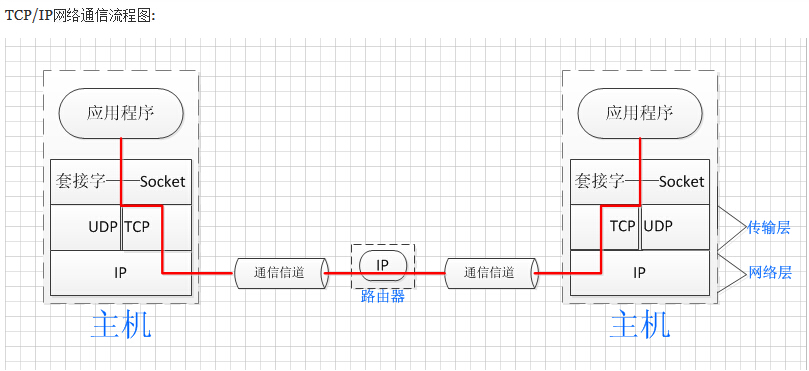
### ****0，计算机网络****

**计算机网络**由一组通过**通信信道(Communication channel)**相互连接的机器组成。这些机器被称为:**主机(hosts)**和**路由器(routers)**:

**通信信道**  ——  是将字节序列从一个主机传输到另一个主机的一种手段(有线、无线(WiFi)等方式)。

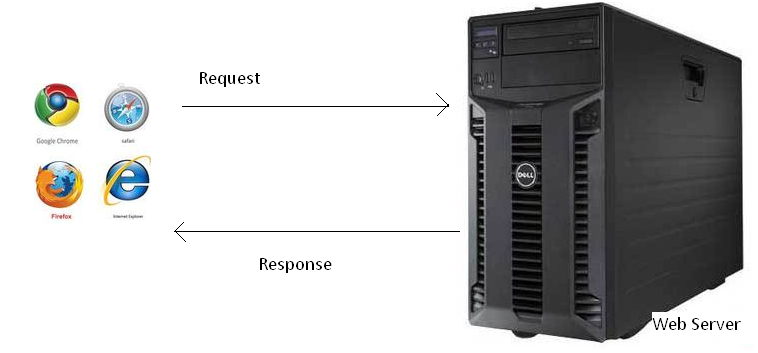
**主机**  ——  作用是运行程序的计算机。

**路由器**——  作用是将信息从一个通信信道传递或转发到另一个通信信道。



## 1， 什么是HTTP协议：

超文本传输协议 (HTTP-Hypertext transfer protocol) 是一种详细规定了浏览器和万维网服务器之间互相通信的规则，通过因特网传送万维网文档的数据传送协议。HTTP是一个应用层协议，由请求和响应构成，是一个标准的客户端服务器模型。HTTP是一个无状态的协议，无状态是指协议对于事务处理没有记忆能力。缺少状态意味着如果后续处理需要前面的信息，则它必须重传，这样可能导致每次连接传送的数据量增大。



## 2， URL

统一[资源](http://baike.baidu.com/view/8439.htm" \t "_blank)定位符（Uniform Resource Locator，缩写为URL）是对可以从互联网上得到的资源的位置和访问方法的一种简洁的表示，是互联网上标准资源的地址。互联网上的每个文件都有一个唯一的URL，它包含的信息指出文件的位置以及浏览器应该怎么处理它

基本格式如下：

schema://host[:port#]/path/.../[;url-params][?query-string][#anchor]

schema 指定所使用的协议（http，https, ftp）

host HTTP服务器的IP地址或者域名

port# HTTP服务器的默认端口是80，这种情况下端口号可以省略

path 访问的资源的路径

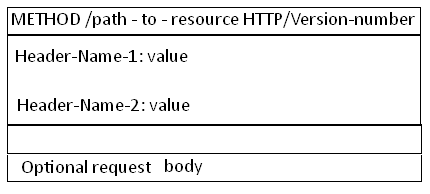
query-string       发送给http服务器的数据

anchor-             锚

## 3，request

是指从客户端到服务器端的请求消息

Request 消息分为3部分，第一部分叫请求行， 第二部分叫http header, 第三部分是body. header和body之间有个空行，结构如下图



Method表示请求方法，比如"POST"，"GET"，

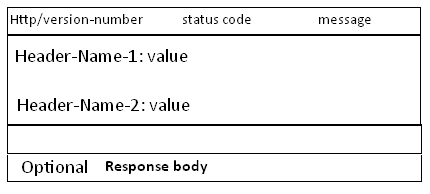
Path-to-resoure表示请求的资源，

Http/version-number 表示HTTP协议的版本号，当使用的是"GET" 方法的时候， body是为空的,当使用"POST"，body不为空，但是没有换行，readline不能读

## 4，response

是指服务器端到客户端的响应信息

和Request消息的结构基本一样。 同样也分为三部分，第一部分叫request line, 第二部分叫request header，第三部分是body. header和body之间也有个空行， 结构如下图



status code状态码:

用来告诉HTTP客户端，HTTP服务器是否产生了预期的Response.

HTTP/1.1中定义了5类状态码:

1XX  提示信息 - 表示请求已被成功接收，继续处理；

2XX  成功 - 表示请求已被成功接收，理解，接受；

3XX  重定向 - 要完成请求必须进行更进一步的处理；

4XX  客户端错误 -  请求有语法错误或请求无法实现；

5XX  服务器端错误 -   服务器未能实现合法的请求

常用的有连接成功200 OK; 找不到对应的资源文件 404 Not Found

## ****5，HTTP请求头和响应头的含义****

**请求头**：

Accept: text/html,image/\*(浏览器可以接收的类型)

Accept-Charset: ISO-8859-1(浏览器可以接收的编码类型)

Accept-Encoding: gzip,compress(浏览器可以接收压缩编码类型)

Accept-Language: en-us,zh-cn(浏览器可以接收的语言和国家类型)

Host: www.it315.org:80(浏览器请求的主机和端口)

If-Modified-Since: Tue, 11 Jul 2000 18:23:51 GMT(某个页面缓存时间)

Referer: http://www.it315.org/index.jsp(请求来自于哪个页面)

User-Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows NT 5.0)(浏览器相关信息)

Cookie：(浏览器暂存服务器发送的信息)

Connection: close(1.0)/Keep-Alive(1.1)(HTTP请求的版本的特点)

Date: Tue, 11 Jul 2000 18:23:51 GMT(请求网站的时间)

**响应头：**

Location: http://www.it315.org/index.jsp(控制浏览器显示哪个页面)

Server:apache tomcat(服务器的类型)

Content-Encoding: gzip(服务器发送的压缩编码方式)

Content-Length: 80(服务器发送显示的字节码长度)

Content-Language: zh-cn(服务器发送内容的语言和国家名)

Content-Type: image/jpeg; charset=UTF-8(服务器发送内容的类型和编码类型)

Last-Modified: Tue, 11 Jul 2000 18:23:51 GMT(服务器最后一次修改的时间)

Refresh: 1;url=http://www.it315.org(控制浏览器1秒钟后转发URL所指向的页面)

Content-Disposition: attachment; filename=aaa.jpg(服务器控制浏览器发下载方式打开文件)

Transfer-Encoding: chunked(服务器分块传递数据到客户端）

Set-Cookie:SS=Q0=5Lb\_nQ; path=/search(服务器发送Cookie相关的信息)

Expires: -1(服务器控制浏览器不要缓存网页，默认是缓存)

Cache-Control: no-cache(服务器控制浏览器不要缓存网页)

Pragma: no-cache(服务器控制浏览器不要缓存网页)

Connection: close/Keep-Alive(HTTP请求的版本的特点)

Date: Tue, 11 Jul 2000 18:23:51 GMT(响应网站的时间)

## web服务器项目

浏览器发送请求：----->

|

服务器接受请求

|

|

解析请求：获取socket，根据http协议解析socket中的内容

|

|

封装请求对象： request（输入流，url,method）

封装响应对象: response（输出流,out,url,path,welcome\_fi）

|

|

执行响应 ：1>普通请求：直接返回用户请求的资源

执行默认的servlet

2>类请求： 执行类

执行自定义的servler

|

|

浏览器接受请求：<-------

推荐参考文献：

《TCP/IP详解》

## HTTP/TCP/IP

HTTP报文是由一行一行的简单字符串组成的，HTTP报文都是纯文本，不是二进制代码，方便读写。HTTP是应用层协议，HTTP无需操心网络通信的具体细节，它将这些细节交给了通用的可靠的因特网传输协议TCP/IP

TCP提供了：无差错的数据传输，按序传输，未分段的数据流（可以在任意时刻以任意尺寸将数据发送出去）

HTTP协议位于TCP的上层，HTTP使用TCP来传输其报文数据，与之类似，TCP位于IP上层

在HTTP客户端向服务器发送报文之前，需要用网际协议（Internet Protocol ，IP)地址和端口号在客户端和服务器之间建立一条TCP/IP连接

HTTP要传送一条报文时，会以流的形式将报文数据的内容通过一条打开的TCP连接按序传输，TCP收到收据流之后，会将数据流砍成被称作为段的小数据块，并将段封装在IP分组中，通过因特网传输。这些工作由TCP/IP软件来处理。

每个TCP段都是由IP分组承载，从一个IP地址发送到另外一个IP地址，每个IP分组中包括：

一个IP分组首部： 源和目的IP地址，长度，其他

一个TCP段首部 ： TCP端口号，TCP控制标识，用于数据排序完整性检查的值

一个TCP数据块 ：

TCP连接通过4个值来识别的：源IP地址，源端口号，目的IP，目的端口号