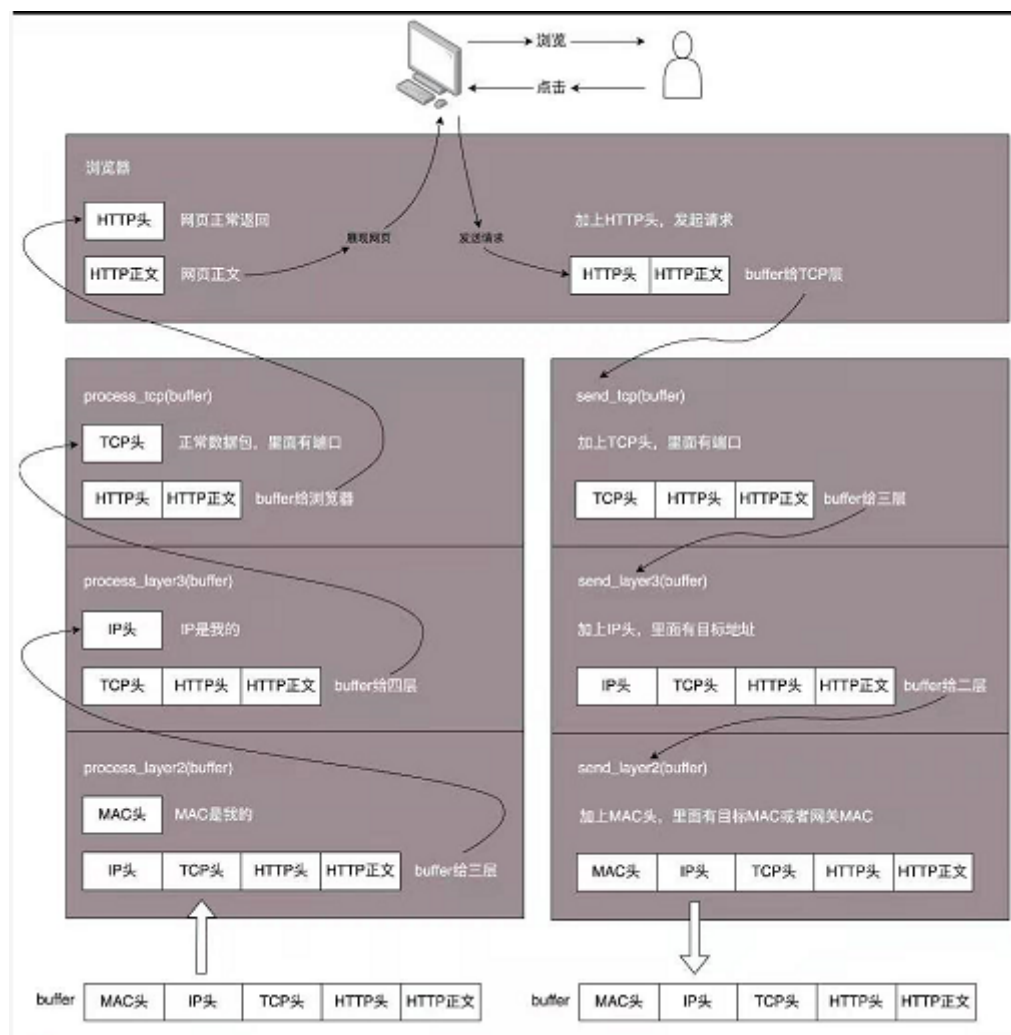


网络为什么要分层

网络为什么要分层？因为，是个复杂的程序都要分层。

可以这样理解网络分层：想象网络包就是一段 Buffer，或者一块内存，是有格式的。同时想象自己是一个处理网络包的程序，而且这个程序可以跑在电脑上，服务器上，交换机上，也可以跑在路由器上。可以想象自己有很多的网口，从某个口拿进一个网络包来，用自己的程序处理以下，再从另一个网口发送出去。

程序是如何工作的？



当一个网络包从一个网口经过的时候，你看到了，首先看看要不要请进来，处理一把。有的网口配置了混杂模式，凡是经过的，全部拿进来。

拿进来后，需要交给程序处理，于是调用 `process_layer2(buffer)`。从 `buffer` 中，摘掉二层的头，看一看应该根据里面的内容做什么操作。

假设发现这个包的 MAC 地址和你的相符，那说明就是发给你的，于是调用 `process_layer3(buffer)`。这个时候 `Buffer` 里已经没有二层的头了，因为在上一层处理的过程中拿

掉了。在这层，摘掉三层的头，看看是发送给自己的，还是希望转发出去的。

如果 IP 地址不是自己的，那就应该转发出去；如果 IP 地址是自己的就是发送给自己的。根据 IP 头里的标示，拿掉三层的头，进行下一层的处理。

假设这个地址是 TCP 的，则调用 `process_tcp(buffer)`。这时候，查看四层的头，看这是一个发起，还是一个应答，又或者是一个正常的数据包，然后分别由不同的逻辑进行处理。如果是发起或者应答，接下来可能要发送一个回复包；如果是一个正常的数据包，就需要交给上层了。

这个时候交给谁呢？是不是有 `process_http(buffer)` 函数呢？没有的，如果是一个网络处理程序，不需要有 `process_http(buffer)`，而是应该交给应用去处理，交给那个应用呢？在四层的头里面有端口号，不同的应用监听不同的端口号，如果发现是浏览器在监听这个端口，那发送给浏览器就行了，至于浏览器怎么处理，不用关心。

浏览器自然是解析 HTML，显示出页面来，电脑主人看见页面很开心，就点了鼠标，点击动作被浏览器捕获，浏览器知道，又要发起另一个 HTTP 请求了，于是使用端口号，将请求发送给了你。

你调用 `send_tcp(buffer)`，`buffer` 就是 HTTP 请求的内容，这个函数里面加一个 TCP 的头，记录下源端口号。浏览器会给你目的端口号，一般为 80 端口。

然后调用 `send_layer3(buffer)`。`buffer` 里面已经有了 Http 的头和内容，以及 TCP 的头。在这个函数里面加一个 IP 的头，记录下源 IP 的地址和目标 IP 的地址。

然后调用 `send_layer2(buffer)`，`buffer` 里已经有了 HTTP 的头和内容，TCP 的头，以及 IP 的头。这个函数里要加一下 MAC 的头，记录下源 MAC 地址，得到的就是本机器的 MAC 地址和目标的 MAC 地址。不过这个还要看当前知道不知道，知道就直接加上，不知道的话就需要通过一定的协议处理过程，找到 MAC 地址。

准备好了，只要 `buffer` 的内容完整，就可以从网口发出去了，你作为一个程序的任务就告一段落。