

网络编程懒人入门(五)：快速理解为什么说UDP有时比TCP更有优势-网络编程/专项技术区 - 即时通讯开发者社区！



关注我的公众号

即时通讯技术之路，你并不孤单！

IM开发 / 实时通信 / 网络编程

本文观点仅作参考，请根据自己系统的应用场景合理地选择数据传输层协议即可，无需盲目崇拜大牛言论。

1、前言

对于即时通讯开者新手来说，在开始着手编写IM或消息推送系统的代码前，最头疼的问题莫过于到底该选TCP还是UDP作为传输层协议。本文延续《网络编程懒人入门》系列文章的风格，通过快速对比分析 TCP 和 UDP 的区别，来帮助即时通讯初学者快速了解这些基础的知识点，从而在IM、消息推送等网络通信应用场景中能准确地选择合适的传输层协议。

随着网络技术飞速发展，网速已不再是传输的瓶颈，UDP

协议以其简单、传输快的优势，在越来越多场景下取代了TCP，如网页浏览、流媒体、实时游戏、物联网。本文作为《网络编程懒人入门》系列文章的第5篇，将为您快速梳理UDP协议在某些场景下对比TCP协议所具有的优势。

另外，即时通讯网的文章：《[简述传输层协议TCP和UDP的区别](#)》、《[为什么QQ用的是UDP协议而不是TCP协议？](#)》、《[移动端即时通讯协议选择：UDP还是TCP？](#)》，更详细地阐述了类似的内容，可以为您提供更多的参考。

2、系列文章

本文是系列文章中的第4篇，本系列文章的大纲如下：

- 《[网络编程懒人入门\(一\)：快速理解网络通信协议__（上篇）__](#)》
- 《[网络编程懒人入门\(二\)：快速理解网络通信协议__（下篇）__](#)》
- 《[网络编程懒人入门\(三\)：快速理解TCP协议一篇就够](#)》
- 《[网络编程懒人入门\(四\)：快速理解TCP和UDP的差异](#)》
- 《[网络编程懒人入门\(五\)：快速理解为什么说UDP有时比TCP更有优势](#)》（本文）
- 《[网络编程懒人入门\(六\)：史上最通俗的集线器、交换机、路由器功能原理入门](#)》
- 《[网络编程懒人入门\(七\)：深入浅出，全面理解HTTP](#)》

[协议》](#)

- [《网络编程懒人入门\(八\): 手把手教你写基于TCP的Socket长连接》](#)
- [《网络编程懒人入门\(九\): 通俗讲解, 有了IP地址, 为何还要用MAC地址? 》](#)

本站的《脑残式网络编程入门》也适合入门学习, 本系列大纲如下:

- [《脑残式网络编程入门\(一\): 跟着动画来学TCP三次握手和四次挥手》](#)
- [《脑残式网络编程入门\(二\): 我们在读写Socket时, 究竟在读写什么? 》](#)
- [《脑残式网络编程入门\(三\): HTTP协议必知必会的一些知识》](#)
- [《脑残式网络编程入门\(四\): 快速理解HTTP/2的服务器推送\(Server Push\)》](#)

如果您觉得本系列文章过于基础, 可直接阅读《不为人知的网络编程》系列文章, 目录如下:

- [《不为人知的网络编程\(一\): 浅析TCP协议中的疑难杂症\(上篇\)》](#)
- [《不为人知的网络编程\(二\): 浅析TCP协议中的疑难杂症\(下篇\)》](#)
- [《不为人知的网络编程\(三\): 关闭TCP连接时为什么](#)

会TIME_WAIT、CLOSE_WAIT》

- [《不为人知的网络编程\(四\)：深入研究分析TCP的异常关闭》](#)
- [《不为人知的网络编程\(五\)：UDP的连接性和负载均衡》](#)
- [《不为人知的网络编程\(六\)：深入地理解UDP协议并用好它》](#)
- [《不为人知的网络编程\(七\)：如何让不可靠的UDP变的可靠？》](#)
- [《不为人知的网络编程\(八\)：从数据传输层深度解密HTTP》](#)
- [《不为人知的网络编程\(九\)：理论联系实际，全方位深入理解DNS》](#)

关于移动端网络特性及优化手段的总结性文章请见：

- [《现代移动端网络短连接的优化手段总结：请求速度、弱网适应、安全保障》](#)
- [《移动端IM开发者必读\(一\)：通俗易懂，理解移动网络的“弱”和“慢”》](#)
- [《移动端IM开发者必读\(二\)：史上最全移动弱网络优化方法总结》](#)

3、参考资料

[《TCP/IP详解 - 第11章·UDP：用户数据报协议》](#)

[《TCP/IP详解 - 第17章·TCP：传输控制协议》](#)

[《TCP/IP详解 - 第18章·TCP连接的建立与终止》](#)

[《TCP/IP详解 - 第21章·TCP的超时与重传》](#)

[《通俗易懂-深入理解TCP协议（上）：理论基础》](#)

[《通俗易懂-深入理解TCP协议（下）：RTT、滑动窗口、拥塞处理》](#)

[《理论经典：TCP协议的3次握手与4次挥手过程详解》](#)

[《理论联系实际：Wireshark抓包分析TCP 3次握手、4次挥手过程》](#)

[《技术往事：改变世界的TCP/IP协议（珍贵多图、手机慎点）》](#)

[《计算机网络通讯协议关系图（中文珍藏版）》](#)

[《高性能网络编程\(一\)：单台服务器并发TCP连接数到底可以有多少》](#)

[《高性能网络编程\(二\)：上一个10年，著名的C10K并发连接问题》](#)

[《高性能网络编程\(三\)：下一个10年，是时候考虑C10M并发问题了》](#)

[《高性能网络编程\(四\)：从C10K到C10M高性能网络应用的理论探索》](#)

[《简述传输层协议TCP和UDP的区别》](#)

[《UDP中一个包的大小最大能多大？》](#)

[《为什么QQ用的是UDP协议而不是TCP协议？》](#)

[《移动端即时通讯协议选择：UDP还是TCP？》](#)

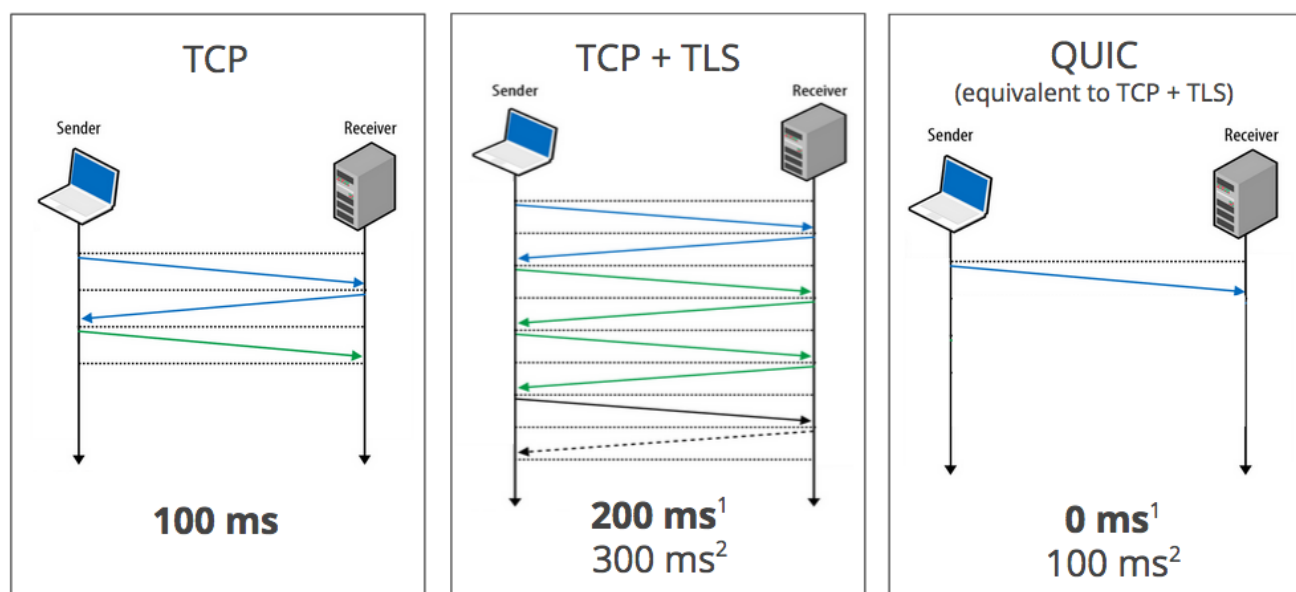
4、网速的提升给UDP稳定性提供可靠网络保障

CDN服务商Akamai报告从2008年到2015年7年时间，各个国家网络平均速率由1.5Mbps提升为5.1Mbps，网速提升近4倍。网络环境变好，网络传输的延迟、稳定性也随之改善，UDP的丢包率低于5%，如果再使用应用层重传，能够完全确保传输的可靠性。

5、对比测试结果UDP性能优于TCP

为了提升浏览速度，Google基于TCP提出了SPDY协议以及HTTP/2。Google在Chrome上实验基于UDP的QUIC协议，传输速率减少到100ms以内。

Zero RTT Connection Establishment



1. Repeat connection
2. Never talked to server before



- Google采用QUIC后连接速率能有效提升75%；
- Google搜索采用QUIC后页面加载性能提升3%；
- YouTube采用QUIC后重新缓冲次数减少了30%。

6、TCP设计过于冗余，速度难以进一步提升

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以及重传策略。由于TCP内置在系统协议栈中，极难对其进行改进。

7、UDP协议以其简单、传输快的优势，在越来越多场景下取代了TCP

7.1网页浏览

使用UDP协议有三个优点：

- 能够对握手过程进行精简，减少网络通信往返次数；
- 能够对TLS加解密过程进行优化；
- 收发快速，无阻塞。

7.2流媒体

采用TCP，一旦发生丢包，TCP会将后续包缓存起来，等

前面的包重传并接收到后再继续发送，延迟会越来越大。基于UDP的协议如实时音视频开源工程WebRTC是极佳的选择。

2010年google 通过收购 Global IP Solutions，获得了WebRTC（网页实时通信Web Real-Time Communication）技术，用于提升网页视频速率。关于WebRTC的介绍，请见：《[访谈WebRTC标准之父：WebRTC的过去、现在和未来](#)》，[更多WebRTC文章点此进入](#)。

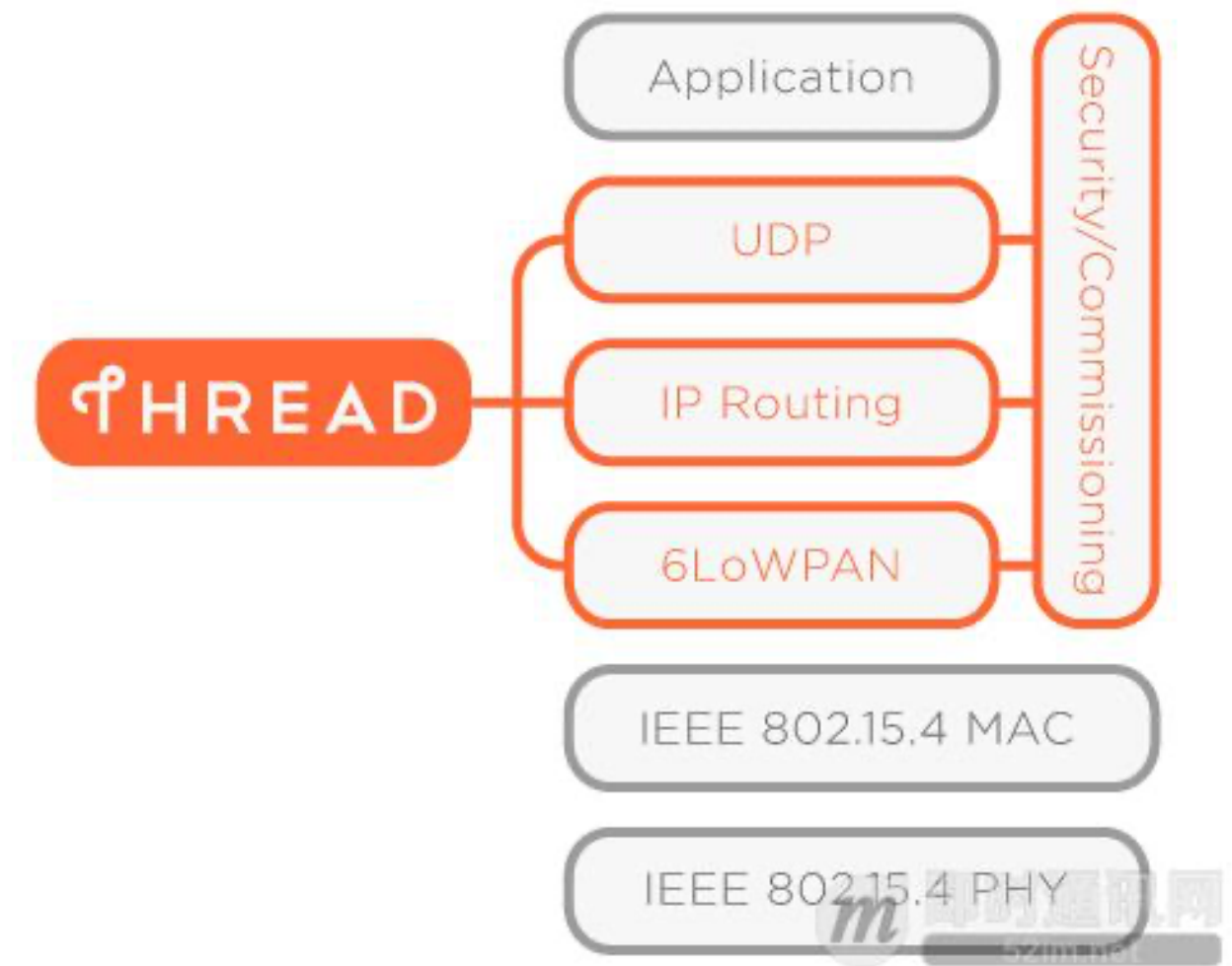
7.3实时游戏

对实时要求较为严格的情况下，采用自定义的可靠UDP协议，比如Enet、RakNet（用户有 sony online game、minecraft）等，自定义重传策略，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的影响。

采用UDP的经典游戏如FPS游戏Quake、CS，著名的游戏引擎Unity3D采用的也是RakNet。

7.4物联网

2014年google旗下的Nest建立Thread Group，推出了物联网通信协议Thread，完善物联网通信。



采用UDP有3个关键点：

- 网络带宽需求较小，而实时性要求高；
- 大部分应用无需维持连接；
- 需要低功耗。

8、本文小结

如今全球将近50%的人都在使用互联网，人们不断的追求更快、更好的服务，一切都在变化，在越来越多的领域，

UDP将会抢占TCP的主导地位。

附录：更多高性能网络编程文章

《[Java新一代网络编程模型AIO原理及Linux系统AIO介绍](#)》

《[有关“为何选择Netty”的11个疑问及解答](#)》

《[开源NIO框架八卦——到底是先有MINA还是先有Netty?](#)》

《[选Netty还是Mina：深入研究与对比（一）](#)》

《[选Netty还是Mina：深入研究与对比（二）](#)》

《[NIO框架入门\(一\)：服务端基于Netty4的UDP双向通信Demo演示](#)》

《[NIO框架入门\(二\)：服务端基于MINA2的UDP双向通信Demo演示](#)》

《[NIO框架入门\(三\)：iOS与MINA2、Netty4的跨平台UDP双向通信实战](#)》

《[NIO框架入门\(四\)：Android与MINA2、Netty4的跨平台UDP双向通信实战](#)》

《[Netty 4.x学习（一）：ByteBuf详解](#)》

《[Netty 4.x学习（二）：Channel和Pipeline详解](#)》

《[Netty 4.x学习（三）：线程模型详解](#)》

《[Apache Mina框架高级篇（一）：IoFilter详解](#)》

《[Apache Mina框架高级篇（二）：IoHandler详解](#)》

《[MINA2 线程原理总结（含简单测试实例）](#)》

《[Apache MINA2.0 开发指南（中文版）\[附件下载\]](#)》

《[MINA、Netty的源代码（在线阅读版）已整理发布](#)》

《[解决MINA数据传输中TCP的粘包、缺包问题（有源](#)

[码\)_》](#)

《[解决Mina中多个同类型Filter实例共存的问题](#)》

《[实践总结：Netty3.x升级Netty4.x遇到的那些坑（线程篇）_》](#)

《[实践总结：Netty3.x VS Netty4.x的线程模型](#)》

《[详解Netty的安全性：原理介绍、代码演示（上篇）_》](#)

《[详解Netty的安全性：原理介绍、代码演示（下篇）_》](#)

《[详解Netty的优雅退出机制和原理](#)》

《[NIO框架详解：Netty的高性能之道](#)》

《[Twitter：如何使用Netty 4来减少JVM的GC开销（译文）_》](#)

《[绝对干货：基于Netty实现海量接入的推送服务技术要点](#)》

《[Netty干货分享：京东京麦的生产级TCP网关技术实践总结](#)》

>> [更多同类文章](#)

(原文链接：<https://yq.aliyun.com/articles/88122>，有改动)