第32卷 第23期 计 算 机 工 程 2006年12月
Vol.32 No.23 Computer Engineering December 2006
·网络与通信· 文章编号：1000—3428(2006)23—0097—03 文献标识码：A 中图分类号：TP393.02

支持NAT用户的IPv6隧道代理设计和实现

吴贤国[1,2]，刘 敏[1,2]
(1. 中国科学院计算技术研究所，北京 100800；2. 中国科学院研究生院，北京 100080)

摘 要：IPv6隧道代理面向分散用户，是IPv4/IPv6过渡技术的重要组成部分。但IPv6隧道代理不支持NAT用户，而在世界各地尤其是在中国，NAT用户广泛存在。该文针对NAT问题对隧道代理机制进行了修改，允许隧道主体上存在任何类型和任意数量的NAT。在Linux平台上实现了修改后的隧道代理系统，并对隧道服务器的性能进行了测试。

关键词：隧道代理；IPv6；过渡；NAT

Design and Implementation of the IPv6 Tunnel Broker to Support NAT Users

WU Xianguo[1,2], LIU Min[1,2]
(1. Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100800;
2. Graduate School of Chinese Academy of Sciences, Beijing 100080)

【Abstract】The IPv6 tunnel broker is designed for scatted users and is an important mechanism of transition from IPv4 to IPv6. However, the mechanism does not support NAT users, which generally exist in the world widely, especially in China. The mechanism is modified to allow NAT users to connect with IPv6 networks. The system is implemented on Linux platforms and the performance of the tunnel server is tested.

【Key words】Tunnel broker; IPv6; Transition; NAT

IPv6隧道代理[1]是IETF制定的一种IPv4/IPv6过渡机制。它的主要功能是为用户提供一种简化的隧道配置方法，由代理服务器负责响应用户的接入请求，自动地在用户和隧道服务器之间建立IPv6-in-IPv4隧道，从而实现用户和IPv6网络的互联。

IPv6隧道代理机制的主要缺点是不支持NAT用户[2]。主要原因有两点：(1)隧道代理采用IPv6-in-IPv4封装方式，但目前绝大部分NAT都不支持这类数据包(协议类型为41)的转发；(2)即使支持，但如果NAT类型是对称的，那么用户的隧道参数即映射地址(用户数据包经NAT转换后的源IP地址)就不再固定，它将随数据包目的地的变化而变化。隧道服务器根据隧道参数配置其与用户之间的隧道，正确的参数应是数据包目的地为隧道服务器时的映射地址，而隧道服务器从代理服务器获得隧道参数，也就是数据包目的地为代理本身时的映射地址，这使得隧道服务器为用户创建的隧道是无效的。基于上述原因，隧道代理机制要求用户必须具有全局IP地址。

本文对隧道代理机制做了改进，由隧道服务器负责创建并管理IPv6-in-UDP隧道，解决了不支持NAT用户的问题。在Linux平台上实现隧道代理系统并对隧道服务器的性能进行了测试。

1 隧道代理机制的改进

1.1 系统描述

改进后的隧道代理系统结构如图1所示，整个系统由客户端、代理服务器、隧道服务器和IPv6节点组成。客户端和隧道服务器之间通过IPv6-in-UDP隧道传输IPv6数据流[3]，几乎所有的NAT设备都支持UDP报文的转发，因此允许隧道主体上存在任意数量的NAT设备。代理服务器不再负责IPv6地址的分配以及隧道的创建和管理，它的主要功能是动态监测隧道服务器的运行状况，然后根据这些信息为用户指定合适的隧道服务器。隧道服务器的主要功能包括为客户端分配IPv6地址，创建并管理和不同客户端之间的隧道，转发客户端和IPv6网络之间的数据流。这样，隧道服务器为建立隧道所需的客户端隧道参数(此时为映射地址和映射端口，即用户数据包经NAT转换后的源IPv4地址和源UDP端口)不是从代理服务器获得，而是直接从它和客户端的交互报文中获得，即使NAT类型是对称的，通过这种方式获得的隧道参数仍然正确，确保了客户端和隧道服务器之间隧道的有效性。

客户端接入IPv6网络的过程如下：为了获得隧道服务器的地址，客户端首先向代理服务器发送接入请求；代理服务器选择一个合适的隧道服务器并向客户端返回它的IPv4地址，在这个过程中需要用到隧道服务器的负载信息，因此隧道服务器必须周期性地向代理服务器发送负载信息报文；客户端得到隧道服务器的地址后向其发送IPv6地址请求报文；隧道服务器从该报文中获得客户端的隧道参数，然后向客户端返回IPv6地址响应报文；客户端根据报文内容配置IPv6地址并建立隧道，完成接入过程。在这之后，为了维持NAT的UDP会话关系，客户端需要周期性地向隧道服务器发送Hello报文，Hello报文是没有数据载荷的IPv6-in-UDP数据包，以尽量减少传输该报文对带宽的消耗。

基金项目：国家基金委基金资助项目“异构网络服务质量控制算法研究”(60273021)

作者简介：吴贤国(1978－)，男，博士生，主研方向：网络协议设计和无线网络；刘 敏，博士生、助理研究员

收稿日期：2005-11-17 **E-mail：**xgwu @ict.ac.cn

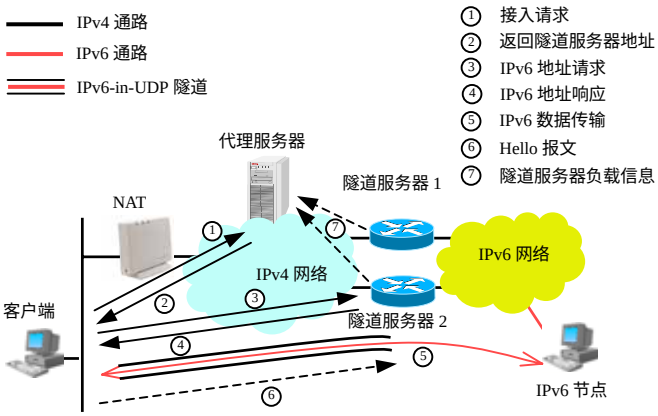

图 1 隧道代理系统结构

客户端接入 IPv6 网络的过程如下：为了获得隧道服务器的地址，客户端首先向代理服务器发送接入请求；代理服务器选择一个合适的隧道服务器并向客户端返回它的 IPv4 地址，在这个过程中需要用到隧道服务器的负载信息，因此隧道服务器必须周期性地向代理服务器发送负载信息报文；客户端得到隧道服务器的地址后向其发送 IPv6 地址请求报文；隧道服务器从该报文中获得客户端的隧道参数，然后向客户端返回 IPv6 地址响应报文；客户端根据报文内容配置 IPv6 地址并建立隧道，完成接入过程。在这之后，为了维持 NAT 的 UDP 会话关系，客户端需要周期性地向隧道服务器发送 Hello 报文，Hello 报文是没有数据载荷的 IPv6-in-UDP 数据包，以尽量减少传输该报文对带宽的消耗。

1.2 自动封装方式

对目的地是客户端的 IPv6 数据包在转发之前进行 UDP 头部和 IPv4 头部的双重封装，是隧道服务器的一个主要功能。隧道服务器通过建立客户端 IPv6 地址和隧道参数之间的映射关系来实现它对 IPv6 数据包的自动封装。

IPv6 地址请求报文是一个数据内容为客户端接口标识符的UDP报文，隧道服务器收到客户端A的地址请求报文后，从中获得它的隧道参数，也就是经过NAT转换后的源IPv4 地址IP_A和源UDP端口号UDP_A，然后根据客户端接口标识符为A构造一个IPv6 地址$IP6_A$，建立$IP6_A$和IP_A、UDP_A之间的映射关系{$IP6_A$：IP_A + UDP_A}。为此，隧道服务器需要维护一张映射表，每增加一个用户，相应地在映射表中增加一个新表项，表示创建一条新的

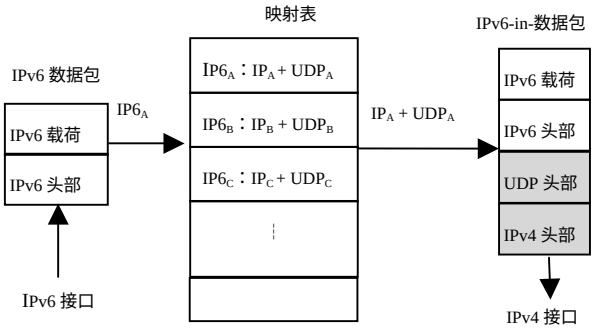

图 2 基于映射关系的自动封装

隧道。从IPv6 接口收到数据包后，通过查表得到IPv4 地址和UDP端口，然后对数据包进行封装，再将封装后的报文从IPv4 接口发送给客户端，如图 2 所示。

和 6to4[4]、ISATAP[5]等基于客户端IPv6 地址来封装的隧道技术不同，采用基于映射关系的自动封装方式，在客户端隧道参数发生改变的情况下，只要重新设置映射关系即可，客户端的IPv6 地址可以保持不变。通过这种方式，隧道服务器可以引入灵活的配址机制，为客户端分配各自需要的IPv6 地址。

1.3 隧道管理

随着用户增多，映射表长度将逐渐增大，隧道服务器的开销也随之增加，这就需要对隧道进行管理，将那些不再使用的隧道删除，使隧道服务器上保存的都是活动的隧道。可采取两种方式来管理隧道：一是客户端程序退出后发送通告，隧道服务器收到通告后删除隧道；二是客户端周期性地告知隧道服务器它的存在，隧道服务器若在一定时间内没有收到告知报文，则删除隧道。

IPv6-in-UDP 隧道是建立在 UDP 会话基础上的，NAT 设备会删除长时间没有数据流的 UDP 会话，为确保隧道有效，客户端需要不时地向隧道服务器发送 Hello 数据包。隧道服务器正好可将 Hello 数据包看作来自客户端的告知报文，因此适合采用第 2 种隧道管理方式。隧道服务器为每个用户设置一个计时器，收到 Hello 数据包后重置计时器，一旦计时器超时，就删除映射表中和此用户对应的映射表项。

2 系统实现和性能测试

2.1 在 Linux 平台上的实现

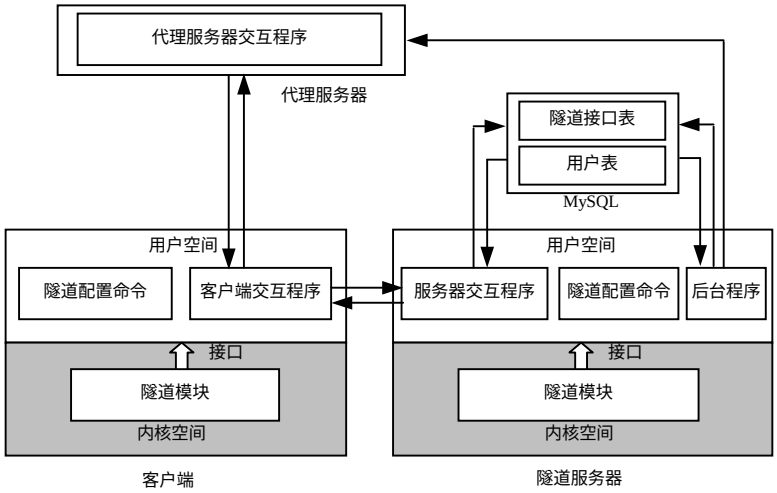

图 3 系统实现框架

实现框架如图 3 所示，整个系统包括客户端软件，代理服务器软件和隧道服务器软件 3 部分。隧道模块是核心模块，它的主要功能是将接收到的 IPv6-in-UDP 数据包解封装后交给上层协议处理，将要转发的 IPv6 数据包封装成 IPv6-in-UDP 数据包后再进行发送。

把隧道作为一个独立的模块放在内核空间实现，一方面，是出于执行封装和解封装效率的考虑；另一方面，封装和解封装是客户端和隧道服务器共同具备的基本功能，将此功能和其它功能分离，隧道模块只向上层的应用程序提供通用的接口，使得应用层功能的具体实现独立于隧道模块的实现，这样的结构有利于提高软件的可扩展性。

隧道在操作系统中以虚拟网络接口(VNI)的形式实现，虽然真正的数据包接收或发送依靠网络适配器完成，但隧道接口在逻辑上具有和网络适配器一样的收发功能。接收函数和

发送函数是隧道模块最主要的两个函数，接收函数的主要工作是修改数据包 skb 的 MAC 层头部指针 mac.raw 和网络层头部指针 nh.raw，设置 protocol 为 ETH_P_IPV6，最后调用 netif_rx 函数；发送函数的主要工作是通过 skb_realloc_headroom 函数增加数据包头部缓存空间，然后重新为传输层头部指针 h.raw 和网络层头部指针 nh.raw 赋值，并设置传输层和网络层头部各个字段的值，最后调用 ip_send 函数。

前面提到，隧道服务器维护一张映射表，以客户端的 IPv6 地址为入口，查表可以得到相应的隧道参数用于封装。在实现中，为每个客户端创建一个虚拟网络接口，将客户端的隧道参数存放在接口指针 net_device->priv 所指向的私有数据区域，然后建立路由，指定目的地为该客户端的所有数据包都交给此接口发送。这样，映射表实际上就是一张路由表，查找隧道参数的过程就是系统查找 IPv6 路由的过程。如图 4，操作系统根据客户端 A 的 IPv6 地址查找路由，确定发送的接口 A，将数据包交给 A 的发送函数，发送函数从 A->priv 指针获得隧道参数，将数据包封装后进行发送。这种实现方式利用系统路由查找功能完成隧道参数的查找，有利于提高查找速率。

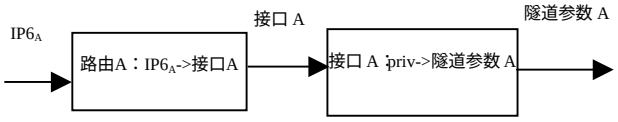

图 4 根据客户端 IPv6 地址获得隧道参数的过程

交互程序是上层应用软件的核心，主要完成客户端 IPv6 地址配置以及隧道的创建工作。隧道服务器交互程序收到客户端的地址请求报文后，按顺序作身份认证、重复地址检测、构建并发送响应报文、创建隧道等处理，在这个过程中，需要和 MySQL 数据库进行通信。数据库维护两张表，一张保存用户名和密码，用于身份认证；另一张保存在线用户的详细信息，如隧道接口名称、状态、IPv6 地址、隧道参数等，用于重复地址检测以及隧道的创建、维护和管理。

隧道服务器收到 Hello 报文(客户端发送周期为 30s)后交给后台程序处理，后台程序根据 Hello 报文的源 IPv6 地址在数据库中查找对应的隧道接口表项，并刷新该表项的状态。后台程序有两个计时器函数，一个用于周期性地检查所有隧道的状态，以删除不再使用的接口及相应的表项；另一个用于统计系统的负载信息并发送给代理服务器。代理服务器交互程序的主要功能是根据这些信息选择一个合适的隧道服务器并向用户返回它的 IPv4 地址。

隧道配置命令是可选的程序，不是客户端和接入服务器软件的必要组成部分。它有建立、删除、修改隧道以及显示隧道信息的功能，主要是方便用户或系统管理员在系统出现故障时使用。

2.2 性能测试

性能测试的对象为隧道服务器软件的应用程序和内核转发模块。隧道服务器运行于中科院计算所，IPv4 地址为 159.226.39.108，可分配的 IPv6 前缀为 2001:250:f007:8::/64，CPU 主频为 667MHz，内存为 256MB，Linux 内核版本为 2.4.18-14，MySQL 版本为 4.1.10。

应用程序的性能以每处理一个地址请求报文和Hello报文消耗的时钟周期来表示，测量次数为 5 000，取平均值作为最终测量值，测量结果如表 1 所示。根据测量结果，可以获得隧道服务器对地址请求报文的最大并发处理能力，比如 667MHz主频最多可以支持约 560 个报文/s(667*10^6/1 200 000)的并发处理；另外，还可以获得隧道服务器因维护隧道付出的开销，比如在线用户数目为 10 000，Hello报文的发送周期为 30s，则因处理Hello报文消耗的CPU资源约占 10%。

表 1 应用程序的性能测试结果

报文类型	消耗的时钟周期
地址请求报文	120 万
Hello 报文	18 万

内核转发模块的性能以吞吐量来表示，包括数据包从客户端转发到另一客户端、从客户端转发到其它 IPv6 节点以及从其它 IPv6 节点转发到客户端的吞吐量。测量每种情况下隧道服务器转发单个数据包消耗的时钟周期，测量次数为 10 000，取平均值作为最终测量值，然后根据 CPU 主频获得吞吐量计算值，结果如表 2 所示。需要注意的是，计算得到的是内核转发模块的吞吐量，实际吞吐量还受到总线速率、网卡速率等硬件因素的影响。

表 2 内核转发模块的性能测试结果

	60bytes		1 472bytes	
	时钟周期	吞吐量	时钟周期	吞吐量
客户端到客户端	10 900	44.2 Mbps	19 000	396 Mbps
客户端到其它 IPv6 节点	9 300	36.9 Mbps	17 500	420 Mbps
其它 IPv6 节点到客户端	9 400	50.4 Mbps	18 200	412 Mbps

3 结论

本文对 IPv6 隧道代理机制进行了修改，由隧道服务器负责 IPv6 地址的分配以及隧道的创建和管理，采用 IPv6-in-UDP 隧道，基于映射关系实现隧道服务器对 IPv6 数据包的封装，并采用客户端发送 Hello 包的方式来维护和管理隧道，解决了隧道代理机制不支持 NAT 用户的问题。在 Linux 平台上实现了修改后的隧道代理系统并已投入到实际网络中运行。

参考文献

1 Durand A, Fasano P, Guardini I, et al. IPv6 Tunnel Broker[S]. RFC 3053, 2001-01.

2 Egevang K, Francis P. The IP Network Address Translator(NAT)[S]. RFC 1631, 1994-05.

3 Huitema C. Teredo: Tunneling IPv6 over UDP Through NATs[Z]. draft-huitema-v6ops-teredo-04.txt, 2005-01.

4 Carpenter B, Moore K. Connection of IPv6 Domain via IPv4 Clouds[S]. RFC 3056, 2001-02.

5 Templin F, Gleeson T, Talwar M, et al. Intra-site Automatic Tunnel Addressing Protocol[Z]. draft-ietf-ngtrans-isatap-24.txt, 2005-01.