

原则：封装

尼克：这个视频是关于我们称之为封装的架构原则。封装是当你把层和包交换结合起来时发生的结果。我们想把数据分解成不连续的单位，称为数据包。然而，每个数据包都包含来自多个层的数据。例如，当你发送一个**TCP**段时，它在一个**IP**包内，而**IP**包又在一个以太网帧内。封装就是这样工作的。封装是一种原则，你通过它来组织数据包中的信息，这样你就可以保持各层，但又让它们分享你的数据包的内容。

分层

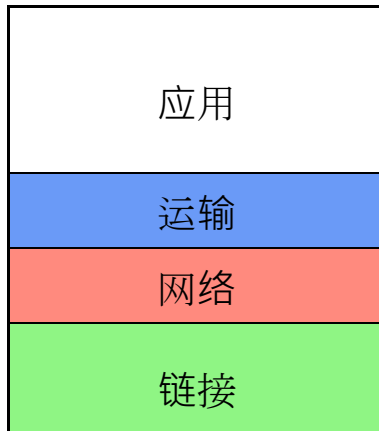


- 分开关注和责任
- 允许每项服务独立发展
- 例子。
 - 传输：应用间通信
 - 链接：共享链接上的主机间通信

菲尔：回顾一下，分层让你把一个复杂的系统分解成更小的部分。每一层都提供一种服务，是对上面各层的网络的一种抽象。它通过使用它下面的层来提供这种抽象。每个层都是自包含的，所以只要它提供了预期的服务，上面的层就不需要担心如何提供。这种关注点的分离意味着每个层可以独立发展。就像网络层的IP不需要担心传输层的TCP的变化一样，应用层如HTTP也不需要担心TCP的变化。例如，在过去的几年里，大多数操作系统已经改变了他们所使用的确切TCP算法，以更好地处理日益增长的网络速度。但网络浏览器使用旧的算法和新的算法都能正常工作。

请注意，这层图片有7层OSI模型。

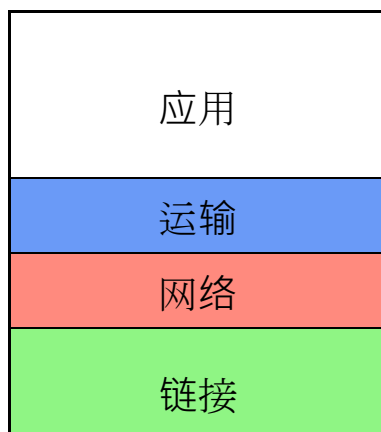
4层模型



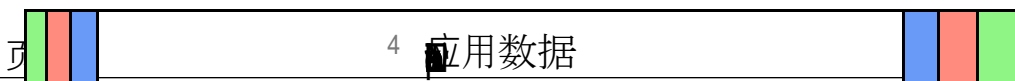
- 分开关注和责任
- 允许每项服务独立发展
- 例子。
 - 传输：应用间通信
 - 链接：共享链接上的主机间通信

尼克：因此，让我们重新回到4层模型。

封装



- 分层如何体现在数据表示中
- 第N层数据是到第N-1层的有效载荷
- 例子。
 - HTTP（网络）应用程序的有效载荷在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个WiFi链接帧。



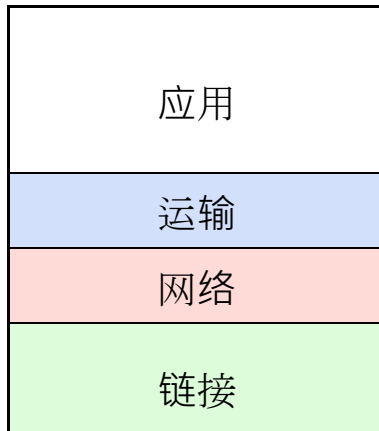
尼克：封装是让我们采取协议层的原则，让他们轻松地分享数据包内的存储。它是分层在实际数据表现中的表现方式。它的工作方式是每个协议层都有一些头，然后是它的有效载荷，最后是一些脚。例如，一个IP包头有一个源地址和一个目标地址。为了用IP发送一个TCP段，我们把TCP格式作为IP包的有效载荷。通过这种方式，IP包“封装”了TCP段。IP不知道也不关心它的有效载荷是什么，它只是把数据包送到终端主机。当数据包到达时，主机会查看有效载荷，看到它是一个TCP段，并对其进行相应处理。

尼克：所以这里有一个更完整的例子。假设你正在使用一台通过WiFi、无线以太网连接的电脑浏览网页。你的网络浏览器产生了一个HTTP GET请求。这个HTTP GET请求是一个TCP段的有效载荷。封装HTTP GET的TCP段是一个IP数据包的有效载荷。这个IP数据包，封装了TCP段和HTTP GET，是一个WiFi帧的有效载荷。如果你要看一下你的电脑发送的字节，它们看起来像这样。最外层的封装格式是WiFi帧，里面是一个IP包，里面是一个TCP段，里面是一个HTTP GET。

菲尔：尼克是如何抽到这个包的，这让你可能觉得很困惑！原来有两种抽包方式。事实证明，有两种方法来绘制数据包。这种区别来自于背景和你所工作的系统的哪一部分。尼克在这里画了一个数据包，标题在右边。数据包的第一部分在右边，数据包的最后部分在左边。

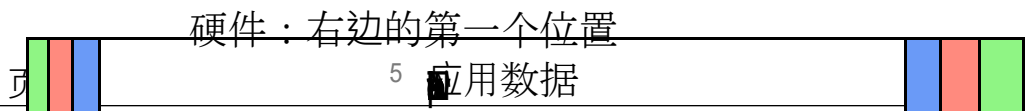
尼克：这完全说得通！当路由器或交换机发送一个数据包时，我们画出数据包从左到右移动。因此，第一个离开路由器或交换机的比特在最右边。

封装



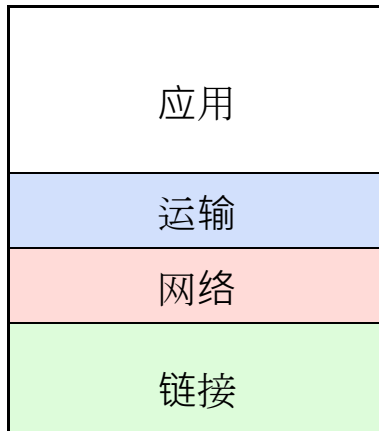
- 分层如何体现在数据表示中
- 第N层数据是到第N-1层的有效载荷
- 例子。
 - HTTP（网络）应用程序的有效载荷在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个WiFi链接帧。

斯坦福大学CS144



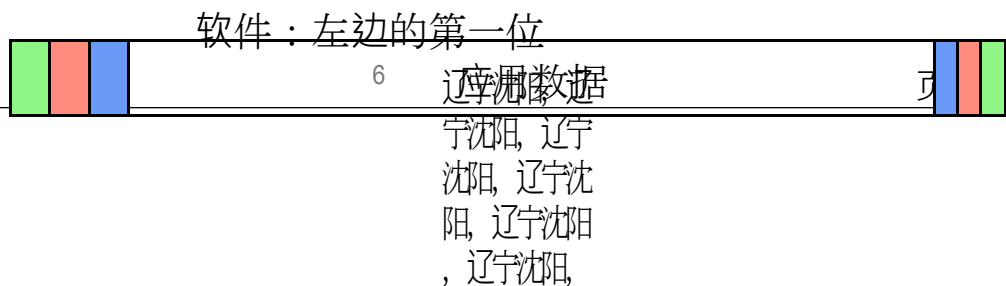
Phil:但我画的是另一种方式的数据包，即页眉在左边，页脚在右边。像这样。

封装



- 分层如何体现在数据表示中
- 第N层数据是到第N-1层的有效载荷
- 例子。
 - HTTP（网络）应用程序的有效载荷在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个WiFi链接帧。

斯坦福大学CS144

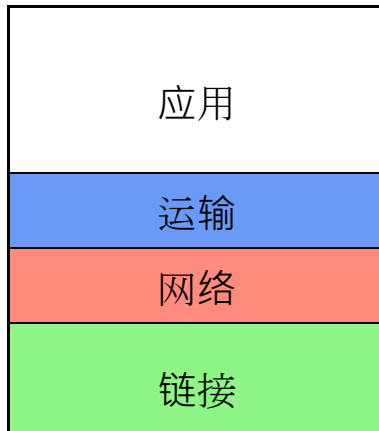


菲尔：这种方法来自于软件。当你阅读RFC文档和许多其他协议规范时，你会看到这种方法。这个想法是，数据包的开始是在地址零处。所以头的第一个字节是在地址零处。由于地址从左到右增加，这意味着数据包的开始在左边，数据包的结束在右边。

尼克：这里没有正确的方法，也没有错误的方法。两种绘制数据包的方法都很有价值，取决于你要用什么来绘制。你应该对两种方式都感到满意。我一般会把标题画在右边。

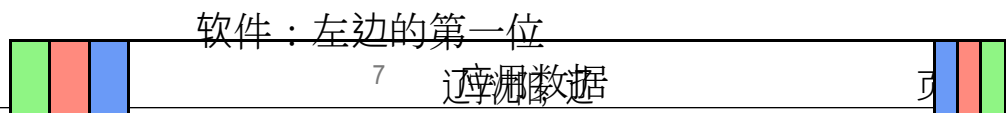
菲尔：而我一般会把它们画在左边。尼克的背景是电气工程和开关设计，我的背景是计算机科学和协议软件。

封装



- 分层如何体现在数据表示中
- 第N层数据是到第N-1层的有效载荷
- 例子。
 - HTTP（网络）应用程序的有效载荷在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个WiFi链接帧。

斯坦福大学CS144



菲尔：现在让我们回到尼克的例子，**HTTP GET**在一个**TCP**段里，一个**IP**包里，一个**WiFi**帧里。让我们用**Wireshark**看看这在实际网络中是什么样子。在我们开始记录之前，我打开了**Wireshark**并记录了一个网络请求的数据包追踪。

我们只看一个数据包。在这里，我们可以看到**Wireshark**是如何告诉我们，这是一个以太网帧，里面是一个**IP**包，里面是一个**TCP**段，里面是一个**HTTP GET**。如果我点击这些协议头的每一个，那么**Wireshark**实际上突出了它们在数据包字节中的位置，下面是这些乱七八糟的东西。**WiFi**是第一位的。

WiFi内部是**IP**。**IP**里面是**TCP**。而在**TCP**里面，我们可以看到我们的**HTTP GET**的文本

封装的灵活性

- 封装使你可以递归分层
- 例子。虚拟专用网络（VPN）。
 - HTTP（网络）应用程序的有效载荷在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个安全的TLS演示信息，在
 - 一个TCP传输段在
 - 一个IP网络数据包在
 - 一个以太网链接帧。

菲尔：这种非常简单的将协议相互封装的方法给了你巨大的灵活性。到目前为止，我们一直在谈论4层模型，认为它是完全静态和不灵活的东西。在实践中，它并不是这样的。你实际上可以使用封装来递归层协议。例如，今天在办公室和企业中非常常用的东西是一个叫做虚拟专用网络的东西。通过虚拟专用网络，你打开一个安全连接到一个你信任的网络，例如你的办公室，例如使用传输层安全（TLS）。当你与互联网通信并发送IP数据包时，你不是正常发送，而是在这个VPN连接内发送。因此，IP数据包会进入你的办公室网络。在这一点上，办公室网络可以正常路由它们。这让你可以做一些事情，如访问办公室里的私人、受保护的网路资源。与其到处撒网保护，你只需小心对待一项服务，即让人们通过虚拟私人网络登录网络的服务。你用一个网关来做这件事，它是一台接受来自允许的VPN客户端的连接并将其流量转发到私人网络的计算机。

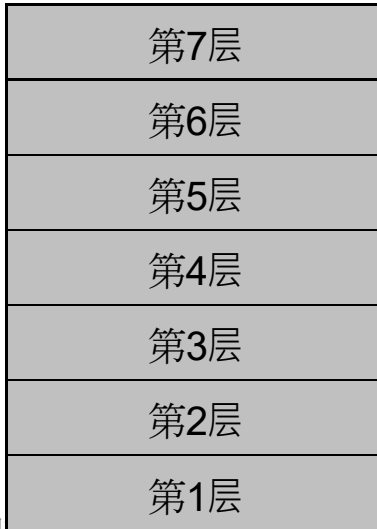
那么，这看起来像什么呢？比方说，我正在访问我的公司内部网站。那么，我的网络浏览器产生一个HTTP GET。像往常一样，它把它放在一个TCP段里，然后把它放在一个IP包里，目的地是公司的内部网络服务器。但是，我的电脑并没有把这个IP包放在一个链路层的框架内--我不能直接与内部网络服务器进行通信--而是把这个IP包放在一个TLS段内。

TLS保护信息并使其保密。这个TLS会话是在一个TCP流内，在虚拟私人网络网关处终止。所以

*外侧的TCP段在一个以虚拟专用网络网关为目的地的IP包内。我们把这个外部IP包放在一个链接帧内，并把它发送到下一跳。所以它看起来像这样。

HTTP里面的TCP里面的IP里面的TLS里面的TCP里面的IP里面的以太网。

封装



- 分层如何体现在数据表示中
- 封装的有效载荷
 - 帮助分离关切
 - 帮助执行边界/分层
 - 简化层的实施

尼克：现在你已经听说了封装，这是统一分层和分组交换的原则。封装是指我们如何将协议层以一种灵活的方式组装成数据包，并保持它们之间的分离。你看到了一个计算机可以封装一个网络请求的例子，以及一个如何以更复杂的方式使用封装的例子，如虚拟私人网络。