

云风的 BLOG

思绪来得快去得也快，偶尔会在这里停留

[« 从 Lua 5.2 迁移到 5.3 | 返回首页 | 如何拼接 PVR 压缩贴图 »](#)

为什么 skynet 提供的包协议只用 2 个字节表示包长度

skynet 提供了一个 lua 库 [netpack](#)，用来把 tcp 流中的数据解析成 长度 + 内容 的包。虽然使用 skynet 的同学可以不使用它，而自己另外实现一套解析协议来解析外部 TCP 数据流（比如 skynet 中的 redis driver 解析 redis server 的数据流就是用的换行符分割包），但依然有很多同学询问，能不能自定义包头长度。

这里的这个库定义的协议中，包长度是用 big-endian 的 2 个字节表示的，也就是说一个包的长度不得超过 64K。这让很多人很为难。已经几次有同学建议，把长度放宽成 4 个字节，因为他们的应用环境中大部分包不会超过 64K，但总有少量超过这个限制的。

历史上，skynet 的 gate 还是用 C 实现的时候（那个版本依然可以使用）的确可以自定义是使用 2 个字节还是 4 个字节表示包长。但经过一番考虑，我还是去掉了这个选择。

一个好的库，应该简洁，且引导使用者用正确的方法做正确的事情；而不应该提供让用户犯错的机会。在和游戏客户端通讯的时候，如果你只采用一个 TCP 连接，那么允许很长的数据包本身就是错误的。甚至 64K 都太大。

游戏通常需要比较快的响应速度，如果你允许在单个 TCP 连接中插入一个太大的数据块，比如 100K，那么在比较弱的网络条件下（例如手机网络），处理这个包可能就需要超过 1 分钟的时间。而这么大的数据块，在业务逻辑上大多不期待立刻能发出或收到的。一个典型的应用场景就是用户在拍卖行中查询所有的上架物品，如果把所有返回数据都放在一个数据包中，很容易就变得很大。而查询大量这个操作，用户本身就对立刻回应没有期待。

而在单个 TCP 连接上，这样一个大数据块会阻塞住整个信道，后面本来需要快速送到的数据全部被延迟了。

如果你想在业务层做一个心跳包检测网络是否超时，很容易就把心跳包拦在后面。而通常网络处理层不会提供接口让业务层探知是否正在接受数据（skynet 的 gateserver 就没有提供这样的接口，虽然它很容易提供，只需要修改几行 lua 代码），只能在一个完整的数据包收到后才会交给业务层处理。

正确的做法是，在长度+内容这个协议上再加一个层次。加一条协议叫大数据块，允许把一个大数据块分几段发送。可以在这条协议中加上数据块 id，在后面引用这个数据块的包中附上数据块 id 即可。

为什么要用这种比较绕的方式，而不直接把包头从 2 字节改成 4 字节？当你做这个设计时，就已经表明你重视了上面提到的问题。

当你把数据包都分割的比较小时，才能实现单个 TCP 连接上承载多个信道的能力。对于网络游戏，并不是所有的数据包都是上下文相关的，你可以看成隐含着有多条线索。比如聊天频道的信息和场景同步的信息就是相互独立的。skynet 为这种场景还提供了额外的 [socket api](#) 支持。socket.lwrite 可以把一个字符串（一个数据包）写到低优先级通道。只有等默认通道（高优先级）的包全部发送完后，低优先级通道上的包才至少被发送一个（单个包可以保证原子性）。

比如，你可以用它来发送聊天信息，就不会因为聊天信息泛滥把其它重要数据包都塞住。同样，你可以用来发送被分割后的大数据包。如果同时你还有很多其它重要的数据需要传输给客户端，那么这些数据块就会被打散穿插在其间。

当然，你也可以把所有给客户端的数据全部用 lwrite 发送，而仅仅把心跳包放在常规高优先级通道，可以保证心跳频率更稳定。

另外，采用 4 字节的包长度还有一个安全漏洞，可能被攻击利用。

一般的分割包的代码，在收到包头时，都会根据长度信息预先分配出相当长的空间，等着后面的数据达到后填入。如果攻击者不断在新建立的连接上发送一个恶意的长度信息，比如 2G，服务器内存很容易被快速消耗光。

早期 skynet 的 gate 实现时，采用的是共享一个固定长度的 ringbuffer 的实现，可以避免这种攻击。但新的版本由于不再允许 4 字节长度，就没有做特别处理了。

如果你的应用环境非常特殊，坚持一个允许更大长度的数据包协议。那么我建议你慎重的实现一个分包模块，而不是简单的把 netpack 库中的 2 改成 4。

云风 提交于 January 8, 2015 11:23 AM | [固定链接](#)

COMMENTS

变长编码可能更合适，固定2个byte长度还是存在不足的，导致心跳包也至少需要3个bytes，大于64k的包还需要特殊处理

比如像mqtt里面就是变长最大4个bytes来表示长度，变长编码意味着前7位encode data，第8位表示是否有下一个byte，最大支持256m大小的包，最小的心跳包就2个bytes，满足绝大多数需求

Posted by: Anonymous | (8) [April 5, 2015 03:40 PM](#)

还可以使用3字节

Posted by: rhn | (7) [February 16, 2015 05:47 PM](#)

表示非常理解。

在看了SPDY协议，了解了TCP单通道，多并发的优点。

小的数据包，可以让套接层响应，分发的频率更高。同时，对数据包分组后，可以实现并发，同时传输N个数据流。因而，将一个个小包合并后，可以组成大数据包，而且大数据包的大小也可以不限制于字节数量。

Posted by: [LuJyKa](#) | (6) [February 9, 2015 04:11 PM](#)

如果把这个设计放到一个前提之下，那就是很好的设计。

毕竟，带着某种目的的设计不会差到哪里去。

作为我，是支持这种设计的，4字节的长度很多时候是一种伤害。

Posted by: [风雪连城](#) | (5) [January 20, 2015 12:59 PM](#)

在看skynet时也对此表示过疑问。

今天得看到云风对此设计上的思考。

我认为各有利弊。所谓的利其实我也认为是在偷懒，比如游戏内好友列表（详情）等，或者你所说的拍卖行，普通程序员估计就直接组包发了呗，文艺点的就分段分页发：)

另外文中提到的分优先级的发送队列，倒是个不错的思路。

Posted by: kevin | (4) [January 12, 2015 04:30 PM](#)

1. 对于服务器之间的内部通信实现4字节长度还是有意义的

2. 那个收到包头长度预分配空间的理由有点牵强

Posted by: [楼兰](#) | (3) [January 8, 2015 07:12 PM](#)

thanks

Posted by: Anonymous | (2) [January 8, 2015 05:18 PM](#)

又了解到一些设计上的知识

Posted by: [黄金](#) | (1) [January 8, 2015 12:34 PM](#)

POST A COMMENT

非这个主题相关的留言请到: [留言本](#)

名字:

Email 地址:

为了验证您是人类，请将六加一的结果（阿拉伯数字七）填写在下面:

URL:

☐ 记住我的信息？

留言：
(不欢迎在留言中粘贴程序代码)

提交