专业技能

熟悉TCP/IP网络模型以及TCP、UDP等协议，对HTTP协议的版本变化有所研究

计算机网络

线程和协程的区别与联系

**区别：**

执行单元：

线程： 是操作系统调度的最小执行单元，由操作系统进行管理和调度。

协程： 是在用户空间进行调度的执行单元，由程序员显式控制。

调度方式：

线程： 由操作系统进行抢占式调度，可以被中断和切换。

协程： 由程序员控制，在需要时显式地挂起和恢复。

并发性：

线程： 线程可以并行执行，由操作系统管理多个线程的调度。

协程： 协程一般在单个线程内执行，通过非抢占式调度实现协作式并发。

状态保存：

线程： 线程的状态由操作系统负责保存和恢复。

协程： 协程的状态由程序员显式保存和恢复，通常保存在栈上。

**联系：**

并发编程：

无论是线程还是协程，都是为了实现并发编程，让程序能够有效地处理多个任务。

异步编程：

协程常用于异步编程，通过非阻塞的方式处理I/O等操作，提高程序的性能。

资源消耗：

协程通常消耗的资源较少，因为它在用户空间进行调度，减少了操作系统线程切换的开销。

可读性和控制：

协程通常具有更好的可读性，因为它在编写代码时更接近自然的同步写法，而不需要考虑锁和线程间通信的复杂性。

程序员对协程的控制更细粒度，可以手动控制协程的执行顺序。

**总体而言**

线程更适用于 CPU 密集型任务

协程更适用于 I/O 密集型任务，以及需要更好的可读性和控制的情况。

浏览器键入baidu.com所有用到的协议

当你在浏览器中键入baidu.com并按下回车时，浏览器会根据输入的网址进行一系列操作，涉及到多个协议。以下是在浏览器键入baidu.com时可能用到的主要协议：

DNS 协议（Domain Name System）：

浏览器首先会将域名 baidu.com 转换成对应的 IP 地址。这个过程使用 DNS 协议进行域名解析。

HTTP 协议（Hypertext Transfer Protocol）：

一般情况下，浏览器会默认使用 HTTP 协议进行网页的请求和响应。浏览器向服务器发送 HTTP 请求，请求获取 baidu.com 的网页内容。

HTTPS 协议（Hypertext Transfer Protocol Secure）：

如果网站启用了 HTTPS，浏览器会使用 HTTPS 协议进行加密的通信。这涉及到 SSL/TLS 协议用于加密数据传输，确保数据的安全性。

TCP 协议（Transmission Control Protocol）：

HTTP 和 HTTPS 协议都是基于 TCP 协议的，它负责在网络上建立可靠的连接，确保数据的可靠传输。

IP 协议（Internet Protocol）：

IP 协议用于在网络上进行数据包的寻址和路由。它是 TCP 和 UDP 的底层协议，负责在网络上进行数据包的传递。

ICMP 协议（Internet Control Message Protocol）：

在进行网络通信时，可能涉及 ICMP 协议，用于处理网络错误、诊断问题等。

以上是在浏览器键入 baidu.com 时可能用到的主要协议。实际情况中，根据网站的具体配置和浏览器的支持情况，可能还涉及到其他协议和技术，如 QUIC 协议、WebSocket 等。

TCP和UDP区别

TCP（Transmission Control Protocol）和UDP（User Datagram Protocol）是两种常用的传输层协议，它们在网络通信中有一些重要的区别：

1. 连接性：

TCP： 提供面向连接的通信。在数据传输之前，必须先建立连接，然后进行可靠的数据传输，最后再释放连接。

UDP： 是面向无连接的通信。每个数据包都是独立的，不需要先建立连接，直接发送数据。因此，UDP通信更为轻量，但不保证可靠性。

2. 可靠性：

TCP： 提供可靠的数据传输，通过序列号、确认和重传机制确保数据的完整性和顺序性。如果数据包丢失，TCP会进行重传。

UDP： 不提供可靠性保证，数据包可能会丢失，也不保证数据包的顺序性。对于某些实时性要求高的应用，可以容忍少量的丢包。

3. 数据传输方式：

TCP： 数据以流的方式进行传输，被划分为小的数据段，然后通过序列号进行重新组装。数据传输是可靠而有序的。

UDP： 数据以数据包的形式传输，每个数据包都是独立的。UDP更适用于需要快速传输的场景，如实时音视频。

4. 头部开销：

TCP： TCP头部相对较大，包含序列号、确认号、窗口大小等字段，用于保证可靠性和流控制。

UDP： UDP头部较小，只包含源端口、目标端口、长度和校验和等基本信息，开销较小。

5. 适用场景：

TCP： 适用于对数据完整性和顺序性有要求的应用，如文件传输、网页访问等。

UDP： 适用于实时性要求高、可以容忍少量数据丢失的应用，如语音通话、视频直播等。

6. 连接和断开：

TCP： 建立连接时有三次握手的过程，断开连接有四次挥手的过程。

UDP： 无连接的特性，不需要建立和断开连接的过程。

总体来说，选择TCP还是UDP取决于具体的应用需求。TCP适用于要求可靠性和有序性的场景，而UDP适用于实时性要求高、可以容忍少量数据丢失的场景。在某些应用中，也可以结合使用TCP和UDP，根据不同的数据传输需求选择合适的协议。