基于 ICE 与 HTTP Tunnel 技术的 NAT/FW 穿越方案 *

陈世锋,陈　怡,徐晓峰,郭东辉
(厦门大学　厦门　361005)

摘 要　在现代网络中,越来越多的主机被部署在 NAT 和防火墙(FW)之后,这就要求 VoIP、P2P 等通信软件必须具备穿越 NAT/FW 的能力,才能实现端到端的数据传输。现有的 STUN、TURN、ICE 等方案虽然可以穿越大部分 NAT/FW,但对有端口或协议限制的防火墙却无能为力。本文提出了将 ICE 和 HTTP Tunnel 技术相结合的穿越方案,设计并实现了可跨平台运行的 HTTP Tunnel 模块来配合 ICE 进行穿越。实验证明,在防火墙只开放 80 端口这种苛刻的条件下,该 HTTP Tunnel 模块也能够成功穿越大部分防火墙,实现数据流的稳定传输。

关键词　防火墙;NAT;HTTP 隧道;ICE

1　引言

如今,采用 NAT 或 FW(firewall)接入 Internet 已经是公司、高校和政府机关的主要上网方式。通过在局域网出口处设置 NAT 和防火墙,一方面可以节省 IP 地址资源,另一方面可以保证局域网内部的安全并控制内部人员的访问权限。但是,NAT/FW 的存在同时也成了普通网络通信的障碍,它要求通信软件必须具备穿越 NAT/FW 的能力,才能实现端到端的数据传输。

目前,已有不少方案被用于解决穿越 NAT/FW 的问题,应用比较广泛的有 STUN (simple traversal of UDP through address translators)[1]、TURN(traversal using relay NAT)[2]、ICE (interactive connectivity establishment)[3]等。这些方法都是针对地址映射以及防火墙对外来数据包的过滤规则提出的解决方案,其中,STUN 是通过从公网上的 STUN 服务器获得内网主机在 NAT 的映射端口,使通信双方直接使用映射后的地址和端口进行通信,这种方式只支持 UDP,无法穿越对称型 NAT。TURN 则是采用 relay 的方式穿越 NAT,即所有报文都需要经过 TURN 服务器转发,支持 TCP 和 UDP,可以穿越对称、非对称的 NAT,这种方式使 TURN 服务器成为网络的瓶颈,并且会造成延时和丢包。ICE 则是一种交互式连接建立方式,它综合应用了 STUN 和 TURN 两种方式,使客户端能够根据所处环境选择最优的穿越路径,以弥补单独使用 STUN 或 TURN 所带来的缺陷,并尽可能地减少包的延时。

然而,当防火墙对外出的通信端口或协议进行限制时,上述方法都将无能为力。由于防火墙一般都会允许 HTTP 通信,因此,采用 HTTP Tunnel 技术是突破防火墙限制的主要途径。本文通过分析防火墙的包过滤规则,提出了将 ICE 和 HTTP Tunnel 技术相结合的穿越方案,设计并实现了可以跨平台运行(Windows 和 Linux)的 HTTP Tunnel 服务器和客户端模块。

2　ICE 和 HTTP Tunnel 技术简介

2.1　交互式连接建立 ICE

ICE 所提供的是一种框架,使各种 NAT/FW 穿越技术

* 国家自然科学基金资助项目,福建省自然科学基金资助项目,国家人事部留学人员创业基金资助项目

可以实现统一。通过综合应用 STUN、TURN 等协议，以及对候选地址进行连通性测试，使其能够最大限度地提供一种最优的路径选择方案，实现最低延迟的通信效果。ICE 的算法流程可概括为 3 个步骤。

(1)收集候选传输地址

客户端首先收集自己的本地地址，然后向 STUN 服务器询问自己的 NAT 映射地址，向 TURN 服务器请求对外中继地址。

(2)候选地址交换

将收集到的传输地址按优先级从高到低排序后，通过信令信道传递给对方。双方都将本地地址集和远程地址集进行配对，构成候选地址对。

(3)连通性测试

依次对每个候选地址对进行连通性测试，测试方法为从一个本地地址向远程地址发送 STUN Request，如果顺利收到 STUN Response 则表示该地址对是连通的，可用来传输数据。

参考文献[4]和[5]介绍了 ICE 方案用于 SIP 信令穿越 NAT 的详细例子。

2.2 HTTP Tunnel 技术

HTTP Tunnel 技术就是指防火墙后的用户将通信的数据包按照 HTTP 的格式进行封装，模拟 HTTP 的通信行为，将数据包发送给外网服务器，由服务器将数据包还原后再转发到目的地址的通信技术。实现 HTTP Tunnel 的方式主要有两种：利用 HTTP 的 80 端口和 HTTP 的 443 端口。对于 HTTP 通信来说，防火墙可以对 HTTP 报文进行解码，通过检查 HTTP 头或者根据访问规则来执行过滤，因此对于经过 80 端口的数据必须严格符合 HTTP 的规定才能顺利通过。但是对于 HTTP 通信来说，由于传输的报文进行了 SSL 加密，防火墙无法对数据进行检查，因此，所有数据报都可以直接穿过。

虽然通过 443 端口可以穿透防火墙，但是基于安全的原因，某些企业会关闭 443 端口[6]，因此，本文主要讨论通过 HTTP 的 80 端口建立隧道的技术。

3 基于 ICE 与 HTTP Tunnel 技术的 NAT/FW 穿越实现思路

由于 ICE 本身没有考虑防火墙可能对外出通信进行限制，因此它只能解决 NAT 映射造成的通信障碍问题，但无法解决防火墙对外出通信的限制。本文提出了采用 HTTP Tunnel 配合 ICE 进行穿越的方案。如图 1 所示，在原有 ICE 的基础上增加了 HTTP Tunnel 模块，图中对网络环境进行如下假设：ICE Server 和 Tunnel Server 均位于公网 IP 地址为 59.77.17.120 的主机上，ICE 中的 STUN Server 和 TURN Server 的默认服务端口分别为 3478 和 5556，Tunnel Server 的服务端口为 80。

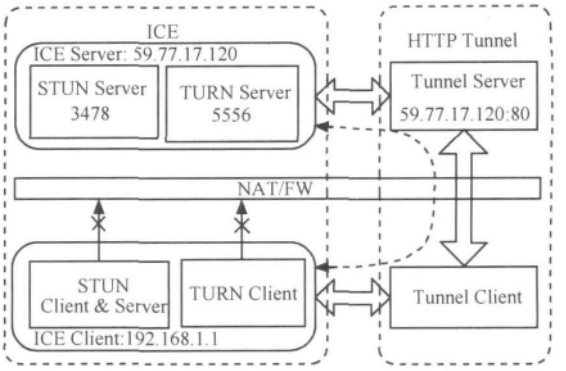

图 1 加入了 HTTP Tunnel 模块的 ICE 系统结构

基于 ICE 与 HTTP Tunnel 技术的穿越方案可具体描述如下。

首先，客户端进行网络环境的探测，以确定防火墙是否禁止了某些协议或端口的通信。具体为：STUN Client 和 TURN Client 分别以 UDP 向公网上的 STUN Server 和 TURN Server 的服务端口发出探测请求包，如果成功收到响应，则说明该防火墙允许该端口上的 UDP 通信，否则为禁止。然后，STUN Client、TURN Client 和 Tunnel Client 再分别向 STUN Server、TURN Server、Tunnel Server 的服务端口发起 TCP 连接，并用 TCP 发送一个探测请求包，如果连接成功并且收到响应，则说明防火墙允许该端口上的 TCP 通信，否则为禁止。

经过上述过程，如果探测结果为 STUN 和 TURN 服务端口上的 TCP 通信和 UDP 通信均被允许，则 STUN 和 TURN 模块均可正常工作，接下来的穿越过程将完全按照参考文献[3] ICE 方式所描述的流程进行。

如果在 STUN 和 TURN 服务端口上只有 TCP 通信被允许，则 STUN 方式无法正常工作，TURN 由于支持 TCP 因此可以正常工作，接下来的穿越过程将完全按照参考文献[2] TURN 方式所描述的流程进行，不采取 ICE 的模式（因为 ICE 模式的连通性检查过程需要 STUN 的配合）。

如果只有 80 端口上的 TCP 通信被允许，则启动 HTTP Tunnel 模块，在 TURN Client 与 TURN Server 之间构造 HTTP 隧道，使 TURN Client 和 TURN Server 可以通过

HTTP 隧道实现无阻碍的数据通信。整个穿越过程的控制(如身份验证、端口分配等)仍然由 TURN 模块负责,HTTP Tunnel 只负责转发数据,这样不仅充分利用了已有的模块,还实现了功能块的分离,使 HTTP Tunnel 还可以被用来转发其他需要穿过防火墙的数据,如 SIP 信令、H.323 信令等。

最后,如果 UDP 和 TCP 探测全部失败,那么整个穿越过程以失败告终。

4 HTTP Tunnel 模块的设计与实现

基于 ICE 与 HTTP Tunnel 技术的 NAT/FW 穿越方案的关键是设计出一个 HTTP Tunnel 模块,使局域网内部的数据能够成功穿越防火墙的限制。本文先就防火墙对 HTTP 报文的过滤规则进行了测试分析,然后根据分析结果设计并实现了 HTTP Tunnel 模块。

4.1 防火墙对 HTTP 报文的过滤规则分析

由于不同防火墙的检查机制不同,要设计出有效的 HTTP Tunnel 模块,必须综合考虑各种防火墙对 HTTP 报文的检查规则。本文通过对各种常用防火墙(如 ISA、天网等)的测试分析,总结出以下规则。

(1)经过 80 端口的数据必须符合 HTTP 协议[5]对 HTTP 报文的格式规定,特别是实际发送的报文长度必须和 HTTP 包头中的 Content-Length 的值相符。经过实验,最精简的 POST 和 GET 请求与响应报文如表 1 所示。

表 1 最精简的 HTTP 报文

POST 请求	POST/HTTP/1.1\r\nContent-Length: 消息长度 \r\n\r\n 消息体
POST 响应	HTTP/1.1 200 OK
GET 请求	GET/HTTP/1.1
GET 响应	HTTP/1.1 200 OK\r\nContent-Length: 消息长度 \r\n\r\n 消息体

(2)客户端发出的 HTTP 报文可以设置 Connection 属性为 Keep-Alive 要求保持连接,但是如果一条连接上长时间没有数据通信,那么即使申请了 Keep-Alive,也有可能被防火墙强制断开。

(3)在同一条 HTTP 连接上,可以进行多次请求和应答,但是 ISA 防火墙要求新的请求必须在旧的请求被响应完毕后才能发送出去,如果上一个请求未收到响应而发送新的请求,防火墙会将新的请求拦下来,直到上一个请求的响应到来再放行。同理,没有请求直接发送回复时也会被防火墙拦截直到请求到来才放行。这种情况下连接不会断开。

(4)ISA 作为一个透明的代理存在于 Client 与 Server 之间。经过测试发现,如果在同一个 IP 地址上透过 ISA 对同一个 Server 发起多条连接请求(假设为 m),那么实际情况是该 Client 与 ISA 建立了 m 条连接,而 ISA 与外网 Server 之间刚开始只建立了 1 条连接,在连接建立之后的数据通信中,如果由 Client 的 m 条连接上发来的请求完全可以由 ISA 与 Server 之间的 1 条连接进行代理的话,那么连接数将不会增加,只有当请求业务比较繁忙时,ISA 才会增加与 Server 之间的连接数,最多增加到 m 条连接。

4.2 隧道模块的设计与实现

根据以上防火墙对 HTTP 报文过滤规则的分析结果,本文设计并实现了 HTTP 隧道模块,包括 Tunnel Client 和 Tunnel Server 两个部分。其中,Tunnel Client 主要为应用程序提供类似 TCP 的接口,包括 tunnel_connect、tunnel_disconnect、tunnel_send、tunnel_recv 等。而 Tunnel Server 则相当于一个中间代理,控制隧道为 Tunnel Client 服务,进行数据的中继。

根据规则(1),客户端可以在发送 POST 请求时,设置 Connection 属性为 Keep-Alive,并将报文中的 Content-Length 属性设置成一个很大的值(本文设为 20 000 000),接下来就可以在该连接上发送不含 HTTP 头的数据,直到所发数据的总量达到 Content-Length 字节后,才需要再发送一次新 POST 请求。同理,可以将 GET 响应的报文中的 Content-Length 设置成很大,用来接收报文。

根据规则(2),如果 Content-Length 设置成很大,那么必须保证隧道中不能太久没有数据通信,可以通过发送维持包的方法来维持隧道不被防火墙断开。

根据规则(3),同一条 TCP 连接同时只能处理一个请求,要实现数据的同时收发,必须建立两条连接,一条用于发送数据,一条用于接收数据,分别以 POST 方法和 GET 方法建立隧道。由于一个 Client 将拥有多条连接,必须由 Server 端为每个 Client 分配一个惟一的 ID 号,用于统一标识属于同一个 Client 的连接。在之后的每个数据包的包头中都加入该 ID 标志,这样 Server 端可以通过 ID 来识别所接收到的数据包的宿主。

根据规则(4),同一个 Client 从不同的两条连接上发送数据,在 Server 端可能在同一条连接上被收到,这就要求 Server 端必须综合考虑各种可能收到的数据,对所有连接上的数据进行统一的处理。

另外,由于 TCP 是以流的形式收发数据,没有包边界,

会造成粘包现象，因此还需在包头中记录包长度，用于Server端进行分包处理。

根据以上分析，将报文包头格式定义如表2所示。

表2 HTTP Tunnel数据包的包头格式

版本号	包类型	客户ID	包长度	保留
1 byte	1 byte	2 byte	2 byte	2 byte

包类型的定义：0，connet请求包；1，disconnect请求包；2，用户数据包；3，隧道维持包；4，探测包。

HTTP Tunnel模块的整体结构如图2所示。

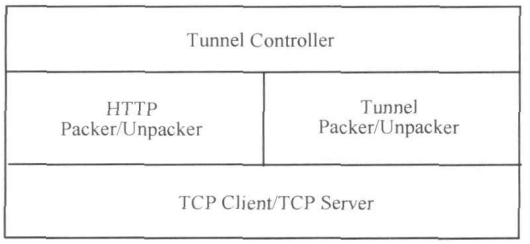

图2 HTTP Tunnel的整体结构

其中，底层的TCP模块以效率最高的Windows IOCP模型和Linux EPOLL模型实现统一的接口，屏蔽了底层操作系统的特殊性，使HTTP Tunnel可以跨平台运行。HTTP Packer、Unpacker负责HTTP包的封装和解析，Tunnel Packer/Unpacker负责表2所示的Tunnel数据包头的封装和解析。最上层的Tunnel Controller则调用底层的模块来控制隧道的建立、销毁以及数据转发。

隧道的建立过程如图3所示。

当应用程序调用tunnel_connect对一个IP地址和端口发起连接时，Tunnel Client先向Tunnel Server发起TCP连接，并利用HTTP的POST方法向Tunnel Server发送连接请求报文（对应表2中包类型为0），报文中包含所要连接的目标Server的地址和端口。Tunnel Server收到请求报文后先判断是否是该TCP连接上的第一个请求，若是则解析HTTP包，并向报文中的目标地址发起TCP连接，若连接成功，则为Client分配一个惟一的ID（用于惟一标识一个客户端），并通过POST响应将该ID返回给客户端，同时在本地建立一个绑定记录，以绑定客户端与目标服务器。绑定记录表的格式如表3所示。建表后将ClientID和DestSocket填入表中，PostSocket和GetSocket将在接下来的隧道建立过程填入。

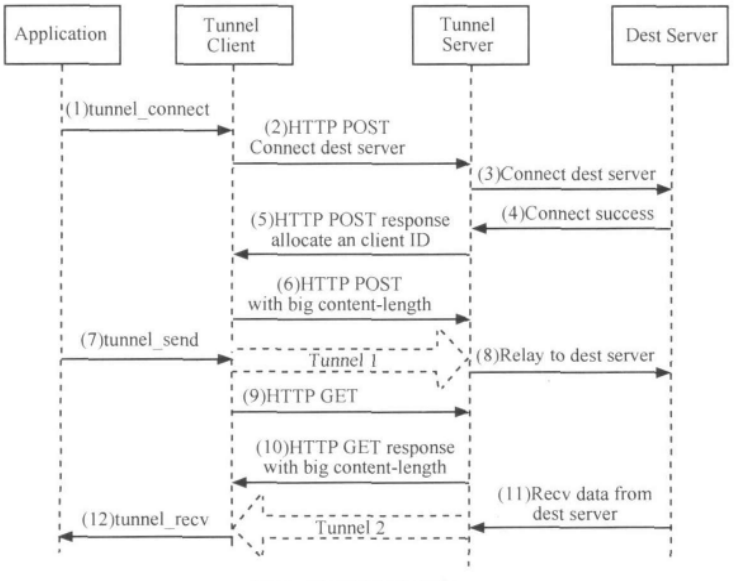

图3 建立隧道的时序

表3 Tunnel Server上的绑定记录表

ClientID	PostSocket	GetSocket	DestSocket
用于惟一标识一个客户端	采用POST方法建立起的隧道所使用的套接字	采用GET方法建立起的隧道所使用的套接字	与目标服务器连接所使用的套接字

Tunnel Client收到连接成功的响应后，在同一条TCP连接上再发送POST请求，但是将Content-Length属性设置成很大，Tunnel Server收到该请求后，首先解析包头中的Content-Length属性值并保存，然后启动该连接上的接收数据计数器，以后每收到一个字节的数据，就将所保存的Content-Length减1。经过上述过程后，第一条隧道被成功建立，Tunnel Server将该TCP连接所用的套接字填入绑定记录表中的PostSocket域，应用程序接下来可以在这条隧道上对目标Server发送数据。当所保存的Content-Length值减至零后，Tunnel Server必须向Tunnel Client发送POST响应，而后Client发送新的POST请求以维持隧道。

为了再建立一条用于接收数据的隧道，Tunnel Client向Tunnel Server请求一条新的TCP连接，然后在该连接上发送GET请求，请求中带有已分配的ClientID。Tunnel Server收到数据包后，判断如果是GET请求，则从报文中解析出Client ID，根据此ID找到Client与目标Server之间的绑定表，将本条TCP连接所用的套接字填入绑定记录表中的GetSocket域，同时向Tunnel Client回复GET响应，在响应报文中将Content-Length属性设置成很大。由此，第二条隧道形成，Tunnel Server从目标Server上接收到的

数据可以通过此隧道转发给客户端。Tunnel Client 每收到一个字节的数据也对 Content-Length 减 1，当 Content-Length 减为零时，向 Tunnel Server 发送新的 GET 请求以维持隧道。

5 HTTP Tunnel 模块的防火墙穿透测试

本文利用所实现的 HTTP Tunnel 对 ISA、SQUID、Check Point 企业防火墙和卡巴斯基、瑞星、天网等个人防火墙进行了穿透测试，测试结果如表 4 所示。

表 4 HTTP Tunnel 防火墙穿透测试结果

防火墙 / 安全等级	天网	瑞星	卡巴斯基	ISA	SQUID	Check Point
低级	√	√	√	√	√	√
中级	√	√	不稳定			
高级	√	√	不稳定			

测试结果表明，只有在穿过卡巴斯基的时候得到的数据不太稳定，其原因在于，卡巴斯基在所收报文的字节数未达到 HTTP 包头中的 Content-Length 的值时，会将收到的一部分数据暂时扣留，直到下一批数据到来再把扣留的数据送出，因此出现数据可穿透但是断断续续的现象。对于其他防火墙，该 HTTP Tunnel 全部可以成功穿越。

6 结束语

本文提出了 ICE 与 HTTP Tunnel 技术相结合的 NAT/FW 穿透方案，只需要对原有的 ICE 模块进行少量改动，就可以解决 ICE 无法穿越防火墙对外出通信的限制这一问题。所实现的 HTTP Tunnel 模块采用分层、模块化的设计思想，具有跨平台运行的特性，并且可以成功穿透大部分防火墙。由于客户端可能采用代理上网，因此，下一步的工作，可以将对 SOCKS 系列代理以及 HTTP 代理的穿透以类似的方式加入到本框架中，以适应各种不同的网络环境。

参考文献

1 Rosenberg J. STUN-Simple traversal of user datagram protocol. RFC3489, 2003

2 Rosenberg J. Traversal using relay NAT (TURN). draft-rosenberg-midcom-turn-08,2005

3 Rosenberg J. Interactive connectivity establishment (ICE):a protocol for network address translator (NAT) traversal for offer/answer protocols ICE.draft-ietf-mmusic-ice-19, 2007

4 Rosenberg J. Examples of network address translation (NAT) and firewall traversal for the session initial protocol (SIP). draft-rosenberg-sipping-nat-scenarios-02

5 刘杨，姜琳颖，张色珍等.基于 ICE 方式的综合性 SIP NAT 解决方案设计与实现.小型微型计算机系统，2006，27(5)：825~827

6 黄伟峰. HTTP Tunnel 技术在 VoIP 系统中的实现. 微型电脑应用，2004，20(2)：43~45

7 Fielding R. Hypertext transfer protocol-HTTP/1.1. RFC2068,1997

Methodology for NAT/FW Traversal Using ICE and HTTP Tunnel

Chen Shifeng, Chen Yi, Xu Xiaofeng, Guo Donghui

(Xiamen University, Xiamen 361005, China)

Abstract In modern networks, a growing number of computers are deployed behind the NAT and firewall, which requires communication systems such as VoIP and P2P which want to transmit data from point to point must be able to traverse NAT/firewall. The existing methods of STUN, TURN and ICE can be helpful in dealing with most kind of NAT/firewall, but helpless when the firewall also restricts the communication port or protocol. The article gives a new solution by syncretizing the ICE and HTTP tunnel technology, and realizes the module of HTTP tunnel. The result of the traversal test shows that the HTTP tunnel module works perfectly with most of firewall.

Key words firewall, NAT, HTTP tunnel, ICE

(收稿日期：2008-10-08)