TCP/IP（Transmission Control Protocol/Internet Protocol，传输控制协议/互联协议）是**指能够在多个不同网络间实现信息传输的协议簇**。

TCP/IP协议不仅仅指的是[TCP](https://baike.baidu.com/item/TCP/33012) 和[IP](https://baike.baidu.com/item/IP/224599)两个协议，而是指一个由[FTP](https://baike.baidu.com/item/FTP/13839)、[SMTP](https://baike.baidu.com/item/SMTP/175887)、TCP、[UDP](https://baike.baidu.com/item/UDP/571511)、IP等协议构成的**协议簇**， 只是因为在TCP/IP协议中TCP协议和IP协议最具代表性，所以被称为TCP/IP协议。

TCP/IP协议在一定程度上参考了[OSI](https://baike.baidu.com/item/OSI/5520)的体系结构。OSI模型共有**七层**，从下到上分别是物理层、数据链路层、网络层、运输层、会话层、表示层和应用层。但是这显然是有些复杂的，所以在TCP/IP协议中，它们被**简化为了四个层**次。[1]

**TCP/IP是互联网相关的各类协议族的总称**

**TCP/IP分为4层：应用层、传输层、网络层、链路层（数据链路层/**网络接口层**）。**

DNS（Domain Name System）位于应用层，提供域名和IP地址之间的解析服务。

基于这个四个分层，则形成了从**客户端和服务端**的一次完整的信息传输过程。

（1）应用层、表示层、会话层三个层次提供的服务相差不是很大，所以在TCP/IP协议中，它们被**合并为应用层**一个层次。[1]

（2）由于**运输层**和**网络层**在网络协议中的地位十分重要，所以在TCP/IP协议中它们被作为独立的两个层次。[1]

（3）因为**数据链路层**和**物理层**的内容相差不多，所以在TCP/IP协议中它们被归并在**网络接口层**一个层次里。只有四层体系结构的TCP/IP协议，与有七层体系结构的OSI相比要简单了不少，也正是这样，TCP/IP协议在实际的应用中**效率更高，成本更低**。

**应用层**：应用层是TCP/IP协议的第一层，是**直接为应用进程提供服务**的。

（1）对不同种类的应用程序它们会根据自己的需要来使用应用层的不同协议，邮件传输应用使用了[SMTP](https://baike.baidu.com/item/SMTP/175887)协议、万维网应用使用了[HTTP](https://baike.baidu.com/item/HTTP/243074)协议、[文件传输](https://baike.baidu.com/item/%E6%96%87%E4%BB%B6%E4%BC%A0%E8%BE%93%E5%8D%8F%E8%AE%AE)的FTP协议、远程登录服务应用使用了有[TELNET](https://baike.baidu.com/item/TELNET/810597)协议。[1]

（2）应用层还能加密、解密、格式化数据。

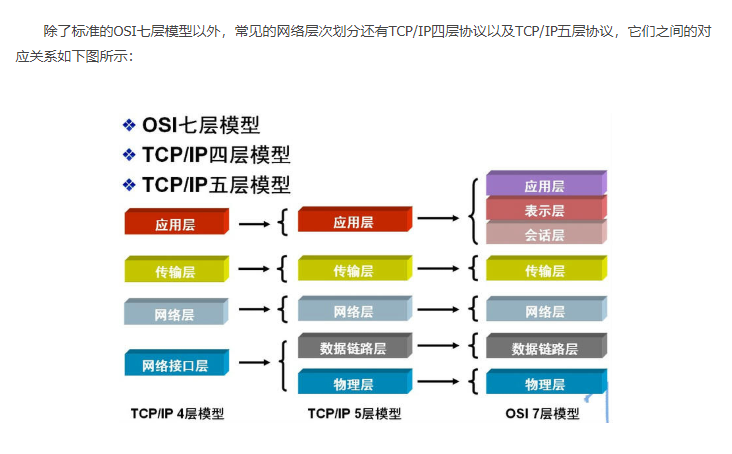
（3）应用层可以建立或解除与其他节点的联系，这样可以充分节省网络资源。

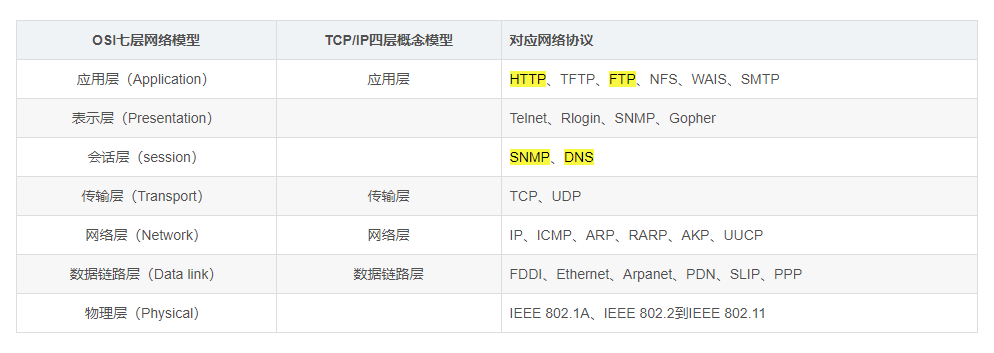
**运输层**：作为TCP/IP协议的第二层，运输层在整个TCP/IP协议中起到了中流砥柱的作用。且在运输层中，TCP和UDP也同样起到了中流砥柱的作用。

**网络层**：网络层在TCP/IP协议中的位于第三层。在TCP/IP协议中网络层可以进行网络连接的建立和终止以及IP地址的寻找等功能。[1]

**网络接口层**：在TCP/IP协议中，网络接口层位于第四层。由于网络接口层兼并了[物理层](https://baike.baidu.com/item/%E7%89%A9%E7%90%86%E5%B1%82/4329158)和[数据链路层](https://baike.baidu.com/item/%E6%95%B0%E6%8D%AE%E9%93%BE%E8%B7%AF%E5%B1%82/4329290)所以，网络接口层既是传输数据的物理媒介，也可以为网络层提供一条准确无误的线路。[1]

HTTP是应用层协议，主要解决如何**包装数据**。  
Socket是对**TCP/IP协议的封装**，Socket本身并不是协议，而是一个**调用接口**（API），通过Socket，我们才能使用TCP/IP协议。  
**可以把ip层想象成公路，tcp、udp是火车，而http，https等是货物**





**一次完整的HTTP请求过程**

​ **DNS域名解析**

​ **建立tcp/ip连接：客户端与服务器通过socket三次握手进行连接建立tcp/ip连接**

​ **建立tcp/ip连接后发起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、图片等）**

​ **浏览器对页面进行渲染呈现给用户**

**HTTP/2**

HTTP/2是HTTP协议自1999年HTTP1.1发布后的首个更新，主要基于**SPDY协议**。  
HTTP2.0的特点是：**在不改动HTTP语义、方法、状态码、URI及首部字段的情况下，大幅度提高了web性能。**

SPDY是Speedy的昵音，意为“更快”。它是Google开发的**基于TCP协议**的应用层协议。目标是优化HTTP协议的性能，通过压缩、多路复用和优先级等技术，缩短网页的加载时间并提高安全性。

**SPDY协议**的核心思想是**尽量减少TCP连接数**。SPDY并不是一种用于替代HTTP的协议，而是对HTTP协议的增强。

HTTP2.0中所有加强性能的核心是**二进制传输**，在HTTP1.x中，我们是通过**文本的方式**传输数据。基于文本的方式传输数据存在很多缺陷，文本的表现形式有多样性，因此要做到健壮性考虑的场景必然有很多，但是二进制则不同，只有0和1的组合，因此选择了二进制传输，实现方便且健壮。  
在HTTP2.0中引入了新的编码机制，所有传输的数据都会被分割，并采用二进制格式编码。

为了保证HTTP不受影响，那就需要在**应用层（HTTP2.0）**和**传输层（TCP or UDP）**之间增加一个

在二进制分帧层上，HTTP2.0会将所有传输的信息分为更小的消息和帧，并采用二进制格式编码，其中HTTP1.x的首部信息会被封装到Headers帧，而Request Body则封装到Data帧。

HTTP/2：

致力于突破http1.1的性能限制，改进传输性能，实现低延迟和高吞吐量。但核心概念上保持不变。

升级 HTTP/2 的前提条件是什么？

答：由于现在所有支持 HTTP/2 的浏览器都强制只使用 TLS(https) 连接，所以：获取证书，并且让服务器支持 https 是必须的先决条件。

对HTTP1.1有4个改进：

二进制分帧、多路复用、头信息压缩、服务器推送。

二进制分帧：

HTTP/1.1 版的头信息必须是文本（ASCII编码），数据体可以是文本或是二进制。HTTP/2 则是一个彻底的二进制协议，头信息和数据体都是二进制，并且统称为"帧"（frame）：头信息帧和数据帧。文本的表现形式有多样性，二进制则不同，只认0和1的组合，所以HTTP2的协议解析决定采用二进制格式，实现方便且健壮。

既然又要保证HTTP的各种动词，方法，首部都不受影响，那就需要在应用层(HTTP2.0)和传输层(TCP or UDP)之间增加一个二进制分帧层。

多路复用（连接共享）：

HTTP/2引入连接共享机制，允许一个连接里，同时发送多个请求和接受多个回应，因此 HTTP/2 可以很容易的去实现多流并行而不用依赖建立多个 TCP 连接。从而提高了性能。

头信息压缩：

HTTP是无状态协议，每次请求都必须附上所有信息。请求的很多字段都是重复的，只需发送一次即可，但每次请求都必须附带，这会浪费性能，影响速度。HTTP/2对这一点做了优化，引入了头信息压缩机制（header compression）。一方面，对头信息压缩后再发送；另一方面，客户端和服务端同时维护一张头信息表，所有字段都会存入这个表，这样后面每次传输只需要传输表里面的索引Id就行，通过索引ID就可以知道表头的值。

服务器推送（缓存推送）：

HTTP/2允许服务器未经请求，主动向客户端发送资源，这叫做服务器推送。

服务器可以对一个客户端请求发送多个响应，服务器可以预期到客户端请求网页后，很可能会再请求静态资源，所以就主动把这些静态资源随着网页一起发给客户端了