

- P27. 在 2.6.2 节的环形 DHT 例子中, 假定对等方 3 知道对等方 5 已经离开。对等方 3 如何更新它的后继状态信息? 此时哪个对等方是它的第一个后继? 哪个是其第二个后继?
- P28. 在 2.6.2 节的环形 DHT 例子中, 假定一个新的对等方 6 要接入该 DHT, 并且对等方 6 最初只知道对等方 15 的 IP 地址。需要采用哪些步骤?
- P29. 因为一个位于  $[0, 2^n - 1]$  的整数能被标识为一个在 DHT 中的  $n$  比特的二进制数, 每个键能被表示为  $k = (k_0, k_1, \dots, k_{n-1})$ , 并且每个对等方标识符能被表示为  $p = (p_0, p_1, \dots, p_{n-1})$ 。我们现在定义键  $k$  和对等方  $p$  的异或 (XOR) 距离为

$$d(k, p) = \sum_{j=0}^{n-1} |k_j - p_j| 2^j$$

描述该度量如何用于为对等方分配 (键, 值) 对。(要学习如何使用这个天然的度量构建有效的 DHT, 参见描述 Kademlia DHT 的文献 [Maymounkov 2002]。)

- P30. 由于 DHT 是覆盖网络, 它们也许不必与底层的物理网络匹配得很好, 即两个相邻的对等方也许物理上相距很远; 例如, 一个对等方可能位于亚洲而它的邻居可能位于北美。如果我们随机并统一地为新加入的对等方分配标识符, 这个分配方案将会引起这种误匹配吗? 揭示原因。这种误匹配如何影响 DHT 的性能呢?
- P31. 在一台主机上安装编译 TCPClient 和 UDPClient Python 程序, 在另一台主机上安装编译 TCPServer 和 UDPServer 程序。
- 如果你在运行 TCPServer 之前运行 TCPClient, 将发生什么现象? 为什么?
  - 如果你在运行 UDPServer 之前运行 UDPClient, 将发生什么现象? 为什么?
  - 如果你对客户端和服务端使用了不同的端口, 将发生什么现象?
- P32. 假定在 UDPClient.py 中在创建套接字后增加了下面一行:
- ```
clientSocket.bind(('', 5432))
```
- 有必要修改 UDPServer.py 吗? UDPClient 和 UDPServer 中的套接字端口号是多少? 在变化之前它们是多少?
- P33. 你能够配置浏览器以打开对某 Web 站点的多个并行连接吗? 有大量的并行 TCP 连接的优点和缺点是什么?
- P34. 我们已经看到因特网 TCP 套接字将数据处理为字节流, 而 UDP 套接字识别报文边界。面向字节 API 与显式识别和维护应用程序定义的报文边界的 API 相比, 试给出一个优点和一个缺点。
- P35. 什么是 Apache Web 服务器? 它值多少钱? 它当前有多少功能? 为回答这个问题, 你也许要看一下维基百科。
- P36. 许多 BitTorrent 客户使用 DHT 来创建一个分布式跟踪器。对于这些 DHT, “键”是什么, “值”是什么?



## 套接字编程作业

配套 Web 网站包括了 6 个套接字编程作业。前四个作业简述如下。第 5 个作业利用了 ICMP 协议, 在第 4 章结尾简述。第 6 个作业使用了多媒体协议, 在第 7 章结尾简述。极力推荐学生们完成这些作业中的几个 (如果不是全部的话)。学生们能够在 Web 网站 <http://www.awl.com/kurose-ross> 上找到这些作业的全面细节, 以及 Python 代码的重要片段。

### 作业 1: Web 服务器

在这个编程作业中, 你将用 Python 语言开发一个简单的 Web 服务器, 它仅能处理一个请求。具体而言, 你的 Web 服务器将: (1) 当一个客户 (浏览器) 联系时创建一个连接套接字; (2) 从这个连接接收 HTTP 请求; (3) 解释该请求以确定所请求的特定文件; (4) 从服务器的文件系统获得请求的文件; (5) 创建一个由请求的文件组成的 HTTP 响应报文, 报文前面有首部行; (6) 经 TCP 连接向请求的浏览