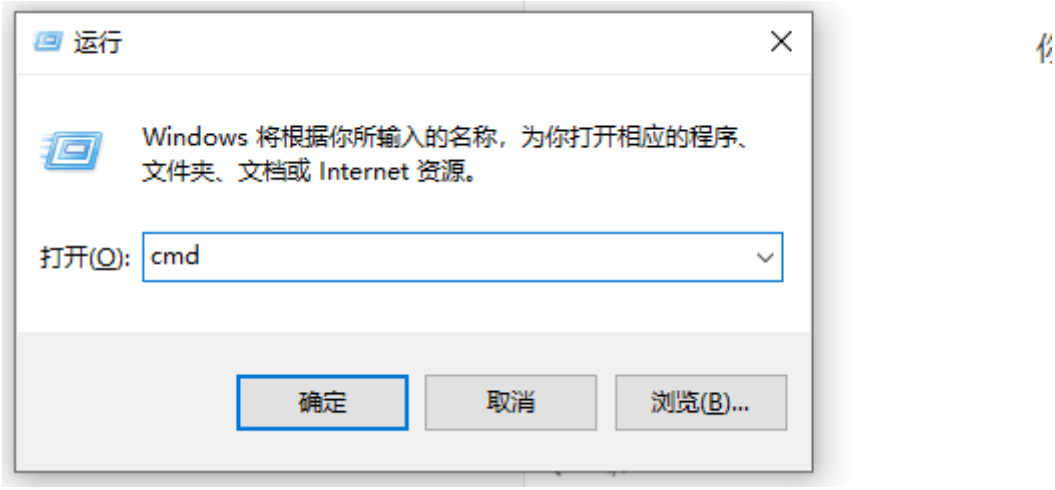
网络入门

一、基本概念

1、mac地址、ip地址

window电脑在命令行模式下输入命令：

怎么进入命令行模式，按　win+R，输入cmd即可




```
ipconfig -all
```

会看到：

```
无线局域网适配器 WLAN:

   描述. . . . . . . . . . . . . . . : Intel(R) Wi-Fi 6 AX201 160MHz
   物理地址. . . . . . . . . . . . . : 8C-C6-81-54-FF-9C
   IPv4 地址 . . . . . . . . . . . . : 192.168.0.109(首选)
   子网掩码 . . . . . . . . . . . . : 255.255.255.0
```

其中物理地址指的就是mac地址、IPv4 地址就是ip。

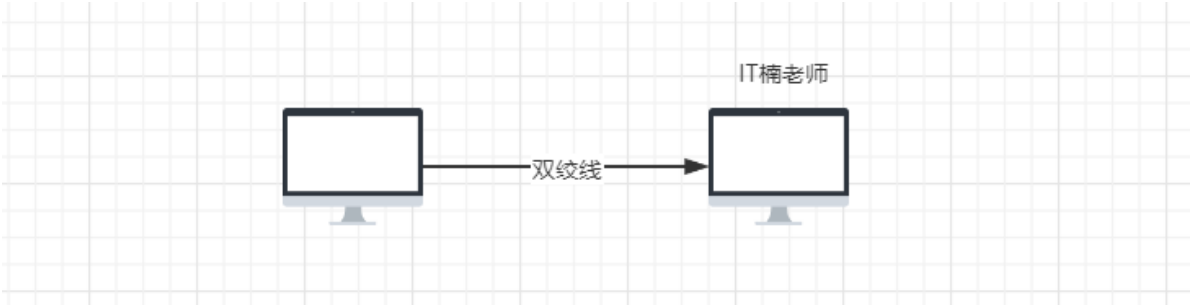
mac地址也叫物理地址和局域网地址，主要用于确认网上设备的地址，类似于身份证号，具有唯一标识，每一个网卡制作完成之后就带有一个mac地址，永远都不会改变。

ip地址，类似于你的现住址，是标记你在网络中的具体位置，一个网卡的ip地址是可以改变的。

2、计算机之间是怎么发送数据的

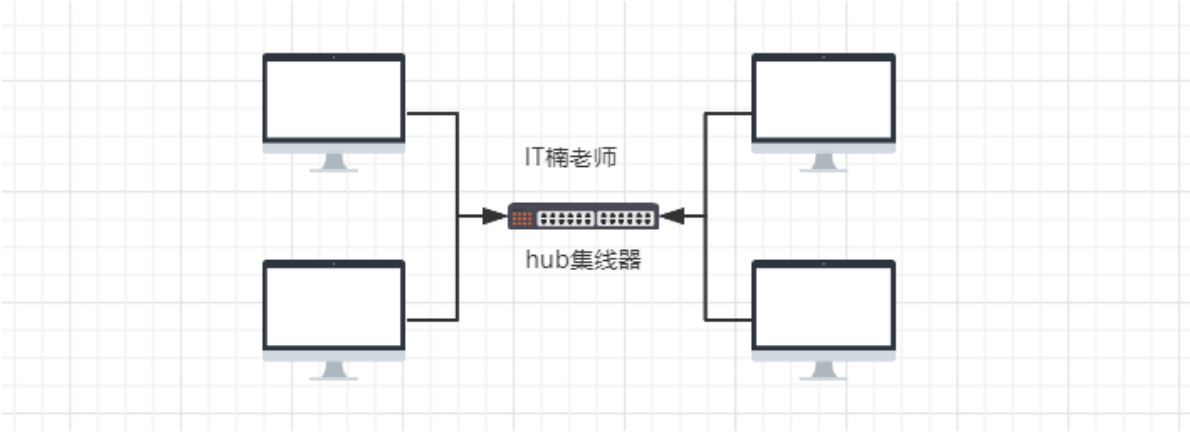
2.1 双绞线

如果只是两台计算机，我能就可以使用双绞线（网线）连载一起，就能互相发送消息，组成一个小网络。



2.2 集线器

如果有多台计算机，可以使用hub，叫做集线器，一个电脑发送信息到集线器，集线器负责广播到其他的计算机。

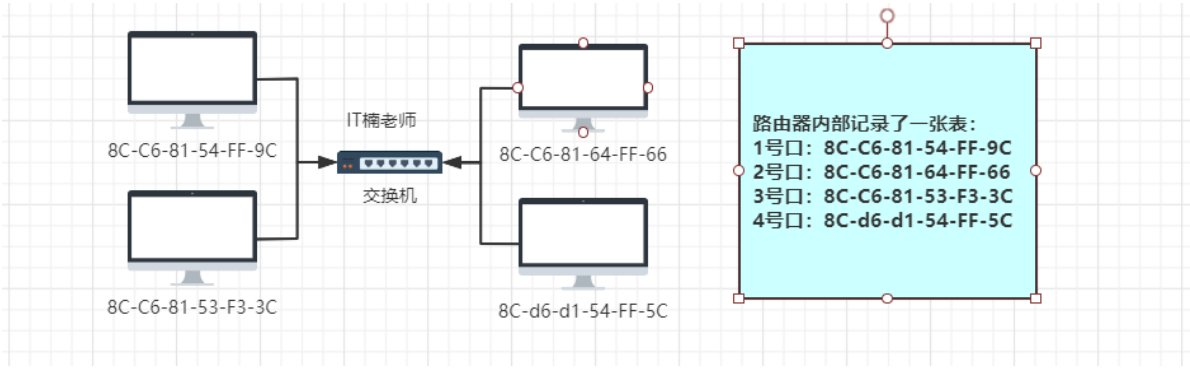


2.3 交换机

由于集线器的问题，我们经常需要发送信息到特定的计算机而不是广播，所以一个新的设备就出现了叫做交换机（switch）。

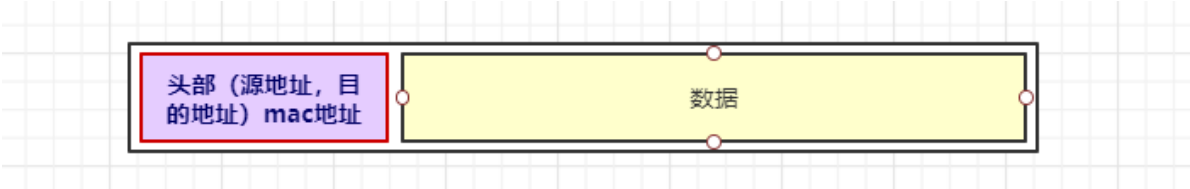
交换机可以记录每一个设备的弟子和接口的对应关系。

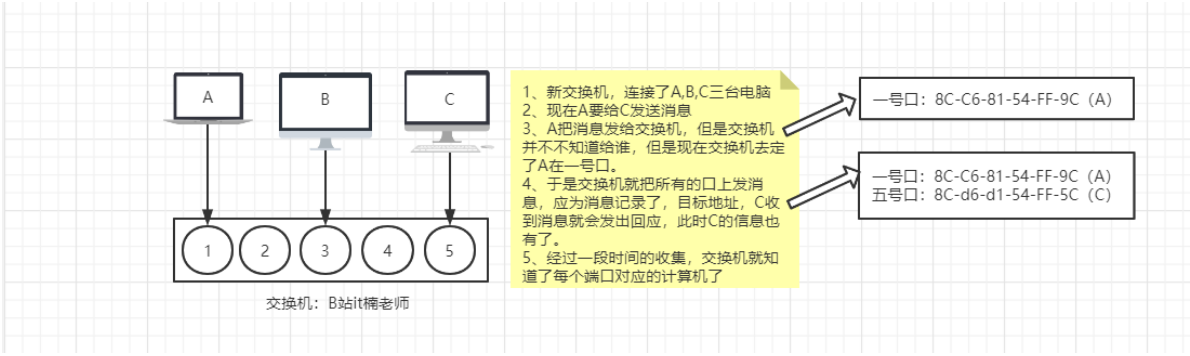
思考问题，交换机要将内容发送给指定的计算机，那么内部一定维护了一张表，记录了哪个电脑链接了我的哪个口。交换机只能识别MAC地址。MAC地址是物理地址，ip地址交换机并不感兴趣。



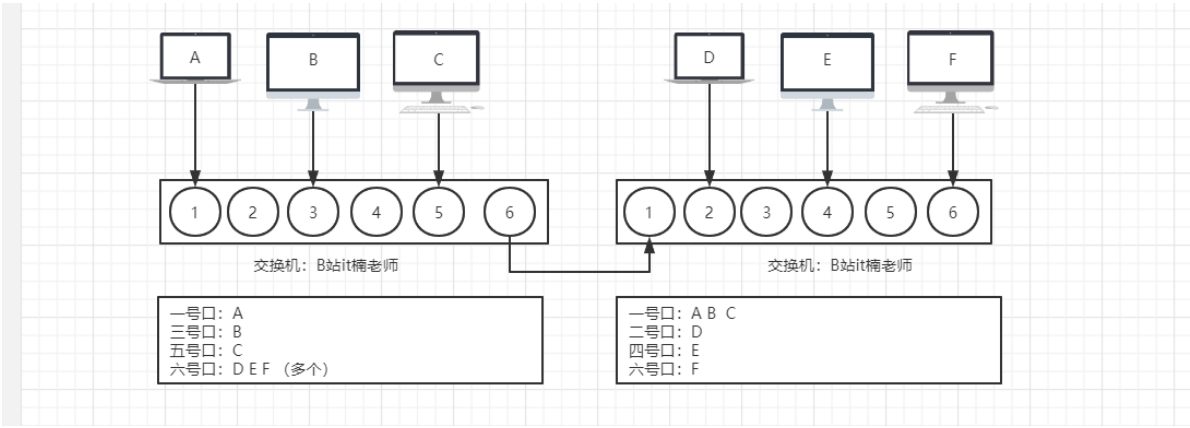
思考一个问题：

交换机啊是怎么知道这个表的：





交换机效率比较高，而且可以进行桥接。



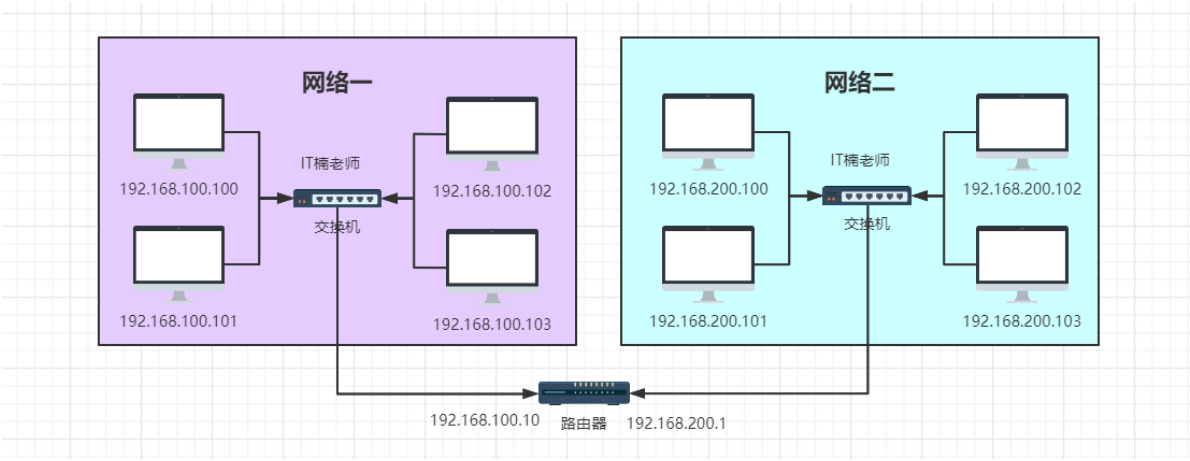
2.4 路由器

思考：使用交换机可以简历一个超大型的网络吗？

一般的交换机的地址表也就能存个几千个地址，当网络内的设备多起来以后，只要交换机找不到对应设备就会广播，地址表如果满了，新地址还会覆盖就地址就会导致重新寻找效率比较低。所以有引入了一个设备叫路由器，谁也听过的一个设备，一般家里都有。

注意：路由器不是猫，猫是调制解调器，调制解调器的作用是将进来的光信号转化为电信号

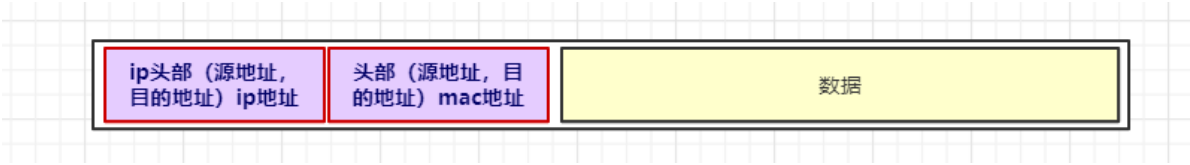
于是提出了以下的设计



这里就有了网络的概念了。以上的几种，哪怕是交换机的桥接也没有设计ip地址这个概念，都是基于mac地址进行数据传输。这里有了网络这个抽象概念之后ip地址就应用而生了，IP地址只要是用来表示计算机的网络位置，他处于哪一个网络。IP地址和子网掩码共同帮助我们定位一个计算机在网络中的位置。

ip地址和子网掩码其实是个32位的二进制数字

此时发送信息就会再包一个消息的头部



家用的路由器，有个wan口，有好几个lan口，wan口用来来接互联网端，lan用来连接家庭设备，这连个口都有一个网卡，一个网卡属于互联网网络，ip可能是10.25.23.65，另一个属于内部网络比如192.168.0.1。内部网络和外部互联网的数据转发由路由器内部实现。

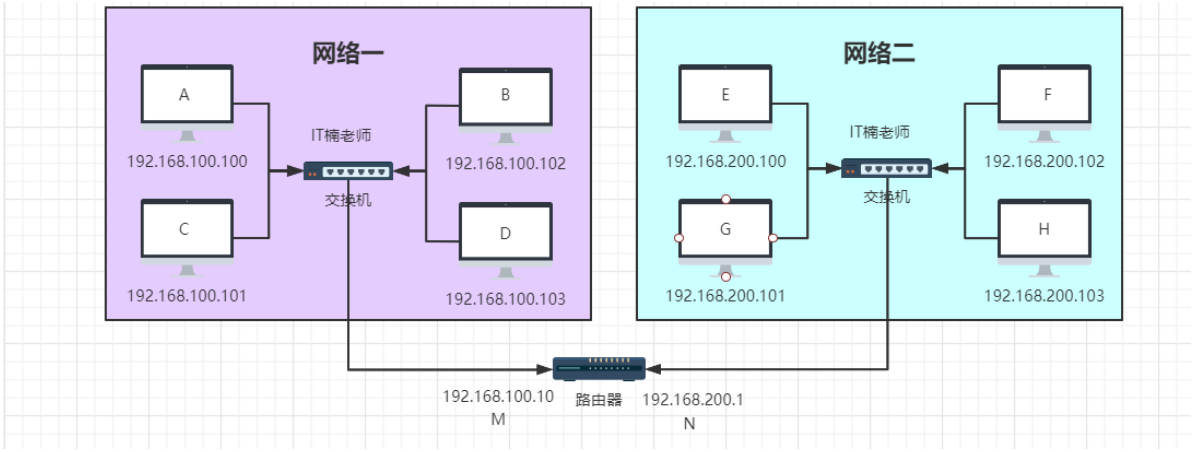
路由器内有路由表。

网络内传输，有了网关的概念了。

ip不能直接进行传输，应为网络内的交换机不支持ip地址，所以通信要转化为mac地址，根据ip查找mac地址。

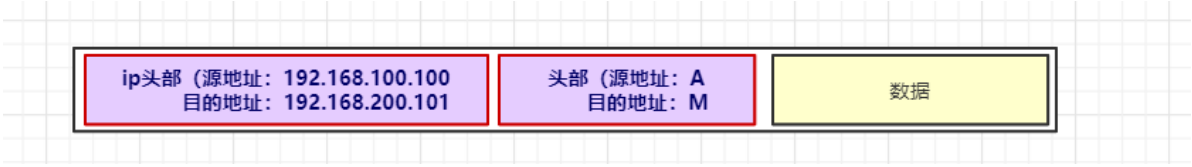
非常重要的arp协议，要进行广播，问一问哪个mac地址的ip是192.168.0.5，他收到就会回应，

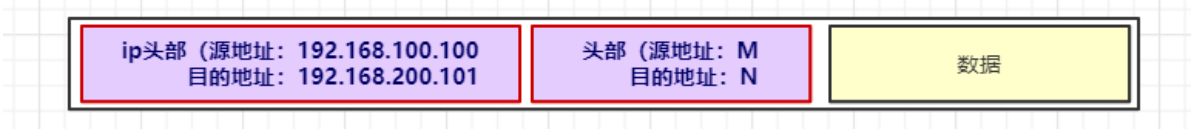
来一个栗子，192.168.100.100 发送信息到 192.168.200.101

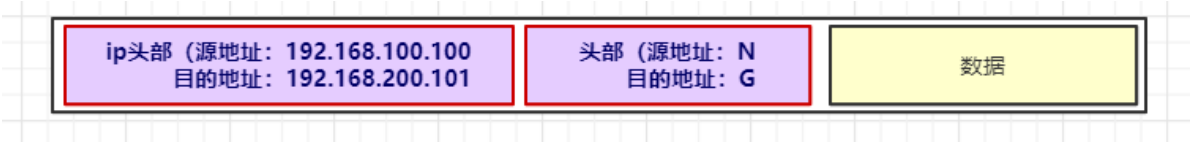


第一步：

192.168.100.100 -> 192.168.100.10







添加一条路由

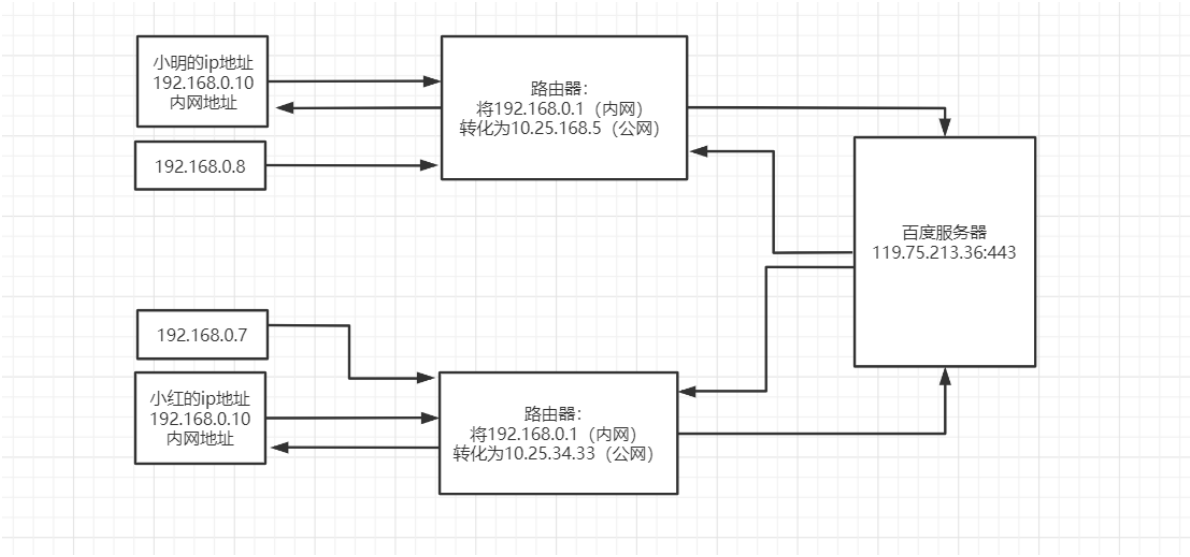
```
route add -host 192.168.110.202 gw 192.168.120.202
route del -host 192.168.110.202 gw 192.168.120.202
```

```
tcpdump -i eth1 host 192.168.1.123 -w /tmp/xxx.cap
```

ICMP（Internet Control Message Protocol）Internet控制报文协议。它是TCP/IP协议簇的一个子协议，用于在IP主机、路由器之间传递控制消息。控制消息是指网络通不通、主机是否可达、路由是否可用等网络本身的消息。这些控制消息虽然并不传输用户数据，但是对于用户数据的传递起着重要的作用。

2.5 NAT协议

每台设备都分配一个ip当然能够实现网络的建立，但是事实上我们只有数量有限的IP地址，能给我们用的大概只有40亿左右，于是一般的做法是家里只有一个ip地址，所有的设备通过路由器连接。连接在内网。那么问题来了，你的手机的地址是192.168.0.10，别人家的也可能是一样的百度怎么区分呀？



这个叫NAT协议 *NAT*（Network Address Translation，网络地址转换）。

现在家里的路由器都是带有交换功能的

能使用的IP地址 25.68亿 共有40多亿

47.111.225.204

3、ip地址的分类

IP地址分类：

1、A类IP地址

一个A类IP地址由1字节的网络地址和3字节主机地址组成，网络地址的最高位必须是“0”， 地址范围从1.0.0.0 到126.0.0.0。可用的A类网络有126个，每个网络能容纳1亿多个主机。

2、B类IP地址

一个B类IP地址由2个字节的网络地址和2个字节的主机地址组成，网络地址的最高位必须是“10”，地址范围从128.0.0.0到191.255.255.255。可用的B类网络有16382个，每个网络能容纳6万多个主机 。

3、C类IP地址

一个C类IP地址由3字节的网络地址和1字节的主机地址组成，网络地址的最高位必须是“110”。范围从192.0.0.0到223.255.255.255。C类网络可达209万余个，每个网络能容纳254个主机。

4、D类地址

用于多点广播（Multicast）。 D类IP地址第一个字节以“lll0”开始，它是一个专门保留的地址。它并不指向特定的网络，目前这一类地址被用在多点广播（Multicast）中。多点广播地址用来一次寻址一组计算机，它标识共享同一协议的一组计算机。224.0.0.0到239.255.255.255用于多点广播 。

5、E类IP地址

以“llll0”开始，为将来使用保留。240.0.0.0到255.255.255.254，255.255.255.255用于广播地址，全零（“0．0．0．0”）地址对应于当前主机。全“1”的IP地址（“255．255．255．255”）是当前子网的广播地址。

IP私有地址：

在IP地址3种主要类型里，各保留了3个区域作为私有地址，其地址范围如下：

A类地址：10.0.0.0～10.255.255.255

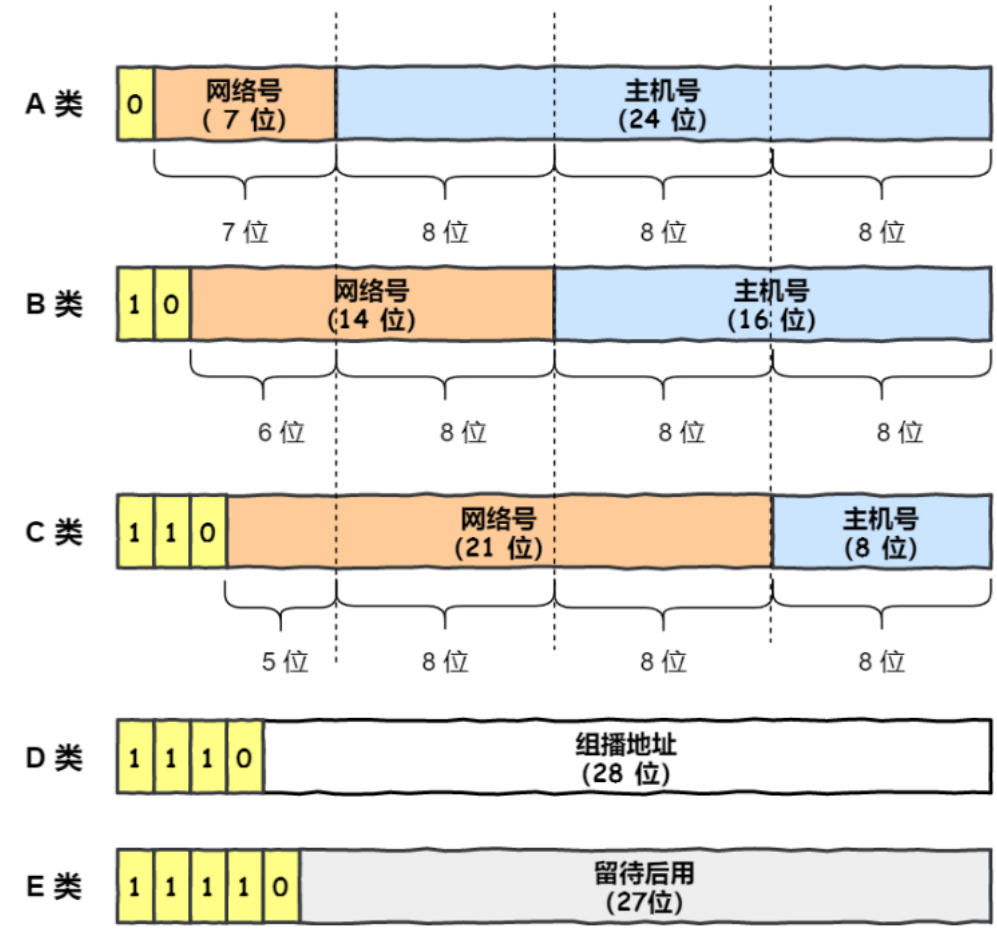
B类地址：172.16.0.0～172.31.255.255

C类地址：192.168.0.0～192.168.255.255

A类地址的第一组数字为1～126。注意，数字0和 127不作为A类地址，数字127保留给内部回送函数，而数字0则表示该地址是本地宿主机，不能传送。

B类地址的第一组数字为128～191。

C类地址的第一组数字为192～223。



上图中黄色部分为分类号，用以区分 IP 地址类别。

4、域名：

你真的能记住每个网站的ip地址吗?

那我们输入的www.itnanls.cn是什么是域名呀

我们可以去阿里云等云服务提供商购买域名。

记录类型：

A- 将域名指向一个IPV4地址

主机记录：

www .itnanls.cn

解析线路：

默认 - 必填！未匹配到智能解析线路时，返回【默认】线路设置结果

* 记录值：

123.56.54.32

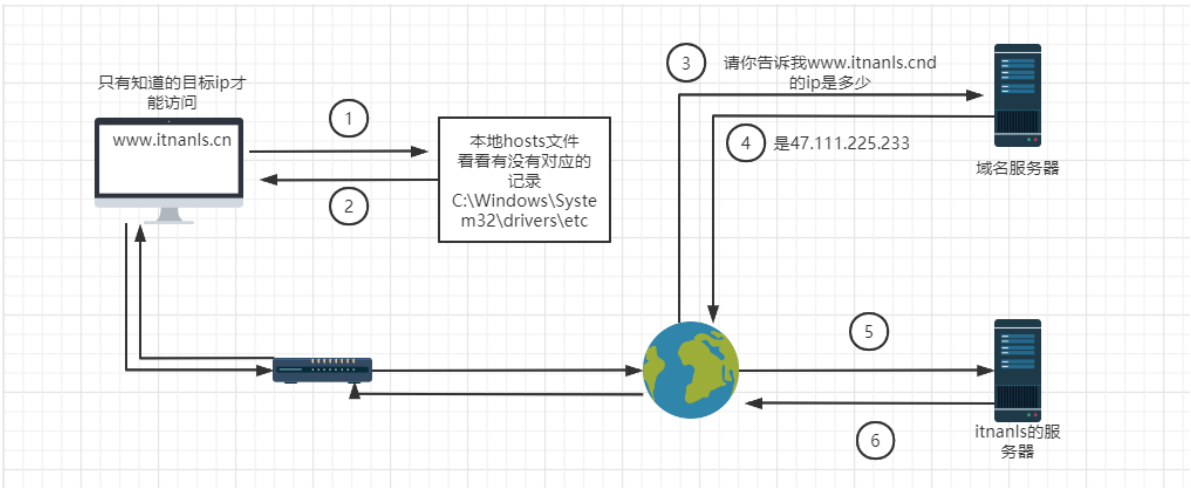
* TTL：

10 分钟

域名是需要解析的，指向一个ip地址。

为了让你容易记忆引入了域名的概念，你不需要记忆，有人帮您记录

DNS解析



4.1 国内的dns

第一名
114DNS：★★★★★
114DNS开启DNS高可靠服务时代的大幕。114DNS开始同时为公众提供高速、稳定、可信的DNS递归解析服务；为网站提供强大抗攻击能力的权威智能DNS解析服务；为ISP提供可靠的DNS灾备及外包服务，作为国内用户量最大的老牌DNS，访问速度快，各地区设有节点，负载各运营商用户，DNS防劫持能力，自然也是名列前茅。
DNS 服务器 IP 地址：
首选：114.114.114.114

备选：114.114.114.115
2017公共DNS服务器地址评估—DNS推荐
第二名
DNSPod DNS：★★★★★
DNSPod创始于2006年3月，是中国最大的第三方域名服务商，全球排名第四位。DNSPod是国内最早提供免费智能DNS产品的网站,致力于为各类网站提供高质量的电信、网通、教育网双线或者三线智能DNS免费解析，作为114DNS竞争对手之一，无论是访问速度，还是各地区节点覆盖率以及防劫持能力都是顶级的。
DNS 服务器 IP 地址：
首选：119.29.29.29
备选：182.254.116.116
2017公共DNS服务器地址评估—DNS推荐
第三名
阿里 DNS：★★★★★
阿里公共DNS是阿里巴巴集团推出的DNS递归解析系统，作为国内最大的互联网基础服务提供商，阿里巴巴在继承多年优秀技术的基础上，通过提供性能优异的公共DNS服务，为广大互联网用户提供最可靠的面向互联网用户提供“快速”、“稳定”、“智能”的免费DNS递归解析服务。
DNS 服务器 IP 地址：
首选：223.5.5.5
备选：223.6.6.6
2017公共DNS服务器地址评估—DNS推荐

4.2 域名的分类

1. **语种分类：**

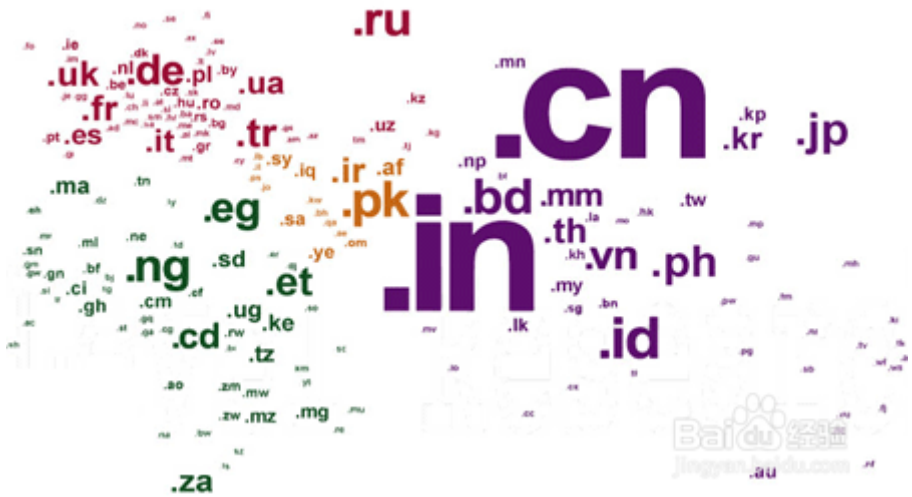
   英文域名、中文域名、日文域名、韩文域名等等。

   现在我们常见的就是英文域名。

   英文域名：baidu.com

   中文域名：百度.com 、百度.中国、baidu.中国

   其它语言域名同上。



2. **地区分类**

   中国、美国、英国、日本等等。

   在国内我们常用的就是.cn域名，.cn是中国大陆的国家一级域名。另外还有.com.cn、.net.cn、.org.cn等等。

   国内不同省市也有自己的顶级域名，例如内蒙古的顶级域名就是：.nm.cn。

美国国家顶级域名是.us 、日本的是.jp、香港的是.hk。

3. **机构分类：**

   .com 商业性的机构或公司

   .org 非盈利的组织、团体 apache.org

   .gov 政府部门 http://www.shanxi.gov.cn/

   .net 从事Internet相关的的机构或公司

   域名虽然有很多分类，但是我们平时在使用的时候，并没有过多的遵循这个原则。比如.com的也有很多作为个人网站、.net很多也用来做了公司网站。

4.3、层级分类

1. **顶级域名（一级域名）：**

   baidu.com

   baidu.cn

2. **二级域名:**

   www.baidu.com

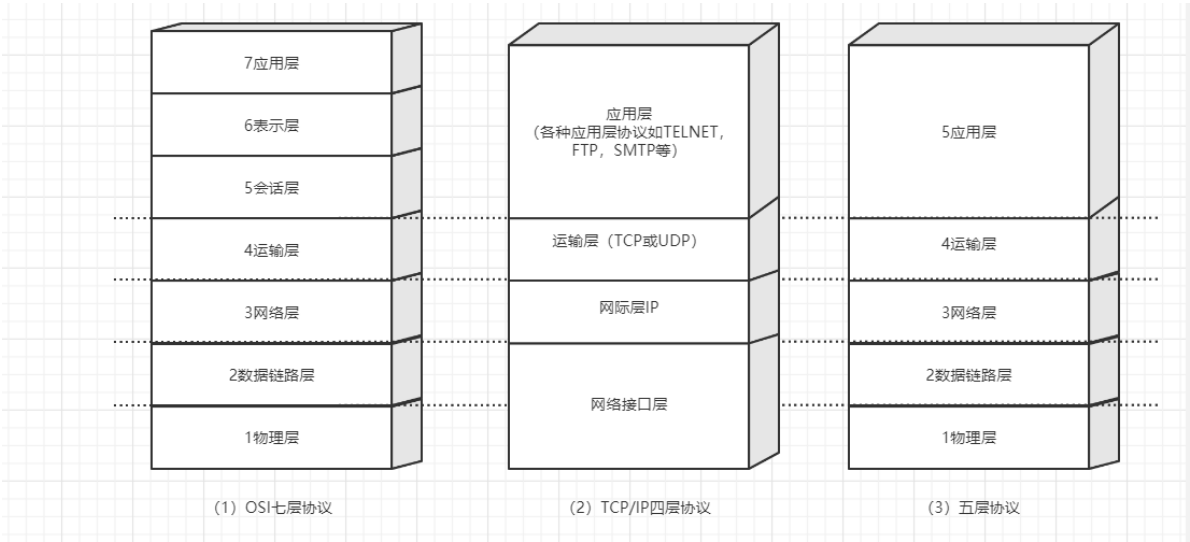
   jingyan.baidu.com

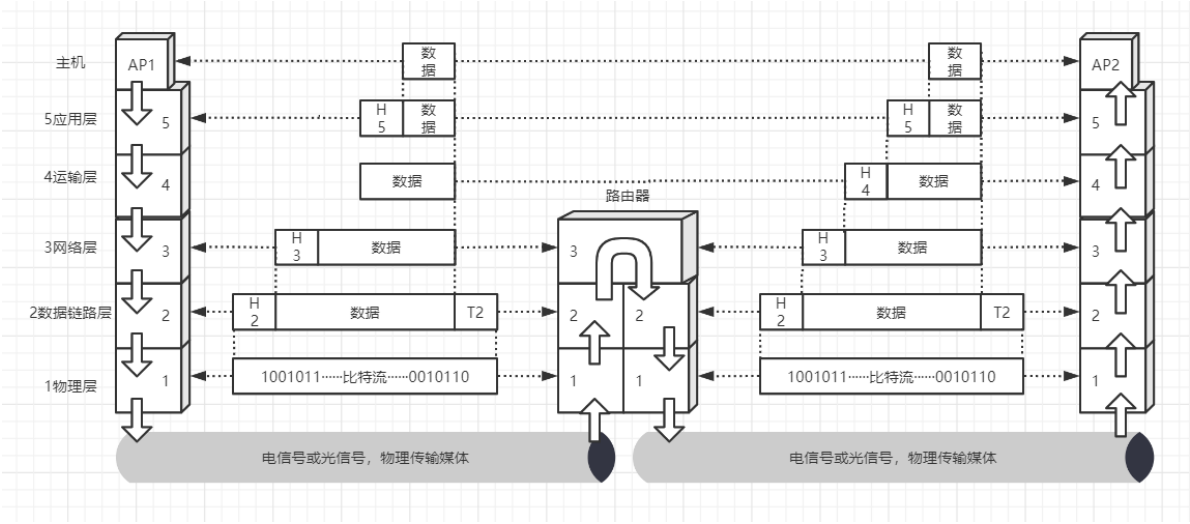
3. **三级域名:**

   wangshangyingxiao.club.1688.com

   hhz.bbs.53hui.com

二、网络七层协议：





OSI层	功能	设备	对应TCP/IP协议
应用层	用户接口、应用程序(文件传输，电子邮件，文件服务，虚拟终端)	网关	TFTP，HTTP，SNMP，FTP，SMTP，DNS，Telnet
表示层	数据的表示、压缩和加密（数据格式化，代码转换，数据加密）	网关	没有协议
会话层	会话的建立和结束（解除或建立与别的接点的联系）	网关	没有协议
传输层	提供端对端的接口	网关	TCP，UDP
网络层	为数据包选择路由，寻址	路由器	IP，ICMP，RIP，OSPF，BGP，IGMP
数据链路层	保证误差错的数据链路，传输有地址的帧以及错误检测功能	交换机、网桥、网卡	SLIP，CSLIP，PPP，ARP，RARP，MTU
物理层	传输比特流，以二进制数据形式在物理媒体上传输数据	集线器、中继器	ISO2110，IEEE802，IEEE802.2

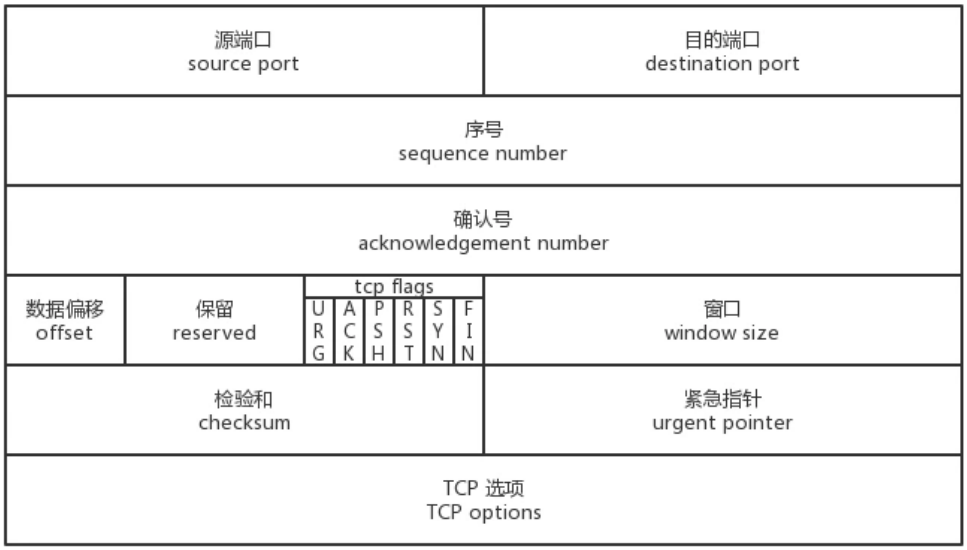
你在浏览器输入www.baidu.com 发生了什么

了解了网络是怎么链接的

三、数据是怎么传输的

1、tcp协议

传输依靠什么进行控制



源端口和目的端口字段

- TCP源端口（Source Port）：源计算机上的应用程序的端口号，占 16 位。
- TCP目的端口（Destination Port）：目标计算机的应用程序端口号，占 16 位。

序列号字段

CP序列号（Sequence Number）：占 32 位。它表示本报文段所发送数据的第一个字节的编号。在 TCP 连接中，所传送的字节流的每一个字节都会按顺序编号。当SYN标记不为1时，这是当前数据分段第一个字母的序列号；如果SYN的值是1时，这个字段的值就是初始序列值（ISN），用于对序列号进行同步。这时，第一个字节的序列号比这个字段的值大1，也就是ISN加1。

确认号字段

TCP 确认号（Acknowledgment Number，ACK Number）：占 32 位。它表示接收方期望收到发送方下一个报文段的第一个字节数据的编号。其值是接收计算机即将接收到的下一个序列号，也就是下一个接收到的字节的序列号加1。

数据偏移字段

TCP 首部长度（Header Length）：数据偏移是指数据段中的“数据”部分起始处距离 TCP 数据段起始处的字节偏移量，占 4 位。其实这里的“数据偏移”也是在确定 TCP 数据段头部分的长度，告诉接收端的应用程序，数据从何处开始。

保留字段

保留（Reserved）：占 4 位。为 TCP 将来的发展预留空间，目前必须全部为 0。

标志位字段

- CWR（Congestion Window Reduce）：拥塞窗口减少标志，用来表明它接收到了设置 ECE 标志的 TCP 包。并且，发送方收到消息之后，通过减小发送窗口的大小来降低发送速率。
- ECE（ECN Echo）：用来在 TCP 三次握手时表明一个 TCP 端是具备 ECN 功能的。在数据传输过程中，它也用来表明接收到的 TCP 包的 IP 头部的 ECN 被设置为 11，即网络线路拥堵。
- URG（Urgent）：表示本报文段中发送的数据是否包含紧急数据。URG=1 时表示有紧急数据。当 URG=1 时，后面的紧急指针字段才有效。
- ACK：表示前面的确认号字段是否有效。ACK=1 时表示有效。只有当 ACK=1 时，前面的确认号字段才有效。TCP 规定，连接建立后，ACK 必须为 1。
- PSH（Push）：告诉对方收到该报文段后是否立即把数据推送给上层。如果值为 1，表示应当立即把数据提交给上层，而不是缓存起来。
- RST：表示是否重置连接。如果 RST=1，说明 TCP 连接出现了严重错误（如主机崩溃），必须释放连接，然后再重新建立连接。

- SYN：在建立连接时使用，用来同步序号。当 SYN=1，ACK=0 时，表示这是一个请求建立连接的报文段；当 SYN=1，ACK=1 时，表示对方同意建立连接。SYN=1 时，说明这是一个请求建立连接或同意建立连接的报文。只有在前两次握手中 SYN 才为 1。
- FIN：标记数据是否发送完毕。如果 FIN=1，表示数据已经发送完成，可以释放连接。

窗口大小字段

窗口大小（Window Size）：占 16 位。它表示从 Ack Number 开始还可以接收多少字节的数据量，也表示当前接收端的接收窗口还有多少剩余空间。该字段可以用于 TCP 的流量控制。

TCP 校验和字段

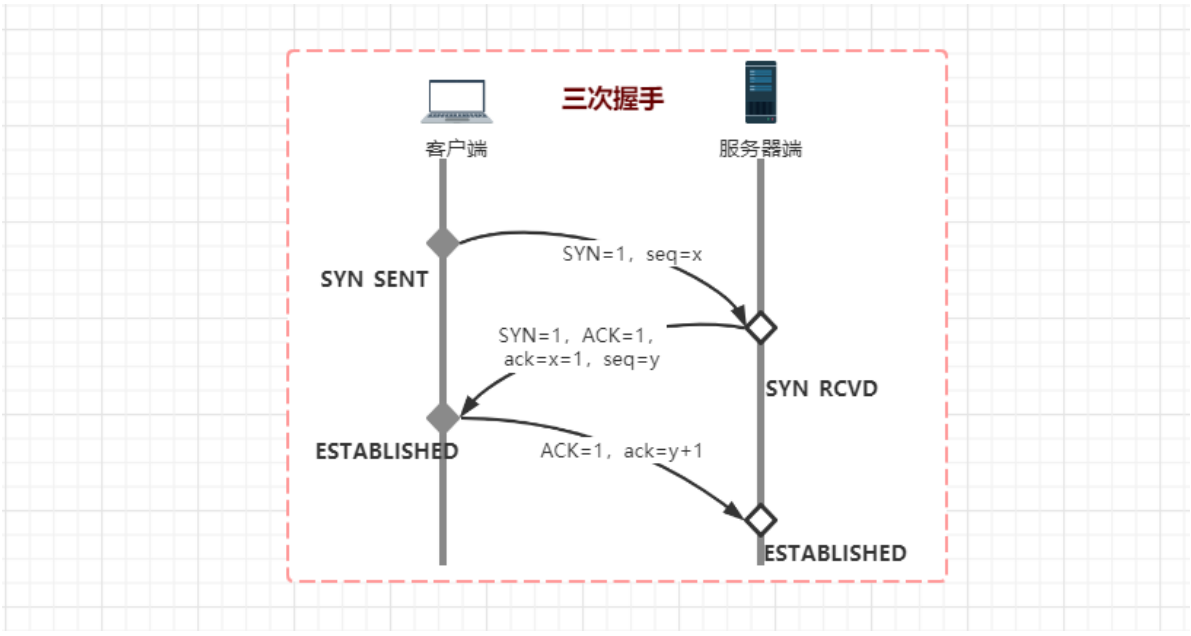
校验位（TCP Checksum）：占 16 位。它用于确认传输的数据是否有损坏。发送端基于数据内容校验生成一个数值，接收端根据接收的数据校验生成一个值。两个值必须相同，才能证明数据是有效的。如果两个值不同，则丢掉这个数据包。Checksum 是根据伪头 + TCP 头 + TCP 数据三部分进行计算的。

紧急指针字段

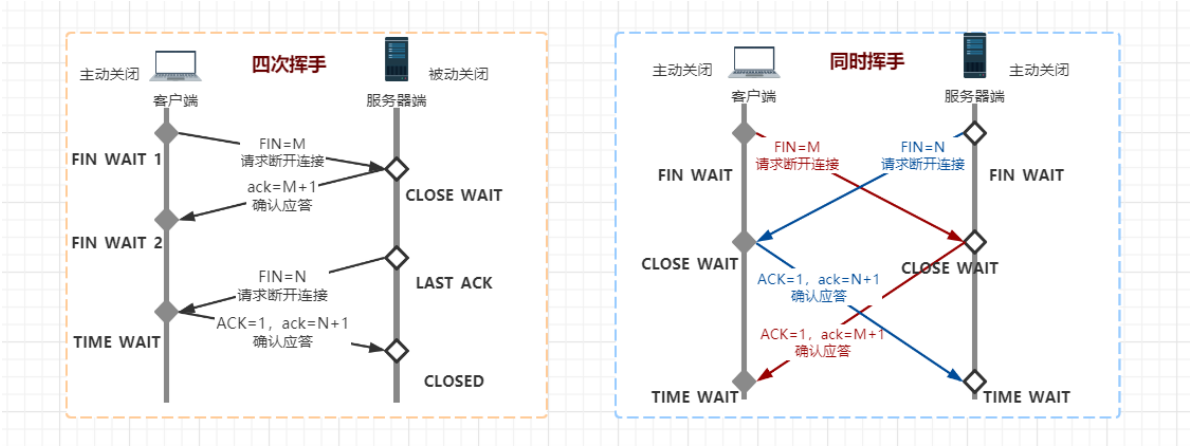
紧急指针（Urgent Pointer）：仅当前面的 URG 控制位为 1 时才有意义。它指出本数据段中为紧急数据的字节数，占 16 位。当所有紧急数据处理完后，TCP 就会告诉应用程序恢复到正常操作。即使当前窗口大小为 0，也是可以发送紧急数据的，因为紧急数据无须缓存。

可选项字段

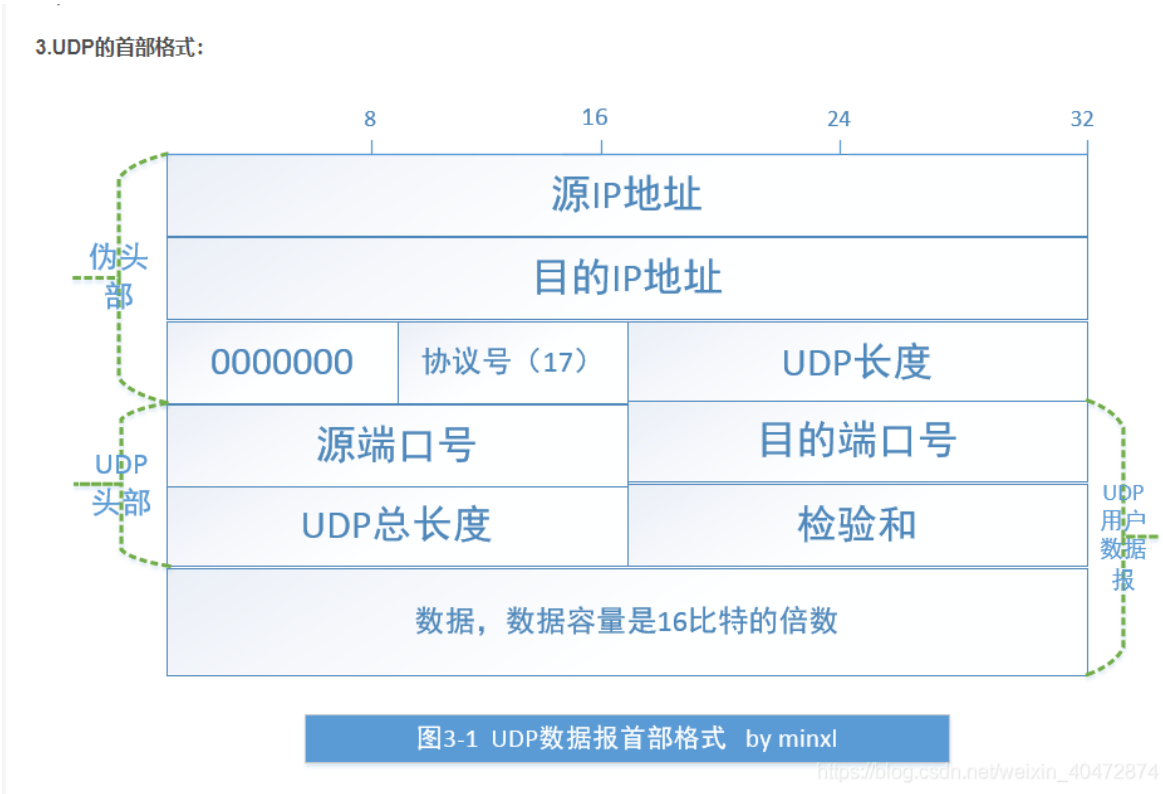
选项（Option）：长度不定，但长度必须是 32bits 的整数倍。



No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	192.168.120.1	192.168.120.200	TCP	66	51138 → 8080 [SYN] Seq=0 Win=64240 Len=0 MSS=1460 WS=256 SACK_PERM=1
2	0.000157	192.168.120.200	192.168.120.1	TCP	66	8080 → 51138 [SYN, ACK] Seq=0 Ack=1 Win=29200 Len=0 MSS=1460 SACK_PERM=1 WS=128
3	0.000235	192.168.120.1	192.168.120.200	TCP	54	51138 → 8080 [ACK] Seq=1 Ack=1 Win=1051136 Len=0



2、udp协议



讲解一下：
*UDP头部很简单，包括源端口，目的端口，UDP总长度，校验和，各占16位/2字节，共8字节。
*源端口：长度16位，指定发送方所使用的端口号，若不需要对方回发消息，则可全置为0。
*目的端口：长度16位，指定接收方所使用的端口号。
*UDP总长度：长度16位，指定了UDP数据报的总长度。
*校验和：长度16位，用于UDP的差错检测，防止UDP报文出错，同时伪首部参与计算，避免UDP用户数据报传送到错误的目的地。UDP的首部，数据部分，伪首部都会参与检验和的计算，各字段是按照16比特为单位进行计算的，因此数据部分是要保证是16比特的倍数，不够用0填充。

3、http协议

1、HTTP协议简介

超文本传输协议（英文：**H**yper**T**ext **T**ransfer **P**rotocol，缩写：HTTP）是一种用于分布式、协作式和超媒体信息系统的**应用层协议。**HTTP是万维网的数据通信的基础。

HTTP的发展是由蒂姆·伯纳斯-李于1989年在欧洲核子研究组织（CERN）所发起。HTTP的标准制定由万维网协会（World Wide Web Consortium，W3C）和互联网工程任务组（Internet Engineering Task Force，IETF）进行协调，最终发布了一系列的RFC，其中最著名的是1999年6月公布的 RFC 2616，定义了HTTP协议中现今广泛使用的一个版本——HTTP 1.1。

2014年12月，互联网工程任务组（IETF）的Hypertext Transfer Protocol Bis（httpbis）工作小组将HTTP/2标准提议递交至IESG进行讨论，于2015年2月17日被批准。 HTTP/2标准于2015年5月以RFC 7540正式发表，取代HTTP 1.1成为HTTP的实现标准。

2、HTTP协议概述

HTTP是一个客户端终端（用户）和服务器端（网站）**请求和应答**的标准（TCP）。

通过使用网页浏览器或者其它的工具，客户端发起一个HTTP请求到服务器上指定端口（默认端口为80）。

我们称这个客户端为用户代理程序（user agent）。

应答的服务器上存储着一些资源，比如HTML文件和图像。我们称这个应答服务器为源服务器（origin server）。

通常，由HTTP客户端发起一个请求，**创建一个到服务器指定端口（默认是80端口）的连接。**HTTP服务器则**在那个端口监听客户端的请求。**一旦收到请求，服务器会向客户端返回一个状态，比如"HTTP/1.1 200 OK"，以及返回的内容，如请求的文件、错误消息、或者其它信息。

3、HTTP工作原理

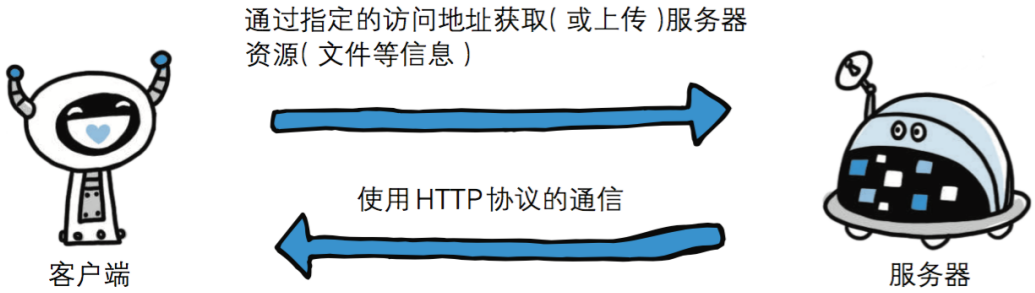
HTTP协议定义Web客户端如何从Web服务器请求Web页面，以及服务器如何把Web页面传送给客户端。HTTP协议采用了**请求/响应模型。**客户端向服务器发送一个请求报文，请求报文包含请求的方法、URL、协议版本、请求头部和请求数据。服务器以一个状态行作为响应，响应的内容包括协议的版本、成功或者错误代码、服务器信息、响应头部和响应数据。

以下是 HTTP 请求/响应的步骤：

1. 客户端连接到Web服务器
   浏览器向 DNS 服务器请求解析该 URL 中的域名所对应的 IP 地址，一个HTTP客户端，通常是浏览器，与Web服务器的HTTP端口（默认为80）建立一个TCP套接字连接。例如，http://www.luffycity.com。
2. 发送HTTP请求
   通过TCP套接字，客户端向Web服务器发送一个文本的请求报文，一个请求报文由**请求行、请求头部、空行和请求数据**4部分组成。
3. 服务器接受请求并返回HTTP响应
   Web服务器**解析请求，定位请求资源。**服务器将资源复本写到TCP套接字，由客户端读取。一个响应由**状态行、响应头部、空行和响应数据**4部分组成。
4. 释放连接TCP连接
   若connection 模式为close，则服务器主动关闭TCP连接，客户端被动关闭连接，释放TCP连接;若connection 模式为keepalive，则该连接会保持一段时间，在该时间内可以继续接收请求;

5. 客户端浏览器解析HTML内容

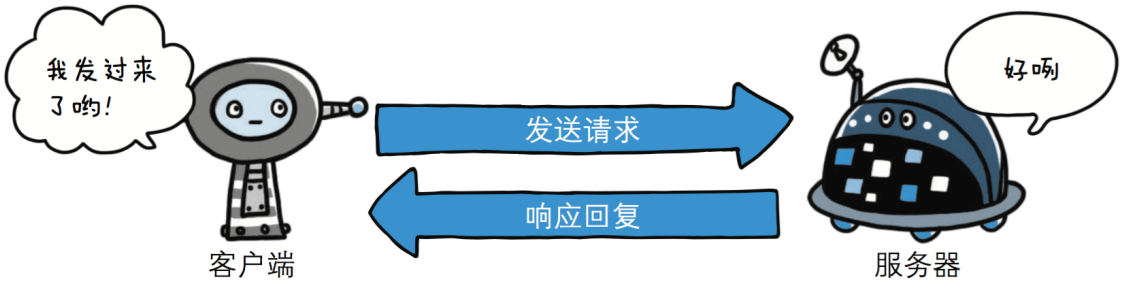
客户端浏览器首先解析状态行，查看表明请求是否成功的状态代码。然后解析每一个响应头，响应头告知以下为若干字节的HTML文档和文档的字符集。客户端浏览器读取响应数据HTML，根据HTML的语法对其进行格式化，并在浏览器窗口中显示。



http协议是基于TCP/IP协议之上的应用层协议。

基于 请求-响应 的模式

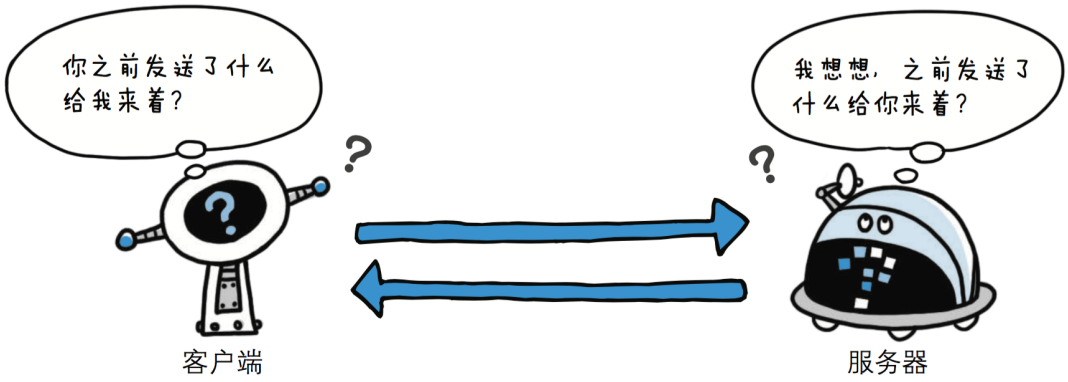
HTTP协议规定,请求从客户端发出,最后服务器端响应该请求并 返回。换句话说,肯定是先从客户端开始建立通信的,服务器端在没有 接收到请求之前不会发送响应，**服务端不能主动说话。**



图：请求必定由客户端发出，而服务器端回复响应

无状态

HTTP是一种不保存状态,即无状态(stateless)协议。HTTP协议 自身不对请求和响应之间的通信状态进行保存。也就是说在HTTP这个 级别,协议对于**发送过的请求或响应都不做持久化处理。**



图：HTTP 协议自身不具备保存之前发送过的请求或响应的功能

使用HTTP协议,每当有新的请求发送时,就会有对应的新响应产 生。协议本身并不保留之前一切的请求或响应报文的信息。这是为了更快地处理大量事务,确保协议的可伸缩性,而特意把HTTP协议设计成 如此简单的。可是,随着Web的不断发展,因无状态而导致业务处理变得棘手 的情况增多了。比如,用户登录到一家购物网站,即使他跳转到该站的 其他页面后,也需要能继续保持登录状态。针对这个实例,网站为了能 够掌握是谁送出的请求,需要保存用户的状态。HTTP/1.1虽然是无状态协议,但为了实现期望的保持状态功能, 于是**引入了Cookie技术。有了Cookie再用HTTP协议通信,就可以管 理状态了。**有关Cookie的详细内容稍后讲解。

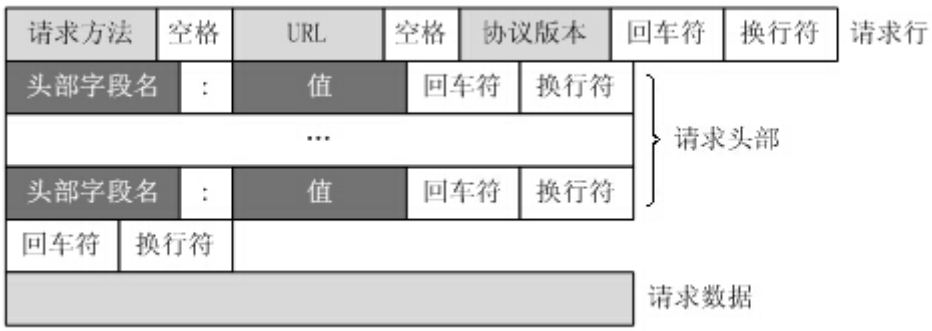
无连接

无连接的含义是限制每次连接只处理一个请求。服务器处理完客户的请求，并收到客户的应答后，即断开连接。采用这种方式可以节省传输时间，并且可以提高并发性能，不能和每个用户建立长久的连接，请求一次相应一次，服务端和客户端就中断了。但是无连接有两种方式，早期的http协议是一个请求一个响应之后，直接就断开了，但是现**在的http协议1.1版本不是直接就断开了，而是等几秒钟，这几秒钟是等什么呢，等着用户有后续的操作，如果用户在这几秒钟之内有新的请求，那么还是通过之前的连接通道来收发消息，如果过了这几秒钟用户没有发送新的请求，那么就会断开连接，这样可以提高效率，减少短时间内建立连接的次数，因为建立连接也是耗时的，默认的好像是3秒中现在**，但是这个时间是可以通过咱们后端的代码来调整的，自己网站根据自己网站用户的行为来分析统计出一个最优的等待时间。

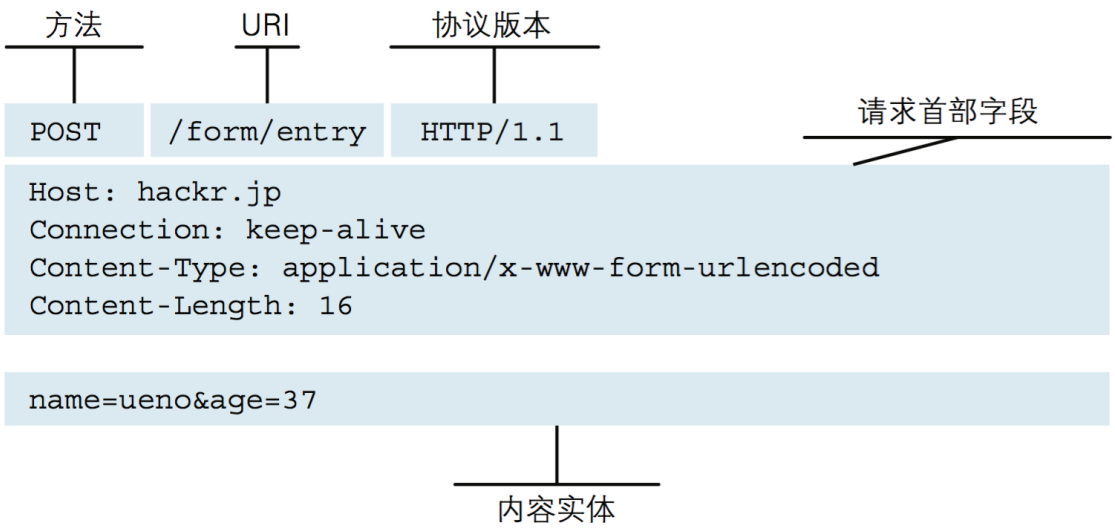
> **来个总结 http 是一种基于请求和响应的无状态、无连接的协议**

4、HTTP请求格式(请求协议)

其实就是个字符串：

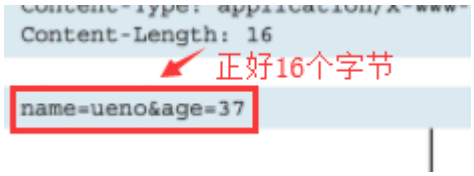


URL包含：/index/index2?a=1&b=2；路径和参数都在这里。

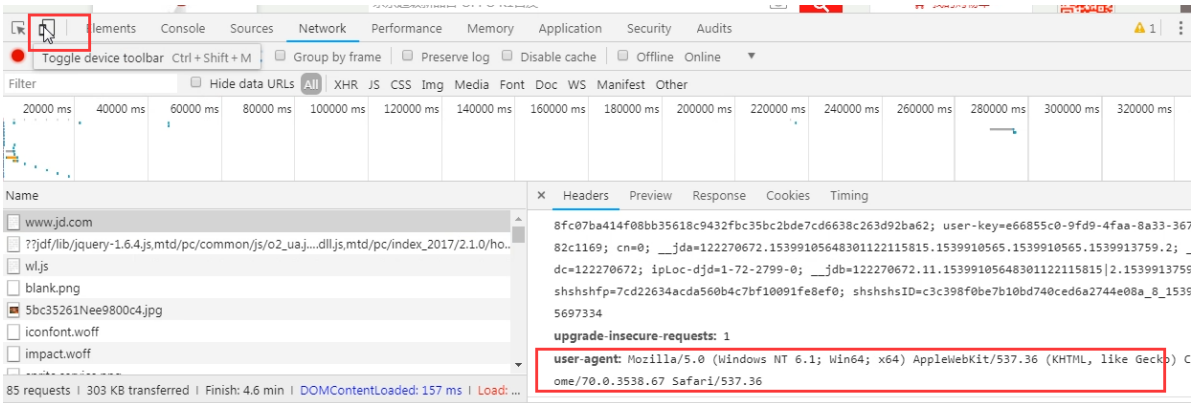


图：请求报文的构成

请求头里面的内容举个例子：这个length表示请求体里面的数据长度，其他的请求头里面的这些键值对，陆续我们会讲的，大概知道一下就可以了，其中有一个user-agent，算是需要你记住的吧，就是告诉你的服务端，我是用什么给你发送的请求，是个啥客户端。



以京东为例，看一下user-agent



看一个爬虫的例子，爬京东的时候没问题，但是爬抽屉的时候必须带着user-agent，因为抽屉对user-agent做了判断，来判断你是不是一个正常的请求，算是反扒机制的一种。

```
import requests

res=requests.get("https://dig.chouti.com/",
                 headers={
                         "user-agent": "Mozilla/5.0 (iPad; CPU OS 1
                         }
                 )

with open("demo.html","w",encoding="utf8") as f:
    f.write(res.text)
```

打开我们保存的demo.html文件，然后通过浏览器打开看看就能看到页面效果。

写上面这些内容的意思是让你知道有这么个请求头的存在，有些是有意义的，请求头我们还可以自己定义，就在requests模块里面那个headers={}，这个字典里面加就行。

5、HTTP请求方法

HTTP/1.1协议中共定义了八种方法（也叫“动作”）来以不同方式操作指定的资源：

（1）GET

向指定的资源发出“显示”请求。使用GET方法应该只用在读取数据，而不应当被用于产生“副作用”的操作中，常用语查询数据的请求。

（2）POST

向指定资源提交数据，请求服务器进行处理（例如提交表单或者上传文件）。数据被包含在请求本文中。这个请求可能会创建新的资源或修改现有资源，或二者皆有。常用于对数据的增删改操作。

> 请求方式: get与post请求（通过form表单我们自己写写看）

- GET提交的数据会放在URL之后，也就是请求行里面，以?分割URL和传输数据，参数之间以&相连，如EditBook?name=test1&id=123456.（请求头里面那个content-type做的这种参数形式，后面讲） POST方法是把提交的数据放在HTTP包的请求体中.
- GET提交的数据大小有限制（因为浏览器对URL的长度有限制），而POST方法提交的数据没有限制.
- GET与POST请求在服务端获取请求数据方式不同，就是我们自己在服务端取请求数据的时候的方式不同了，这句废话昂。

6、URL

超文本传输协议（HTTP）的统一资源定位符将从因特网获取信息的五个基本元素包括在一个简单的地址中：

- 传送协议。
- 层级URL标记符号(为[//],固定不变)
- 访问资源需要的凭证信息（可省略）
- 服务器。（通常为域名，有时为IP地址）
- 端口号。（以数字方式表示，若为HTTP的默认值“:80”可省略）
- 路径。（以“/”字符区别路径中的每一个目录名称）
- 查询。（GET模式的窗体参数，以“?”字符为起点，每个参数以“&”隔开，再以“=”分开参数名称与数据，通常以UTF8的URL编码，避开字符冲突的问题）
- 片段。以“#”字符为起点

```
以http://www.xinzhi.com:80/news/index.html?id=250&page=1 为例，其中：

http，是协议；
www.xinzhi.com，是服务器；
80，是服务器上的默认网络端口号，默认不显示；
/news/index.html，是路径（URI：直接定位到对应的资源）；
?id=250&page=1，是查询条件。
大多数网页浏览器不要求用户输入网页中“[http://”的部分，因为绝大多数网页内容是超文本传输协议文件。
“80”是超文本传输协议文件的常用默认端口号，因此一般也不必写明。一般来说用户只要键入统一资源定位符的一部分
```

7、请求中常见的ContentType

就是告诉服务器，我给你发了个参数，参数是什么类型，你应该怎么接受，怎么解析。

> application/x-www-form-urlencoded

这应该是最常见的 POST 提交数据的方式了。浏览器的原生 form 表单，如果不设置 enctype 属性，那么最终就会以 application/x-www-form-urlencoded 方式提交数据。请求类似于下面这样（无关的请求头在本文中都省略掉了）

格式： name='lisi'&age=13

> application/json

application/json 这个 Content-Type 作为响应头大家肯定不陌生。实际上，现在越来越多的人把它作为请求头，用来告诉服务端消息主体是序列化后的 JSON 字符串。由于 JSON 规范的流行，除了低版本 IE 之外的各大浏览器都原生支持 JSON.stringify，服务端语言也都有处理 JSON 的函数，使用 JSON 不会遇上什么麻烦。

格式： {“name”:"lisi","age":13}

> multipart/form-data

这又是一个常见的 POST 数据提交的方式。我们使用表单上传文件时，必须让 form 的 enctyped 等于这个值。直接来看一个请求示例：

8、HTTP响应格式（响应协议）

图：响应报文的构成

9、HTTP状态码

所有HTTP响应的第一行都是状态行，依次是当前HTTP版本号，3位数字组成的状态代码，以及描述状态的短语，彼此由空格分隔。

状态代码的第一个数字代表当前响应的类型：

- 1xx消息——请求已被服务器接收，继续处理
- 2xx成功——请求已成功被服务器接收、理解、并接受
- 3xx重定向——需要后续操作才能完成这一请求
- 4xx请求错误——请求含有词法错误或者无法被执行，客户端
- 5xx服务器错误——服务器在处理某个正确请求时发生错误，500

虽然 RFC 2616 中已经推荐了描述状态的短语，例如"200 OK"，"404 Not Found"，但是WEB开发者仍然能够自行决定采用何种短语，用以显示本地化的状态描述或者自定义信息。

	类别	原因短语
1XX	Informational（信息性状态码）	接收的请求正在处理
2XX	Success（成功状态码）	请求正常处理完毕
3XX	Redirection（重定向状态码）	需要进行附加操作以完成请求
4XX	Client Error（客户端错误状态码）	服务器无法处理请求
5XX	Server Error（服务器错误状态码）	服务器处理请求出错

8、响应中常见的contentType

text/html

告诉浏览器，我给你返回的是html，请渲染

```
application/json
```

告诉浏览器，我给你返回的是json数据，一般会让js去处理

常见的contenttype对照表

https://tool.oschina.net/commons/

10、一些常见的响应头：

常见的请求头对照表

http://tools.jb51.net/table/http_header

```
Server: Apache-Coyote/1.1：服务器的版本信息；
Content-Type: text/html;charset=UTF-8：响应体使用的编码为UTF-8；
Content-Length: 724：响应体为724字节；
Set-Cookie: JSESSIONID=C97E2B4C55553EAB46079A4F263435A4; Path=/hello：响应给客户端的Cookie；
Date: Wed, 25 Sep 2012 04:15:03 GMT：响应的时间，这可能会有8小时的时区差；

```

```
Last-Modified：最后的修改时间；
If-Modified-Since：把上次请求的index.html的最后修改时间还给服务器；

//告诉浏览器不要缓存：
Expires: -1；
Cache-Control: no-cache；
Pragma: no-cache；
**自动刷新**响应头，浏览器会在3秒之后请求http://www.itcast.cn：
Refresh: 3;url=http://www.itcast.cn
```

```
Request URL: http://127.0.0.1:8888/user?username=23423&password=4234
Request Method: GET
Status Code: 200 OK
Remote Address: 127.0.0.1:8888
Referrer Policy: no-referrer-when-downgrade
Content-Length: 1422
Content-type: text/html;charset=utf-8
Accept: text/html,application/xhtml+xml,application/xml;q=0.9,image/webp,image/apng,*/*;q=0.8,application/signed-exchange;v=b3;q=0.9
Accept-Encoding: gzip, deflate, br
Accept-Language: zh-CN,zh;q=0.9,en;q=0.8,en-GB;q=0.7,en-US;q=0.6
Connection: keep-alive
Host: 127.0.0.1:8888
Referer: http://127.0.0.1:8888/index    //告诉服务器这个请求从哪里来
Sec-Fetch-Dest: document
Sec-Fetch-Mode: navigate
Sec-Fetch-Site: same-origin
Sec-Fetch-User: ?1
Upgrade-Insecure-Requests: 1
User-Agent: Mozilla/5.0 (Windows NT 10.0; Win64; x64) AppleWebKit/537.36 (KHTML, like Gecko) Chrome/80.0.3987.163 Safari/537.36 Edg/80.0.361.111
username: 23423
password: 4234
```

```
POST /index HTTP/1.1
Content-Type: application/x-www-form-urlencoded
User-Agent: PostmanRuntime/7.24.1
Accept: */*
Cache-Control: no-cache
Postman-Token: b976a185-e71b-4ef0-bd3a-58bf61eec48a
Host: 127.0.0.1:8888
Accept-Encoding: gzip, deflate, br
Connection: keep-alive
Content-Length: 21

name=zhangsna&age=123&nam=1
```

的，