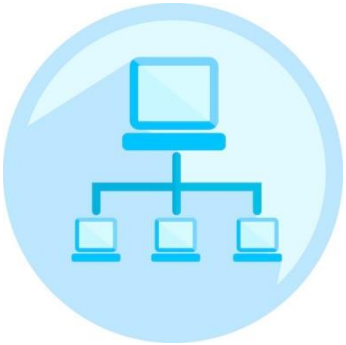
计算机网络

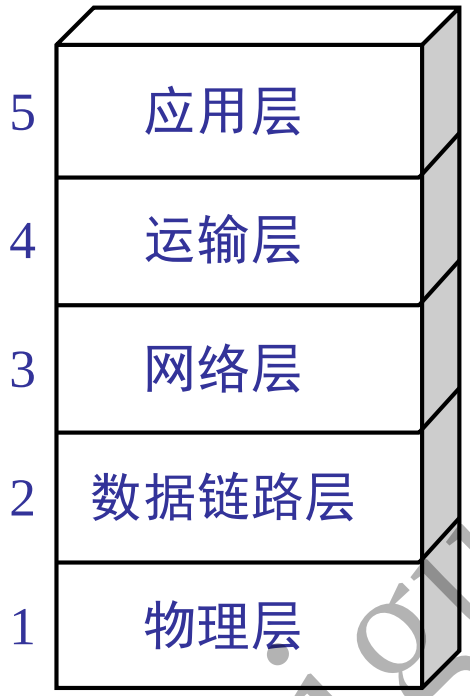
顾 军

计算机学院

jgu@cumt.edu.cn

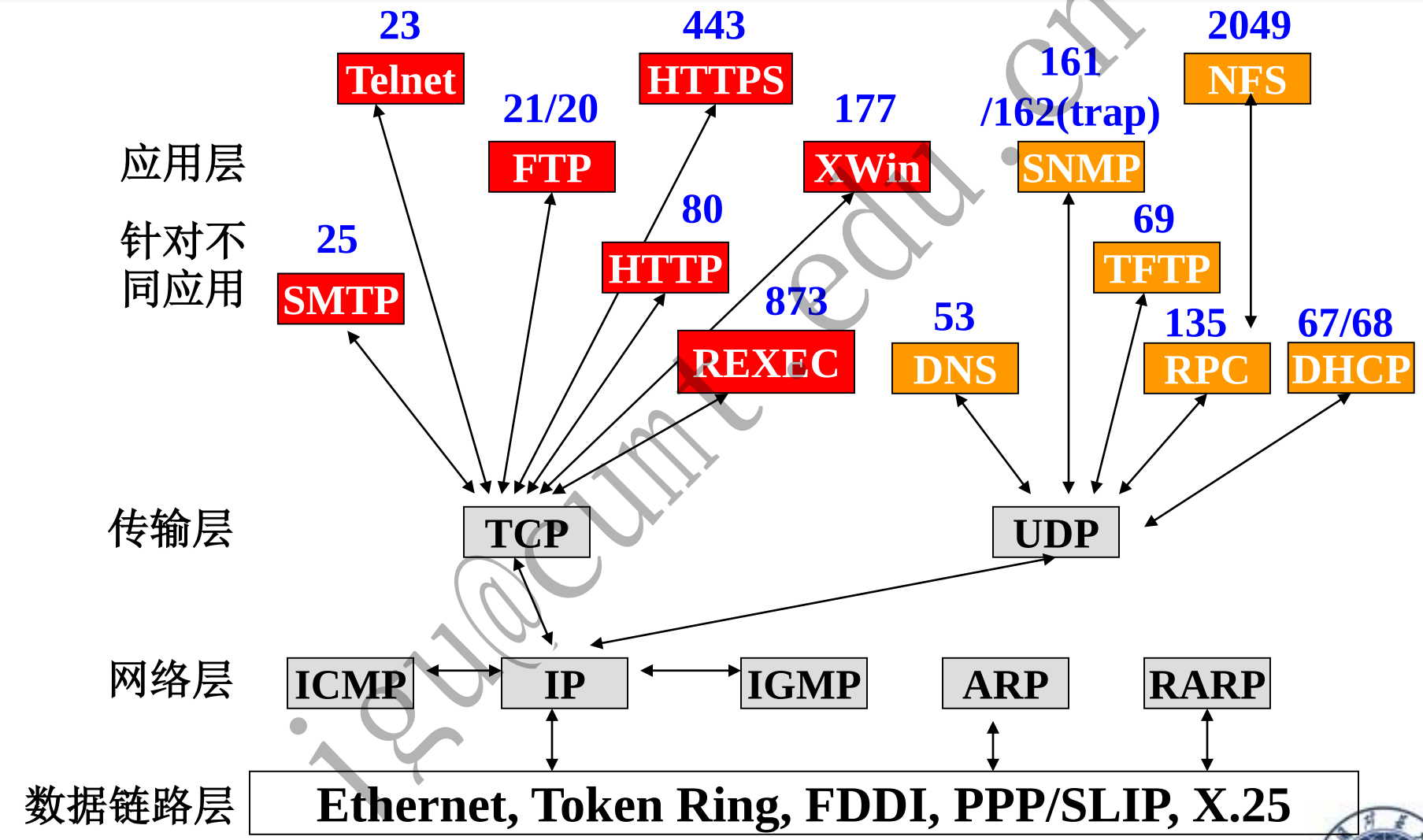


专题6：互联网提供了哪些高层应用



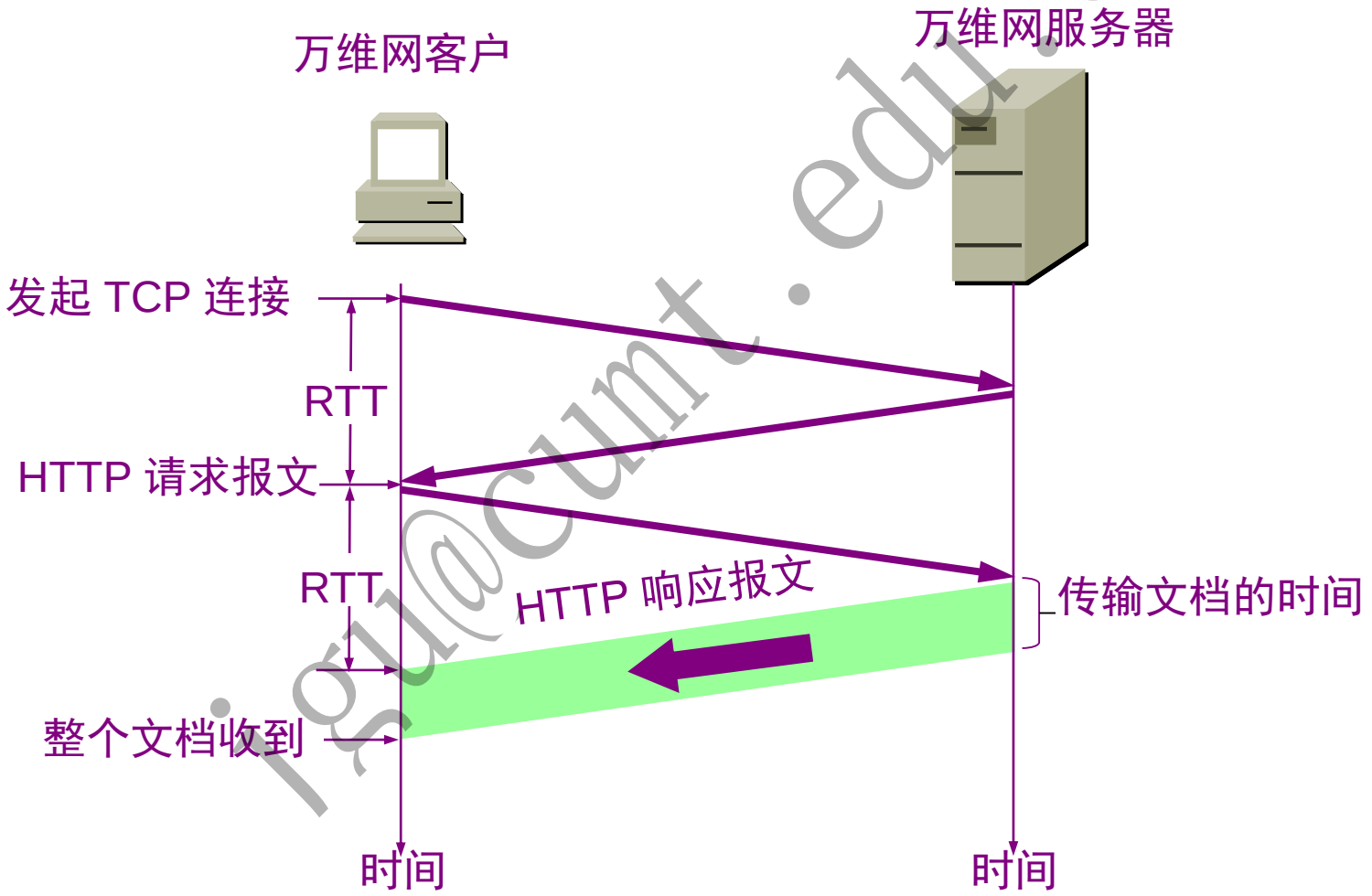
- 应用层(application layer)
- 运输层(transport layer)
- 网络层(network layer)
- 数据链路层(data link layer)
- 物理层(physical layer)

TCP/IP协议族中的应用层协议



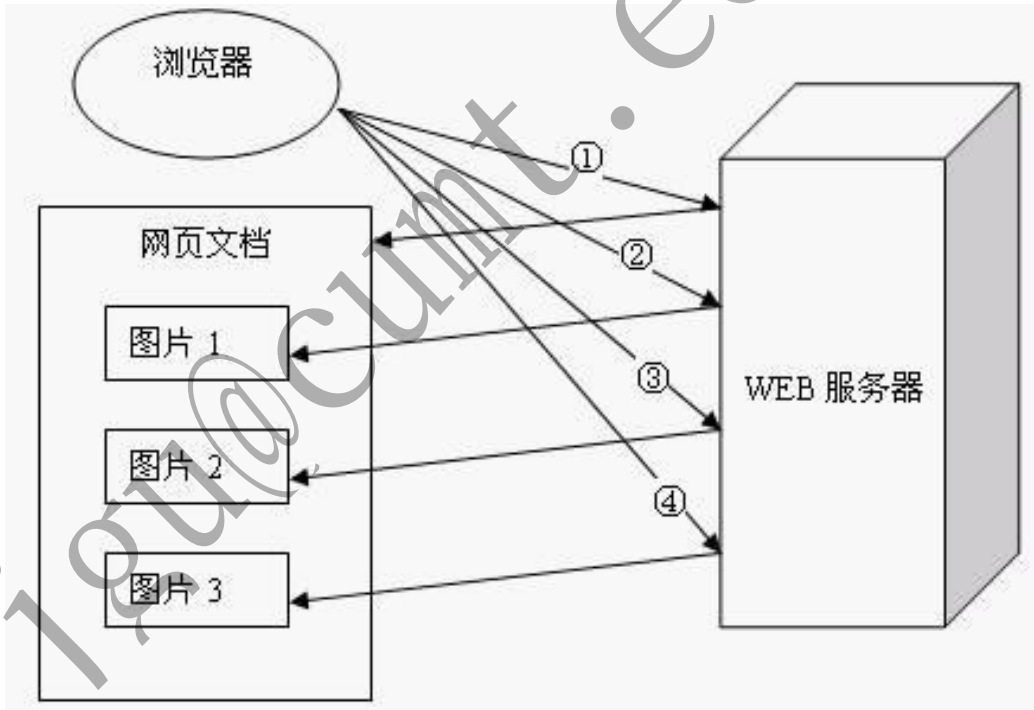
HTTP/1.0非持续连接

- 每请求一个文档就要有两倍RTT的开销。

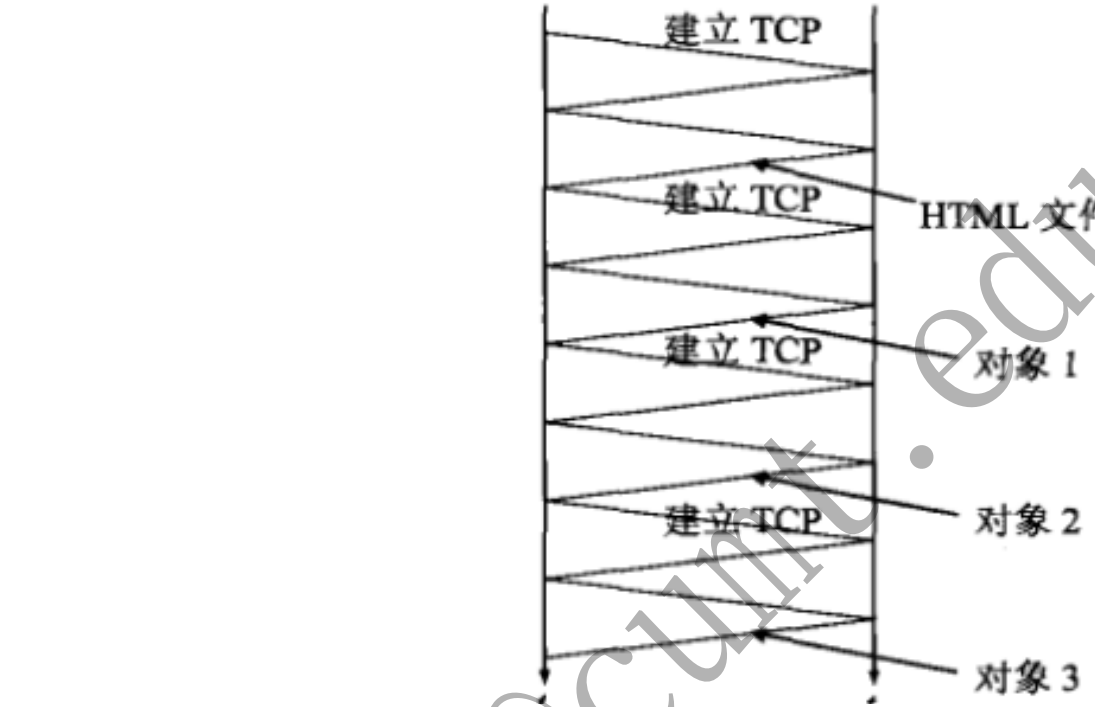


一个万维网文档包含多个链接对象的情况

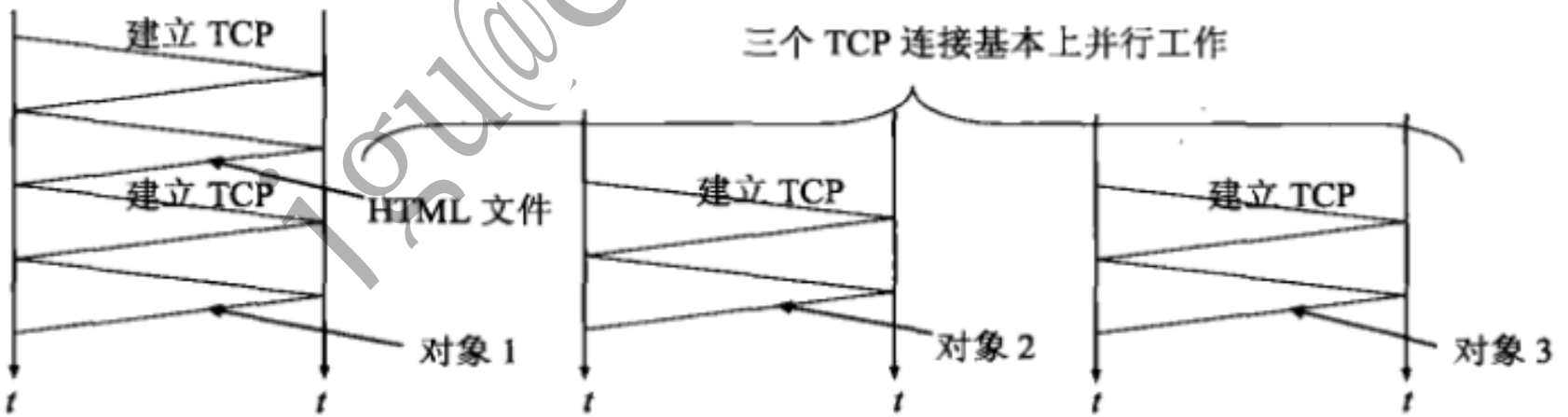
- 若一个主页上的很多链接的对象(如图像等)需要依次进行链接，那么每一次链接下载都会导致2×RTT的开销。



非持续连接的并行和非并行工作方式

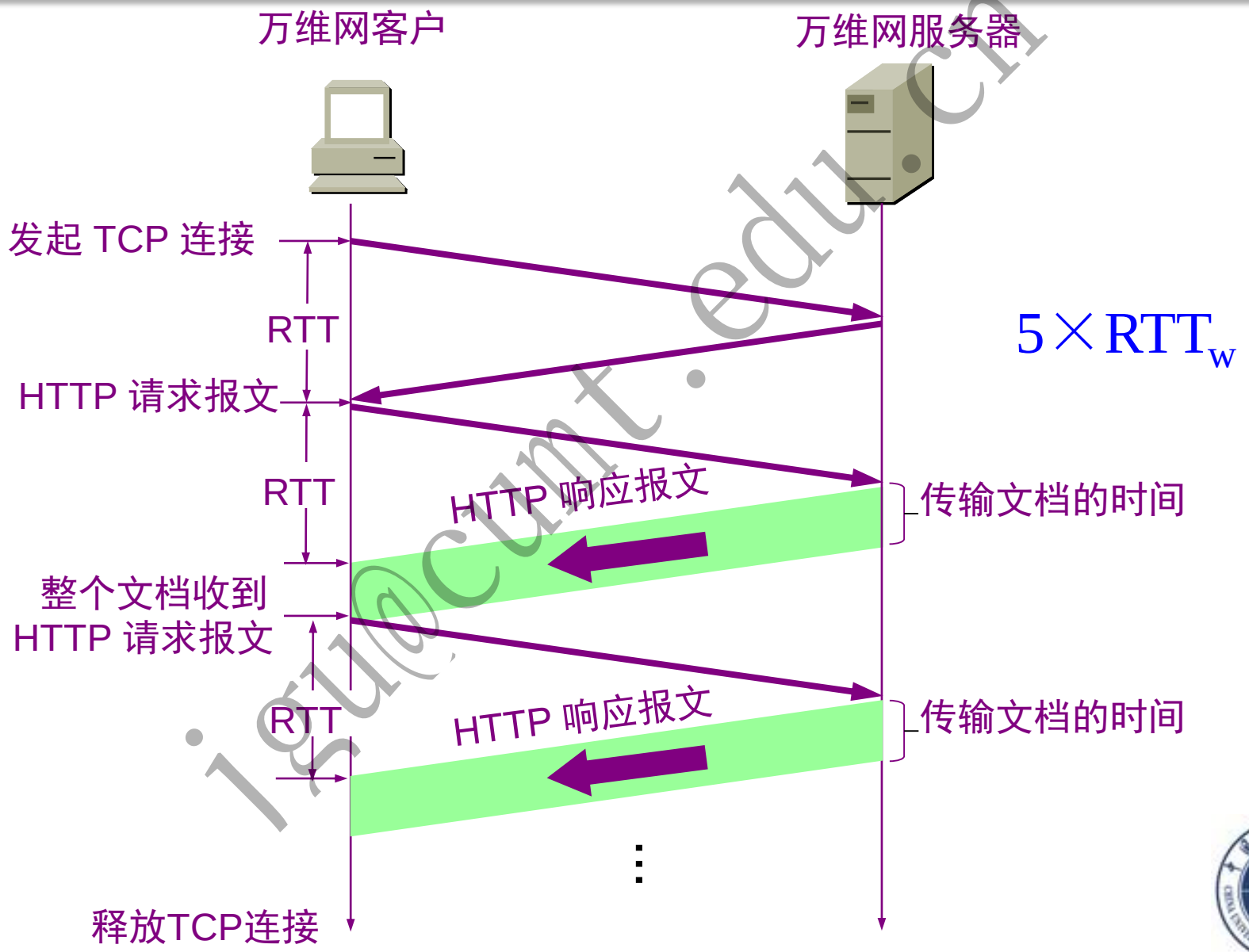


$8 \times RTT_w$

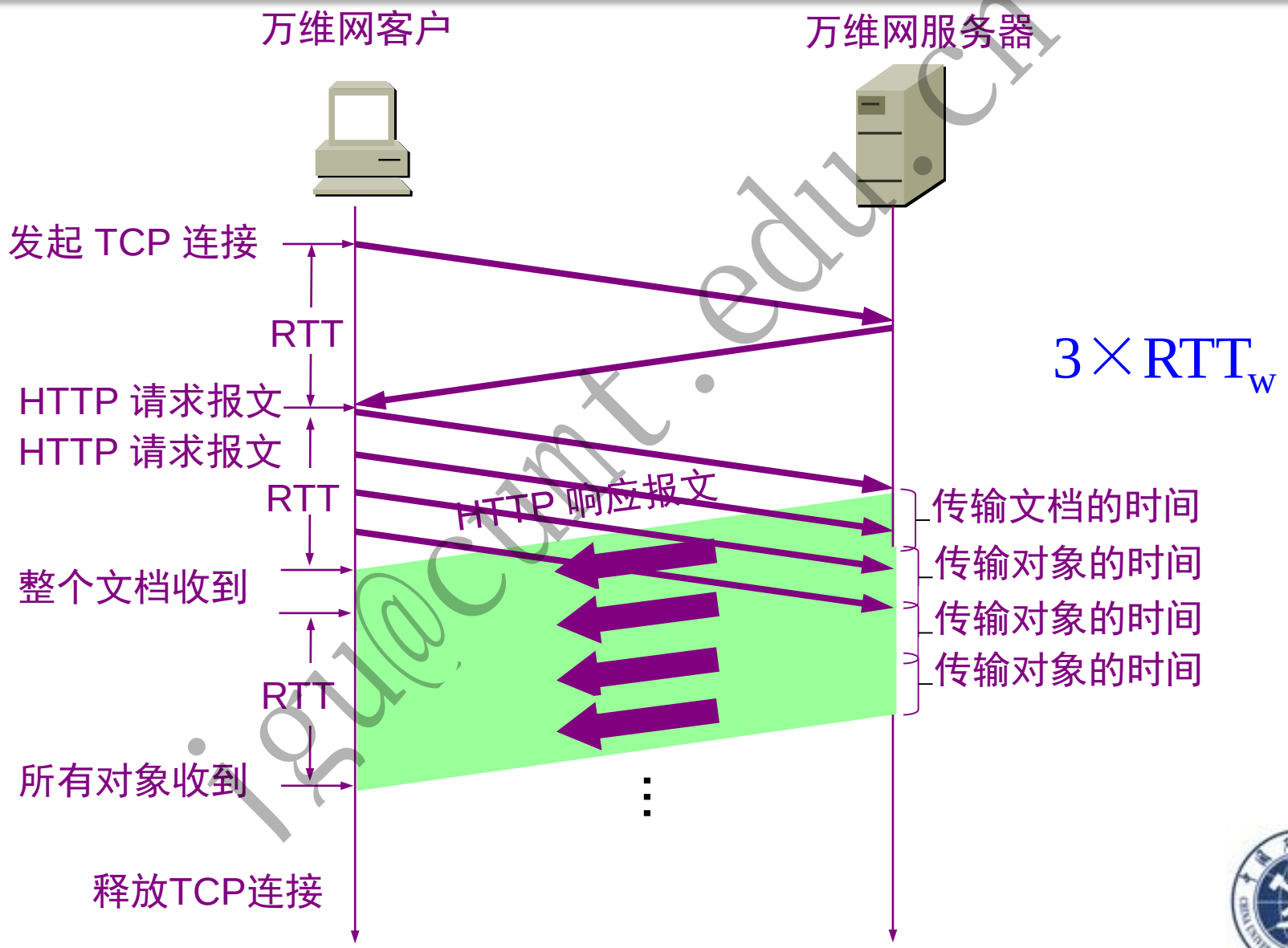


$4 \times RTT_w$

非流水线方式的持续连接



流水线方式的持续连接



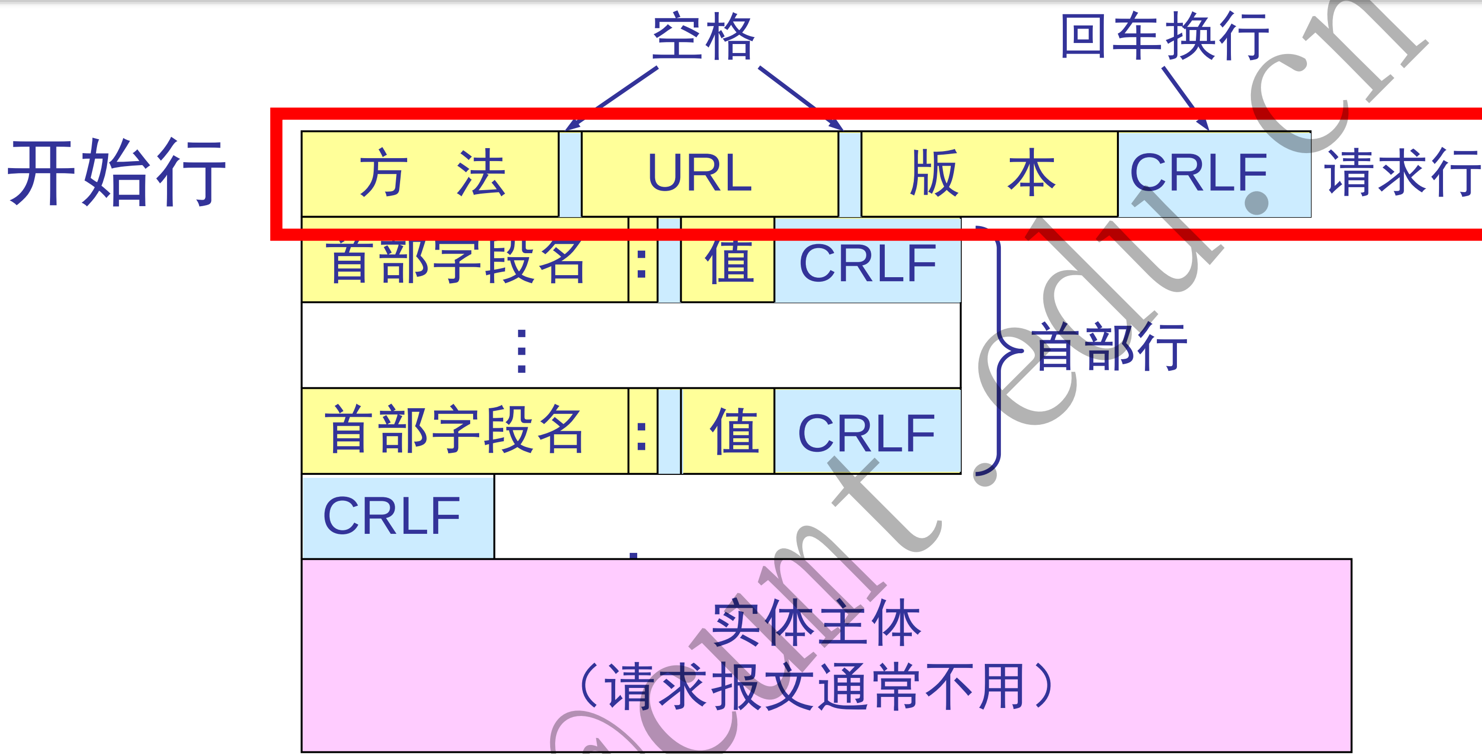
Q18：HTTP/1.x 的报文结构？

HTTP 有两类报文：

- 请求报文——从客户向服务器发送请求报文。
- 响应报文——从服务器到客户的回答。

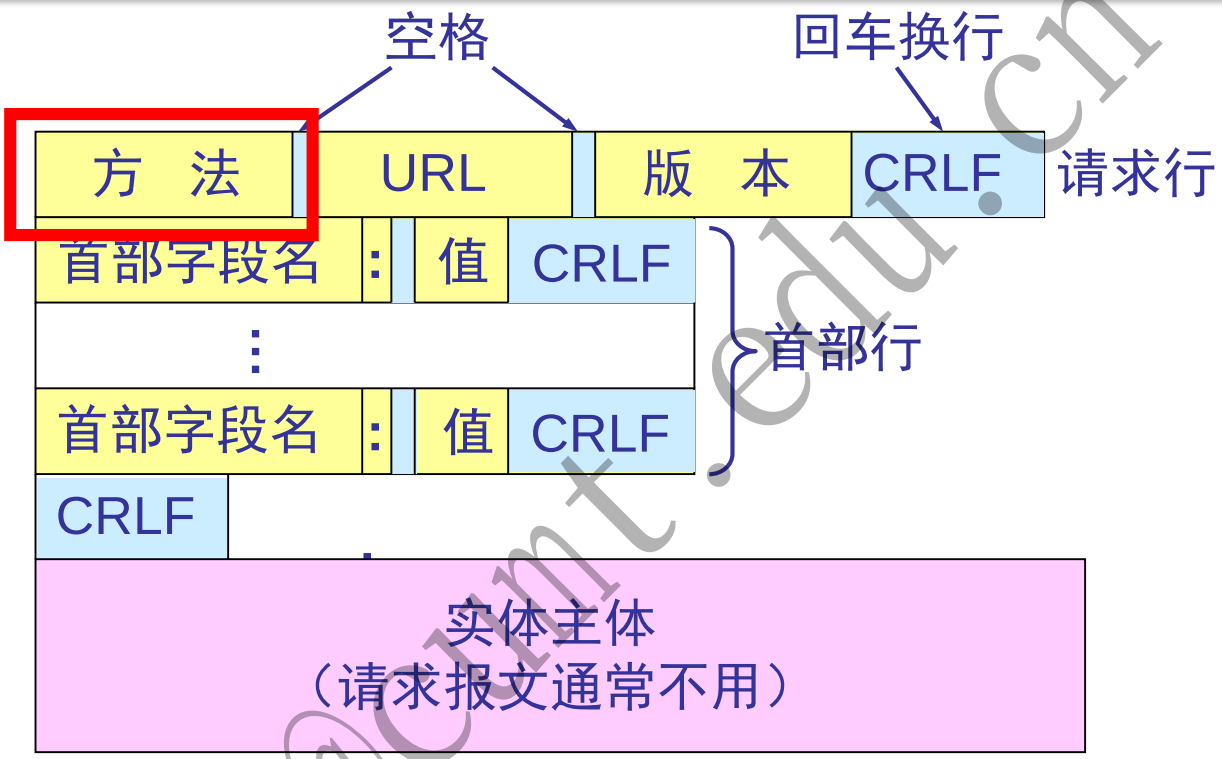
- 由于 HTTP 是面向正文的(text-oriented)，因此在报文中的每一个字段都是一些 ASCII 码串，因而每个字段的长度都是不确定的。

HTTP/1.x 的报文结构（请求报文）



报文由三个部分组成，即开始行、首部行和实体主体。
在请求报文中，开始行就是请求行。

HTTP/1.x 的报文结构（请求报文）

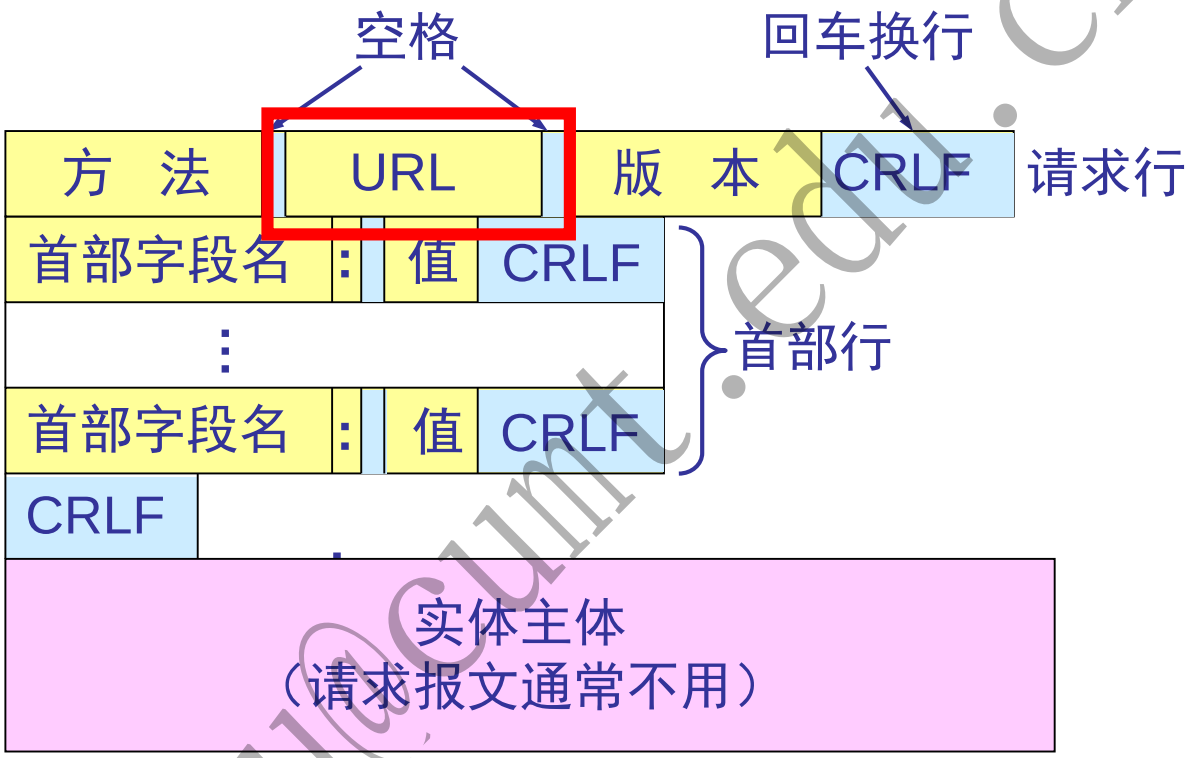


“方法”是面向对象技术中使用的专门名词。所谓“方法”就是对所请求的对象进行的操作，因此这些方法实际上也就是一些命令。因此，请求报文的类型是由它所采用的方法决定的。

HTTP 请求报文的一些方法

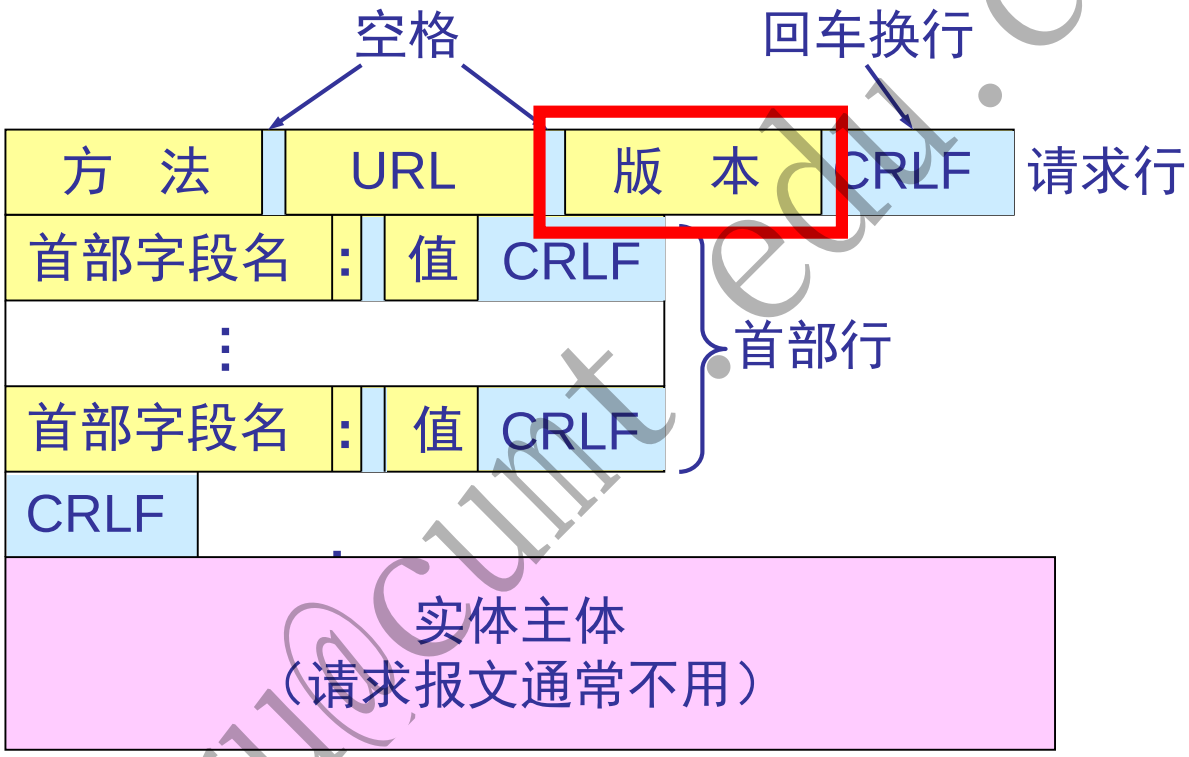
方法（操作）	意义
OPTION	请求一些选项的信息
GET	请求读取由 URL所标志的信息
HEAD	请求读取由 URL所标志的信息的首部
POST	给服务器添加信息（例如，注释）
PUT	在指明的 URL下存储一个文档
DELETE	删除指明的 URL所标志的资源
TRACE	用来进行环回测试的请求报文
CONNECT	用于代理服务器

HTTP/1.x 的报文结构（请求报文）



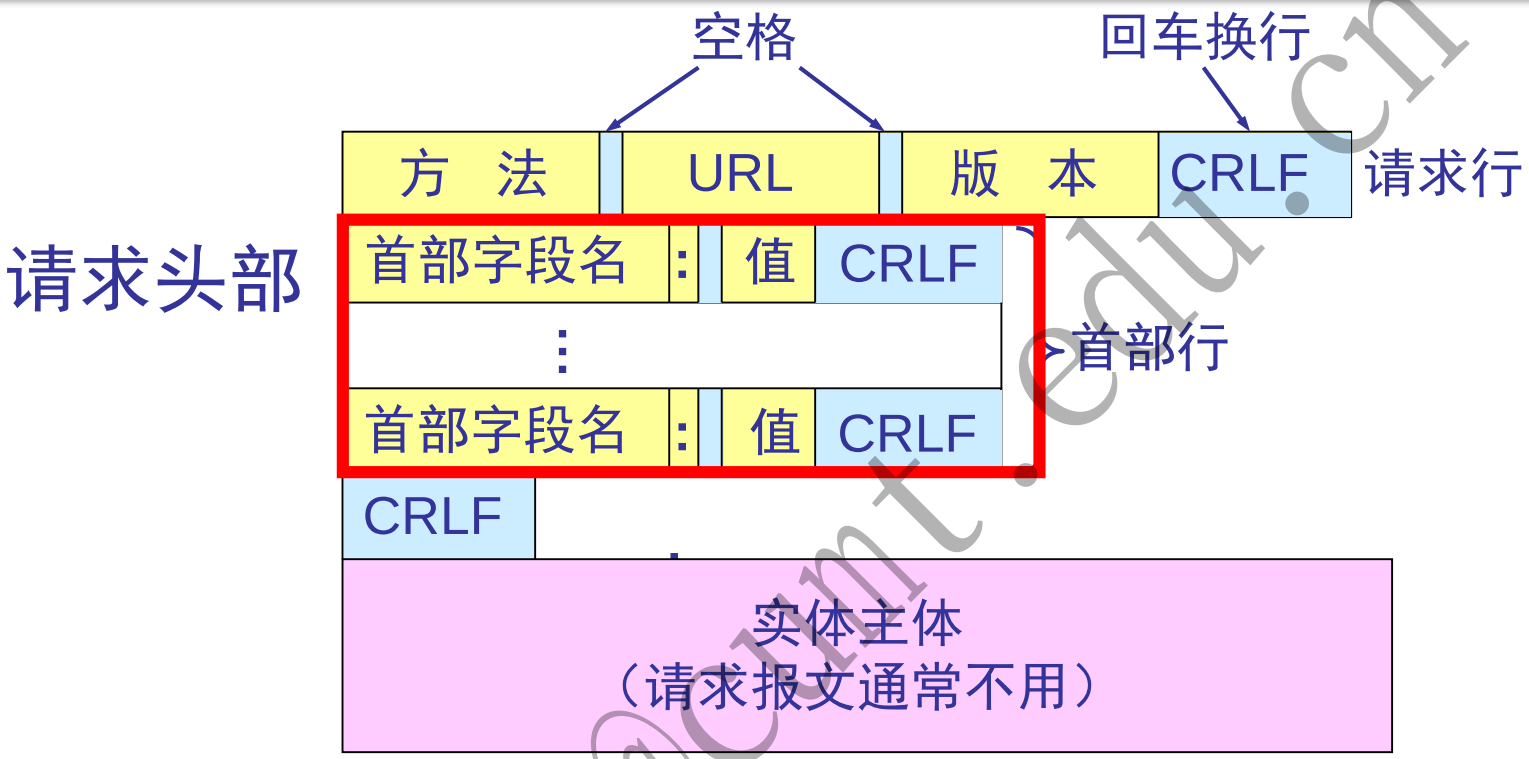
“URL”是所请求的资源的 URL。

HTTP/1.x 的报文结构（请求报文）



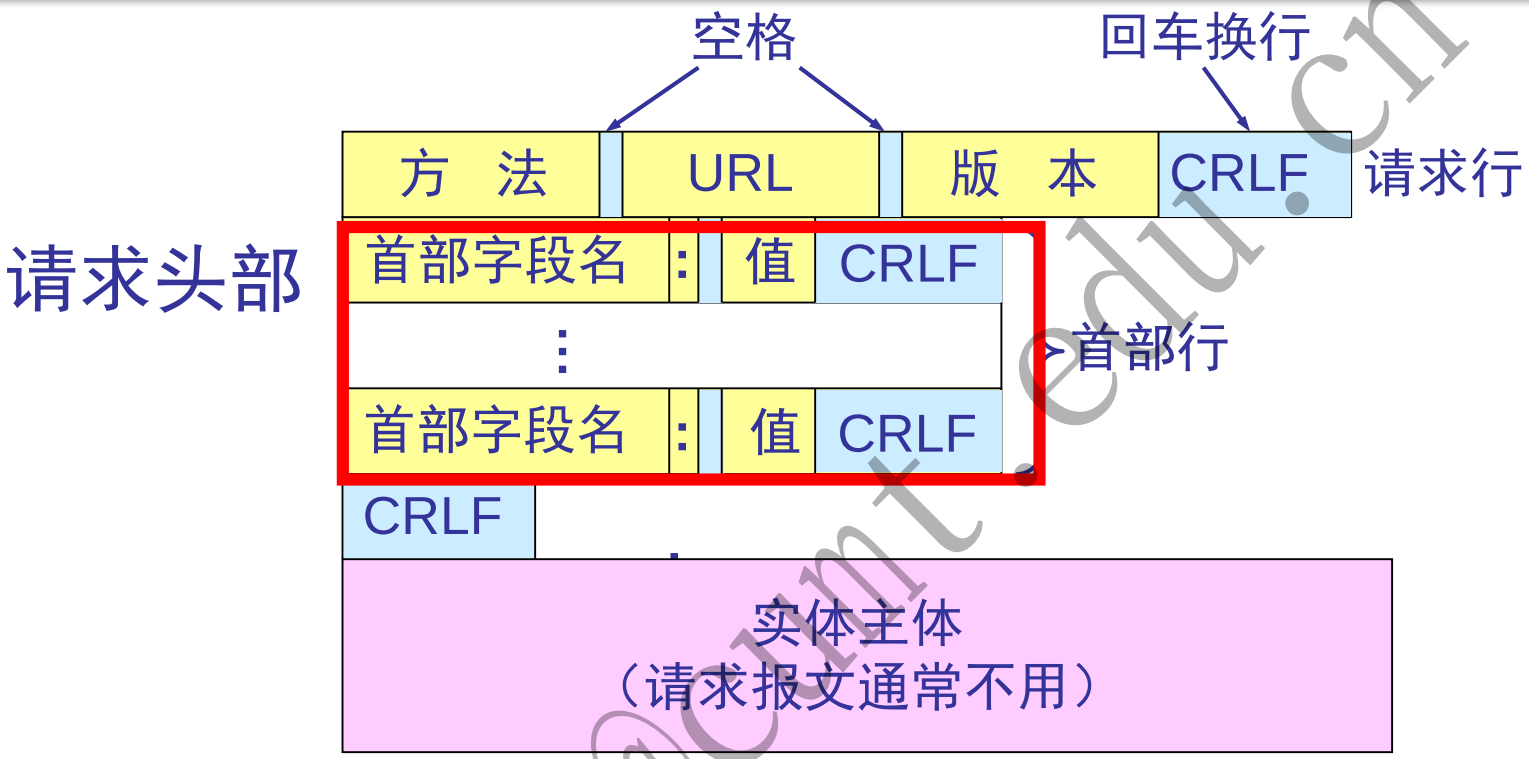
- “版本”是 HTTP 的版本。

HTTP/1.x 的报文结构（请求报文）



请求头部由关键字/值对组成，每行一对，关键字和值用英文冒号“:”分隔。请求头部通知服务器有关于客户端请求的信息。

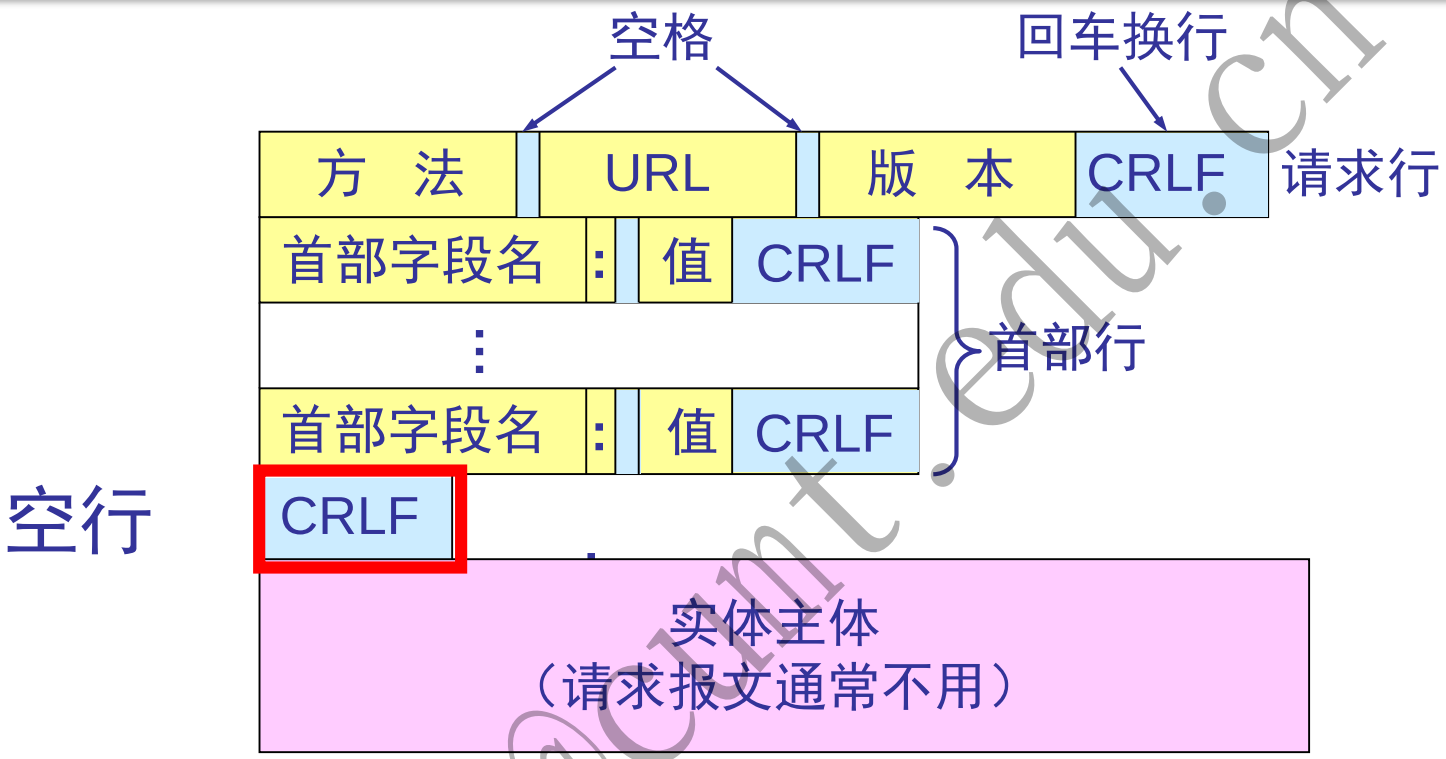
HTTP/1.x 的报文结构（请求报文）



典型的请求头有：

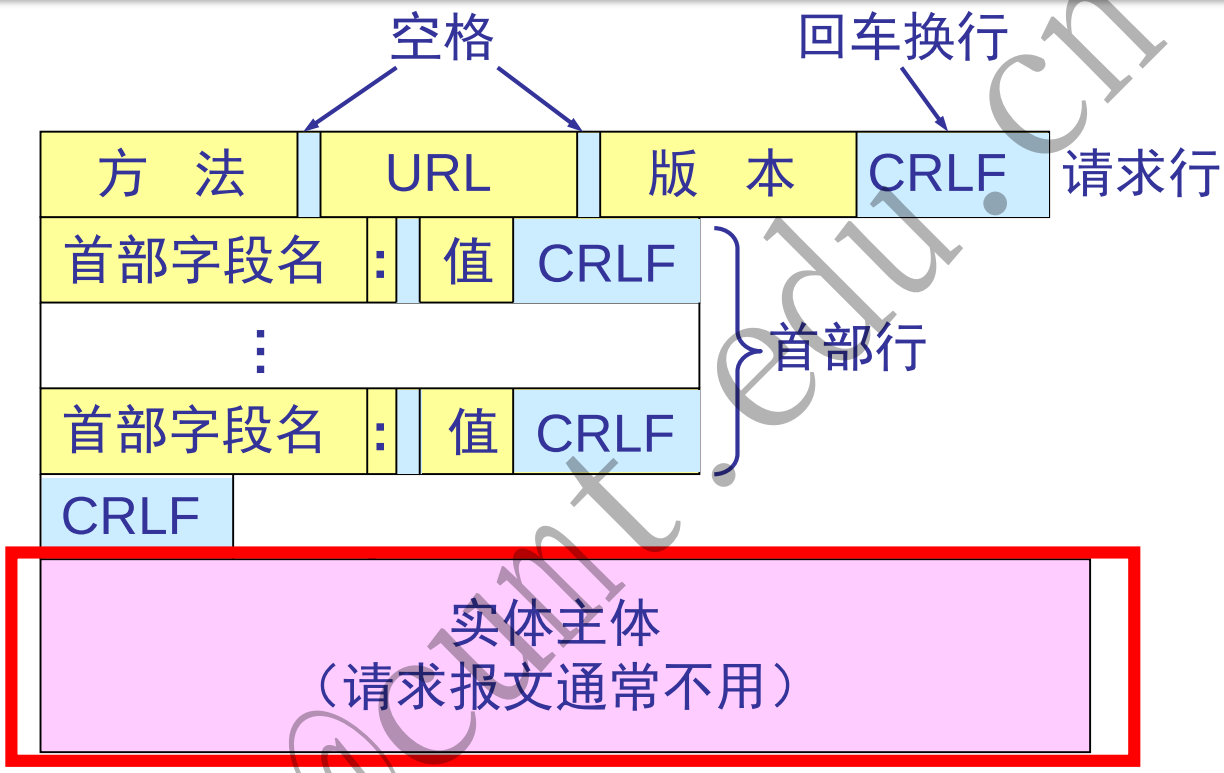
- User-Agent：产生请求的浏览器类型。
- Accept：客户端可识别的内容类型列表。
- Host：请求的主机名，允许多个域名同处一个IP地址，即虚拟主机。

HTTP/1.x 的报文结构（请求报文）



最后一个请求头之后是一个空行，发送回车符和换行符，通知服务器以下不再有请求头。

HTTP/1.x 的报文结构（请求报文）



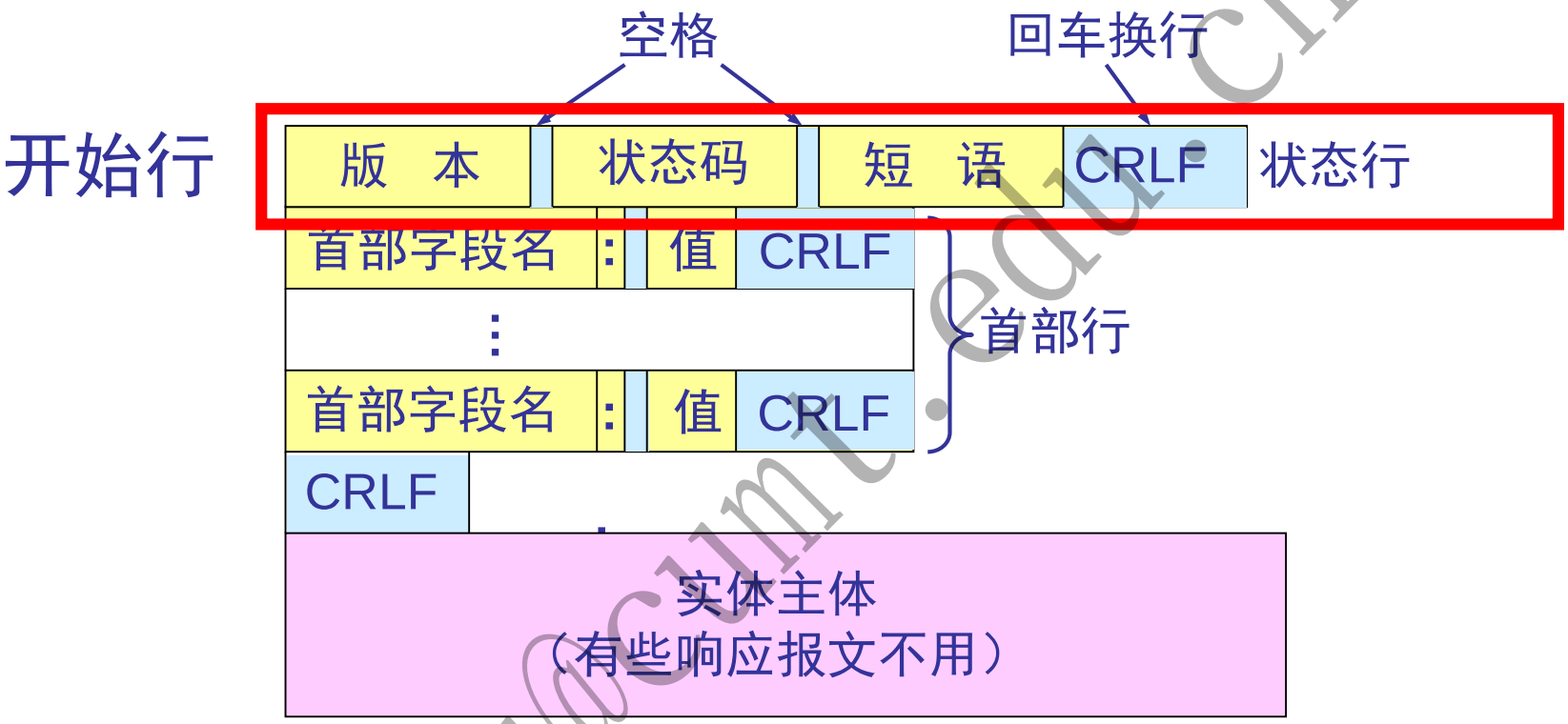
请求数据

请求数据不在GET方法中使用，而是在POST方法中使用。POST方法适用于需要客户填写表单的场合。与请求数据相关的最常使用的请求头是Content-Type和Content-Length。

HTTP请求报文例子

GET /chn/yxsz/index.htm HTTP/1.1 {请求行使用了相对URL}
Host： www.tsinghua.edu.cn {此行是首部行的开始。这行给出主机的域名}
Connection： close {告诉服务器发送完请求的文档后就可释放链接}
User-Agent：Mozilla/5.0 {表明用户代理是使用Netscape浏览器}
Accept-Language：cn {表示用户希望优先得到中文版本的文档}
{请求报文的最后还有一个空行}

HTTP/1.x 的报文结构（响应报文）



响应报文的开始行是状态行。
状态行包括三项内容，即 HTTP 的版本，状态码，以及解释状态码的简单短语。

状态码都是三位数字

- 1xx 表示通知信息的，如请求收到了或正在进行处理。
- 2xx 表示成功，如接受或知道了。
- 3xx 表示重定向，表示要完成请求还必须采取进一步的行动。
- 4xx 表示客户的差错，如请求中有错误的语法或不能完成。
- 5xx 表示服务器的差错，如服务器失效无法完成请求。

HTTP实体报头

- 请求和响应消息都可以传送一个实体。一个实体由实体报头域和实体正文组成，但并不是说实体报头域和实体正文要在一起发送，可以只发送实体报头域。实体报头定义了关于实体正文（eg：有无实体正文）和请求所标识的资源的元信息。

HTTP实体报头

- Content-Type：用于指明发送给接收者的实体正文的媒体类型。
- Expires：给出响应过期的日期和时间。为了让代理服务器或浏览器在一段时间以后更新缓存中(再次访问曾访问过的页面时，直接从缓存中加载，缩短响应时间和降低服务器负载)的页面，我们可以使用Expires实体报头域指定页面过期的时间。
- Last-Modified：用于指示资源的最后修改日期和时间。

HTTP实体报头

- Content-Encoding，被用作媒体类型的修饰符，它的值指示了已经被应用到实体正文的附加内容的编码，因而要获得Content-Type报头域中所引用的媒体类型，必须采用相应的解码机制。Content-Encoding主要用于记录文档的压缩方法。
- Content-Language，描述了资源所用的自然语言。没有设置该域则认为实体内容将提供给所有的语言阅读者。
- Content-Length，用于指明实体正文的长度，以字节方式存储的十进制数字来表示。即一个数字字符占一个字节，用其对应的ASCII码来存储传输。

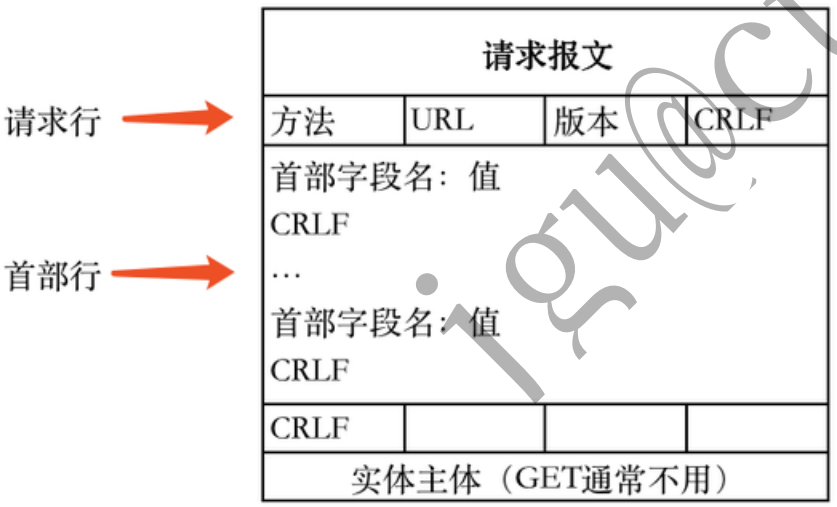
HTTP响应报文例子

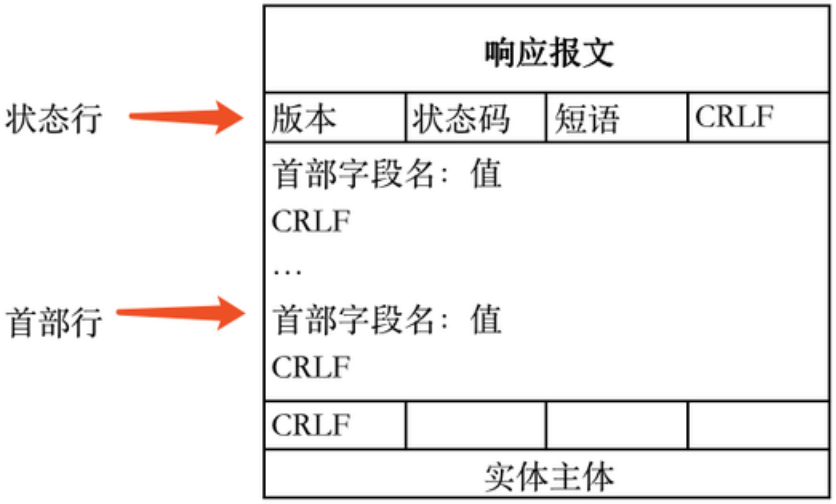
```
HTTP/1.1 200 OK
Date: Sat, 31 Dec 2015 23:59:59 GMT
Content-Type: text/html; charset=ISO-8859-1
Content-Length: 122

<html>
<head>
<title>Wrox Homepage</title>
</head>
<body>
<!-- body goes here -->
</body>
</html>
```

HTTP/1.x有安全缺陷

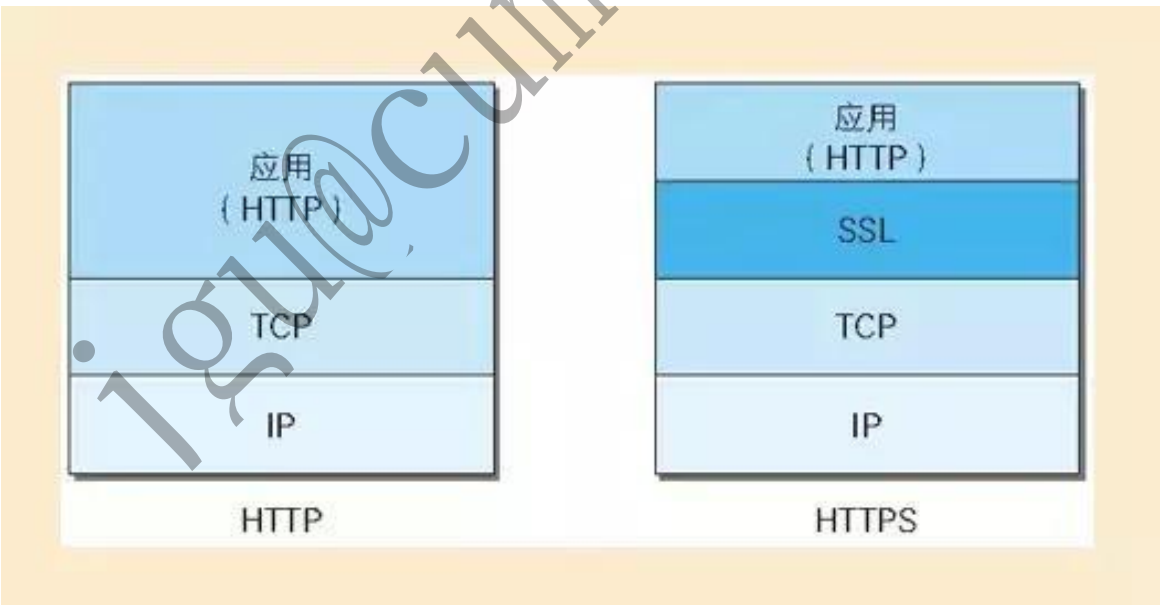
- HTTP/1.x诞生的时候是明文协议，其格式由三部分组成：start line（request line或者status line）、header、body。
- 要识别这3部分就要做协议解析，HTTP/1.x的解析是基于文本，健壮性较差。





Q19：HTTPS如何工作？

- HTTPS是一种基于SSL/TLS的HTTP，所有的HTTPS数据都是在SSL/TLS协议封装之上传输的。
  - TLS是传输层加密协议，前身是SSL协议，由网景公司1995年发布，有时候两者不区分。
- 简单来说，HTTPS是安全版的HTTP，将HTTP的数据进行加密，其所用的端口是443。



HTTPS的特点

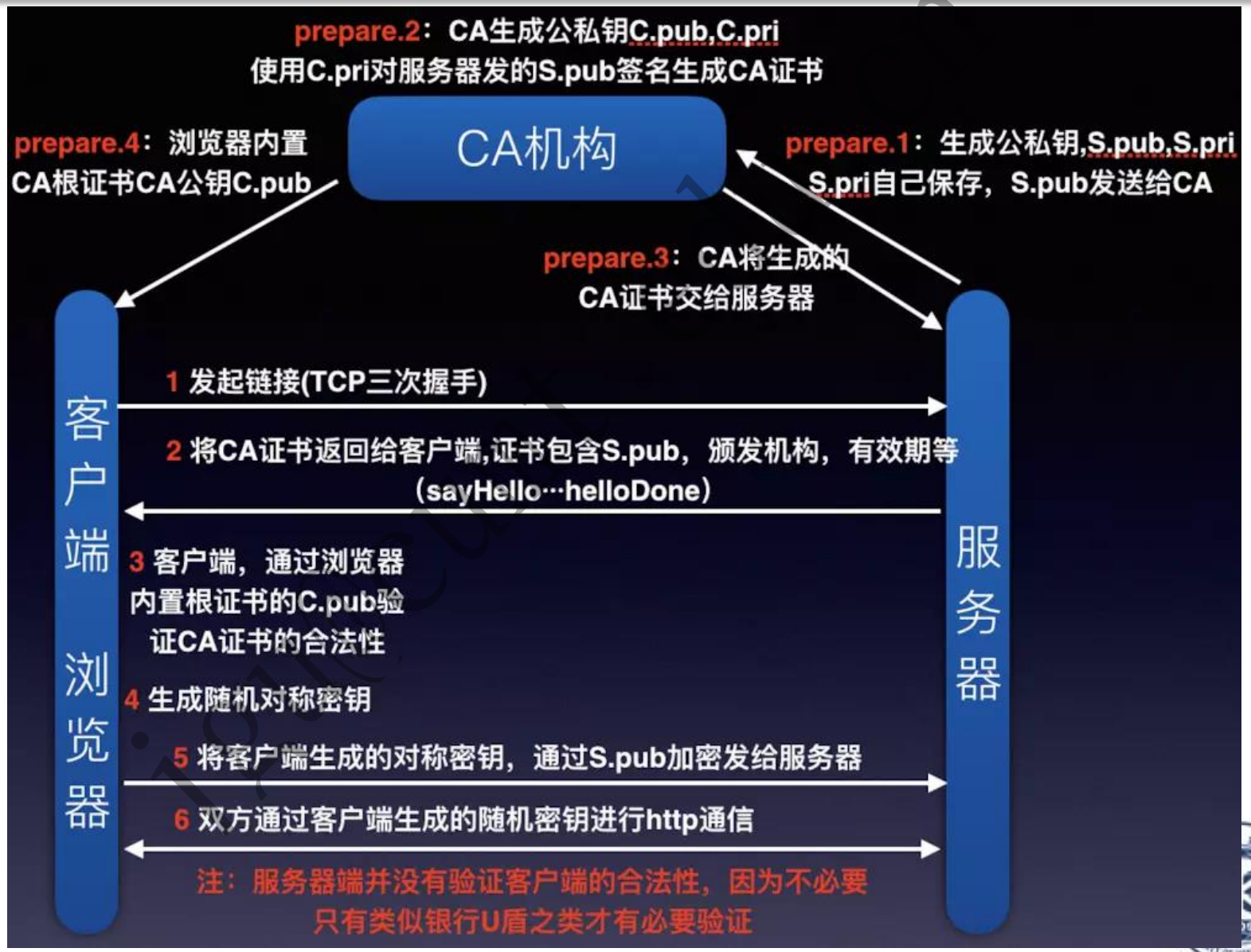
- 内容加密：采用混合加密技术（结合非对称加密和对称加密技术），中间者无法直接查看明文内容
- 验证身份：通过证书认证客户端访问的是自己的服务器
- 保护数据完整性：防止传输的内容被中间人冒充或者篡改

HTTPS的建立过程

- HTTPS协议安全通信步骤中有三个主角元素：客户端，服务器，第三方可信任的证书颁发机构。
- HTTPS建立过程相当复杂
  - 在客户端向服务器发起请求前，需要完成一些准备工作，如：从CA证书颁发机构，获取数字证书，颁发给申请者（服务器）；客户端（比如浏览器），为了安全性，会内置一份CA根证书。
  - 完成上述准备工作后才能发起链接。

HTTPS的建立过程

先完成准备工作，再发起链接

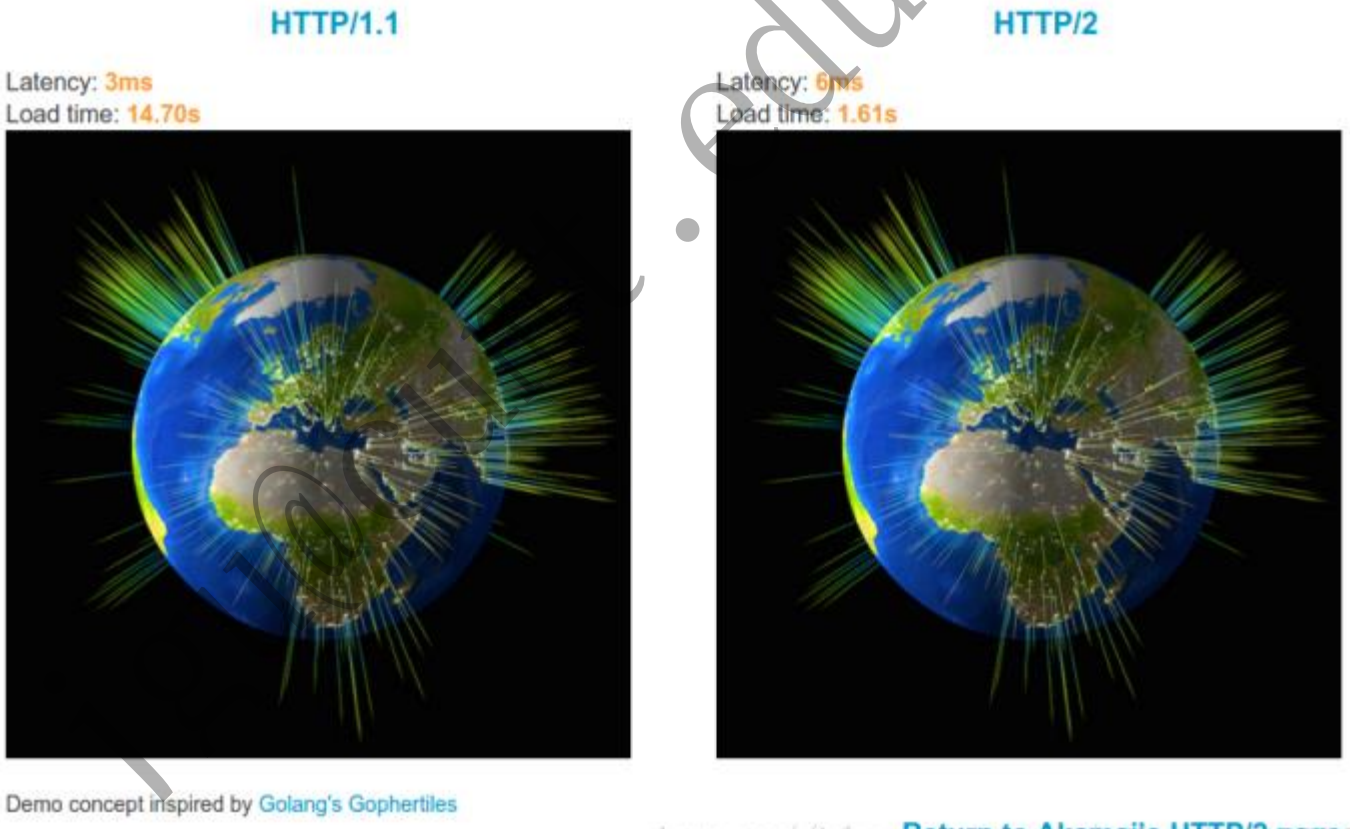


Q20：HTTP/2如何工作？

- HTTP/2 （原名HTTP/2.0）即超文本传输协议 2.0，是下一代HTTP协议，由互联网工程任务组IETF的Hypertext Transfer Protocol Bis (httpbis)工作小组进行开发。
- HTTP/2是自1999年HTTP1.1发布后的首个更新，在2013年8月进行首次合作共事性测试。
- 在开放互联网上HTTP 2.0将只用于https://网址，而 http://网址将继续使用HTTP/1，目的是在开放互联网上增加使用加密技术，以提供强有力的保护去遏制主动攻击。

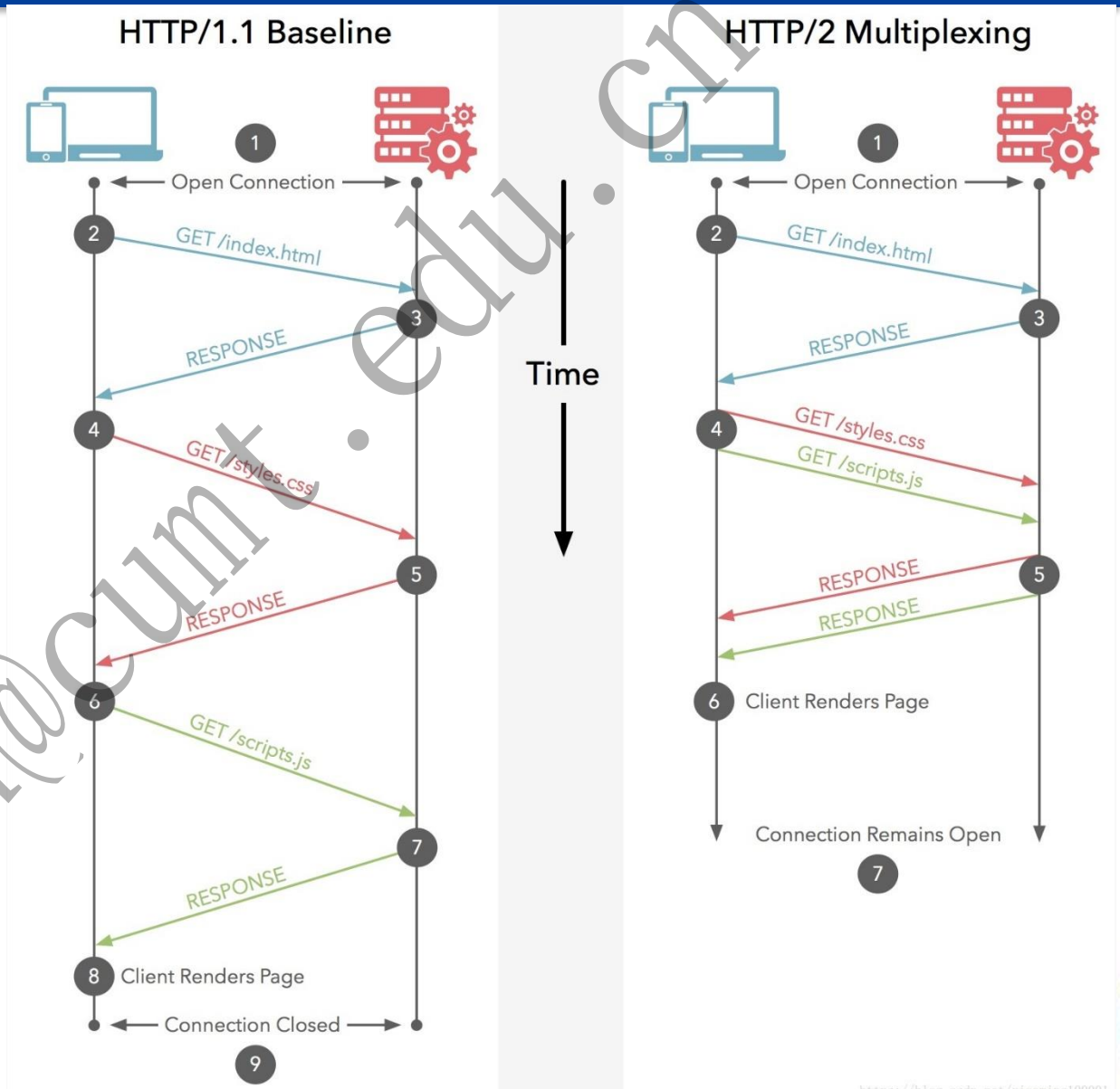
HTTP1.1和HTTP/2的性能对比

Akamai公司建立的一个官方的演示，使用HTTP/1.1和HTTP/2同时请求379张图片，观察请求的时间，明显看出HTTP/2性能占优势。



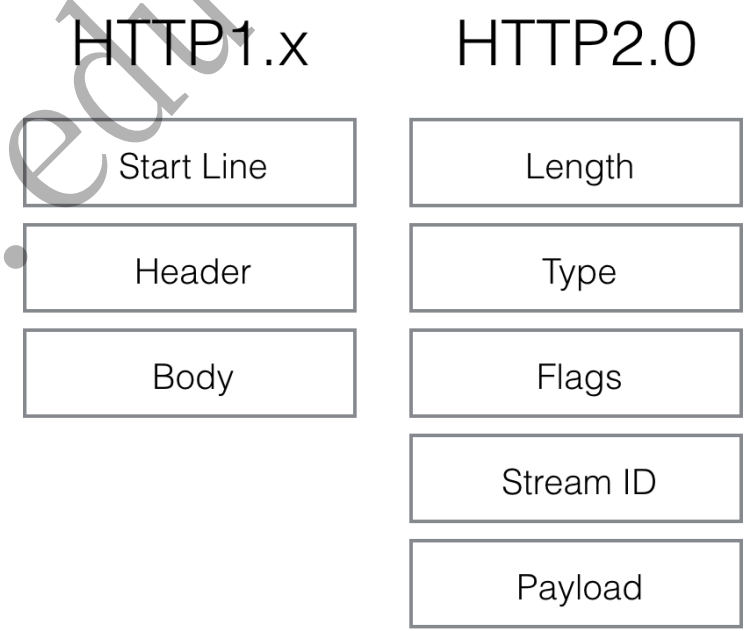
HTTP1.1和HTTP/2的技术对比

多路复用：通过单一的HTTP/2连接请求发起多重的请求-响应消息，多个请求stream共享一个TCP连接，实现多路并行而不是依赖建立多个TCP连接。

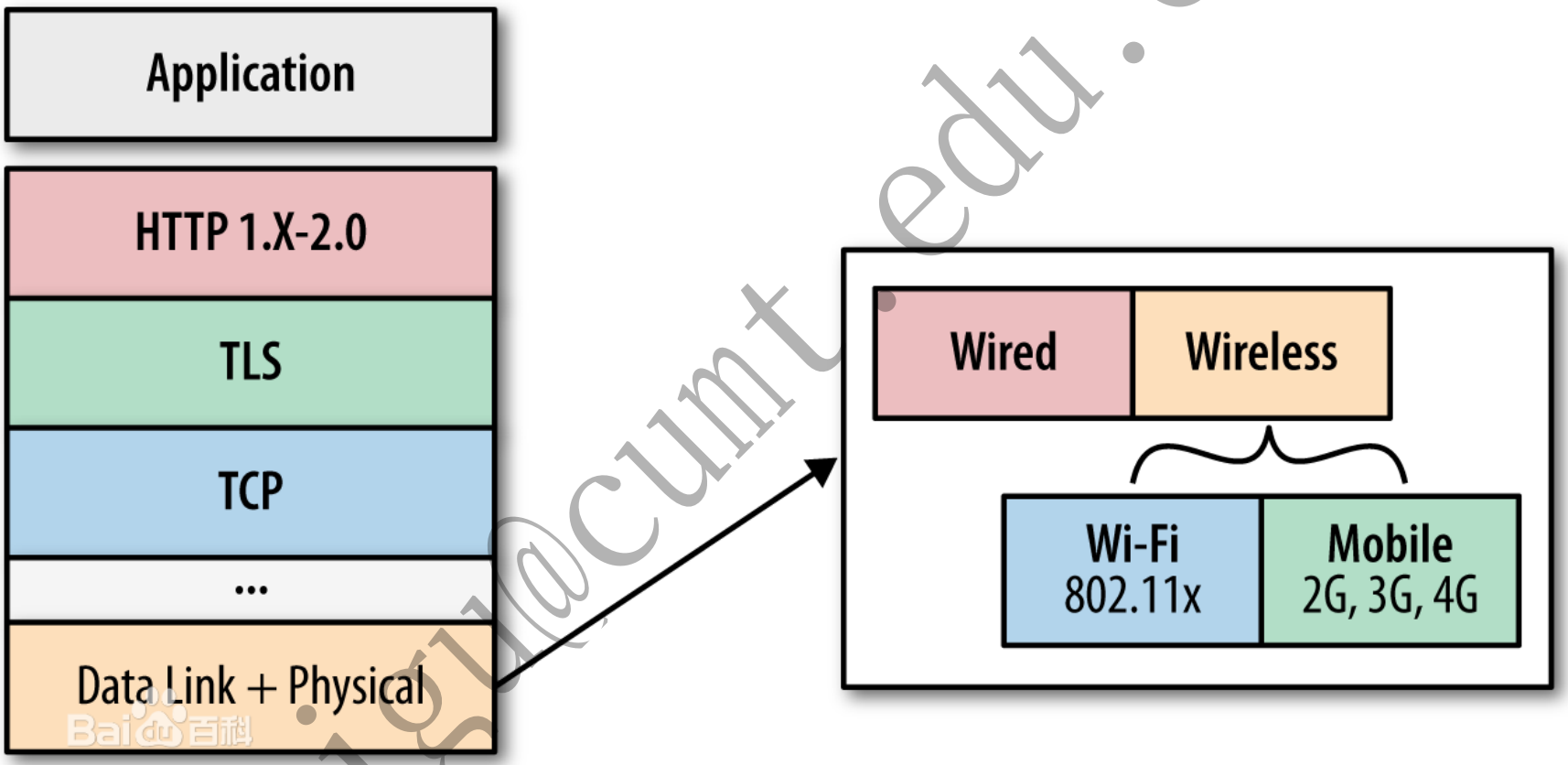


HTTP/2的报文格式

- HTTP2.0用binary格式定义了一个一个的frame，length定义了整个frame的开始到结束，type定义frame的类型（10种），flags用bit位定义一些重要参数，stream id用作流控制，剩下的payload就是request的正文。



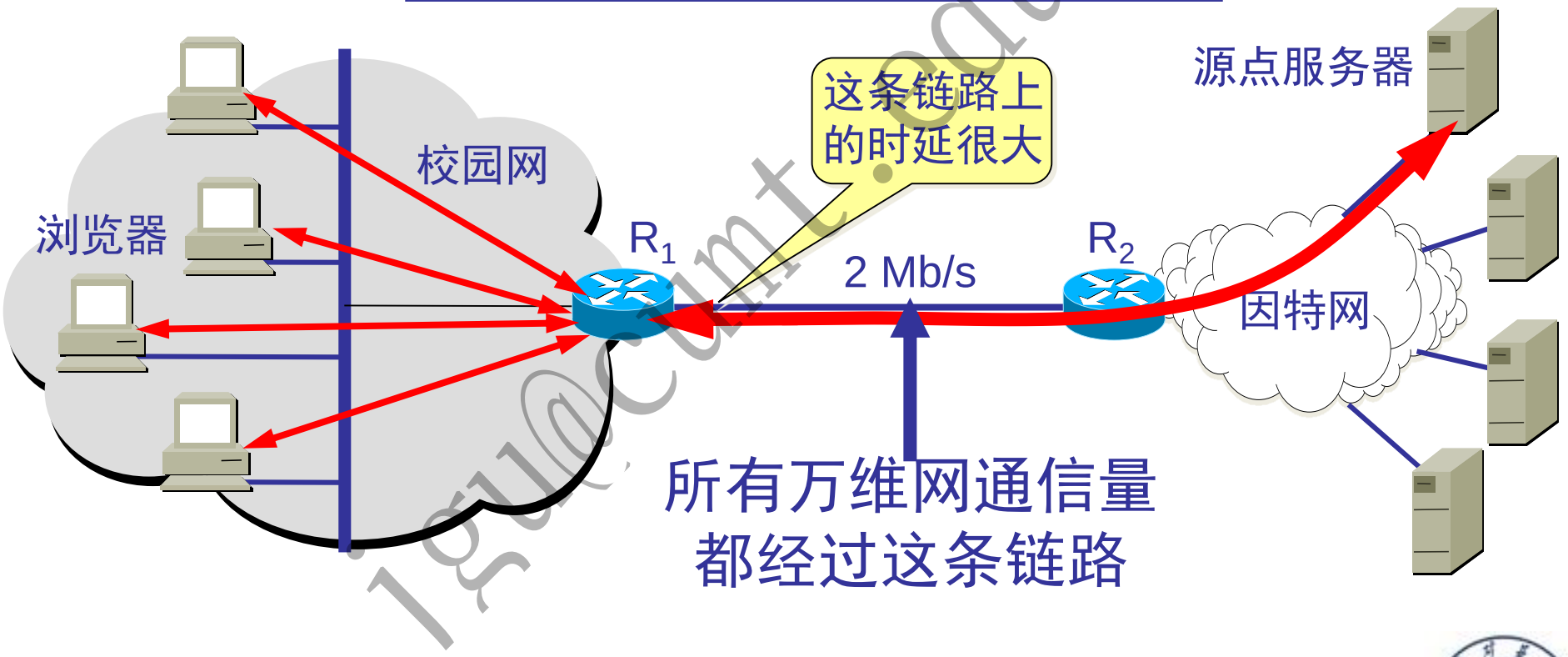
HTTP/2优化Web应用的访问



Q21：如何提高万维网的访问效率？

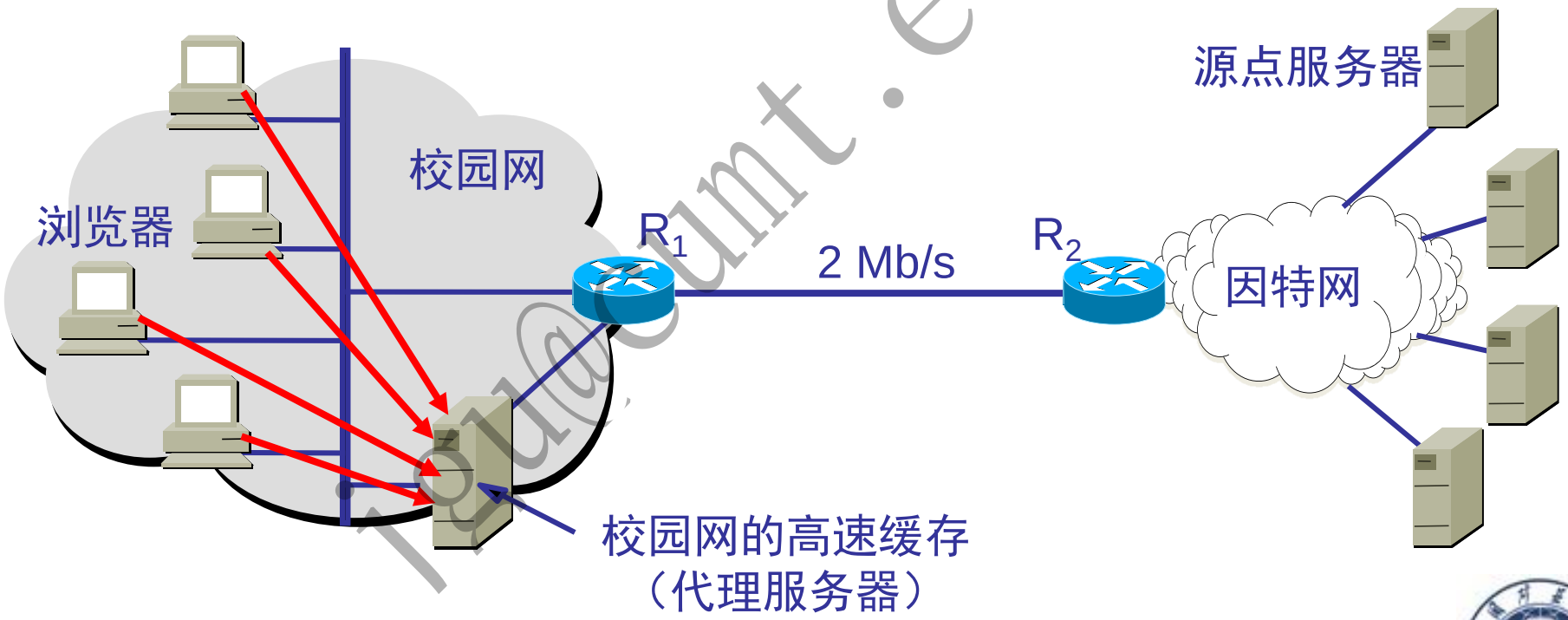
- 代理服务器(proxy server)又称为万维网高速缓存(Web cache)，它代表浏览器发出 HTTP 请求。
- 万维网高速缓存把最近的一些请求和响应暂存在本地磁盘中。
- 当与暂时存放的请求相同的新请求到达时，万维网高速缓存就把暂存的响应发送出去，而不需要按 URL 的地址再去因特网访问该资源。

使用高速缓存可减少访问因特网服务器的时延



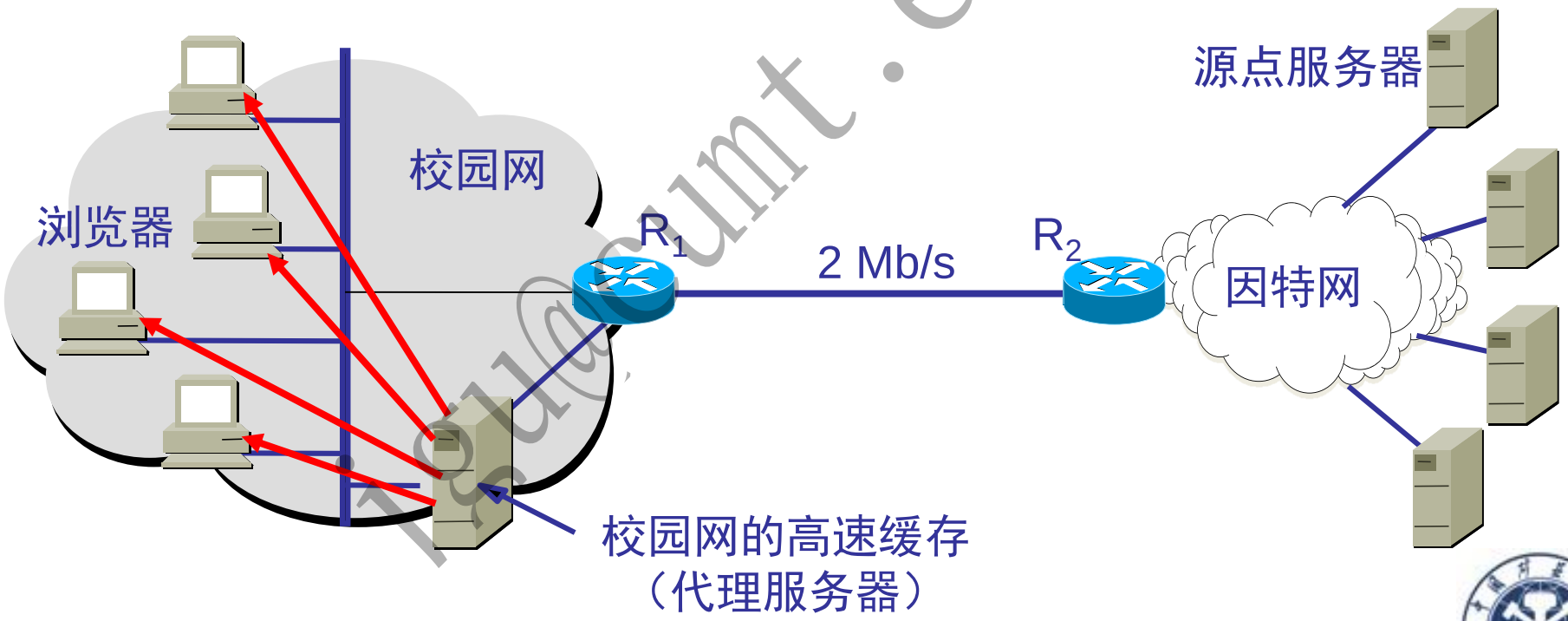
使用高速缓存的情况

(1) 浏览器访问因特网的服务器时，要先与校园网的高速缓存建立 TCP 连接，并向高速缓存发出 HTTP 请求报文



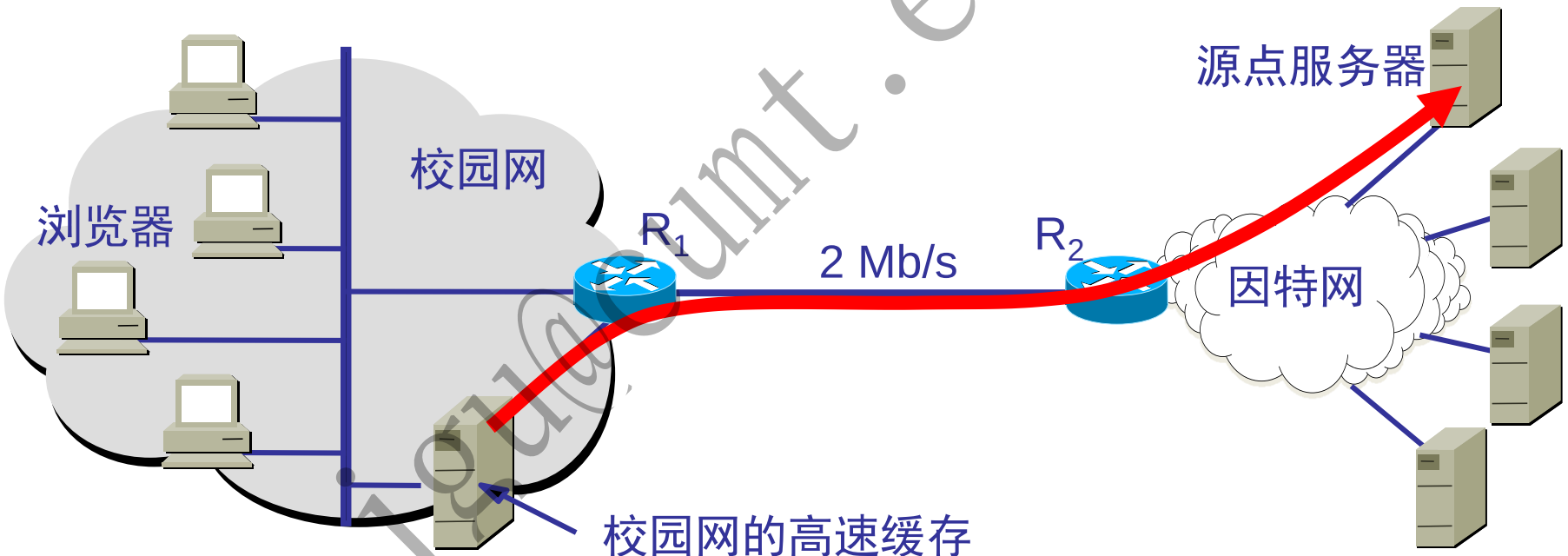
使用高速缓存的情况

(2) 若高速缓存已经存放了所请求的对象，则将此对象放入 HTTP 响应报文中返回给浏览器。



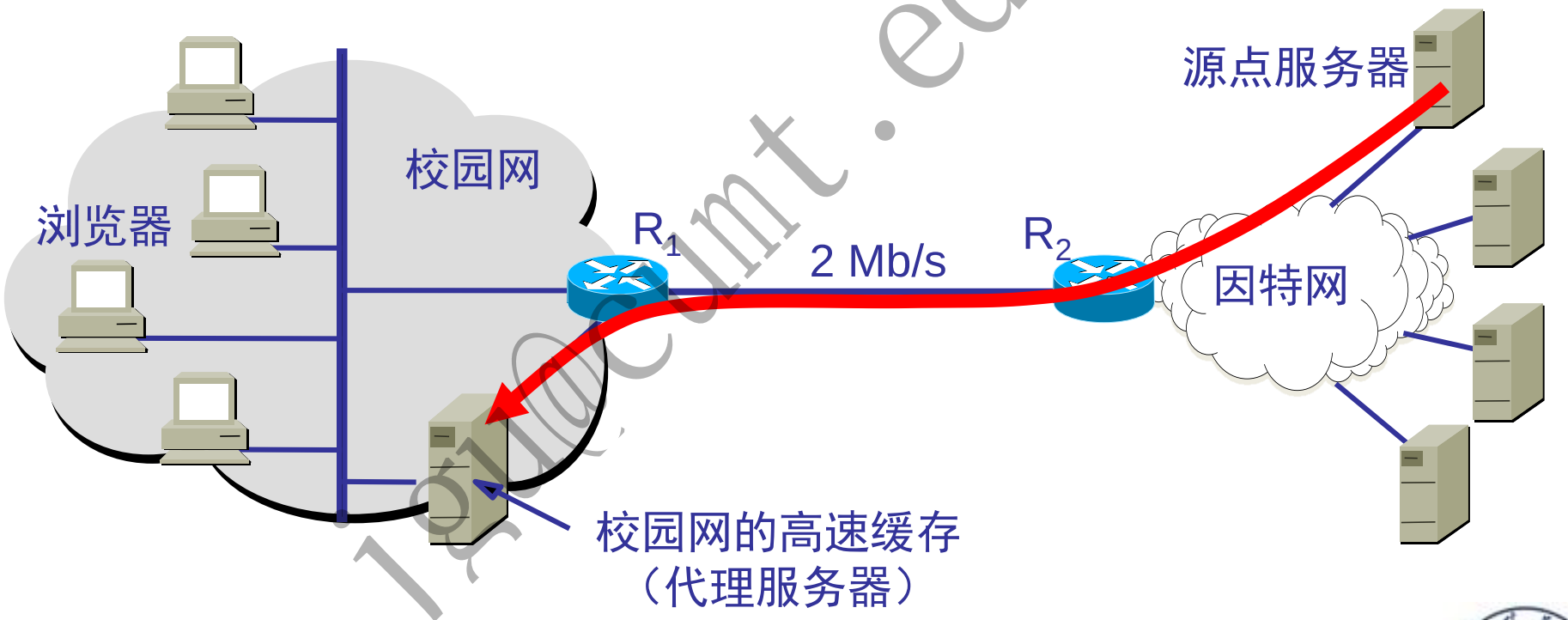
使用高速缓存的情况

(3) 否则，高速缓存就代表发出请求的用户浏览器，与因特网上的源点服务器建立 TCP 连接，并发送 HTTP 请求报文。



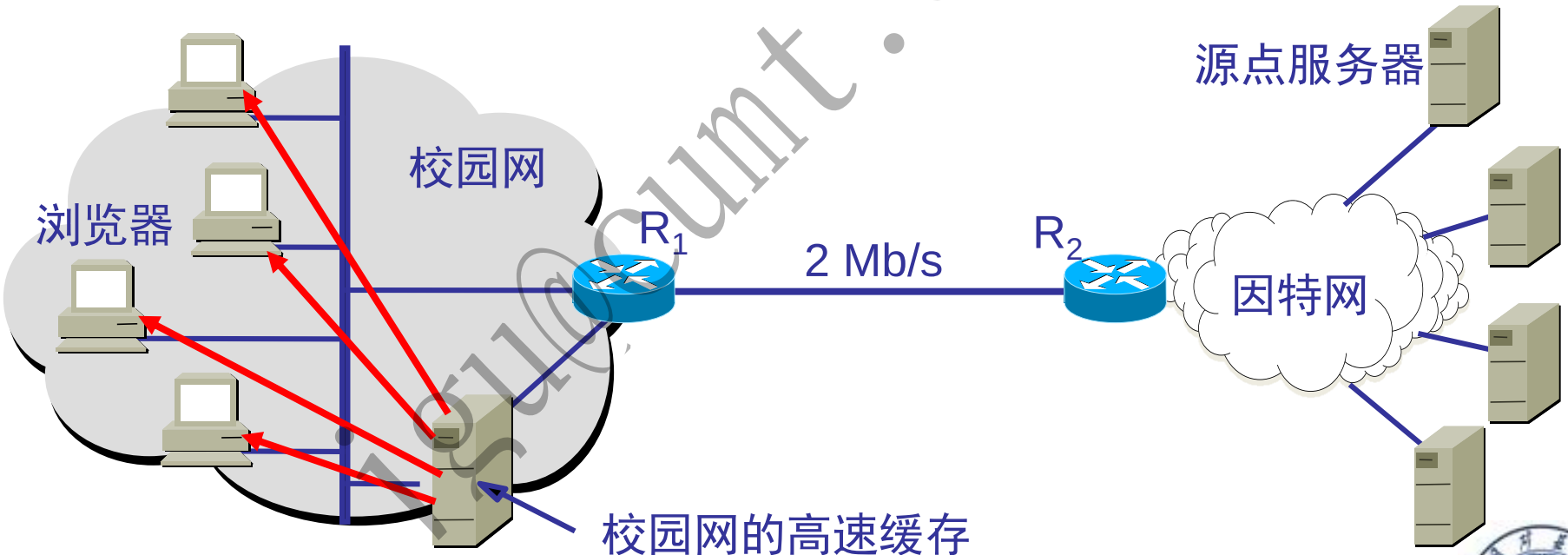
使用高速缓存的情况

(4) 源点服务器将所请求的对象放在 HTTP 响应报文中返回给校园网的高速缓存。

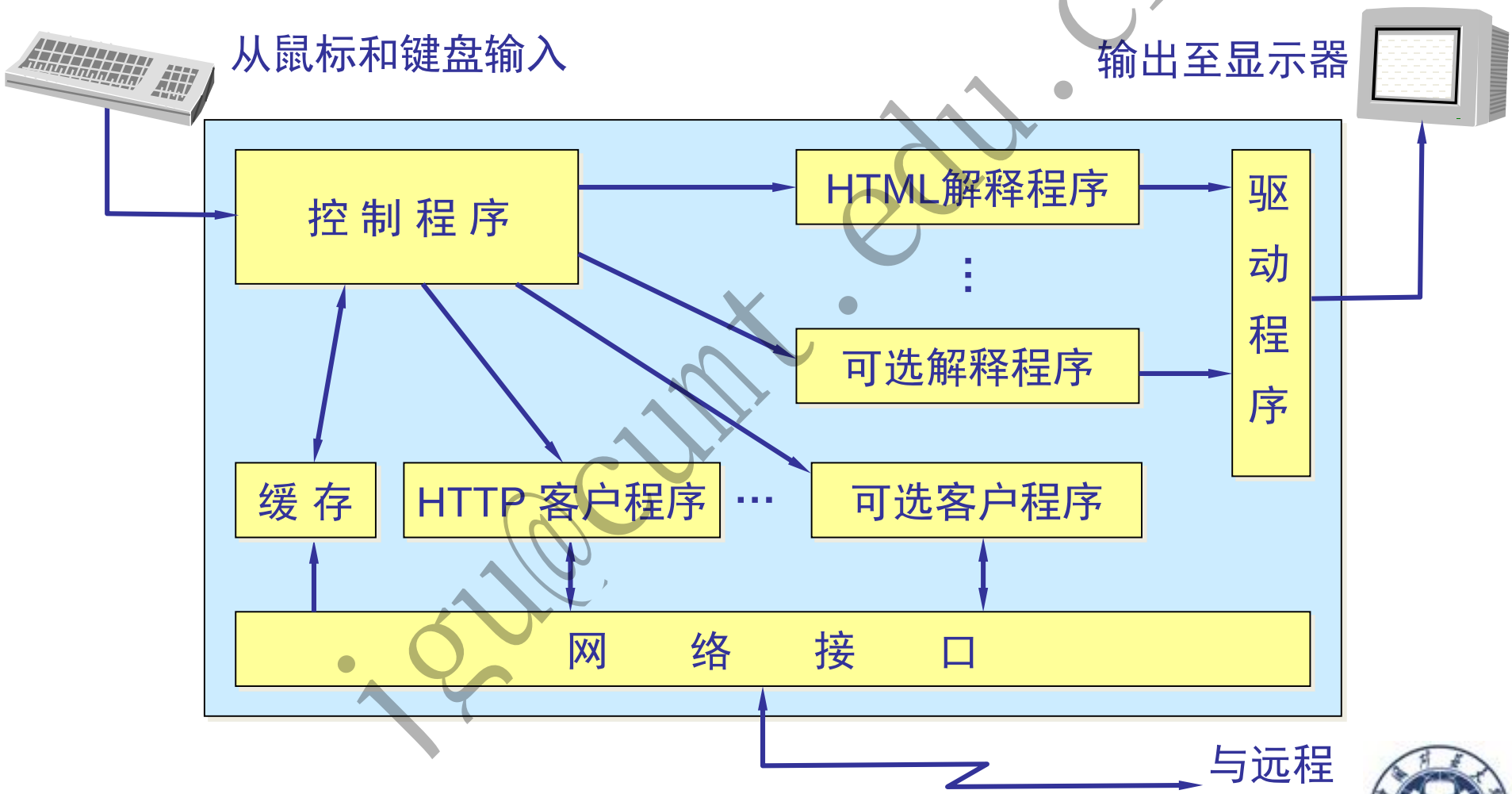


使用高速缓存的情况

(5) 高速缓存收到此对象后，先复制在其本地存储器中（为今后使用），然后再将该对象放在 HTTP 响应报文中，通过已建立的 TCP 连接，返回给请求该对象的浏览器。



Q22：浏览器是如何工作的？



浏览器中的缓存

- 浏览器将它取回的每一个页面副本都放入本地磁盘的缓存中。
- 当用户用鼠标点击某个选项时，浏览器首先检查磁盘的缓存。若缓存中保存了该项，浏览器就直接从缓存中得到该项副本而不必从网络获取，这样就明显地改善浏览器的运行特性。
- 但缓存要占用磁盘大量的空间，而浏览器性能的改善只有在用户再次查看缓存中的页面时才有帮助。
- 许多浏览器允许用户调整缓存策略。

Q23：动态万维网文档的实现？

- 静态文档是指该文档创作完毕后就存放在万维网服务器中，在被用户浏览的过程中，内容不会改变。

静态文档不能满足需求

- 随着时间的推移，人们发现静态的HTML着实无聊而乏味，增加动态生成的内容才会令Web应用程序变得更加有用。
- 于是，HTML的语法在不断膨胀
  - 最重要的是增加了表单（Form）；
  - 客户端增加了诸如脚本处理、DOM处理（文档对象模型，Document Object Model，一种用于HTML和XML文档的编程接口，给文档提供了一种结构化的表示方法，可以改变文档的内容和呈现方式）等功能；
  - 对于服务器，则相应的出现了CGI（Common Gateway Interface）以处理包含表单提交在内的动态请求。

动态文档

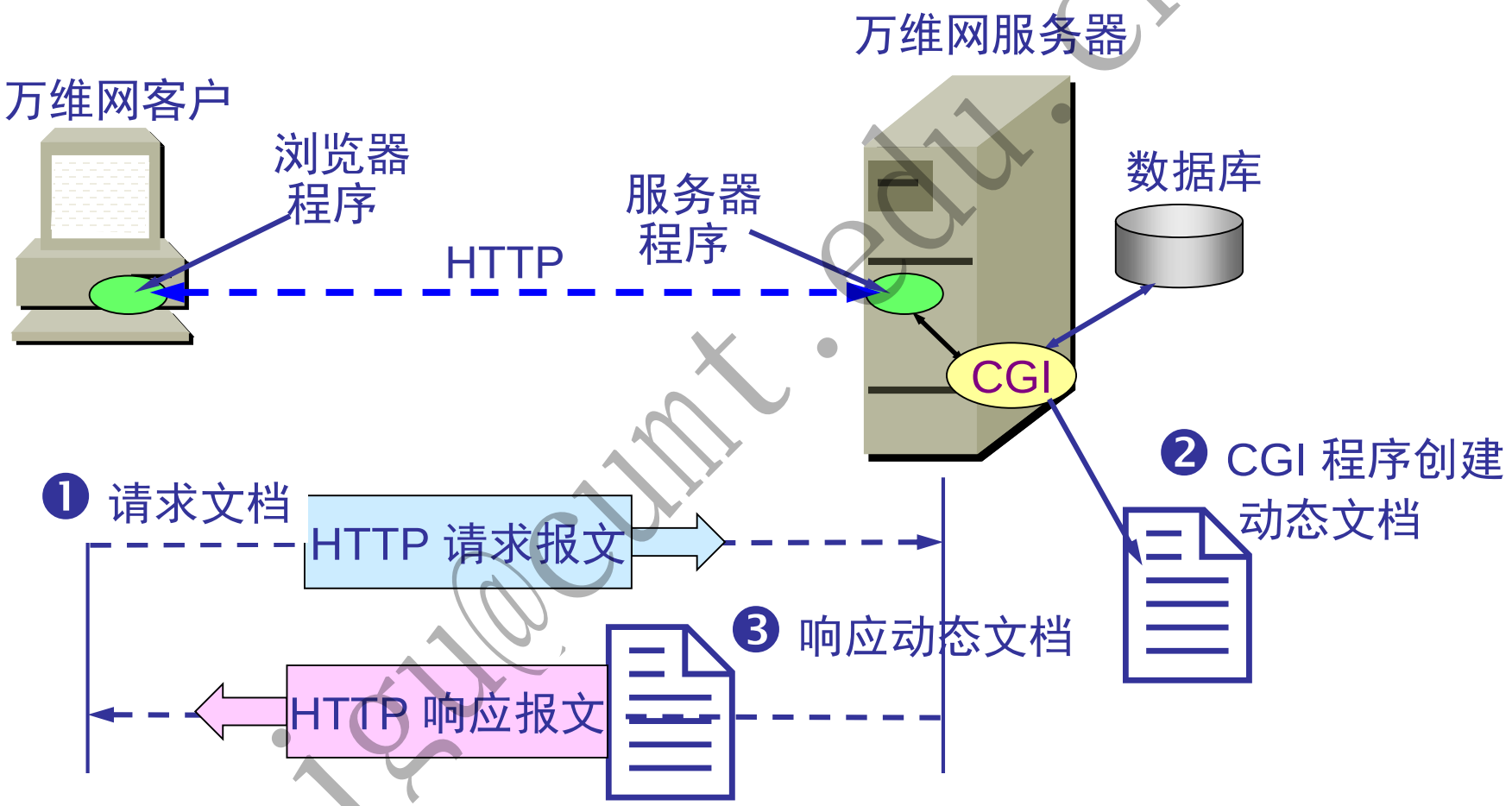
- 动态文档是指文档的内容是在浏览器访问万维网服务器时才由应用程序动态创建。
- 动态文档和静态文档之间的主要差别体现在服务器一端。这主要是文档内容的生成方法不同。而从浏览器的角度看，这两种文档并没有区别。

万维网服务器功能的扩充

(1) 应增加另一个应用程序，用来处理浏览器发来的数据，并创建动态文档。

(2) 应增加一个机制，用来使万维网服务器把浏览器发来的数据传送给这个应用程序，然后万维网服务器能够解释这个应用程序的输出，并向浏览器返回 HTML 文档。

扩充了功能的万维网服务器



通用网关接口 CGI (Common Gateway Interface)

- CGI 是一种标准，它定义了动态文档应如何创建，输入数据应如何提供给应用程序，以及输出结果应如何使用。
- 万维网服务器与 CGI 的通信遵循 CGI 标准。
- “通用”：CGI 标准所定义的规则对其他任何语言都是通用的。
- “网关”：CGI 程序的作用像网关。
- “接口”：有一些已定义好的变量和调用等可供其他 CGI 程序使用。

CGI 程序

- CGI 程序的正式名字是 CGI 脚本(script)。
- “脚本”指的是一个程序，它被另一个程序（解释程序）而不是计算机的处理机来解释或执行。
- 脚本运行起来要比一般的编译程序要慢，因为它的每一条指令先要被另一个程序来处理（这就要一些附加的指令），而不是直接被指令处理器来处理。

HTTP对状态信息的需求

- 在这种客户端与服务器进行动态交互的Web应用程序出现之后，HTTP无状态的特性严重阻碍了这些交互式应用程序的实现，毕竟交互是需要承前启后的，简单的购物车程序也要知道用户到底在之前选择了什么商品。
- 于是，两种用于保持HTTP状态的技术就应运而生了，一个是Cookie，而另一个则是Session。

Cookie

- Cookie是客户端的存储空间，用来表示在 HTTP 服务器和客户之间传递的状态信息，由浏览器来维持。
- 万维网站点使用 Cookie 来跟踪用户。使用 Cookie 的网站服务器为用户产生一个唯一的识别码，并以此作为索引在服务器的后端数据库中产生一个项目。
- 利用此识别码，网站就能够跟踪该用户在该网站的活动。

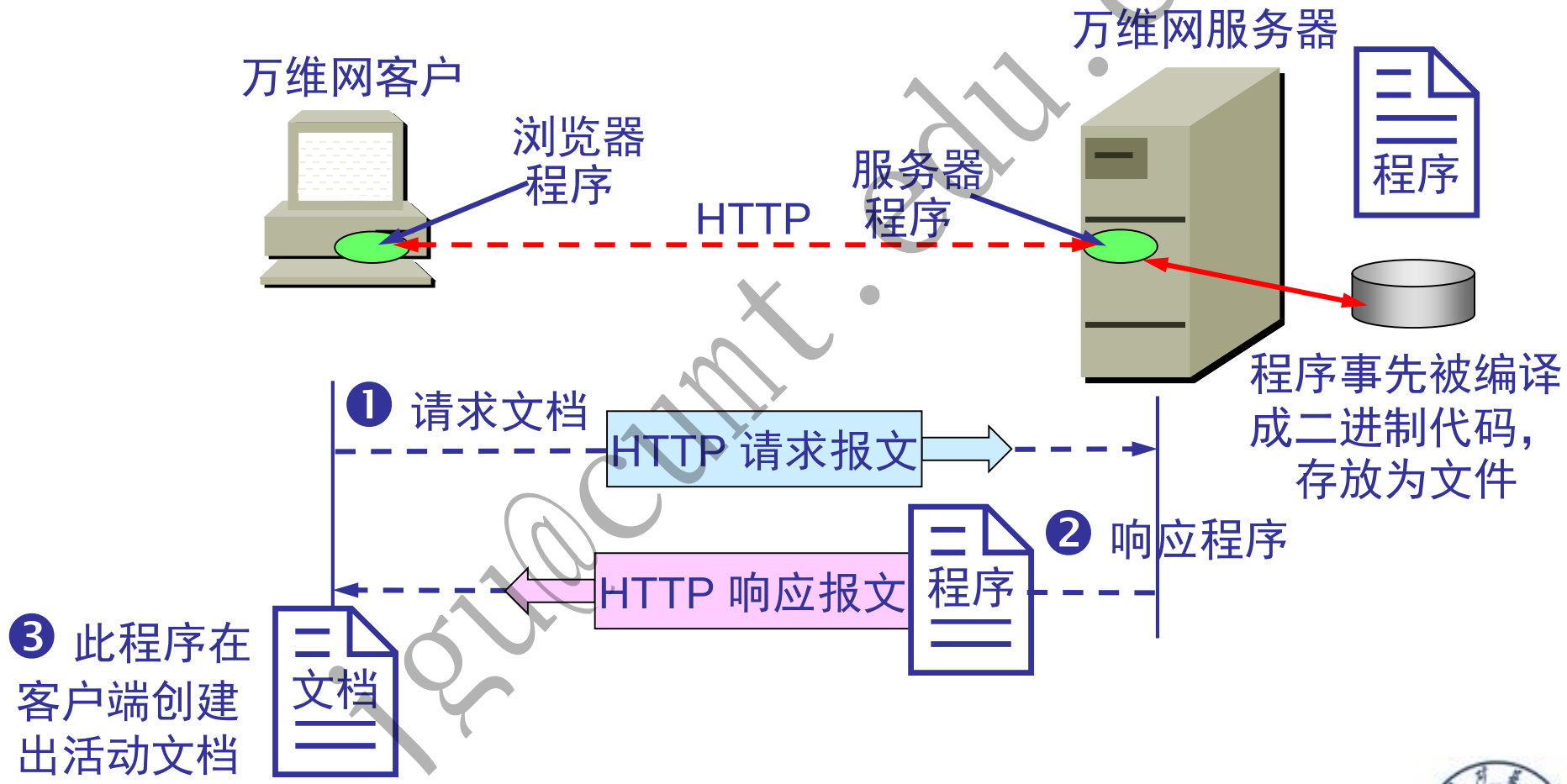
Session

- Session机制采用的是在服务器端保持状态的方案。
- 由于在服务器端保持状态的方案在客户端也需要保存一个标识，所以Session机制可能需要借助于Cookie机制来达到保存标识的目的，但实际上还有其他选择，比如说重写URL和隐藏表单域。

Q24：活动万维网文档的实现？

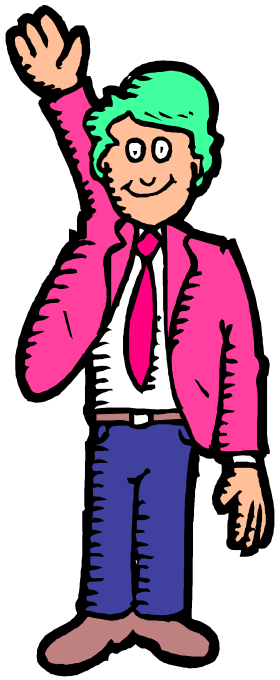
- 活动文档(active document)技术把所有的工作都转移给浏览器端。
- 每当浏览器请求一个活动文档时，服务器就返回一段程序副本在浏览器端运行。
- 活动文档程序可与用户直接交互，并可连续地改变屏幕的显示。
- 由于活动文档技术不需要服务器的连续更新传送，对网络带宽的要求也不会太高。

活动文档在客户端创建



用 Java 技术创建活动文档

- 由美国 Sun 公司开发的 Java 语言是一项用于创建和运行活动文档的技术。
- 在 Java 技术中使用 “小应用程序”(applet)来描述活动文档程序。
- 用户从万维网服务器下载嵌入了 Java 小应用程序的 HTML 文档后，可在浏览器的屏幕上点击某个图像，就可看到动画效果，或在下拉式菜单中点击某个项目，就可看到计算结果。
- Java 技术是活动文档技术的一部分。



THANK YOU!