

# 短信互联互通实现方案

周智洪 肖 璋 陈东明

(暨南大学电子信息工程系, 广东 广州 510632)

**摘要** 参照中国移动的短信网关协议(CMPP)、联通短消息网关协议(SGIP)、中国电信小灵通短消息网关协议(SGMP)以及公众互联网短消息网关协议(SMPP), 构建一个短消息互联互通的网关系统. 论述该网关系统的通信实现方式以及计费、加密认证、监控的模块.

**关键词** 短消息网关, 多线程, 线程池, 套接字

**中图分类号** TN 915.02

**文献标识码** A

随着移动短消息业务的蓬勃发展, 诸如中国移动、联通、中国电信都在各自的移动平台上提供了短消息业务. 但因各自所采用的通信协议不一, 无法互通, 尤其中国电信推出的小灵通目前还无法与现存的移动、联通这两大运营商进行短消息互通. 这使移动业务提供商开展各项业务受到限制. 为此, 以互联互通为目的的短消息综合网关成为迫切的需要. 作为一个综合的、可扩充性的短信互联互通方案, 主要功能表现为短消息中心互联网关、CP/SP 业务接入网关, 以及与固网短消息互联网关、移动短消息互联网关、联通短消息互联网关和内部业务接入网关, 等等<sup>[1]</sup>. 同时, 要求综合网关具备大容量处理能力、高可靠性、灵活的路由、计费、流量控制, 支持各种广义的 SP 互连等.

## 1 系统逻辑结构

整个系统按照逻辑结构可以分两层——功能支撑层和接口层. 整体结构, 如图 1 所示. 功能支撑层是整个短消息网关的核心部分. 它包括给移动、电信等短消息运营商之间短消息包的路由、转发, 系统平台对数据包的计费, 用户管理以及系统监控等功能. 接口层包括接入接口层和业务接口层. 从图 1 可以看出, 接入接口层主要是根据各短消息中心及网关使用的不同协议, 建立相应的接口连接.

通过系统留出的接口, 业务提供商(SP)或因特网内容提供商(ICP), 可方便的进行用户应用层的开发. 从而, 达到组件化、可重用性强、易升级扩展的商业化三层结构.

## 2 系统功能层次结构

整个系统按照功能层次来分类, 主要由 5 个部分组成. 下面, 我们分析其细节部分.

### 2.1 通信处理模块

整个通信模块运行, 在 RedhatLinux 7.3 平台下. 所有通信连接都使用流套接字(Socket), 采用 POSIX 多线程技术. 建立线程池<sup>[2]</sup>, 通过线程的调度使每个模块中都得到独立的线程进行处理, 最大限度地减少因资源而阻塞线程运行等. 下面, 我们谈谈具体的通信流程、各模块之间的关系以及其中的一些关键技术.

**2.1.1 通信处理流程及路由** 图 2 中箭头表示产生的各线程, 箭头方向表示数据的流向. 整个流程分成上行和下行两个部分. 首先, 上行接收模块从短消息中心(SMSC)或网关(SMG)接收到短消息包后, 不做任何处理就放入上行接收缓冲队列. 接着上行处理模块通过调度线程池中的空闲线程, 从接收缓冲

队列中取出接收下来的消息包进行分析处理. 根据分析结果, 数据会有两种流向. (1) 如果需要对 SMSC

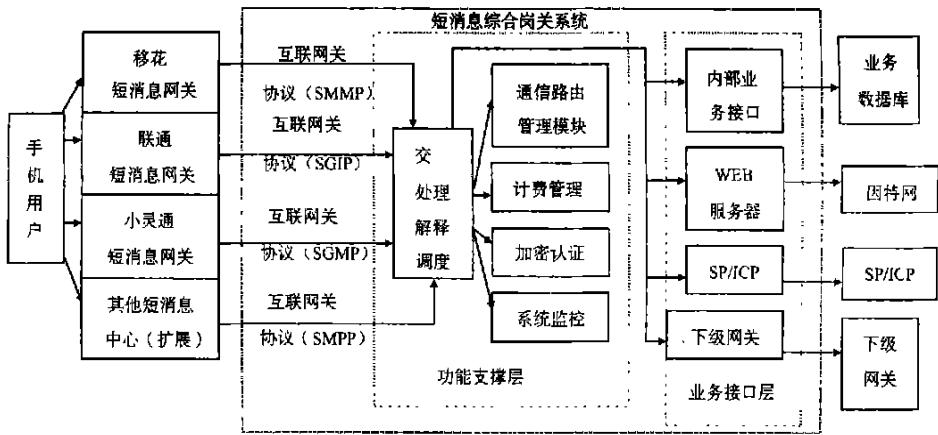


图 1 网关系统结构图

(SMG) 作出应答, 则产生应答包; 如果是不同的 SMSC(SMG) 之间的短消息转发, 则把要转发的短消息包放入下行发送缓冲队列中. 下行发送模块通过调度线程池中的空闲线程, 从下行发送缓冲队列中取出待发送数据包转发至相应 SMSC(SMG). (2) 如果 SMSC(SMG) 下发至业务提供商(SP) 的短消息包, 则放入上行发送缓冲队列. 上行发送模块通过调度线程池中的空闲线程, 从上行发送缓冲队列中取出待发送数据包转发至相应 SP. 同理, 从 SP 发送来的短消息数据包, 首先进入下行接收模块, 整个处理流程类似上行处理过程.

2.1.2 收发线程与缓冲队列关系及缓冲队列的实现 对应各缓冲队列, 我们采取各自终端节点对应一个独立的接收或发送线程, 各线程对应一个独立缓冲队列. 现用上行接收缓冲队列为例, 解释这种策略的思路和好处. 如图 2 所示, 上行接收模块对应每个 SMSC(SMG) 建立一个独立的接收线程后, 并对应自己的接收缓冲队列. 这样做的有两个好处, 其一是便于对接收数据包进行流量控制. 当对应的缓冲队列已经存满, 则阻塞对应的接收线程. 这样在不影响其它接收线程的情况下, 防止了缓冲区溢出及丢包现象的出现. 其二, 可以对不同的 SMSC(SMG) 设定优先级别. 优先级别高的 SMSC(SMG) 在上行处理模块中, 会从线程池中优先分配空闲线程, 从该对应的缓冲队列中提取数据进行处理. 对于缓冲队列单

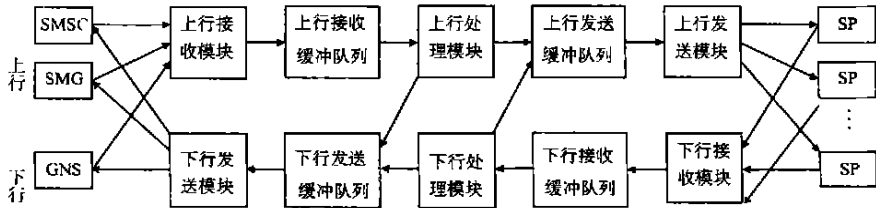


图 2 通信系统模块图

元的实现, 采用的共享内存的方法. 用 calloc 分配  $N$  个内存块. 选择适当的内存块大小, 使一个内存块作为缓冲队列单元, 单元可以容纳大部分类型的协议数据单元(PDU), 某些特大数据包用多个内存块来存储. 每个内存块的第 1 个字节, 用来标识此内存块的数据是否已处理完. 第 2 个字节标识此内存块, 属于整个数据包的第几个分段. 数据包的第 1 个分段的第 3~6 个字节, 表示开始被处理的时间. 这种线程间共享内存法的缺点是浪费一些内存空间, 优点是便于统计缓冲队列的占用量, 便于用多线程技术来并发处理各数据包<sup>[3]</sup>.

2.1.3 线程池管理与自动性能调整 采用多个线程池完成不同的功能模块. 对于包括上行处理线程池、下行处理线程池、上行发送线程池、下行接收线程池等. 而且对应不同 SMSC(SMG) 所采用的不同的应用层协议, 用一个专门的线程池来进行处理. 每个线程池对应一个调配线程, 由调配线程给空闲的处理线程分配任务, 利用一个全局的线程池状态结构体链表来对线程池管理<sup>[4]</sup>. 全局的线程池信息结构体链表, 定义为 struct threadpool { int poolid; /\* 线程池编号 \*/ int size; /\* 线程数 \*/ int dispatcher; /\* 调配

线程号\* / struct workthread \* workers; /\* 工作线程信息结构体链表头指针\* / } . 工作线程信息结构体链表, 定义为 struct workthread{ int threaded; /\* 工作线程号\* / int poolid; /\* 所属线程池\* / int state; /\* 线程状态(0 为空闲, 1 为忙)\* / }<sup>[5]</sup>. 任务繁重时(线程池中已无空闲线程), 调配线程会实时产生线程加入到线程池中, 以增强并行处理性, 调整系统性能. 总线程数不能超过一定上限(上限是由操作系统和机器配置情况决定). 空闲线程太多时, 由系统监管线程(或定时处理线程)通过线程间的信号机制, 杀死线程池中过度冗余的空闲线程. 当发现线程泄漏的时候, 系统监管线程产生新线程替代, 以维护系统性能.

2.1.4 数据流量控制的实现 CMPP v2.0 只支持发送端的流量控制(向 SMG 发送 CMPP - Submit 或向 SP 发送 CMPP - Deliver 时要实现). 未收到应答的包达到一定数量就暂停发送. SMPP v5.0 支持接收端的流量控制(接收处理来自 SMSC 的应答包中的 congestion - state, 控制自己的发送速度;同时向 SMSC 发送应答时加入 congestion - state 以控制对方的发送速度). 每 M 个应答包中都有一个应答包含有拥塞状态 TLV, 以反映所对应的接收缓冲队列占有率( congestion - state  $\theta$  100). 从而, 实现从接收方进行流量控制. SGIP v1.2 中没有流量控制方面的规定, 但可利用数据包中的保留字段参照 SMPP v5.0 的方式来实现.

## 2.2 计费管理

该网关的设计目的不仅可供移动运营商使用, 同时也可用于 SP 的进行移动商务开发. 因此, 要有完善的计费管理功能, 并提供各种样式的计费信息提取接口.

## 2.3 系统管理和业务管理

系统管理和业务生成, 由系统管理工作站和业务管理工作站和业务生成工作站组成. 系统管理提供对系统维护管理、运行操作、运行状态监视等功能, 提供维护人员对本系统进行日常维护和管理. 业务管理和业务生成, 提供管理人员对业务的开放、关闭、业务定制、业务流程管理、业务数据维护、报表生成和管理、业务统计分析等等业务方面的维护的管理.

## 2.4 加密处理和密钥管理部分

加密处理部分由加密机构成. 加密机一般采用工业控制计算机, 以完成和相应的加密单元构成. 系统为保障支付的安全性, 对系统中的数据库数据、数据报文等数据进行加密处理的加密/解密处理. 系统和本系统以外的所有设备和关联系统之间的通信, 均采用严格的加密处理. 本系统主要使用 MD5(信息摘要)算法和 RSA(数字签名)算法实施加密处理. 密钥管理部分由密钥管理工作站构成, 负责管理和处理短信中心与短信网关, SP 与短信网关之间加密密钥的管理.

## 2.5 系统监控部分

系统监控模块提供对系统实时运行状态监控, 提供系统各个部件障碍、网络连接故障、系统运行异常等告警功能. 及时提醒系统维护人员, 对系统进行维护和故障处理. 系统监控部分独立于应用系统. 应用系统定期将系统运行过程中的各类事件写入 MIB 库, 监控部分通过 SNMP 2.0 网管协议实时读取系统运行情况.

# 3 业务接口层

## 3.1 WEB 服务器

处于一级防火墙和应用服务器之间, 用户和操作/管理员(营业厅)都可通过互联网按不同权限进行访问. Web 服务器通过应用服务器透过二级防火墙向客户服务器提交服务请求, 实现相应的资料管理、客户服务和业务管理等功能. 按权限实现部分资料管理和客户服务功能. UNIX 下运行, 用 Weblogic, Jsp 技术实现.

## 3.2 数据库部分

由数据库服务器、磁盘阵列、数据备份设备、备份服务器等组成. 它存储用户信息、SP 业务信息、通信交易过程记录, 以其它系统运行所需数据. 采用实时备份的方式. 主用数据库和备用数据库内容一致, 对数据库的一切修改动作都对双备份同时进行. 备用数据库可执行查询操作, 以分担主用数据库的负担, 并可在主用数据库被破坏时用于恢复数据. UNIX 下运行 ORACLE 9i 数据库系统. 数据库操作用 Pro C 编程和调用存储过程实现<sup>[6]</sup>, 采用双机热备份、配磁盘冗余阵列.

## 4 系统优化

作为电信级别的互联互通网关,通信的稳定和速度是首要指标.这取决于实现系统的并发性,从而产生大量的线程.决定系统运行状况的一大关键是,线程的属性设置和调度策略.CPU 是面对轻进程 LWP 进行调度的,对于需要快速响应的线程可以绑定在 LWP 上,并且设置较高的优先级以便实时调度.比如,维护连接的线程、接收数据包的线程.至于数据包的处理线程、监听 SP 连接的线程,可以是分时调度的非绑定的线程.为防止线程间的影响,各个处理线程一般都是独立的.为了更好的实时并发性,可以采用多处理器技术,以更好地发挥多线程的优势.

## 5 系统测试方案

由于移动、联通等移动运营商的特殊性,直接在移动网络进行实际的测试无可行性.同时,为了突出测试网关系统通信模块的稳定性,在北京某 IT 公司帮助下,设计出测试方案.由该公司提供两台服务器,同时以  $2\,000\text{条}\cdot\text{s}^{-1}$  的速度发送短信.服务器 1 发送的短信目的方为服务器 2 的 IP,这是测试网关系统的短信网关之间的转发功能.服务器 2 发送的短信目的方设为我们(我方)的 1 台 PC 机.机上运行目的方自行编写的短信接收模拟程序,这是测试网关系统对 SP 的转发功能.短信网关程序运行在目的方提供的奔 3 处理器,128 M 内存的机器上.

系统经过 1 h 测试,整个系统的不丢包率达到 100%.从而,证明本文所述的网关方案切实可行,实为一个稳定的系统.

## 6 结束语

限于篇幅,只能对系统的通信、路由、监控等主要模块进行简要的描述.对所采用技术的优缺点,也进行简单的比较.本研究为各移动业务开发商进行短消息的商业开发,提供一个有效的解决方案.

### 参 考 文 献

- 1 周学泳,石 丹.短信息(SMS)与 WAP 的开发及应用[M].北京:电子工业出版社,2002.12~13
- 2 Stevens W R 著.UNIX 网络编程[M].施振川等译.北京:清华大学出版社,1999.1~92
- 3 Gray J S 著.UNIX 进程间通信[M].第 2 版.张 宁等译.北京:电子工业出版社,2000.272~362
- 4 Beveridge J 著.Win32 多线程程序[M].候 捷译.武汉:华中科技大学出版社,2002.335~364
- 5 谭浩强.C 程序设计[M].第 2 版.北京:清华大学出版社,1999.263~286
- 6 于自跃.Oracle 9i 开发教程[M].北京:北京希望电子出版社,2002.36~60

## A Plan for Implementing the Intercommunication of Short Message

Zhou Zhihong   Xiao Zhang   Chen Dongming

(Dept. of Electron. Info. Eng., Jinan Univ., 510632, Guangzhou, China)

**Abstract** Referring to such gateway agreements as CMPP, SGIP, SGMP and SMPP, a gateway system is constructed for the intercommunication of short message. The way for realizing communication of this gateway system is discussed. Its modules of charge, encryption and authentication, and monitoring are also discussed.

**Keywords** short message gateway, Multi thread, thread pool, socket