

TCP 连接握手需要经过以下几个步骤。

- (1) 请求新的 TCP 连接时，客户端要向服务器发送一个小的 TCP 分组（通常是 40 ~ 60 个字节）。这个分组中设置了一个特殊的 SYN 标记，说明这是一个连接请求。（参见图 4-8a）。
- (2) 如果服务器接受了连接，就会对一些连接参数进行计算，并向客户端回送一个 TCP 分组，这个分组中的 SYN 和 ACK 标记都被置位，说明连接请求已被接受（参见图 4-8b）。
- (3) 最后，客户端向服务器回送一条确认信息，通知它连接已成功建立（参见图 4-8c）。现代的 TCP 栈都允许客户端在这个确认分组中发送数据。

HTTP 程序员永远不会看到这些分组——这些分组都由 TCP/IP 软件管理，对其是不可见的。HTTP 程序员看到的只是创建 TCP 连接时存在的时延。

通常 HTTP 事务都不会交换太多数据，此时，SYN/SYN+ACK 握手（参见图 4-8a 和图 4-8b）会产生一个可测量的时延。TCP 连接的 ACK 分组（参见图 4-8c）通常都足够大，可以承载整个 HTTP 请求报文<sup>4</sup>，而且很多 HTTP 服务器响应报文都可以放入一个 IP 分组中去（比如，响应是包含了装饰性图片的小型 HTML 文件，或者是对浏览器高速缓存请求产生的 304 Not Modified 响应）。

82

最后的结果是，小的 HTTP 事务可能会在 TCP 建立上花费 50%，或更多的时间。后面的小节会讨论 HTTP 是如何通过重用现存连接，来减小这种 TCP 建立时延所造成的影响的。

#### 4.2.4 延迟确认

由于因特网自身无法确保可靠的分组传输（因特网路由器超负荷的话，可以随意丢弃分组），所以 TCP 实现了自己的确认机制来确保数据的成功传输。

每个 TCP 段都有一个序列号和数据完整性校验和。每个段的接收者收到完好的段时，都会向发送者回送小的确认分组。如果发送者没有在指定的窗口时间内收到确认信息，发送者就认为分组已被破坏或损毁，并重发数据。

由于确认报文很小，所以 TCP 允许在发往相同方向的输出数据分组中对其进行“捎带”。TCP 将返回的确认信息与输出的数据分组结合在一起，可以更有效地利用网络。为了增加确认报文找到同向传输数据分组的可能性，很多 TCP 栈都实现了一种

---

注 4：因特网流量中的 IP 分组通常是几百字节，本地流量中的 IP 分组为 1500 字节左右。