高度扩展的数据库具备多种特性，使其能与类似ANS的系统相匹配。首先是 HyperDB 的 tries 使用：搜索树。其中各个节点为其子节点的前缀。通过利用 tries 存储姓名，我们能够保证即使数据库里有成千上万条条目，查找也很经济而迅速。tries 内的查找为 `O(n)`，其中 `n` 为正在搜索的关键词的长度。HyperDB 也使用矢量时钟，其在分布式系统内追踪事件的因果关系，以防止节点不同步的情况 [10]。

4. 鸣谢

本白皮书承蒙 Littlstar与Token Foundry之惠而得以成文。特别感谢 Logan Dwight、Andrew Grathwohl 与 Brandon Plaster 作出的贡献。

缩写

AFS Ara File System. 3–5, 8

ANS Ara Name System. 9

CFS Conflict-Free File System. 4, 5

CFSNet Conflict-Free File System Network. 4

DCDN Decentralized Content Delivery Network. 3–5

DDO DID Descriptor Object. 4, 5, 9

DID Decentralized Identifier. 4, 5, 8, 9

DNS Domain Name System. 4, 9

DPKI Decentralized Public Key Infrastructure. 4

FHS Filesystem Hierarchy Standard. 4, 5

PII Personally-Identifiable Information. 4

SLEEP Syncable Ledger of Exact Events Protocol. 4

参考文献

[1] Code for Science Buus, Ogden. Sleep - syncable ledger of exact events protocol. https://github.com/datproject/docs/blob/master/papers/sleep.pdf, Aug 2017.

[2] Mathias Buus. Hyperdb. https://github.com/mafintosh/hyperdb, Aug 2017.

[3] Cisco. Cisco visual networking index predicts global annual ip traffic to exceed three zettabytes by 2021. https://newsroom.cisco.com/press-release-content?type=webcontent&articleId=1853168, Jun 2017.

[4] Stewart Clarke. Piracy set to cost streaming players more than $50 billion, study says. http://variety.com/2017/tv/news/piracy-cost-streaming-players-over-50-billion-1202602184/, Oct 2017.

[5] Palatinus et al. Mnemonic code for generating deterministic keys. https://github.com/bitcoin/bips/blob/master/bip-0039.mediawiki, Sept 2013.

[6] Russell Hardin. The free rider problem. https://plato.stanford.edu/entries/free-rider, May 2003.

[7] Ahamad Jun. Incentives in bittorrent induce free riding. https://disco.ethz.ch/courses/ws0506/seminar/papers/freeriding_incentives.pdf, Aug 2005.

[8] The Linux Foundation LSB Workgroup. Filesystem hierarchy standard. https://refspecs.linuxfoundation.org/FHS_3.0/fhs-3.0.pdf, Mar 2015.

[9] Paul Mockapetris. Domain names - implementation and specification. https://tools.ietf.org/html/rfc1035, Nov 1987.

[10] Multiple. Vector clock. https://en.wikipedia.org/wiki/Vector_clock.

[11] Markus Sabadello. A universal resolver for self-sovereign identifiers. https://medium.com/decentralized-identity/a-universal-resolver-for-self-sovereign-identifiers-48e6b4a5cc3c, Nov 2017. Accessed on 2018-04-24.

[12] Stephen E. Siwek. The true cost of sound recording piracy in the us economy. https://www.riaa.com/wp-content/uploads/2015/09/20120515_SoundRecordingPiracy.pdf, Aug 2007.

[13] Rebooting the Web-of Trust. Decentralized public key infrastructure. http://www.weboftrust.info/downloads/dpki.pdf, Dec 2015. Accessed on 2018-04-19.

[14] Buterin Vogelsteller. Erc-20 token standard. https://github.com/ethereum/EIPs/blob/master/EIPS/eip-20.md, Nov 2015.

[15] W3C. Decentralized identifiers. https://w3c-ccg.github.io/did-spec, Apr 2018. Accessed on 2018-05-01.

[16] Music Business Worldwide. Why does the riaa hate torrent sites so much? https://www.musicbusinessworldwide.com/why-does-the-riaa-hate-torrent-sites-so-much/, Dec 2014.

[17] Inc ZyTrax. Dns resource records (rrs). http://www.zytrax.com/books/dns/ch8/, Oct 2015.

附录

I. 成熟的 Ara 平台代币经济（草案）

综述

传统的云服务因其在购买与管理内部架构方面提供的灵活性、机敏性和经济性而一炮走红。众所周知，许多云服务使用复杂且模糊的定价模式，且需要与每个客户单独协商长期的投入与合同，这直接损害了其最初打算提供的灵活性与机敏性 [1]。近几年，P2P CDN 越来越受欢迎，且在视频传输方面，也出现了 SaaS 所鼓吹的混合解决方案。尽管 SaaS 利用视频观众群体建立了具备更简单的定价模式的、可高度扩展的解决方案，但其仍旧维护着多个集中化资源，即需要受制于其商业模式，且将用户归类为“二等公民”，因此必须经过其同意或实行经济补偿才能使用其硬件设施。而 Ara 对使用其设备的网络参与者提供奖励，因而又前进了一步。

为了取代当前的标准云基础架构费用，Ara 平台发行了本地协议等效代币：Ara 代币。此代币可在 Ara 平台上使用，用以构建加密的经济激励措施，鼓励健康诚实的网络行为，从而使得内容消费者与创作者之间建立更加直接的雇佣关系，并鼓励用户使用该平台。Ara 代币可视为成员们在网络上提供的价值的度量，所奖励的每个代币都代表了网络效用的提高。

Ara 代币

使用 Ara 的 SDK 所创建的分布式服务被称为模块，可在 Ara 网络上购买、出售、请求与兑现。可在网络内向对等用户外包模块任务，而对方在成功完成任务后，可获得 Ara 代币作为报酬。为了推动建立一个开放且具有竞争性的市场，Ara 允许服务请求者为其所请求的的服务确定奖励分配或奖金，并允许服务提供商确定最低可接受的奖金金额。与“亚马逊土耳其机器人”（需要人类智能支持的任务的众包市场）类似，Ara 为外包的分布式计算或网络连接任务建立了市场。模块可为一次性按需任务，例如分布式转码，也可为长期反复提供的服务，例如 P2P 多人游戏服务器。

功能

Ara 代币可通过多种方式在整个网络内使用。

- 就消费者而言，Ara 代币可用以购买各种各样的内容，从具有娱乐性质的数字内容到可参与的新模块等等
- 服务请求者可使用 Ara 代币以发起工作请求并为成功完成这些工作者设置奖金
- 服务提供者可支付 Ara 代币押金，作为对完成任务以获得奖励的承诺（经要求可退还押金，但是提供者无法再赚取奖励）
- 开发者可使用 Ara 代币在网络内部署新的模块

由于这些角色存在严重重叠的情况，所以服务提供者可直接使用完成任务而赚取的 Ara 代币奖金来购买新的内容，开发者也能够使用因出售模块而赚取的 Ara 代币发起新的工作请求。

市场动态

由于网络成员在决定他们要参与的任务或服务方面具有完全的自主性，因此该网络内会形成一个经济均衡的自由市场。各个模块可能拥有自己的经济，此经济受请求者与提供者对该模块的工作作出的行为管控。例如，分布式视频转码对视频制作人来说可能是一个特别紧急的工作，使得分布式视频转码的需求价格缺乏弹性（即视频制作人相对而言不太关心他们需要向服务提供者提供多少奖励）。这因此形成了卖方市场，使得分布式转码服务提供者在确定奖励分配方面占有优势，促使奖金金额上涨。同样地，P2P 游戏服务器模块的需求较多，但提供该模块的服务提供者相对较少。这又形成了卖方市场，并且奖金也被抬高。另一方面，可能只有少数人需求分布式机器学习模块，但是却有很多符合条件的服务提供者能够提供这一模块。由于需求相对较少，则若提供者将其最低奖金要求设得过高，则将错失机会。此时，这形成了买方市场，并且奖金被压低。

值得注意的是，没有强制要求模块设置奖金，也如何设置奖金也并不存在任何标准化模式。其旨在协调服务请求者与服务提供者的激励措施、支持各种激励模式，并根据全球各地的基础架构与联网固定费用作出相应调整。

激励结构

为了更好地了解该模式如何实现服务请求者与服务提供者之间的激励匹配，我们将从整体上对双方的经济利益进行概述。服务请求者想要其提出的请求能够以最低的成本得以完成，然而服务提供者却想要以最高的价格提供服务。换句话说，在双方对价格取得一致意见的情况下将网络效用最大化，会为双方带来最佳利益。因此，最有可能完成的是在奖金与工作之间达到最佳平衡的服务，这为市场带来了压力，并提高了分布式服务设计的奖金竞争力与分布式服务设计的创新性，最终提高了效率。

可在该平台上运行的分布式服务的多样性使得在确定奖励的工作单位（UWR，可证明的工作的基础单位，根据其划分并提供奖金）与奖金模式（管控如何支付奖金的条件）时必须灵活。P2P 多人游戏服务器可使用已服务的请求数作为其 UWR，而调用该服务器模块的游戏开发者可能会认为基于订阅的、重复式奖金模式是最为合理的。另一方面，分布式转码服务可使用每分钟转码的字节数作为其 UWR，视频制作人可就每次转码支付一次奖金。

服务提供者可提供 Ara 代币作为押金来参与工作并赚取奖励。如同支付奖金一样，服务请求者可决定提供者为了加入该服务而必须缴付的最低押金（若有）。最低押金暗示了服务要求的投入水平，且在服务成功完成后与奖金一起退还。作为确定 *UWR* 的一部分，服务也必须提供证明以核实其是否完成。

相反，服务提供者也可根据其为服务而专门投入的资源来设置订阅费用。这些专门的提供者被称为 *supernodes*，他们缴付的押金将保管在一份智能合同内，直至订阅结束。超节点常常比非专门的提供者更加可靠，因此其能够通过设置订阅费用的方式来左右市场动态。波哥大的超节点的收费可能比洛杉矶的超节点高，原因在于其硬件与互联网费用更高。

网络效应

向网络添加新内容涉及 DCDN 超节点的调用——专注于内容冗余与可用性的 Ara 节点。假设某个文件的奖金不变，而某个用户共享了该文件，促使此文件的可用性有了微小提升，这对 DCDN（以及一切具有固定激励措施的基于货币的 P2P 文件共享系统 [2]）中一切用户可能从此文件上获得的奖励产生的影响，是可以通过一个子模块集函数来模拟的。更直观地讲，可以把此函数想象为描述收益递减的函数。由于这一特性，DCDN 超节点可采用订阅奖金模式来抵消随着可用性的提高而不断增加的托管内容机会成本。

内容发布者可以根据具体的内容分别在其地理位置内任意调整调用与订阅的超节点的数量。然后，发布者可自由决定在全世界范围内提供其内容的程度。无论是想要在全世界范围内调用所有可用的超节点以涵盖全球受众的大型传媒，还是确定主要受众为欧洲并决定优先考虑欧洲超节点的独立内容创作者，这都能够提供支持。内容发布者也可为每次的内容下载决定奖励分配。因此，所有人都能找到最理想的奖励分配与超节点分配，达到理想的参与程度（即文件可用性）。

参考文献

[1] Enterprise Strategy Group (2015, June), *Price Comparison: Google Cloud Platform vs. Amazon Web Services*, https://cloud.google.com/files/esg-whitepaper.pdf

[2] M. Salek, S. Shayandeh, and D. Kempe, *You Share, I Share: Network Effects and Economic Incentives in P2P File-Sharing Systems* https://arxiv.org/pdf/1107.5559.pdf

II. DCDN 成本分析 by Lester Kim

简介

为了计算潜在合作伙伴的流媒体成本，我们需要知晓其在每个时间单位 T（秒）内需要向消费者传输的数据量 B（字节）。假设有 N 组的上传者节点，其中 $N \in \mathbb{N}$。$\forall n \in \{1, ..., N\}$，每组的平均带宽 n 为 b_n（字节/秒/节点）。设 q_n 为第 n 组节点的数量。设 $\vec{b} = [b_1 ... b_N]^\top$，且 $\vec{q} = [q_1 ... q_N]^\top$。因此，在 $\frac{B}{T}$ 限制下每秒传输的字节为

$$g(\vec{q}) = \vec{b} \cdot \vec{q} = \frac{B}{T}. \tag{1}$$

我们想要找到最理想的 $\vec{q}^*$ 以将分发的成本 $C(\vec{q})$ 降至最低。若 $\vec{p} = [p_1 ... p_N]^\top$，其中 p_n 为第 n 组的每节点价格，我们得到

$$C(\vec{q}) = \vec{p} \cdot \vec{q}. \tag{2}$$

上传者的利润最大化

为了确定 $\vec{p}$，我们来观察一下将利润最大化的公司的行为。设 f 为生产函数，其中能源输入为 E（千瓦时），输出为 q（节点数）。我们为该生产函数建模，得到

$$f(E) = AE^\alpha \tag{3}$$

其中 A 为生产要素（节点数/千瓦时 $^\alpha$），$\alpha \in [0, 1]$ 为生产弹性（输出增加百分比/输入增加百分比）[1]。

设 P 千瓦时/秒）为节点开始上传数据时的功率提高值。这包括通过网络接口控制器 (NIC) 发送数据，但也包括机器的开启（从关闭或待机状态开启）。若各个节点的功率为 P，则对于特定的 E，单个节点可运行 $\frac{E}{P}$ 秒。但是，假设为完成工作的时间限制为 T，则必须有 $\frac{E}{PT}$ 个节点。因此可得：

$$A = \frac{D}{(PT)^\alpha} \tag{4}$$

其中 D 为总要素生产力 [2]（节点数）。

设 p 为每节点价格，p_E 为能源价格（每千瓦时）。则该公司的利润函数 π 为

$$\pi(q, E) = pq - p_E E. \tag{5}$$

带宽成本可以忽略不计，因为相对于一个月的时间，它在短短数秒内的成本可以视作是固定不变的[1]。

假定输出要求至少为 q，要将公司的利润最大化，则有：

[1]即使包含了带宽，每秒成本的数量级也与能源的一样。在纽约，50 MBps 的成本为 \$3 x 10^{-5}/秒 [3]。

$$\max_{q,E} \ \pi(q,E) \quad \text{s.t. } f(E) \geq q. \tag{6}$$

为了解出这一函数，设拉格朗日乘数为：

$$\mathcal{L}(q,E,\lambda) = pq - p_E E - \lambda(AE^{\alpha} - q). \tag{7}$$

采用偏导函数，并将其设置为零，得出

$$\frac{\partial \mathcal{L}}{\partial q} = p + \lambda = 0 \tag{8}$$

$$\frac{\partial \mathcal{L}}{\partial E} = -p_E - \lambda A \alpha E^{\alpha-1} = 0 \tag{9}$$

$$\frac{\partial \mathcal{L}}{\partial \lambda} = q - AE^{\alpha} = 0. \tag{10}$$

解出这些一阶条件，得出

$$q^* = \left(\frac{\alpha A^{\frac{1}{\alpha}} p}{p_E}\right)^{\frac{\alpha}{1-\alpha}}. \tag{11}$$

重写这一函数（去掉 q 的星标），得出

$$p = \frac{p_E}{\alpha}\left(\frac{q^{1-\alpha}}{A}\right)^{\frac{1}{\alpha}}. \tag{12}$$

这一公式会使得创作者知晓 p 应获取预期数量的节点。
从（12）中，我们发现对于特定的 q，最理想收入为

$$pq = \frac{p_E}{\alpha}\left(\frac{q}{A}\right)^{\frac{1}{\alpha}}. \tag{13}$$

分发者的成本最小化

由于该公司将收入花在了消费者（创作者）身上，可将（2）写为

$$C(\vec{q}) = \frac{p_E}{\alpha}\sum_{n=1}^{N}\left(\frac{q_n}{A_n}\right)^{\frac{1}{\alpha}}. \tag{14}$$

在将创作者的成本降至最低方面存在的问题为

$$\min_{\vec{q}} \ C(\vec{q}) \quad \text{s.t. } g(\vec{q}) \geq \frac{B}{T}. \tag{15}$$

拉格朗日乘数为

$$\mathcal{L}(\vec{q},\lambda)=C(\vec{q})-\lambda(g(\vec{q})-\frac{B}{T}). \tag{16}$$

一阶条件为

$$\frac{\partial\mathcal{L}}{\partial\vec{q}}=\frac{\partial C}{\partial\vec{q}}-\lambda\frac{\partial g}{\partial\vec{q}}=0 \tag{17}$$

$$\frac{\partial\mathcal{L}}{\partial\lambda}=\frac{B}{T}-g(\vec{q})=0. \tag{18}$$

在（14)、(4）与（1）中，

$$\frac{\partial C}{\partial\vec{q}}=\frac{p_E T\vec{P}}{\alpha^2}\circ(\vec{q}^{\circledast(1-\alpha)}\oslash\vec{D})^{\circ\frac{1}{\alpha}} \tag{19}$$

$$\frac{\partial g}{\partial\vec{q}}=\vec{b} \tag{20}$$

其中 $(\vec{D},\vec{P})=([D_1...D_N]^\top,[P_1...P_N]^\top)$。“$\circ$”、“$^\circ$”、“$\oslash$” 分别为阿达马变换（各元素）的产品、功率与分配 [4]。所以 $\forall m,n\in\{1,...,N\}$，

$$\frac{P_m^\alpha q_m^{1-\alpha}}{b_m^\alpha D_m}=\frac{P_n^\alpha q_n^{1-\alpha}}{b_n^\alpha D_n}. \tag{21}$$

因此，

$$b_m q_m=\left(\frac{b_m D_m P_n^\alpha}{P_m^\alpha b_n D_n}\right)^{\frac{1}{1-\alpha}} b_n q_n. \tag{22}$$

将（22）与（18）结合，得出

$$\boxed{\vec{q}^* = \frac{B\vec{b}^{\circ -1}}{T\kappa} \circ (\vec{b} \circ \vec{D} \oslash \vec{P}^{\circ \alpha})^{\circ \frac{1}{1-\alpha}}} \tag{23}$$

$$\boxed{C^* = \frac{p_E T}{\alpha} \left(\frac{B}{T\kappa^{1-\alpha}} \right)^{\frac{1}{\alpha}}} \tag{24}$$

其中

$$\kappa \equiv \sum_{m=1}^{N} \left(\frac{b_m D_m}{P_m^{\alpha}} \right)^{\frac{1}{1-\alpha}}. \tag{25}$$

特例: $\alpha = 1$

当 $\alpha = 1$ 时，(23) 与 (24) 变成了

$$q_n^* = \begin{cases} \frac{B}{|\Upsilon| T b_n} & n \in \Upsilon \\ 0 & n \notin \Upsilon \end{cases} \tag{26}$$

$$C^* = \frac{p_E B P_n}{b_n D_n} \qquad \text{any } n \in \Upsilon \tag{27}$$

其中

$$\Upsilon \equiv \left\{ n \in \{1, ..., N\} \;\middle|\; n = \arg\max_{1 \le m \le N} \frac{b_m D_m}{P_m} \right\}. \tag{28}$$

$\forall n \in \Upsilon$，在第 n 组内的各个节点将交付 $\frac{B}{|\Upsilon| q_n^*} (= b_n T)$ 数据并接收至少 $\frac{p_E P_n T}{D_n}$ 的报酬。但是，$\vec{q}^*$ 存在多重解。例如，对于任何 $n \in \Upsilon$，第 n 组可通过采用 $\frac{B}{T b_n}$ 节点来完成所有工作。

实例

我们对纽约的一个实例进行计算，其中

$$\alpha = 1 \tag{29}$$
$$B = 1 \text{ GB} \tag{30}$$
$$N = 2 \tag{31}$$
$$p_E = \$0.2321/\text{kWh [5]} \tag{32}$$
$$T = 1 \text{ s} \tag{33}$$
$$\vec{b} = \begin{bmatrix} 100 \text{ MB/s} \\ 1 \text{ MB/s} \end{bmatrix} \text{ [6]} \tag{34}$$
$$\vec{D} = \begin{bmatrix} 1 \text{ node} \\ 1 \text{ node} \end{bmatrix} \tag{35}$$
$$\vec{P} = \begin{bmatrix} 200 \text{ W} \\ 2 \text{ W} \end{bmatrix} \text{ [7][8]} \tag{36}$$

为创作者找到 $\vec{q}^*$ 与 C^* 的实例。然后，

$$\vec{q}^* = \begin{bmatrix} 5 \text{ nodes} \\ 500 \text{ nodes} \end{bmatrix} \tag{37}$$
$$C^* \approx \$1.29 \text{ x } 10^{-4}. \tag{38}$$

这比 AWS Cloudfront 的按需定价（$0.020/GB - $0.085/GB）要便宜 155 至 659 倍（99.35% - 99.85%）[9]。第 1 组内的每个节点将负责处理 100 MB，然而第 2 组内的每个节点将负责处理 1 MB。第 1 组与第 2 组内的各个节点分别需要超过 $1.29 x 10^{-5} 与 $1.29 x 10^{-7}。

换个角度说，假定 Netflix 是一个潜在合作伙伴。2017 年，Netflix 平均每天被观看的内容超过 1.4 亿小时 [10]。平均起来，Netflix 视频为 1GB/小时 [11]。在 Ara 平台上，51.1 艾字节 [12] 的年花费仅为 660 万美元/年 ($0.2106/秒)。若估计 Netflix 的流媒体费用为 $0.03/GB [13]，可得到 $1.5 亿/年 ($46.61/秒)。若使用 Ara 网络，将使得 Netflix 在 2017 年的净收入 5.589 亿美元 [14] 提高到接近四倍（曼哈顿拥有 166 万的人口 [15]，其中 287008Netflix 用户[2]的流量为 321.45TB/天，3.72GB/秒。这需要 3721 个第 2 组内的节点，费率为 $41.47/天)。

[2]截止 2018 年第一季度末，Netflix 在美国的订阅人数达到 5671 万，在全世界的订阅人数达到 1.25 亿 [16]。美国总人口 3.28 亿 [17]，按比例计算，则有 287008 位 Netflix 订阅者位于曼哈顿。

参考文献

[1] Wikipedia (2018, April 22), *Cobb–Douglas production function*, https://en.wikipedia.org/wiki/Cobb%E2%80%93Douglas_production_function

[2] Wikipedia (2018, June 5), *Total factor productivity*, https://en.wikipedia.org/wiki/Total_factor_productivity

[3] Spectrum (2017, December 29), *Broadband Label Disclosure*, p.2, https://www.spectrum.com/content/dam/spectrum/residential/en/pdfs/policies/Broadband_Label_Disclosure_Charter_122917.pdf

[4] Wikipedia (2018, March 10), *Hadamard product (matrices)*, https://en.wikipedia.org/wiki/Hadamard_product_(matrices)

[5] Electricity Local (2018, June 19), https://www.electricitylocal.com/states/new-york/new-york

[6] Wikipedia (2018, June 19), *Bandwidth (computing)*, https://en.wikipedia.org/wiki/Bandwidth_(computing)

[7] Energuide.be (2018, June 19), *How much power does a computer use? And how much CO2 does that represent?*, https://www.energuide.be/en/questions-answers/how-much-power-does-a-computer-use-and-how-much-co2-does-that-represent/54/

[8] R. Sohan, A. Rice, A. W. Moore, and K. Mansley, "Characterizing 10 Gbps Network Interface Energy Consumption," *The 35th Annual IEEE Conference on Local Computer Networks (LCN) Short Papers*, University of Cambridge, Computer Laboratory, July 2010, https://www.cl.cam.ac.uk/ acr31/pubs/sohan-10gbpower.pdf.

[9] Amazon Web Services Pricing (2018, June 19), *Amazon Cloudfront Pricing*, https://aws.amazon.com/cloudfront/pricing

[10] L. Matney (2017, December 11), "Netflix users collectively watched 1 billion hours of content per week in 2017," *Techcrunch*, https://techcrunch.com/2017/12/11/netflix-users-collectively-watched-1-billion-hours-of-content-per-week-in-2017

[11] K. Hubby (2017, May 23), "The surprising amount of data Netflix uses," *The Daily Dot*, https://www.dailydot.com/debug/how-much-data-netflix-use/

[12] Wikipedia (2018, June 20), *Exabyte*, https://en.wikipedia.org/wiki/Exabyte

[13] D. Rayburn (2009, July), "Stream This!: Netflix's Streaming Costs," *Streaming Media (June/July 2009)*, http://www.streamingmedia.com/Articles/Editorial/Featured-Articles/Stream-This!-Netflixs-Streaming-Costs-65503.aspx

[14] Netflix (2018, January 1), p.40, https://ir.netflix.com/static-files/20c3228d-bf1f-4956-a169-c8b76911ecd5

[15] Wikipedia (2018, July 5), *Manhattan*, https://en.wikipedia.org/wiki/Manhattan

[16] Statistica (2018, July 5), *Number of Netflix streaming subscribers in the United States from 3rd quarter 2011 to 1st quarter 2018 (in millions)*, https://www.statista.com/statistics/250937/quarterly-number-of-netflix-streaming-subscribers-in-the-us/

[17] United States Census Bureau (2018, July 5), *U.S. and World Population Clock*, https://www.census.gov/popclock/

III. 预期 Ara 奖励分析 by Lester Kim

网络模式

我们将 Ara 节点网络表示为一个完整的图表 [1] $G = (V, E)$，其中 V 包含 N 个顶点，各顶点代表一个节点，E 包含 $\frac{N(N-1)}{2}$ 条边，其中每条边代表了两个节点之间的通信信道。设 C 为所有节点均想要拥有的某内容集（在这里即一组数字娱乐文件），设子集 $S \subseteq V$ 内包含拥有 C 的所有节点 (即 $S = \{v \in V : C \in v\}$)。

设时间 $t \in \mathbb{N}$。若 $t = 0$，$|S| = 1$，所以仅有一个 $v_0 \in V$ 含有内容 C；因此，仅有一个顶点能够在网络内向其他节点传输一份 C。假定所有其他 $N-1$ 节点都想要 C，但 v_0 的带宽只够向一个节点传输 C。从 $t = 0$ 到 $t = 1$，$|S|$ 从 1 增加至 2。推广而言，时间为 t 时，

$$|S| = \begin{cases} 2^t & 0 \le t < \log_2 N \\ N & t \ge \log_2 N. \end{cases} \tag{39}$$

注意，$|S| = N$ 开始于 $t = \lceil \log_2 N \rceil$。

只有当 v 向 s 支付 p 的金额后，$\forall s \in S$ s 才会将 C 传输给 $v \in V \setminus S$。设 M 为网络中娱乐文件传输的总预算。将其除以 N 个节点，从而得出 $p = M/N$。

若 $t = 0$，唯一的 $v_0 \in S$ 会从某些 $v_1 \in V \setminus S$ 处接收 p。然后，若 $t = 1$，每个 v_0 $v_1 \in S$ 分别从某些 $v_2, v_3 \in V \setminus S$ 处接收 p。若 $t < \lfloor \log_2 N \rfloor$，$S$ 内的 $|S| = 2^t$ 个节点各自会从 $V \setminus S$ 内的 2^t 个节点处接收 p。若 $t = \lfloor \log_2 N \rfloor$，$|S| > \frac{N}{2}$，则 C 提供者的数量要比需求者多的多。当发生这种情况时，将从 S 中随机选择 $N - 2^t$ 个节点来传输 C。若 $t = \lceil \log_2 N \rceil$ $S = V$。

在此模式中，v_0 至少赚取

$$\frac{M \lfloor \log_2 N \rfloor}{N}; \tag{40}$$

v_1 至少赚取 $\frac{M(\lfloor \log_2 N \rfloor - 1)}{N}$；$v_k$ 至少赚取

$$\frac{M(\lfloor \log_2 N \rfloor - \lceil \log_2 (k+1) \rceil)}{N}. \tag{41}$$

v_k 至少赚取 $\frac{M}{N}$ 时的最大 k 出现于

$$\lfloor \log_2 N \rfloor - \lceil \log_2 (k+1) \rceil \ge 1 \tag{42}$$

这意味着

$$\log_2 (k+1) \le \log_2 \frac{N}{2}. \tag{43}$$

因此，要赚取至少 $\frac{M}{N}$，k 的最大值为 $k = \lfloor \frac{N}{2} \rfloor - 1$。平均起来，每个节点赚取

$$\frac{M - \frac{M}{N}}{2^{\lceil \log_2 N \rceil - 1}} = \frac{M(1 - \frac{1}{N})}{2^{\lceil \log_2 N \rceil - 1}}. \tag{44}$$

分子为 $M - \frac{M}{N}$，不包括 v_0 的娱乐预算，因为其在 $t = 0$ 时已含有 C。分母为 $2^{\lceil \log_2 N \rceil - 1}$，因为若 $t = \lceil \log_2 N \rceil - 1$，$|S| = 2^{\lceil \log_2 N \rceil - 1}$，且在此时，$S$ 包含在过程中可能赚取奖励的所有节点。这意味着有 $N - 2^{\lceil \log_2 N \rceil - 1}$ 个节点无法赚取奖励。

实例

大约 80% 的美国人拥有可联网的计算机 [2]。由于美国人口为 3.27 亿 [3]，所以有 $(0.8)(327\text{M}) = 261.6\text{M}$ 的美国人拥有可联网的设备。假设每个人只有一台设备，则 $N = 261.6\text{M}$。美国每一年的娱乐消费为 \$734B [4] [5]。假设未来这项开销的大部分为数字娱乐支出，如果仅考虑能够联网的这 80% 的美国人的预算，则他们的花销为 $(0.8)(734\text{B}) = \$587\text{B}$。假设这些花销中的 10% 用于支付分发费用。随后可得，$M = (0.1)(\$587\text{B}) = \$58.7\text{B}$。随后可得，$p = M/N = \$58.7\text{B}/261.6\text{M} = \$224.39$。根据 (44)，每年收益的平均值为每节点 \$437.35。根据 (40)，v_0 能够赚取的最多奖励为 \$6282.87。因此，该实例证明了最初共享内容的对等点将赚取最多奖励。

参考文献

[1] Wikipedia (2018, June 19), *Complete graph*, https://en.wikipedia.org/wiki/Complete_graph

[2] C. Ryan and J. M. Lewis, "Computer and Internet Use in the United States: 2015," *American Community Survey Reports* U.S. Census Bureau, September 2017 https://www.census.gov/content/dam/Census/library/publications/2017/acs/acs-37.pdf

[3] World Population Review (2018, June 18) *United States Population 2018* http://worldpopulationreview.com/countries/united-states-population/

[4] SelectUSA (2018, August 22), *MEDIA AND ENTERTAINMENT SPOTLIGHT*, https://www.selectusa.gov/media-entertainment-industry-united-states

[5] Bureau of Labor Statistics (2017, August 29), *CONSUMER EXPENDITURES–2016* https://www.bls.gov/news.release/cesan.nr0.htm