## 7.3.4 路由跟踪

路由跟踪功能用来识别一个设备到另一个设备的通路。在一个简单的网络上，这个通路可能只经过一个路由器，甚至一个都不经过。但在复杂的网络中，数据包可能要经过数十个路由器才会到达最终目的地。确定数据包从一个目的地到另一个目的地的实际路径，对于通信检修十分重要。

通过使用 ICMP（在 IP 协议的帮助下），路由跟踪可以画出数据包的路径。举例来说，文件 icmp_traceroute.pcap 中的第一个数据包，和我们在上一节中看到的 echo 请求（见图 7-32）很类似。

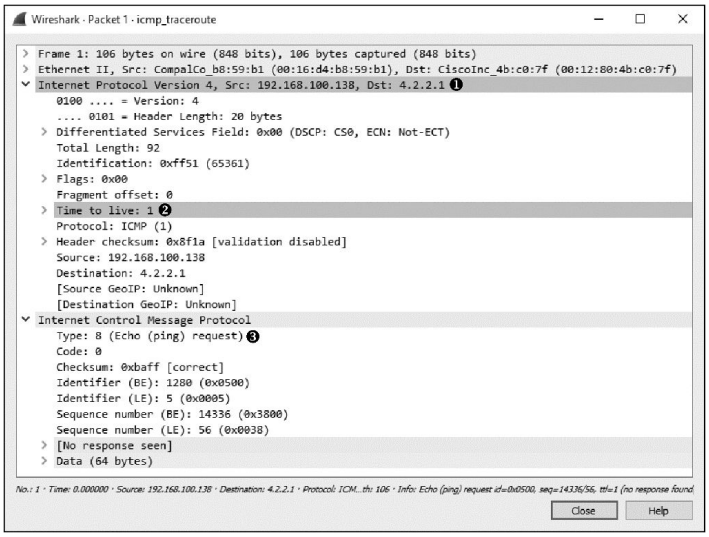


图 7-32　一个 TTL 值为 1 的 ICMP echo 请求数据包

乍看起来，这个数据包就是一个从 192.168.100.138 到 4.2.2.1 的简单 echo 请求，并且 ICMP 中的每一个部分都与 echo 请求数据包相同。但是当展开这个数据包的 IP 头时，你可以注意到一个奇怪的数字。这个数据包的 TTL 被设为了 1，也就意味着这个数据包会在它遇到的第一个路由器处被丢掉。因为目标地址 4.2.2.1 是一个互联网地址，我们就会知道源设备和目的设备之前至少会有一个路由器，所以这个数据包不会到达目的地。这对我们来说是个好事，因为路由跟踪正是需要这个数据包只到达它传输的第一个路由器。

第二个数据包正如所期望的那样，是前往目的地路径上第一个路由器发回的响应（见图 7-33）。在数据包到达 192.168.100.1 这个设备后，它的 TTL 减为 0，所以它不能继续传输，于是路由器回复了一个 ICMP 响应。这个数据包的类型是 11，代码是 0，也就是告诉我们由于数据包的 TTL 在传输过程中超时，因此目的不可达。

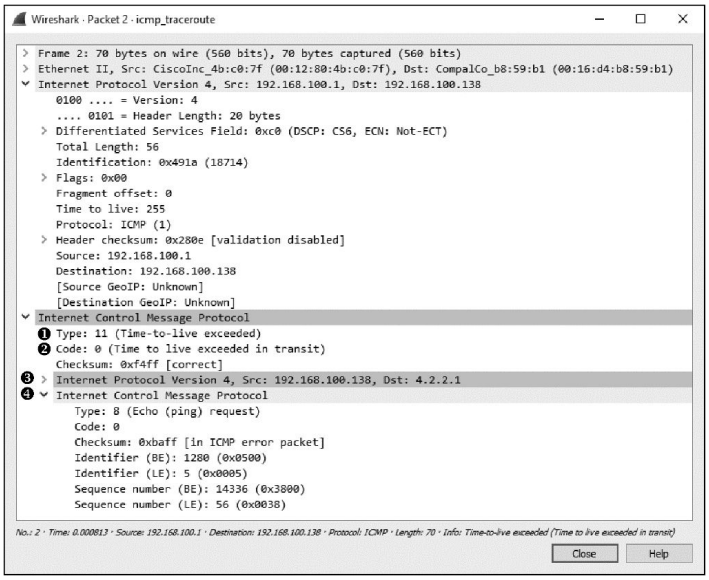


图 7-33　来自路径上第一个路由器的 ICMP 响应

这个 ICMP 数据包有时候被叫作双头包，因为这个 ICMP 的结尾部分包含了原先 echo 请求的 IP 头和 ICMP 数据的副本。这个信息被证明在网络检修的时候非常有用。

在第 7 个数据包前，这种发送 TTL 自增数据包的过程又出现了两次。在这里，除了 IP 头的 TTL 值被设为了 2，从而保证这个数据包会在被丢弃前到达第二跳路由，你还可以看到和第一个数据包相同的东西。和我们所期望的一样，我们从下一跳的路由 12.180.241.1 接到了一个有着同样 ICMP 目的不可达和 TTL 超时的响应消息。这种将 TTL 自增 1 的过程，一直持续到数据包到达目的地址 4.2.2.1。

总结来说，路由跟踪要与路径上的每一个路由器进行通信，从而画出前往目的地的路由图，如图 7-34 所示。

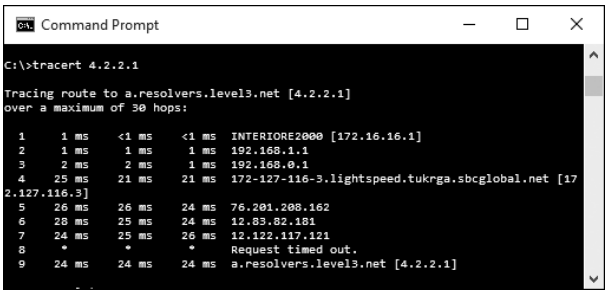


图 7-34　路由跟踪功能的样例输出

> 注意
>
> 我们这里所讨论的路由跟踪主要基于 Windows，因为只有它在使用 ICMP。Linux 上的路由跟踪更复杂一些，并使用了其他协议来进行路由路径的跟踪。