1.2.4 数据封装

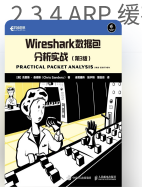
OSI 参考模型不同层次上的协议在数据封装的帮助下进行通信传输。议栈中的每层协议都负责在传输数据上增加一个协议头部或尾部，其中包了使协议栈之间能够进行通信的额外信息。例如，当传输层从会话层接收据时，它会在将数据传递到下一层之前，附上自己的头部信息数据。

数据封装过程将创建一个协议数据单元（PDU），其中包括正在发送网络数据，以及所有增加的头部与尾部协议信息。随着网络数据沿着 OS考模型向下流动，PDU 逐渐变化、增长，各层协议均将其头部或尾部信添加进去，直到物理层时达到其最终形式，并发送给目标计算机。接收计机收到 PDU 后，沿着 OSI 参考模型往上处理时，逐层剥去协议头部和尾当 PDU 到达 OSI 参考模型的最上层时，将只剩下原始传输数据。

> 注意
>
> OSI 参考模型使用特别的术语来描述每一层的数据。物理层叫比特，数据链路层叫帧，网络层叫数据包，传输层叫数据段。最上面的三层可以统称数据，但这些叫法实际上用得并不多，我们一般会使用报文来表示一个完整或部分 PDU，该 PDU 从多个 OSI 参考层中包含了表头和表尾息。

让我们通过一个实际的例子来理解数据的封装过程，这个例子描述了据包是如何在 OSI 参考模型中被创建、传输和接收的。作为数据包分析师你需要了解，我们经常会忽略掉会话层和表示层，所以它们将不会在这个子中出现（包括本书的其余部分）。[1]

假设这样一个情形：我们试着在计算机上浏览 Google。在这个过程中我们必须首先产生一个请求数据包，从客户端计算机传输到目标服务器上这里我们假设 TCP/IP 通信会话已经被建立，图 1-3 则展示了此案例中的数据封装处理过程。



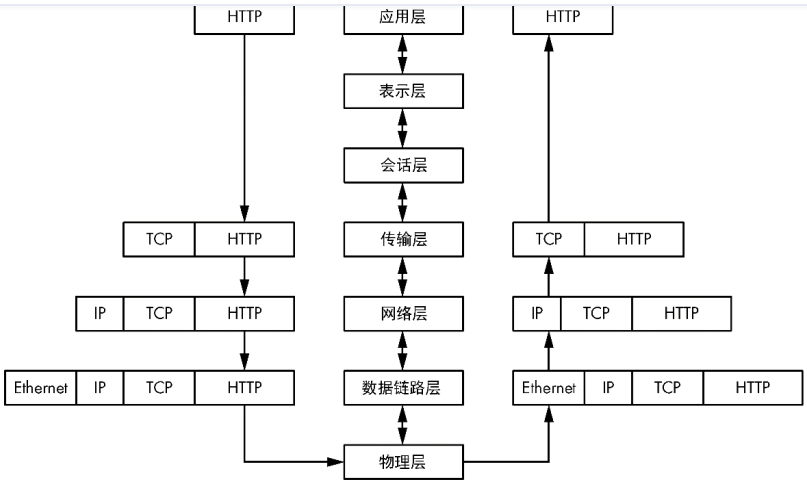


图 1-3　客户端和服务器之间数据封装过程图示

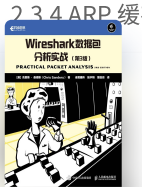
我们从客户端计算机的应用层开始，在我们浏览一个网站时，所使用
应用层协议是 HTTP，通过此协议发出请求命令，从 Google 下载
index.html 文件。

> 注意
>
> 在实践中，浏览器会向网站的根目录文件发出请求，通常使用正斜杠（/）来表示。当 Web
> 服务器接收到该请求时，它会根据服务器的网页根目录设定对浏览器重定向。根目录文件名通常
> 是 index.html 或 index.php。我们会在第 9 章讨论更多有关 HTTP 的内容。

应用层协议发送出指令后，我们就开始关心数据包是如何被发送到目
地的。数据包中的应用层数据将沿着 OSI 参考模型的协议栈传递给传输层
HTTP 是一个使用 TCP（或在 TCP 协议之上）的应用层协议，因此传输层
将使用 TCP 协议来确保数据包的可靠投递。一个包括序列号和其他数据的
TCP 协议头部将被创建，并添加到 PDU 中，如图 1-3 所示。该 TCP 表头
含了序列号和其他信息，以确保数据包能够被正确交付。

> 注意
>
> 我们常说一个协议在其他协议之上，是因为 OSI 参考模型的分层设计。例如 HTTP 等应用层
> 协议提供了一个特定的服务，并依靠 TCP 协议来保证服务的可靠交付。正如你学习到的，DNS
> 议架构于 UDP 之上，而 TCP 架构在 IP 之上。

在完成这项工作之后，TCP 协议将数据包交给 IP 协议，也就是在第
层上负责为数据包进行逻辑寻址的协议。IP 协议创建一个包含有逻辑寻址
息的头部，并将数据包传递给数据链路层上的以太网协议，然后以太网协
地址会被添加并存储在以太网帧头中。现在数据包已经完全组装好并传递



从中提取到所包含的物理以太网寻址信息，确定数据包是否是发往这台服器的。一旦处理完这些信息，第 2 层头部与尾部的信息将被剥除，并进入第 3 层的信息处理过程中。

第 3 层 IP 寻址信息会被读取，以便确认数据包被正确转发，以及数据包并未进行分片处理。这些信息也同样被剥除，并交到下一层进行处理。

第 4 层 TCP 协议信息现在被读取，以确保数据包是按序到达的。然后第 4 层报头信息被剥离，留下的只有应用层数据。这些数据会被传递到 Web 服务器应用程序。为了响应客户端发过来的这个数据包，服务器应该发回一个 TCP 确认数据包，使客户端知道它的请求已经被接收，并可以等待获取 index.html 文件内容了。

所有数据包都会以这个例子中描述的过程进行创建和处理，而无论使的是哪种协议。

但同时，请牢记并非每个网络数据包都是从应用层协议产生的，所以会进一步看到只包含第 2 层、第 3 层或第 4 层协议信息的数据包。