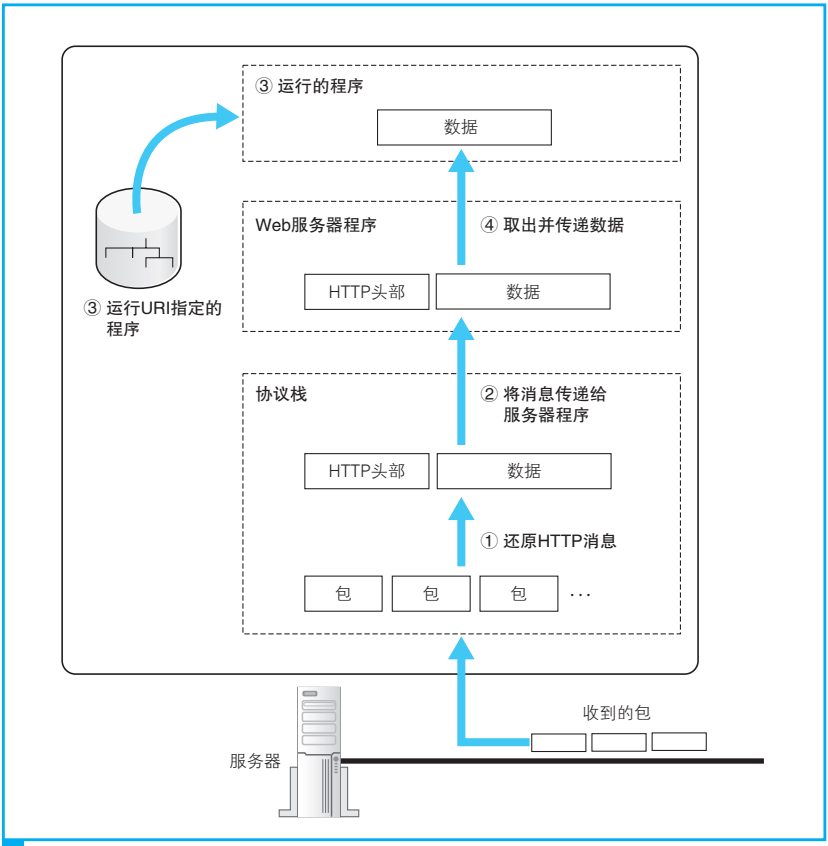


图 6.11　数据如何传递给 Web 服务器上运行的程序
Web 服务器程序在组装网络包、还原数据之后，会运行其中指定的程序（实际是委托操作系统来运行），然后将数据传递给已运行的程序。

器并不过问，也不会去改变程序输出的内容[①]。

6.3.3 Web 服务器的访问控制

正如我们前面讲的，Web 服务器的基本工作方式就是根据请求消息的内容判断数据源，并从中获取数据返回给客户端，不过在执行这些操作之

① 但可以添加一些 HTTP 消息的头部字段。

前，Web 服务器还可以检查事先设置的一些规则，并根据规则允许或禁止访问。这种根据规则判断是否允许访问的功能称为访问控制，一些会员制的信息服务需要限制用户权限的时候会使用这一功能，公司里也可以利用访问控制只允许某些特定部门访问。

Web 服务器的访问控制规则主要有以下 3 种。

（1）客户端 IP 地址
（2）客户端域名
（3）用户名和密码

以上规则可针对作为数据源的文件和目录进行设置，当收到客户端的请求消息时，服务器会根据 URI 判断数据源，并检查数据源对应的访问控制规则，只有允许访问时才读取文件或运行程序。下面我们来看一下设定访问控制规则时，服务器是如何工作的。

首先是根据客户端 IP 地址设置的规则，这个情况很简单，在调用 accept 接受连接时，就已经知道客户端的 IP 地址了，只要检查其是否允许访问就可以了。

当根据客户端域名设置规则时，需要先根据客户端 IP 地址查询客户端域名，这需要使用 DNS 服务器。一般我们使用 DNS 服务器都是根据域名查询 IP 地址，其实根据 IP 地址反查域名也可以使用 DNS 服务器。具体来说，这个过程是这样的。收到客户端的请求消息后，Web 服务器（图 6.12 ①）会委托协议栈告知包的发送方 IP 地址，然后用这个 IP 地址生成查询消息并发送给最近的 DNS 服务器（图 6.12 ②）。接下来，DNS 服务器找出负责管辖该 IP 地址的 DNS 服务器，并将查询转发给它（图 6.12 ③），查询到相应的域名之后返回结果（图 6.12 ④），然后 Web 服务器端的 DNS 服务器再将结果转发给 Web 服务器（图 6.12 ⑤）。这样一来，我们就可以根据发送方 IP 地址查询到域名。接下来，为了保险起见，还需要用这个域名查询一下 IP 地址，看看结果与发送方 IP 地址是否一致（图 6.12 ⑥）。这是因为有一种在 DNS 服务器上注册假域名的攻击方式，因此我们需要进行双重检查，

如果两者一致则检查相应的访问控制规则，判断是否允许访问。从图 6.12 中可以看出，这种方式需要和 DNS 服务器进行多次查询，整个过程比较耗时，因此 Web 服务器的响应速度也会变慢。

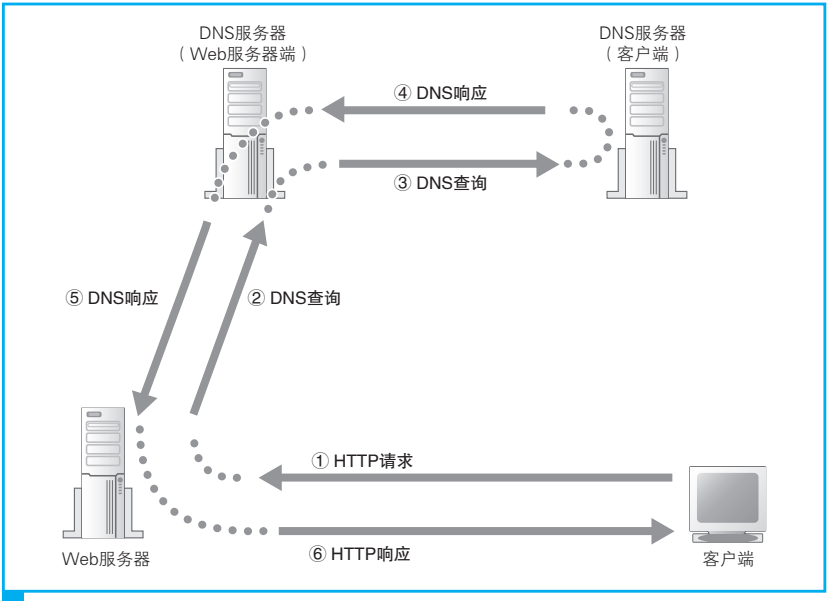


图 6.12　根据域名进行访问控制

如果用户名和密码已设置好，那么情况如图 6.13[①]。通常的请求消息中不包含用户名和密码，因此无法验证用户名和密码（图 6.13 ①）。因此，Web 服务器会向用户发送一条响应消息，告诉用户需要在请求消息中放入用户名和密码（图 6.13 ②）。浏览器收到这条响应消息后，会弹出一个输入用户名和密码的窗口，用户输入用户名和密码后（图 6.13 ③），浏览器将这

① 这里介绍的内容是使用 Web 服务器提供的密码认证功能时的工作过程，除此之外，还可以通过 Web 服务器运行 CGI 认证程序来验证密码。这种情况下，认证程序会生成一个含有密码表单的网页并发送给用户，用户填写密码后发送回服务器，由认证程序进行校验。这种方式会包含密码表单页面和用户提交的密码数据的交互过程，因此和图 6.13 的过程是有区别的。

些信息放入请求消息中重新发送给服务器（图 6.13 ④）。然后，Web 服务器查看接收到的用户名和密码与事先设置好的用户名和密码是否一致，以此判断是否允许访问，如果允许访问，则返回数据（图 6.13 ⑤）。

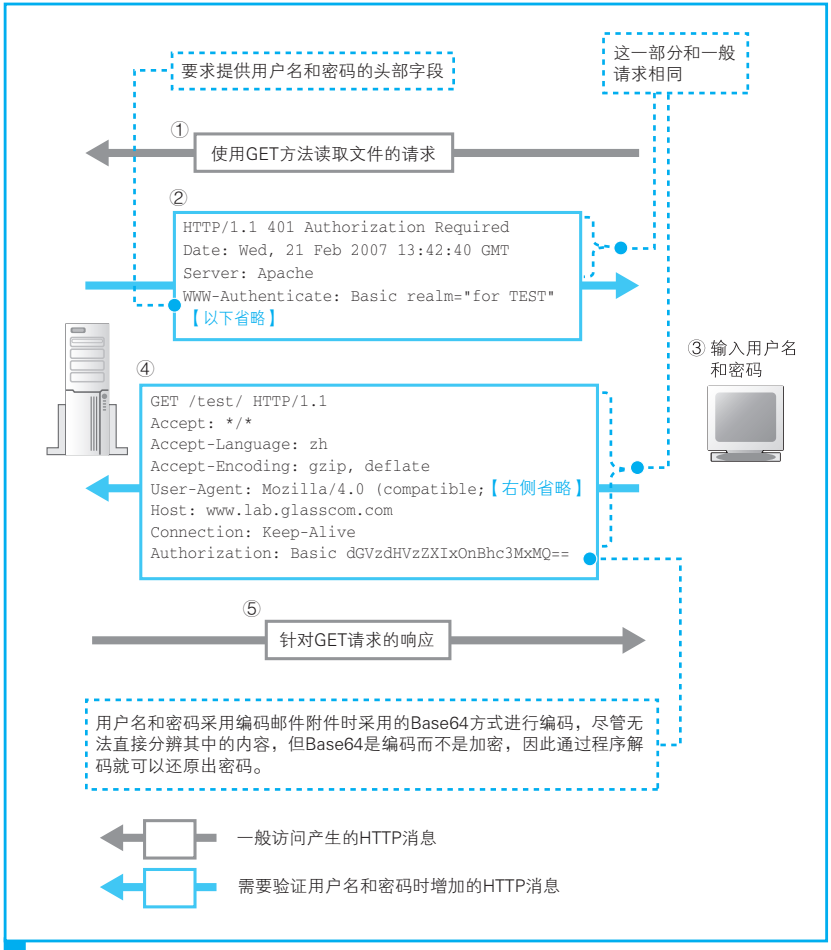


图 6.13　利用 HTTP 验证用户名和密码

当访问设置了用户名和密码保护的页面时，需要在 HTTP 请求消息中添加包含用户名和密码的头部字段（Authorization）。否则，Web 服务器不会返回请求的页面内容，而是会返回一个要求提供用户名和密码的头部字段（WWW-Authenticate）消息。

6.3.4 返回响应消息

当服务器完成对请求消息的各种处理之后，就可以返回响应消息了。这里的工作过程和客户端向服务器发送请求消息时的过程相同。

首先，Web 服务器调用 Socket 库的 write，将响应消息交给协议栈。这时，需要告诉协议栈这个响应消息应该发给谁，但我们并不需要直接告知客户端的 IP 地址等信息，而是只需要给出表示通信使用的套接字的描述符就可以了。套接字中保存了所有的通信状态，其中也包括通信对象的信息，因此只要有描述符就万事大吉了。

接下来，协议栈会将数据拆分成多个网络包，然后加上头部发送出去。这些包中包含接收方客户端的地址，它们将经过交换机和路由器的转发，通过互联网最终到达客户端。

6.4 浏览器接收响应消息并显示内容

6.4.1 通过响应的数据类型判断其中的内容

Web 服务器发送的响应消息会被分成多个包发送给客户端，然后客户端需要接收数据。首先，网卡将信号还原成数字信息，协议栈将拆分的网络包组装起来并取出响应消息，然后将消息转交给浏览器。这个过程和服务器的接收操作相同。接下来，我们来看一看浏览器是如何显示内容的。

要显示内容，首先需要判断响应消息中的数据属于哪种类型。Web 可以处理的数据包括文字、图像、声音、视频等多种类型，每种数据的显示方法都不同，因此必须先要知道返回了什么类型的数据，否则无法正确显示。

这时，我们需要一些信息才能判断数据类型，原则上可以根据响应消息开头的 Content-Type 头部字段的值来进行判断。这个值一般是下面这样的字符串。

```
Content-Type: text/html
```

其中“/”左边的部分称为“主类型”，表示数据的大分类；右边的“子类型”表示具体的数据类型。在上面的例子中，主类型是 text，子类型是 html。主类型和子类型的含义都是事先确定好的[①]，表 6.1 列出了其中主要的一些类型。上面例子中的数据类型表示遵循 HTML 规格的 HTML 文档。

表 6.1　消息的 Content–Type 定义的数据类型

主类型	含　义	子类型示例	
text	表示文本数据	text/html	HTML 文档
		text/plain	纯文本
image	表示图像数据	image/jpeg	JPEG 格式的图片
		image/gif	GIF 格式的图片
audio	表示音频数据	audio/mpeg	MP2、MP3 格式的音频
video	表示视频数据	video/mpeg	MPEG 格式的视频
		video/quicktime	Quicktime 格式的视频
model	表示对物体等的形状和动作进行建模的数据	model/vrml	VRML 格式的建模数据
application	除上述以外的数据，Excel、Word 等应用程序的数据都属于这一类型	application/pdf	PDF 格式的文档数据
		application/msword	MS-WORD 格式的文档数据
message	直接存放邮件等消息时使用的类型，表示直接存放某种格式的消息	message/rfc822	一般的邮件数据，包含 From:、Date: 等头部数据
multipart	消息体中包含多个部分的数据	multipart/mixed	消息体中包含各种不同格式的数据，其中每个部分的数据都有单独定义的媒体类型

此外，当数据类型为文本时，还需要判断编码方式，这时需要用 charset 附加表示文本编码方式的信息，内容如下。

① 类型的含义和公有 IP 地址、端口号一样，都是全世界统一管理的。

```
Content-Type: text/html; charset=utf-8
```

这里的 utf-8 表示编码方式为 Unicode，如果是 euc-jp 就表示 EUC 编码，iso-2022-jp 表示 JIS 编码，shift_jis 表示 JIS 编码[①]。

除了通过 Content-Type 判断数据类型，还需要检查 Content-Encoding 头部字段。如果消息中存放的内容是通过压缩或编码技术对原始数据进行转换得到的，那么 Content-Encoding 的值就表示具体的转换方式，通过这个字段的值，我们可以知道如何将消息中经过转换的数据还原成原始数据。

Content-Type 字段使用的表示数据类型的方法是在 MIME[②] 规格中定义的，这个规格不仅用于 Web，也是邮件等领域中普遍使用的一种方式。不过这种方式也只不过是一种原则性的规范，要通过 Content-Type 准确判断数据类型，就需要保证 Web 服务器正确设置 Content-Type 的值，但现实中并非总是如此。如果 Web 服务器管理员不当心，就可能会因为设置错误导致 Content-Type 的值不正确。因此，根据原则检查 Content-Type 并不能确保总是能够准确判断数据类型。

因此，有时候我们需要结合其他一些信息来综合判断数据类型，例如请求文件的扩展名、数据内容的格式等。比如，我们可以检查文件的扩展名，如果为 .html 或 .htm 则看作是 HTML 文件，或者也可以检查数据的内容，如果是以 <html> 开头的则看作是 HTML 文档。不仅是 HTML 这样的文本文件，图片也是一样。图片是经过压缩的二进制数据，但其开头也有表示内容格式的信息，我们可以根据这些信息来判断数据的类型。不过，这部分的逻辑并没有一个统一的规格，因此不同的浏览器以及不同的版本都会有所差异。

---

① 中文常用的编码包括 gb2312、gbk、gb18030、big5 等。——译者注

② MIME：Multipurpose Internet Mail Extensions，多用途因特网邮件扩充。原本是为在电子邮件中附加图片和附件等非文本信息而制定的一种规格，后来在 Web 的领域也得到了广泛使用。

6.4.2　浏览器显示网页内容！访问完成！

判断完数据类型，我们离终点就只有一步之遥了。接下来只要根据数据类型调用用于显示内容的程序，将数据显示出来就可以了。对于 HTML 文档、纯文本、图片这些基本数据类型，浏览器自身具有显示这些内容的功能，因此由浏览器自身负责显示。

不同类型的数据显示操作的过程也不一样，我们以 HTML 文档为例来介绍。HTML 文档通过标签表示文档的布局和字体等样式信息，浏览器需要解释这些标签的含义，按照指定的样式显示文档的内容。实际的显示操作是由操作系统来完成的，浏览器负责对操作系统发出指令，例如在屏幕上的什么位置显示什么文字、使用什么样的字体等。

网页中还可以嵌入图片等数据，HTML 文档和图片等数据是分别存在在不同的文件中的，HTML 文档中只有表示图片引用的标签[①]。在读取文档数据时，一旦遇到相应的标签，浏览器就会向服务器请求其中的图片文件。这个请求过程和请求 HTML 文档的过程是一样的，就是在 HTTP 请求消息的 URI 中写上图片文件的文件名即可。将这个请求消息发送给 Web 服务器之后，Web 服务器就会返回图片数据了。接下来，浏览器会将图片嵌入到标签所在的位置。JPEG 和 GIF 格式的图片是经过压缩的，浏览器需要将其解压后委托操作系统进行显示。当然，为了避免图片和文字重叠，在显示文字的时候需要为图片留出相应的位置。

像 HTML 文档和图片等浏览器可自行显示的数据，就会按照上述方式委托浏览器在屏幕上显示出来。不过，Web 服务器可能还会返回其他一些类型的数据，如文字处理、幻灯片等应用程序的数据。这些数据无法由浏览器自行显示，这时浏览器会调用相应的程序。这些程序可以是浏览器的插件，也可以是独立的程序，无论如何，不同类型的数据对应不同的程序，这一对应关系是在浏览器中设置好的，只要按照这一对应关系调用相应的

① 扩展名为 .html 或 .htm 的文件中只包含 HTML 文档的数据。

程序，并将数据传递给它就可以了。然后，被调用的程序会负责显示相应的内容。

到这里，浏览器的显示操作就完成了，可以等待用户的下一个动作了。当用户点击网页中的链接，或者在网址栏中输入新的网址时，访问 Web 服务器的操作就又开始了。

小测验

本章的旅程告一段落，我们为大家准备了一些小测验题目，确认一下自己的成果吧。

问题

1. 在包收发操作中，服务器和客户端的区别是什么？
2. 当包到达服务器时，网卡会接收信号并通知 CPU，此时使用的机制叫什么？
3. Web 服务器可以同时处理多个客户端的访问，这里利用了操作系统的什么功能？
4. 当需要对 Web 服务器的访问进行限制的时候，可以根据哪些条件来判断是否允许访问？
5. Web 服务器返回的数据包括文档、图片等多种类型，客户端如何判断返回数据的不同类型？

Column
网络术语其实很简单

Gateway 是通往异世界的入口

探索队员：网关（gateway）有各种不同的种类呢。

探索队长：是啊。

队员：话说，gateway 这个词到底是什么意思啊？

队长：在问别人之前呢……

队员：我知道，我现在就查字典。唔，字典上说是墙上的像门一样的入口。

队长：没错，入口的里面是什么呢？

队员：里面？是什么呢？天堂？

队长：天堂……怎么说呢，不算对也不算错吧，总之，入口的里面是和外面不一样的世界。

队员：噢……

队长：通往异世界的入口就是 gateway 啦。

队员：怎么感觉像问禅一样的……

队长：哪有。要不我们还是举个例子吧。Web 服务器有一种叫 CGI（Common Gateway Interface，通用网关接口）的功能，这又是什么东西呢？

队员：Web 服务器运行 CGI 程序，然后 CGI 程序处理用户发来的数据，对吧？

队长：从 Web 服务器的角度来看确实如此，但如果从客户端发送的消息的角度来看呢？

队员：消息首先会到达 Web 服务器。

队长：没错，然后呢？

队员：然后……会进入 Web 服务器中，接着又会进入 CGI 程序中，是吗？

队长：没错。准确来说，CGI 指的不是 CGI 程序本身，而是连接程序与 Web 服务器程序的接口规格。所以说，客户端发送的消息是通过 CGI 这样一个接口，从 Web 服务器程序进入 CGI 程序的。

队员：原来如此。那么这个接口就是通往 CGI 程序这个异世界的入口咯？

队长：看来你总算是有点长进了。除了 CGI 之外，还有其他一些通往异世界的入口，这些都叫 gateway。

队员：那么 TCP/IP 设置窗口中的默认网关也是一种 gateway 咯？

队长：这里的网关就是路由器的意思。

队员：那么为什么不叫默认路由器，而是叫默认网关呢？

队长：因为一开始并没有路由器这个词，那时候是管路由器叫网关的。从某种角度来说，路由器就是通向另一个网络的入口，所以默认网关的叫法也是那时候遗留下来的。

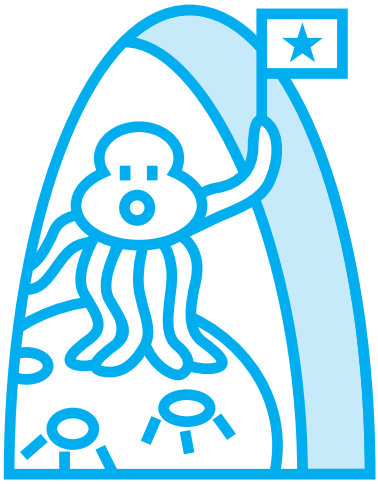
队员：这样啊，那后来为什么又出来路由器这个词了呢？

队长：以前，对于相当于路由器这样的东西，有很多不同的叫法。TCP/IP 中叫网关，TCP/IP 之外的路由器又有别的叫法。即便是现在，像交换机、集线器之类的叫法也不是很明确呢，是不是？

队员：这样下去可不行呢。

队长：我觉得也是。所以说，后来就统一叫路由器了。

队员：原来如此。那现在的交换机、



集线器之类的名字能不能也统一一下呢？

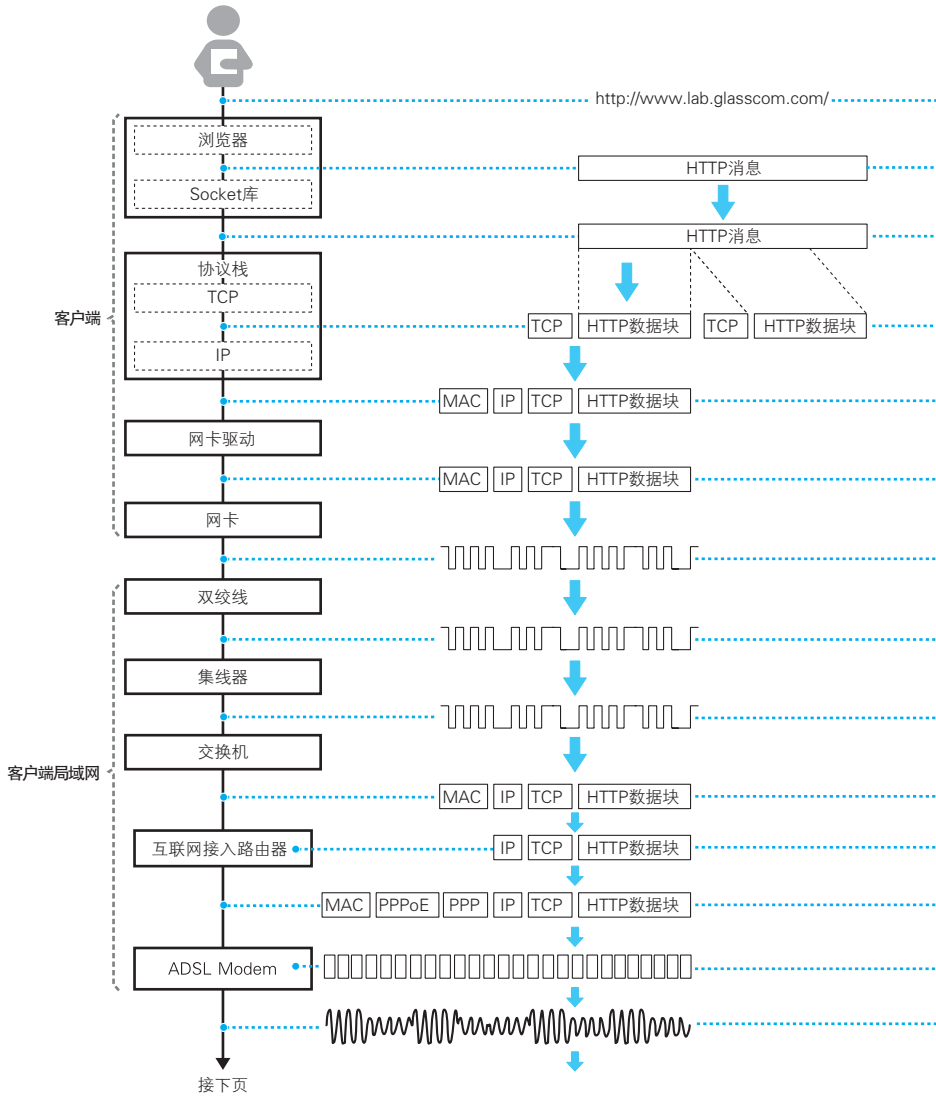
队长：跟我说也没用啊。

队员：别这样嘛，帮帮忙吧！

小测验答案

1. 没有区别（参见【6.1.1】）
2. 中断（参见【6.1.2】）
3. 多任务和多线程（参见【6.2.3】）
4. （a）客户端 IP 地址；（b）客户端域名；（c）用户名和密码（参见【6.3.3】）
5. 原则上根据响应消息的 Content-Type 头部字段的值来判断（参见【6.4.1】）

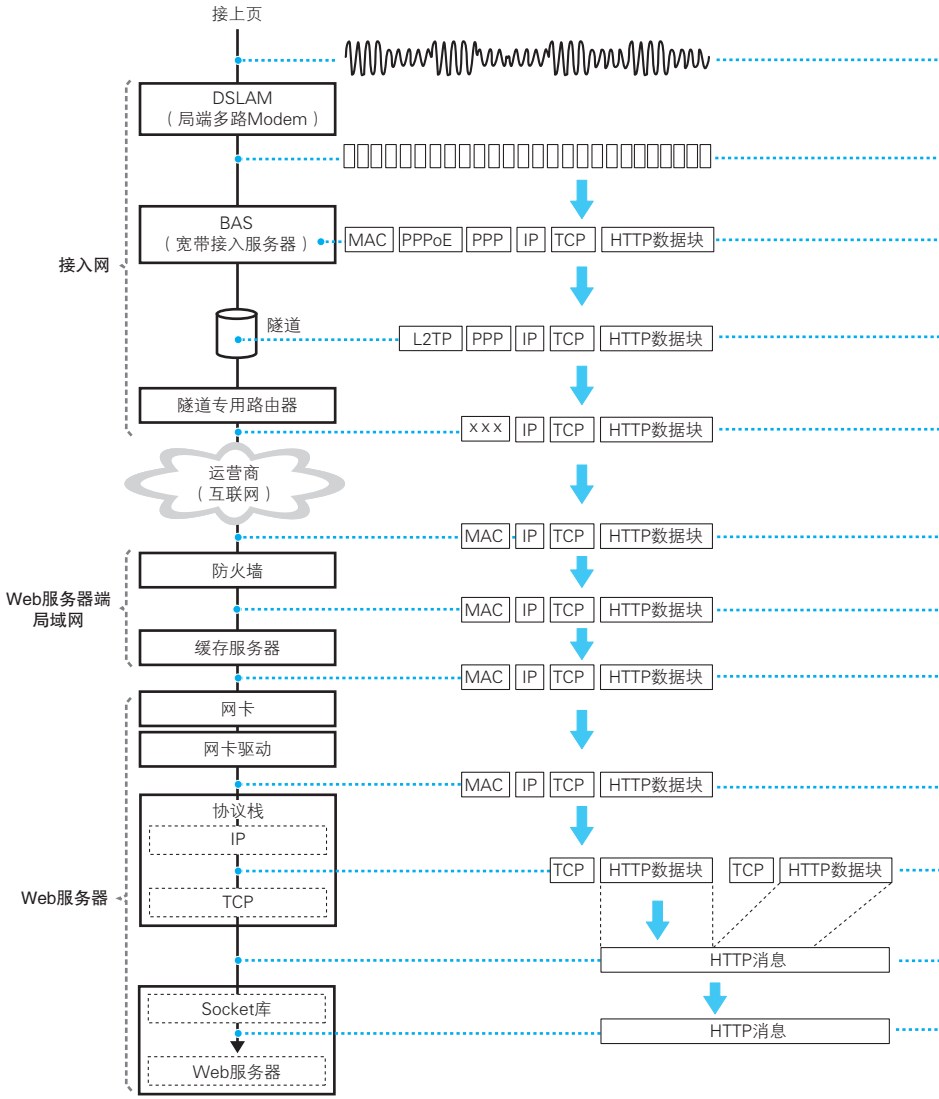
附录　网络包的旅程


http://www.lab.glasscom.com/
浏览器
Socket库
HTTP消息
HTTP消息
协议栈
TCP
IP
客户端
TCP
HTTP数据块
TCP
HTTP数据块
MAC
IP
TCP
HTTP数据块
网卡驱动
MAC
IP
TCP
HTTP数据块
网卡
双绞线
集线器
交换机
客户端局域网
MAC
IP
TCP
HTTP数据块
互联网接入路由器
IP
TCP
HTTP数据块
MAC
PPPoE
PPP
IP
TCP
HTTP数据块
ADSL Modem
接下页

如图，计算机生成的网络包其内容是不变的，但在传输到 Web 服务器的过程中，其形态会发生变化。

接收方MAC地址	接收方IP地址	概览	对应章节
		首先，用户输入网址	第1章
		浏览器解析网址，生成HTTP消息并转交给Socket库	
		Socket库将收到的HTTP消息作为数据转交给协议栈	
		TCP按照网络包的长度对数据进行拆分，在每个包前面加上TCP头部并转交给IP	第2章
最近的路由器的MAC地址	Web服务器的IP地址	IP在TCP包前面加上IP头部，然后查询MAC地址并加上MAC头部，然后将包转交给网卡驱动	
最近的路由器的MAC地址	Web服务器的IP地址	网卡驱动收到IP发来的包，将其转交给网卡并发出发送指令	
		网卡检查以太网的可发送状态，将包转换成电信号通过双绞线发送出去	
		信号通过双绞线到达集线器	第3章
		集线器将信号广播到所有端口，这样信号便到达交换机	
最近的路由器的MAC地址	Web服务器的IP地址	交换机根据收到的包的接收方MAC地址查询自身的地址表找到输出端口，并将包转发到输出端口	
	Web服务器的IP地址	互联网接入路由器根据收到的包的接收方IP地址查询自身的路由表找到输出端口，并将包转发到输出端口	第4章
BAS的MAC地址	Web服务器的IP地址	互联网接入路由器输出到互联网的包带有PPPoE头部和PPP头部	
		ADSL Modem将收到的包拆分成ATM信元	
		ADSL Modem将拆分后的 ATM 信元转换成电信号通过电话线发送出去	

网络包的旅程（续）


接上页
DSLAM
（局端多路Modem）
BAS
（宽带接入服务器）
接入网
MAC
PPPoE
PPP
IP
TCP
HTTP数据块
隧道
L2TP
隧道专用路由器
×××
运营商
（互联网）
防火墙
Web服务器端
局域网
缓存服务器
网卡
网卡驱动
协议栈
IP
TCP
Web服务器
HTTP消息
Socket库
Web服务器

接收方MAC地址	接收方IP地址	概览	对应章节
		ADSL Modem发送的信号经过电线杆上的电话线到达电话局的DSLAM（局端多路Modem）	第4章
		DSLAM将收到的电信号还原成ATM信元并发送给BAS	
	Web服务器的IP地址	BAS将ATM信元还原成网络包，根据接收方IP地址进行转发	
	Web服务器的IP地址	BAS转发的包被加上L2TP头部并通过隧道	
	Web服务器的IP地址	网络包到达位于隧道出口的隧道路由器，L2TP头部和PPP头部被丢弃，通过互联网流向Web服务器	
缓存服务器或者Web服务器的MAC地址	Web服务器的IP地址	服务器端的局域网中有防火墙，对进入的包进行检查，判断是否允许通过	第5章
缓存服务器或者Web服务器的MAC地址	Web服务器的IP地址	Web服务器前面如果有缓存服务器，会拦截通过防火墙的包。如果用户请求的页面已经缓存在服务器上，则代替服务器向用户返回页面数据	
Web服务器的MAC地址	Web服务器的IP地址	如果请求的页面没有被缓存，缓存服务器会将请求转发给Web服务器	
	Web服务器的IP地址	Web服务器收到包后，网卡和网卡驱动会接收这个包并转交给协议栈	第6章
		协议栈依次检查IP头部和TCP头部，如果没有问题则取出HTTP消息的数据块并进行组装	
		HTTP消息被恢复成原始形态，然后通过Socket库转交给Web服务器	
		Web服务器分析HTTP消息的内容，并根据请求内容将读取的数据返回给客户端	