第 6 卷 第 11 期 2006 年 6 月
1671 - 1815 (2006) 11 - 1556 - 05

科 学 技 术 与 工 程
Science Technology and Engineering

Vol. 6 No. 11 Jun. 2006

针对 SIP 的 STUN 解决方案的设计与实现

郭常清
(湖南大学软件学院, 长沙 410082)

摘 要 SIP 是一个基于文本的应用层协议, 但 SIP 协议本身无法实现让 SIP 消息安全地穿过 NAT 和防火墙。从 SIP 消息的特点出发, 提出一种无需扩展 SIP 协议的应用层解决方案, 引入 STUN 协议, 取得 IP 地址和端口的映射关系, 修改 SIP 和 SDP 消息的内容来保证通信连接, 从而实现对 NAT 的穿越。

关键词 VoIP 协议(SIP) 网络地址翻译器 STUN
中图法分类号 TP393; **文献标识码** A

目前, VoIP 技术在世界范围内已获得广泛应用, 而基于 SIP[1] (Session Initiation Protocol[RFC2543], 会话初始化协议) 的软交换技术已成为 VoIP 技术研究的一个新的热点。

SIP[1] 是 IETF 提出的在 IP 网络上进行多媒体通信的应用层控制协议。可用于建立、修改、终结多媒体会话和呼叫。其特点是简单、便于扩展和扩充, 且 SIP 借鉴了许多已有的 Internet 协议, 是实现增值综合业务的理想手段, 具有很好的发展潜力。

由于我国广泛使用的宽带城域网、企业网中普遍采用 NAT[2] (Network Address Translator[RFC1631], 网络地址翻译器), SIP 是一个基于文本的应用层协议, 建立会话所需的地址信息描述均存在于 SIP 消息中。而包含丰富地址信息的 SIP 消息处于应用层, NAT 只对 TCP/UDP 和 IP 包头中的地址和端口进行翻译, 从而造成载荷内的地址和端口与 IP 包头的源地址和源端口不一致, 会导致 NAT 外和 NAT 内的用户之间无法从 SIP 消息中得到有效的地址信息, 无法完成正常的会话建立过程。

为解决以上问题, 文中采用 STUN[3] (Simple Traversal of UDP Through NAT[RFC3489]) 实现对 NAT的穿越, 将 STUN 取得的(IP: 端口)映射直接写入 SIP 消息来建立正确的连接。系统采用 Microsoft 主推的 RTC Client API 进行开发, 以满足 SIP 软交换系统的全球统一标准, 在文中对解决方案和实现方法进行了详述。

2006年1月6日收到

1 SIP 技术简介

SIP 协议采用基于文本格式的客户端——服务器方式, 以文本形式表示消息的语法、语义和编码, 客户机发起请求, 服务器进行响应。SIP 独立于底层协议——TCP、UDP 或 SCTP, 采用自己的应用层可靠机制来保证消息的可靠传递。SIP 消息中使用的 SDP (Session Description Protocol[RFC2327], 会话描述协议) 用于 SIP 端点之间媒体通道建立所需参数的描述。

一个基本的 SIP 呼叫一般包括以下几个头域: Call - ID、Request - URI、Via、From、To、Cseq、Content - Length、Content- Type 和 Contact。这几个头域中携带了会话过程的基本地址信息, 通过它们即可完成一个较完整的 SIP 呼叫。文中将详述对这几个头域的改写。RFC2543 中对 SIP 请求 (Request) 定义了 INVITE、ACK、OPTIONS、BYE、CANCEL 和 REGISTER 等 6 个方法。通过 INVITE、ACK、BYE 和 REGISTER 等几个方法即可完成一个相对完整的 SIP 呼叫过程。同样, RFC2543 对 SIP 响应(Response) 定义了相关的回答, 与基本连接直接相关的有 100(Trying)、180(Ringing) 、200(OK) 和 404(Not found) 等, 文中将对以上信息进行分析。

2 STUN 穿越原理及系统实现方案

2.1 开发环境简介

系统所采用的开发环境如图 1 所示。

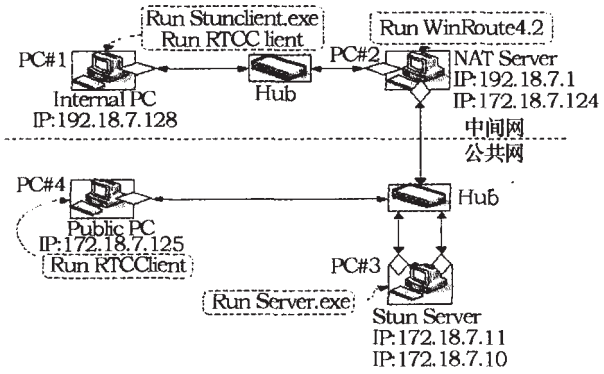

图 1 硬件拓扑图

PC#1 的 IP 配置为 192.18.7.128,在此机器上运行 STUN Client; 在 PC#2 上安装两块网卡,IP 分别配置成 192.18.7.1 和 172.18.7.124,PC#2 充当 NAT Server,由 PC#1 和 PC#2 共同组成私网。由 PC#3 充当 STUN Server, 安装两块网卡,IP 分别配置为 172.18.7.10 和 172.18.7.11,鉴于 STUN 协议要求使用四个(IP:端口)对来完成对 NAT 类型的判断,使用端口 3478、3479, 即可得到 (172.18.7.10:3478) (172.18.7.10:3479) (172.18.7.11:3478) (172.18.7.11:3479)等四个(IP:端口)对,PC#4 充当公网用户,IP 配置为 172.18.7.125。分别在 PC#1 和 PC#4 上运行所开发的 RTC Client 进行通话。

2.2 NAT 所带来的问题

NAT 顾名思义就是实现了网络地址的翻译,目前广泛应用于私有地址域与公有地址域的转换以缓解 IPv4 地址匮乏的问题。由于 NAT 的存在,那些使用到 IP 地址的高层应用将受到限制, 如图 1 所示, 以 PC#1 和 PC#4 之间建立 SIP 连接的情况为例, 当 NAT 内的 PC#1 向公网上的 PC#4 发送 INVITE 请求时,请求的头为:

INVITE: PC#1@ 172.18.7.125 SIP/2.0

Via: SIP/2.0/UDP 192.18.7.128

……(略)

INVITE消息的源 IP 地址经过 NAT 后, 被翻译为 172.18.7.124:45040。然而在 SIP 协议中,PC#4 不会按此地址返回 100(Trying)响应,而是按 Via 中的 192.18.7.128:29362 返回。由于这是个私有地址,发回的响应无法送到 PC#1 中,连接无法建立。

同样地,对于 SIP 所控制的媒体流的传输,有关地址和参数信息保存在 SIP 消息的 SDP 消息体中。因此,除了消息本身,SIP 所控制的媒体流的传输也无法完成。

2.3 STUN 所定义的 NAT 类型分类

STUN 协议用来帮助位于 NAT 之后的实体发现 NAT 的存在,判断 NAT 的类型,从而得到由 NAT 分配的绑定映射地址。STUN 无需对 NAT 进行改变, 适用于处理在应用程序实体与公网间有多重 NAT 的情况。

在实际应用中,UDP 在不同类型的 NAT 中处理是不一样的。根据 NAT 工作方式的不同,STUN 协议将 NAT 分为四类。

Full cone NAT:是指所有来自相同内部 IP 地址和端口的请求都映射到某一个相同的外部 IP 地址和端口上。此外,任意的外部主机可以通过发送信息包到内部主机对应的外部 IP 地址上来发送信息包(在 PC#2 上安装软件 WinRoute Lite 4.2,可模拟出此环境)。

Restricted cone NAT: 是指所有来自相同内部 IP 地址和端口的请求都映射到某一个相同的外部 IP 地址和端口上。与 Full cone NAT 不同的是,一个外部主机只能在内部主机曾经发送过信息包给它的情况下,才能够发送信息包到该内部主机(IP 必须一致, 端口号可以不一样,10.0.0.1:6666->172.18.7.10:3478,172.18.7.10:3478->10.0.0.1:6666 成功, 172.18.7.10:3479->10.0.0.1:6666 也成功)(在 PC#2 上安装软件 WinGate V5.0.2, 可模拟出此环境)。

Port restricted cone NAT:类似于 restricted cone NAT,但是限制的还包括端口。也就是说,只有在内部主机(通过 IP 地址和端口号)给外部主机发送过信息包的情况下, 该外部主机才可以发送信息包给该内部主机, 并需要明确地表明原来接收内部主机信息时的 IP 地址和端口号 (Restricted cone NAT 不需要相同的端口号,10.0.0.1:6666->172.18.7.10:3478,172.18.7.10:3478->10.0.0.1:6666 成功,172.18.7.10:3479->10.0.0.1:6666 不成功)(在 PC#2 上安装软件 Sygate4.0,可模拟出此环境)。

Symmetric NAT: 指的是所有来自相同内部 IP 地址和端口到同一目的地的请求都映射到某一个相同的外部 IP 地址和端口上。假如相同的主机使用相同的源 IP 地址和端口号发送信息包到另一目的

地,那么将会映射到另一外部地址上。而且,只有当外部主机接收到内部主机的信息包后才能发送信息包回给该内部主机(10.0.0.1:6666->172.18.7.10:3478,外部地址为 115.68.45.123, 10.0.0.1:6666->172.18.7.11:3479,采用的外部地址为115.68.45.134)(STUN无法穿越此类 NAT)。

2.4 STUN 技术绑定原理简介

STUN Client:客户端是一个发送 STUN 请求的实体。STUN 客户端可以运行在终端系统上(end system),譬如 PC 机,或者是网络中的一个元素,譬如一个会议服务器。

STUN Server:服务器是接收 STUN 请求,发送 STUN响应的实体。一般来说,STUN 服务器都附属在公用因特网上。

STUN 协议是一种简单的 C/S 协议。客户端发送请求到服务器,然后服务器给予响应。请求有两种,一种是绑定请求,通过 UDP 发送;另一种是共享秘密请求,通过 TLS over TCP 传送。STUN 绑定请求被用于发现 NAT 的存在, 并且发现 NAT 所产生的对应公共 IP 地址和端口号。绑定请求通过 UDP 协议发送到服务器上, 当绑定请求到达服务器后,说不定已经通过了一层或多层的 NAT 了。因此,服务器所收到的请求的源地址肯定就是由最靠近服务器的 NAT 所提供的对应地址,STUN 服务器复制源地址和端口号到绑定响应中去,并且按照源地址和端口号发回响应。经过多层的 NAT 后,响应还是会回到客户端上的。

技巧在于利用 STUN 协议来发现 NAT 的存在,判断出类型,并得到和利用它们所分配的绑定。

几个主要的 STUN 属性定义如下。

(1) MAPPED-ADDRESS(IP 地址和端口号),它总是位于绑定响应中,代表了服务器所收到的绑定请求中的源 IP 地址和端口号。如有 NAT 存在,该(IP:端口)就是内部(IP:端口)所对应的外部映射。

(2) RESPONSE-ADDRESS(IP 地址和端口号),它在绑定请求中, 代表需要送到的 IP 地址和端口号。(不一定需要有,没有的话,服务器会回送到请求的源地址和端口)。

(3) CHANGE-REQUEST,它包括两个标志负责来控制发送响应的 IP 地址和端口号。这两个标志分别是“改变 IP”和“改变端口号”。CHANGE-REQUEST 属性只存在于绑定请求中。这两个属性分别用于判定客户端是处于 Restricted cone NAT 或是 Port restricted cone NAT 后面。它们指示服务器使用另外的 IP 地址和端口号来发送绑定响应。这个属性是可选项。

(4) CHANGED-ADDRESS,它位于绑定响应当中,假如客户端需要‘改变 IP’和‘改变端口号’,它通知客户端改变后所用的源 IP 地址和端口号。

(5) SOURCE-ADDRESS,它只位于绑定响应当中,它表明响应是由什么地方所发送的,源 IP 和端口号是什么。在探测二次 NAT 配置中非常有效。

由以上所述可以得出, 在实际的网络环境中,共有 6 种可能的情况:

(1) 在公众 internet 上,

(2) 在拦截 UDP 的防火墙内,

(3) 可以允许 UDP 出去,而响应必须回到请求的主机上 (很像 symmetric NAT, 但是没有地址翻译,叫这种为 symmetric UDP Firewall),

(4) Full-cone NAT,

(5) Symmetric NAT,

(6) Restricted cone or Port restricted cone NAT。

正确识别 NAT 类型是实现穿越的关键,该过程的原理如图 2 所示。

现结合硬件拓扑图 1 和判别原理图 2 解释判

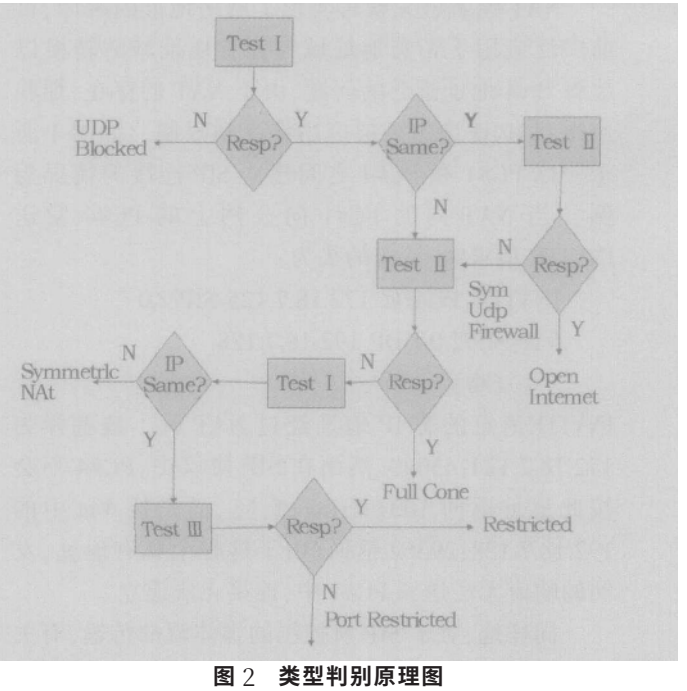

图 2 类型判别原理图

断过程。

该流程使用三个测试。在测试 I 上,客户端发送 STUN 绑定请求到服务器上 (没有设定 CHANGE-REQUEST 属性的标记, 也没有 RESPONSE-ADDRESS 属性)。这导致服务器发送响应到发出请求的 IP 和端口号。在测试 II 中,客户端发送绑定请求(带有 CHANGE-REQUEST 属性中的'改变 IP'和"改变端口号'的标记)。在测试 III 中,客户端会发送仅带有'改变端口号'标记的绑定请求。

客户端(192.18.7.128:29362)先开始测试 I,在 STUN Server 上使用(172.18.7.10:3478),假如该测试没有产生任何响应,那么客户端就会知道不能使用 UDP 连接。假如产生了响应,客户端就需要检测 MAPPED-ADDRESS 属性。假如得到的地址和端口号和发送信息包的 IP 地址和端口号一致的话,就证明并没有经过 NAT 转换。然后进行测试 II。

此时 STUN Server 发送响应时使用(172.18.7.11:3479),假如收到一个响应,那么客户端就会知道它拥有与 internet 开放的接口(至少,它位于一个功能类似 full-cone NAT 的防火墙后面,但是缺乏了地址翻译功能)。假如收不到响应,那么客户端就会知道它位于 symmetric UDP firewall 的后面。

那么当 IP 地址和端口号并不符合通过测试 I 所得到响应的 MAPPED-ADDRESS 属性,那就证明它必定位于 NAT 背后。然后进行测试 II(此时 STUN Server 上发回响应的是(172.18.7.11:3479)),假如收到响应的话, 客户端就会知道它处于 full-cone NAT 后面。假如没有收到响应,再进行测试 I,但是这次, 将使用从响应中 CHANGED-ADDRESS 属性所得到的 IP 和端口号来进行测试 I, 即使用(172.18.7.11:3479)。假如返回的 MAPPED-ADDRESS 属性中的 IP 和端口号与第一次测试 I 所得到的不一样,就证明它处于 symmetric NAT 后面。假如地址和端口号相同的话, 就证明客户端处于 Restricted NAT 或 Port restricted NAT 后面。为了确定究竟是处于哪个后面, 将要进行测试 III。即让 STUN Server 上使用(172.18.7.11:3478)。假如得到一个响应,就是处于 Restricted NAT 后面,反之,就是 Port restricted NAT。

类型判断结束后, 将返回映射端口的绑定结果,如图 3 所示。

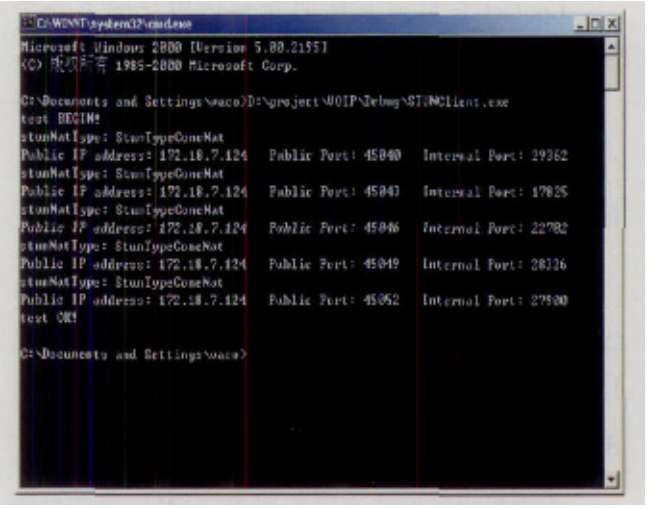

图 3　映射绑定关系图

由图 3 可知,得到五对绑定的映射地址,例如:内部(IP:端口)为(192.18.7.128:29362),外部为(172.18.7.124:45040),应在后台持续运行此程序,使 STUN Client 不断向 STUN Server 发送请求,保证映射地址的实时更新,才能保证会话连接的长时间有效。

2.5　使用所取映射改写 SIP 消息

1) 对 Via 头域的改写

Via 纪录请求所通过的路径。

原为 Via: SIP/2.0/UDP 192.18.7.128:29362,

改为 Via: SIP/2.0/UDP 172.18.7.124:45040。

由内网用户发出的 INVITE、ACK、BYE 和 REGISTER 等 4 个请求消息和 100(Trying)、180(Ringing)、200(OK)等 3 个回答中的 Via 头域均应进行以上修改。

2) 对 SDP 消息体的修改

SDP 消息体中描述了与会话及相关媒体特性的有关内容。与连接直接相关的头域为:

"Connection c =",描述主机的 IP 地址。

"Media m =" 接收端口以及接收媒体流的类型信息。

原为 c = IN IP 192.18.7.128,

m = audio 29362 RTP/AVP 0,

改为 c = IN IP 172.18.7.124,

m = audio 45040 RTP/AVP 0。

由公网用户返回的 200(OK)中的 SDP 消息体中的相应部分应进行同样的修改.

3) 对 Contact 头域的修改

Contact 头域通常标明能到达用户的下一个 URL,与注册、重定向等操作密切相关。

原为 Contact:〈sip: pc#1@192.18.7.128: 29362〉

改为 Contact:〈sip: pc#1@172.18.7.124: 45040〉

对 REGISTER 中的 Contact 头域,应进行同样的修改。

4) 对 Content- Length 头域进行修改。

由于 SIP 消息被改写,将导致消息体长度发生改变,应对相关的 INVITE 及 200 (OK) 等消息的 Content- Length 进行改写。

3 结论

本文提出的解决方案实现于应用层,无需对协议进行扩展。可以直接利用现有的 SIP 协议栈等资源。最根本的依据是 SIP 本身是一个纯文本形式的协议,完全可以在网络层实现对有效数据的定位,从而完成对文本数据的解析和修改。同时,采用在底层对 SIP 消息进行改写的方法,避免了将消息先送往应用层,然后再返回底层完成发送,可以提高效率。

参 考 文 献

1 Handley M, Schulzrinne H, Schooler E, Rosenberg J SIP: Session initiation protocol. RFC 2543, March 1999

2 Egevang K, Francis P. The IP network address translator. (NAT). RFC1631, May 1994

3 Rosenberg J, Weinberger J, Mahy R. STUN: Simple traversal of user datagram protocol (UDP) through network address translators (NATs). RFC 3489, March 2003

4 Srisuresh P, Kuthan J, Rosenberg J. Middlebox communication architecture and framework. IETF, Feb 2001

5 Rosenberg J, Weinberger J, Schulzrinne H. SIP Extension for NAT Traveral. IETF, Nov 21 2001

Implementation the Traversing of NAT in SIP- based VoIP System

GUO Changqing

(Department of Software, Hunan University, Changsha 410082)

[Abstract] SIP is a text- based application- layer protocol. It can't be used in NAT or firewall environment. According to the specialties of entities of SIP messages, an application layer solution without expanding SIP, by achieving the mapped relation of IP address and Ports with STUN and overwriting the SIP messages and SDP messages to assure the communication connection are presented. In this way the system can implement the traversing of the NAT.

[Key words] VoIP session initiation protocol network address translator STUN