多点 UDP 互连的透明代理

Transparent Proxy for Multi-connected UDP

时 曦[1] 苏思妮[2] 陈笑容[1]

(贵州大学理工学院计算机科学系 贵阳550025)[1]

(贵州省移动通信公司技术研发中心 贵阳550001)[2]

摘 要 本文阐述了 UDP 与 NAT 的一个兼容性问题,并提出了解决方案,介绍了一个实施例。

关键词 UDP,内部地址,NAT,代理,透明代理,多点互连

1. 引言

目前 Internet 上使用 UDP 协议进行多点互连的应用程序不断增加,这给广泛使用的 NAT 技术提出了新的课题。

2. 问题的提出

多点互连指多于两个节点之间的相互通讯。使用 UDP 协议的优点是不需要在节点间两两建立连接,即可相互发送数据报。但是当使用 NAT 内部网中的计算机加入到这样的通讯组中时,则产生了如下问题:

假设有 A,B,C,D 四台主机,其中 B,C,D 为拥有正常 Internet 地址的主机,B 拥有 NAT 功能,A 为一台内部网中的主机,它通过 B 与 C,D 互连。如果 A,C,D 进行相互通讯,则它们之间首次互发 UDP 数据报时的连通情况如下:

源 \ 目的	A	C	D
A	—	通	通
C	不通	—	通
D	不通	通	—

C,D 首次发送给 A 的数据报,均不可达。只有当 A 发送了数据报给 C 后,C 发给 A 的数据报才能送达;同样,只有当 A 发送了数据报给 D 后,D 发给 A 的数据报才能送达。如果每一个节点发送数据报给 A 都必须等待 A 首先发送数据报,这就给上层的应用程序的编写带来了很大的困难。那么当 A 发送了数据报给 C 后,如果 D 通过 C 知道了 A,这时 D 发送数据报给 A 能否送达呢?请看如下试验:

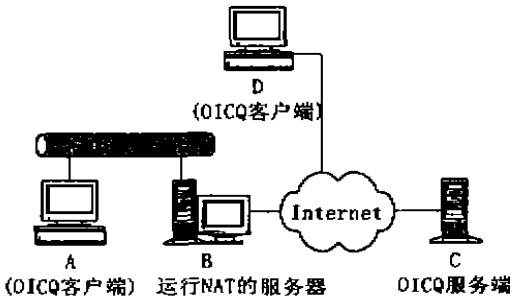

图1

OICQ 是上述使用 UDP 协议进行多点互连的典型的应用,我们拥有上图中的 A, B 和 D。首先在 A 和 D 上,运行了 OICQ 客户端,登录均能成功。显然,A,D 发送给 C 的数据报均能送达,而 C 发送给 A,D 的应答数据报也能送达,由于篇幅所限,这里就不详细列出其通讯过程的细节了。现在,我们在 D 上发送一条消息给 A,OICQ 客户端出现了约20秒的等待,然后成功发送。A 也能收到信息。但是 D 显示:"通过服务器中转"。

事实上,D 未能将 UDP 数据报直接送到 A,而是通过 C"中转"的。

那么等待的20秒在干什么呢?我们在 B 上运行了跟踪程序 tcpdump,摘录了部分输出信息如下:

```
09: 55: 50.973238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 55: 50.973238 > 202.101.67.4 > 202.101.79.62: icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 55: 51.013238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 55: 53.353238 > 202.101.67.4 >
```

时 曦 讲师,主要从事计算机网络方向的研究;苏思妮 助理工程师,从事网络信息系统方面的研究。

202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 55: 53.363238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 55: 53.363238 > 202.101.67.4 > 202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 55: 56.343238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 55: 59.343238 > 202.101.67.4 > 202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 55: 59.373238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 55: 59.373238 > 202.101.67.4 > 202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 55: 59.383238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 56: 16.383238 > 202.101.67.4 > 202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 56: 16.393238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 56: 16.393238 > 202.101.67.4 > 202 101.79.62:icmp: 202.101 67.4 udp port 61551 unreachable [tos 0xc0]
09: 56: 16.403238 < 202.101.79.62.4000 > 202.101.67.4.61551: udp 68
09: 56: 16.423238 > 202.101.67.4 > 202.101.79.62:icmp: 202.101.67.4 udp port 61551 unreachable [tos 0xc0]
09: 56: 26.683238 > 202.101.67.4.61551 > 61.144.238.155.8000: udp 36
09: 56: 26.953238 < 61.144.238.155.8000 > 202.101.67.4.61551: udp 32 (DF)

其中B的地址为202.101.67.4，D的地址为202.101.79.62，D在约26秒时间内先后多次发出了到B的UDP数据报："202.101.79.62.4000 > 202.101.67.4.61551: udp 68"，而收到的应答是"icmp: 202.101.67.4 udp port 61551 unreachable"，这是ICMP目的端口不可达报文。

在这里，D认为B就是A，这是NAT的特点，而B的61551端口只接受来自C的信息，对其他任何主机发来的信息都返回ICMP目的端口不可达报文，这是NAT的限制，对于TCP来说，这不是问题，因为TCP总是点对点的；而对UDP来说，这是需解决的兼容性问题。

3. 解决方案

解决这一问题的方法显然不止一种，最直接的方法是：修改NAT的操作规则，这样做困难较大。最简单的方案则是安装一个支持UDP的代理，比如socks5，我们也做了相关试验，能够解决这一问题。但是，这种方案需要为每一个OICQ客户端做相关配置。这和NAT的思想背道而驰。

那么能不能自己设计一个UDP的透明代理，并能像NAT那样工作，无需配置客户端呢？我们进行了如下构思：

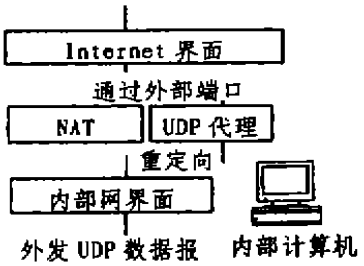

图2

如图2所示，使用数据报重定向功能把内部计算机外发的UDP数据报重定向到指定端口。在该端口UDP代理接收全部外发UDP数据报，通过Internet界面上的外部端口转发它们，并且，UDP代理记录下所有的外部端口与内部地址及端口的对应关系，在今后的操作中，所有该外部端口收到的UDP数据报均转发给对应的内部地址及端口；所有该内部地址及端口外发的UDP数据报均从对应的外部端口转发。这就实现了UDP的透明代理功能。

4. 实例

我们在Linux下实施了上述构思，使用ipchains很容易实现数据报重定向功能：命令如下：

```
ipchams -A input -s 192.168.0.0/24 \
                 -d ! 192.168.0.0/24 \
                 -p udp\
                 -j REDIRECT 1088
```

UDP代理使用C语言编写，由于源程序篇幅比较长，这里就不敷述了，其数据流图如图3。

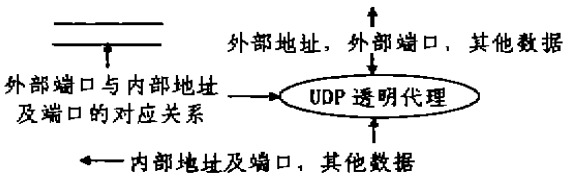

图3

结束语 笔者在Linux下配合ipchains的数据报重定向功能做了一个UDP透明代理，解决了UDP协议与NAT的一个兼容性问题，该方法在实际应用中取得了很好的效果。

参 考 文 献

1 Egevang K, Francis P. RFC1631[DB/OL]. ftp://ds.internic.net/rfc, 1994
2 Rekhter Y, et al. RFC1597[DB/OL]. ftp://ds.internic.net/rfc, 1994
3 Holdrege M, Srisuresh P. RFC3027[DB/OL]. ftp://ds.internic.net/rfc, 2001