值波动较小，那么 DevRTT 的值就会很小；另一方面，如果波动很大，那么 DevRTT 的值就会很大。β 的推荐值为 0.25。

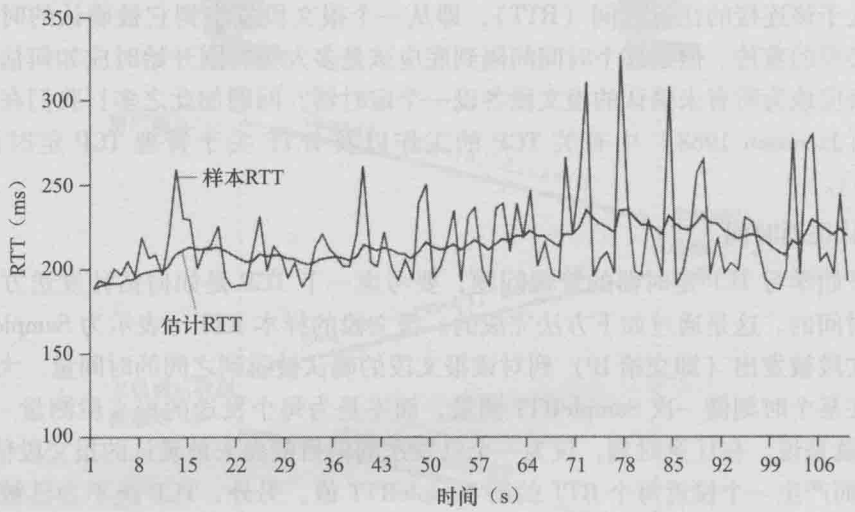

图 3-32 RTT 样本和 RTT 估计

2. 设置和管理重传超时间隔

假设已经给出了 EstimatedRTT 值和 DevRTT 值，那么 TCP 超时间隔应该用什么值呢？很明显，超时间隔应该大于等于 EstimatedRTT，否则，将造成不必要的重传。但是超时间隔也不应该比 EstimatedRTT 大太多，否则当报文段丢失时，TCP 不能很快地重传该报文段，导致数据传输时延大。因此要求将超时间隔设为 EstimatedRTT 加上一定余量。当 SampleRTT 值波动较大时，这个余量应该大些；当波动较小时，这个余量应该小些。因此，DevRTT 值应该在这里发挥作用了。在 TCP 的确定重传超时间隔的方法中，所有这些因素都考虑到了：

$$\text{TimeoutInterval} = \text{EstimatedRTT} + 4 \cdot \text{DevRTT}$$

推荐的初始 TimeoutInterval 值为 1 秒［RFC 6298］。同样，当出现超时后，TimeoutInterval 值将加倍，以免即将被确认的后继报文段过早出现超时。不管怎样，一旦报文段收到并更新 EstimatedRTT 后，TimeoutInterval 就又使用上述公式计算了。

实 践 原 则

与我们在 3.4 节中所学的方法很像，TCP 通过使用肯定确认与定时器来提供可靠数据传输。TCP 确认正确接收到的数据，而当认为报文段或其确认报文丢失或受损时，TCP 会重传这些报文段。有些版本的 TCP 还有一个隐式 NAK 机制（在 TCP 的快速重传机制下，收到对一个特定报文段的 3 个冗余 ACK 就可作为对后面报文段的一个隐式 NAK，从而在超时之前触发对该报文段的重传。TCP 使用一系列编号以使接收方能识别丢失或重复的报文段。像可靠数据传输协议 rdt3.0 的情况一样，TCP 自己也无法确认一个报文段或其 ACK 是丢失了还是受损了，或是时延过长了。在发送方，TCP 的响应是相同的：重传有问题的报文段。