Linux 下传统代理、IP 伪装与透明代理的设计与实现

A Design and Implementation of Traditional Proxy、IP Masquerade and Transparent Proxy

何文华　梁竟敏 （广东女子职业技术学院网络中心 510300）

摘要：考虑到有限的 IP 地址资源以及网络安全性，在内部网络应用中代理服务器的使用十分普通，Linux 系统由于其卓越的性能及遵循自由软件协议，应用 Linux 作为代理服务器越来越广泛。本文试图通过对 Linux 下的传统代理、IP 伪装与透明代理的实现，分析这几种代理服务器实现方法的特点和应用。

关键词：传统代理 IP 伪装 透明代理

本文所使用的服务器安装 Redhat Linux 9.0(服务器模式)系统，在服务器上安装三块网卡，三块网卡的 IP 地址分别为 eth0：211.66.80.52/24(默认网关为 211.66.80.10)，eth1：192.168.10.1/24，eth2：172.16.1.1/16。其中 211.66.80.52/24 是 Internet 中的有效 IP 的地址，另外两个 IP 则是内部 IP 地址。网络拓扑结构如图 1 所示：

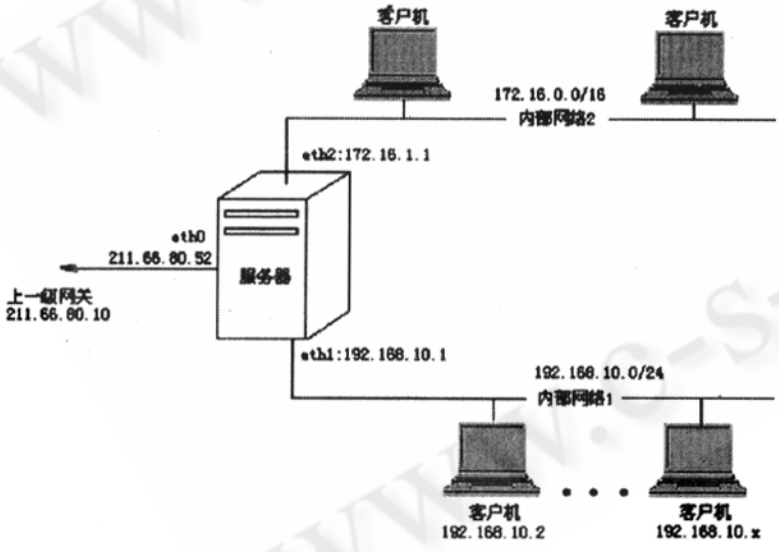

图 1　网络拓扑结构

1　传统代理的实现

在 Linux 中使用最广泛的传统代理是 Squid，Redhat Linux 9.0 中已经自带有了 squid。squid 的服务程序是 squid，位于 /usr/sbin 目录下。控制 squid 服务的程序是 /etc/rc.d/init.d/squid，缺省的配置文件为 /etc/squid/squid.conf。可以用 squid 作单网卡代理，也可以作双网卡代理，应用较普遍的是双网卡代理，即一块网卡与外部网络连接，另一块网卡与内部网络连接，外部网络与内部网络在物理上是隔离的。

通过修改配置文件 /etc/squid/squid.conf 的内容完成 squid 服务的配置，squid.conf 是一个文本文件，用文本编辑器打开 squid.conf 文件对 squid 进行配置。在实现传统代理时，主要配置该文件中的如下几项：

```
http_port 192.168.10.1:3128
http_port 172.16.1.1:3128
cache_mem 32 MB
visible_hostname PROXY_SERVER
maximum_object_size 4 MB
cache_dir ufs /cache 100 16 256
cache_effective_user squid
httpd_accel_host virtual
httpd_accel_port 80
httpd_accel_with_proxy on
httpd_accel_uses_host_header on
http_access allow all
```

配置好 squid.conf 文件后，运行命令 squid -z 建立缓存区，即在指定的缓存目录下建立两重目录。完成后则配置好一台代理服务器，其中网卡 eth1(192.168.10.1)向 192.168.10.0 网络提供代理服务，网卡 eth2 向 172.16.0.0 网络提供代理服务。

配置好 squid.conf 后，启动 squid 代理服务即可。下面是 squid 服务程序的启动方法：

(1) 启动 squid 代理服务

```
# /etc/rc.d/init.d/service squid start
```

(2) 停止 squid 用以下命令

```
# /etc/rc.d/init.d/service squid stop
```

(3) 重新启动 squid 代理服务

/etc/rc.d/init.d/service squid restart

(4) 显示当前的 squid 代理服务的状态

/etc/rc.d/init.d/squid status

(5) 重新输出 squid 代理系统

/etc/rc.d/init.d/squid reload

某些版本的 Linux 不能以"root"用户运行 squid,需要创建特殊的用户。例如创建一个 squid 用户,并通过如下命令创建页面缓冲文件的目录 /cache:

/usr/sbin/useradd - d /cache/ - r - s /dev/null squid

#mkdir /cache

#chown - R squid.squid /cache

启动 squid 代理服务后,只需在客户端浏览器中设置代理服务器的 IP 地址和端口号即可。

2 IP 伪装

实现 IP 伪装仍按图 1 的网络拓扑图结构及网络配置。IP 数据包的结构如图 2 所示,每一个 IP 数据包都有一个源发送地址(Sending Address),简称为源地址,一个目的地址(Destination Address),若在客户机 192.168.10.2 上运行命令:

ping 211.66.80.10

则 IP 数据包的源地址是 192.168.10.2,目的地址是 211.66.80.10,IP 数据包的路由如下:

192.168.10.2 - - >192.168.10.1 - - >211.66.80.52 - - >211.66.80.10

发送的 IP 数据包到达 211.66.80.10 后由 211.66.80.10 响应并回送 IP 数据包,回送 IP 数据包的源 IP 地址是 211.66.80.10,目的 IP 地址是 192.168.10.2,但目的 IP 地址 192.168.10.2 对于网关 211.66.80.10 而言是不可识别的,只能发往它的上一级网关 211.66.80.52,即 IP 数据包不能如期返回。显然这种情况下用内网的 IP 地址是无法访问 Internet 的。

若希望用内部 IP 也能直接访问 Internet,最有效的方法是利用 Linux 的路由功能将服务器配置为一台路由器,并且对 eth0 设置 IP 伪装功能,对于内部网络发送的数据包,路由器将源地址转换为合法的 IP 地址再发送出去,返回的数据包再由路由器(服务器)将目标地址转换成内部网络的 IP 地址。

<table>
<tr><td>Version Number</td><td>Header Length</td><td>Type of Service (服务类型)</td><td colspan="4">Datagram Length (数据包长度)</td></tr>
<tr><td colspan="3">Identification(标识)</td><td>0</td><td>DF</td><td>MF</td><td>Fragment Offset(数据块偏移)</td></tr>
<tr><td>Time to Live (TTL) (生存时间)</td><td colspan="2">Transport Protocol(传输协议)</td><td colspan="4">Header Checksum(标题检查和)</td></tr>
<tr><td colspan="7">Sending Address(发送端地址)</td></tr>
<tr><td colspan="7">Destination Address(目的地地址)</td></tr>
<tr><td colspan="6">Options(选项)</td><td>Padding(填充)</td></tr>
</table>

图 2　IP 数据包结构

从客户机 192.168.10.2 上运行命令:ping 211.66.80.10 时,IP 数据包的路由和伪装过程如图 3 所示。

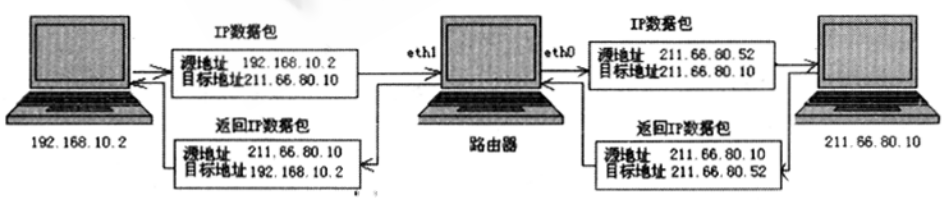

图 3　IP 数据包地址的伪装过程

在路由器内部,eth0 和 eth1 之间 IP 数据包是可以相互转发的,在 eth0 上实现 IP 伪装,实现的过程如下:

(1) 开启 IP 转发

echo 1 > /proc/sys/net/ipv4/ip_forward

(2) 在 eth0 上实现 IP 伪装

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

(3) 运行 route 命令检查路由表，若还未设置默认网关可用以下命令手工添加

route add default gw 211.66.80.10

这样在客户机端将 192.168.10.1 或 172.16.1.1 设置为网关，并配置好 DNS 服务器后，则拥有内网 IP 的客户机可以象拥有 Internet 上的有效 IP 地址一样进行上网游览和收发 E-Mail。

3 透明代理

透明代理是指客户端感觉不到代理的存在，不需要在内部的浏览器中设置代理，只需要设置默认网关，客户端访问外部网络的数据包被发送到默认网关，默认网关运行着一个代理服务器，数据实际上被重定向到代理服务器的代理端口(例如 3128)，由本地的代理服务器向外请求所需数据并拷贝给客户端。

透明代理一方面能利用代理服务器的缓存功能，另一方面理论上它对任何协议都适用。将传统代理和 IP 伪装结合起来就可以实现透明代理。

实现透明代理使用图 1 的网络拓扑结构，在客户端计算机 192.168.10.2 使用浏览器访问站点 202.101.66.100，透明代理的实现过程如下：

(1) 客户端浏览器从端口 1050 向 202.101.66.100 的 Web 站点发出一个 http 连接请求。

(2) 当请求的数据包从客户机的 1050 端口送往 202.101.66.100 的 80 端口时，被重定向到代理服务器的 3128 端口，即客户机使用端口 1050 与代理服务器的 3128 端口建立一个连接(连接 1)。

(3) 代理服务器从端口 1025 与 202.101.66.100 的端口 80 建立另一个连接(连接 2)。

(4) 当代理服务器从 Web 站点通过"连接 2"传来页面后，通过已经建立的"连接 1"把页面拷贝给客户机。

(5) 从客户机的角度来看，连接是 192.168.10.2 的 1050 端口与 202.101.66.100 的 80 端口建立一个连接来传送数据包的，但实际上客户机是和代理服务器建立的"连接 1"，由代理服务器和 202.101.66.100 建立另一个"连接 2"来传送。这两个连接过程对用户是透明的，如图 4 中的虚线所示。

若已经配置好代理服务器，用 iptables 增加如下两条规则就可以实现透明代理：

iptables -t nat -A PREROUTING -i eth1 -p tcp -m tcp --dport 80 -j REDIRECT --to-ports 3128

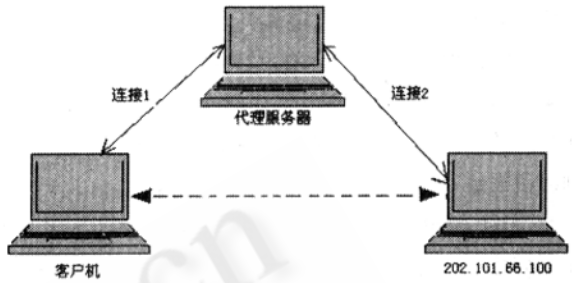

图 4　透明代理示意图

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

其中第一条规则的作用是将所有 80 端口的包转发到 3128 端口。第二条规则的作用是对 eth0 的端口进行欺骗。

4 传统代理、IP 伪装、透明代理的比较

传统代理的优点是对客户端要求很少，只要能连通 squid 服务器即可实现，但具体应用中需要对每一台客户机设置代理服务器的 IP 地址和端口号，而且能为客户机提供缓存服务，以提高访问速度，减轻网络的负担。但传统代理是基于网络应用层的，对一些新的协议不一定能支持，象 ICQ、P2P 等应用软件需要安装代理服务器的客户端软件。

IP 伪装是基于网络层进行数据交换的，理论上可以对任何协议都适用，但不能提供代理服务器的缓存功能。

透明代理兼顾了传统代理服务器和 IP 伪装的优点，一方面能提供代理服务器的缓存功能，另一方面它对任何协议都适合用，象 ICQ、P2P 等应用软件不需要另行安装代理服务器的客户端软件。但需要象 IP 伪装一样将每台客户机的默认网关都设为代理服务器，并在客户端来进行 DNS 解析。

参考文献

1 肖明、胡金柱等，基于 Linux 的路由器和防火墙技术，计算机应用研究，1999，5：P54-56。

2 付炜，Linux 下网络路由器的实现，四川通信技术，2001(31)2：30-32。

3 《Squid Web Proxy Cache》，http://www.squid-cache.org/