11.1.1 TCP 重传

重传数据包是 TCP 最基本的错误恢复特性之一，它被设计用来对付数据包丢失。

数据包丢失可能有很多原因，包括出故障的应用程序、流量负载沉重的路由器或者临时性的服务中断。数据包层次上的移动速度非常快，而且数据包丢失通常是暂时的，因此 TCP 能否检测到数据包丢失并从中恢复显得至关重要。

决定是否有必要重传数据包的主要机制叫作重传计时器。这个计时器负责维护一个叫重传超时（Retransmission timeout，RTO）的值。每当使用 TCP 传输一个数据包时，就启动重传计时器。当收到这个数据包的 ACK 时，计时器就会停止。从发送数据包到接收 ACK 确认之间的时间被称为往返时间（Round-trip time，RTT）。将若干个这样的时间平均下来，可算出最终的 RTO 值。

在最终算出 RTO 值之前，传输操作系统将一直依赖于默认配置的 RTT 值。此项设定用于主机间的初始通信，并基于接收到的数据包 RTT 进行调整，以形成真正的 RTO。

一旦 RTO 值确定下来，重传定时器就被用于每个传输的数据包，以确定数据包是否丢失。图 11-1 阐述了 TCP 重传过程。

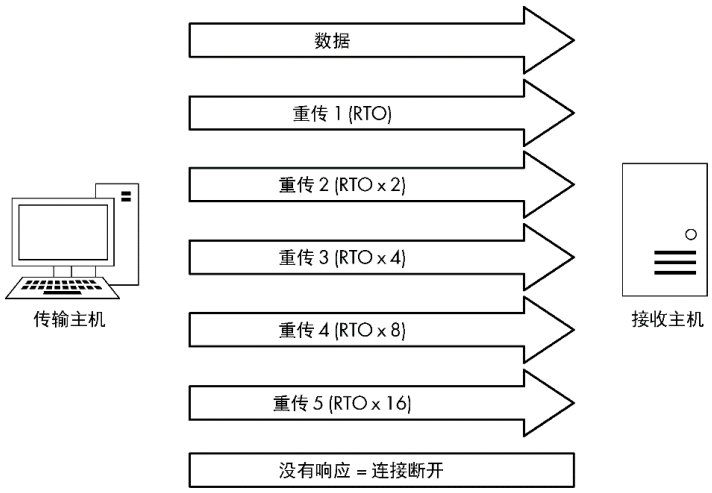

图 11-1　TCP 重传过程的概念视图

当数据包被发送出去，但接收方没有发送 TCP ACK 数据包时，传输主机就假设原来的数据包丢失了，并重传它。重传之后，RTO 值翻倍。如果在到达那个值之前一直没有接收到 ACK 数据包，则将发生另一次重传。如

果下一次重传还是没有收到 ACK，那么 RTO 值将翻倍。每次重传，RTO 值都将翻倍，这个过程会持续到收到一个 ACK 数据包，或者发送方达到配置的最大重传次数为止。

最大重传次数取决于传输操作系统上的配置。默认情况下，Windows 主机最多重传 5 次，大部分 Linux 主机则默认重传 15 次。这个选项在两个操作系统中都是可配置的。

要看 TCP 重传的例子，请打开 tcp_retransmissions.pcap 文件，它包含了 6 个数据包。第一个数据包如图 11-2 所示。

这是一个 TCP PSH/ACK 数据包 ❶，包含 648 字节的数据 ❷，从 10.3.30.1 发送到 10.3.71.7❸。这是一个典型的数据包。

在正常条件下，你会期待在发送第一个数据包之后，很快就能看到响应的 TCP ACK 数据包。然而，在这个例子中，下一个数据包是一次重传。通过在 Packet List 面板中查看这个数据包，你就能得出这个结论。Info 列明确表明了[TCP Retransmission]，并且这个数据包以黑底红字出现。图 11-3 显示了 Packet List 面板中列出的重传例子。

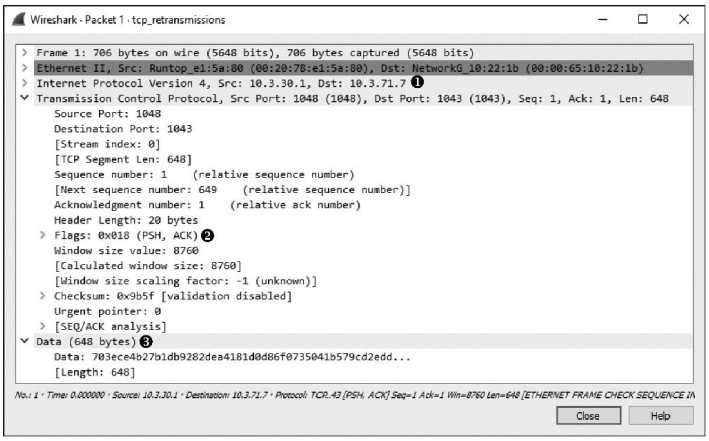

图 11-2　包含数据的简单 TCP 数据包

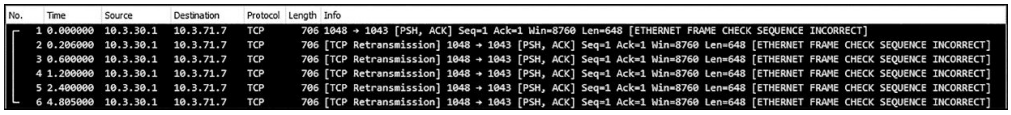

图 11-3　Packet List 面板中的重传

如图 11-4 所示，你也可以通过查看 Packet Details 和 Packet Bytes 面板确定它是否是重传数据包。

注意，除了 IP identification 和 Checksum 域之外，这个数据包与最初的数据包完全一致。为了验证这个结论，可在 Packet Bytes 面板中比较这个重传数据包和最初的数据包 ❶。

在 Packet Details 面板中，注意到重传数据包的 SEQ/ACK Analysis 标题下有一些额外的信息 ❷。这个有用的信息是由 Wireshark 提供的，实际上并不包含在数据包里。SEQ/ACK analysis 告诉我们这确实是一个重传 ❸，RTO 值 0.206s ❹ 是基于与数据包 1❺ 的时间差值算出来的。

查看剩下的数据包应该是类似的结果，唯一的不同在于 IP identification、Checksum 域以及 RTO 值。为了显示每个数据包之间的时间间隔，如图 11-5 所示，可以查看 Packet List 面板的 Time 列。在这里，你看到每一次重传后 RTO 值翻倍，时间呈指数增长。

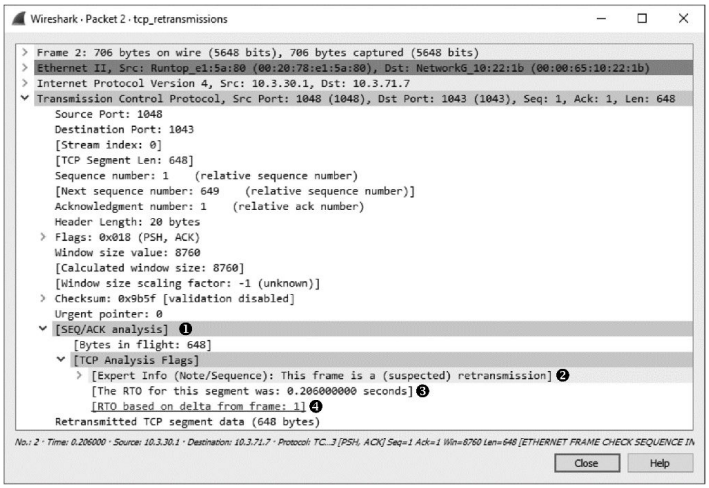

图 11-4　一个重传数据包

No.	Time
1	0.000000
2	0.206000
3	0.600000
4	1.200000
5	2.400000
6	4.805000

图 11-5　Time 列显示了 RTO 值的增长

传输设备使用 TCP 的重传特性来检测数据包丢失并从中恢复。下一步，我们将查看 TCP 的重复确认特性，它被接收方用于检测数据包丢失并从中恢复。