

2^n 规则, 在具有最小标识符的对等方中存储 (键, 值) 对。]

此时假定某对等方 Alice 要将一个 (键, 值) 对插入 DHT。从概念上讲, 这是直接的: 她首先确定标识符最邻近该键的对等方; 然后她向那个对等方发送一个报文, 指示它存储该 (键, 值) 对。但是 Alice 怎样确定最邻近该键的对等方呢? 如果 Alice 要联系系统中所有对等方 (对等方 ID 和相应的 IP 地址), 她能够在本地确定最邻近对等方。但是这样的方法要求每个对等方联系在该 DHT 中的所有其他对等方, 而这对于具有数以百万计对等方的大规模系统而言, 是完全不现实的。

1. 环形 DHT

为了处理规模的问题, 我们现在考虑将对等方组织为一个环。在这种环形设置中, 每个对等方仅与它的直接后继和直接前任联系 (模 2^n)。在图 2-27a 中显示了这种环的一个例子。在该例中, n 仍取为 4, 有与前例同样的 8 个对等方。每个对等方仅知道它的直接后继和直接前任, 例如, 对等方 5 知道对等方 8 和 4 的 IP 地址和标识符, 但不必知道在该 DHT 中任何其他对等方的情况。对等方的这种环形设置是覆盖网络的一个特殊情况。在一个覆盖网络中, 对等方形成了一个抽象的逻辑网, 该网存在于由物理链路、路由器和主机组成的“底层”计算机网络之上。在覆盖网络中的链路不是物理链路, 而仅是对等方对之间的虚拟联络。在图 2-27a 所示的覆盖网络中, 有 8 个对等方和 8 条覆盖链路; 在图 2-27b 所示的覆盖网络中, 有 8 个对等方和 16 条覆盖链路。一条单一的覆盖链路通常使用了底层网络中的许多条物理链路和物理路由器。

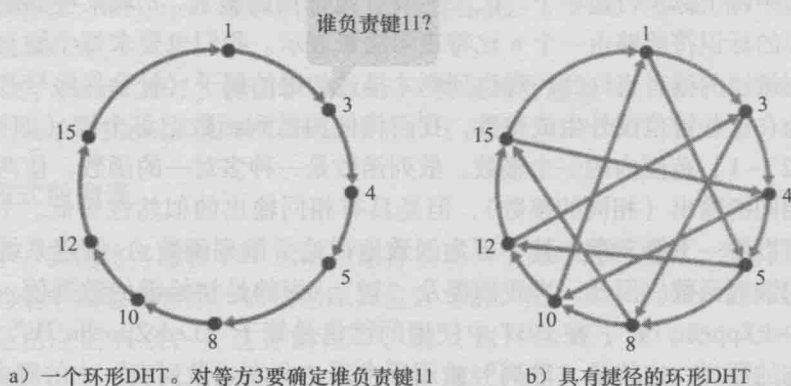


图 2-27 在环形 DHT 中确定最邻近键

使用图 2-27a 中的环形覆盖网络, 现在假定对等方 3 要确定 DHT 中的哪个对等方负责键 11。使用该环形覆盖网络, 初始对等方 (对等方 3) 生成一个报文, 问“谁负责键 11?”并绕环顺时针发送该报文。无论何时某对等方接收到该报文, 因为它知道其后继和前任的标识符, 它能够确定是否由它负责 (即最邻近) 查询中的键。如果某对等方不负责该键, 它只需将该报文发送给它的后继。因此, 例如当对等方 4 接收到询问键 11 的报文, 它确定自己不负责该键 (因为其后继更邻近该键), 故它只需将该报文传递给对等方 5。这个过程直到该报文到达对等方 12 才终止, 对等方 12 确定自己是最邻近键 11 的对等方。此时, 对等方 12 能够向查询的对等方即对等方 3 回送一个报文, 指出自己负责键 11。

为减少每个对等方必须管理的覆盖信息的数量, 环形 DHT 提供了一种非常精确有效