

SIP 协议栈的设计及性能优化

郑智

武汉理工大学, 武汉 (430070)

E-mail: joko1985@163.com

摘 要: 本文介绍了会话初始协议 (SIP) 的基本内容, 采用分层、模块化的方法设计了 SIP 协议栈, 并提出了一种优化的内存管理机制。测试结果表明协议栈的性能有了较大的提高。

关键词: SIP, 协议栈, 内存管理

1. 引言

会话初始协议 (SIP) 是用来生成、修改和终结一个或多个参与者之间的会话的信令协议^[1]。这些会话包括因特网多媒体会议, 因特网 (或任何 IP 网络) 电话呼叫和多媒体发布。会话中的成员能够通过多播或单播联系的网络来通信。SIP 支持会话描述, 允许参与者在一组兼容媒体类型上达成一致。同时通过代理和重定向请求到用户当前位置来支持用户移动性。

2. SIP 协议栈的总体框架

SIP 协议栈采用开放式的分层体系结构进行设计, 如图 1 所示。这种框架的特点是上层模块可以调用下层模块提供的接口, 反向调用则通过回调函数来实现^{[2][3]}。因此, 只要模块间接口的语法和语义进行了明确的定义, 各子模块都可以进行更改和替换, 从而最大限度的提高了 SIP 协议栈代码的可移植性。同时这种设计允许从上而下的裁剪, 子模块及其下层支撑模块可以独立运行, 增加了子模块代码的可复用性。

SIP 协议栈分为对话层、事务层、编解码层以及传输层四个层次, 每层都提供了相应的 API 接口。图 1 说明了协议栈的总体结构以及各层之间的关系。

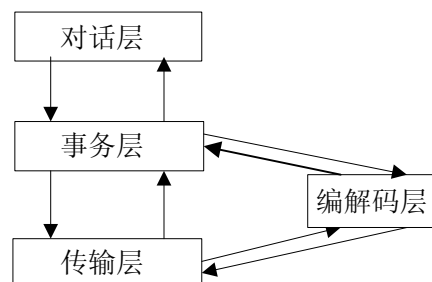


图1 SIP 协议栈的总体架构

协议栈的最底层是传输层, 其承担的任务是侦听网络端口 (默认 SIP 端口号 5060)。当传输层收到对端 SIP 实体发送的原始 SIP 消息, 调用编解码层提供的 API, 做简单的语法检查后 (例如是否缺少必备的消息头), 把消息解码成内部的数据结构, 并在指定的端口 (例如 5060 端口) 发送至事务层。由于 SIP 协议同时支持 UDP 和 TCP, 因此传输层对这两种传输协议的 socket 接口都进行了封装。

事务层实现了 RFC3261 中规定的 4 类事务模型, 将传输层投递过来的消息, 纳入对应的状态机中处理, 把处理后的事件上报给对话层处理。而对于从对话层传送过来的消息 (例如发送 INVITE、BYE 等等), 直接调用编解码层提供的 API, 把内部数据结构编码成文本

的 SIP 消息，然后调用传输层提供的 API 把 SIP 消息发送至网络。

对话层位于事务层上方，它负责把事务层上报的单独的一个个消息组织成呼叫的信息。例如，对于同一个呼叫的 INVITE 和 BYE 消息，在事务层是互相独立的，各自按照各自的状态机运作，而对话层则把两者纳入到同一个呼叫中进行管理。对话层还负责和协议栈用户的通信。

3 动态内存管理

3.1 传统内存管理方式

任何 SIP 协议栈，在解码过程中都无法避免动态内存申请。这是因为 SIP 协议的消息变化非常大，不可能设计一个固定大小的数据结构来存储解码的结果。例如，根据协议，INVITE 消息每经过一个 proxy，该 proxy 将增加一个 Via 消息头到此消息中，这样，当 INVITE 消息到达最终的被叫的时候，它携带的 Via 个数是不定的。对这样的 INVITE 解码生成的数据结构，必须用动态的链表或者指针数组来存储不定数目的 Via 结构。对许多协议栈而言，当需要增加一个 Via 节点到上图的链表中的时候，简单地调用了 malloc 函数。动态内存分配方式如图 2 所示。如果解码 N 个消息头，需要等待 N 次全局性互斥锁时间，大大的降低了协议栈的性能。

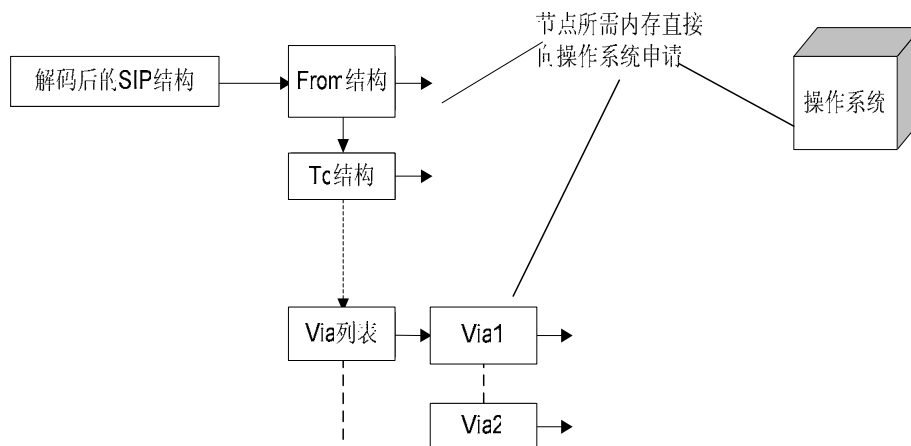
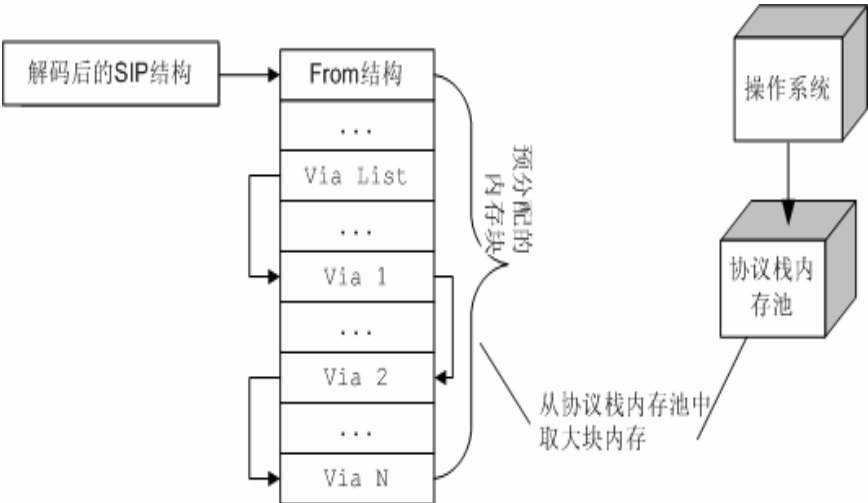


图 2 动态内存分配

3.2 改进后的内存管理方式

SIP 协议栈的设计采用了内存池（MemPool）技术，并在此基础上做了改进。经典的内存池技术，是一种用于分配大量大小相同的小对象的技术。通过该技术可以极大加快内存分配/释放过程。而本文将内存池技术进行了改良，将内存块大小改为非固定长度。即第 1 次申请 32 字节内存，第 2 次申请时 64 字节内存，第 3 次申请 128 字节内存，以此类推。由于 SIP 协议栈的设计采用了分层、模块化的方法，因此对于各个模块所需的内存块大小进行了预分配。

图 3 描述了 SIP 协议栈中编解码模块内存缓冲区的设计方案。解码后的所有 SIP 消息头的数据结构可以共用同一内存缓冲区，这样的处理，在消息流程中减少了内存申请的开销；同时解码过程字符串指针直接指向消息内存块部分，减少了内存拷贝的开销；而且内存块可以串联起来，提供了无限制的内存使用的可能。



4. 性能测试

本文对运行 SIP 协议栈的呼叫服务器进行性能测试，得到了 SIP 协议栈内存使用率以及 CPU 占用率的数据。在每一次呼叫中，呼叫服务器需要处理 14 条 SIP 消息。其中使用大话呼模拟器来模拟主叫 SIP 用户和被叫 SIP 用户。测试拓扑图如下：

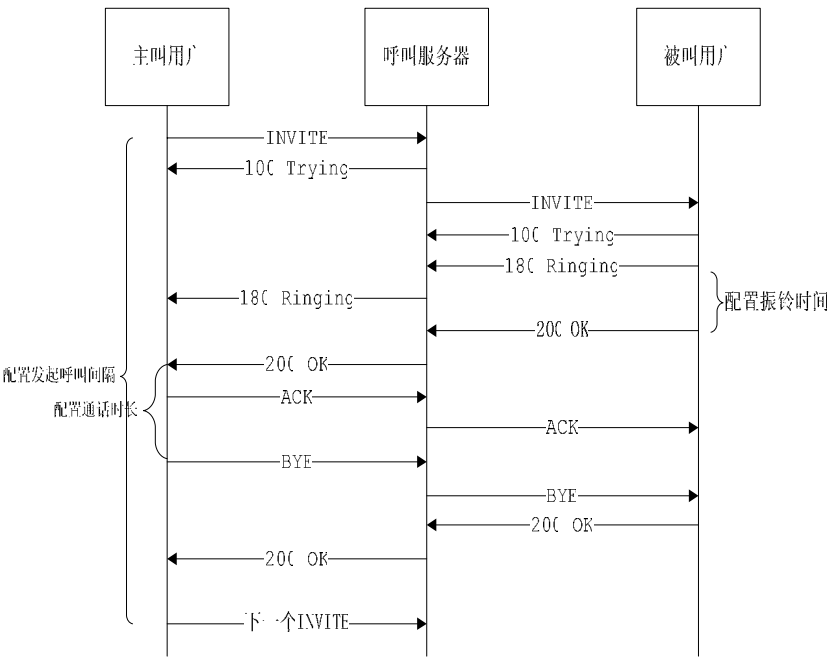


表 1 中的测试结果表明，本协议栈的性能得到了很大的提高，能够在大呼叫量的情况下保持绝大部分呼叫的正常接续，基本能符合网络运营商的要求，显示了改进方案的可行性和有效性。

表 1 测试结果

测试条件 测试结果	正常大话呼	超长时间大话呼	过载大话呼
成功率	100%	99%	60%
启动时每秒呼叫数量	15	40	70
发起呼叫总数	108000	1547120	392015
接受呼叫总数	108000	1546000	235209
高峰期 SIP 协议栈的 CPU 使用率	12.8%	55.8%	95%
时间长度	2 小时	12 小时	4 小时
忙时每秒呼叫数量	10	14	25

5. 结论

SIP 协议凭借其简单、易于扩展、便于实现等诸多优点越来越得到业界的青睐，并且已经成为 3G IMS 和 NGN（下一代网络）中的核心协议，并且市场上出现越来越多的支持 SIP 的客户端软件和智能多媒体终端，以及用 SIP 协议实现的服务器和软交换设备，SIP 的功能和应用越来越广泛，其相应的规范和扩展也就越来越多，这就要求我们在 SIP 协议栈的实现上，必须有一个易扩展的结构框架。而本文设计的 SIP 协议栈的框架已经能够适合当前 SIP 协议的发展要求，并且提供优化的内存缓冲区机制，使不同模块处理同一消息时，共用同一内存缓冲区，最小化内存申请拷贝的次数，提高了协议栈的性能。

参考文献

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, et al. SIP: Session Initiation Protocol[S]. RFC3261, June2002
- [2] 雷正雄,朱晓民,廖建新. SIP 协议栈的设计与实现[J].现代电信科技,2004,(3):17-20.
- [3] 白建军, 彭晖, 田敏. SIP 解密[M]. 人民邮电出版社, 2003.

The Design and performance optimization of SIP Stack

Zheng Zhi

Department of Information Engineer, Wu Han University of Technology, Wu Han (430070)

Abstract

This paper introduces the Session Initiation Protocol (SIP).We design the SIP stack by utilizing layered, modularized technology, and presents an enhanced memory management mechanism. The test results show that the stack's performance had improved greatly.

Keywords: SIP, protocol stack, memory management