

基于 Netfilter 的 NAT 技术及其应用*

乐德广, 郭东辉, 吴伯僖
(厦门大学技术物理研究所, 福建厦门 361005)

摘要: NAT 技术是为了解决 IPv4 网络地址空间的不够, 而提出的一种过渡技术, 并由于其简单、高效的特性, 而得到了广泛的应用。本文详细介绍了 NAT 技术及在 Linux 2.4 内核中基于 Netfilter 框架的 NAT 实现原理, 并结合实验室的网络建设, 给出其在实验室网络建设中应用实例。

关键词: NAT, Netfilter, Iptables

中图分类号: TP393.03

1 引言

随着 Internet 的飞速发展, IP 地址短缺已成为一个十分突出的问题, 使得 Internet 本身的发展遇到了障碍。面对这个问题, 业界提出了各种的解决方案。由于现有基于类的 IPv4 地址体系结构的低效率, 无法从根本上解决 IP 地址短缺的问题, 因此从长期的角度看, IPv6 是最终的解决手段。但是 IPv6 的全面实现还需要一段很长的时间。这样, 就需要有过渡的解决方法, 来暂时解决 IP 地址耗尽的问题。为此人们开发出了许多扩展 IP 地址的方法, 如使用子网掩码^[1]、VLSM (Variable Length Subnet Masks)^[2]、CIDR (Classless Inter-Domain Routing)^[3]等技术以提高 IPv4 地址空间的使用效率; 另一个得到广泛应用的方法就是 NAT (Network Address Translation) 技术^[4], 它允许在内部网络使用保留 IP 地址的主机连入 Internet, 使得 Internet 上的主机数可以大大多于实际 IPv4 地址数。本文首先介绍了 NAT 技术, 并详细阐述了其在 Linux 2.4 中基于 Netfilter 框架的实现原理, 最后给出基于 Netfilter 的 NAT 网关应用实例。

2 NAT 技术

2.1 NAT 技术简介

所谓 NAT 就是在内部网络中使用 IETF (Internet Engineering Task Force) 在 RFC1918 中定义的不可路由的保留 IP 地址: 10.0.0.0~10.255.255.255, 172.16.0.0~172.16.255.255, 192.168.0.0~192.168.255.255。当内部主机要访问 Internet 上的资源时, 在边缘路由器或者网关处通过 NAT 技术将保留地址替换成可路由的全局 IP 地址, 实现内部网络与 Internet 的透明通信。NAT 作为一种过渡解决手段, 是对 IPv4 地址体系结构的补充, 用来减少对全局 IP 地址的需求。

2.2 NAT 的类型

一般 NAT 可分为静态地址转换和动态地址转换。静态地址转换是最简单的一种转换方式, 它在 NAT 转换映射表中为每一个需要转换的内部保留地址创建一个固定的转换条目, 称为简单条目, 映射唯一的全局地址。保留地址与全局地址一一对应。每当内部主机与外界通信时, 保留地址就会转化为对应的全局地址。

* 本文工作得到国家自然科学基金项目 (No: 60076015)、福建省自然科学基金项目 (No: A0010019) 和福建省高新技术项目等经费的资助。作者简介: 乐德广, 男, 1975 年 10 月, 厦门大学物理系博士研究生; 通讯联系人: 郭东辉教授 (博导)。

动态地址转换在边缘路由器或网关中定义了一个全局地址池。当内部主机要与外界进行通信时，如果是第一次与外接连接，则边缘路由器或网关将会动态的从 NAT 地址池中选择一个全局地址对内部保留地址进行地址转换，并记录在 NAT 转换映射表中，每个转换条目在连接建立时动态建立，当连接终止时被释放。这样，网络的灵活性大大增强，并提高全局地址的使用效率。

端口地址转换（NAPT，Network Address Port Translation）是动态地址转换的一种变形，它可以使多个内部主机共享一个全局地址，而通过源和目的 TCP/UDP 端口号来区分 NAT 转换映射表中的转换条目及保留地址，这种包含 IP 地址和 TCP/UDP 端口号的转换条目被称为扩展条目。NAPT 可以使整个内部局域网主机都使用一个全局地址来连入 Internet。这样，进一步提高了全局地址的使用效率。

3 NAT 在 Netfilter 中的实现原理

3.1 Netfilter 结构框架

Netfilter 是 Linux 2.4 内核实现数据包过滤、处理、NAT 等功能的结构框架^[5]，它为 IPv4 网络协议定义了 5 个钩子（Hooks）函数，内核模块可以对一个或多个这样的钩子函数进行注册挂接，使这些钩子函数在数据包流过协议栈时被调用，从而可以修改这些数据包，并向 Netfilter 返回如下值：NF_ACCEPT，继续正常传输数据包；NF_DROP，丢弃该数据包，不再传输；NF_STOLEN，模块接管该数据包，不再继续传输该数据包；NF_REPEAT，再次调用该钩子函数；NF_QUEUE，对该数据包进行排队，这些排队的数据包被传递给用户空间进程进行异步处理，用户进程能检查、修改数据包。Netfilter 的结构框架如图 1 所示：

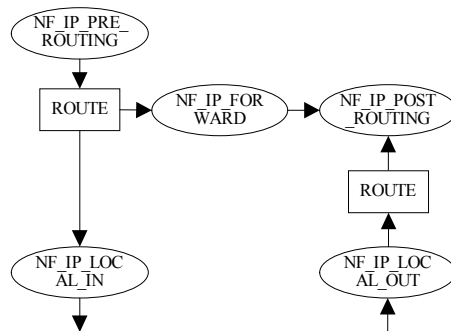


图 1 Netfilter 结构框架示意图

Fig.1 Netfilter Architecture

从图中可以看出 IPv4 的 5 个钩子函数分别是：`NF_IP_PRE_ROUTING`、`NF_IP_LOCAL_IN`、`NF_IP_FORWARD`、`NF_IP_POST_ROUTING`、`NF_IP_LOCAL_OUT`。数据包从左边进入系统，进行 IP 校验以后，数据包经过第一个钩子函数 `NF_IP_PRE_ROUTING` 进行处理；然后就进入路由代码，其决定该数据包是需要转发还是发给本机；若该数据包是发给本机，则该数据经过钩子函数 `NF_IP_LOCAL_IN` 处理后，传递给上层协议；若该数据包应该被转发，则它被 `NF_IP_FORWARD` 处理；经过转发的数据包经过最后一个钩子函数 `NF_IP_POST_ROUTING` 处理以后，再传输到网络上。本地产生的数据经过钩子函数 `NF_IP_LOCAL_OUT` 处理以后，进行路由选择处理，然后经过 `NF_IP_POST_ROUTING` 处理后，发送到网络上。

3.2 Netfilter 的 NAT 实现原理

Netfilter 在 Linux 2.4 内核中提供了一系列的表，每个表由若干链组成，而每个链中可以

由一条或数条规则组成。内核模块可以注册一个新的规则表，并要求数据包流经指定的规则表，用于实现数据包过滤（Filter 表），网络地址转换（NAT 表）及数据包处理（Mangle 表）。在 NAT 表中包含三个链：PREROUTING 链、POSTROUTING 链和 OUTPUT 链^[6]。进行 NAT 时，Netfilter 监听钩子函数：NF_IP_PRE_ROUTING、NF_IP_POST_ROUTING 及 NF_IP_LOCAL_OUT，并根据 NAT 表中的规则对数据包进行地址转换处理。图 2 是 Netfilter 的 NAT 实现原理图。NAT 只对新连接的第一个数据包查询 NAT 表，随后同一个连接的数据包将根据第一个数据包的结果进行同样的转换处理。

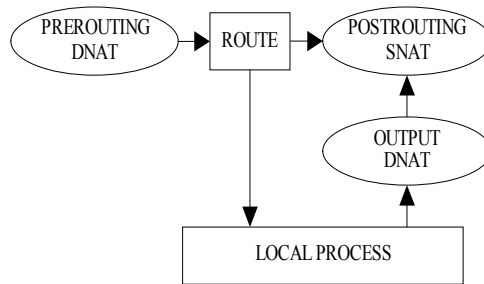


图 2 Netfilter 的 NAT 实现原理
Fig.2 Principle of NAT of Netfilter

Linux 2.4 以前的内核仅仅支持 IP 伪装（Masquerading）等有限的 NAT 功能。Netfilter 则支持 IP 伪装、重定向和端口转发等各种 NAT 技术。Netfilter 将 NAT 分成了两种类型，即源 NAT（SNAT）和目的 NAT（DNAT）。SNAT 是指修改数据包的源地址，SNAT 会在数据包经过 NF_IP_POST_ROUTING 时，根据 NAT 表中 POSTROUTING 链的规则修改数据包的源地址。IP 伪装是一种特殊的 SNAT；DNAT 是指修改数据包的目的地址，DNAT 在数据包经过 NF_IP_LOCAL_OUT 或 NF_IP_PRE_ROUTING 时，根据 NAT 表中 OUTPUT 链或 PREROUTING 链的规则修改数据包目的地址。重定向和端口转发都属于 DNAT。

Linux 2.4 提供了一个简洁、强大的工具 iptables^[7]来实现 NAT 功能。iptables 由两个子系统组成：内核模块和用户接口应用程序。iptables 内核模块能够对输入、输出的 IP 包进行过滤和管理，它是 Linux 2.4 内核中 Netfilter 框架的一个组成部分；iptables 用户接口程序可以添加、插入或删除内核表中的规则，利用 iptables 工具，并用选项“-t nat”来创建、修改 NAT 表。

(1) SNAT

SNAT 用“-j SNAT”选项表示，用“-o”选项指定数据包将要发送到的接口名称，用“--to-source”来指定需要改变的源 IP 地址、IP 地址的范围或者端口号。例如，更改所有来自 192.168.1.0/24 的数据包的源 IP 地址为 210.34.16.59：

```
#iptables -t nat -A POSTROUTING -s 192.168.1.0/24 -o eth0 -j SNAT --to 210.34.16.59
```

有一种 SNAT 的特殊情况是 IP 伪装，在 IP 伪装中不需要明确指定源地址，只要指明正确的网络接口，它会使用包送出的那个接口地址作为源地址。例如：

```
# iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

(2) DNAT

DNAT 用“-j DNAT”选项表示，用“-i”选项指定输入数据包的接口名称，用“--to-destination”选项来指定需要改变的目的 IP 地址、IP 地址的范围或者端口号。例如，更改所有来自 eth1 网络接口的数据包的目的 IP 地址为 210.34.16.59：

```
#iptables -t nat -A PREROUTING -i eth1 -j DNAT --to 210.34.16.59
```

有一种 DNAT 的特殊情况是重定向，它将符合条件的数据包的目的地址改为数据包进

入系统时的网络接口的 IP 地址。例如，把发送到 80 端口的 Web 连接重定向到 Squid 代理端口上：

```
# iptables -t nat -A PREROUTING -i eth1 -p tcp --dport 80 -j REDIRECT --to-port 3128
```

DNAT 的另外一种特殊情况是端口转发，例如，要将 210.34.16.59:8080 变为 192.168.1.1:80：

```
# iptables -A PREROUTING -t nat -p tcp -d 210.34.16.59 --dport 8080 -j DNAT --to 192.168.1.1:80
```

4 基于 Netfilter 的 NAT 路由应用实例

下面我们介绍基于 Netfilter 的 NAT 技术在厦门大学 EDA 实验室的应用实例，厦门大学 EDA 实验室网络于 1997 年建立，经过几年的发展，实验室教师和研究队伍不断壮大，增加了许多网络设备和服 务，使实验室网络规模不断扩大，原有的基于 IPCHAINS 的简单 IP 伪装型网关已不适应实验室网络的发展需求。为此我们在分析最新 Linux 2.4 内核和 Netfilter 框架的基础上对实验室里原有的 NAT 网关作了升级^[8]。现有的实验室网络结构如图 3 所示：

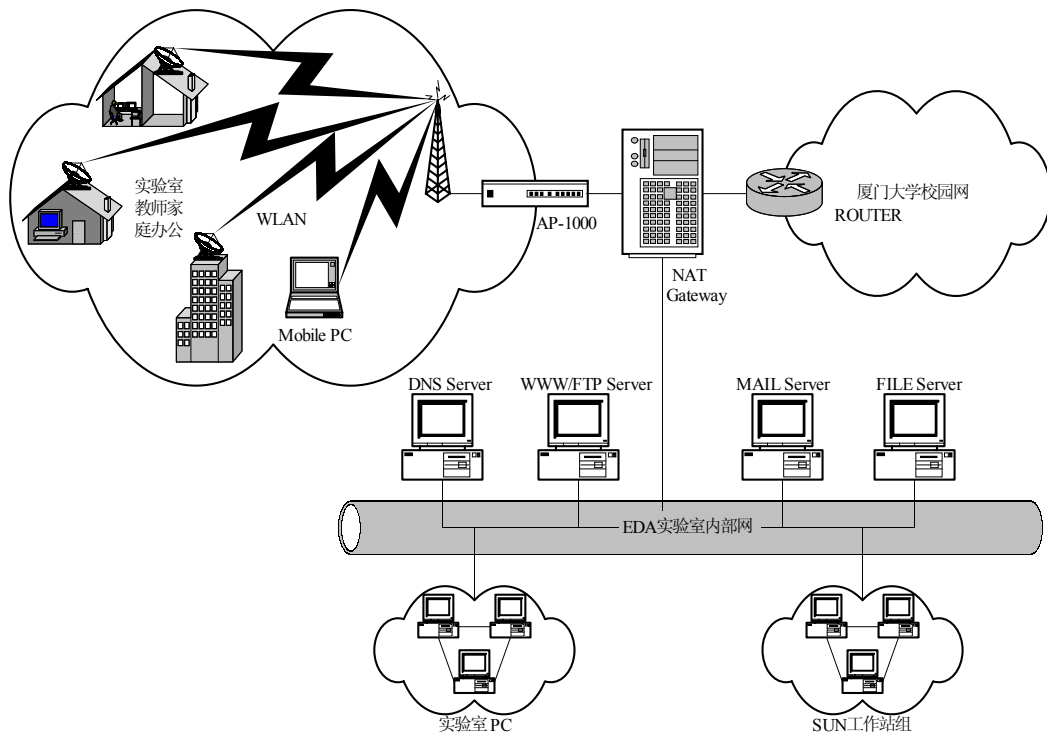


图 3 实验室网络拓扑结构图

Fig.3 Network's Structure of Our Laboratory

图中，由于学校 IP 地址资源有限，实验室目前只分配到五个全局 IP 地址。它们分别是 210.34.16.59~210.34.16.63，其中 210.34.16.59 作为 NAT 网关，其它几个全局 IP 在 NAT 网关中采用 IP 映射的方式对实验室内部的部分 SUN 工作站和服务器进行地址映射，使学校的其他老师 和学生能够使用实验室内的部分资源；实验室内部老师及研究生的工作用 PC 机及各种服务器和工作站均采用保留 IP 地址，网络地址为 192.168.1.0/24，IP 地址范围是 192.168.1.1~192.168.1.254；同时为了方便实验室教师及部分研究生的家庭办公需要，实验室于去年建立了厦门大学第一个无线局域网，在无线局域网中采用保留 IP，网络地址是

10.10.10.0/24。NAT 网关的具体配置如下：

(1) 启动内核的 IP 转发功能，命令如下：

```
# /bin/echo "1" > /proc/sys/net/ipv4/ip_forward
```

(2) 载入 NAT 所需模块，命令如下：

```
# /sbin/insmod /lib/modules/2.4.18-3/kernel/net/ipv4/netfilter/ip_tables.o
# /sbin/insmod /lib/modules/2.4.18-3/kernel/net/ipv4/netfilter/ip_conntrack.o
# /sbin/insmod /lib/modules/2.4.18-3/kernel/net/ipv4/netfilter/iptables_nat.o
# /sbin/insmod /lib/modules/2.4.18-3/kernel/net/ipv4/netfilter/ipt_MASQUERADE.o
```

其中，`ip_tables` 是 `iptables` 主模块，实现 Netfilter 的接口框架；`ip_conntrack` 是连接跟踪模块，是后面两个模块所需要的；`iptables_nat` 是实现 NAT 功能的模块；`ipt_MASQUERADE` 则是实现 IP 伪装。

(3) 进行 IP 伪装，命令如下：

```
# /sbin/iptables -t nat -A POSTROUTING -o eth0 -j MASQUERADE
```

其中，“`-t nat`”指明对 NAT 表操作；“`-A POSTROUTING`”指明添加 POSTROUTING 链中的规则；“`-o eth0 -j MASQUERADE`”表示对所有通过 `eth0` 接口向外发送的包进行 IP 伪装。

(4) IP 映射

实验室的 SUN 工作组中有一台 SUN E250 工作站，IP 地址是 192.168.1.60，为了学校的其他师生能使用该工作站，在 NAT 路由器上进行地址映射，其映射地址是 210.34.16.60，另外也对实验室内的 WEB/FTP 服务器、DNS 服务器以及 MAIL 服务器作了相应的地址映射，使 Internet 上的其他用户也能访问实验室的这些资源。为此我们在 NAT 的外部网络接口 (`eth0`) 上绑定多个全局 IP 地址，命令如下：

```
# ifconfig eth0 add 210.34.16.60 netmask 255.255.255.0
# ifconfig eth0 add 210.34.16.61 netmask 255.255.255.0
# ifconfig eth0 add 210.34.16.62 netmask 255.255.255.0
# ifconfig eth0 add 210.34.16.63 netmask 255.255.255.0
```

然后，对 NAT 网关收到的目的 IP 为 210.34.16.60~210.34.16.63 的所有数据包进行 DNAT，命令如下：

```
# iptables -A PREROUTING -i eth0 -d 210.34.16.60 -j DNAT --to 192.168.1.60
# iptables -A PREROUTING -i eth0 -d 210.34.16.61 -j DNAT --to 192.168.1.61
# iptables -A PREROUTING -i eth0 -d 210.34.16.62 -j DNAT --to 192.168.1.62
# iptables -A PREROUTING -i eth0 -d 210.34.16.63 -j DNAT --to 192.168.1.63
```

最后，对 NAT 网关接收到的源 IP 地址为 192.168.1.60~192.168.1.63 的数据包进行 SNAT，命令如下：

```
# iptables -A POSTROUTING -o eth0 -s 192.168.1.60 -j SNAT --to 210.34.16.60
# iptables -A POSTROUTING -o eth0 -s 192.168.1.61 -j SNAT --to 210.34.16.61
# iptables -A POSTROUTING -o eth0 -s 192.168.1.62 -j SNAT --to 210.34.16.62
# iptables -A POSTROUTING -o eth0 -s 192.168.1.63 -j SNAT --to 210.34.16.63
```

这样，所有目的 IP 为 210.34.16.60~210.34.16.63 的数据包都将分别被转发给 192.168.1.60~192.168.1.63；而所有来自 192.168.1.60~192.168.1.63 的数据包都将分别被伪装成由 210.34.16.60~210.34.16.63，从而实现了 Internet 对实验室内部资源的访问。

5 结束语

Netfilter 作为 Linux 2.4 内核中用于扩展各种网络服务的结构化底层框架, 简洁灵活, 易于使用。经过一年多的稳定运行, 证明现有的基于 Netfilter 的 NAT 比原来的基于 IPCHAINS 的 NAT 转发效率更高、速率更快, 取得了很好的运行效果。此外, 通过 IP 映射功能对外隐藏了内部网络各节点的真实 IP 地址, 阻断了非法人员对内部网络的工作站/服务器的非法入侵和攻击, 增强了网络安全性。

参考文献:

- [1] Mogul J, Postel J. Internet Standard Subnetting Procedure [EB/OL] . <http://www.ietf.org/rfc/rfc950.txt>. 1985-08/2002-10-25.
- [2] Braden R, Postel J. Requirements for Internet Gateways [EB/OL] . <http://www.ietf.org/rfc/rfc1009.txt>. 1987-06/2002-10-25.
- [3] Fuller V, Li T, Yu J, Varadhan K. Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy [EB/OL] . <http://www.ietf.org/rfc/rfc1519.txt>. 1993-09/2002-10-25.
- [4] Egevang K, Francis P. The IP Network Address Translator (NAT) [EB/OL] . <http://www.ietf.org/rfc/rfc1631.txt>. 1994-05/2002-10-25.
- [5] Netherlabs BV, Gregory Maxwell, Remco van Mook, Martijn van Oosterhout, Paul B Schroeder, Jasper Spaans. Linux 2.4 Advanced Routing HOWTO v0.3.0. 2001-09-09.
- [6] Rusty Russell. Linux 2.4 NAT HOWTO v1.0.1. 2000-05-01.
- [7] Oskar Andreasson. iptables Tutorial 1.1.0. Boingworld organization. 2001.
- [8] 郭东辉, 李立峰, 纪安妮, 刘瑞堂. 基于 linux 的网络系统管理及其 Internet 服务的配置 [J] . 厦门大学学报 (自然科学版), 1999, 38 (6): 399—444.

The Netfilter Technology and Its Application for NAT Implementation

LE De-guang , GUO Dong-hui, WU Bo-xi
(Institute of Technical Physics, Xiamen University, 361005)

Abstract: NAT is one of the most important technologies to resolve the lack of IPv4 addresses, and has been widely applied in the extension of LAN for Internet. In this paper, we detail on the NAT technology and its implementation principle based on Netfilter in Linux 2.4 kernel. With the Netfilter technology, we built up a LAN in our laboratory accessed to Internet, and the outline of our LAN is also presented in this paper.

Key words: NAT, Netfilter, Iptables