

问题。尽管这在概念上简单，但是当在如 TCP 这样的实际协议中实现超时/重传机制时还是会产生许多微妙的问题。也许最明显的一个问题就是超时间隔长度的设置。显然，超时间隔必须大于该连接的往返时间（RTT），即从一个报文段发出到它被确认的时间。否则会造成不必要的重传。但是这个时间间隔到底应该是多大呢？刚开始时应如何估计往返时间呢？是否应该为所有未确认的报文段各设一个定时器？问题如此之多！我们在本节中的讨论基于 [Jacobson 1988] 中有关 TCP 的工作以及 IETF 关于管理 TCP 定时器的建议 [RFC 6298]。

1. 估计往返时间

我们开始学习 TCP 定时器的管理问题，要考虑一下 TCP 是如何估计发送方与接收方之间往返时间的。这是通过如下方法完成的。报文段的样本 RTT（表示为 SampleRTT）就是从某报文段被发出（即交给 IP）到对该报文段的确认被收到之间的时间量。大多数 TCP 的实现仅在某个时刻做一次 SampleRTT 测量，而不是为每个发送的报文段测量一个 SampleRTT。这就是说，在任意时刻，仅为一个已发送的但目前尚未被确认的报文段估计 SampleRTT，从而产生一个接近每个 RTT 的新 SampleRTT 值。另外，TCP 决不为已被重传的报文段计算 SampleRTT；它仅为传输一次的报文段测量 SampleRTT [Kan 1987]。（本章后面的一个习题请你考虑一下为什么要这么做。）

显然，由于路由器的拥塞和端系统负载的变化，这些报文段的 SampleRTT 值会随之波动。由于这种波动，任何给定的 SampleRTT 值也许都是非典型的。因此，为了估计一个典型的 RTT，自然要采取某种对 SampleRTT 取平均的办法。TCP 维持一个 SampleRTT 均值（称为 EstimatedRTT）。一旦获得一个新 SampleRTT 时，TCP 就会根据下列公式来更新 EstimatedRTT：

$$\text{EstimatedRTT} = (1 - \alpha) \cdot \text{EstimatedRTT} + \alpha \cdot \text{SampleRTT}$$

上面的公式是以编程语言的语句方式给出的，即 EstimatedRTT 的新值是由以前的 EstimatedRTT 值与 SampleRTT 新值加权组合而成的。在 [RFC 6298] 中给出的 α 参考值是 $\alpha = 0.125$ （即 $1/8$ ），这时上面的公式变为：

$$\text{EstimatedRTT} = 0.875 \cdot \text{EstimatedRTT} + 0.125 \cdot \text{SampleRTT}$$

值得注意的是，EstimatedRTT 是一个 SampleRTT 值的加权平均值。如在本章后面习题中讨论的那样，这个加权平均对最近的样本赋予的权值要大于对老样本赋予的权值。这是很自然的，因为越近的样本越能更好地反映网络的当前拥塞情况。从统计学观点讲，这种平均被称为指数加权移动平均（Exponential Weighted Moving Average, EWMA）。在 EWMA 中的“指数”一词看起来是指一个给定的 SampleRTT 的权值在更新的过程中呈指数型快速衰减。在课后习题中，将要求你推导出 EstimatedRTT 的指数表达形式。

图 3-32 显示了当 $\alpha = 1/8$ 时，在 gaia.cs.umass.edu（在美国马萨诸塞州的 Amherst）与 fantasia.eurecom.fr（在法国南部）之间的一条 TCP 连接上的 SampleRTT 值与 EstimatedRTT 值。显然，SampleRTT 的变化在计算 EstimatedRTT 的过程中趋于平缓。

除了估算 RTT 外，测量 RTT 的变化也是有价值的。[RFC 6298] 定义了 RTT 偏差 DevRTT，用于估算 SampleRTT 一般会偏离 EstimatedRTT 的程度：

$$\text{DevRTT} = (1 - \beta) \cdot \text{DevRTT} + \beta \cdot |\text{SampleRTT} - \text{EstimatedRTT}|$$

注意到 DevRTT 是一个 SampleRTT 与 EstimatedRTT 之间差值的 EWMA。如果 SampleRTT