华中科技大学

硕士学位论文

基于Linux的SIP穿越NAT研究与实现

姓名：张连靖

申请学位级别：硕士

专业：通信与信息系统

指导教师：杨宗凯;杜旭

20040509

摘 要

VoIP(Voice over IP)是一种以 IP 电话为主，并推出相应的增值业务的技术。VoIP 最大的优势是能广泛地采用 Internet 和全球 IP 互连的环境，提供比传统业务更多、更好的服务。它也是目前 Internet 应用领域的一个热门话题。同时，由于 SIP 协议与 H.323 协议相比具有更大的优势，所以 VoIP 系统越来越多的采用 SIP 协议。本文对基于 SIP 协议的 VoIP 系统中关键技术进行了研究，无疑具有重要的实用意义。

VoIP 的大规模应用将极大地增加对 IP 地址的需求。当前 VoIP 重点考虑使用 NAT 穿越技术来解决地址匮乏的问题。与支持 HTTP 等数据穿越的传统 NAT 防火墙不同的是，基于 H.323、SIP 等协议的 VoIP 应用需通过信令消息中的 IP 地址和端口来实现目的地寻址，因此 NAT 穿越时不仅需要对 TCP/UDP 层的端口信息以及 IP 层的源地址和目的地址进行变换，还需对 IP 包载荷中的相关地址信息进行变换。因此 NAT 穿越问题是目前开展 VoIP 业务最大的障碍，迫切需要解决。

本文首先分析了基于 SIP 的 VoIP 系统中相关协议，介绍了防火墙与 NAT 以及 Linux 下防火墙和 NAT 的实现方式，并介绍了目前业界 NAT 穿越的几种解决方案。本文着重给出了一种 Linux 平台上应用层网关（ALG）解决 NAT 穿越的具体实现方案。实现方案归纳了 SIP 穿越 NAT 时所需修改的字段以及修改方法，并提出了一个 ALG 状态机，该状态机不仅能够使 SIP 信令顺利穿越 NAT 还可以记录当前 SIP 通话状态并控制 RTP 媒体信道。本文遵循软件工程的要求，对方案进行了功能测试、性能测试等。研究和实践表明，本文所提出的 ALG 的方案能够有效的解决典型的 SIP 穿越 NAT 问题，为 VoIP 的大规模应用提供了参考。

关键词：VoIP　防火墙　NAT 穿越 NAT 应用层网关

Abstract

VoIP is a technology that mostly utilized in IP phone, and relevant value-added services. The most advantage of VoIP is that it can make use of the global IP internetworking environment, to provide the more and more、the better and better services than tradition PSTN network. It is also the hot research field of Internet. At the same time, as SIP has more advantages than H.323, VoIP trends to adopt SIP protocol instead of H.323. This thesis does some research on the VoIP system based on SIP.

The wide application of VoIP greatly increases the demand of IP addresses. VoIP mainly considers using NAT traversal to solve this problem. Compared to general data traversal such as HTTP, the VoIP application based on H.323 and SIP uses the IP address and port in the message body of singnaling to route. So besides transforming the TCP/UDP layer port and IP layer address, VoIP NAT traversal needs to transform parts of the payload of IP packages. NAT traversal is now the biggest drawback in VoIP application and should be solved in short time.

This thesis first analyses the related protocols in VoIP system based on SIP, introduces the firewall and NAT including the NAT and firewall implementation in Linux, and discusses some kinds of method to solve this problem. The thesis mainly implements a prototype of an Application Level Gateway (ALG) for SIP traversal. The implementation summaries the modifications to SIP messages and puts forward an ALG state machine. This state machine not only passes SIP signaling through NAT gateway but also records the SIP communication states and controls the RTP channel. Following the request of software engineering, this thesis takes a function testing and a performance testing on ALG; the testing result shows that the ALG has fulfilled the requirement of SIP traversal NAT. It will be a good help to wide usage of VoIP applications.

Key words: VoIP SIP firewall NAT traversal ALG

独创性声明

本人声明所呈交的学位论文是我个人在导师指导下进行的研究工作及取得的研究成果。尽我所知，除文中已经标明引用的内容外，本论文不包含任何其他个人或集体已经发表或撰写过的研究成果。对本文的研究做出贡献的个人和集体，均已在文中以明确方式标明。本人完全意识到本声明的法律结果由本人承担。

学位论文作者签名：张莲靖

日期：2004年4月20日

学位论文版权使用授权书

本学位论文作者完全了解学校有关保留、使用学位论文的规定，即：学校有权保留并向国家有关部门或机构送交论文的复印件和电子版，允许论文被查阅和借阅。本人授权华中科技大学可以将本学位论文的全部或部分内容编入有关数据库进行检索，可以采用影印、缩印或扫描等复制手段保存和汇编本学位论文。

保密□，　　在____年解密后适用本授权书。

本论文属于

不保密☑。

（请在以上方框内打“√”）

学位论文作者签名：张莲靖　　指导教师签名：

日期：2004年5月9日　　日期：2004年5月9日

1 绪 论

1.1 课题背景

在当前的网络应用中，VoIP 是一大热点，具有很大的发展潜力和广阔的市场前景。伴随宽带入户，信息服务的多样化需求与多功能智能设备等高新技术产品的不断涌现推动了网络的融合，传统的电路交换网也有逐渐地被分组交换网所替代的趋势。目前应用于 VoIP 的主要标准有 ITU 的 H. 323[1] 建议和 IETF 的 SIP[2]协议。SIP 因其简单、灵活、可扩展性强的特点，成为实现 VoIP 系统的热点技术。进入二十世纪九十年代以来，因特网规模飞速膨胀，IPv4 地址资源面临着耗尽的危机，同时在世界的各个角落，存在着大量的局域网资源，也就是说大量的用户在使用着私有 IP 地址，由于相关标准的不完善，IPv6 技术的广泛应用还有待时日，对于如何解决局域网用户接入和访问 INTERNET 资源，已经有很多成熟的标准和具体的应用，NAT[3] 就是其中的一种技术。NAT 技术目前主要应用于 IP 、UDP 和 TCP 层，而 SIP 协议是基于 UDP 和 TCP 之上的应用层协议，SIP 包头中含有很多对于路由、接续 SIP 信令和建立呼叫连接的必不可少的地址信息，SIP 如何穿越 NAT 目前还没有相关标准，业界存在许多解决方案。

本文是与台湾 Adigital 公司合作项目“VoIP 网关路由器”的一个子项目。该网关路由器能够为局域网用户分配虚拟 IP，同时它能够与传统电信网互通。为了增加产品的功能与卖点，网关需要局域网内网用户也能享用 VoIP 功能，能够与其它网关的内网用户进行通信。

本文所要解决问题即为：网关内网用户的 SIP 信令能够穿越 NAT 与外部 SIP 用户或者其它网关的内网用户进行通信。

1.2 课题关键技术以及研究现状

1.2.1 VoIP 系统

随着 Internet 的快速发展，VoIP 的应用迅速普及，通过 IP 网络进行语音通讯已经十分普遍。VoIP 是建立在 IP 技术上的分组化、数字化传输技术，其基本原理[4]是：通过语音压缩算法对话音进行压缩编码处理，然后把这些语音数据按 IP 等相关协议进行打包，经过 IP 网络把数据包传输到目的地，再把这些语音数据包串起来，经过解码解压处理后，恢复成原来的语音信号，从而达到由 IP 网络传送话音的目的。

当前 Internet 的应用也日益广泛，特别是目前骨干网速率在高速增长，接入网的速率不断增长，因而 Internet 上的业务正在从窄带走向宽带，从非实时走向实时，VoIP 则为其中的一个重要的业务发展方向。

运用 VoIP 技术，通过技术整合，将传统企业 PSTN 语音业务与传统 LAN 数据业务合二为一，使之能够在一个网络上实现低成本的 IP 语音和 IP 数据服务，这对于增强企业局域网网络非 IP 呼叫处理能力、扩展其使用功能、降低企业对外经营业务成本费用（如大量长途电话、传真、视频会议、语音多媒体网络信息服务等）具有非常重要的现实意义。

目前 VoIP 正在逐步采纳 SIP 协议，SIP 协议由 IETF 工作组定义，基于文本编码的构建在 UDP 和 TCP 之上的，用于建立呼叫连接的应用层控制信令协议。SIP 消息中使用的 SDP 协议用于 SIP 端点之间媒体通道建立所需参数的描述。基于 SIP 的系统协议栈结构如图所示：

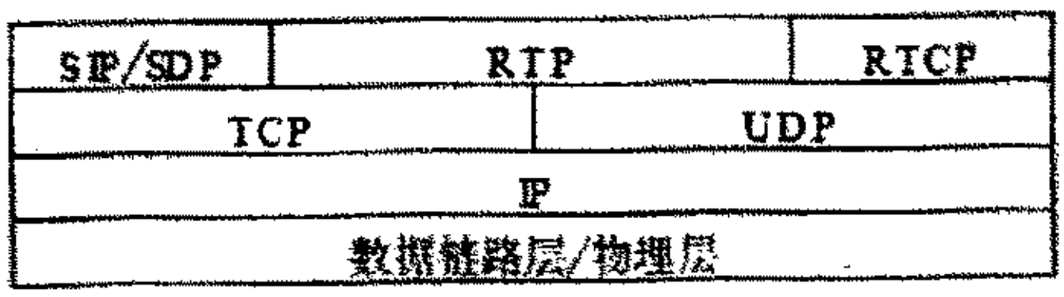

图 1-1 SIP 系统协议栈结构图

1.2.2 NAT 防火墙以及对 VoIP 的影响

通常，企业为使内部的计算机不受外部网络的攻击影响，都部署了防火墙。一个

防火墙（作为阻塞点、控制点）能极大地提高一个内部网络的安全性，并通过过滤不安全的服务而降低风险。由于只有经过精心选择的应用协议才能通过防火墙，所以网络环境变得更安全。如防火墙可以禁止诸如众所周知的不安全的 NFS 协议进出受保护网络，这样外部的攻击者就不可能利用这些脆弱的协议来攻击内部网络。防火墙同时可以保护网络免受基于路由的攻击，如 IP 选项中的源路由攻击和 ICMP 重定向中的重定向路径。防火墙应该可以拒绝所有以上类型攻击的报文并通知防火墙管理员。

而基于 IP 的语音和视频通讯的 SIP 协议，要求终端之间使用 IP 地址和数据端口来建立数据通信通道。为了建立数据连接，终端必须随时侦听外来的呼叫，而防火墙却通常被配置来阻止任何不请自到的数据包通过。VoIP 在通信时，即使防火墙可以让最初建立呼叫的发往固定端口的数据包进入，由于语音和视频通信需要通过动态端口分配来建立发送和接收数据的通道，其范围较大且无法事先预知内部终端的 IP 地址和端口信息，因此防火墙不可能无限制地开放这么大的包过滤范围，否则就失去了防火墙存在的意义。

随着 Internet 的快速膨胀，IPv4 地址空间即将严重耗尽。为解决这个问题，人们设计出了网址地址转换（NAT）[2] 。NAT 置于内网与外网的边界，其功能是将外网可见的 IP 地址与内网所用的地址相映射，这样，每一受保护的内网可重用特定范围的虚拟 IP 地址，而这些地址是不用于公网的。从外网来的含公网地址信息的数据包先到达 NAT，NAT 使用预设好的规则（其组元包含源地址、源端口、目的地址、目的端口、协议）来修改数据包，然后再转发给内网接收点。对于流出内网的数据包也须经过这样的转换处理。从安全性上来看 NAT 提供了对外隐藏内网拓扑的一个手段，但也给 VoIP 应用带来很大的麻烦。SIP、H.323 等语音视频协议消息包一般是在特定区段中内嵌 IP 地址和端口号，而不是放置在 IP 包头，这样如果仅仅使用 NAT，协议里的 IP 和端口号就不能指向正确的地方。

1.2.3 研究现状

目前业界对 SIP 穿越 NAT 有很多解决方案，比如应用层网关（Application Layer Gateway，NAT/ALG）方式，UDP 对 NAT 的简单穿越（Simple Traversal of UDP Through Network Address Translator，STUN）方式以及 SIP 协议扩展方式。各种解决方案互有

利弊，业界并没有统一标准。但由于本文所在项目的特殊性，本文结合 SIP 协议着重介绍 NAT/ALG 方式，并给出了一种较好的实现方案。

1.3 本文主要工作以及组织结构

以 SIP 为基础的 VoIP 系统牵涉到许多协议比如 SIP，SDP 和 RTP，本文深入分析了各个协议的运作过程。基于 SIP 协议存在多种通信状态这一特点，本文提出的 ALG 方案记录了 SIP 协议通信状态，在不同的状态对信令消息进行不同的修改。并根据通信状态打开和关闭 RTP 语音通道。同时本文研究了 Linux 下 NAT 防火墙的实现，结合这种方式给出了 ALG 在 Linux 平台上的实现方案。

本文第一章介绍了 VoIP 系统、防火墙和 NAT 的概念，提出了项目研究背景和意义。

第二章介绍了 VoIP 系统中所使用到的协议。

第三章介绍了防火墙与 NAT 以及它们在 Linux 中的实现。

第四章对现有的 SIP 穿越 NAT 技术做了研究，介绍了现在流行的解决方案。

第五章描述了 Linux 系统上应用层网关的实现方案，总结了字段修改表，提出了一个用于 SIP 穿越 NAT 的状态机。

第六章探讨了课题实施情况及下一步研究的方向。

本文工作对 SIP 穿越 NAT 做了有益的探索。

2 基于 SIP 信令的 VoIP 系统

2.1 SIP 协议介绍

2.1.1 SIP 背景和功能

随着传统电信网络和计算机网络的逐渐融合，出现了采用分组交换技术替代传统电路交换技术传送话音业务的 VoIP，VoIP 的出现不可避免的给传统电信市场带来了强大冲击，大多长途电信运营商都开始关注和采用 IP 电话，目前国际上的 VoIP 主要有 ITU-T 的 H.323 和 SIP。目前前者使用较多，但由于 SIP 与 H.323 相比有许多优势，所以，对 SIP 的研究已经越来越深入、充分，SIP 的使用也越来越广泛，大有取代 H.323 之势。本文着重介绍 SIP。

会话发起协议（Session Initiation Protocol，SIP）[2]是由 Internet 工程任务组（IETF）提出的 IP 电话信令协议。正如其名字所隐含的，SIP 信令用于发起会话，它能控制多个参与者参加的多媒体会话的建立和终结，并能动态调整和修改会话属性，如会话带宽要求、传输的媒体类型（语音、视频和数据等）、媒体的编解码格式、对组播和单播的支持等。

SIP 的开发目的是用来帮助提供跨越因特网的高级电话业务。IP 电话（因特网电话）正在向一种正式的商业电话模式演进，SIP 协议正是用来确保这种演进实现而产生的，它是 VoIP 系列协议中重要的一员。

2.1.2 SIP 网络元素

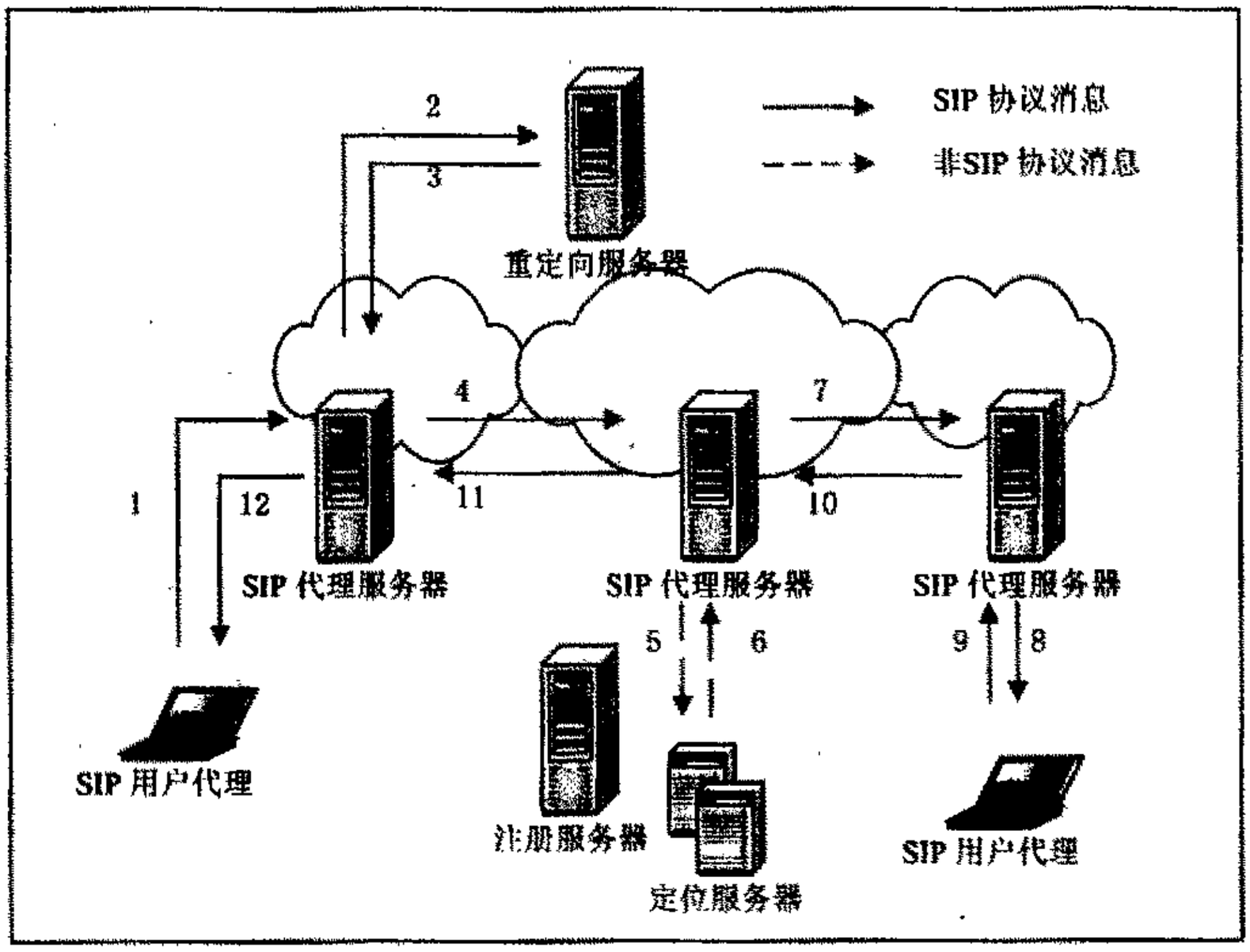

图 2-1 SIP 网络元素

用户代理本身具有一客户机元素（用户代理客户机 UAC）和一服务器元素（用户代理服务器 UAS）。客户机元素初始呼叫而服务器元素应答呼叫。这允许点到点的呼叫通过客户机-服务器协议来完成[5]。

SIP 服务器元素提供多种类型的服务器。有三种服务器形式存在于网络中--SIP 有状态代理服务器，SIP 无状态代理服务器和 SIP 重定向服务器。由于呼叫者未必知道被呼叫方的 IP 地址或主机名，SIP 服务器的主要功能是提供名字解析和用户定位。可以获得的是 email 形式的地址或与被呼叫方关联的电话号码。使用该信息，呼叫者的用户代理能够确定特定服务器来解析地址信息--这可能涉及网络中很多服务器。

SIP 代理服务器接收请求，决定将这些请求传送到何处，并且将它们传送到下一服务器（使用下一跳路由原理）。在网络中可以有多跳。

有状态和无状态代理服务器的区别是有状态代理服务器记住它接收的入请求，以及回送的响应和它转送的出请求。无状态代理服务器一旦转送请求后就忘记所有的信

息。这允许有状态代理服务器生成请求以并行地尝试多个可能的用户位置并且送回最好的响应。无状态代理服务器可能是最快的，并且是 SIP 结构的骨干。有状态代理服务器可能是离用户代理最近的本地设备，它控制用户域并且是应用服务的主要平台。

重定向服务器接收请求，但不是将这些请求传递给下一服务器而是向呼叫者发送响应以指示被呼叫用户的地址。这使得呼叫者可以直接联系在下一服务器上被呼叫方的地址。

2.1.3 SIP 协议实现机制

SIP 是一个分层结构的协议[6]，这意味着它的行为根据一组平等独立的处理阶段来描述，每一阶段之间只是松耦合。协议分层描述是为了表达，从而允许功能的描述可在一个部分跨越几个元素。它不指定任何方式的实现。当我们说某元素包含某层，我们是指它顺从该层定义的规则集，不是协议规定的每个元素都包含各层。而且，由 SIP 规定的元素是逻辑元素，不是物理元素。一个物理实现可以选择作为不同的逻辑元素，甚至可能在一个个事务的基础上。

SIP 的最底层是语法和编码。它的编码使用增强 Backus-Nayr 形式语法（BNF）来规定。

第二层是传输层。它定义了网络上一个客户机如何发送请求和接收响应以及一个服务器如何接收请求和发送响应。所有的 SIP 元素包含传输层。

第三层是事务层。事务是 SIP 的基本元素。一个事务是由客户机事务发送给服务器事务的请求（使用传输层），以及对应该请求的从服务器事务发送回客户机的所有响应组成。事务层处理应用层重传，匹配响应到请求，以及应用层超时。任何用户代理客户机（UAC）完成的任务使用一组事务产生。用户代理包含一个事务层，有状态的代理也包含事务层。无状态的代理不包含事务层。事务层具有客户机组成部分（称为客户机事务）和服务器组成部分（称为服务器事务），每个代表有限的状态机，它被构造来处理特定的请求。

事务层之上的层称为事务用户（TU）。每个 SIP 实体，除了无状态代理，都是事务用户。当一个 TU 希望发送请求，它生成一个客户机事务实例并且向它传递请求和 IP 地址、端口和用来发送请求的传输机制。一个 TU 生成客户机事务也能够删除它。

当客户机取消一个事务时，它请求服务器停止进一步的处理，将状态恢复到事务初始化之前，并且生成特定的错误响应到该事务。这由 CANCEL 请求完成，它构成自己的事务，但涉及要取消的事务。

SIP 通过 EMAIL 形式的地址来标明用户地址。每一用户通过一等级化的 URL 来标识，它通过诸如用户电话号码或主机名等元素来构造（例如：SIP:user@company.com）。因为它与 EMAIL 地址的相似性，SIP URLs 容易与用户的 EMAIL 地址关联。

SIP 提供它自己的可靠性机制从而独立于分组层，并且只需不可靠的数据包服务即可。SIP 可典型地用于 UDP 或 TCP 之上。

当一用户希望呼叫另一用户，呼叫者用 INVITE 请求初始呼叫，请求包含足够的信息用以被呼叫方参与会话。如果客户机知道另一方的位置，它能够直接将请求发送到另一方的 IP 地址。如果不知道，客户机将请求发送到本地配置的 SIP 网络服务器。如果服务器是代理服务器它将解析被呼叫用户的位置并且将请求发送给它们。有很多方法可完成上步，例如搜索 DNS 或访问数据库。服务器也可以是重定向服务器，它可以返回被呼叫用户的位置到呼叫客户机用以它直接与用户联系。在定位用户的过程中，SIP 网络服务器当然能够代理或重定向呼叫到其它的服务器，直到到达一个明确地知道被呼叫用户 IP 地址的服务器。

一旦发现用户地址，请求就发送给该用户，此时将产生几种选择。在最简单的情况，用户电话客户机接收请求——也就是，用户的电话振铃。如果用户接受呼叫，客户机用客户机软件的指定能力响应请求并且建立连接。如果用户拒绝呼叫，会话将被重定向到语音邮箱服务器或另一用户。“指定能力”参照用户想启用的功能。例如，客户机软件可以支持视频会议，但用户只想使用音频会议，那则只会启用音频功能。

2.1.4 SIP 消息的组成

有两种类型的 SIP 消息：

- 请求：从客户机发到服务器
- 响应：从服务器发到客户机

SIP 请求消息包含三个元素：请求行、头、消息体。

SIP 响应消息包含三个元素：状态行、头、消息体。

请求行和头域根据业务、地址和协议特征定义了呼叫的本质，消息体独立于 SIP 协议并且可包含任何内容。

每条 SIP 消息由以下三部分组成：

- 起始行（Start Line）：每个 SIP 消息由起始行开始。起始行传达消息类型（在请求中是方法类型，在响应中是响应代码）与协议版本。起始行可以是一请求行（请求）或状态行（响应）。
- SIP 头：用来传递消息属性和修改消息意义。它们在语法和语义上与 HTTP 头域相同（实际上有些头就是借自 HTTP），并且总是保持格式：<名字>: <值>。
- 消息体：用于描述被初始的会话（例如，在多媒体会话中包括音频和视频编码类型，采样率等）。消息体能够显示在请求与响应中。SIP 清晰区别了在 SIP 起始行和头中传递的信令信息与在 SIP 范围之外的会话描述信息。可能的体类型就包括本文将要描述的 SDP 会话描述协议。

其中描述消息体（Message body）的头称为实体头（Entity header），其格式如下：

Generic-message = start-line
*message-header
CRLF
[message -body]

起始行分请求行（Request-line）和状态行（Status-line）两种，其中请求行是请求消息的起始行，状态行是响应消息的起始行。

SIP 定义了下述方法：

INVITE——邀请用户加入呼叫。

BYE——终止一呼叫上的两个用户之间的呼叫。

OPTIONS——请求关于服务器能力的信息。

ACK——确认客户机已经接收到对 INVITE 的最终响应。

REGISTER——提供地址解析的映射，让服务器知道其它用户的位置。

INFO——用于会话中信令。

2.1.5 SIP 消息头域简介

消息头域分成四类：general-header, request-header, response-header 和 entity-header。每个头域都是一个“句子”，即以 CRLF 行结束符标志一个头域的结束。它们都由域名 field-name 和域值 field-value 两部分组成，中间以“:”冒号相隔。一般对域值 field-value 的规定和解释与具体各个域名 field-name 有关。

各个头域的详细说明请参阅 SIP 协议第六章，这里对一些最常用最基本的头域作简要介绍。

Call-ID：用来唯一标志“INVITE”请求或者相同客户机发出的所有登记请求。如果会议通过“INVITE”多个人参加，就有多个不同的 Call-ID。

Request-URL：用来只是请求消息的寻址，它的值可以被代理服务器修改。

From：只是请求的发起者，From 头域可以包含 Tag 参数，Tag 用来区分用户使用相同 Call-ID 发起呼叫的情况。

To：只是请求消息的接受者。语法形式与 From 完全相同。当请求消息中包含多个 Via 头域时，响应消息的 To 必须增加 Tag 参数，以便主叫的 UAC 可以相同请求多个响应消息。

CSeq：由请求方法和十进制的数字组成。相同 Call-ID 发出的请求必须使用单调增长的数字。ACK 和 CANCEL 请求使用和初始 INVITE 请求相同的 Cseq 值。

Contact：给用户提供 URL 进行通信。Contact 可以出现在 INVITE，ACK 和 REGISTER 请求中，也可以使用在 1XX，2XX 等响应消息中。

Via：跟踪请求消息目前走过的路径。Via 头域的使用有助于防止请求循环并可保证响应和请求采用相同的路由。Via 基本元素是主机和使用的端口。

下图所示的场景为参与会话的两个 User agent，从发起呼叫、建立链路、媒体流传输、到拆除 SIP 信令链路的全过程[71]。

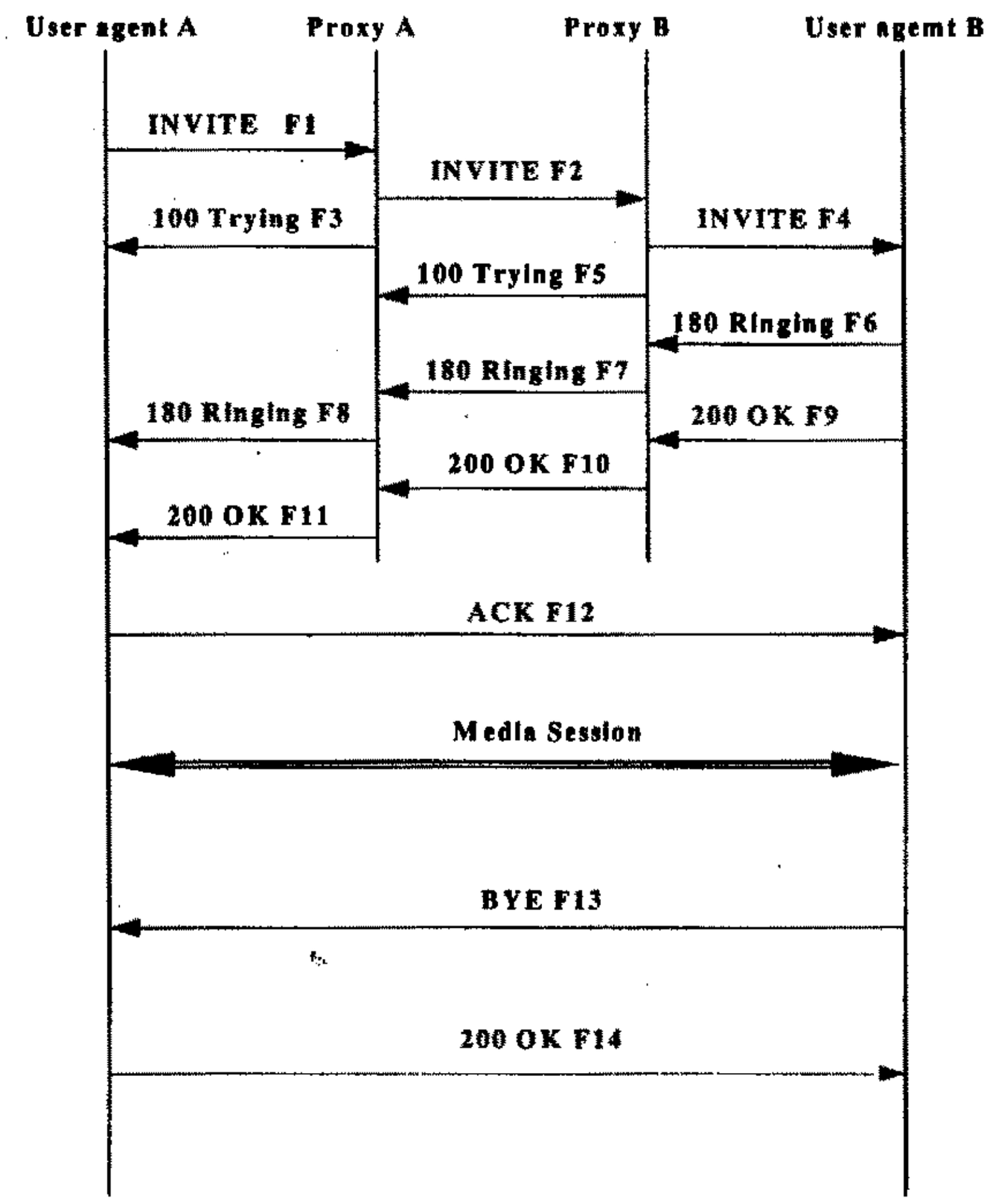

图 2-2 SIP 信令过程

2.2 SDP 协议

目前 SIP 多数使用 SDP 进行媒体协商，实际上 SDP[8]就是用来描述多媒体会话通告，多媒体会话邀请和其他形式的多媒体会话初始化的协议。SDP 包通常包括以下信息：

- 媒体类型，例如视频和音频。
- 传输协议，例如 RTP/UDP/IP 和 H.320。
- 媒体格式，例如 H.261 视频和 MPEG 视频。

- 多播地址和媒体传输端口（IP 多播会话）。
- 用于联系地址的媒体和传输端口的远端地址（IP 单播会话）。

SDP 描述由许多文本行组成，文本行的格式为<类型>=<值>，<类型>是一个字母，<值>是结构化的文本串，其格式依<类型>而定。

下面是 SDP 的一个例子：

```
---------- SDP Message ----------------------
v=0
o=lj 53655765 2353687637 IN IP4 10.1.1.1
s=Session SDP
c=IN IP4 10.1.1.1
m=audio 49172 RTP/AVP 0
```

2.3 RTP 协议

RTP 协议的含义为即时传输协议，它的英文名字为 Real-time Transport Protocol，是用于 Internet 上针对多媒体数据流的一种传输协议[9]。该协议被定义为在一对一或一对多的传输情况下工作，其目的是提供时间信息和实现流同步。RTP 协议通常使用 UDP 来传送数据，但 RTP 协议也可以在 TCP 或 ATM 等其他协议之上工作。RTP 本身并不能为按顺序传送数据包提供可靠的传送机制，也不提供流量控制或拥塞控制，它依靠 RTCP 提供这些服务。

在基于 SIP 的 VoIP 系统中，对于实时媒体的传输多采用 RTP 协议进行传输。而对 RTP 传输特性的描述则是在 SDP 协议中协商的。如前面例子所描述，SDP 消息主体中的 c 字段为接收 RTP 包的 IP 地址而 m 字段则为接收 RTP 包的端口号。

每个RTP 数据包都由一个头部和不定长的媒体数据组成,其中,RTP 包头的前12个字节是固定的。RTP包头结构如下图所示。

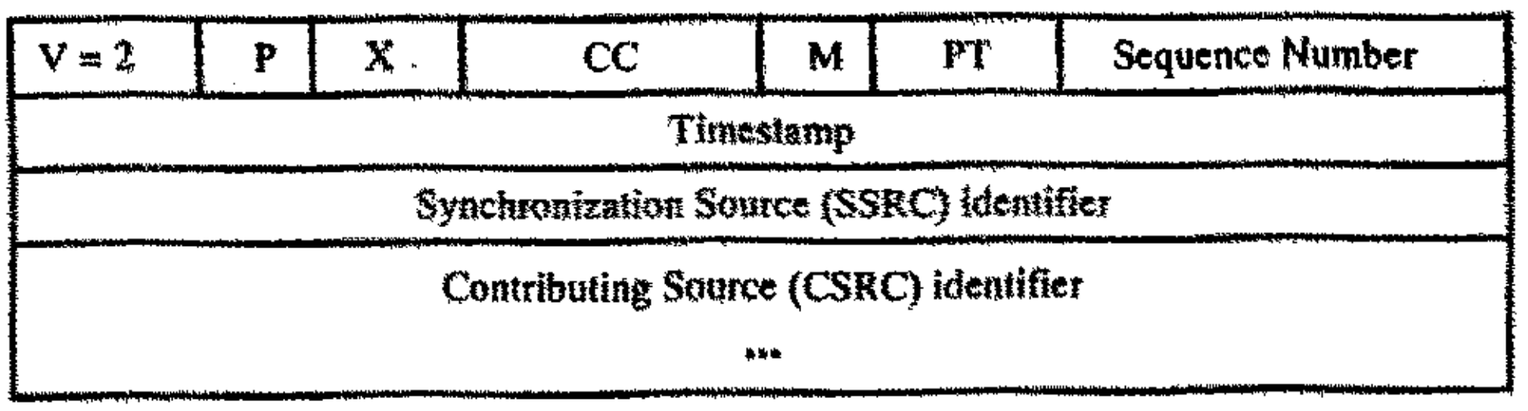

图 2-3 RTP 包头的格式

前面 12 个字节是所有 RTP 包中都必须包含有的内容。CSRC 标识符列表域只在混合器（Mixer）的输出中才产生,用于多点会议。其中几个重要的域的意义及实现如下：

- CSRC 计数器（CC）：4 比特，这个域含有固定头部后面跟着的 CSRC（Contributing Source）的数目。
- 净荷类型（PT）：7 比特，该域标识了 RTP 净荷的格式，它决定了应用程序如何对净荷解码。RFC1890 定义了一个缺省的净荷类型码到净荷格式的静态映射集合，例如 G723 为 4，G729 为 18。需要注意的是：在任何一个给定的时刻，RTP 包的发送者只能发送一种净荷类型的包。
- 序列号（Sequence Number）：16 比特，发送方在每发送完一个 RTP 包后就将该域的值增加一，接收方可以由该域检测包的丢失及恢复包序列。序列号的初始值是随机的，使得即便在发送端本身不加密时（如数据包通过翻译器时），对采用公开加密算法的普通文本进行攻击也会很困难。这个初始值的选择，与后面对时间戳初始值和 SSRC 的选取相同。
- 时间戳（Tmestamp）：32 比特，该域记录了该包中数据的第一个字节的采样时刻。在媒体同步和抖动计算中是不可缺少的。用于采样的时钟分辨率必须达到足够的精度，以便进行同步和测量接收包的抖动。例如，以每个视频帧间隔作为一个采样时钟单位明显精度不够。时钟频率由净荷所承载的数据的格式和媒体的类型决定，它由配置或净荷格式规范明确定义或根据净荷格式通过非 RTP 方式动态规定。这里的时间戳也像序列号一样有个随机初始值的问题，对于这个初始值，请参阅

SSRC 的产生。对于时间戳，要注意两点：第一，几个连续的 RTP 包可以拥有相同的时间戳（如:同一帧图像分别封装到了几个 RTP 包中）。第二，连续的 RTP 包中所含的时间戳可能不是单调增加的。这是因为数据没有按照取样的先后顺序来传输所造成的，如 MPEG 的插值帧。

- 同步源（Synchronization Source,SSRC）:32 比特，该域是同步源的一个标识符。所谓同步源，就是指 RTP 包流的来源。该标识符是随机选取的，在同一个 RTP 会话中不能有两个相同的 SSRC 值。对于如何选取一个 SSRC，这和选取序列号和时间戳的初始值一样，RFC1889 推荐了 MD5 随机算法。
- 贡献源列表（CSRC List）:0～15 项，每项 32 比特，它包含了所有对该 RTP 包中数据存在贡献的同步源的标识符。CSRC 标识符由混合器负责插入，用于有贡献的 SSRC 标识符。如，语音包如果在混合器进行了混合处理，混合产生新包的所有源的 SSRC 标识符都被列出来，以便接收端能正确指出交谈双方的身份。

3 防火墙与 NAT 网关

3.1 防火墙概述

随着全球信息高速公路的建设，Internet 的发展，给人类社会的科学与技术、经济与文化带来巨大的推动与冲击，同时也带来了许多的挑战[10]。资源共享和信息安全历来是一对矛盾。近年来随着 Internet 的飞速发展，计算机网络资源共享进一步加强，随之而来的安全问题日益突出。解决网络安全问题的一个有效办法是内部网和外部网之间设置防火墙。

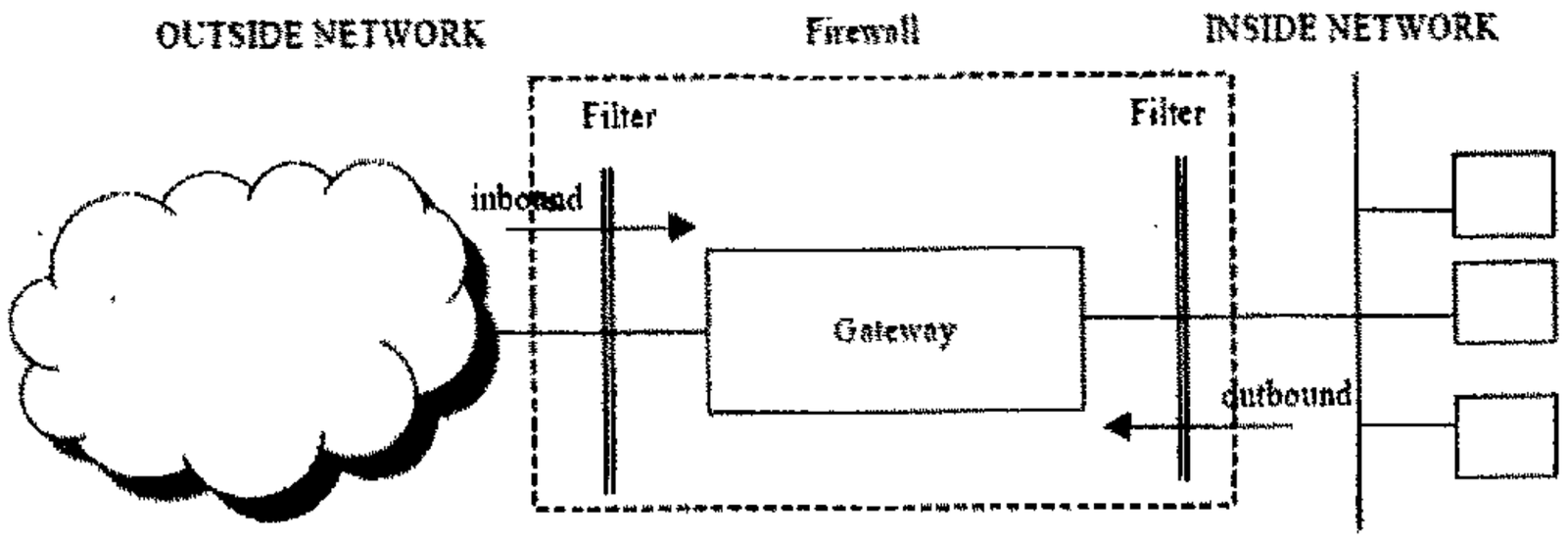

图 3-1防火墙拓扑图

如图为防火墙的一种典型应用。防火墙技术作为目前用来实现网络安全措施的一种主要手段，它可通过监测、限制、更改跨越防火墙的数据流，尽可能地对外部屏蔽网络内部的信息、结构和运行情况，以实现网络安全保护。防火墙是同时具有下述特征的安装在两个网络之间的软/硬件的集合：从里到外和从外到里的所有通信都必须通过防火墙；只有本地安全策略授权的通信才允许通过；防火墙本身是免疫的,不会被穿透的[11]。

3.1.1 防火墙技术

防火墙技术有很多种，其中有的技术已广泛应用，但更多新技术正在研究和改进之中，下面是几种常见的防火墙技术[12]

动态监视技术

它采用了一个检测模块（一个在网关上执行网络安全策略的软件引擎）。检测模块在不影响网络正常工作的前提下，采用抽取相关数据的方法对网络通信的各层实施监测，抽取部分数据（状态信息）并动态地保存起来，作为以后制定安全决策的参考。检测模块支持多种协议和应用程序，并可以很容易地实现应用和服务的扩充。

适应代理技术

自适应代理技术是根据用户定义的安全规则，动态适应传送中的分组流量。它允许用户根据具体需求，自己定义防火墙策略，而不会牺牲速度或安全性。如果安全要求高，那么最初的安全检查仍在应用层进行，保证实现传统代理防火墙的最大安全性；而一旦代理明确了会话的所有细节，那么其后的数据包就可以直接经过速度快得多的网络层。这样一来,自适应代理防火墙就拥有了和传统代理防火墙一样的安全性，同时又具有了分组状态检查防火墙的速度。

网络地址转换(NAT)

内部网络采用的是私有地址（未注册的 IP 地址），若要与外部通信，需要公网地址。网络地址转换能将未注册的 IP 地址映射成合法的公网地址。网络地址转换分静态地址和动态地址两种转换模式，静态地址转换支持一对一的永久地址映射,而动态地址转换则是根据内部用户的需求临时分配公网地址，其地址映射是暂时的。 这种技术既解决了 IP 地址不足的问题，同时又隐藏了内部网络真正的 IP，使黑客无法直接攻击内部网络，加强了内部网的安全性。

3.2 NAT 介绍

3.2.1 NAT—解决 IPv4 地址不足

随着计算机网络的普及和主机数目爆炸性的增长，传统的基于 TCP/IP 体系结构

的网络地址结构和地址分配策略已越来越不适应当前计算机网络的发展状况。TCP/IP 中地址结构（IPv4）由 32 位字节组成，分成 A、B、C、D、E 五类。由于主机数目的增长，32 位字节地址结构已经濒临耗尽的边缘，阻碍了网络的发展[13]。

另外，传统的地址分配策略是对每一个入网的主机分配一个全球唯一的 IP 地址，然而其中的某些主机很少甚至根本不与英特网中的主机通信，同时某些企业或组织在申请 IP 地址时考虑到将来的发展，往往注册大量多余的 IP 地址，造成了 IP 地址得不到充分利用。Internet 地址分配委员会（IANA）针对上述两种情况，分别提出了长期和短期解决策略。一方面采用 IPv6 协议[14]（128 位地址格式）取代 IPv4，从根本上扩大 IP 地址数目；另一方面采用私有网络地址和网络地址转换 NAT（Network Address Translation）技术来解决 IP 地址的匮乏问题。

IPv6 是新一代网络层协议，它采用 128 位地址格式，可容纳 2128 个主机，根除了主机数目的扩展性问题。IPV6 与 IPV4 不兼容，但能支持 IPV4 支持的所有上层协议和应用。IPV6 弥补了 IPV4 中的安全功能，将 IPSec 融入 IPV6 中，保证了信息传输的安全。

私有网络地址技术是利用 IPV4 中预留的一个 A 类地址、16 个 B 类地址和 255 个 C 类地址分配给企业网内不与 Internet 通信的主机，仅为与 Internet 通信的主机分配全球唯一的 IP 地址[15]。这样的地址为：

10.0.0.0 - 10.255.255.255 (10/8 prefix)

172.16.0.0 - 172.31.255.255 (172.16/12 prefix)

192.168.0.0 – 192.168.255.255 (192.168/16 prefix) （RFC1918）。

而将私有网和公共网连接起来的一种方法就是 NAT（network address translator）。同时，如防火墙一节所述，它本身也能作为防火墙提供网络安全的保护。NAT 方法的基本工作原理是在私有网和公共网的地址间建立起一定的映射关系，以保证公共网中的地址唯一性并使私有网中的地址得以复用[2]。

在 NAT 建立起的早期，人们普遍怀疑 NAT 对 IP 头的操作会对应用的透明产生影响，并且得出了它不适合长期的应用，甚至是短期应用的结论[16]。但是现在不但它得到了广泛的应用，并且还有人提出可以将它作为长期的解决方案并对 IPV6 的必要性提出了质疑。支持者认为它可以支持大部分诸如浏览，电邮，文件传输的网络应用，

而反对者认为它阻碍了网络的正常发展。不论如何，NAT 都影响了端到端链接的透明性因为传输的透明取决于IP头的一致性并且有的应用不知在IP头中包含了地址信息。应用特殊配置的应用层网关可以解决部分问题，但是仍然要接受特殊应用的挑战，当没有多个 NAT 设备时还容易解决问题，但是当有多个 NAT 设备时，或是多点和分布的应用时，情况将变得非常复杂。如果 NAT 得到应用，名称的解析和反解析将要依赖请求的发出地，而这样的结果是强调应用客户端/服务器模式而不是端到端的模式(peer-to-peer)并且服务器应该在公共地址领域中。

3.2.2 NAT 的分类

根据 NAT 在地址分配策略，可以分为三种类型：静态 NAT(Static NAT)、动态地址 NAT(Pooled NAT)、网络地址端口转换 NAPT（Port－Level NAT）[17]。

静态 NAT 是设置起来最为简单和最容易实现的一种，内部网络中的每个主机都被永久映射成外部网络中的某个合法的地址。而动态地址 NAT 则是在外部网络中定义了一系列的合法地址，采用动态分配的方法映射到内部网络。NAPT 则是把内部地址映射到外部网络的一个 IP 地址的不同端口上。根据不同的需要，三种 NAT 方案各有利弊。

动态地址 NAT 只是转换 IP 地址，它为每一个内部的 IP 地址分配一个临时的外部 IP 地址，主要应用于拨号，对于频繁的远程联接也可以采用动态 NAT。当远程用户联接上之后，动态地址 NAT 就会分配给他一个 IP 地址，用户断开时，这个 IP 地址就会被释放而留待以后使用。

网络地址端口转换 NAPT（Network Address Port Translation）是人们比较熟悉的一种转换方式。NAPT（network address and port translation）扩展了 NAT，它将传输标示符（TCP 和 IP 中的端口号，ICMP 的查询标识符）也进行转换，这样就允许私有网中的多个主机的传输标示复用到一台主机上，也就是多台主机共享一个公共地址，它也可以与基本的 NAT 结合起来，让外部地址池被更多的私有网中的主机所共享。值得注意的是 NAPT 可以和传统的 NAT 结合在一起运用。对于从私有网到公共网的包 NAPT 将转换源地址，源传输标示符和相关的字段，如像 IP、TCP、UDP、ICMP 的头，而传输标示符指的是 TCP/UDP 的端口号，ICMP 的查询号。对于从公共网到私有

网的包，则也要相应的转换目的地址，目的传输标示符和相关的字段。

3.3 Linux 下防火墙与 NAT 的实现[18]

2.4.x 版本 Linux 内核使用 netfilter/iptables 实现防火墙与 NAT。

netfilter/iptables 是与最新的 2.4.x 版本 Linux 内核集成的 IP 信息包过滤系统。该系统能够很好的控制防火墙配置和启用 NAT。

3.3.1 Linux 安全性和 netfilter/iptables

Linux 因其健壮性、可靠性、灵活性以及好象无限范围的可定制性而在 IT 业界变得非常受欢迎。Linux 具有许多内置的能力，使开发人员可以根据自己的需要定制其工具、行为和外观，而无需昂贵的第三方工具。如果 Linux 系统连接到因特网或 LAN、服务器或连接 LAN 和因特网的代理服务器，所要用到的一种内置能力就是针对网络上 Linux 系统的防火墙配置。可以在 netfilter/iptables IP 信息包过滤系统（它集成在 2.4.x 版本的 Linux 内核中）的帮助下运用这种能力。

对于 Linux 系统管理员、网络管理员以及家庭用户（他们想要根据自己特定的需求来配置防火墙、在防火墙解决方案上节省费用和对 IP 信息包过滤具有完全控制权）来说，netfilter/iptables 系统十分理想。

3.3.2 Linux 中防火墙配置概念

对于连接到网络上的 Linux 系统来说，防火墙是必不可少的防御机制，它只允许合法的网络流量进出系统，而禁止其它任何网络流量。为了确定网络流量是否合法，防火墙依靠它所包含的由网络或系统管理员预定义的一组规则。这些规则告诉防火墙某个流量是否合法以及对于来自某个源、至某个目的地或具有某种协议类型的网络流量要做些什么。术语"配置防火墙"是指添加、修改和除去这些规则。稍后，将详细讨论这些规则。

网络流量由 IP 信息包（或，简称信息包）— 以流的形式从源系统传输到目的地系统的一些小块数据 — 组成。这些信息包有头，即在每个包前面所附带的一些数据位，它们包含有关信息包的源、目的地和协议类型的信息。防火墙根据一组规则检查

这些头，以确定接受哪个信息包以及拒绝哪个信息包。我们将该过程称为信息包过滤。

3.3.3 netfilter/iptables 系统的优点

netfilter/iptables 的最大优点是它可以配置有状态的防火墙，这是 ipfwadm 和 ipchains 等以前的工具都无法提供的一种重要功能。有状态的防火墙能够指定并记住为发送或接收信息包所建立的连接的状态。防火墙可以从信息包的连接跟踪状态获得该信息。在决定新的信息包过滤时，防火墙所使用的这些状态信息可以增加其效率和速度。这里有四种有效状态，名称分别为 ESTABLISHED、INVALID、NEW 和 RELATED。

状态 ESTABLISHED 指出该信息包属于已建立的连接，该连接一直用于发送和接收信息包并且完全有效。INVALID 状态指出该信息包与任何已知的流或连接都不相关联，它可能包含错误的数据或头。状态 NEW 意味着该信息包已经或将启动新的连接，或者它与尚未用于发送和接收信息包的连接相关联。最后，RELATED 表示该信息包正在启动新连接，以及它与已建立的连接相关联。

netfilter/iptables 的另一个重要优点是，它使用户可以完全控制防火墙配置和信息包过滤。您可以定制自己的规则来满足您的特定需求，从而只允许您想要的网络流量进入系统。

另外，netfilter/iptables 是免费的，这对于那些想要节省费用的人来说十分理想，它可以代替昂贵的防火墙解决方案。

3.3.4 netfilter/iptables 系统工作方式

netfilter/iptables IP 信息包过滤系统是一种功能强大的工具，可用于添加、编辑和除去规则，这些规则是在做信息包过滤决定时，防火墙所遵循和组成的规则。这些规则存储在专用的信息包过滤表中，而这些表集成在 Linux 内核中。在信息包过滤表中，规则被分组放在我们所谓的链（chain）中[19][20]。后面会详细讨论这些规则以及如何建立这些规则并将它们分组在链中。

虽然 netfilter/iptables IP 信息包过滤系统被称为单个实体，但它实际上由两个组件 netfilter 和 iptables 组成。

netfilter 组件也称为内核空间（kernel space），是内核的一部分，由一些信息包过滤表组成，这些表包含内核用来控制信息包过滤处理的规则集。

iptables 组件是一种工具，也称为用户空间（user space），它使插入、修改和除去信息包过滤表中的规则变得容易。

通过使用用户空间的 iptables，可以构建自己的定制规则，这些规则存储在内核空间的信息包过滤表中。这些规则具有目标，它们告诉内核对来自某些源、前往某些目的地或具有某些协议类型的信息包做些什么。如果某个信息包与规则匹配，那么使用目标 ACCEPT 允许该信息包通过。还可以使用目标 DROP 或 REJECT 来阻塞并杀死信息包，也可以使用目标 REDIRECT 将信息包发送到本地 IP 的某个端口上。

根据规则所处理的信息包的类型，可以将规则分组在链中。处理入站信息包的规则被添加到 INPUT 链中。处理出站信息包的规则被添加到 OUTPUT 链中。处理正在转发的信息包的规则被添加到 FORWARD 链中。这三个链是基本信息包过滤表中内置的缺省主链。另外，还有其它许多可用的链的类型（如 PREROUTING 和 POSTROUTING），以及提供用户定义的链。每个链都可以有一个策略，它定义"缺省目标"，也就是要执行的缺省操作，当信息包与链中的任何规则都不匹配时，执行此操作。

建立规则并将链放在适当的位置之后，就可以开始进行真正的信息包过滤工作了。这时内核空间从用户空间接管工作。当信息包到达防火墙时，内核先检查信息包的头信息，尤其是信息包的目的地。我们将这个过程称为路由。

如果信息包源自外界并前往系统，而且防火墙是打开的，那么内核将它传递到内核空间信息包过滤表的 INPUT 链。如果信息包源自系统内部或系统所连接的内部网上的其它源，并且此信息包要前往另一个外部系统，那么信息包被传递到 OUTPUT 链。类似的，源自外部系统并前往外部系统的信息包被传递到 FORWARD 链。

接下来，将信息包的头信息与它所传递到的链中的每条规则进行比较，看它是否与某条规则完全匹配。如果信息包与某条规则匹配，那么内核就对该信息包执行由该规则的目标指定的操作。但是，如果信息包与这条规则不匹配，那么它将与链中的下一条规则进行比较。最后，如果信息包与链中的任何规则都不匹配，那么内核将参考

该链的策略来决定如何处理该信息包。理想的策略应该告诉内核 DROP 该信息包。如下图说明了这个过程：

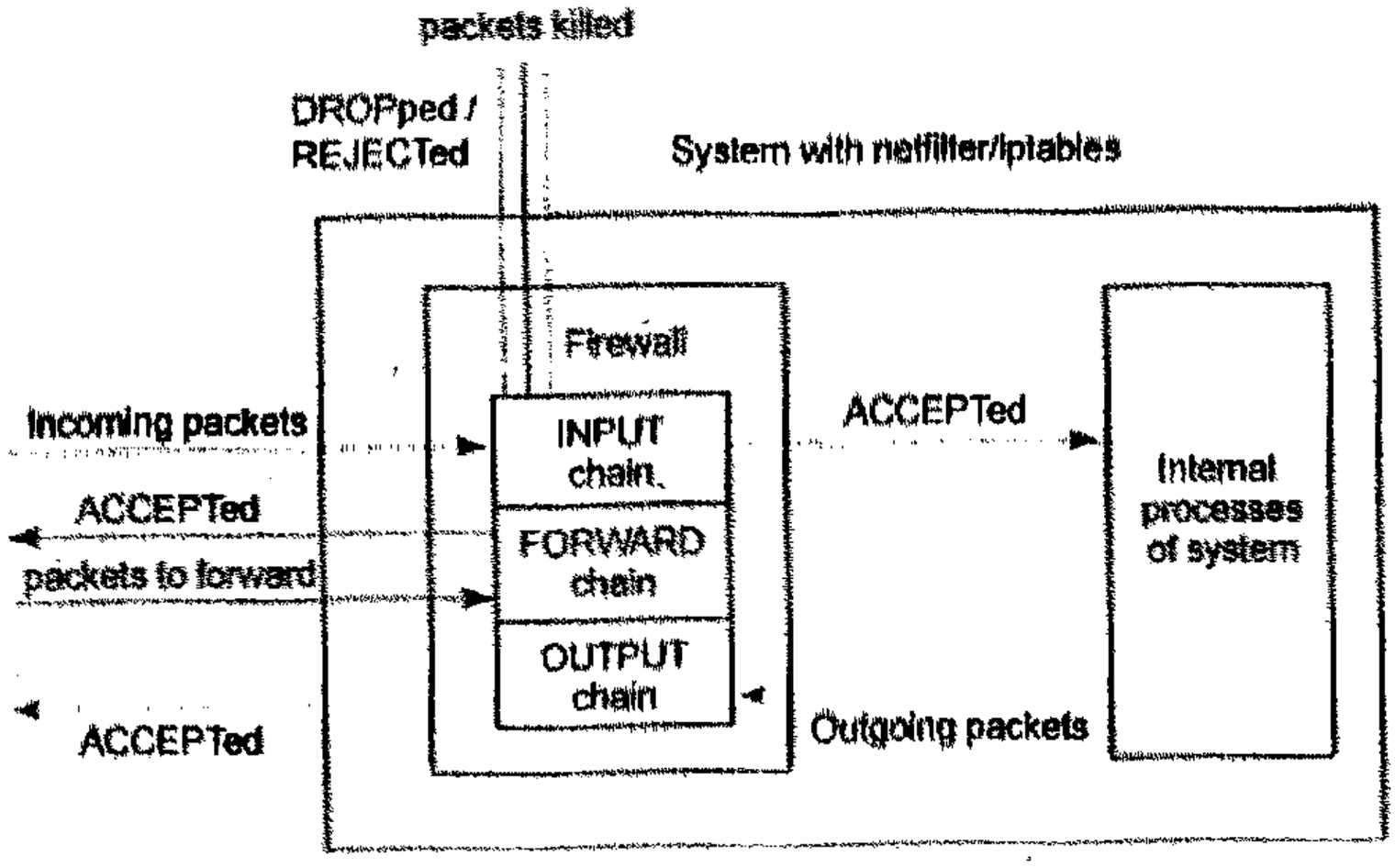

图 3-2 netfilter 实现

3.3.5 iptables 的基本使用

通过向防火墙提供有关对来自某个源、到某个目的地或具有特定协议类型的信息包要做些什么的指令，规则控制信息包的过滤。通过使用 netfilter/iptables 系统提供的特殊命令 iptables，建立这些规则，并将其添加到内核空间的特定信息包过滤表内的链中。关于添加 / 除去 / 编辑规则的命令的一般语法如下：

$ iptables [-t table] command [match] [target]

具体 iptables 的使用方法请参看 iptables 手册[21][22]。下面主要介绍它的两个应用。

3.3.6 用 iptables 实现 NAT

在 linux 的 NAT-HOWTO[23]中，作者从原理的角度将 NAT 分成了两种类型，即源 NAT(SNAT)和目的 NAT(DNAT)，顾名思义，所谓 SNAT 就是改变转发数据包的源地址，所谓 DNAT 就是改变转发数据包的目的地址。

如前面所述，netfilter 是 Linux 核心中一个通用架构，它提供了一系列的"表

"(tables)，每个表由若干"链"(chains)组成，而每条链中可以有一条或数条规则(rule)组成。并且系统缺省的表是"filter"。但是在使用 NAT 的时候，我们所使用的表不再是"filter"，而是"nat"表，所以我们必须使用"-t nat"选项来显式地指明这一点。

同 filter 表一样，nat 表也有三条缺省的"链"(chains)，这三条链也是规则的容器，具体 nat 表的操作语法可以参看 iptables 的 Howto。下面介绍利用 iptables NAT 的一种典型应用。

使用拨号带动局域网上网

小型企业、网吧等多使用拨号网络上网，通常可能使用代理，但是考虑到成本、对协议的支持等因素，建议使用 ip 欺骗方式带动区域网上网。

```
#进行 ip 伪装
iptables -t nat -A POSTROUTING -o ppp0 -j MASQUERADE
```

3.3.7 用 iptables 实现 REDIRECT

在某些应用场景下，需要将某些特殊的包截取由网关自行处理。Iptables 中的 REDIRECT 即可满足这样的需求。

对于 REDIRECT 具体描述同样可以参看 iptables 的 Howto。这里我们仅以网关要截取并处理内网 SIP 包为例。

由于 SIP 包是 UDP 包，而且的目的端口为知名端口 5060，所以我们可以设计规则将所有目的端口为 5060 的 UDP 包重定向到本地 IP 的某个端口（比如 7078）

4 SIP 穿越 NAT 研究

4.1 存在问题

目前，由于相比传统电信的诸多优势，VoIP 正以极快的速度发展，在此过程中遇到了很多实际问题，特别是 VoIP 用户的接入问题。VoIP 是一个基于分组网承载的网络，接入用户都是通过 IP 地址来寻址的，但当前网络的实际情况是，由于 IP 地址资源紧缺以及安全等原因，网上大量的企业网和驻地网基本都采用私有 IP 地址通过出口的 NAT / FW（Fire Wall，防火墙）接入公网，而目前在 IP 上承载语音和视频的协议（如 H.323、SIP、MGCP、H.248 等），由于其本身并不在 IP 头携带地址信息，在私网用户接入应用中，这些协议的控制通道 / 媒体通道难以与传统的 NAT/FW 设备与公网进行互通，或者说目前的 NAT/FW 大多支持 HTTP 的数据应用协议穿透，而不支持会话业务的控制与媒体 NAT/FW 穿透。因此上面所提到的问题就成为目前开展 VoIP 业务最大的障碍，迫切需要解决。针对本文背景以及 SIP 协议正在逐步取代 H.323 的现实，本文将讨论 SIP 穿越 NAT 的方式[24][25]。

4.2 配置方式

SIP 穿越 NAT 最简单的方式就是通过配置 SIP phone 达到穿越的效果[26]。配置方式为：

1. 通过询问网络管理员或者 NAT 网关的配置得到 NAT 网关外部 IP。同时得到 SIP 电话的本地 IP。
2. 配置 SIP 电话，使得其利用 NAT 的外部 IP 来生成 SIP，SDP 以及 RTP 包，同时它仍然侦听本地 IP 上的 SIP 消息。例如在 NAT 网关上启用 DMZ 主机该 SIP 电话能够接受所有发给 NAT 外部 IP 的包。
3. 根据 SIP 电话的 SIP 信令端口以及媒体端口配置 NAT 网关，使得网关将发向这些端口的数据包转发至内部 SIP 电话。这就意味着该 SIP 电话将收到所有外部发向

NAT 的 SIP 包和 RTP 包。

4. 外部 SIP 用户则通过拨打 NAT 外部 IP 地址与内部 SIP 电话进行通信。

这种 SIP 穿越 NAT 的方式缺点是整个 NAT 内网只能有一个 SIP 电话能够与外界通信，而且一旦 SIP 电话的 IP 地址或者端口有所变化就必须重新配置 NAT 网关，所以它扩展性很差。但是由于它利于实现所以仍然被很多小型 NAT 网关使用。

4.3 STUN in B2BUA 方式

简单 UDP 包穿越协议[27]（Simple Traversal of User Datagram Protocol）是一种 UDP 包穿越 NAT 的协议。它使用一种简单的客户—服务器模型，它通过这种模型得到 NAT 给 UDP 包分配的 IP 地址以及端口，从而修改 SIP 中相应的字段。概括的说 STUN 工作方式为：

1. STUN 首先检测自己处于何种 NAT 之下。
2. STUN 客户端在转发 UDP 包之前，发送一个 Binding 请求（使用 UDP 包封装）穿过 NAT 到达 STUN 服务器端。服务器端回送一个响应包，该响应包的记录为被 NAT 修改后 Binding 请求包的源 IP 地址以及端口。
3. 该源 IP 地址和端口即为 STUN 将要转发的 UDP 包穿越 NAT 后的源 IP 地址以及端口。STUN 客户端从而可以修改该 UDP 包内容中关于源 IP 地址和端口的部分。

STUN 也可以解决 SIP 穿越 NAT 的问题。其中一种解决方案是结合 STUN 与 SIP 的 B2BUA。B2BUA 是背靠背 SIP 用户代理（Back to back User Agent），一方面它接收 SIP 客户端用户代理的请求，另一方面它也发送 SIP 请求至 SIP 服务器端用户代理。B2BUA 接合的 STUN 客户端部分，当它接收 SIP INVITE 或者其它 SIP 请求后，作为 STUN 客户端它会去得到 NAT 转换后的地址并且修改 SDP 和 SIP 中相应内容，接着转发 SIP 请求[28]。

4.4 SIP 协议扩展方式

与 H.323 相比，SIP 的灵活性可扩展性很强，所以对 SIP 协议进行扩展，增加字段也可以实现 SIP 穿越 NAT[29][30]。这里介绍其中一种实现方式。

为了能够在 SIP 消息中体现穿越 NAT 后 IP 地址的改变。定义一种新的 Via header 参数：response port，编码为“rport”。 客户端在 Via header 中插入这个参数，但其值为空。

当 SIP 服务器端收到这样的 SIP 请求的时候会将“rport”设置为请求包的源地址端口，并且在发送响应包的时候发送至列“rport”所指端口。

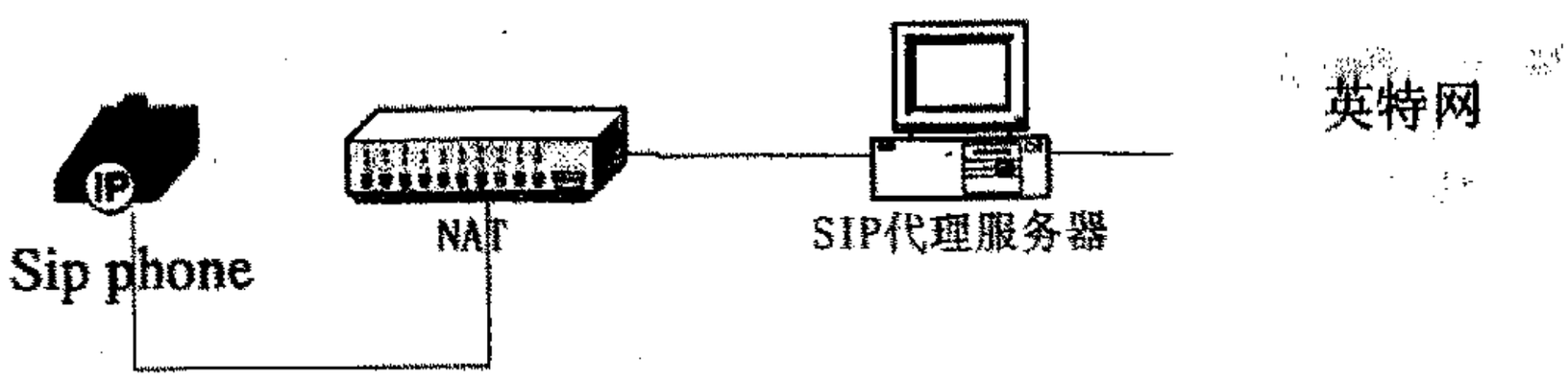

图 4-1 协议扩展拓扑图

如图，以上面拓扑图为例介绍利用协议扩展交互信令的过程。

1. SIP 电话发送一个 INVITE 消息：

```
INVITE sip:user@example.com SIP/2.0
Via:SIP/2.0/ UDP 10.1.1.1: 4540;
                    rport;branch=z9hG4bKkjshdyff
```

2. 这个消息穿过到达 SIP Proxy，SIP Proxy (192.168.139.66: 5060) 将 SIP 消息进行扩展：

```
INVITE sip:user@example.com SIP/2.0
Via: SIP / 2.0 /UDP proxy.example.com;
                    branch=z9hG4bKkjsh77
Via: SIP/2.0/UDP 10.1.1.1:4540;
                    received=192.0.2.1;rport=9988;
                    branch=z9hG4bKkjshdyff
```

Via 头中 rport 参数，表明 SIP 消息穿越 NAT 时，NAT 网关发送 SIP 消息所用端口，并且该消息是向外发出的。

3. 通过信令交互 SIP 代理服务器收到 200 OK 响应消息：

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP proxy.example.com;
                    branch=z9hG4bKkjsh77
Via: SIP/2.0/UDP 10.1.1.1:4540;
                    received=192.0.2.1;rport=9988;
                    branch=z9hG4bKkjshdyff
```

4. 按协议扩展要求，SIP 代理服务器将响应消息发送至 192.0.2.1 端口 9988。这样，NAT 能够正确的把响应消息回送至 NAT 背后的 SIP 电话。

4.5 ALG 方式研究

另外一种行之有效的解决 SIP 穿越防火墙的办法是使用应用层 NAT 网关。应用层 NAT 网关分析 IP 数据报的内容从而对 SIP 消息进行修改实现 SIP 穿越。

IETF 草案《SIP 应用层网关实现方式》[31]概述应用层 NAT 网关实现 SIP 穿越防火墙的方式。它对这种网关的实现目标进行了分析：SIP-ALG 网关要求对整个 SIP 消息进行解析，能够加密和解密，提供验证功能并且需要记录 SIP 的状态以便确定何时中止媒体流的传输。草案提出了应用层网关处理简单的 SIP 信令必须修改的字段。应用层 NAT 网关的实现正是基于这样的修改。下面简单介绍这个草案：

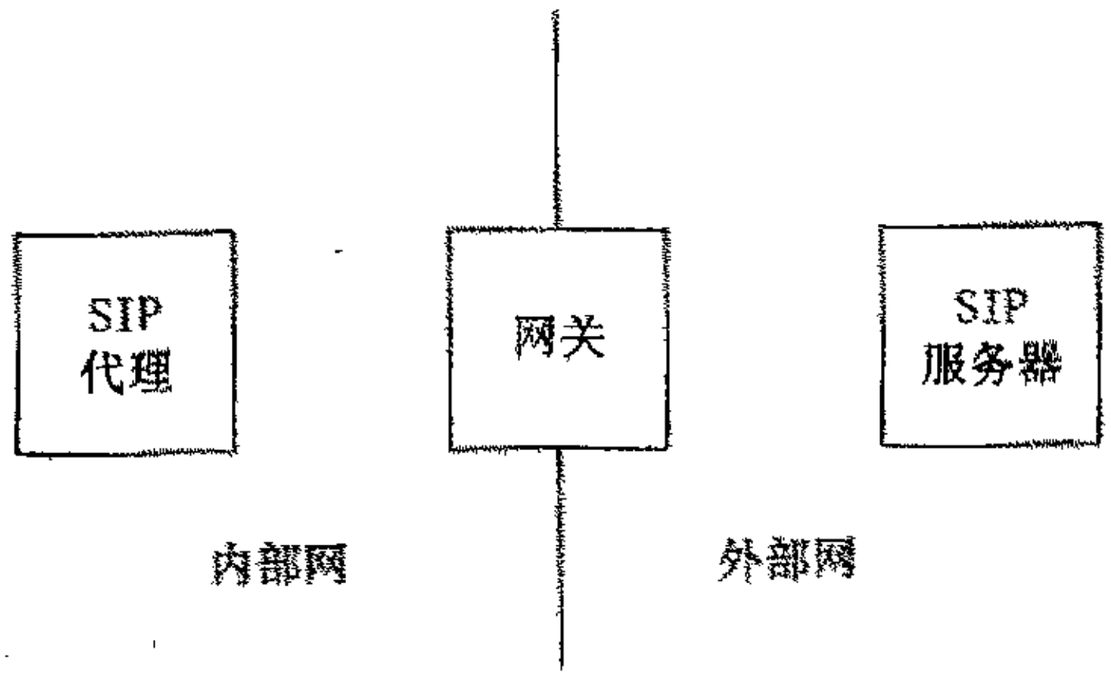

图 4-2 草案拓扑

草案考虑的是一种典型的 SIP 穿越 NAT 的情形，即为内部 SIP 代理向外部 SIP 服务器或者 SIP 终端发起一个呼叫并成功建立连接。草案要求应用层 NAT 网关必须能够修改出口和入口 SIP 包的内容。特别的，在草案所针对的特定情形中，应用层网关对于出口的 SIP 包，必须要修改的字段为：Via, Contact 和 Content-Length。如果 SIP 使用 SDP 来交换媒体信息，那么 SDP 中的内容也必须得到修改[32]。而对于入口的 SIP 包，那么需要修改的则只是 SDP 内容。为了实现这样的修改，应用层 NAT 网关还必须维护一张端口映射表。应用层 NAT 网关能够根据这张表将外部网络传入的 SIP 消息转发至所对应的 SIP 代理。

5 SIP NAT 应用网关的实现

5.1 为什么使用 ALG

5.1.1 拓扑结构

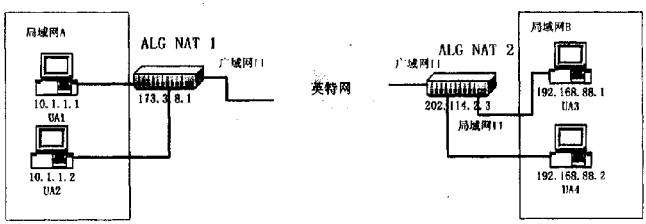

图 5-1 ALG 拓扑图

如图为方案所要解决的拓扑结构。其中 UA 为 SIP 电话。ALG NAT 使用的是 PLT-501 开发板。PLT-501 开发板为我们自行研发的产品。它拥有无线网卡、交换芯片、FXS/FXO 接口以及 4 个局域网接口和 1 个广域网接口。它运行的是 vLinux 操作系统，在内核中已经实现了 NAT。图中 NAT1 背后的 UA1 和 UA2 能够与同时与另一个 NAT 背后的 UA3 和 UA4 通信。

本拓扑有以下两个前提：

1、 所有的 SIP UA 均使用 5060 端口作为 SIP 接收端口

2、 SIP UA 不使用代理服务器，利用自己的 IP 地址作为 SIP 地址的主机部分。

这种拓扑结构能够满足大部分的使用 PLT-501 开发板的用户。因为该开发板用于小型公司或者家庭用户，而且开发板推荐使用的 SIP UA 是免费的 JAVA phone，该软件缺省配置就是使用 5060 作为 SIP 接收端口。

5.1.2 方案选择

选择使用 ALG 有如下原因

1、 配置方式虽然简单易行，但是它只能满足一个 SIP UA 穿越 NAT，不符合图中多个 SIP UA 同时穿越 NAT 的要求。

2、 STUN in B2BUA 方式需要为 SIP 穿越 NAT 搭建另外的的 STUN 服务器，而在小型应用中搭建这样的服务器花费太大并不合适。

3、 SIP 扩展方式需要修改 JAVA phone 的代码，然而我们并没有这样的代码，并且修改 SIP 存在和其它 SIP phone 兼容的问题。所以没有使用这样的方式。

由于我们拥有所有开发板代码，熟悉 Linux 操作系统，对 SIP、RTP、SDP 等的协议的实现也有一定经验。使用 ALG 对用户的开销也很小，所以我们选择使用 ALG 来实现如上拓扑。

5.2 软件架构

5.2.1 模块接口

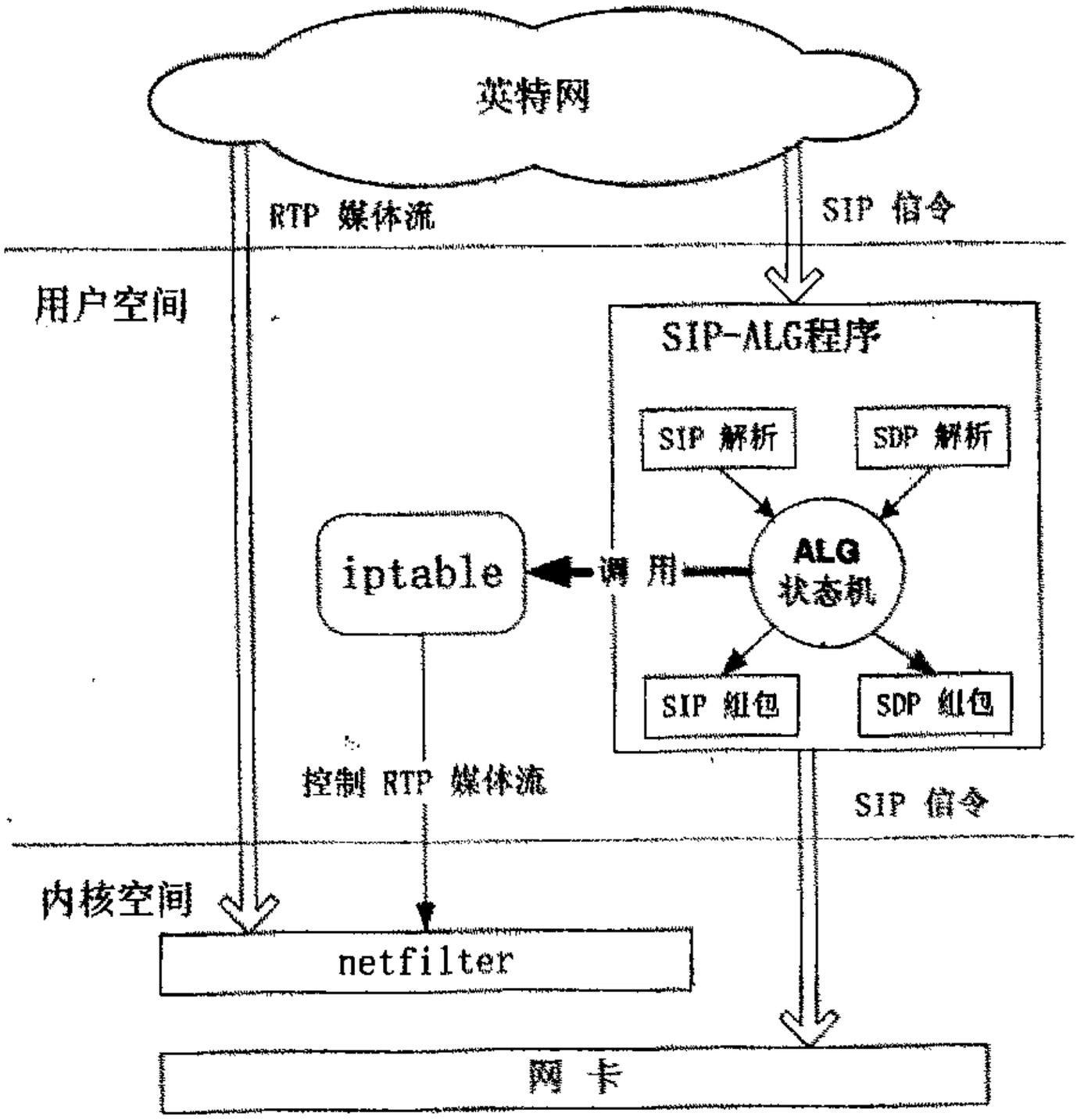

图 5-2 ALG 模块接口图

图中为 SIP-ALG 模块与 vLinux 其它模块的联系（接口），以及 SIP-ALG 本身的结构划分[33][34][35]。

SIP-ALG 核心部分为其 ALG 状态机。SIP-ALG 从英特网上接收 SIP 信令后，调用 SIP 解析函数和 SDP 解析函数对 SIP 信令进行解析。解析的结果交由 ALG 状态机进行处理。状态机会根据 SIP 信令包的内容已经当前状态对消息进行修改。而后，SIP 组包和 SDP 组包函数将修改结果重新组包然后调用内核的套接口程序发送 SIP 消息[36]。

ALG 状态机除了修改 SIP 消息，它还根据 SIP 消息交互的信息调用 vlinux 中的 iptables 命令，修改 NAT 规则，以便 RTP 媒体流能够正确地穿越 NAT 到达 SIP 用户。

5.2.2 软件框架

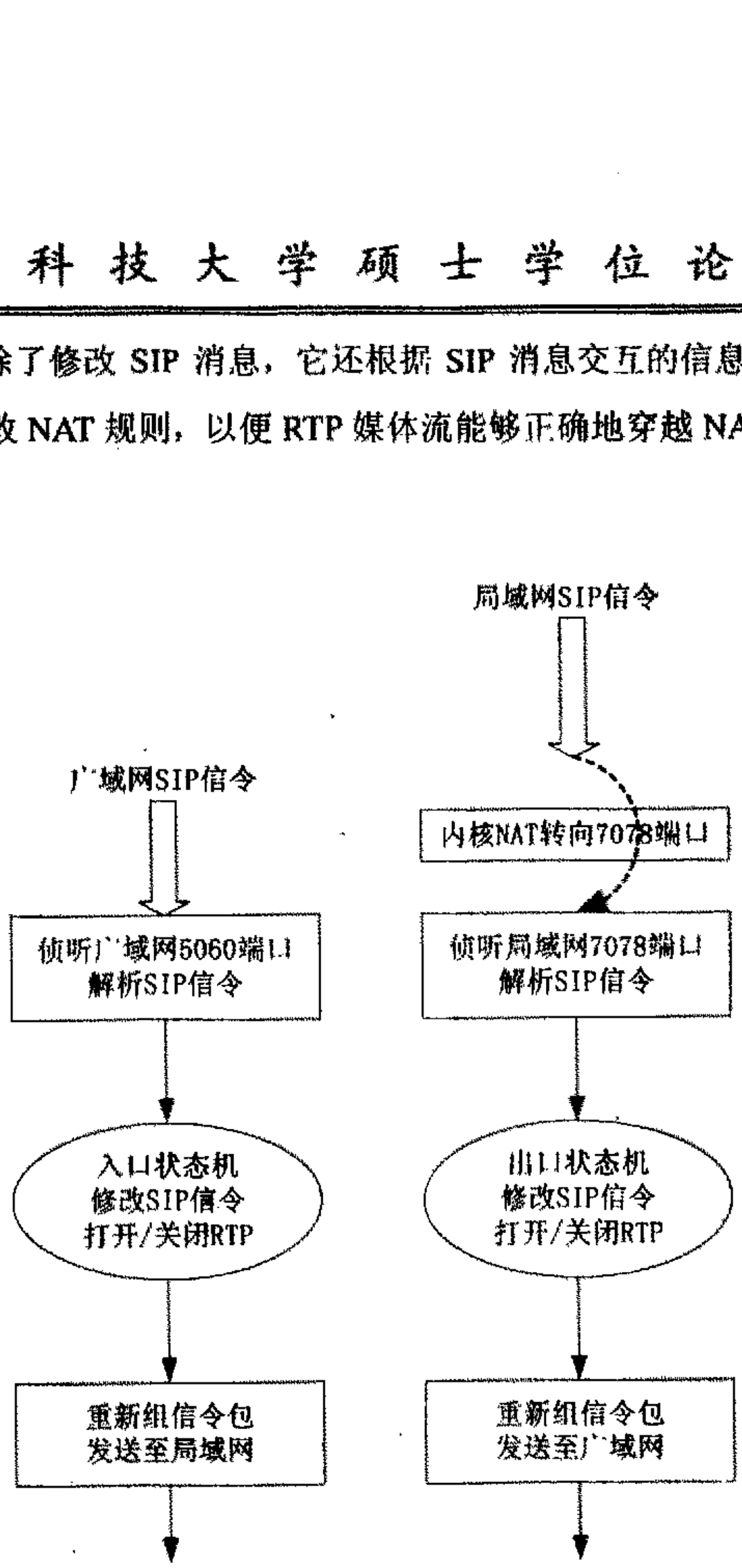

图 5-3 ALG 软件框架图

如图为 SIP-ALG 的软件结构[37]。SIP-ALG 将信令分为两种类型进行处理。从广域网发送至局域网的包叫做入口包（inbound packets），反之从局域网发出到广域网的包叫做出口包（outbound packets）。可以看出，软件主要功能由两个状态机完成。而对于消息和媒体的发送接收也做了一定的处理。5.2.3 节介绍消息和媒体发送的处理，而 5.3 节将着重介绍状态机。

5.2.3 消息和媒体的发送接收

由于状态机的修改，广域网发给局域网的信令消息目的地址就是网关的广域网口，并且端口为 5060。所以 SIP-ALG 只需要侦听网关的广域网口的 5060 端口即可收到广域网发来的 SIP 信令消息[38]。

然而，即使有状态机的修改，局域网 SIP 用户发出的信令包目的地址并不为网关的局域网口地址，而是一个广域网地址。根据 Linux NAT 实现原理，这样的包仅会被网关内核[39]的 NAT 处理并由它转发至广域网，用户空间得不到这样的包。但是 SIP-ALG 是运行在用户空间的程序，所以必须有一个办法将局域网用户的信令包也转交到用户空间的 SIP-ALG 程序。

考虑到局域网用户的信令包有一个显著特点，即信令包为 UDP 包，目的端口为 5060。结合 iptables 的 redirect 功能。我们可以设计一个 NAT 规则：将所有目的端口为 5060 的局域网内 UDP 包重定向至本地 7078 端口。这样 SIP-ALG 就能收到所有局域网 SIP 用户发出的信令包，并且能够根据信令包的 start line 得知信令包重定向前的目的 IP 地址。这也就是图中局域网信令用虚线经过 NAT 的含义。

由于媒体流本身不包含 IP 地址信息。所以状态机会建立针对 RTP 媒体流的 NAT 规则，根据这个规则，媒体流能够在 Linux 内核中及时得到处理转发至局域网或者广域网，而不需要发送至用户空间由 SIP-ALG 程序处理。

5.3 状态机设计

5.3.1 设计基础

SIP 穿越 NAT 主体工作由两个状态机来完成。这两个状态机分别处理入口 SIP 消息和出口 SIP 消息。

我们使用端口映射表来记录和设置从内网到外网 IP 地址和端口的修改。基于 SIP 在呼叫过程中 Call-ID 是不变而且唯一的这个特点，端口映射表的索引为 Call-ID。这个端口映射表的内容是根据 SIP 消息逐步建立的。而网关中 RTP 通道的打开也正是根据这个端口映射表调用 iptables 命令建立 NAT 规则。

我们同时也使用一个用户表（或者通过 REGISTER 消息建立这张表），该表记录了用户名和内网 SIP 电话的 IP 地址以及端口。以便能够知道将入口 SIP 消息发向内网的哪个地址。

对 SIP 消息的修改可以总结成为一张修改表，该表记录了对出口/入口消息各个字段修改的方式，状态机实际上是一种结合信令状态以及这张修改表的方式。所以，本质上而言状态机是建立在对呼叫成功 SIP 消息修改和呼叫失败 SIP 消息修改方式的基础之上。

下面我们首先讨论 NAT 对两种呼叫修改流程 SIP 消息的修改，尔后将修改方式总结为一张修改表。最后用两个状态机来实现这样的修改。

5.3.2 呼叫成功消息修改

首先介绍对成功建立呼叫的消息修改方式。

如图为 SIP 消息穿越的拓扑图：

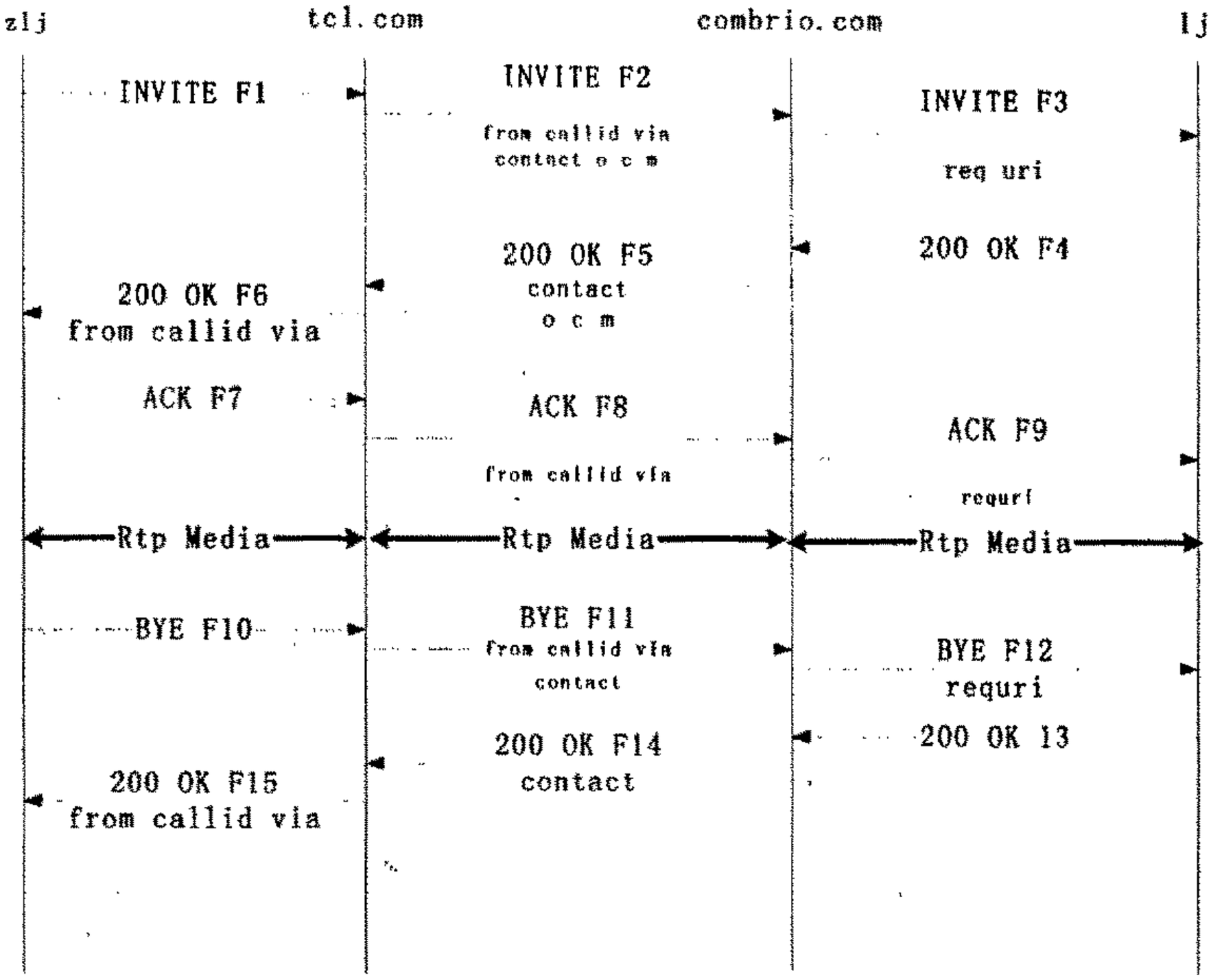

图 5-4呼叫成功修改图

图中，zlj 和 lj 为两个 SIP 电话，tcl.com 和 combrio.com 为分别负责 zlj 和 lj 的两个 ALG-NAT 网关。F1 到 F15 为从呼叫建立到呼叫拆除过程中的 15 个信令消息。每个信令消息第一行为信令名，第二行为需要修改的 SIP 头域，第三行则为需要修改的 SDP 头域。

下面将详细阐述 ALG 如何修改字段以及记录信令状态。为了简便起见，除了第一个请求包和响应包外，我们仅列举出需要修改的字段以及修改的结果。其中删除线代表需要修改的地方。

F1 INVITE zlj->tcl.com

~~==========~~

~~---------- IP Header ----------~~

~~SIP: 10.1.1.1:5060~~

```
DIP: 202.114.2.3:5060
---------- SIP Message ----------
INVITE sip:lj@202.114.2.3:5060 SIP/2.0
Via: SIP/2.0/UDP 10.1.1.1:5060
To: lj <sip:lj@202.114.2.3:5060>
From: zlj <sip:zlj@10.1.1.1:5060>
Call-ID: a84b4c76e66710@10.1.1.1
Contact: <sip: zlj@10.1.1.1:5060>
---------- SDP Message ----------
v=0
o=lj 53655765 2353687637 IN IP4 10.1.1.1
s=Session SDP
c=IN IP4 10.1.1.1
m=audio 49172 RTP/AVP 0
==========
```

这是 zlj 发起的呼叫，可以看到，他所呼叫的目的地址即为负责 lj 用户的 NAT 网关。

F2 INVITE tcl.com->combrio.com

```
==========
---------- SIP Message ----------
Via: SIP/2.0/UDP 10.1.1.1:5060 173.3.8.1:5060
From: zlj <sip: zlj@ 10.1.1.1:5060 173.3.8.1:5060>;
Call-ID: a84b4c76e66710@10.1.1.1 173.3.8.1
Contact: <sip: zlj@ 10.1.1.1:5060 173.3.8.1:5060>
---------- SDP Message ----------
o=lj 53655765 2353687637 IN IP4 10.1.1.1 173.3.8.1
c=IN IP4 10.1.1.1 173.3.8.1
m=audio 49172 60000 RTP/AVP 0
```

F2 为经过 combrio 网关修改后的 SIP 消息。

可以看到在 F1 消息中 from callid via contact 以及 o c m 头域都是使用的 zlj 本身的虚拟 IP 地址。而这样的 IP 地址在英特网上是不可见的，所以，ALG 把它们均改为

NAT 本身 IP 地址进行传输。

在这个时候 ALG 需要建立一个映射规则以记录 zlj 所发起的呼叫。

客户端建立完整规则：

当 F2 INVITE 离开 ALG 网关的时候，它即可建立该 Call-ID 对应映射表的全部内容[40]。

表格 5-1 端口映射表

Call-ID	Source IP &Port	RTP C field & m port
a84b4c76e66710	10.1.1.1:5060⇔173.3.8.1:5060	10.1.1.1⇔173.3.8.1 49172⇔60000
.....	.....	.....

图中外网的 IP 地址为 tcl.com 的 IP 地址。

```
F3 INVITE     combrio.com=>lj
              ==========================================
              ---------- SIP Message ---------------------
              INVITE sip: lj@202.114.2.3:5060 192.168.88.1:5060 SIP/2.0
              ---------- SDP Message ---------------------
```

当 F2 消息到达 combrio 网关的时候，combrio 的网关根据一个用户表网关就能够将 SIP 消息发向内网的 SIP 电话（lj）。

表格 5-2 端口映射表

User name	IP	PORT
lj	192.168.88.1	5060
.....	.....	.....

同时，对于 combrio 的网关也应该建立起响应的映射。然后将 SIP 消息发送给内

网。注意到这个时候 SIP 消息需要修改的只是 request－URL。

服务器端建立 SIP 规则：

表格 5-3端口映射表

Call-ID	Destination IP &Port	RTP C field & port
a84b4c76e66710	202.114.2.3:5060⇔192.168.88.1:5060	空
.....	.....	.....

F4 200 OK　　lj -> combrio.com

```
==========================================================
---------- SIP Message ---------------------
SIP/2.0 200 OK
Via: SIP/2.0/UDP 173.3.8.1:5060
To: lj <sip:lj@192.168.88.1:5060>
From: zlj <sip:zlj@173.3.8.1:5060>;
Call-ID: a84b4c76e66710@173.3.8.1
Contact: <sip: lj@ 192.168.88.1:5060>
---------- SDP Message ---------------------
  v=0
  o=lj 53655765 2353687637 IN IP4 192.168.88.1
  s=Session SDP
  c=IN IP4 192.168.88.1
  m=audio 49160 RTP/AVP 0
```

根据 SIP 协议，响应包会发向请求包的 Via 字段，由于已经 tcl 网关已经将 Via 修改为自己的外部 IP 地址，所以，200 OK 响应包能够发向 tcl.com 网关。

F5 200 OK　　combrio.com->tcl.com

```
==========================================================
---------- SIP Message ---------------------
Contact: <sip: lj@ 192.168.88.1:5060202.114.2.3:5060>
```

---------- SDP Message ----------

c=IN IP4 ~~192.168.88.1~~202.114.2.3

m=audio ~~4916~~60200 RTP/AVP 0

当 F4 200 OK 包离开 combrio 网关的时候，combrio 网关发现它的 Contact 字段为内网 IP 地址，于是它将 Contact 字段改为自己的外网 IP 地址。同时在 SDP 部分它修改 o 和 c 字段为外网 IP 地址。

对于 m 字段，当 combrio 网关发送这个 SIP 消息的时候，它认为应该打开 RTP 通道了。于是，网关将分配一个外网发送 RTP 媒体的端口（例子中为 60200）。这个端口将与 lj 发出的信令消息中 RTP 媒体端口进行 NAT 映射，使得发给外网这个端口的 RTP 包转发到内网 SIP 电话所要求的端口上。网关使用 iptables 中增加一个映射的规则来实现这个转发。

服务器端建立完整规则：

F4 200 OK 离开 combrio 网关的时候网关应该建立和记录以下规则，并使用 iptables 来实现 RTP 的转发，也就是建立了 RTP 通道

表格 5-4端口映射表

Call-ID	Destination IP &Port	RTP C field & port
a84b4c76e66710	202.114.2.3:5060⇔ 192.168.88.1:5060	192.168.88.1⇔ 202.114.2.3 49160⇔60200
……	……	……

F6 200 OK tcl.com->zlj

==========

---------- SIP Message ----------

Via: SIP/2.0/UDP ~~173.3.8.1:506~~10.1.1.1:5060

From: zlj <sip: zlj@ ~~173.3.8.1:506~~10.1.1.1:5060>

Call-ID: a84b4c76e66710@~~173.3.8.1~~10.1.1.1

```
---------- SDP Message ----------------------
==========================================
```

当 lj 的 F6 200OK 消息到达 tcl 网关的时候，网关 ALG 则通过 Call-ID 查找映射表将 200OK 转发至 zlj SIP 电话。同时网关也应该打开 RTP 通道，准备 RTP 的传输。

F7 到 F9 的 ACK 穿越通过前面建立的规则也作类似 INVITE 的修改:

到此，呼叫成功建立，并且语音媒体流也开始传输。

假设 zlj 首先挂机:

F10 BYE zlj -> tcl.com

```
==========================================
------------ IP Header -------------------
SIP: 10.1.1.1:5060
DIP: 202.114.2.3:5060
---------- SIP Message ----------------------
   BYE sip:lj@202.114.2.3:5060 SIP/2.0
   Via: SIP/2.0/UDP 10.1.1.1:5060
   From: zlj <sip:zlj@10.1.1.1:5060>
   To: lj <sip:lj@202.114.2.3:5060>
   Call-ID: a84b4c76e66710@10.1.1.1
==========================================
```

挂机后，zlj 则会发出以上的 BYE 请求消息。

BYE 消息穿越两个 NAT 到达 lj 的过程和 INVITE 类似，就不重复描述，下面主要描述 BYE 的 200 OK 响应的处理。

F13 200 OK lj->combrio.com

```
==========================================
------------ IP Header -------------------
SIP: 192.168.88.1:5060
DIP: 173.3.8.1:5060
---------- SIP Message ----------------------
SIP/2.0 200 OK
   Via: SIP/2.0/UDP 173.3.8.1:5060
```

```
From: zlj <sip:zlj@173.3.8.1:5060>
To: lj <sip:lj@192.168.88.1:5060>
Call-ID: a84b4c76e66710@173.3.8.1
```

Lj 会对 BYE 消息发出以上响应。

F14 200 OK combrio.com->tcl.com

```
---------- SIP Message ----------
SIP/2.0 200 OK
   Via: SIP/2.0/UDP 173.3.8.1:5060
   From: zlj <sip:zlj@173.3.8.1:5060>
   To: lj <sip:lj@192.168.88.1:5060>
Call-ID: a84b4c76e66710@173.3.8.1
```

combrio 网关会转发 200 OK 响应消息。处理方法与 F5 类似。这里需要说明的是，当网关收到对 BYE 消息的 200 OK 回应的时候，它就应该将该端口映射表删除。考虑到 SIP 协议中会重传 200 OK 响应消息，所以网关不马上删除端口映射表而是等待一段时间候再删除它。

F15 200 OK tcl.com->zlj

```
---------- SIP Message ----------
SIP/2.0 200 OK
   Via: SIP/2.0/UDP 173.3.8.1:506010.1.1.1:5060
   From: zlj <sip:zlj@173.3.8.1:506010.1.1.1:5060>
Call-ID: a84b4c76e66710@173.3.8.110.1.1.1
```

当 tcl 网关修改 200 OK 响应消息发送给 zlj 后，它同样不马上删除端口映射表而是等待一段时间候再删除它。

5.3.3 呼叫失败消息修改流程

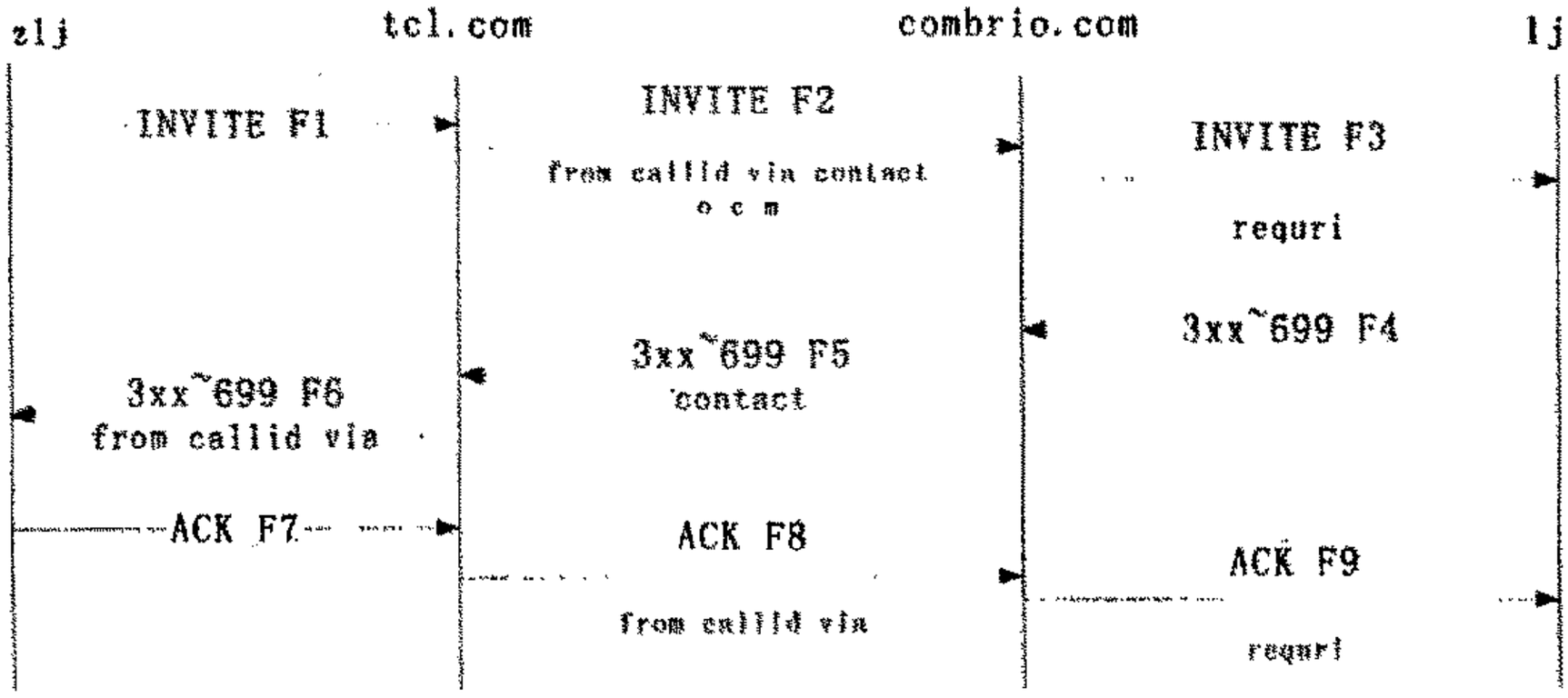

图 5-5 呼叫失败修改图

如图，呼叫失败修改流程和成功流程实际上是类似的，这里就不再列举所有的 SIP 消息。注意到这种情况下，删除端口映射表的时机是在两个网关收到针对 3xx～699 的 ACK 的时候。

5.3.4 消息修改表

根据以上修改 SIP 消息的流程总结以下对字段修改的修改表[41][42]：

表格 5-5 SIP 消息修改表

	字段	客户端	服务器端
入口请求消息	To	将内网IP改为ALG的WAN口IP	不予修改
	From	不予修改	不予修改
	Call_ID	将内网IP改为ALG的WAN口IP	不予修改
	Via	不予修改	不予修改
	Request_URI	将内网IP改为ALG的WAN口IP	将内网 IP 改为 ALG 的 WAN 口 IP
	Contact	不予修改	不予修改
入口响应消息	To	不予修改	不予修改
	From	将内网 IP 改为 ALG 的 WAN 口 IP	不予修改
	Call_ID	将内网 IP 改为 ALG 的 WAN 口 IP	不予修改
	Via	将内网 IP 改为 ALG 的 WAN 口 IP	将内网 IP 改为 ALG 的 WAN 口 IP
	Request_URI	(I)	(I)
	Contact	不予修改	不予修改
出口请求消息	To	不予修改	不予修改
	From	将ALG的WAN口IP改为对应的内网IP	不予修改
	Call_ID	将ALG的WAN口IP改为对应的内网IP	不予修改

	Via	将ALG的WAN口IP改为对应的内网IP	将ALG的WAN口IP改为对应的内网IP
	Request_URI	不予修改	不予修改
	Contact	将ALG的WAN口IP改为对应的内网IP	将ALG的WAN口IP改为对应的内网IP
出口响应消息	To	将ALG的WAN口IP改为对应的内网IP	不予修改
	From	不予修改	不予修改
	Call_ID	将ALG的WAN口IP改为对应的内网IP	不予修改
	Via	不予修改	不予修改
	Request_URI	(I)	(I)
	Contact	将ALG的WAN口IP改为对应的内网IP	将ALG的WAN口IP改为对应的内网IP

(I) 回应消息不存在这个字段。

5.3.5 出口状态机

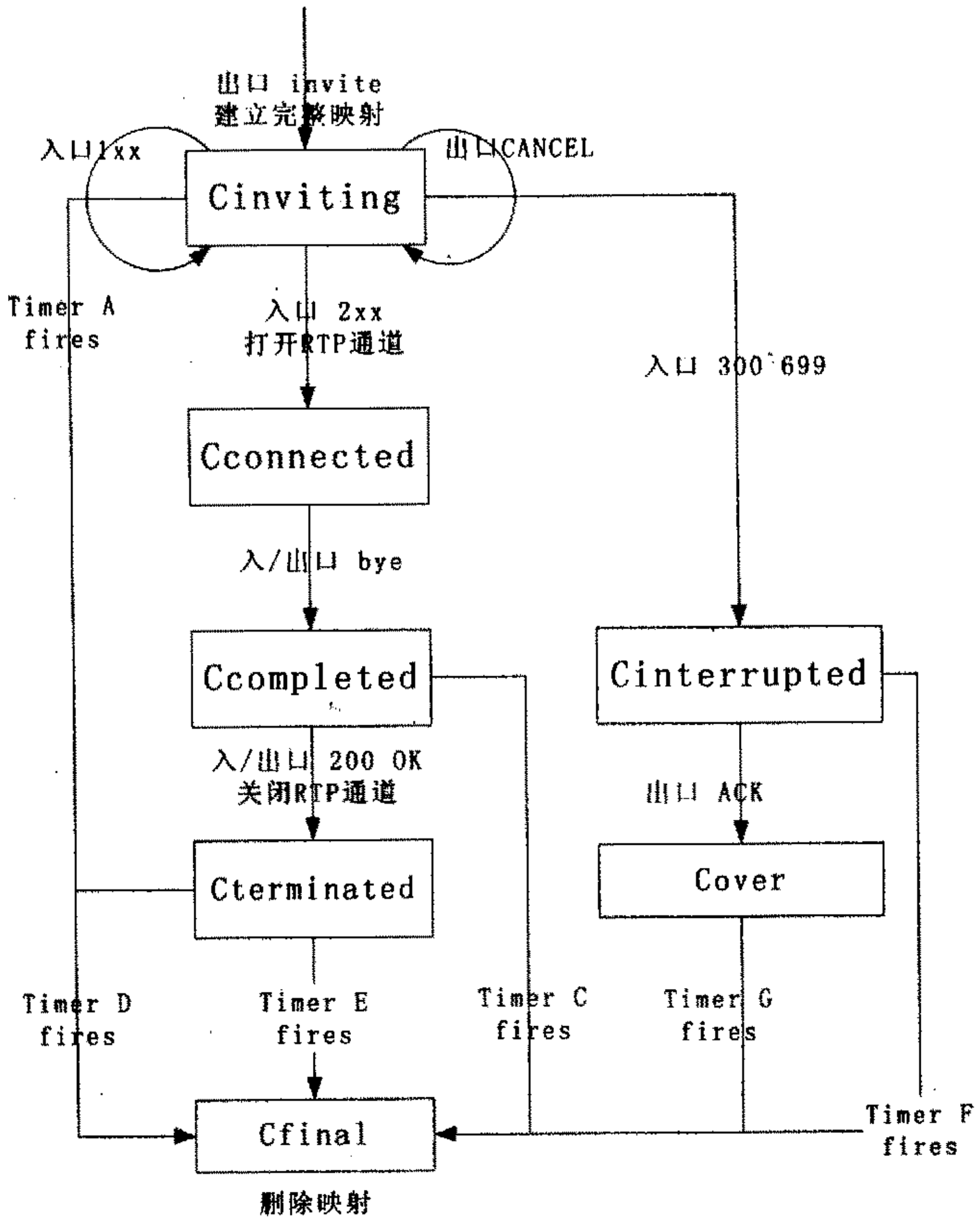

图 5-6 出口状态机

如图为出口状态机的设计图[43]。当网关收到一个出口的 INVITE 的时候，它即建立一个出口状态机，状态机索引为 Call-ID[44]。每个状态都以 C 打头表示这个状态机是 SIP 客户端发送 INVITE 后产生的。也可以称之为客户端状态机。图中，对各个信

令消息修改的方式遵循消息修改表，这里就不再一一列出，下面主要阐述状态机的变迁。

收到一个出口的INVITE的后，网关进入Cinviting状态，它将修改字段发送INVITE出去，同时根据 SIP 消息建立一个完整的端口映射。在这个状态收到对 1xx 响应和 CANCEL 请求均不进行状态变迁，而它针对呼叫成功和呼叫失败有两个分支。

对于呼叫成功分支，即图中变迁到 Cconnected 状态分支，它的变迁条件为收到入口的相同 Call-ID 的 2xx 响应包。收到后，状态机认为出口的 RTP 通道应该打开，则它根据端口映射调用 iptables 打开 RTP 通道，进行通话。通话结束后，通话双方中，有一方发出 BYE 请求，也就是图中所示入/出口 bye 消息。这个消息会导致状态变迁到 Ccomplete 状态。这个状态表明该状态机已经收到了 BYE 消息。该状态下如果再收到 200OK 的响应消息，状态机认为 RTP 通话已经结束，于是它将关闭 RTP 通道，进入 Cterminated 状态。Cterminated 状态虽然不能进行 RTP 通信但是还是能够收发相同 Call-ID 的 SIP 消息，这是为了状态机处理可能出现的重发 SIP 消息的情况。Cterminated 状态经过一段时间后则变迁至 Cfinal 状态，并删除映射和状态机本身。

对于呼叫失败分支，即图中变迁到 Cinterrupted 状态分支。这个变迁是收到 300~699 消息的时候变化的。进入这个状态说明这次呼叫不成功，当收到出口 ACK 的时候这个状态进入 Cover 阶段，这个阶段也是为删除状态机作准备，过一段时间后状态机进入到 Cfinal 阶段并删除状态机以及端口映射。

状态机中的几个 Timer 都是遵循 SIP 状态机的超时时间，可以查阅 SIP 状态机。比如 TimerA 是 64×T1。

5.3.6 入口状态机

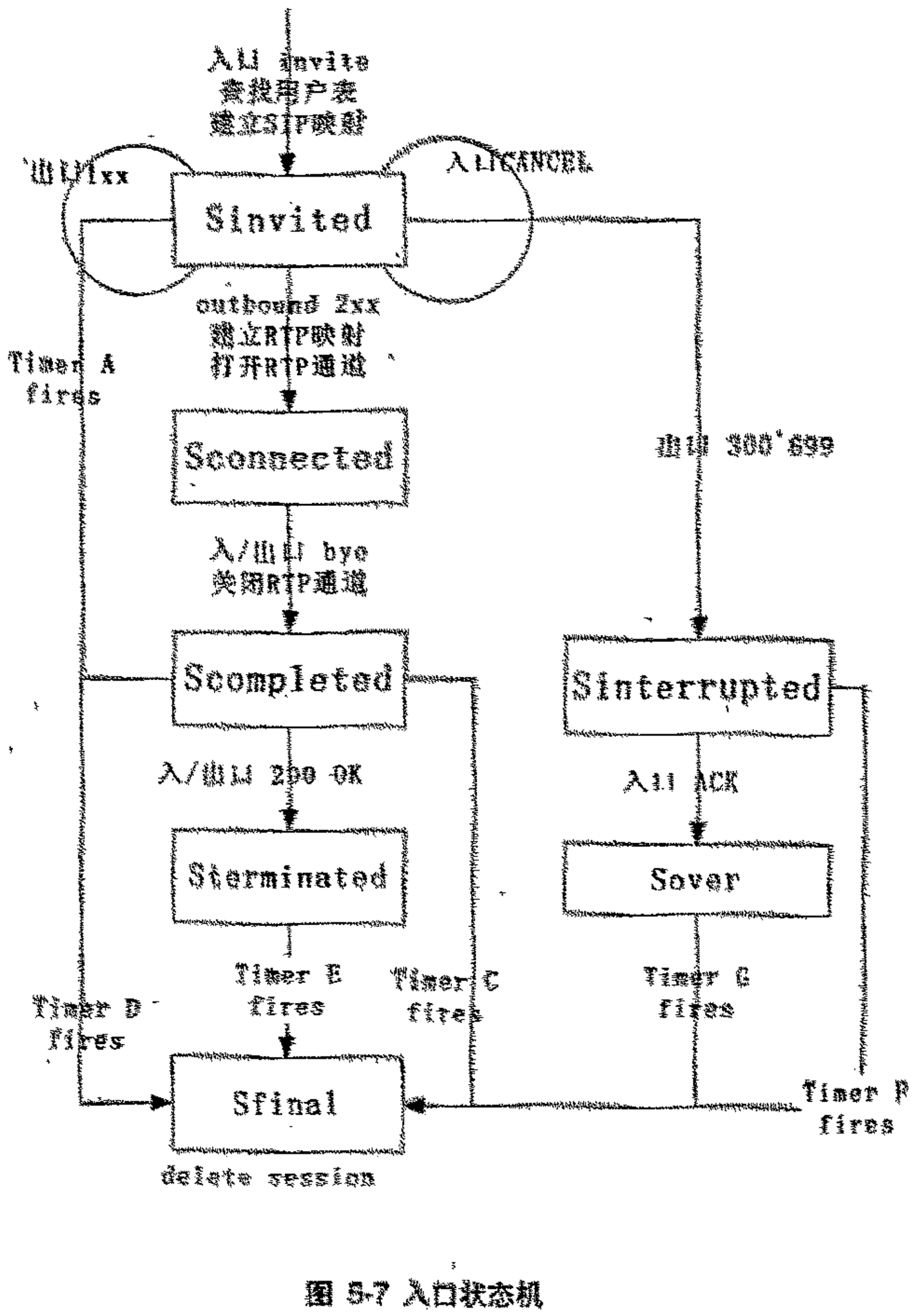

图 5-7 入口状态机

如图为入口状态机的设计图。当网关收到一个入口的 INVITE 的时候，它即建立一个入口状态机，状态机索引为 Call-ID。每个状态都以 S 打头表示这个状态机是 SIP 服务器端收到 INVITE 后产生的。也可以称之为服务器状态机。下面将主要阐述状态

机的变迁。

收到一个入口的 INVITE 的后，网关进入 Sinviting 状态，它查找用户表得到这个 INVITE 所要发送的对象，修改字段转发 INVITE，同时根据 SIP 消息建立一个的端口映射。这个端口映射并不完整，它仅记录了这个 Call-ID 所对应的 SIP 用户的内网 IP 地址端口和网关地址端口映射，在这个状态收到对 1xx 响应和 CANCEL 请求均不进行状态变迁，而它针对呼叫成功和呼叫失败有两个分支。

对于呼叫成功分支，即图中变迁到 Sconnected 状态分支，它的变迁条件为收到相同 Call-ID 的出口 2xx 响应包。收到后，状态机能够根据 SDP 的 c 和 m 字段内容建立端口映射中 RTP 内网 IP 地址端口和外网 IP 地址端口的映射。同时，状态机认为出口的 RTP 通道应该打开，则它根据端口映射调用 iptables 打开 RTP 通道，进行通话。通话结束后，通话双方中，有一方发出 BYE 请求，也就是图中所示入/出口 bye 消息。这个消息会导致状态变迁到 Scomplete 状态。这个状态表明该状态机已经收到了 BYE 消息。该状态下如果再收到 200OK 的响应消息，状态机认为 RTP 通话已经结束，于是它将关闭 RTP 通道，进入 Sterminated 状态。Sterminated 状态经过一段时间后则变迁至 Sfinal 状态，并删除映射和状态机本身。

对于呼叫失败分支，即图中变迁到 Sinterrupted 状态分支。这个变迁是收到出口 300～699 消息的时候变化的。进入这个状态说明这次呼叫不成功，当收到出口 ACK 的时候这个状态进入 Sover 阶段，这个阶段也是为删除状态机作准备，过一段时间后状态机进入也 Sfinal 阶段并删除状态机以及端口映射。

状态机中的几个 Timer 都是遵循 SIP 状态机的超时时间，可以查阅 SIP 状态机。

5.4 软件测试

5.4.1 测试环境

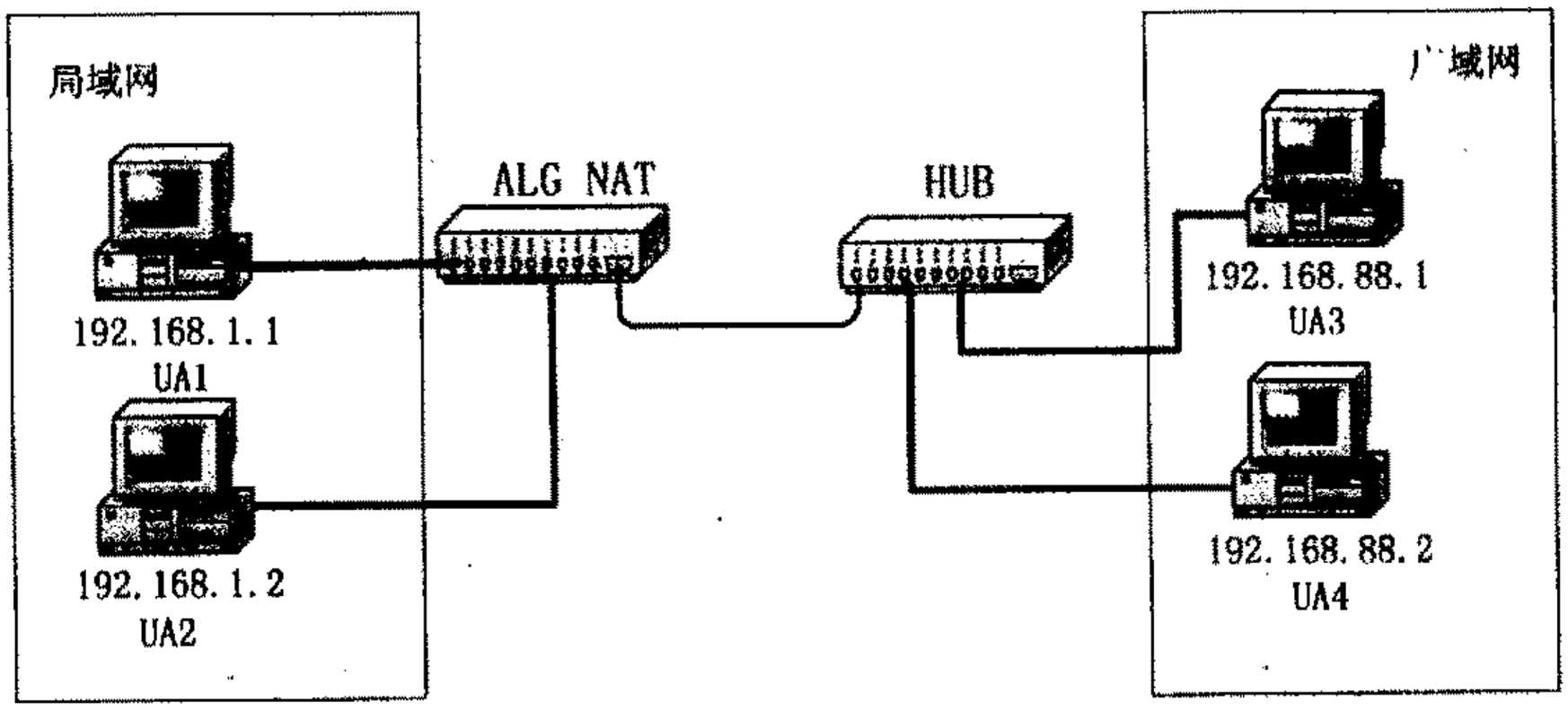

图 5-8 测试环境

如图，将两台电脑通过 HUB 连接于广域网，IP 配置为 88 网段，默认网关设置为 192.168.88.254。每台电脑上配置一个 JAVA PHONE，使用 5060 端口。

将 ALG -NAT 的 WAN 口与 HUB 相连。

将两台电脑连接于 ALG NAT 的 LAN 口。IP 配置为 192.168.1.0 网段，默认网关设置为 ALG NAT 的 LAN 口 IP：192.168.1.2。

5.4.2 每台电脑上配置一个 JAVA PHONE，使用 5060 端口，功能测试

表格 5-6 测试 1

测试 1	两局域网 java phone 与两广域网 java phone 互相拨打
测试过程	1. UA1 拨打 UA3,sip:zlj@192.168.88.1. 2. UA3 按 ANSWER 键，双方建立连接; 3. UA1 按 disconnected 键。 4. UA4 拨打 UA2,sip:pj@192.168.88.108 5. UA2 按 ANSWER 键，双方建立连接; 6. UA4 按 disconnected 键。 7. UA1 拨打 UA4,sip:xjh@192.168.88.2 8. UA4 按 ANSWER 键，双方建立连接; 9. UA1 按 disconnected 键。

	10. UA2 拨打 UA3,sip:zlj@192.168.88.1 11. UA3 按 ANSWER 键，双方建立连接; 12. UA3 按 disconnected 键。 13. ……。
检验项目及期望结果	1. UA1 显示进入 DAILING 状态,紧接着 UA1 显示进入 RINGING 状态，同时有铃声响起，UA3 显示进入 ALERTING 状态，同时有铃声响起。 2. UA1.UA3 均显示连接进入 connected 状态。 双方开始通话。 3. UA1,UA3 均显示进入 disconnected 状态。双方结束通话。 4. UA4 显示进入 DAILING 状态,紧接着 UA4 显示进入 RINGING 状态，同时有铃声响起，UA2 显示进入 ALERTING 状态，同时有铃声响起。 5. UA2.UA4 均显示连接进入 connected 状态。 双方开始通话。 6. UA2,UA4 均显示进入 disconnected 状态。双方结束通话。 7. UA1 显示进入 DAILING 状态,紧接着 UA1 显示进入 RINGING 状态，同时有铃声响起，UA4 显示进入 ALERTING 状态，同时有铃声响起。 8. UA1.UA4 均显示连接进入 connected 状态。 双方开始通话。 9. UA1,UA4 均显示进入 disconnected 状态。双方结束通话。 10. UA2 显示进入 DAILING 状态,紧接着 UA2 显示进入 RINGING 状态，同时有铃声响起，UA3 显示进入 ALERTING 状态，同时有铃声响起。 11. UA2.UA3 均显示连接进入 connected 状态。 双方开始通话。 12. UA2,UA3 均显示进入 disconnected 状态。双方结束通话。

	13. …… 通话正常清晰及状态转换正常，且多次拨打不会引起死机。
测试结果	测试通过。

表格 5-7 测试 2

测试 2	*两广域网 java phone 拨打同一局域网 java phone*
测试过程	1. UA4 拨打 UA2,sip:pj@192.168.88.108 2. UA2 按 ANSWER 键，双方建立连接; 3. UA3 拨打 UA2, sip:pj@192.168.88.108 4. UA2 按 ANSWER 键，双方建立连接; 5. UA2 点击第一个通话，按 disconnected 键。 6. UA2 点击第二个通话，按 disconnected 键。
检验项目及期望结果	1. UA4 显示进入 DAILING 状态,紧接着 UA1 显示进入 RINGING 状态，同时有铃声响起，UA2 显示进入 ALERTING 状态，同时有铃声响起。 2. UA2,UA4 均显示连接进入 connected 状态。 双方开始通话。 3. UA3 显示进入 DAILING 状态,紧接着 UA3 显示进入 RINGING 状态，同时有铃声响起，UA2 第一个通话下方会显示绿色闪动 ALERTING 状态，同时有铃声响起。 4. UA2,UA4 均显示连接进入 connected 状态。 双方开始通话。 5. UA2,UA4 均显示进入 disconnected 状态。双方结束通话。 6. UA1,UA3 均显示进入 disconnected 状态。双方结束通话。 通话正常清晰及状态转换正常，且多次拨打不会引起死机。
测试结果	测试通过。

5.4.3 性能测试

表格 5-8 测试 3

测试3	*一局域网java phone与一广域网java phone不停互相拨打*
测试过程	1. UA1 拨打 UA3,sip:zlj@192.168.88.1. 2. UA3 按 ANSWER 键，双方建立连接; 3. UA1 按 disconnected 键。 4. UA3 拨打 UA1,sip:lx@192.168.88.108 5. UA1 按 ANSWER 键，双方建立连接; 6. UA3 按 disconnected 键。 7. UA1 拨打 UA3,sip:zlj@192.168.88.1. 8. 重复以上操作…… 共拨打大约 50 次。
检验项目及期望结果	通话正常清晰及状态转换正常，且多次拨打不会引起死机。
测试结果	测试通过。

6 结论与展望

本文基于与台湾 Adigital 公司合作开发项目“基于 Linux 的 VoIP 路由器网关研发”，目前该项目已经进入产品化结束期。作为该项目中的一个子项目，本方案经过了严格测试，证明能够稳定完成 SIP 穿越 NAT 的工作。全文包括以下方面的工作以及创新：

1. 对 VoIP 相关协议如 SIP、SDP、RTP 等协议进行了学习研究；对 NAT 防火墙以及它们在 Linux 下实现进行了研究
2. 总结和归纳了现有的 SIP 穿越 NAT 方式。
3. 提出了一种实现 SIP-ALG 的解决方案。该方案运行于用户空间，用 REDERECT 方式得到内网 SIP 信令，总结了 NAT 对信令包的修改，设计了一个状态机记录信令过程并控制 RTP 通道。

基于以上研发成果，还可以继续研发工作如下：

1. 本文假设了所有 SIP 用户均使用 SIP 的知名端口进行通信。而实际情况可能并非如此，在后面的工作需要考虑 SIP 用户任意选择端口的情况。
2. 本文没有处理 Register 消息，而是由用户自行设置 ALG 中的用户表。在今后的工作中，可以首先处理 Register 消息，根据消息内容填写 ALG 用户表以及端口映射表。
3. 对于 SIP 协议中验证、加密，本文未作考虑，如果用户需要，可在今后的工作中加入对这方面的处理。

致 谢

子在川上曰：逝者如斯夫！

光阴似箭，转眼三年的研究生生活即将结束，值此论文完成之际，谨向给予我关心、支持和帮助的所有老师、同学和亲人们致以衷心的感谢。正是有了他们的支持和鼓励，我才能顺利的完成硕士学业。

首先，我要向我的导师杨宗凯教授表示衷心的感谢，杨老师学识渊博、治学严谨、诲人不倦。他务实的工作作风、积极进取的人生精神对我产生了巨大的影响。不仅使我在专业知识上获得了提高，也使我明白了许多做人的道理，使我受益匪浅。同样也要感谢师母程文青副教授，在攻读硕士期间，程老师在科研、学习和生活等方面都给了我许多关心和帮助，在此向她表示深深的谢意。

其次，我要感谢何建华副教授、杜旭博士、黄佳庆博士、在我的论文写作期间给我的很多宝贵的意见和建议。

再次，我要感谢余江博士、黄建博士、雷嘉硕士、杨明硕士、王敏硕士、徐静华硕士等互联网中心的所有成员，在课题的研究过程中给我的启迪与帮助；感谢台湾专家林祖宏先生等在科研工作中给予我的指导和提供这样一个研发的机会。

特别要感谢雷嘉硕士，协助我设计并实现了 ALG 的代码，以及对代码进行了全面测试。

最后，我要感谢我的父母、姐姐等亲人们对我生活上的关怀与学业上的鼓励，他们不求回报的亲情永远是我最强大的支持力量。

张连靖

二零零四年四月于华工园

参考文献

[1] ITU-T Recommendation H.323-Version 5, Packet Based Multimedia Communication Systems. May 2003

[2] Jonathab Rosenberg, Henning Schulzrinne, Gonzalo Camarillo et al. SIP: Session Initiation Protocol. IETF RFC 3261. June 2002.

[3] K. Egevang, P. Francis. The IP Network Address Translator (NAT). ITEF RFC1631May 1994.

[4] 陈德来. IP 电话原理及相关技术标准. 电信快报，1999，第 8 期：22 页

[5] 魏春城. SIP 协议的特点及应用. 电信科学，2002，第 9 期： 84-85 页

[6] 林铮. 软交换中的关键技术：SIP. 通信世界，2002，第 19 期： 39-40 页

[7] 赵慧玲，叶华等. 以软交换为核心的下一代网络技术. 北京：人民邮电出版社，2002 年 2 月. 120-122 页

[8] M. Handley, V. Jacobson. SDP: Session Description Protocol. IETF RFC 2327 April 1998.

[9] Audio-Video Transport Working Group, H. Schulzrinne, S. Casner, R. Frederick, V. Jacobson. RTP: A Transport Protocol for Real-Time Applications. RFC 1889 January 1996

[10] 庞向阳，欧阳柳波. 防火墙技术分析及其研究进展. 长沙大学学报，2002 年 6 月，第 16 卷，第 2 期：67～70 页

[11] 黄允聪，严望佳. 防火墙的选型、配置、安装和维护 [M] . 北京，清华大学出版社，1999，第二期：22～24 页

[12] 黄天成,宋小芹,任清珍. 防火墙技术综述[J] . 现代计算机技术，2000，第二期：56～57 页

[13] 陈晓峰. IPv4 和下一代 IP 地址[J]. 福建电脑，2003 年第 1 期：121～123 页

[14] R. Hinden, S. Deering. IP Version 6 Addressing Architecture. RFC2373. July 1998.

[15] Y. Rekhter, B. Moskowitz, D. Karrenberg, G. J. de Groot &E. Lear. Address Allocation for Private Internets. RFC 1918. February 1996.

[16] T. Hain. Architectural Implications of NAT. RFC 2993. November 2000.

[17] 孙盛源. 简述 NAT—网络地址转换. 甘肃科技， 2001 年 5 月第 7 期：23～25 页

[18] 卢宁 李定主 等. 防火墙及其 Linux 实现. 电脑开发与应用. 2001 年 15 卷第 2 期：113～116 页

[19] 吴兴. 用 iptales 实现包过滤型防火墙. 现代计算机技术. 2002-06-24：25～28 页

[20] 王中，郭兰申等. Linux 防火墙分析. 河北工业大学学报. 2001 年 4 月第 30 卷第 2 期：43～45 页

[21] Rusty Russell. Linux 2.4 Packet Filtering HOWTO. http://www.netfilter.org. 2002/01/24

[22] Rusty Russell and Harald Welte. Linux netfilter Hacking HOWTO. http://www.netfilter.org. 2002/07/02

[23] Rusty Russell. Linux 2.4 NAT HOWTO. http://www.netfilter.org. 2002/01/14.

[24] J. Rosenberg, R. Mahy, S. Sen. NAT and Firewall Scenarios and Solutions for SIP. IETF Draft. June 24, 2002

[25] J. Rosenberg, D. Drew, and H. Schulzrinne. Getting SIP through firewalls and NATs. IETF Draft. Feb. 2000.

[26] F. Thernelius. SIP firewall solution. IETF Draft. July 2000.

[27] J. Rosenberg, J. Weinberger, C. Huitema, and R. Mahy. STUN-simple traversal of UDP through NATs. IETF Draft. Apr. 2002.

[28] D. Yon. Connection-oriented media transport in SDP. IETF Draft. May 2002.

[29] J. Rosenberg, J. Weinberger, and H. Schulzrinne. SIP extensions for NAT traversal. IETF Draft Nov. 2001.

[30] Rosenberg J, Weinberger J, Schulzrinne H. NA T Friendly SIP (J). IETF Draft. July 20,2001

[31] B. Biggs. A SIP application level gateway for network addresses translation. IETF Draft. Mar. 2000.

[32] R. Mahy. Requirements for connection reuse in the session initiation protocol (SIP). IETF Draft. June 2002.

[33] W.Richard Stevens. UNIX 环境高级编程. 尤晋元等译. 北京机械工业出版社, 2001 年. 48~153 页

[34] W.Richard Stevens. TCP/IP 详解（卷 1）：协议. 范建华，胥光辉，张涛等译. 北京机械工业出版社.2000 年. 24~243 页

[35] Gary R.Wright W.Richard Stevens. TCP/IP 详解（卷 2）：实现. 陆雪莹，蒋慧等译. 北京机械工业出版社. 2000. 20~250 页

[36] 杜吉友，张艳霞，董德存. 穿越防火墙/NAT 的 SIP 通信研究. 中国数据通信，2004 年，第 2 期： 71～72 页

[37] 邵海东，周鹏，胡南军，陈道蓄，谢立，基于 Linux 的嵌入式系统设计与实现，计算机工程，2002 年 6 月：233～235 页

[38] Jon Postel. User Datagram Protocol. IETF RFC 768. Aug 1980

[39] 毛德操，胡希明，Linux 内核源代码情景分析，杭州：浙江大学出版社，2001. 20~500 页

[40] 严蔚敏，吴伟民. 数据结构. 第二版. 北京：清华大学出版社，1992 年. 30~103 页

[41] C. Huitema, Short term NAT requirements for UDP based peer-to-peer applications. IETF Draft. Feb. 2001.

[42] Fredrik Thernelius. SIP, NAT, and Firewalls. Department of teleinformatics. May 2000：56 页

[43] 张海藩. 软件工程导论. 第三版. 北京清华大学出版社，1998. 12~153 页

[44] 何永龙，林浒，雷为民. 一种 SIP NAT 应用网关的设计与实现. 小型微型计算机系统. 2002 年 8 月，第 23 卷，第 8 期 ： 56～58 页

附录 1 攻读硕士学位期间发表论文目录

杜旭，张连靖，余江等. IGMP Snooping 协议实现方案. 计算机应用，已录用. 署名单位：华中科技大学.

基于Linux的SIP穿越NAT研究与实现

作者：张连靖
学位授予单位：华中科技大学
被引用次数：1次

参考文献(45条)

1. 参考文献

2. ITU-T Recommendation H.323-Version 5,Packet Based Multimedia Communication Systems 2003

3. Jonathab Rosenberg.Henning Schulzrinne.Gonzalo Camarillo SIP:Session Initiation Protocol 2002

4. K Egevang.P Francis The IP Network Address Translator (NAT) 1994

5. 陈德来 IP电话原理及相关技术标准 1999(08)

6. 魏春城 SIP协议的特点及应用[期刊论文]-电信科学 2002(9)

7. 林铮 软交换中的关键技术:SIP 2002(19)

8. 赵慧玲.叶华 以软交换为核心的下一代网络技术 2002

9. M Handley.V Jacobson SDP:Session Description Protocol 1998

10. H Schulzrinne.S Casner.R Frederick.V.Jacobson RTP:A Transport Protocol for Real-Time Applications 1996

11. 庞向阳.欧阳柳波 防火墙技术分析及其研究进展[期刊论文]-长沙大学学报 2002(2)

12. 黄允聪.严望佳 防火墙的选型、配置、安装和维护 1999

13. 黄天成.宋小芹.任清珍 防火墙技术综述 2000(02)

14. 陈晓峰 IPv4和下一代IP地址 2003(01)

15. R Hinden.S Deering IP Version 6 Addressing Architecture 1998

16. Y Rekhter.B Moskowitz.D Karrenberg.G.J.de Groot E.Lear Address Allocation for Private Internets 1996

17. T Hain Architectural Implications of NAT 2000

18. 孙盛源 简述NAT-网络地址转换[期刊论文]-甘肃科技 2001(5)

19. 卢宁.李定主.郭爱红.郭朝辉 防火墙及其Linux实现[期刊论文]-电脑开发与应用 2002(2)

20. 吴兴 用iptales实现包过滤型防火墙 2002

21. 王中.郭兰申.王永滨.杨威 Linux防火墙分析[期刊论文]-河北工业大学学报 2001(2)

22. Rusty Russell Linux 2.4 Packet Filtering HOWTO 2002

23. Rusty Russell.Harald Welte Linux netfilter Hacking HOWTO 2002

24. Rusty Russell Linux 2.4 NAT HOWTO 2002

25. J Rosenberg.R Mahy.S Sen NAT and Firewall Scenarios and Solutions for SIP 2002

26. J Rosenberg.D Drew.H Schulzrinne Getting SIP through firewalls and NATs 2000

27. F Thernelius SIP firewall solution 2000

28. J Rosenberg.J Weinberger.C Huitema.R.Mahy STUN-simple traversal of UDP through NATs 2002

29. D Yon Connection-oriented media transport in SDP 2002

30. J Rosenberg.J Weinberger.H Schulzrinne SIP extensions for NAT traversal 2001

31. Rosenberg J.Weinberger J.Schulzrinne H NAT Friendly SIP 2001

32. B Biggs A SIP application level gateway for network addresses translation 2000

33. R Mahy Requirements for connection reuse in the session initiation protocol (SIP) 2002

34. W Richard Stevens.尤晋元 UNIX环境高级编程 2001

35. W Richard Stevens.范建华.胥光辉.张涛 TCP/IP详解(卷1):协议 2000

36. Gary R.Wright W.Richard Stevens.陆雪莹.蒋慧 TCP/IP详解(卷2):实现 2000

37. 杜吉友.张艳霞.董德存 穿越防火墙/NAT的SIP通信研究[期刊论文]-中国数据通信 2004(2)

38. 邵海东.周鹏.胡南军.陈道蓄.谢立 基于LinuX的嵌入式系统设计与实现[期刊论文]-计算机工程 2002(6)

39. Jon Postel User Datagram Protocol 1980

40. 毛德操.胡希明 Linux内核源代码情景分析 2001

41. 严蔚敏.吴伟民 数据结构 1992

42. C Huitema Short term NAT requirements for UDP based peer-to-peer applications 2001

43. Fredrik Thernelius SIP,NAT,and Firewalls 2000

44. 张海藩 软件工程导论 1998

45. 何永龙.林浒.雷为民 一种SIP NAT应用网关的设计与实现[期刊论文]-小型微型计算机系统 2002(8)

引证文献(1条)

1. 徐宏亮 基于SIP协议的软交换系统[学位论文]硕士 2006