

为了决定她响应哪个请求, BitTorrent 使用了一种机灵的对换算法。其基本想法是, Alice 根据当前能够以最高速率向她提供数据的邻居, 给出其优先权。特别是, Alice 对于她的每个邻居都持续地测量接收到比特的速率, 并确定以最高速率流入的 4 个邻居。每过 10 秒, 她重新计算该速率并可能修改这 4 个对等方的集合。用 BitTorrent 术语来说, 这 4 个对等方被称为疏通 (unchoked)。重要的是, 每过 30 秒, 她也要随机地选择另外一个邻居并向其发送块。我们将这个被随机选择的对等方称为 Bob。因为 Alice 正在向 Bob 发送数据, 她可能成为 Bob 前 4 位上载者之一, 这样的话 Bob 将开始向 Alice 发送数据。如果 Bob 向 Alice 发送数据的速率足够高, Bob 接下来也能成为 Alice 的前 4 位上载者。换言之, 每过 30 秒 Alice 将随机地选择一名新的对换伴侣并开始与那位伴侣进行对换。如果这两名对等方都满足此对换, 它们将对方放入其前 4 位列表中并继续与对方进行对换, 直到该对等方之一发现了一个更好的伴侣为止。这种效果是对等方能够以趋向于找到彼此的协调的速率上载。随机选择邻居也允许新的对等方得到块, 因此它们能够具有对换的东西。除了这 5 个对等方 (“前” 4 个对等方和一个试探的对等方) 的所有其他相邻对等方均被 “阻塞”, 即它们不能从 Alice 接收到任何块。BitTorrent 有一些有趣的机制没有在这里讨论, 包括片 (小块)、流水线、随机优先选择、残局模型和反怠慢 [Cohen 2003]。

刚刚描述的关于交换的激励机制经常被称为 “一报还一报” (tit-for-tat) [Cohen 2003]。已证实这种激励方案能被回避 [Liogkas 2006; Locher 2006; Piatek 2007]。无论如何, BitTorrent “生态系统” 取得了广泛成功, 数以百万计的并发对等方在数十万条洪流中积极地共享文件。如果 BitTorrent 被设计为不采用一报还一报 (或一种变种), 然而在别的方面却完全相同的协议, BitTorrent 现在将很可能不复存在了, 因为大多数用户将成为搭便车者了 [Sarouiu 2002]。

[Guo 2005; Piatek 2007] 提出了 BitTorrent 协议的有趣变种。此外, 许多 P2P 直播流式应用如 PPLive 和 PPstream 从 BitTorrent 中获得了灵感 [Hei 2007]。

2.6.2 分布式散列表

在本节中, 我们将考虑在 P2P 网络中怎样实现一个简单的数据库。我们先从描述这种简单数据库的集中式版本开始, 该数据库只包含 (键, 值) 对。例如: 键可以是社会保险号, 值可以是相应的人名; 在这种情况下, 键 - 值对的例子如 (156-45-7081, Johnny Wu)。或者键可以是目录名 (例如, 电影、唱片和软件的名字), 值可以是存储内容的 IP 地址; 在这种情况下, 键 - 值对的例子如 (Led Zppelin IV, 128.187.123.38)。我们用键来查询数据库。如果在该数据库中有一个或多个键 - 值对匹配该查询键, 该数据库就返回相应的值。例如, 如果该数据库存储了社会保险号和它们对应的人名, 我们能够用特定的社会保险号查询, 该数据库就返回具有那个社会保险号的人的名字。或者, 如果数据库存储了目录名及其相应的 IP 地址, 我们能够用特定的目录名查询, 该数据库返回存储该特定内容的 IP 地址。

构建这样一个数据库直接采用客户 - 服务器体系结构, 以在一个中心服务器中存储所有 (键, 值) 对。因此在本节中, 我们将考虑构建这种数据库的一个分布式、P2P 的版本, 在数以百万计的对等方上存储 (键, 值) 对。在该 P2P 系统中, 每个对等方将保持 (键, 值) 对仅占总体的一个小子集。我们将允许任何对等方用一个特别的键来查询该分布式数据库。分布式数据库则将定位拥有该相应 (键, 值) 对的对等方, 然后向查询的对