

SIP 代理服务器的设计与实现

Design and Implementation of SIP Proxy Server

(中国人民解放军 66455 部队 075000)

刘泽新 (中国科学院计算技术研究所 100080)

(中国科学院研究生院 100039)

何 彬 (中国科学院计算技术研究所 100080)

摘要:SIP(Session Initiation Protocol,会话初始化协议)是一个应用层信令控制协议,用来创建、修改和终结一个或多个参与者参加的会话,这些会话包括 IP 电话、分布式多媒体、多媒体会议等。本文基于 SIP 协议,提出了一个 SIP 代理服务器的设计与实现方案。

关键词:SIP 代理服务器 事务 客户事务 服务器事务

1 引言

SIP 代理服务器是 SIP 网络中的一个中介组件,是 SIP 网络的重要组成部分之一。代理服务器的主要功能是路由功能,通过代理服务器对 SIP 消息的转发,用户请求被正确地传输到目的用户,实现多媒体会话。

2 SIP 协议简述

2.1 SIP 发展概述

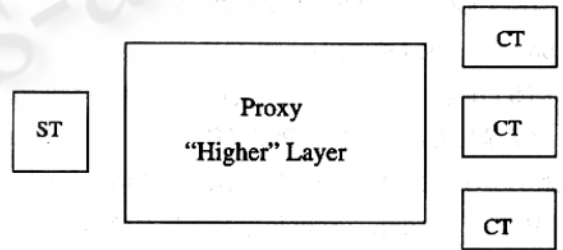
SIP 协议是由 IETF 的 MMUSIC 工作组制定的,1999 年提出第一版本 RFC2543,2002 年提出第二版本 RFC3261。SIP 是一个基于文本的作用于应用层的多媒体会话信令协议。它的主要功能是在 IP 网络上创建、修改和终结一个或多个参与者参加的会话,这些会话包括 IP 电话、分布式多媒体、多媒体会议等。SIP 并不是一个垂直型通信系统,它必须和其它协议(如 RTP、SDP 等)共同使用来构建一个完整的 IP 网络多媒体通信系统。但它又不依赖于其它任何协议,在功能上它是独立的,是一个开放的分布式协议。

2.2 SIP 网络组件

SIP 网络结构采用客户机/服务器结构,包含两个基本组件:SIP 用户代理(SIP User Agent)和 SIP 服务器(SIP Server)。SIP 通信将在这两种组件之间进行。SIP 消息是客户机与服务器之间通信的基本信息单元。

2.3 SIP 代理服务器分类

根据对事务处理方式的不同,SIP 代理服务器分为有状态代理服务器(stateful proxy server)和无状态代理服务器(stateless proxy server)两类。有状态与无状态代理服务器的区分是 SIP 适应 IP 网络现状的一种有效方法。在网络的核心使用无状态代理,提供快速、高效的路由,在网络的边缘使用有状态代理,为用户提供更好的服务。



T = Client Transaction ST = Server Transaction

图 1 有状态代理服务器结构

(1) 有状态代理模式。有状态代理服务器要在路由过程中存储通信事务信息,直至这个事务结束并被释放为止。每个有状态代理服务器均有一个服务事务模块(ST)与多个客户事务模块(CT),它们被一个更高层的称为代理内核(Proxy Core)的模块管理。其结构如图 1。

CT 与 ST 是代理服务器事务层的主要功能模块,它们分别完成 UAC 和 UAS 的功能。一个 ST 对应多个 CT,当一个 ST 接收到一个请求后,它通过高层的代理内核产生派生代理,再由派生代理去调用多个 CT 转发该请求,这样一个代理服务器可以向用户的多个注册地址转发该请求。

(2) 无状态代理模式。无状态代理服务器不包括四层结构体系中的事务层。无状态代理服务器直接从传输层接收 SIP 消息并对消息进行路由,它不存储与事务有关的任何信息。

3 SIP 代理服务器的实现

3.1 SIP 协议栈(见图 2)

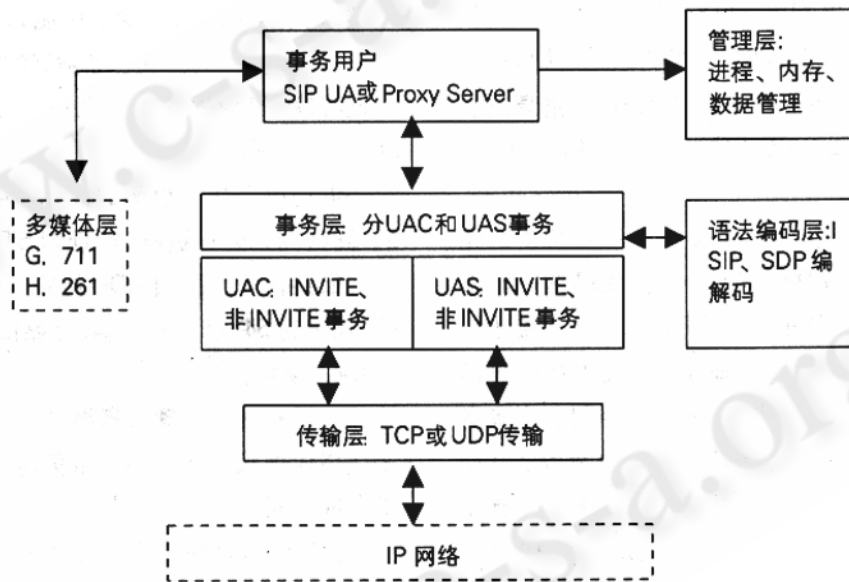


图 2 SIP 协议栈

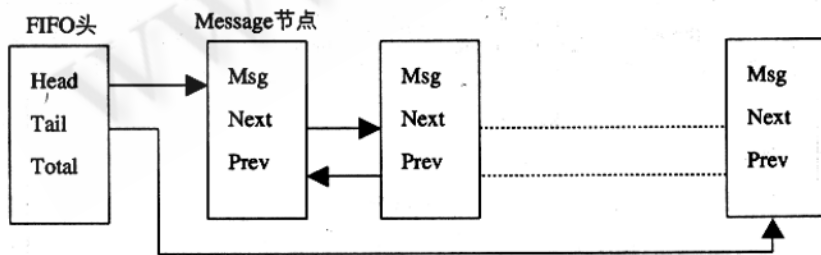


图 3 消息对列数据结构

(1) 管理层。它是为软件设计的需要而建立的, SIP 标准中并不包括此层。它的主要功能包括:系统环

境初始化、日志文件的建立、数据结构的定义与操作、内存空间的申请与释放及用于对各种应用进程进行出错、超时处理的管理线程的建立。

(2) 多媒体层。只在 SIP UA 中用到,完成用户端多媒体音、视频信息的编码、解码、压缩与解压缩。

(3) 事务用户。除无状态代理服务器外,每一个 SIP 实体就是一个事务用户。如 SIP UA 或有状态代理服务器。

(4) 事务层。它是 SIP 层次结构的核心层,负责会话的控制与管理,主要内容是保存事务的状态、匹配请求消息的响应消息、对消息进行超时、重传处理。

(5) 传输层。负责端到端的连接与通信。采用套接字方式进行 TCP 或 UDP 数据传输。

(6) 语法编码层。SIP 消息采用基于 UTF - 8 (RFC2279) 的文本编码格式,语法信息采用扩充的巴

柯斯范式 (BNF) 形式描述, 语法编码层的主要功能是完成 SIP 请求与响应消息的构造与解析。

3.2 数据结构

(1) 消息队列。传输层对 SIP 消息的管理与记录采用 FIFO 队列的形式, 其数据结构如图 3。

(2) 事务记录。由于代理服务器在通信过程中存在大量的 SIP 事务, 因此对 SIP 事务内容的记录与操作采用 Hash 表方式, 以提高服务器的事务检索速度, 保证服务器的处理能力。

3.3 有状态代理服务器的实现

3.3.1 语法编码层

SIP 消息分为标题行 (命令行或状态行)、头域、消息体三部分, SIP 消息的构造与解析也分为三个模块。代理服务器对消息体是透明的, SIP 代理服务器并不对消息体进行解析。该层主要由 SIP 标题行、相关头域的构造与解析函数构成。SIP 是基于文本的协议, 消息的构造与解析函数比较简单, 主要由字符串操作函数完成。

3.3.2 传输层

代理服务器初始化时, 创建接收、发送、解析三个线程, 接收、发送两个 FIFO 队列。接收线程按 TCP 或 UDP 协议采用套接字方式接收 SIP 消息, 放入接收队列, 由解析线程从队列中调出 SIP 消息, 交本层处理函数, 处理函数根据消息是请求还是响应分别上交事务层的相关接口。对于出向呼叫, SIP 消息由事务层交给处理函数, 处理函数将其放入发送队列, 由发送线程从队列中读取该消息, 经网络发出。其消息流程图见图 4。

3.3.3 事务层

事务层由 CT 与 ST 两个部件组成, 分为 UAC INVITE、UAC NINVITE、UAS INVITE、UAS NINVITE 四种状态机, 程序设计中要按 RFC3261 状态机的要求, 分别设计相应函数。

传输层处理函数接收到 SIP 请求消息时, 将该请求交给 ST 处理模块, 如果是第一次接收到该事务, 则要对事务进行记录同时上交代理服务程序, 代理程序处理后, 交给 CT 处理模块。如不是第一次, 则说明是重传的请求, 直接交给 CT 模块按 SIP 请求的类型, 分别调用 UAC INVITE 或 UAC NINVITE 状态机进行处理, 状态机函数将该 SIP 请求加入发送队列, 经传输层发送线程发出。当接收到响应消息时, CT 模块根据该响应所对应的请求类型调用响应状态机函数, 根据响应消息的状态码, 更改状态机的状态参数, 然后再将该响应上交代理程序, 代理程序处理后再将该响应下传给 ST 模块, 加入发送队列。

调用每个状态机函数时, 都要启动该事务状态机的超时、调死处理进程, 实现对 SIP 事务处理进程的管理。如果传输层采用 UDP 传输, 还要启动事务重传进程, 以保证 SIP 消息的正确、可靠传输。

3.3.4 事务用户

对于有状态代理服务器, 它就是代理主程序, 对于无状态代理服务器不存在事务用户。有状态代理服务器实现过程如下:

(1) 代理处理响应。如果代理服务器接收到响应消息, 首先要与已存储的请求消息进行匹配, 取消不匹

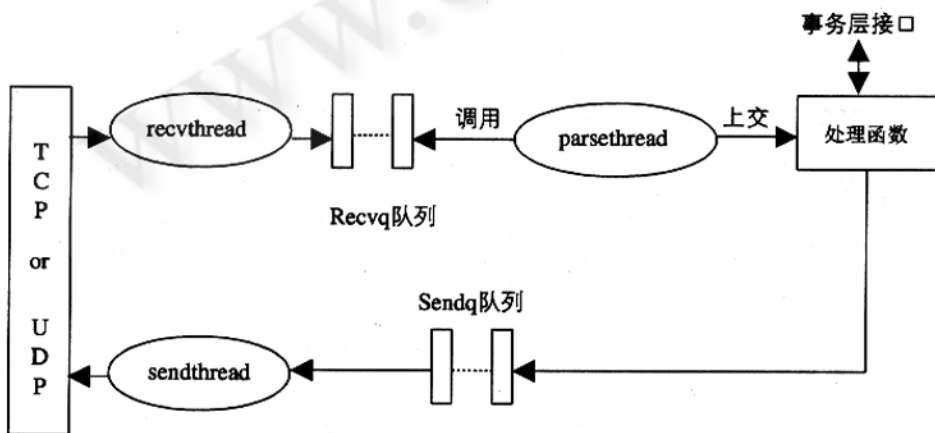


图 4 传输层消息流程图

配响应,然后对成功响应再判断是临时响应还是最终响应。

① 如是临时响应,代理服务器不转发,退出代理程序;

② 如是最终响应,再判断是对 INVITE 的响应还是对其它请求的响应,如果收到 INVITE 请求的响应,则由 Proxy_Cancel 函数取消 INVITE 的重发,然后对响应消息的 Via 头域进行解析,按与请求相反的路径向下一站转发,并由 Delete_Top_Via 函数从响应消息的 Via 域中去掉本代理服务器的地址行。

(2) 代理处理请求。对于收到的请求消息,代理主进程 (Proxy_Server) 判断该请求是本域请求还是远程请求:

① 如果是本域请求,通过 Get_Contacts 函数查询用户注册服务器或定位服务器,找出目的用户所注册的 SIP 地址组 Address (即最终用户的逻辑地址与他的联系地址的动态映射关系表),为目的用户的每一个联系地址派生一个代理进程即派生代理进程 (Proxy_Fork) 和一个超时处理进程 (Proxy_Timeout),由派生代理进程调用事务层 CT 模块中的请求进程,向地址组中的每一地址转发该请求,同时,启动响应、出错处理进程,对请求事务过程进行管理。

② 如果是对远程用户的请求,则从用户的请求资源标识 (Request-URI) 中解析出目的地址并与定位服务器中已存储的目的地址构成一个新的地址组,采用同①一样的派生代理方式经传输层向下一跳转发,下一跳站点地址的查找采用 DNS 方式。

3.4 无状态代理服务器的实现

由于它没有事务层,因此它的实现方式非常简单,对 SIP 消息的代理仅在传输层完成。接收到消息后,判断是请求还是响应。

① 对于请求,按有状态代理服务器的方法判断是本域还是远程请求,本域请求处理方法是注册服务器或定位服务器中查出最新的目标联系地址直接发送,即它只向一个目标地址发送 SIP 请求消息,而不象有状态代理服务器那样派生出多个代理进程,向目标地址组中的所有地址发送请求消息;远程请求的处理办法是通过解析请求资源标识,获取目的地址后,向下一站点转发。

② 对于响应按 SIP 消息的 via 域中的第二个地址

向下一站发送。无状态代理不需要为每一请求或响应派生代理进程和进行事务与状态的记录。

3.5 进程管理

对通信过程中的事务进程采用链表结构进行记录,并由事务管理进程对它们进行管理,以便在事务进程收到响应或超时、出错时,调用相应的功能函数。

以派生代理进程为例说明其工作过程:代理服务器系统初始化时,生成三个管理线程,分别为连接、监测、解除线程,一个在用进程链表,一个释放进程链表。当代理服务器生成派生代理进程 (Proxy_Fork) 时,连接线程将 Proxy_Fork 进程及其超时处理进程 (Proxy_Timeout) 的信息加入到在用进程链表中,监测线程不断遍历该链表,对派生代理进程的计时器进行查询,当计时器为零时,则把它从在用进程链表中清除到释放进程链表,解除线程从释放进程链表中取出派生代理进程的记录信息,调用它的超时处理进程,根据超时处理进程的所设定的内容,进行后续的处理。

4 结束语

SIP 是一个近两年才提出的基于 IP 网络的新协议,SIP 协议软件的开发还处在不断完善中,本文依据 RFC3261 建议,设计了一个代理服务器软件,但这个代理服务器软件只实现了 RFC3261 建议中关于代理服务器的主要功能,尚有许多部分,如 SIP 消息安全性、SIP 路由记录、系统监控等未在软件中实现。

参考文献

- 1 J. Rosenberg, H. Schulzrinne, G. Camarillo. SIP: Session Initiation Protocol IETF RFC 3261 June 2002.
- 2 A. Johnston, S. Donovan, C. Cunningham. Session Initiation Protocol (SIP) Basic Call Flow Examples IETF RFC 3665 December 2003.
- 3 Gonzalo Camarillo, SIP 揭密,白建军、彭晖、田敏等译,人民邮电出版社,2003.6.
- 4 Olivier Hersent, David Gurle, Jean-Pierre Petit, IP 电话—基于分组的多媒体通信系统,邝坚、戴志涛译,人民邮电出版社,2000.11.
- 5 于明俭、陈向阳、方汉, Linux 程序设计权威指南,机械工业出版社,2001.4.