第 35 卷 第 6 期 计 算 机 工 程 2009 年 3 月
Vol.35 No.6 Computer Engineering March 2009
·网络与通信· 文章编号：1000—3428(2009)06—0127—03 文献标识码：A 中图分类号：TP393.02

NAT-PT 协议转换网关的设计与实现

李随意
(南京陆军指挥学院军事训练与管理系，南京 210045)

摘 要：介绍 NAT-PT 的技术原理，给出 NAT-PT 协议转换网关的概念和系统结构。根据 RFC 2766 文档和 RFC 2765 文档的相关说明，参照 Linux 内核关于 IPv4/IPv6 网络实现部分的源代码，设计并实现 NAT-PT 网关应用程序。该程序由主体模块、应用层网关模块、网络地址转换模块和协议转换模块组成。

关键词：NAT-PT 协议转换网关；NAT-PT 网关应用程序；Linux 内核

Design and Implementation of NAT-PT Protocol Translation Gateway

LI Sui-yi
(Department of Military Training and Management, Nanjing Army Command College, Nanjing 210045)

【Abstract】 This paper introduces NAT-PT's technology principle, depicts the concept of NAT-PT protocol translation gateway and its system framework. According to RFC 2766 document and RFC 2765 document, and referring to source codes about the realization of IPv4/IPv6 network in Linux kernel, it designs and implements NAT-PT gateway application program, which is composed of main model, application level gateway model, network address translation model and protocol translation model.

【Key words】 NAT-PT protocol translation gateway; NAT-PT gateway application program; Linux kernel

本文研究 NAT-PT 协议转换网关，探讨如何在 IPv4 向 IPv6[1-2]过渡的过程中构建下一代路由器。在不改变原有网络节点的情况下，实现了 IPv4 网络与 IPv6 网络的互通。

1 NAT-PT 技术原理

NAT-PT[3]是用于实现 IPv4 向 IPv6 平滑过渡的协议转换技术。它通常与 SIIT 技术[4]结合以实现地址、协议的转换，与 DNS ALG 技术结合实现 IPv4 和 IPv6 间的域名系统解析或寻址。NAT-PT 由网络地址翻译技术(Network Address Translation, NAT)和协议翻译技术(Protocol Translation, PT)2 个部分组成。

NAT 技术是 IPv4 和 IPv6 间的网络地址转换技术，一般需要设置一个 IPv4 地址池，当 IPv6 地址转换成 IPv4 地址时，从地址池中获取一个 IPv4 地址，并把 IPv6 源地址转换成该 IPv4 地址，提取 IPv6 目的地址中的主机地址部分作为 IPv4 目的地址，并重新计算包头的校验和。把该地址转换记录在地址映射表中。将 IPv4 地址转换成 IPv6 地址时，可以根据地址映射表中的地址转换记录来实现。

NAT 技术存在局限性，当 IPv4 地址池的地址使用完后，其他 IPv6 节点不能与网络外部的 IPv4 节点建立会话连接。因此，研究者提出 NAPT(Network Address Port Translation)技术，它允许多个 IPv6 节点通过一个 IPv4 地址和 IPv4 节点进行通信，IPv6 节点的 TCP/UDP 端口被转换为合法 IPv4 地址的 TCP/UDP 端口。NAPT 解决了 NAT 无法解决的问题，当预留的 IPv4 地址耗费完毕时，它能够提供 6.3×10^4 个 TCP 或 UDP 会话。但该方法只适用于有端口号的协议类型(如 TCP, UDP 协议)。

PT 技术是 IPv4 和 IPv6 间的协议转换技术。它主要根据 IPv4 和 IPv6 在语义上的不同对 IP 包头的对应字段进行转换，构建新的数据包。PT 技术基于 SIIT 协议，在 SIIT 协议中定义了以下 2 种转换形式：

(1)IPv4/IPv6[1]报头之间的转换。在 IPv4 中需要根据报头标志字段判断数据包在何处分片。而在 IPv6 中数据包的分片在主机上完成。因此，如果在数据传输时，IPv4 的数据包发生分片，那么进行 IPv4/IPv6 报头转换处理时，就需要在 IPv6 报头后加上分片扩展头。

(2)ICMPv4/ICMPv6[2]之间的转换。ICMP 协议主要负责为路由器或主机提供路径信息、路由可达信息以及报告传输过程中出现的错误信息。在纯 IPv4 节点和纯 IPv6 节点之间通信时，需要进行 ICMPv4 和 ICMPv6 间的转换。它们之间的转换较简单，主要内容是重新计算校验和以及对包含错误信息 IP 报头的转换。

2 NAT-PT 协议转换网关

NAT-PT 协议转换网关[3]又称为 NAT-PT 翻译网关，一般被配置在边界路由器上，用来连接 IPv4 网络和 IPv6 网络，负责 IPv4 和 IPv6 网络地址和协议的翻译工作，实现纯 IPv4 节点和纯 IPv6 节点之间的网络通信。为了保证 IPv4 网络和 IPv6 网络间的互连互通，需要在边界路由器上安装 IPv4/IPv6 双协议栈。NAT-PT 协议转换网关的网络拓扑如图 1 所示。NAT-PT 协议转换网关的系统结构如图 2 所示，其中，2 个网络设备接口层分别连接 IPv4 网络和 IPv6 网络。在 NAT-PT 协议转换网关中，使用一个全球可路由的 IPv4 地址池。当会话穿越 IPv4/IPv6 边界路由器时，NA(P)T-PT 层根据 IPv4 地址池中的 IPv4 地址进行地址翻译和协议转换，并在 IPv4/IPv6 地址映射表中为每个转换建立相应信息。IPv4/IPv6 协议转换管理层对 IPv4/IPv6 数据包进行转换、路由等控制管理。最上面的 ALG 模块层对数据包负载中包含的 IP 地址进行转换，

作者简介：李随意(1980－)，男，助教、硕士，主研方向：嵌入式系统，计算机网络

收稿日期：2008-09-09 **E-mail：**flying_wind2008@163.com

ALG 与 NAT-PT 配合使用可以提供对多种应用层(如 DNS, FTP)的支持。

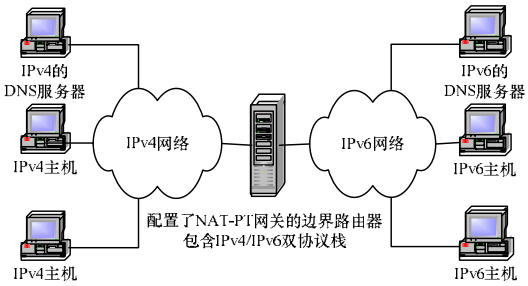

图 1　NAT-PT 协议转换网关的网络拓扑

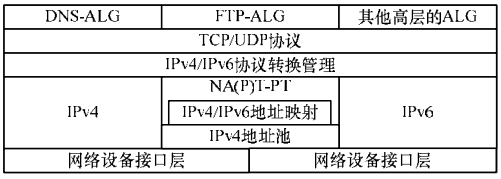

图 2　NAT-PT 协议转换网关的系统结构

网络间的会话必须在一个 NAT-PT 网关上完成，因为它会跟踪它支持的所有会话，并要求进出的数据包通过同一个 NAT-PT 转换器。NAT-PT 网关允许大量应用程序在纯 IPv4 节点和纯 IPv6 节点间进行互操作，是目前实现从 IPv4 向 IPv6 过渡的较理想方法。IPv4 网络使用者或 IPv6 网络使用者通过 NAT-PT 网关发送或接收数据包时，可以像使用本地网络一样，感觉不到网络间的差异。

3　NAT-PT 网关应用程序的设计与实现

根据文献[3-4]的相关说明，参照 Linux 内核关于 IPv4/IPv6 网络实现部分的源代码，设计并实现 NAT-PT 网关应用程序。该程序由以下 4 个模块组成：

(1)主体模块 nat-pt.c 是 NAT-PT 的主体程序，它通过初始化数据包过滤器 packet filter 来接收 IPv4 或 IPv6 的数据包，将经过协议转换处理后的新数据包转发到其他网络上。它还负责处理 ARP(地址解析协议)请求响应以及 Neighbour Discovery(邻居发现协议)的请求响应。其中，对 ARP 和 Neighbour Discovery 的处理是为了获取 IPv4 目的主机和 IPv6 目的主机的以太网地址。

(2)网络地址转换模块 address_pool.c 提供了将 IPv4 地址转换为 IPv6 地址的地址转化功能，它使用一个 IPv4 的地址池，用来在一个会话中实现 IPv4 与 IPv6 的地址映射。

(3)协议转换模块 protocol_translator.c 主要负责将 IPv4 数据包的 IPv4 和 ICMPv4 报头翻译成语义上等价的 IPv6 数据包的 IPv6 和 ICMPv6 报头，反之亦然。

(4)应用层网关模块 alg_manager.c 主要用来判断数据包是否需要交由上层 ALG(应用层网关)进行进一步处理，如果在数据包的载荷中包含 IP 地址应用，那么必须由 ALG 进行特定的地址翻译和协议转换，目前在该模块中主要实现了 DNS ALG 功能。

在上述 4 个模块中，主体模块 nat-pt.c 是核心程序，用来检测网卡接口是否有新的数据包到来，若有则通过地址翻译和协议转换，根据数据包的目的地址将转换后的数据包转发到其他网络上。其他 3 个模块程序实现具体的功能函数，主体模块 nat-pt.c 通过调用它们来实现新数据包的地址翻译和协议转换。

nat-pt.c 是在 Linux 下实现的源程序，需要包含 Linux 内核的一些头文件，如<net/bpf.h>, <net/if.h>, <net/if_arp.h>, <netinet/in.h>, <netinet/if_ether.h>, <netinet/icmp6.h>, <arpa/inet.h>, <net/route.h>等。在 nat-pt_global.h 头文件中建立了一些 nat-pt.c 和其他 3 个模块需要使用的全局数据结构和函数。例如，数据结构 IP_header_info 用来存储 IPv4 和 IPv6 的报头信息，在 nat-pt.c 中被经常使用，其格式如下：

```
 struct IP_header_info     /*数据结构 IP_header_info 的格式*/
{       Network_Source        Network_source;
/*定义数据包的源网络*/
        Network_Destination   Network_destination;
 /*定义数据包的目的网络*/
        unsigned short        Payload_length;
/*IP 数据包中载荷的字节长度*/
        Packet_Type           Payload_type;
/*数据包使用的协议*/
        unsigned char         TTL;
 /*数据包的生存周期*/
        struct
{       struct in_addr *Source_address;
        struct in_addr *Destination_address;
}IPv4;
struct
{       struct in6_addr *Source_address;
        struct in6_addr *Destination_address;
}IPv6;};
```

在 nat-pt_global.h 头文件中声明了宏#define IPv6_PREFIX_HOST “fe80:0000:0000:0000:0000:0000”用来表示 IPv6 网络侧所有主机的地址前缀，这些主机的默认路由器是 NAT-PT 网关。nat-pt.c 主要以双网卡系统实现，一个网卡(eth0)用来连接 IPv4 网络，另一个网卡(eth1)用来连接 IPv6 网络。

nat-pt.c 程序实现 IPv4 与 IPv6 之间地址翻译和协议转换的步骤如下：先为 IP 数据包分配内存缓冲区 pBufferIPin 和 pBufferIPout，为 ARP 和 Neighbour Discovery 数据包分配内存缓冲区 pBufferARP4ReqIn, pBufferARP4RplIn, pBufferND6in 和 pBufferARPout。然后将 IP_header_info 结构变量 pIP_header 作为存储 IPv4 或 IPv6 报头信息的缓冲区。利用 initialise_network_interface()函数初始化由 Berkeley Packet Filter(BPF)接口定义的 5 个网络接口 IPv4, IPv6, ARPv4Req, ARPv4Rpl 和 NDv6，其中，IPv4, ARPv4Req 和 ARPv4Rpl 与 eth0 接口相关联，IPv6 和 ND6 与 eth1 接口相关联。利用宏 IPv6_PREFIX_HOST 生成 IPv6 地址前缀。获取 NAT-PT 主机 eth0 的硬件地址、IPv4 地址和 IPv4 地址池中的地址，然后获取 eth1 的硬件地址和它的 IPv6 地址。再利用 initialise_packet_filter()函数初始化上述 5 个网络接口，清空数据输入缓冲区。初始化地址池和 ALG_Manager 堆，将 NAT-PT 开始工作的时间信息加入数据包统计信息日志文件。

完成上述处理后，开始运行读取、翻译和发送数据包的无限循环体，它是程序的主要语句体，用来把接收到的数据包翻译成不同地址格式的数据包，并根据目的地址的不同，通过不同网络接口转发出去，其主具体步骤如下：

(1)检查 IPv4 接口(即 eth0 接口)是否有 IPv4 数据包到来，若有数据包到来，则读取缓冲区中的所有数据包，并调用协议转换模块 protocol_translator.c 程序中的 PT_Translate_IPv4_Packet()函数，检验和翻译每个 IPv4 数据包。若翻译失败，则丢弃数据包，将 morestatsIP4.bs_discarded 变量的值增 1。

如果翻译后 IPv4 报头的目的地址是 IPv4 网络，那么使用 ARP 协议获得 IPv4 网络中目的主机的物理硬件地址(目的主机的 IP 地址在 IP_header_info 结构中)，并将该 IPv4 数据包通过 IPv4 接口(即 eth0 接口)转发出去。若转发失败，则丢弃数据包，将 morestatsIP4.bs_discarded 变量的值增 1。如果目的地址不是 IPv4 网络，则一定是 IPv6 网络，此时使用 ICMPv6 的邻居发现协议获得 IPv6 网络中目的主机的物理硬件地址(目的主机的 IP 地址在 IP_header_info 结构中)。将此翻译的 IPv4 数据包通过 IPv6 接口(即 eth1 接口)转发出去，如果转发失败，则丢弃数据包，将 morestatsIP6.bs_discarded 变量的值增 1。

(2)检查 IPv6 接口(即 eth1 接口)是否有 IPv6 数据包到来，如果有数据包到来，那么读取缓冲区中的所有数据包，并调用协议转换模块 protocol_translator.c 程序中的 PT_Translate_IPv6_Packet()函数，检验和翻译每个 IPv6 数据包。若翻译失败，则丢弃数据包，将 morestatsIP6.bs_discarded 变量的值增 1。

如果翻译后 IPv6 报头的目的地址是 IPv4 网络，那么使用 ARP 协议获得 IPv4 网络中目的主机的物理硬件地址(目的主机的 IP 地址在 IP_header_info 结构中)，并将此翻译的 IPv6 数据包通过 IPv4 接口(即 eth0 接口)转发出去。如果转发失败，则丢弃数据包，将 morestatsIP4.bs_discarded 变量的值增 1。如果目的地址不是 IPv4 网络，那么一定是 IPv6 网络，此时使用 ICMPv6 的邻居发现协议获得 IPv6 网络中目的主机的物理硬件地址(目的主机的 IP 地址在 IP_header_info 结构中)，并将此翻译的 IPv6 数据包通过 IPv6 接口(即 eth1 接口)转发出去，如果转发失败，则丢弃数据包，将 morestatsIP6.bs_discarded 变量的值增 1。

(3)检查 ARPv4Req 接口(即 eth0 接口)是否有数据包到来，如果有数据包到来，则读取缓冲区中的所有数据包，判断目的 IPv4 主机的 IP 地址是否在 IPv4 地址池中。如果在地址池中，则表明该地址已经被使用，丢弃数据包。如果不在，则构建数据帧和 ARP 报头，交换源地址和目的地址，将 ARP 请求模式改为 ARP 响应模式，并通过 ARPv4Req 接口(即 eth0 接口)将数据包发送出去。

(4)完成上述步骤后，回到第(1)步循环重复执行。

4 结束语

NAT-PT 协议转换网关用于实现 IPv4 向 IPv6 的平稳过渡[5]。未来 IPv6 网络将快速发展，因此，有必要进一步研究 NAT-PT 技术。

参考文献

[1] Hinden R. Internet Protocol Version 6(IPv6) Addressing Architecture[S]. RFC 3513, 2003.

[2] Conta A, Deering S. Internet Control Message Protocol(ICMPv6) for the Internet Protocol Version 6(IPv6) Specification[S]. RFC 2463, 1998.

[3] Tsirtsis G, Srisuresh P. Network Address Translation-protocol Translation(NAT-PT)[S]. RFC 2766, 2000.

[4] Nordmark E. Stateless IP/ICMP Translation Algorithm(SIIT)[S]. RFC 2765, 2000.

[5] 陈晓梅, 王宝生, 赵 峰, 等. 一种新型 IPv6 路由器控制平面的设计与实现[J]. 计算机工程, 2007, 33(22): 119-120, 126.

编辑 陈 晖

(上接第 126 页)

```
    {
      Boolean lookup(in string type_id)raises(NotFound);
    };
    };
```

4 基于 CORBA 的 P/S 使能技术

一个典型的 P/S 通信系统包括事件模型、订阅模型、匹配算法、路由算法和服务质量保证等。基于 CORBA 的 P/S 通信使能技术包括如下几方面：

(1)事件模型又称发布模型，利用 CORBA 的 IDL 数据类型几乎可定义所有已知的数据模型。

(2)订阅模型也称注册模型，OMG Trader Constraint Language 为基于内容的 P/S 订阅模型提供了基础。CORBA 的通告服务的订阅模型就是在 OMG Trader Constraint Language 的基础上进行扩充后得到的一个注册模型。

(3)通知交付模型是指订阅者获取通知的方式，发布者可通过 notify 将事件发布到 P/S 中间件，P/S 中间件将满足订阅者兴趣的通知交付给订阅者，这种方式是 push 方式，在 CORBA 中，通知服务只要有订阅者的某种形式的对象引用就可调用订阅者的操作向其推送内容，很容易实现 push 模型。如果订阅者主动向 P/S 中间件获取订阅，这种方式是 pull 模型，在这种模型中，订阅者在获取通知的同时会指定自己的兴趣，若没有符合自己兴趣的通知，订阅者立即返回(采用非阻塞方式)。在基于 CORBA 的 P/S 系统中，利用基于方法级的异步模型(如国防科大的 AP 模型)很容易实现。混合 push/pull 模型可以在基于 CORBA 的 push 和 pull 模型的基础上实现。

(4)通知传播模型是指底层采用何种通信协议传播通知，IIOP/IGOP 协议可作为一种 P/S 系统传输协议，因为 CORBA 具有平台无关、语言无关、操作系统无关和硬件体系结构无关等特性，所以使用 IIOP/IGOP 作为底层传输协议的 P/S 系统同样可受益于这些特性。网络层可使用 TCP/IP 协议，因为 ORB 负责处理底层网络通信的细节，它可使用不同的底层网络协议(如 TCP/IP 等)，从而避免了订阅端和发布端复杂的网络编程。

5 结束语

基于 CORBA 的方法可以仅用标准 IDL 定义事件，不需要专门的编码包处理事件机制。编译 IDL 后形成 P/S 模式中的发布端(框架)和订阅端(码根)，发布者可以通过通知接口(notify)将事件发布到 P/S 中间件，信息订阅者以订阅的形式指定自己感兴趣的事件，而双方事件的匹配完全交由 P/S 中间件处理，实现了时间、空间和流程的三方解耦。

参考文献

[1] 张志伟. 面向对象异步通信中间件的设计与实现[D]. 长沙: 国防科技大学, 2004.

[2] 马建刚, 黄 涛. 面向大规模分布式计算发部订阅系统核心技术[J]. 软件学报, 2006, 17(1): 134-147.

[3] 郭银章, 徐玉斌, 曾建潮. 分布对象及主流技术比较研究[J]. 太原重型机械学院学报, 2004, 25(2): 142-143.

编辑 顾姣健