

## 第1讲 | 为什么要学习网络协议？

笔记本： P.趣谈网络协议

创建时间： 2018/5/18 9:27

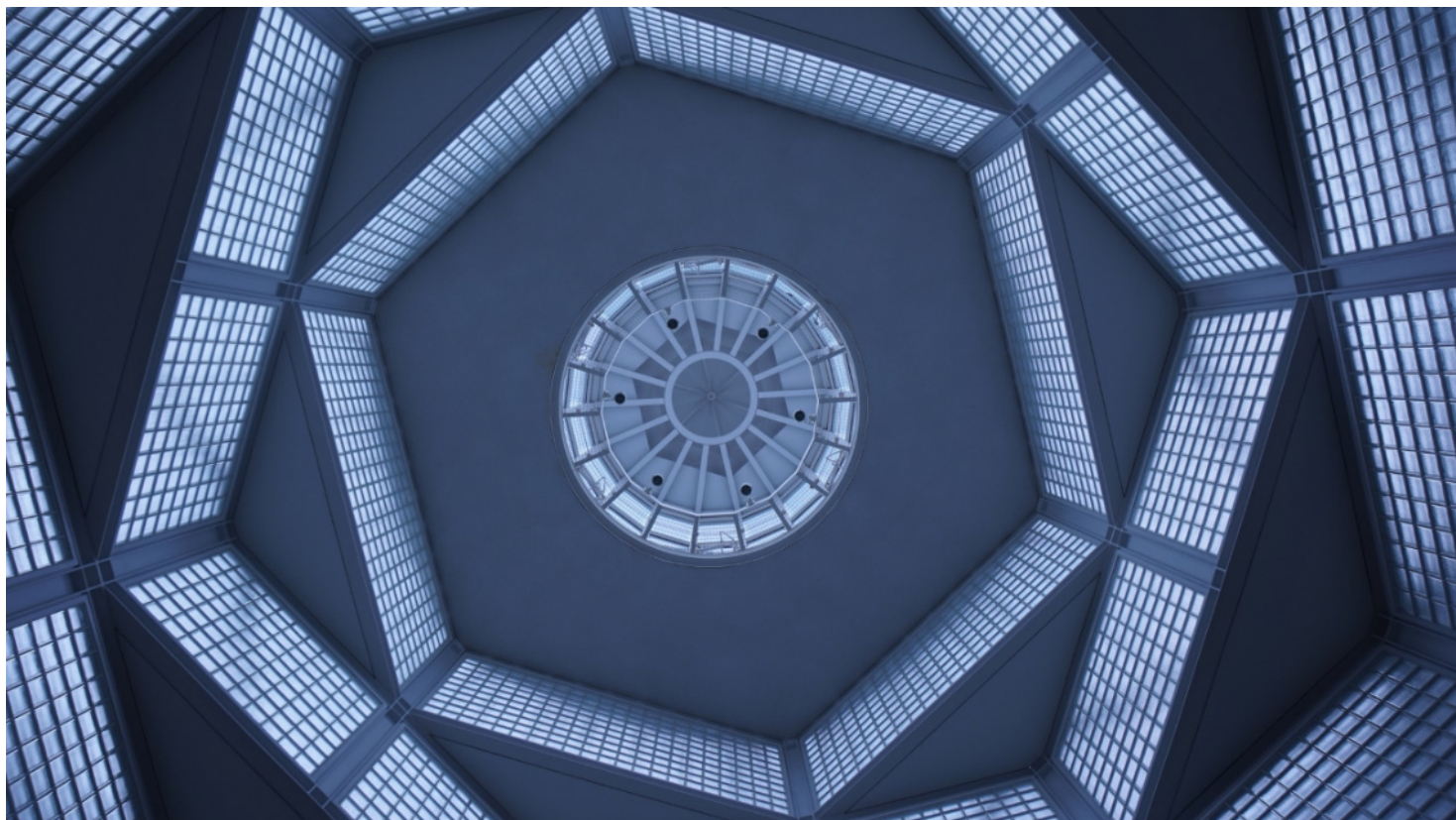
更新时间： 2018/5/18 9:28

作者： hongfenghuoju

URL：

## 第1讲 | 为什么要学习网络协议？

2018-05-18 刘超



《圣经》中有一个通天塔的故事，大致是说，上帝为了阻止人类联合起来，就让人类说不同的语言。人类没法儿沟通，达不成“协议”，通天塔的计划就失败了。

但是千年以后，有一种叫“程序猿”的物种，敲着一种这个群体通用的语言，连接着全世界所有的人，打造这互联网世界的通天塔。如今的世界，正是因为互联网，才连接在一起。

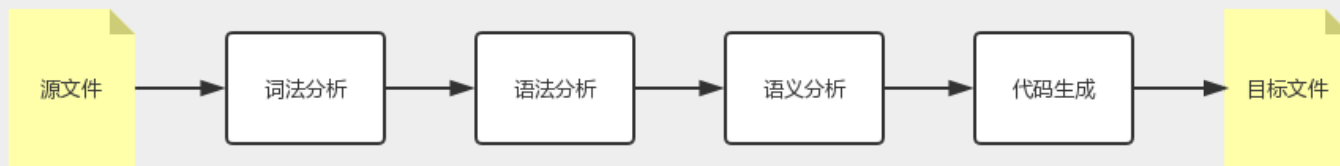
当 "Hello World!" 从显示器打印出来的时候，还记得你激动的心情吗？

```
public class HelloWorld {  
    public static void main(String[] args){  
        System.out.println("Hello World!");  
    }  
}
```

如果你是程序员，一定看得懂上面这一段文字。这是每一个程序员向计算机世界说“你好，世界”的方式。但是，你不一定知道，这段文字也是一种协议，是人类和计算机沟通的协议，只有通过这种协议，计算机才知道我们想让它做什么。

协议三要素

当然，这种协议还是更接近人类语言，机器不能直接读懂，需要进行翻译，翻译的工作教给编译器，也就是程序员常说的 compile。这个过程比较复杂，其中的编译原理非常复杂，我在这里不进行详述。



但是可以看得出，计算机语言作为程序员控制一台计算机工作的协议，具备了协议的三要素。

- 语法，就是这一段内容要符合一定的规则和格式。例如，括号要成对，结束要使用分号等。
- 语义，就是这一段内容要代表某种意义。例如数字减去数字是有意义的，数字减去文本一般来说就没有意义。
- 顺序，就是先干啥，后干啥。例如，可以先加上某个数值，然后再减去某个数值。

会了计算机语言，你就能够教给一台计算机完成你的工作了。恭喜你，入门了！

但是，要想打造互联网世界的通天塔，只教给一台机器做什么是不够的，你需要学会教给一大片机器做什么。这就需要网络协议。只有通过网络协议，才能使一大片机器互相协作、共同完成一件事。

这个时候，你可能会问，网络协议长啥样，这么神奇，能干成啥事？我先拿一个简单的例子，让你尝尝鲜，然后再讲一个大事。

当你想要买一个商品，常规的做法就是打开浏览器，输入购物网站的地址。浏览器就会给你显示一个缤纷多彩的页面。

那你有没有深入思考过，浏览器是如何做到这件事情的？它之所以能够显示缤纷多彩的页面，是因为它收到了一段来自 HTTP 协议的“东西”。我拿网易考拉来举例，格式就像下面这样：

```
HTTP/1.1 200 OK
Date: Tue, 27 Mar 2018 16:50:26 GMT
Content-Type: text/html; charset=UTF-8
Content-Language: zh-CN

<!DOCTYPE html>
<html>
<head>
<base href="https://pages.kaola.com/" />
<meta charset="utf-8"/> <title> 网易考拉 3 周年主会场 </title>
```

这符合协议的三要素吗？我带你来看一下。

首先，符合语法，也就是说，只有按照上面那个格式来，浏览器才认。例如，上来是状态，然后是首部，然后是内容。

第二，符合语义，就是要按照约定的意思来。例如，状态 200，表述的意思是网页成功返回。如果不成功，就是我们常见的“404”。

第三，符合顺序，你一点浏览器，就是发送出一个 HTTP 请求，然后才有上面那一串 HTTP 返回的东西。

浏览器显然按照协议商定好的做了，最后一个五彩缤纷的页面就出现在你面前了。

我们常用的网络协议有哪些？

接下来揭秘我要说的大事情，“双十一”。这和我们要讲的网络协议有什么关系呢？

在经济学领域，有个伦纳德·里德（Leonard E. Read）创作的《铅笔的故事》。这个故事通过一个铅笔的诞生过程，来讲述复杂的经济学理论。这里，我也用一个下单的过程，看看互联网世界的运行过程中，都使用了哪些网络协议。

你先在浏览器里面输入 <https://www.kaola.com>，这是一个URL。浏览器只知道名字是“www.kaola.com”，但是不知道具体的地点，所以不知道应该如何访问。于是，它打开地址簿去查找。可以使用一般的地址簿协议DNS去查找，还可以使用另一种更加精准的地址簿查找协议HTTPDNS。

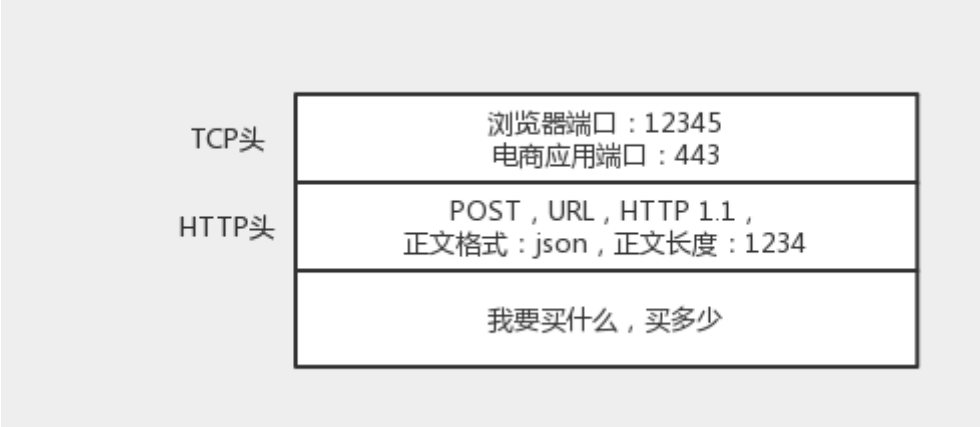
无论用哪一种方法查找，最终都会得到这个地址：106.114.138.24。这个是IP地址，是互联网世界的“门牌号”。

知道了目标地址，浏览器就开始打包它的请求。对于普通的浏览请求，往往会使用HTTP协议；但是对于购物的请求，往往需要进行加密传输，因而会使用HTTPS协议。无论是什么协议，里面都会写明“你要买什么和买多少”。



DNS、HTTP、HTTPS 所在的层我们称为应用层。经过应用层封装后，浏览器会将应用层的包交给下一层去完成，通过 socket 编程来实现。下一层是传输层。传输层有两种协议，一种是无连接的协议UDP，一种是面向连接的协议TCP。对于支付来讲，往往使用 TCP 协议。所谓的面向连接就是，TCP 会保证这个包能够到达目的地。如果不能到达，就会重新发送，直至到达。

TCP 协议里面会有两个端口，一个是浏览器监听的端口，一个是电商的服务器监听的端口。操作系统往往通过端口来判断，它得到的包应该给哪个进程。



传输层封装完毕后，浏览器会将包交给操作系统的网络层。网络层的协议是 IP 协议。在 IP 协议里面会有源 IP 地址，即浏览器所在机器的 IP 地址和目标 IP 地址，也即电商网站所在服务器的 IP 地址。

IP头	客户端电脑IP : 192.168.1.101 电商服务器IP : 106.114.138.24
TCP头	浏览器端口 : 12345 电商应用端口 : 443
HTTP头	POST , URL , HTTP 1.1 , 正文格式 : json , 正文长度 : 1234
	我要买什么 , 买多少

操作系统既然知道了目标 IP 地址，就开始想如何根据这个门牌号找到目标机器。操作系统往往会判断，这个目标 IP 地址是本地人，还是外地人。如果是本地人，从门牌号就能看出来，但是显然电商网站不在本地，而在遥远的地方。

操作系统知道要离开本地去远方。虽然不知道远方在何处，但是可以这样类比一下：如果去国外要去海关，去外地就要去网关。而操作系统启动的时候，就会被 DHCP 协议配置 IP 地址，以及默认的网关的 IP 地址 192.168.1.1。

操作系统如何将 IP 地址发给网关呢？在本地通信基本靠吼，于是操作系统大吼一声，谁是 192.168.1.1 啊？网关会回答它，我就是，我的本地地址在村东头。这个本地地址就是 MAC 地址，而大吼的那一声是 ARP 协议。

MAC头	客户端电脑MAC : 192.168.1.101的MAC 网关的MAC : 192.168.1.1的MAC
IP头	客户端电脑IP : 192.168.1.101 电商服务器IP : 106.114.138.24
TCP头	浏览器端口 : 12345 电商应用端口 : 443
HTTP头	POST , URL , HTTP 1.1 , 正文格式 : json , 正文长度 : 1234
	我要买什么 , 买多少

于是操作系统将 IP 包交给了下一层，也就是 MAC 层。网卡再将包发出去。由于这个包里面是有 MAC 地址的，因而它能够到达网关。

网关收到包之后，会根据自己的知识，判断下一步应该怎么走。网关往往是一个路由器，到某个 IP 地址应该怎么走，这个叫作路由表。

路由器有点像玄奘西行路过的一个个国家的一个个城关。每个城关都连着两个国家，每个国家相当于一个局域网，在每个国家内部，都可以使用本地的地址 MAC 进行通信。

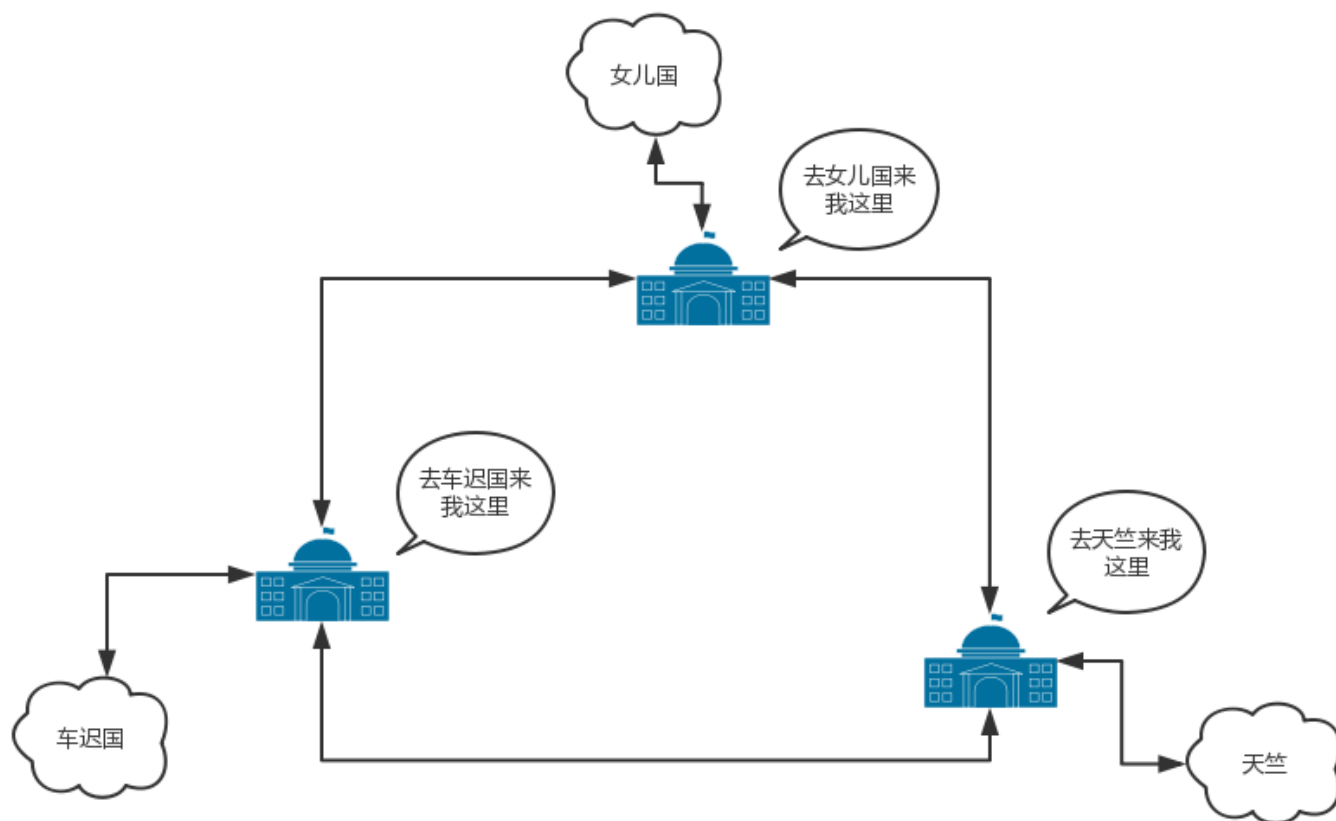
一旦跨越城关，就需要拿出 IP 头来，里面写着贫僧来自东土大唐（就是源 IP 地址），欲往西天拜佛求经（指的是目标 IP 地址）。路过宝地，借宿一晚，明日启行，请问接下来该怎么走啊？



去IP段1，走网口1，下一跳为路由器A
去IP段2，走网口2，下一跳为路由器B
去IP段3，走网口3，下一跳为路由器C

去女儿国，走左面的路，下一站为女儿国东门
去车迟国，走中间的路，下一站为车迟国南门
去天竺，走右面的路，下一站为天竺西门

城关往往是知道这些“知识”的，因为城关和临近的城关也会经常沟通。到哪里应该怎么走，这种沟通的协议称为**路由协议**，常用的有**OSPF**和**BGP**。



城关与城关之间是一个国家，当网络包知道了下一步去哪个城关，还是要使用国家内部的 MAC 地址，通过下一个城关的 MAC 地址，找到下一个城关，然后再问下一步的路怎么走，一直到走出最后一个城关。

最后一个**城关**知道这个网络包要去的地方。于是，对着这个国家吼一声，谁是目标 IP 啊？目标服务器就会回复一个 MAC 地址。网络包过关后，通过这个 MAC 地址就能找到目标服务器。

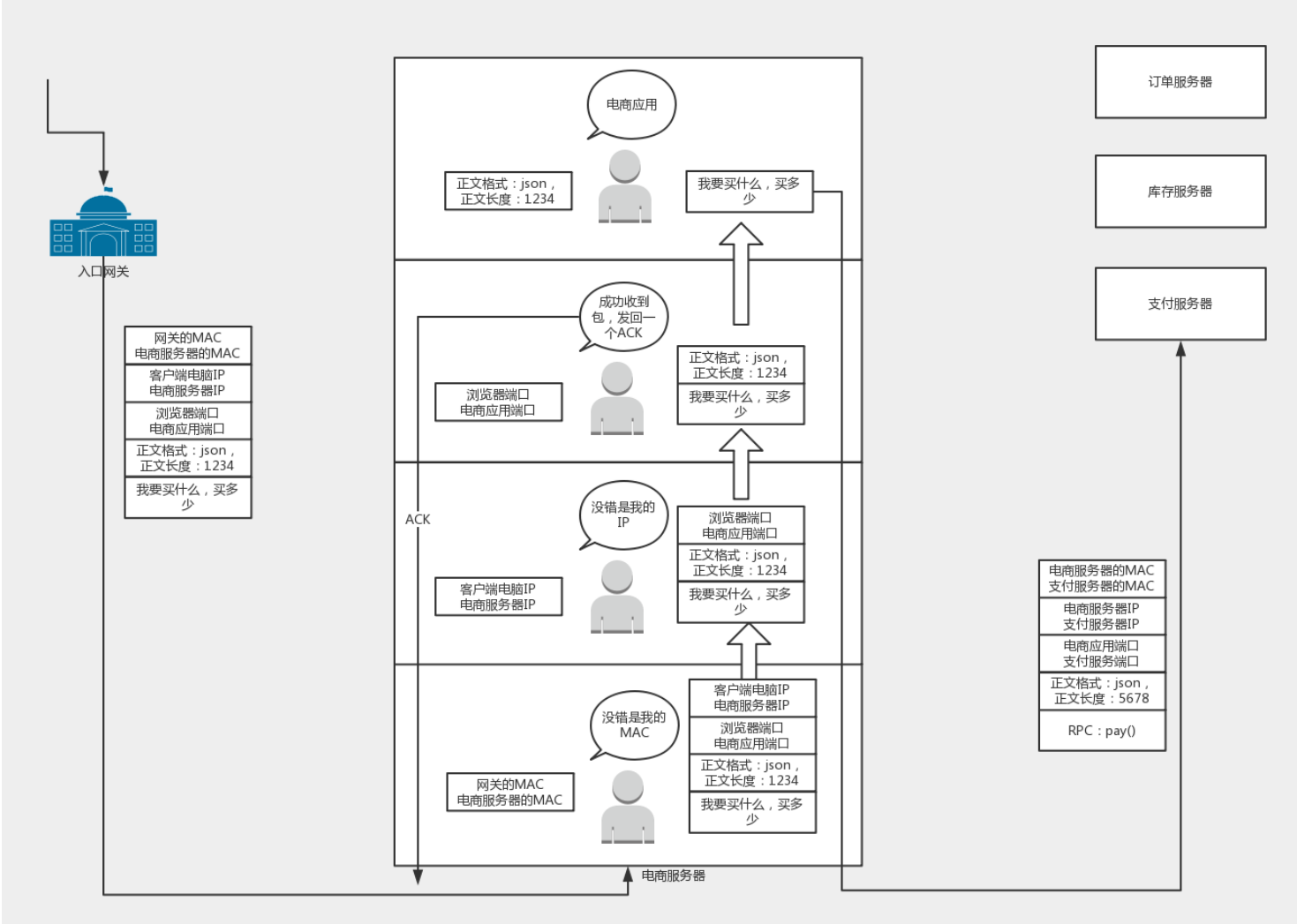
目标服务器发现 MAC 地址对上了，取下 MAC 头来，发送给**操作系统的网络层**。发现 IP 也对上了，就取下 IP 头。IP 头里会写上一层封装的是 TCP 协议，然后将其交给传输层，即 TCP 层。

在这一层里，对于收到的每个包，都会有一个回复的包说明收到了。这个回复的包绝非这次下单请求的结果，例如购物是否成功，扣了多少钱等，而仅仅是 TCP 层的一个说明，即收到之后的回复。当然这个回复，会沿着刚才来的方向走回去，报个平安。

因为一旦出了国门，西行路上千难万险，如果在这个过程中，网络包走丢了，例如进了大沙漠，或者被强盗抢劫杀害怎么办呢？因而到了要报个平安。

如果过一段时间还是没到，发送端的 TCP 层会重新发送这个包，还是上面的过程，直到有一天收到平安到达的回复。这个重试绝非你的浏览器重新将下单这个动作重新请求一次。对于浏览器来讲，就发送了一次下单请求，TCP 层不断自己闷头重试。除非 TCP 这一层出了问题，例如连接断了，才轮到浏览器的应用层重新发送下单请求。

当网络包平安到达 TCP 层之后，TCP 头中有目标端口号，通过这个端口号，可以找到电商网站的进程正在监听这个端口号，假设一个 Tomcat，将这个包发给电商网站。



电商网站的进程得到 HTTP 请求的内容，知道了要买东西，买多少。往往一个电商网站最初接待请求的这个 Tomcat 只是个接待员，负责统筹处理这个请求，而不是所有的事情都自己做。例如，这个接待员要告诉专门管理订单的进程，登记要买某个商品，买多少，要告诉管理库存的进程，库存要减少多少，要告诉支付的进程，应该付多少钱，等等。

如果告诉相关的进程呢？往往通过 RPC 调用，即远程过程调用的方式来实现。远程过程调用就是当告诉管理订单进程的时候，接待员不用关心中间的网络互连问题，会由 RPC 框架统一处理。RPC 框架有很多种，有基于 HTTP 协议放在 HTTP 的报文里面的，有直接封装在 TCP 报文里面的。

当接待员发现相应的部门都处理完毕，就回复一个 HTTPS 的包，告知下单成功。这个 HTTPS 的包，会像来的时候一样，经过千难万险到达你的个人电脑，最终进入浏览器，显示支付成功。

小结

看到了吧，一个简简单单的下单过程，中间牵扯到这么多的协议。而管理一大片机器，更是一件特别有技术含量的事情。除此之外，像最近比较火的云计算、容器、微服务等技术，也都需要借助各种协议，来达成大规模机器之间的合作。

我在这里列一下之后要讲的网络协议，之后我会按照从底层到上层的顺序来讲述。

极客时间

# 趣谈网络协议

像小说一样的网络协议入门课



扫码订阅《趣谈网络协议》

应用层	DHCP	HTTP	HTTPS	RTMP	P2P	DNS	GTP	RPC
传输层	UDP	TCP						
网络层	ICMP	IP	OSPF	BGP	IPSec	GRE		
链路层	ARP	VLAN	STP					
物理层	网络跳线							

上面的“双十一”故事只是为了给你一个大致框架，这里面有些协议，我在故事里已经提到了，有些还没有提到。在这门课的最后一章，当所有的协议都讲过之后，我会再重新讲一遍这个故事，到时候你就能明白更多的细节。

最后，学完了这一节，给你留一个问题吧。

当网络包到达一个城关的时候，可以通过路由表得到下一个城关的 IP 地址，直接通过 IP 地址找就可以了，为什么还要通过本地的 MAC 地址呢？

欢迎你留言和我讨论。趣谈网络协议，我们下期见！