

idea's blog

高性能分布式 Web 和网络应用开发, 架构设计, 移动应用开发

[首页](#) [文章精选](#) [简单想法](#) [SSDB](#) [iComet](#) [Cpy](#) [Tovi](#) [CocoaUI](#) [留言](#) [关于](#)

炮打TCP – 关于一而再再而三的粘包拆包问题的大字报

2018-06-13

C/C++语言编程, Computer System

Views: 3172 | 10 Comments

TCP 所谓的粘包和拆包问题，是技术圈里最奇葩的问题之一！

一而再，再而三，就跟傻逼的中国球迷支持中国足球队一样，前赴后继。有时候同一个人多次在犯同一个错误，有时候是前脚一个犯错了后脚又来一个还犯同样的错。即使是最优秀的程序员，也会在这个问题上栽跟头，思维甚至很难转过弯，很久才能意识到自己的错误。而低水平的程序员就更不用说了，很多人到死都没有理解这个错误并解决掉，只是逃掉了而已。

我们固然可以认为原因是某些人学艺不精，但那么多的人，其中包括无数的优秀程序员在 TCP 粘包和拆包问题在犯错误，难道我们不能说，这其实是 TCP 自身的原因吗？

在我看来，这个问题的出现，原因就在于 TCP 协议是有原罪的 -- 也就是 TCP 协议所谓的“流式”协议。所以，我要炮轰 TCP！

经过几十年的验证，除了少数几个网络协议会用到 TCP 所谓的流式特性之外，没有任何应用协议使用流式特性。我们必须承认，所有的应用层协议都是基于报文的协议，而不是流式协议。而某些名字中带有“流（Stream）”字样的协议，如 RTP，流媒体等，其本质是无数小体积的报文按顺序拼接而成，根本就和 TCP 的流式没有任何关系！

那么我们就可以确定，数据的本质是报文，流数据是某类报文数据的一种伪称。事实，TCP 的流就是基于 IP 报文的。

因为“流”是一个伪抽象的概念，所以流式协议是违反人的天性和事物的内在逻辑的。万物的本质是报文。这因为如此，“流”所引出的粘包拆包问题，就必然会一而再，再而三，大量地出现。

炮轰之后，我们要怎么解决问题呢？

由于 TCP 协议已经成为事实上的基础，所以淘汰掉 TCP 是不可想象的。我们要做的是，找到正确的编程代码，解决粘包拆包问题。经过无数人的探索，以及无数人一次又一次重复的愚蠢错误的反证，我发现了解决 TCP 粘包和拆包问题只有一条路径，没有第二条！我断言，所有和我的解决方案不同的代码，都是错误的。

彻底解决 TCP 粘包和拆包问题的代码架构如下：

```
char tmp[];
Buffer buffer;
// 网络循环：必须在一个循环中读取网络，因为网络数据是源源不断的。
while(1){
    // 从TCP流中读取不定长度的一段流数据，不能保证读到的数据是你期望的长度
    tcp.read(tmp);
    // 将这段流数据和之前收到的流数据拼接到一起
    buffer.append(tmp);
    // 解析循环：必须在一个循环中解析报文，应对所谓的粘包
    while(1){
        // 尝试解析报文
        msg = parse(buffer);
        if(!msg){
            // 报文还没有准备好，糟糕，我们遇到拆包了！跳出解析循环，继续读网络。
            break;
        }
        // 将解析过的报文对应的流数据清除
        buffer.remove(msg.length);
        // 业务处理
        process(msg);
    }
}
```

这段代码是终极地解决 TCP 粘包和拆包问题的代码！

这段代码之所以正确，是因为它包含了两个循环：网络循环和解析循环。

IT牛人



带有此标志，表示本博客
已经被加入到IT牛人博客
聚合网站。

[加关注](#) [ideawu](#) (粉丝9003)

Polls

你想加入百度吗？

- ☐ 想
- ☐ 不太想
- ☐ 不想

[投票](#) [查看结果](#)

Categories

-未分类 (2)

C# (1)

C/C++语言编程 (68)

Cocoa (15)

Computer System (61)

Diary (89)

etc (84)

IT技术和评论 (75)

Linux (84)

MacOSX (27)

Music/Movie (24)

Python (13)

SSDB (69)

Web (163)

Java Web (16)

PHP (55)

网络编程 (37)

资讯 (3)

高性能Web架构 (23)

Recent Posts

炮打TCP – 关于一而再再而三的粘包拆包问题的大字报

PHP 获取周一，上个月的正确做法

给Nginx设置默认主机

快速排序算法(QuickSort)的代码实现

Mac 下最好用的看图软件 Tovi 免费下载了！

NSView NSImage NSData转换

国内公有云（云主机）服务商选择

国内某品牌云主机SSH远程后无法在终端显示远程路径的问题

基于 TCP UDP 协议的实时流媒体的实时性分析

SIP INVITE 会话建立过程

Recent Comments

suyanlong: Buffer buffer；缓存也要有最大的限制，不然网

网络循环用于从 TCP socket 中读取流式数据，每一次读取到的数据的长度是不可预期，也就是，读取到的数据长短不一，无法保证，这就是所谓“流式”引出的问题。

而解析循环的功能是从拼接后流数据中，尝试解析出多个报文。注意，是多个报文，不是一个。因为所谓的粘包问题存在，所以可能是多个，而不是一个。如果解析不成功，那说明是遇到了拆包问题，我们继续读网络数据。

你只需要死记硬背上面的正确代码即可。不死记硬背也一样，最终你还是要得出和我相同的结论写出和我一样的代码。那么，何不现在就死记硬背呢？

最后，附上经典的错误代码：

```
tcp.read(tmp, HEADER_LEN);
header = parse_header(tmp);
tcp.read(tmp, header.body_len);
body = parse(tmp);
```

这样的代码当然是错误的，这么简单代码怎么可能是对的？如果对了，TCP 还是 TCP 吗？

如果你认为本文有用，请关注这个 GitHub 项目：https://github.com/ideawu/FUCK_TCP 让更多人一起炮打 TCP！

该项目还提供了模拟粘包和拆包的代码，你如果不信邪，可以写一个自己的 client 试试。

Related posts:

- 1. 关于TCP粘包和拆包的终极解答
- 2. 经典的 TCP socket 读取报文错误
- 3. 通过 HTTP POST 发送二进制数据
- 4. 在Linux进行IO的正确姿势
- 5. 使用 Channel 进行可靠传输

Posted by **ideawu** at 2018-06-13 15:05:11

PHP 获取周一，上个月的正确做法 »

didadi: 你好，请问对于java客户端的，大量的SSDB对象创建会

tkboy: @ideawu: 这句话给人的感觉不是像讨论技术，而是就

tkboy: @ideawu: 你这说的，我上面的描述是看不懂么？是

tgf: 关键问题是解析部分需要写成状态机或者用一个corouti

ideawu: @tkboy: 你看不懂吗？

tkboy: 你的喷点是啥？TCP不能说是流协议么？ 人家RFC

ideawu: @Levi: 更正：不是避免，是应对。

Levi: 为什么使用循环解析报文能避免粘包？

ideawu: @Thomas: 不算是 trick，不然还能放哪？

Tags

ajax Apache BPM C1000K CocoaUI CocoaUI源码解析 comet Cpy Facebook Firefox GIPS icomet iOS iphp JavaScript jQuery KeyValue KV LevelDB libevent mis MySQL nginx NoSQL phpssdbadmin Redis rocksdb RTP SIP SSDB TCP VoIP WebRTC WordPress wordpress插件 Xcode ZendFramework 交互设计 程序员 视频聊天 语音引擎 语音聊天 阿里云 面试IT业界顶尖企业所应该知道的 10道题 高并发

Links

Amit's A* Pages 游戏开发 IT牛人博客聚合 JS 在线统计图表制作工具 My English Blog Tovi.app Mac 下的看图软件 Unreal Engine 3 开发者中心 搞笨蛋 – 互联网最新搞笑图片

Meta

Entries (RSS) Comments (RSS) Log in

10 Responses to "炮打TCP – 关于一而再再而三的粘包拆包问题的大字报"




suyanlong 2018-07-01 at 17:35:03

Buffer buffer；缓存也要有最大的限制，不然网络攻击，势必造成内存溢出，

既然buffer已经有最大限制了，意味着需要结构化的字节流也要限制，即：业务`包` < buffer大小 < MAX_BUFFER_SIZE && socket buffer大小 < buffer大小

关于那些TCP粘包、黏包的说法，我觉得他们就是这些东西认识不清楚。


Reply



tgf 2018-06-17 at 18:38:40

关键问题是解析部分需要写成状态机或者用一个coroutine或promise或者某种monad之类的结构包起来

Reply



tkboy 2018-06-15 at 13:15:39

你的喷点是啥？TCP不能说是流协议么？

人家RFC 793原文描述是

The TCP is able to transfer a continuous stream of octets in each direction between its users by packaging some number of octets into segments for transmission through the internet system

注意这里的词用的是“is able to”能处理的意思，不是"TCP is stream protocol"吧？

Reply

 **ideawu** 2018-06-15 at 14:34:07


@tkboy: 你看不懂吗?

Reply

 **tkboy** 2018-06-19 at 14:10:32

@ideawu: 这句话给人的感觉不是像讨论技术，而是就是为了喷而喷哇。如果是这样那讨论结束了。

Reply

 **tkboy** 2018-06-19 at 14:09:26

@ideawu: 你这话说的，我上面的描述是看不懂么？是哪里描述的不对么？为啥讨论技术的方式不是直入话题说出自己的说法以及反驳点，而是说这种没分量的话？

Reply

 **Levi** 2018-06-14 at 09:52:42


为什么使用循环解析报文能避免粘包？

Reply

 **ideawu** 2018-06-15 at 11:15:33

@Levi: 更正：不是避免，是应对。

Reply

 **Thomas** 2018-06-13 at 19:42:38

这里的trick 关键也是把拆包的逻辑放在 parse 这里吧

Reply

 **ideawu** 2018-06-13 at 19:47:49

@Thomas: 不算是 trick，不然还能放哪？

Reply

Leave a Comment

suyanlong

Name *

59293841@qq.com

Email *(will not be published)

Website

Post Comment

You are subscribed to this entry. [Manage your subscriptions.](#)