**TCP和UDP的区别和优缺点**

2017年08月06日 20:32:16 阅读数：70652 标签： [TCP](http://so.csdn.net/so/search/s.do?q=TCP&t=blog)[UDP](http://so.csdn.net/so/search/s.do?q=UDP&t=blog) 更多

个人分类： [c](https://blog.csdn.net/xiaobangkuaipao/article/category/6803622)

版权声明：本文为博主原创文章，未经博主允许不得转载。 https://blog.csdn.net/xiaobangkuaipao/article/details/76793702

1、TCP与UDP区别总结：  
1、TCP面向连接（如打电话要先拨号建立连接）;UDP是无连接的，即发送数据之前不需要建立连接

2、TCP提供可靠的服务。也就是说，通过TCP连接传送的数据，无差错，不丢失，不重复，且按序到达;UDP尽最大努力交付，即不保证可靠交付

Tcp通过校验和，重传控制，序号标识，滑动窗口、确认应答实现可靠传输。如丢包时的重发控制，还可以对次序乱掉的分包进行顺序控制。

3、UDP具有较好的实时性，工作效率比TCP高，适用于对高速传输和实时性有较高的通信或广播通信。

4.每一条TCP连接只能是点到点的;UDP支持一对一，一对多，多对一和多对多的交互通信

5、TCP对系统资源要求较多，UDP对系统资源要求较少。

2、为什么UDP有时比TCP更有优势?

UDP以其简单、传输快的优势，在越来越多场景下取代了TCP,如实时游戏。

（1）网速的提升给UDP的稳定性提供可靠网络保障，丢包率很低，如果使用应用层重传，能够确保传输的可靠性。

（2）TCP为了实现网络通信的可靠性，使用了复杂的拥塞控制算法，建立了繁琐的握手过程，由于TCP内置的系统协议栈中，极难对其进行改进。

采用TCP，一旦发生丢包，TCP会将后续的包缓存起来，等前面的包重传并接收到后再继续发送，延时会越来越大，基于UDP对实时性要求较为严格的情况下，采用自定义重传机制，能够把丢包产生的延迟降到最低，尽量减少网络问题对游戏性造成影响。

3、UDP和TCP编程步骤也有些不同，如下：

**TCP:**  
TCP编程的服务器端一般步骤是：   
　　1、创建一个socket，用函数socket()；     SOCKET SocketListen =socket(AF\_INET,SOCK\_STREAM, IPPROTO\_TCP);  
　　2、设置socket属性，用函数setsockopt(); \* 可选   
　　3、绑定IP地址、端口等信息到socket上，用函数bind(); SOCKET\_ERROR = bind(SocketListen,(const sockaddr\*)&addr,sizeof(addr))  
　　4、开启监听，用函数listen()；                 SOCKET\_ERROR == listen(SocketListen,2)  
　　5、接收客户端上来的连接，用函数accept()；    SOCKET SocketWaiter = accept(SocketListen,

                                                  \_Out\_    struct sockaddr \*addr

 \_Inout\_  int \*addrlen);  
　　6、收发数据，用函数send()和recv()，或者read()和write();   
　　7、关闭网络连接； closesocket(SocketListen);closesocket(SocketWaiter);  
　　8、关闭监听；   
SOCK\_STREAM这种的特点是面向连接的，即每次收发数据之前必须通过connect建立连接，而SOCK\_DGRAM这种是User Datagram Protocol协议的网络通讯，它是无连接的，不可靠的。  
TCP编程的客户端一般步骤是：   
　　1、创建一个socket，用函数socket()；   
　　2、设置socket属性，用函数setsockopt();\* 可选   
　　3、绑定IP地址、端口等信息到socket上，用函数bind();\* 可选   
　　4、设置要连接的对方的IP地址和端口等属性；   
　　5、连接服务器，用函数connect()；   
　　6、收发数据，用函数send()和recv()，或者read()和write();   
　　7、关闭网络连接；

int send(

\_In\_  SOCKET s, //向哪个socket发送，accept返回的socket。

\_In\_  const char \*buf,

\_In\_  int len,

\_In\_  int flags

); 由于

send(SocketClient,(const char \*)&fh,sizeof(fh),0);

recv(SocketClient,szbuf,sizeof(szbuf),0);  
**UDP:**  
与之对应的UDP编程步骤要简单许多，分别如下：   
　　UDP编程的服务器端一般步骤是：   
　　1、创建一个socket，用函数socket()；   
　　2、设置socket属性，用函数setsockopt();\* 可选   
　　3、绑定IP地址、端口等信息到socket上，用函数bind();   
　　4、循环接收数据，用函数recvfrom();   
　　5、关闭网络连接；   
  
UDP编程的客户端一般步骤是：   
　　1、创建一个socket，用函数socket()；   
　　2、设置socket属性，用函数setsockopt();\* 可选   
　　3、绑定IP地址、端口等信息到socket上，用函数bind();\* 可选   
　　4、设置对方的IP地址和端口等属性;   
　　5、发送数据，用函数sendto();   
　　6、关闭网络连接；

int recvfrom(

\_In\_         SOCKET s, //绑定的socket

\_Out\_        char \*buf,

\_In\_         int len,

\_In\_         int flags,

\_Out\_        struct sockaddr \*from, //用来接收对方的

\_Inout\_opt\_  int \*fromlen

);

int nres=recvfrom(pThis->m\_socketListen,szBuf,sizeof(szBuf),0,(sockaddr\*)&addrClient,&nSize);//0处标志位  
sendto(m\_socketListen,szBuffer,nSize,0,(const sockaddr\*)&addr,sizeof(sockaddr\_in))  
TCP和UDP是OSI模型中的运输层中的协议。TCP提供可靠的通信传输，而UDP则常被用于让广播和细节控制交给应用的通信传输。

4、将socket设置为广播属性  
bool optval=true;  
setsockopt(m\_socketListen,SOL\_SOCKET,SO\_BROADCAST,(const char \*)&optval,sizeof(bool));

5、将Socket设置为非阻塞。  
//bool benable=true;  
//ioctlsocket(m\_socketListen,FIONBIO,(u\_long\*)&benable);

6、Tcp头，20字节

7、UDP首部,8个字节



**TCP和UDP的9个区别是什么**

2016年03月21日 11:56:51 阅读数：5479 标签： [网络开发](http://so.csdn.net/so/search/s.do?q=%E7%BD%91%E7%BB%9C%E5%BC%80%E5%8F%91&t=blog) 更多

个人分类： [PHP学习与实践](https://blog.csdn.net/yanglr2010/article/category/1435859)[Java](https://blog.csdn.net/yanglr2010/article/category/1435844)[网络与安全](https://blog.csdn.net/yanglr2010/article/category/1435845)

       TCP和UDP是两个传输层协议,广泛应用于网络中不同主机之间传输数据。对任何程序员来说，熟悉TCP和UDP的工作方式都是至关重要的。这就是为什么TCP和UDP是一个流行的[Java编程面试问题](https://blog.csdn.net/lzuacm/article/details/Java)。我曾经在各种不同的Java面试中见过这个问题,尤其是对服务器端Java开发人员的面试。由于FIX(金融信息交换)协议也是一个基于TCP协议,一些投资银行,对冲基金,和交换解决方案提供商也在寻找有良好的TCP和UDP的知识的Java开发人员。为高速电子交易平台编写FIX引擎和服务器端组件需要能够对数据结构、算法以及网络等基本知识都理解透彻的的开发人员的

       顺便说一下,使用TCP和UDP并不局限于一个领域,它处于互联网的中心位置。HTTP是基于TCP，而HTTP是整个互联网的核心部分。另一个Java开发人员应该理解这两个协议的理由是, java广泛用于编写多线程、并行以及可伸缩的服务器程序。Java还提供了丰富的基于TCP和UDP套接字编程API。在这篇文章中,我们将学习TCP和UDP协议之间的关键差异。首先,TCP代表传输控制协议和UDP代表用户数据报协议,两者都广泛的应用于构建互联网应用。

**TCP与UDP之间的区别**

       我喜欢比较两事物的不同点，这不仅使他们更加容易比较，而且会更容易记住他们之间的差异。当我们比较TCP，UDP，我们需要了解TCP和UDP各自的工作方式，了解哪种协议提供了可靠保证交付和而哪个又没有。哪种协议是快速的，他为什么更加快速，最重要的是，当我们了解了这些，在建设我们自己的分布式应用程序时，就知道该选择TCP还是 UDP。在这篇文章中，我们将看到9点UDP和TCP之间的差异，例如，连接步骤，排序，速度，可靠性，开销，头大小，拥塞控制，应用以及基于TCP和UDP协议不同，他们如何传输数据。

**1. 基于连接vs无连接**

他们之间的第一点并且最重要的区别是：TCP是面向连接的协议，而UDP是无连接的协议。这意味着当一个客户端和一个服务器通过TCP发送数据之前，必须先建立连接，他们可以通过TCP发送数据。建立连接的过程也被称为TCP握手，他通过控制消息在客户端和服务器之间互换来实现。下面的图形象描述了TCP握手过程。客户端，它也是TCP连接的发起者，发送一个SYN消息给服务器，该服务器端正在监听某个TCP端口。服务器接收该消息并发送一个SYN-ACK消息，客户端接受到该消息之后会再回一个ACK消息。一旦服务器收到ACK消息，TCP连接就建立成功，准备数据传输了。另一方面，UDP是无连接的协议，和点对点连接之前不需要发送消息。这就是为什么，UDP更加适合消息的[多播发布](http://javarevisited.blogspot.sg/2010/10/common-terms-used-in-tibco-rv.html)，从单个点向多个点传输消息。

图1：TCP消息握手(TCP - Handshake Message)

**2. 可靠性 不同**

TCP提供交付保证,这意味着一个使用TCP协议发送的消息是保证交付给客户端的。如果消息在传输过程中丢失,那么它将重发,这是由TCP协议本身控制的。另一方面,UDP是不可靠的,它不提供任何交付的保证。一个数据报包在运输途中可能会丢失。这就是为什么UDP是不适合保证交付的项目。

**3.有序性**

除了提供交付保证，为TCP也保证了消息的有序性。该消息将以从服务器端发出的同样的顺序发送到客户端，尽管这些消息到网络的另一端时可能是无序的。TCP协议将会为你排好序。UDP不提供任何有序性或序列性的保证。数据包将以任何可能的顺序到达。这就是为什么TCP是适合需要顺序交付方式的应用，尽管有基于UDP的协议通过使用序列号和重传来提供有序和可靠性的应用，如[TIBCO Rendezvous](http://javarevisited.blogspot.sg/2010/10/tibco-rv-messagging.html)，他实际上就是一个基于UDP的应用。

**4.数据边界**

TCP不保存数据的边界，而UDP保证。在传输控制协议，数据以字节流的形式发送，并没有明显的标志表明传输信号消息（段）的边界。在UDP中，数据包单独发送的，只有当他们到达时，才会再次集成。包有明确的界限来哪些包已经收到，这意味着在消息发送后，在接收器接口将会有一个读操作，来生成一个完整的消息。虽然TCP也将在收集所有字节之后生成一个完整的消息，但是这些信息在传给传输给接受端之前将储存在TCP缓冲区，以确保更好的使用网络带宽

**5.速度**

总而言之，TCP速度比较慢，而UDP速度比较快，因为TCP必须创建连接，以保证消息的可靠交付和有序性，他需要做比UDP多的多的事。这就是为什么UDP更适用于对速度比较敏感的应用，例如：在线视频媒体，电视广播和多人在线游戏。

**6.重量级vs轻量级**

由于上述的开销，TCP被认为是重量级的协议，而与之相比，UDP协议则是一个轻量级的协议。因为UDP传输的信息中不承担任何间接创造连接，保证交货或秩序的的信息。这也反映在用于承载元数据的头的大小。

**7. 头大小**

TCP具有比UDP更大的头。一个TCP数据包报头的大小是20字节，UDP数据报报头是8个字节。TCP报头中包含序列号，ACK号，数据偏移量，保留，控制位，窗口，紧急指针，可选项，填充项，校验位，源端口和目的端口。而UDP报头只包含长度，源端口号，目的端口，和校验和。下图是TCP和UDP头：

图2：TCP包格式(TCP Packet Format)

图3：UDP包格式(UDP Packet format)

**8. 拥塞或流控制**

TCP有流量控制。在任何用户数据可以被发送之前，TCP需要三数据包来设置一个套接字连接。TCP处理的可靠性和拥塞控制。另一方面，UDP不能进行流量控制。

**9. 用法和应用**

在互联网中，TCP和UDP都运行在哪些环境中了？在了解了TCP和UDP之间的关键差异之后，我们可以很容易地得出结论，哪种情况适合他们。由于TCP提供可靠交付和有序性的保证，它是最适合需要高可靠并且对传输时间要求不高的应用。UDP是更适合的应用程序需要快速，高效的传输的应用，如游戏。UDP是无状态的性质，在服务器端需要对大量客户端产生的少量请求进行应答的应用中是非常有用的。在实践中，TCP被用于金融领域，如FIX协议是一种基于TCP的协议，而UDP是大量使用在游戏和娱乐场所。

**基于TCP和UDP的协议**

基于TCP协议的最好例子是HTTP协议和HTTPS协议，他们几乎存在于互联网的任何地方，实际上，绝大多数你所熟悉的通常协议，都是基于TCP的，例如：Telnet，FTP以及SMTP协议。UDP协议没有TCP协议那么受欢迎，但是也被广泛应用，比如DHCP以及DNS协议，其他还有一些基于UDP的协议如SNMP,TFTP,BOOTP以及NFS（早期版本）。

特别需要记住的是，TCP是面向连接的，可靠的，缓慢的，可靠交付以及保证消息顺序的，而UDP是无连接的，不可靠的，没有序列保证，但是一个快速传输的协议。TCP头开销也比UDP高得多，因为它每个数据包中药发送更多的元数据。值得一提的是，TCP头的大小是20个字节，而UDP头大小是8个字节。如果你不想丢失任何消息，使用TCP协议，而UDP能够高速传输数据，并且丢失少量的数据包是可以接受的，如视频流或在线多玩家游戏。对于基于TCP / UDP协议，运行在Linux上的应用，需要牢记的[基本网络命令](http://javarevisited.blogspot.sg/2010/10/basic-networking-commands-in-linuxunix.html)，如Telnet和netstat，他们极大的帮助调试和排除任何连接问题。

https://blog.csdn.net/u011957758/article/details/72353485