

SIP ALG 面向事务过程的队列调度算法及性能分析

黄永峰¹, 侯冠男¹, 李建庆²

(1. 清华大学电子工程系网络研究所, 北京 100084; 2. 澳门科技大学资讯科技学院, 澳门)

摘要: NAT 成为 SIP 端到端多媒体通信应用中一个巨大的障碍, 如何解决 SIP 应用穿透 NAT 的问题成为当前互联网应用领域研究的热点问题. 论文通过剖析 NAT 的工作原理, 并针对 SIP 协议的信令过程的特点, 提出了面向 NAT 穿透的 SIP ALG 体系结构模型. 同时, 针对 ALG 多路并发处理以及消息存储修改转发的特点, 重点研究了 ALG 对多路呼叫处理的调度策略问题, 提出了基于呼叫事务的多队列优先调度算法, 并建立 ALG 最大同时处理数据数的计算模型. 实际测试表明, 论文所提出计算模型和实际测试结果比较吻合, 证明了计算模型的正确性.

关键词: NAT; 应用级网关; 多队列调度算法

中图分类号: TP393.2 **文献标识码:** A **文章编号:** 0372-2112(2007)07-1223-05

SIP ALG Multi-Queue Priority Scheduling Algorithm Based on Calling Events and the Performance Analysis

HUANG Yong feng¹, HOU Guai nan¹, LI Jian qing²

(1. Network Research Institute, Department of Electronic Engineering, Tsinghua University, Beijing 100084, China;
2. Faculty of Information Technology, Macau University of Science and Technology, Macau, China)

Abstract: NAT is a significant barrier in SIP peer to peer multimedia communication. The problem of penetrating NAT in SIP application is of great research value. Considering the process of SIP protocol, we propose an SIP ALG architecture that solves the problem of penetrating NAT. Moreover, the paper studies extensively on the scheduling strategy of multi process calling. A multi queue priority scheduling algorithm based on calling events is proposed, as well as a model that calculates the maximum simultaneous data number in ALG. The proposed models are verified by testing results.

Key words: network address translation(NAT); application level gateway (ALG); multi queue priority scheduling algorithm

1 引言

基于网络安全和 IPv4 地址资源匮乏等问题的考虑, 目前许多企业内部网络采用了私有 IP 地址, 并通过 NAT(Network Address Translation)来控制与公共网络的通信^[1]. NAT 能够完成私有 IP 地址与全局 IP 地址的相互转化, 并设置相应的分组过滤规则, 让不满足条件的 IP 分组不能够穿透 NAT.

随着互联网的飞速发展, 基于互联网的多媒体通信应用越来越普遍, 其中最典型的的就是 VoIP. 然而, 在 IP 电话的部署过程中, 存在非常大的障碍就是 NAT 穿透问题. 因为 NAT 设备允许一个企业内部网使用私有的 IP 地址, 互联网的路由器只能将数据送到具有全局 IP 地址的设备. 因此 NAT 后面的终端可对于互联网的路由器来说是不可路由的, 这样将导致 NAT 后面的终端

不能接收互联网上的呼叫. 另外, 即使 NAT 后面的终端可以向 NAT 外的终端发起呼叫, 建立了呼叫链接, 但是在媒体传输时, 也存在 NAT 后面的终端将无法接收对方的媒体分组的问题.

目前, 很多人都在研究 VoIP 穿透 NAT 问题, 并提出了很多解决方法和策略. 例如, 美国的 Ridgeway 公司提出的隧道方案^[2]、IETF 制定的 UDP 流协议穿透 NAT 的协议(STUN)^[3]、中间盒(MidCom)^[4]、全代理服务器(Full Proxy)^[5]、应用级网关(ALG, Application Level Gateway)以及会话边界控制器^[6]等. 这些方案, 各有千秋, 都只是解决特定条件下 NAT 穿透问题, 而且大部分的研究主要是局限于系统的实现, 对 NAT 穿透的性能分析和优化研究论文还没有. 本论文设计了基于 SIP 的面向 NAT 穿透的 ALG 体系结构, 提出了基于呼叫事务的队列调度算法, 建立了 ALG 性能的分析数学模型, 并进行了实验

收稿日期: 2006-04-06; 修回日期: 2007-05-15

基金项目: 国家发改委 CNGI 项目; 国家 863 高技术研究发展计划(No. 2006AA01Z4444); 澳门特别行政区科学技术发展基金(No. 372/043/SAF/2005)

© 1994-2010 China Academic Journal Electronic Publishing House. All rights reserved. <http://www.cnki.net>

仿真和验证.

本论文的结构是:第2节分析了ALG穿透NAT穿透基本原理,提出了一种基于SIP(Session Initiation Protocol)的ALG体系结构.第3节SIP ALG多路并发处理中基于呼叫事务的队列调度算法,建立了性能分析数学模型.第4节对所提出的模型进行的实验仿真和验证.

2 SIP ALG工作原理和体系结构

2.1 基本原理分析

与HTTP应用穿透NAT原理有所不同,基于SIP的多媒体通信中的媒体分组传输和SIP信令传输通道是分离的,媒体分组需要通过SIP信令消息中携带IP地址和端口号等信息来实现目的地寻址.因此信令消息在NAT穿透中不仅需要对TCP/UDP层的端口以及IP层的源地址和目的地址进行变换,还需对IP媒体分组中头部地址信息进行变换.而且,对于SIP应用来说,媒体分组头部中的地址信息是通过信令来动态协商,不是静态设定的.因此准确获取每路呼叫的相关地址信息(IP地址和端口),建立地址映射表,并按照地址映射表为每路呼叫的信息分组进行实时、高效转换是SIP ALG设计中需要解决的核心问题.

论文所提出的NAT穿透方法是基于SIP协议的注册机制来实现.SIP协议规定:SIP终端需要周期性的发送Register消息到注册服务器^[1].由于周期性Register信令经过NAT设备,使得NAT设备对通过的消息流始终保持一个确定的端口;同时,当Register消息经过ALG时,ALG就会记录信令穿透NAT时被转换后的IP地址和端口等信息,建立地址映射表,并且将此信息与NAT后面的用户URI等信息进行绑定.另外,SIP ALG对内网和外网发起呼叫的处理方式有所不同,因此ALG在处理一个呼叫之前首先需要判断是呼出呼叫(外网终端呼叫内网终端)还是呼入呼叫(内网终端呼叫外网终端),当ALG接收到一个呼入呼叫时,ALG将通过NAT的地址映射表将呼叫转发给网内被叫方.当NAT后面的SIP终端收到来自外网的呼叫请求后,就可以通过主叫方的Invite消息和被叫方的200(OK)响应消息中SDP(Session Description Protocol)消息体来协商媒体流所使用的IP地址和端口等相关信息,从而来实现NAT的穿透.

2.2 SIP ALG的体系结构设计

ALG实现SIP对NAT的穿透,是通过对流经ALG的SIP请求消息和响应消息进行解析、修改和转发.因此,整个ALG可以分为如下3个功能模块:消息解析模块、修改模块和消息转发模块.

(1) 消息解析模块

消息解析模块主要对流经ALG的SIP消息进行解析,并根据消息分组的From、To、Via等头域中的IP地址

等信息来判断该消息是呼入呼叫还是呼出呼叫,然后,调用修改模块对这些消息中的相关参数进行修改.图2显示了消息解析模块的算法流程.

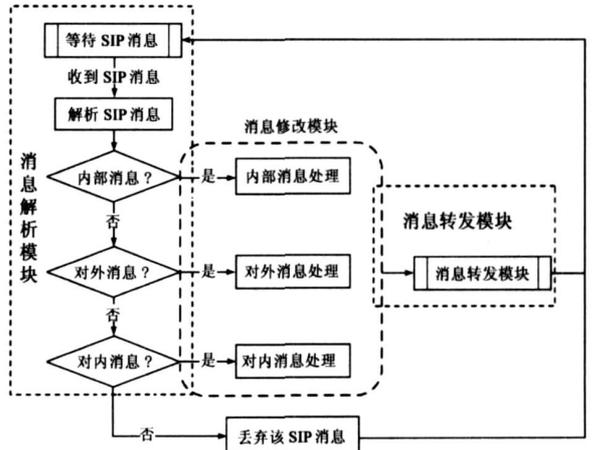


图1 SIP ALG体系整体

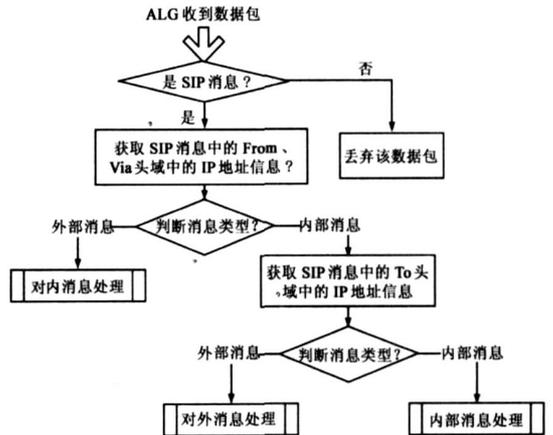


图2 消息解析模块工作流程

(2) 消息修改模块

根据呼出和呼入的不同,消息修改模块对消息中相关参数进行修改.呼入消息的修改主要是根据地址映射表把外网IP和端口号转换成内部IP和端口号,以便外部请求能准确的到达NAT内侧的用户端;呼出消息修改则是根据地址映射表把内部IP和端口号转化成外网可用的IP和端口号,这样来自外网的请求就能准确定位到该地址.图3给出了消息修改模块的算法流程.

对呼叫事务中,需要修改的请求消息有:Invite, ACK, BYE和Register等,而修改的响应消息有100(Tring), 180(Ringing)以及200(OK)等,修改的内容包括这些消息中的Via、From、To、Call ID、Contact等头域中包含的IP地址和端口等信息.另外,需要特别处理的是对SDP消息体的修改,这也是SIP ALG实现其功能的关键所在.因为SDP消息体的交换是实现呼叫双方媒体流通信的前提和关键,它决定了相应的媒体分组中的

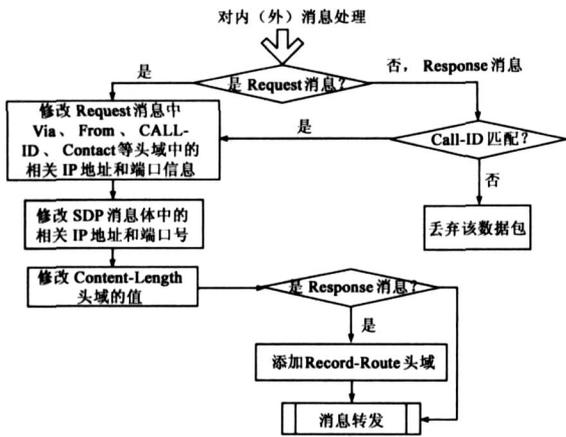


图 3 消息处理模块

地址信息. 例如: 当一个呼叫的媒体分组由外向内需要穿透 NAT 时, ALG 需要分析呼出的 Invite 和 200(OK) 消息的 SDP 消息体中“Media Description”和“Connection Information”参数行, 根据这两行参数描述的地址信息来启动 NAT 上的未用端口, 从而来保证媒体流分组能被正确地转发到内网中的目的地址, 实现媒体流对 NAT 的穿透. 另外, 在对响应消息进行修改时, 还需要通过 Call-ID 来匹配呼叫的上下文环境, 并要在响应消息中加入 Record Route 头域, 以此来告知内网的 SIP 终端, 要求本次呼叫事务中的所有后续消息都要流经 SIP ALG, 这样才能保证 ALG 对 SIP 消息进行正确转发. 最后, 对消息修改完毕后, 需要重新计算消息的长度, 并保存在 Content-Length 参数中.

(3) 消息转发模块

消息转发模块主要是通过控制 NAT 对修改后的 SIP 消息进行转发, 以便完成一个会话的建立. 消息转发模块在 ALG 中建立并维护当前呼叫的上下文环境, 也就是一个地址映射表(如表 1), 通过该表格来记录相关的地址映射关系.

表 1 地址映射信息表

用户号码	源 IP	源端口	目标 IP	目标端口
81000004	10.10.15.44	5060	202.205.11.230	60012

表 1 中的数据表示所有目的地址为 202.205.11.230:

60012 的 SIP 分组都会被 NAT 网关转发到主机 10.10.15.4:5060 上处理. 消息修改模块在进行消息修改之前, 首先需要查询 ALG 内地址映射表, 如果表内已经建立了该用户的信息, 则按此信息对向 NAT 处的 SIP 消息做相应的修改, 否则丢弃该消息.

为了生成相关地址映射信息表, ALG 首先需要分析的 Register 消息. ALG 记录 Register 中的 To 和 Contact 头域, 按它们的参数值在 NAT 设备上启用一个未用的端口, 然后 ALG 记录生成的映射关系, 并保存在地址映射表中. 生成地址映射表后, 消息修改模块再按映射关系修改呼出呼叫的 Invite, Register 等消息, 然后再通过消息转发模块来实现 NAT 的穿透.

3 消息队列调度算法及性能分析

3.1 消息队列调度算法

SIP ALG 作为一个多路并发处理服务器系统, 如何提高系统并发处理性能(即单位时间内实时处理最大呼叫数)是其关键技术, 影响系统并发处理性能最主要的因素是 ALG 消息队列调度算法.

在 ALG 中, 我们采用异步 I/O 消息机制. 异步 I/O 消息处理队列采用主动对象设计模式, 它异步接收和处理多个服务请求, 为了实现多路呼叫并发处理, ALG 使用了线程池来处理 SIP 请求, 设计模式中的代理实现主动对象接口, 服务者提供对象的实现, 代理和服务者在不同的线程中运行, 因此方法调用和方法执行可以并行.

另外, 在 SIP 协议中, 引入了“呼叫事务”和“事务状态”概念^[5]. 所谓呼叫事务是指发起一个请求消息到接收其响应消息的全过程. 为了保证一个事务的“原子”性, 服务器需要维护“事务状态”. 因此, 我们在设计 ALG 的消息队列调度算法时, 提出了一种基于呼叫事务优先的多队列调度算法. 该算法的基本思想是以 SIP 中呼叫事务为队列单位, 不同的事务排列在不同的队列中, 队列之间的调度遵循 FIFO, 且不可抢占, 优先完成一个队列全部呼叫.

该算法的基本过程如图 4:

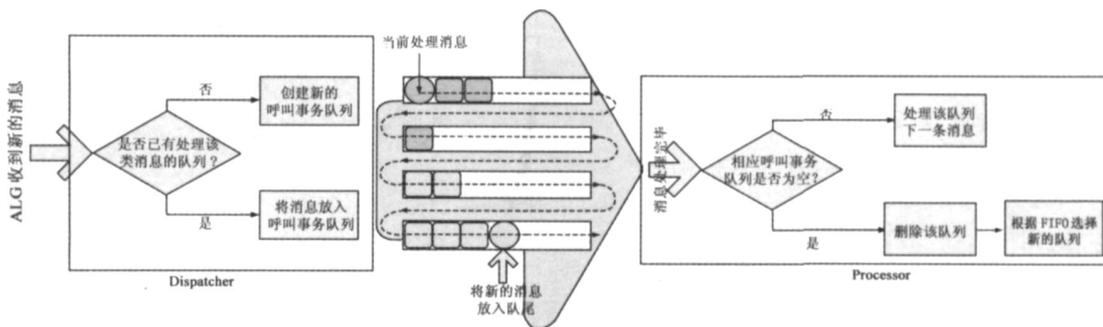


图 4 消息队列调度算法

可以很容易证明, 基于呼叫事务优先的队列调度算法具有如下两个优点:

- (1) 减少呼叫连接的建立时间, 提高了系统的实时性和系统并发处理性能;
- (2) 减少了呼叫事务的维护时间, 同时也减少了系统内存资源的消耗.

3.2 并发处理性能分析

对于一个实时通信系统来说, 最主要的性能是呼叫建立时间; 同理, 对一个 ALG 来说, 关键性能是其并发呼叫的处理能力, 也就是一个 ALG 同时处理的最大话路数, 这两者是相关的. 当一个 ALG 的并发性能增强, 则呼叫的转换时间就减少, 相应地呼叫连接的建立时间也就减少了. 下面根据呼叫事务调度算法的基本过程, 分析 ALG 的并发处理能力与呼叫建立时间等因素的关系, 同时, 建立了 ALG 同时处理的最大话路数的计算模型.

根据呼叫事务队列调度算法的基本原理, 用户在发起呼叫的第 1 个消息(例如: Invite)之后, 希望在尽可能短的时间内得到服务, 也就是说, 希望该消息的事务尽快结束, 从而完成呼叫的建立. 一个呼叫事务基本过程如图 5 所示.

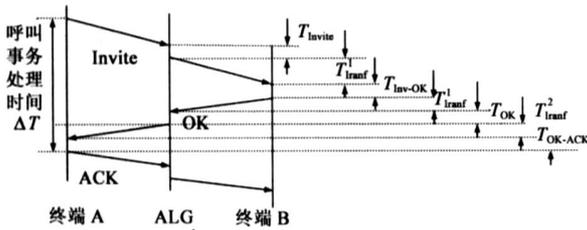


图 5 一个呼叫事务的处理过程

假设 1 个呼叫事务的处理时间为 ΔT , 一个呼叫事务是由 3 个消息组成 (Invite、OK 和 ACK), 每个消息的处理时间分别是 T_{invite} 、 T_{ok} 、 T_{ack} , 同时还假设 ALG 分别到终端 B 和 A 的消息传输时间为 T_{tranf}^1 、 T_{tranf}^2 , 以及终端 B 接收处理 Invite 消息和构造 OK 消息的时间为 T_{inv-OK} , 终端 A 接收处理 OK 消息和构造 ACK 消息的时间为 T_{OK-ACK} . 因此 ALG 处理 1 个呼叫事务的所需要的时间 ΔT 可以表示为式(1).

$$\Delta T = T_{invite} + T_{ok} + T_{ack} + 2T_{tranf}^1 + 2T_{tranf}^2 + T_{OK-ACK} + T_{inv-OK} \quad (1)$$

假设一个 ALG 能同时的最大呼叫处理数目为 N_{max} , 也就是说: 在某时刻内, ALG 接收到呼叫数目不超过 N_{max} 时, 这样就可以保证每个呼叫事务能及时处理, 不会在事务队列中排队等待, 这样也保证每路呼叫事务的处理时间相等. 但是如果 ALG 在某时刻内接收的呼叫事务数大于 N_{max} 时, 呼叫消息则进入相应的事务队列进行排队, 则每路呼叫建立的时间也就相应地延

长.

一般情况下, 我们将呼叫到达的规律是按泊松过程来处理. 假设将新呼叫建立的数量定义为“事件 i ”, 而其时间戳为 T_i , 事件 $i-1$ 与事件 i 之间的时间间隔为 t_i .

假设在时间间隔 D 内恰好有 K 个呼叫到达的概率为 $P_k(D)$, 于是可以表示为:

$$P_1(d) = \lambda * d + o(d) \quad (2)$$

$$1 - P_0(d) - P_1(d) = o(d)$$

$$P_{k+1}(D+d) = P_{k+1}(D)(1 - \lambda * d) + P_k(D) * \lambda * d + o(d) \quad (3)$$

式(3)表示在时间 D 有 $k+1$ 个事件, 且在 D 和 $D+d$ 之间没有事件发生, 或者是在 D 有 k 个事件且在时间 $(D+d)$ 时有 $k+1$ 事件的概率.

对式(3)产生差分方程, 其解为式(4):

$$P_k(D) = (\lambda D)^k * e^{-\lambda D} / k! \quad (4)$$

前面假设了一台 ALG 的最多可以同时为 N_{max} 个呼叫服务, 其他的呼叫按 FIFO 模型排队. 我们进一步假设在 ALG 在单位时间内接收到的呼叫数为 M , 在正常情况下, 一个呼叫事务处理需要时间为 ΔT , 则在一个呼叫处理时间内的呼叫数可以表示为 ΔM , $\Delta M = M * \Delta T$, 且 $\lambda = M$.

于是我们可以算出在同时到达 ALG 的呼叫数目大于 N_{max} 时的呼叫处理时间 w 大于 t 的概率, 如公式(5)所示.

$$P(w > t) = \frac{\frac{N_{max}}{N_{max} - \Delta M} * \frac{\Delta M^{N_{max}}}{N_{max}!}}{(1 + \Delta M + \frac{\Delta M^2}{2!} + \frac{\Delta M^3}{3!} + \dots + \frac{\Delta M^{N_{max}}}{N_{max}!}) + \frac{N_{max}}{N_{max} - \Delta M} * \frac{\Delta M^{N_{max}}}{N_{max}!}} * e^{-t(N_{max} - \Delta M) / \Delta T}$$

通过式(5), 我们可以分析 ALG 在最大同时处理呼叫数 N_{max} 和呼叫事务最小处理时间 ΔT 一定的条件下, 单位时间内接收到的呼叫数 M 与实际呼叫处理时间 t 之间的关系. 同时也通过公式(5)来计算一个 ALG 最大同时处理呼叫数 N_{max} .

4 实验仿真与分析

为了对上述数学模型进行验证, 我们在 P4 2.4G 单 CPU 联想服务器实现 SIP ALG. 操作系统采用 Linux 2.4.20, ALG 采用 OSIP 的协议栈来实现 SIP 协议功能, 网络接口采用 100Mbps 以太网卡, 测试工具采用 TCPdump 网络协议分析软件. 通过分析所捕获的分组的时间戳来计算分组的传输延迟以及分组处理延迟等参数.

通过该实验, 主要目标是测试一个特定的 ALG 的

最大同时处理呼叫数 N_{\max} . 测试方法如下: 首先测试在单路呼叫情况下的一个呼叫事务处理延迟 ΔT 的大小 (如表 2 所示), 然后逐步增加单位时间内的呼叫次数 M , 当 M 增加到一定大小时, 如果 ΔT 的大小发生了明显的增大, 则可以确定此时的 M 值为该 ALG 最大同时处理呼叫数 N_{\max} . (测试结果如表 2 中的 M 测试值所示).

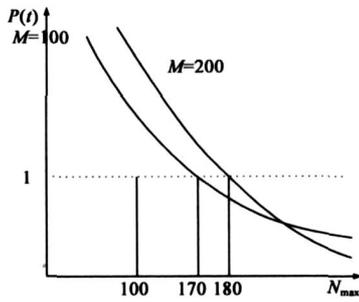


图 6 N_{\max} 与 M 的关系曲线

按照上述测试条件, 我们假设单位时间内发起的呼叫次数不变 ($M=200$), 按照公式 (5) 可以计算 $t = \Delta T$ 的概率 P 与 N_{\max} 的变化曲线, 如图 6 所示.

表 2 ALG 的测试和计算结果

T_{invite}	1.4ms	ΔT (实测)	39.1ms
T_{ok}	1.4ms	M (实测)	200 次/s
T_{ack}	1.4ms	ΔM (实测)	79 次/s
$2T_{\text{invite}}^1 + T_{\text{inviteOK}}$	8.1ms	N_{\max} (计算值)	175
$2T_{\text{invite}}^2 + T_{\text{OK-ACK}}$	26.8ms	N_{\max} (实测值)	182

从实验结果可看出, 论文所提出的 ALG 性能分析的数据模型计算结果和实验测试结果比较接近, 说明了该数学模型是正确的, 对分析和优化 ALG 性能具有一定的理论指导意义.

5 小结

论文根据 SIP 应用的实际需要, 设计了并实现了面向 NAT 穿透的 SIP ALG. 论文在如下 2 个方面进行创新研究: 第 1 提出了基于 SIP 的注册机制来实现 NAT 穿透的方法; 第 2 提出了基于呼叫事务的调度算法; 同时还建立了 ALG 并发处理性能分析模型. 测试表明, 论文所提出的算法能在一定程度上提高 ALG 的性能, 所建立数学模型是正确的, 对 ALG 的设计具有理论指导意义, 所设计 ALG 具有较大的实用性.

参考文献:

- [1] A Gulbrandsen, P Vixie, L Esibov. A DNS RR for specifying the location of services (DNS SRV) [S]. Request for Comments 2782, Internet Engineering Task Force, Feb. 2000.
- [2] A T Campbell, J Gomez, S Kim, A Valk, C-Y Wan, Z R Turnyi. Design, implementation, and evaluation of cellular IP

[J]. IEEE Personal Communications Magazine, 2000, 7(4): 42 - 49.

- [3] D Marples. Naming and accessing internet appliances using extensions to the session initiation protocol [R]. in Proc. of SIP 2000 Conference and Exhibition, (Paris, France), May 2000.
- [4] H Schulzrinne, J Rosenberg. The session initiation protocol: Internet centric signaling [J]. IEEE Communications Magazine, 2000, 38(10): 134-141.
- [5] J Rosenberg, J Weinberger, C Huitema, R Mahy. STUN- Simple Traversal of User Datagram Protocol (UDP) Through Network Address Translators (NATs) [S]. RFC3489, IETF, March 2003.
- [6] J Rosenberg, J Weinberger, H Schulzrinne. SIP Extensions for NAT Traversal [S]. Internet Draft, IETF, November 2001.
- [7] J Rosenberg, et al. SIP extensions for presence [S]. Internet Draft, Internet Engineering Task Force, June 2000. Work in progress.
- [8] K Darilion, W kampichler, K m Goeschka. Event based radio communication signaling using the session initiation protocol [A]. In networks, 2003, ICON 2003. 11TH IEEE International Conference on [C]. IEEE, 2003. 489-494
- [9] M Handley, H Schulzrinne, E Schooler, J Rosenberg. SIP: Session Initiation Protocol [S]. RFC3261, IETF, June 2002.
- [10] M Handley, V Jacobson. SDP: Session Description Protocol [S]. RFC2327, IETF, April 1998.
- [11] M S Branicky, S M Phillips, W Zhang. Stability of networked control system: Explicit analysis of delay [A]. Proceedings of American Control Conference [C]. Chicago, Illinois, 2000. 314 - 321.
- [12] R Ramjee, T F LaPorta, L Salgarelli, S Thuel, K Varadhan, L Li. IP-based access network infrastructure for next generation wireless networks [J]. IEEE Personal Communications Magazine, 2000, 7(4): 34-41.
- [13] Y Rekhter, B Moskowitz, D Karrenberg, E Lear. Address Allocation for Private Internets [S]. RFC1918, IETF, February 1996.

作者简介:

黄永峰 男, 生于 1967 年 12 月, 计算机系统结构专业博士, 副教授, 研究方向为计算机网络、多媒体通信与网络安全。
E-mail: yfhuang@Tsinghua.edu.cn

侯冠男 女, 生于 1984 年 9 月, 2005 年毕业于清华大学电子工程系, 目前清华大学电子工程系在读硕士, 专业: 信息与通信. 研究方向: 多媒体通信技术与网络安全.

李建庆 男, 生于 1969 年 6 月, 电子科学与技术专业博士, 助理教授, 主要从事计算机网络研究.