



runyon的博客

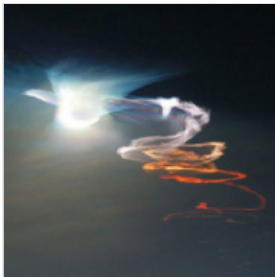
<http://blog.sina.com.cn/finaldaytmp1982> [订阅] [手机订阅]

[首页](#) [博文目录](#) [图片](#) [关于我](#)

个人资料

正文

字体大小: 大 中 小



runyon

微博

加好友

发纸条

写留言

加关注



博客等级: **18**
博客积分: **876**
博客访问: **292,670**
关注人气: **26**
获赠金笔: **15**
赠出金笔: **0**
荣誉徽章:

相关博文

[武汉](#)

[一米爱拍照](#)

[利威尔与团长的命中注定! 兵长外
动漫菌姐](#)

[1249: 试借取几分泰然
雨烟-n](#)

[5月1日赛事参考:俄超主场球队可高
李小五](#)

[《喜欢你》周冬雨率真随性演技呆
影评人太侃](#)

[品茗谈天地, 赏花语人生
晋文](#)

[一个坑折射的六种人生 \(2017.4.10
龙飞扬0101](#)

[托福阅读加试汇总](#)

谈谈网络编程中应用层(基于TCP/UDP)的协议设计 (2013-04-27 19:11:00)

转载 ▼

标签: tcp udp 协议设计 it 分类: 编程

对于初涉网络编程的开发人员来说,在通信协议的设计上一般会有所困惑。一般的网络编程书籍上也较少涉及这方面的内容。估计是觉得太简单了。这块确实是不难,但如果不了解,又很容易出篓子或者绕弯路。下面我就来谈谈基于TCP/UDP的协议设计。

1、基于TCP的协议设计

TCP是基于流的协议。但大部分网络应用一般会有个更小的处理单元,我们称之为帧(FRAME)。

• 是否分帧

如上所述,大部分网络应用是需要分帧的。举IM为例,用户登录是一个帧,用户发送文本信息是一个帧。少部分应用可以不需要分帧,比如:echo服务器,接收到什么直接回复即可;转发服务器,同样是接收到数据直接转给目标机器;更常见的情况是一个TCP连接只发送/处理一个请求之后就直接关闭,这种也就没必要分帧了。

考虑到除了学习网络编程,没人做echo server。所以只要服务端不是一次连接只处理一个请求,或者纯转发,就应该采用分帧的设计。

• 如何分帧

注意:帧是业务处理的单元,是具体应用Care的,但这不关TCP的事情!初学者往往认为tcp这端write一次,tcp那端就会read一次,然后惊呼“粘包”、“丢包”,其实这都是程序处理不当。在这边推荐一本书籍《TCP/IP协议详解 卷1》,挺薄的,看完可以减少很多对TCP的错误认识。实际上发送方发送一帧,接收方可能要N次才能读取完成,而且可能同时读到下帧的数据。那要怎么在接收方把一帧数据不多不少的读取出来呢?

常用做法有两个:基于长度和基于终结符(Delimiter)。基于长度,就是在帧前先发送帧的长度,一般用固定长度的字节来发送此长度,比如2个字节(最大帧长不能大于65535),4个字节。(ps:我也见过使用可变长度的字节来发送此长度,比如netty中的ProtobufVarint32FrameDecoder,看代码那是相当的蛋疼,我觉得完全是折腾自己,强烈不推荐。)使用基于长度的分帧方式,接受方处理流程一般是这样:“读取固定长度的字节->解析出帧长->读取帧长字节->处理帧”。

基于终结符(Delimiter),最典型的应用就是HTTP协议了,使用\r/n/r/n作为终结符。使用基于终结符的分帧方式,接收方的处理流程一般是这样:“读数据->在读取的数据中定位终结符->没找到,将数据缓存->继续读数据->定位终结符->找到终结符,将终结符之前的数据作为一帧进行处理”。

使用终结符的方式务必要考虑转义问题,不然在帧的数据中出现终结符,乐子就大了。

注意不管采用哪种方式,在开发的时候都需要考虑最大帧长的问题。不然如果对方说要发送4G长度的帧(恶意or程序错误),真的去new 4G字节的缓存;或者对方一直发送数据,没有终结符。都可能造成程序内存耗尽。

一般来说,基于长度的分帧方式。开发更简单,程序执行效率也更高,使用更广泛些。基于终结符也不是一无是处:可读性更好,容易模拟和测试(如用telnet)。下面重点讨论基于长度的分帧方式。

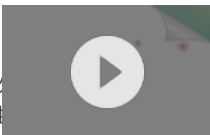
• 基于长度的的帧设计 (length based frame design)

一般来说,我们会将帧分为帧头(frame header,一般是固定长度)和帧体(frame body,一般是可变长度,也有固定长度的)。如上所述,最简单的帧头只要一个字段——帧长。但在实际应用中,一个典型的帧头可能还有以下字段:

a) 消息类型(message type): 在一个网络应用中,往往有多种类型的帧。比如对于IM,有登陆/登出/发送消息/.....。接收方需要根据帧头的消息类型字段,解码出不同种类的消息,交给相应处理模块进行处理。也就是帧的结构是Length-Type-Message,Length-Type可以视为帧头,Message是帧体。消息类型一般也是使用固定长度,比如Length 4个字节,Type 4个字节,那么帧头的长度就是8个字节。接收方处理流程:“读帧头长度字节数据-解码帧头获得长度和消息类型-读帧体长度字节数据-根据消息类型解码消息-处理消息”。Length-Type-Message结构的帧设计是使用最广泛的,普适性最好也最精简的设计。

b) 请求序列号(serials): 这个不是必选项,但我觉得对于非echo式的服务(echo式的服务:总是客户端发送请求-服务端针对该请求应答,应答保证严格按照请求顺序),加上这个字段肯定不后悔。这样对于乱序(如果有消息队列后台线程池,很正常)的执行结果,才能够和请求对上号,从而做出正确的处理。的服务端要保证响应的严格有序,是比较麻烦和影响性能的。

c) 版本号(version): 很多人这么用,但我觉得大部分情况下这不是个好主意。帧头应该放需要的字段。而版本号可能只有少数包如登录会用到,所以放到登录包体里可能更合适。单独本工作量会比较大,开发起来会比较繁琐易错。至于担心解码失败,更好的方式是采用类似Protobuf这种可以向



2017-5-7

佳峰老师

如何说服别人相信你的主意才是最

洛阳人才网官博

下周A股王者归来的概率大吗?

轩阳论市

更多>>

推荐博文

20170421【早评】反抽后

王祖蓝和李亚男大曝婚姻保鲜秘籍

台湾科技挣扎，人祸大于天灾?

收入份额=市场份额，虎嗅想干什

传奇的谢幕，谈岩田聪和他的任天

家常主食轻松做之——培根香葱花

盘点2015最惊艳流行的婚礼蛋

非洲荒漠“精灵怪圈”引发诸多猜

20170327【早评】多空将

大白鲨海滩搁浅 路人

查看更多>>

谈谈网络编程中应用层(基于TCP/UDP)的协议设计_runyon_新浪博客

下兼容的编解码方案。

注意：在帧头设计时应该要尽可能的精简和通用，因为帧头长度是每个帧都需要的额外开销。如果某个字段（如序列号）只有少数帧会使用到，完全可以放在帧体里去。反之，如果某个字段大部分包都有，却不定义在包头，会导致难以统一处理，增加开发工作量。这些需要根据具体业务需求来进行权衡，没有统一的答案。举个例子，Length-Type-Message结构适用于大部分情况，但如果业务要求每个帧都需要表明操作者，在帧头增加UID字段变成Length-Type-UID-Message，程序的开发会更简单。

• 帧体的设计

帧体就是字段的集合，举个例子，登录帧体包含用户名、密码这两个字段（只是举例，现实的登录包往往复杂得多）。在帧体设计上，大家往往也是八仙过海各显神通。比如基于XML、json，基于字段Pos（举登录包为例，就先写/读用户名，再写/读密码。这种方式不是太好，很难向下兼容：比如登录包需要在用户名和密码间加一个用户状态，如果服务端/客户端没有同步升级，就会斯巴达）。我甚至见过狂野得离谱的直接使用C struct的，这种脑残到爆：兼容性渣不说，类对齐（可以用pragma pack避免不一致）、byte order、机器字长都会造成麻烦。

比较推荐的做法：骚年，用Google Protobuf吧！如果要可读性好，json相比XML更省带宽。

2、基于UDP的协议设计

一般来说，UDP的服务器要比TCP简单得多（不过如果要实现基于UDP的可靠消息传输，就当我说）。而且udp本来就是基于数据包的协议。write/read是可以一一对应的（不考虑丢包），所以不需要有长度字段/终结符。但是要注意：为了避免丢包率过高，udp包的长度一般不应该大于1500字节（大概，为了安全起见，我一般保证小于1K嘿），如果数据量较大，就需要分包了，这是比TCP麻烦的地方。

典型的UDP的协议设计就是：Type-Message。Type长度固定，用于说明消息类型；Message是消息体，和tcp的帧体设计同样即可。

谁看过这篇博文

落差	5月6日
维子	5月1日
祁衍Hankow	4月29日
-陨落星	4月20日
通和	4月17日
蓝天下的我	4月17日
1848546292	4月14日
苦行边缘	4月11日
用户32057...	4月2日
__ICE__	3月17日
新生产者	2月7日
fipo514984	2月3日

分享：

阅读(3077) | 评论 (1) | 收藏(0) | 转载(1) | 喜欢▼ | 打印 | 举报

已投稿到： 排行榜

前一篇：对软件架构的一些思考(V2.0)

后一篇：QT程序如何获知鼠标enter/leave通知栏图标

评论 重要提示：警惕虚假中奖信息 [发评论]

林彬泽不要那么2啊

嗯，说得很对，用json的时候也是通过长度来分包的。

2014-3-2 15:03 来自 林彬泽不要那么2啊 的评论 回复(0)

发评论

更多>>

坚持不买你

股市

陈永聪

被裁了

贴你

登录名： 密码： [找回密码](#) [注册](#) ☒ 记住登录状态

☒ 分享到微博 ☐ 评论并转载此博文

按住左边滑块，拖动完成上方拼图

发评论

以上网友发言只代表其个人观点，不代表新浪网的观点或立场。

[< 前一篇](#)

[对软件架构的一些思考\(V2.0\)](#)

[后一篇 >](#)

[QT程序如何获知鼠标enter/leave通知栏图标](#)

新浪BLOG意见反馈留言板 不良信息反馈 电话：4006900000 提示音后按1键（按当地市话标准计费） 欢迎批评指正
新浪简介 | About Sina | 广告服务 | 联系我们 | 招聘信息 | 网站律师 | SINA English | 会员注册 | 产品答疑

Copyright © 1996 - 2017 SINA Corporation, All Rights Reserved
新浪公司 版权所有