

HTTP 网络

七层协议、四层协议、TCP、HTTP、SOCKET、长短连接

七层协议

物理层

为数据链路层提供物理连接，在其上串行传送比特流，即所传送数据的单位是比特。此外，该层中还具有确定连接设备的电气特性和物理特性等功能。

数据链路层

负责在网络节点间的线路上通过检测、流量控制和重发等手段，无差错地传送以帧为单位的数据。为做到这一点，在每一帧中必须同时带有同步、地址、差错控制及流量控制等控制信息。

网络层

为了将数据分组从源（源端系统）送到目的地（目标端系统），网络层的任务就是选择合适的路由和交换节点，使源的传输层传下来的分组信息能够正确无误地按照地址找到目的地，并交付给相应的传输层，即完成网络的寻址功能。

传输层

传输层是高低层之间衔接的接口层。数据传输的单位是报文，当报文较长时将它分割成若干分组，然后交给网络层进行传输。传输层是计算机网络协议分层中的最关键一层，该层以上各层将不再管理信息传输问题。

会话层

该层对传输的报文提供同步管理服务。在两个不同系统的互相通信的应用进程之间建立、组织和协调交互。例如，确定是双工还是半双工工作。

表示层

该层的主要任务是把所传送的数据的抽象语法变换为传送语法，即把不同计算机内部的不同表示形式转换成网络通信中的标准表示形式。此外，对传送的数据加密（或解密）、正文压缩（或还原）也是表示层的任务。

应用层

该层直接面向用户，是OSI中的最高层。它的主要任务是为用户提供应用的接口，即提供不同计算机间的文件传送、访问与管理，电子邮件的内容处理，不同计算机通过网络交互访问的虚拟终端功能等。

OSI 中的层	功能	TCP/IP 协议族
应用层	文件传输, 电子邮件, 文件服务, 虚拟终端	TFTP, HTTP, SNMP, FTP, SMTP, DNS, Telnet
表示层	数据格式化, 代码转换, 数据加密	
会话层	解除或建立与别的接点的联系	
传输层	提供端对端的接口	TCP, UDP
网络层	为数据包选择路由	IP, ICMP, RIP, OSPF, BGP, IGMP
数据链路层	传输有地址的帧以及错误检测功能	SLIP, CSLIP, PPP, ARP, RARP, MTU
物理层	以二进制数据形式在物理媒体上上传输数据	ISO2110, IEEE802。 IEEE802.2

Telnet:

提供远程登录（终端仿真）服务，好象比较古老的BBS就是用的这个登陆。

FTP

提供应用级的文件传输服务，说的简单明了点就是远程文件访问等等服务；

SMTP

不用说拉，天天用到的电子邮件协议。

TFTP

提供小而简单的文件传输服务，实际上从某个角度上来说是对FTP的一种替换（在文件特别小并且仅有传输需求的时候）。

SNTP

简单网络管理协议。看名字就不用说什么含义了吧。

DNS

域名解析服务，也就是如何将域名映射成IP地址的协议。

HTTP

不知道各位对这个协议熟不熟悉啊？这是超文本传输协议，你之所以现在能看到网上的图片，动画，音频，等等，都是仰仗这个协议在起作用啊

Socket

Socket是一组编程接口（API）。介于传输层和应用层，向应用层提供

统一的编程接口。应用层不必了解 TCP/IP 协议细节。直接通过对 Socket 接口函数的调用完成数据在 IP 网络的传输。

基于传输层差异，4 种类型的 Socket：

1. 基于 TCP 的 Socket：提供给应用层可靠的流式数据服务，使用 TCP 的 Socket 应用程序协议：BGP, HTTP, FTP, TELNET 等。优点：基于数据传输的可靠性。
2. 基于 UDP 的 Socket：适用于数据传输可靠性要求不高的场合。基于 UDP 的 Socket 应用程序或协议有：RIP, SNMP, L2TP 等。
3. 基于 RawIp 的 Socket：非连接，不可靠的数据传输。特点：能使应用程序直接访问网络层。基于 RawIp 的 Socket 有 ping, tracert, ospf 等。
4. 基于链路层的 Socket。为 IS-IS 协议提供的 Socket 接口。使 IS-IS 协议可通过 Socket 直接访问链路层。非连接，不可靠通信服务。

TCP 连接

手机能够使用联网功能是因为手机底层实现了 TCP/IP 协议，可以使手机终端通过无线网络建立 TCP 连接。TCP 协议可以对上层网络提供接口，使上层网络数据的传输建立在“无差别”的网络之上。

建立起一个 TCP 连接需要经过“三次握手”：

第一次握手：客户端发送 syn 包 ($syn=j$) 到服务器，并进入 SYN_SEND 状态，等待服务器确认；

第二次握手：服务器收到 syn 包，必须确认客户的 SYN ($ack=j+1$)，同时自己也发送一个 SYN 包 ($syn=k$)，即 SYN+ACK 包，此时服务器进入 SYN_RECV 状态；

第三次握手：客户端收到服务器的 SYN + ACK 包，向服务器发送确认包 ACK($ack=k+1$)，此包发送完毕，客户端和服务器进入 ESTABLISHED 状态，完成三次握手。

握手过程中传送的包里不包含数据，三次握手完毕后，客户端与服务器才正式开始传送数据。理想状态下，TCP 连接一旦建立，在通信双方中的任何一方主动关闭连接之前，TCP 连接都将被一直保持下去。断开连接时服务器和客户端均可以主动发起断开 TCP 连接的请求，断开过程需要经过“四次握手”（过程就不细写了，就是服务器和客户端交互，最终确定断开）

HTTP 连接

HTTP 协议即超文本传送协议 (Hypertext Transfer Protocol)，是 Web 联网的基础，也是手机联网常用的协议之一，HTTP 协议是建立在 TCP

协议之上的一种应用。

HTTP 连接最显著的特点是客户端发送的每次请求都需要服务器回送响应，在请求结束后，会主动释放连接。从建立连接到关闭连接的过程称为“一次连接”。

在 HTTP 1.0 中，客户端的每次请求都要求建立一次单独的连接，在处理完本次请求后，就自动释放连接。

在 HTTP 1.1 中则可以在一次连接中处理多个请求，并且多个请求可以重叠进行，不需要等待一个请求结束后再发送下一个请求。

由于 HTTP 在每次请求结束后都会主动释放连接，因此 HTTP 连接是一种“短连接”，要保持客户端程序的在线状态，需要不断地向服务器发起连接请求。通常的做法是即时不需要获得任何数据，客户端也保持每隔一段固定的时间向服务器发送一次“保持连接”的请求，服务器在收到该请求后对客户端进行回复，表明知道客户端“在线”。若服务器长时间无法收到客户端的请求，则认为客户端“下线”，若客户端长时间无法收到服务器的回复，则认为网络已经断开。

长连接与短连接

所谓长连接，指在一个 TCP 连接上可以连续发送多个数据包，在 TCP 连接保持期间，如果没有数据包发送，需要双方发检测包以维持此连接，一般需要自己做在线维持。

短连接是指通信双方有数据交互时，就建立一个 TCP 连接，数据发送完成后，则断开此 TCP 连接，一般银行都使用短连接。

比如 http 的，只是连接、请求、关闭，过程时间较短，服务器若是一段时间内没有收到请求即可关闭连接。

其实长连接是相对于通常的短连接而说的，也就是长时间保持客户端与服务端的连接状态。

长连接与短连接的操作过程

通常的短连接操作步骤是：
连接 → 数据传输 → 关闭连接；

而长连接通常就是：
连接 → 数据传输 → 保持连接 (心跳) → 数据传输 → 保持连接 (心跳) → →
关闭连接；
这就要求长连接在没有数据通信时，定时发送数据包 (心跳)，以维持连接状态，短连接在没有数据传输时直接关闭就行了

HTTP请求过程

域名解析 → 与服务器建立连接 → 发起 HTTP 请求 → 服务器响应 HTTP 请求，浏览器得到 html 代码 → 浏览器解析 html 代码，并请求 html 代码中的资源（如 js、css、图片） → 浏览器对页面进行渲染呈现给用户

状态码

状态码	说明
200	响应成功
302	跳转，跳转地址通过响应头中的Location属性指定（JSP中Forward和Redirect之间的区别）
400	客户端请求有语法错误，不能被服务器识别
403	服务器接收到请求，但是拒绝提供服务（认证失败）
404	请求资源不存在
500	服务器内部错误