

我们首先来确定对于客户-服务器体系结构的分发时间,我们将其表示为 D_{cs} 。在客户-服务器体系结构中,没有对等方参与来帮助分发文件。我们做下列观察:

- 服务器必须向 N 个对等方的每个传输该文件的一个副本。因此该服务器必须传输 NF 比特。因为该服务器的上载速率是 u_s , 分发该文件的时间必定是至少为 NF/u_s 。
- 令 d_{\min} 表示具有最小下载速率的对等方的下载速率, 即 $d_{\min} = \min\{d_1, d_p, \dots, d_N\}$ 。具有最小下载速率的对等方不可能在少于 F/d_{\min} 秒时间内获得该文件的所有 F 比特。因此最小分发时间至少为 F/d_{\min} 。

将这两个观察放在一起, 我们得到

$$D_{cs} \geq \max\left\{\frac{NF}{u_s}, \frac{F}{d_{\min}}\right\}$$

该式提供了对于客户-服务器体系结构的最小分发时间的下界。在课后习题中将请你给出服务器能够调度它的传输以便实际取得该下界的方法。因此我们取上面提供的这个下界作为实际发送时间, 即

$$D_{cs} = \max\left\{\frac{NF}{u_s}, \frac{F}{d_{\min}}\right\} \quad (2-1)$$

我们从式 (2-1) 看到, 对足够大的 N , 客户-服务器分发时间由 NF/u_s 确定。所以, 该分发时间随着对等方 N 的数量线性地增加。因此举例来说, 如果从某星期到下星期对等方的数量从 1000 增加了 1000 倍, 到了 100 万, 将该文件分发到所有对等方所需要的时间就要增加 1000 倍。

我们现在来对 P2P 体系结构进行简单的分析, 其中每个对等方能够帮助服务器分发该文件。特别是, 当一个对等方接收到某些文件数据, 它能够使用自己的上载能力重新将数据分发给其他对等方。计算 P2P 体系结构的分发时间在某种程度上比计算客户-服务器体系结构的更为复杂, 因为分发时间取决于每个对等方如何向其他对等方分发该文件的各个部分。无论如何, 能够得到对该最小分发时间的一个简单表示式 [Kumar 2006]。至此, 我们先做下列观察:

- 在分发的开始, 只有服务器具有文件。为了使社区的这些对等方得到该文件, 该服务器必须经其接入链路至少发送该文件的每个比特一次。因此, 最小分发时间至少是 F/u_s 。(与客户-服务器方案不同, 由服务器发送过一次的比特可能不必由该服务器再次发送, 因为对等方在它们之间可以重新分发这些比特。)
- 与客户-服务器体系结构相同, 具有最低下载速率的对等方不能够以小于 F/d_{\min} 秒的分发时间获得所有 F 比特。因此最小分发时间至少为 F/d_{\min} 。
- 最后, 观察到系统整体的总上载能力等于服务器的上载速率加上每个单独的对等方的上载速率, 即 $u_{\text{total}} = u_s + u_1 + \dots + u_N$ 。系统必须向这 N 个对等方的每个交付 (上载) F 比特, 因此总共交付 NF 比特。这不能以快于 u_{total} 的速率完成。因此, 最小的分发时间也至少是 $NF/(u_s + u_1 + \dots + u_N)$ 。

将这三个观察放在一起, 我们获得了对 P2P 的最小分发时间, 表示为 D_{p2p} 。

$$D_{p2p} \geq \max\left\{\frac{F}{u_s}, \frac{F}{d_{\min}}, \frac{NF}{u_s + \sum_{i=1}^N u_i}\right\} \quad (2-2)$$