视频 - Wireshark 中的 IPv6 报头示例 （ 6 分钟 ）

此屏幕截图显示了一个使用 Wireshark 的数据包捕获和来自一次 IPv6 对话的网络层信息。让我们看一下。在此屏幕截图中，可以看到高亮显示的数据包是 46 号数据包数据包列表窗口中的源地址显示它是一个全局单播 IPv6 地址。可以看到此地址以 2001:6f8 开头。目的地址也是全局单播地址 2001:6f8:900 等等。如果查看协议字段，我们会在上层看到，这是一个 TCP 数据包，是一次与一个 HTTP Web 服务器建立初始通信的尝试。如果在网络层信息区域中向下看，可以看到 IPv6 信息已展开。我们看看针对 Internet 协议第 6 版的一些协议字段信息。首先，可以看到 IPv6 报头中的信息量比 IPv4 报头中小得多。

有一些有趣的功能。首先，可以看到版本字段是相同的。在本例中显示为 6，将这个数据包标识为 IPv6。还可以在这里看到二进制数 6。下一个字段是流量类型字段。流量类型字段的作用与 IPv4 数据包中的差异化服务字段相同。它处理流量优先级和拥塞。可以看到的下一部分是流标签。流标签字段是一个针对 IPv6 协议的新字段。其目的是在路由器和交换机中维持相同的数据包流，以便为需要数据包按相同顺序到达的实时应用程序提供帮助。可以看到下一个字段是负载长度字段。它与 IPv4 报头中的总长度字段相同。此字段告诉我们数据包的总大小--在本例中为 40 字节。下一个报头字段的作用与针对 IPv4 的协议字段相同。可以看到它表明此数据包的上层数据部分是 6 或 TCP。跳数限制的作用与 IPv4 数据包中的 TTL 字段相同。可以看到当前的跳数限制设置为 64 跳。当此值下降到 0 时，将丢弃该数据包。接下来，我们有源 IPv6 地址，目的 IPv6 地址，然后在上层，可以看到这是一个包含 TCP 报头信息的 TCP 数据包。我们看看下一个屏幕截图。在下一个屏幕截图中，可以看到我们高亮显示了 49 号数据包。

我们已与此 Web 服务器建立连接。此数据包现在是一个发送到 Web 服务器的 GET 请求。如果在展开的 Internet 协议第 6 版数据包详细信息窗口中向下看，可以看到负载长度大得多。可以在 IPv6 信息下方看到 TCP 信息，现在我们的 GET 请求中还有 HTTP 协议信息。这是我们获取网页的 GET 请求。转到下一个屏幕截图，最后一个屏幕截图显示了一条 ICMP 第 6 版邻居请求消息。如果查看此窗口中高亮显示的数据包，在 1 号数据包中，我们将看到源地址这一次不是全局单播 IPv6 地址，而是一个本地链路地址。可以从这里的 fe80 确定这一点。还可以看到此本地链路地址使用了 EUI-64 来解析地址的使用了 EUI-64 来标识地址的接口标识部分。可以通过该地址内的 ff:fe 确定这一点。目的地址是一个 ff02 IPv6 地址，表明这是一个组播数据包。如果查看协议，可以看到它是 ICMP 第 6 版，然后我们从有关数据包的信息了解到，这是对我们在之前的屏幕截图中联系的同一个设备的邻居请求消息。此数据包的功能实质上类似于 IPv4 中的 ARP 请求。我们需要发现此设备的本地链路地址，所以我们发出一条 ICMP 第 6 版邻居请求消息，将其以组播形式发送，我们希望从此邻居返回一个本地链路地址。如果在展开的详细信息窗口向下看，可以看到第 6 版、流量类型、流标签、负载长度（数据包的总长度）；像显示为 58 的 IPv4 的协议字段一样，下一个报头字段是数据包的数据部分中的一条 ICMP 第 6 版消息；跳数限制 -- 255 跳。这类似于 TTL 字段。然后是源本地链路地址和目的组播 IPv6 地址。在底部的 IPv6 信息下方，可以看到有一个特定于 Internet 控制消息协议第 6 版的可展开区域。