●通信保密●

NAT 在嵌入式系统中的设计与实现

张振宇 杨宗凯
(华中科技大学电信系,武汉 430074)

【摘 要】网络地址转换(NAT)是路由器实现共享接入和网络安全的重要机制。介绍了 NAT 的原理及其在实时操作系统 VxWorks 上的一种实现。

【关键词】网络地址转换 网络地址端口转换 VxWorks 应用级网点 路由器

The Design and Implementation of NAT Based in Embedded System

Zhang Zhenyu Yang Zhongkai
(Department of Electronics and Information of Huazhong University of Science and Technology, Wuhan 430074)

【**Abstract**】NAT is an important sharing access and security mechanism of router. This paper will describe the theory of NAT and its implementation under the real time operating system VxWorks.

【**Keywords**】NAT, NAPT, VxWorks, ALG, Router

1 引言

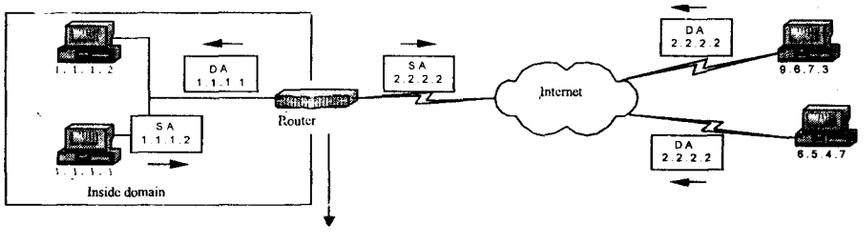
NAT 最初是作为一个解决 IPv4 地址不足的过渡方案,它允许多台主机复用或共享一个全局的合法 IP 地址。由于 NAT 隐藏了内部 IP 地址,具有隔离内部网与 Internet 的功能,因此是各种路由器实现共享 Internet 接入和防火墙功能的一种重要方法。这里描述在实时操作系统 VxWorks 上实现的以 NAT 为主体的一种 Internet 共享接入和防火墙机制。

2 NAT 及实时操作系统 VxWorks 简介

2.1 NAT 简介

NAT 的作用是将 IP 地址从一个域映射到另一个域,在两个地址域之间提供透明的包传递,而包的发送方和接收方都不会意识到 NAT 的存在[1]。其工作原理如图 1 所示。内部局域网主机发往 Internet 的 IP 包的源地址为 1.1.1.2 或 1.1.1.1,包通过路由器(router)中 NAT 处理后,其地址将转换成共享地址 2.2.2.2;接收的 IP 包通过路由器后,其目的地址将相应恢复成内部私有地址 1.1.1.1 或 1.1.1.2。至于区别两台主机建立的不同连接的方法是设置不同的端口号 1024、1723。

NAT 有多种类型,主要包括 static NAT、dynamic NAT、NAPT 等[2]。static NAT 是一种最简单的情况,它在私有地址(private address)和公有地址(public address)间实现一对一映射。在转换过程中,不仅包的首部、包的负载必须被考虑及 IP 校验和必须被重新计算,且由于 TCP 校验和的计算包含伪首部,因此 TCP 校验和也必须重新计算。static NAT 隐蔽内部局域网简单,但是它不能根据连接建立的需要来分配共有地址,即每个私有地址都必须有动态地址对应。



地址映射表

Protocol	source address: port	NAPT address: port	destination address: port
TCP	1.1.1.2:1723	2.2.2.2:1723	6.5.4.7:23
TCP	1.1.1.1:1024	2.2.2.2:1024	9.6.7.3:23

图 1 NAT 工作原理图

与 static NAT 不同的是,dynamic NAT 的实现是通过在内

收稿日期:2002-12-26。
张振宇:1978 年生,硕士。现从事现代通信技术、通信网及移动通信的研究工作。
杨宗凯:1963 年生,教授,博士生导师。现从事现代信息网络理论及其应用和现代数字信号处理技术的研究工作。

部网中设置一个共享公有地址池来达到地址复用，从而减少对合法注册公有地址的需求。内部网的主机从地址池中将得到一个 IP 地址，初始化一个连接。当连接被中止或超时后，主机将归还地址池其使用的 IP 地址，让其它主机使用。但 dynamic NAT 需要维护连接的状态，因而比 static NAT 实现更复杂，且当公有地址池被耗尽时，新的连接也将被拒绝。

针对 dynamic NAT 的缺点，NAPT(Network Address Port Translation，又称 masquerading)对其进行了改进。NAPT 通过复用不同的流来共享一个 IP 地址，使建立的连接数不受公有地址数的限制。NAPT 通常应用于 SOHO(small Office/Home Office) 路由器，使一个局域网共享只有一个公有地址的 Internet 接入[3][4]。

在本系统中，综合实现了 static NAT、dynamic NAT 及 NAPT，并通过命令行来灵活进行配置。

2.2 VxWorks 简介

VxWorks 操作系统是美国 WindRiver 公司设计开发的一种嵌入式实时操作系统(RTOS)，具有先进的网络功能。VxWorks 网络协议栈是 VxWorks 内部的一个 BSD4.4 兼容的高性能实时 TCP/IP 协议栈，相对于 BSD4.3，它增加了完全的路由支持以及 Internet 的一些新特性，使得 VxWorks 的网络性能更加优越，能适合从高性能的网络交换设备到低价的网络接入设备的需求，如 10M/100M 以太网交换机、广域网接入设备、ATM 交换机等。OSPF、RIP、DHCP、DNS、SNTP 和其他应用协议通过 socket 与 UDP、TCP、IP 协议通信，接口层(Interface Layer) 即 MUX 隔离了底层驱动程序与协议层的实现细节，使得协议能通过标准的库函数与驱动程序实现灵活的绑定和通信。

由于 VxWorks 网络协议栈缺少 NAT 和 Firewall 模块，所以 VxWorks 提供了两种 hook 函数来抓获 IP 包，这是 NAT 和 Firewall 模块对包进行分析和转换的基础。两种之一是 EtherHook 函数，另一是 IpFilterHook 函数[5]。

EtherHook 可以直接捕获收、发的原始双向以太网包，但是只有特定的 VxWorks 驱动程序才支持 EtherHook，而且因 EtherHook 是在 IP 层以下进行操作，所以改动包的内容后，也必须相应改动 IP 校验和。

IpFilterHook 作为 IP 层的一部分，只能捕获、处理所收的原始 IP 包，在进行处理后再选择是否交给 IP 层选路。IpFilterHook 的优点在于不需驱动程序的支持，IP 校验和则可交由 IP 层计算。

本系统 NAT 将利用 IpFilterHook 来实现。IpFilterHook 通过 ipFilterLib 库提供 IpFilterHookAdd 函数动态加载。

3 NAT 的设计实现

系统将实现 static NAT、dynamic NAT、NAPT 以及能进行包过滤的 Firewall 功能，整个模块在协议栈中的位置如图 2，NAT 模块的整体结构如图 3。模块可以通过命令行动态加载，然后根据不同的需要分别配置为 static NAT、NAPT、Firewall 等功能。

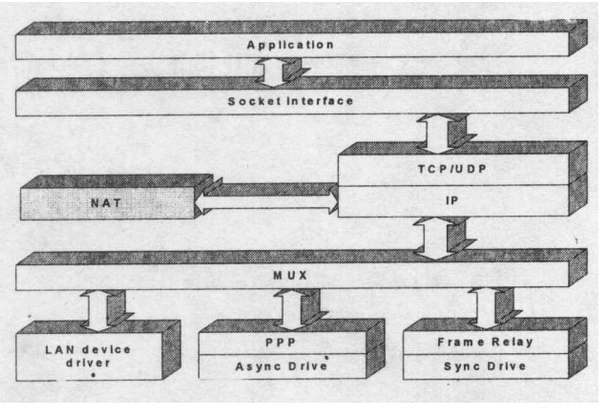


图 2 NAT 在 VxWorks TCP/IP 协议栈中的实现位置

3.1 NAT 模块总流程

如图 3，当 IP 协议运行时，任务 tNetTask 中的 ipintr() 函数将调用 hook 函数捕获 IP 包，NAT 将检查其 IP 首部和 TCP/IP 首部。进一步的 NAT 和 Firewall 功能将基于从 IP 首部和 TCP/IP 首部获得的信息。

首先，管理模块(management module)将根据配置、包首部及底层接口信息选择进一步处理的模块。

包过滤模块(Packet Filter module)将捕获的 IP 包的源地址、目的地址、协议类型、端口号等与配置的规则进行匹配过滤，以决定丢弃或进行下一步处理。

其次，伪装模块(masq module)和去伪装模块(demasq module)将对 IP 包进行 static NAT、dynamic NAT 或 NAPT。如果应用协议为 FTP，则在伪装和去伪装过程中调用 FTP 应用级网关模块(FTP ALG module)进行处理。

最后，hook 函数返回 FALSE，将包交还任务 tNetTask，由 IP 协议选路。

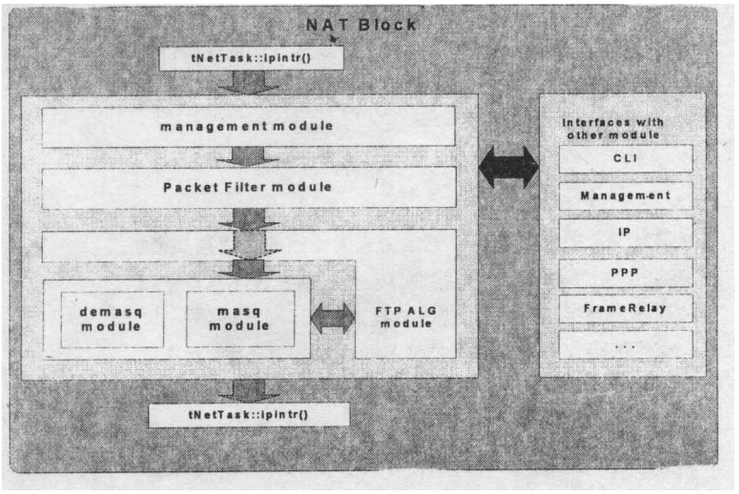


图 3 NAT 的整体结构

3.2 NAPT 的工作流程

相对于 NAT，NAPT 执行伪装功能时，它首先将检查地址

(下转第 111 页)

边界 id 的可能性是$\frac{1}{2^k}$。在有 m 个攻击者的情况下，对任意特定的距离 d，在最坏的情况下最多有 m 个独立的攻击路径（实际上，独立的攻击路径仅仅存在于离受害者非常近的地方，而且也只是很小一部分。在决大多数情况下，多攻击都是共享着攻击路径上的大部分）。因此，对任意的距离 d 的边界 id 被不正确接受的可能性最多为$1-(1-\frac{1}{2^k})^{m'}$。

这里可能存在着有 m^k 碎片合并的最坏情况，划分碎片的参数 k 对这个结果的影响很大。如当 $h=32$ 和 $k=4$，并要求重构攻击路径的确定性不低于 97% 时，在相同的距离上（$d=10$, $p=\frac{1}{25}$）有着 100 个独立的服务器（100 条独立的攻击路径）。然而，对 $h=32$ 和 $k=8$，（这个值是实际用来实施的）在相同的距离上获得同样的确定性仅仅会有 10 个独立的路由器。

压缩的边界采样算法的最大缺点是当出现多个攻击路径分岔的时候要考虑到大量的合并。虽然这些合并可以离线计算，但如果出现大的 k 和 m 值，可能会变的不可控。例如，即使用 $k=8$ 和 $m=10$，如果分离的攻击路径导致对每一个攻击者来说出现 10 个彻底独立的边界，那么这将需要考虑大约 10 亿次的合并。

在一般情况下，对 IP 数据包的调整仅仅是增加它的距离域，报头检查和（Header CheckSum）根本用不着调整。对报头的操作也仅仅是增加距离域一次，减少 TTL 域一次，再就是如果两者中有谁溢出就做一次检查。所以，这种布局调整可以使压缩的边界采样算法的执行达到很高的性能，并保持着一个低的路由器流量负载。

IP 标识域的重新布局需要注意 IP 碎片流量引起的后向兼容性问题。在对这问题没有极好的解决方案的前提下只能容忍一些碎片流量的缺点，幸运的是，实验表明：只有少于 0.25% 的数据包被分成碎片。所以，应该相信该种压缩编码在绝大多数的情况下是可以和现存的 IP 协议实行无缝连接的[3]。

参考文献

1 CERT Coordination Center in collaboration with Neil Long and Rob Thomas. http://www.cert.org/archive/pdf/DoS_trends.pdf

2 Distributed Denial of Service Defense Tactics Simple Nomad, Bind View RAZOR. http://razor.bindview.com/publish/papers/DDSA_Defense.html

3 Stefan Savage, David Wetherall, Anna Karlin and Tom Anderson: Practical Network Support for IP Traceback. Department of Computer Science and Engineering University of Washington Seattle, WA, USA, 2001

（上接第 87 页）

映射表以确定某个连接的表项是否存在，如果存在，则用伪装地址、伪装端口号分别替换包首部中的源地址、源端口号；如果不存在，则建立一个新表项，并使用它进行伪装。因为首部进行了替换，所以校验和需要重新计算。

NAPT 执行去伪装功能时，它将与目的地址、目的端口号与地址映射表的表项进行匹配，如果不存在匹配表项，则将包直接交于 IP 层处理；如果存在，则将首部中的目的地址、目的端口号用地址映射表中记录的指向内部网主机真实的目的地址、目的端口号替换。当然也需要重新计算校验和。

在 NAPT 实现过程中，对于特定协议包（如 FTP 包），需要调用相应应用级网关（ALG）处理。NAPT 将检查地址映射表每一表项并记录其状态，并定时清理检查地址映射表，删去超时表项，释放系统资源。

3.3 NAT 中的应用级网关（ALG）

NAT 本身并不能透明地支持所有的协议，因此使用 NAT 时，常常需要应用级网关（ALG）来为应用程序建立一个透明的环境。

FTP 是最常见的一个需要 ALG 才能透明通过 NAT 的协议。这是因为在 FTP 的报文中，包含有用 ASCII 表示的 IP 地址和端口号，当必须改变它们时，有可能会引起报文长度的变化，由此也将引起 TCP 序列号的偏移。本系统的 FTP ALG 将解决这些问题。

4 结束语

这里介绍了 NAT 的原理及其在实时操作系统 VxWorks 上的一种设计与实现。已在多种型号路由器上得到应用，实际表明，设计可行，实现稳定。需要进一步解决的问题是设计实现更多种 ALGs 以支持尽可能多的高层协议，以及实现 bidirectional NAT 等更多类型的 NAT，以满足各种应用需求。

参考文献

1 Egevang K B, Francis P. The IP Network Address Translator(NAT). RFC1631, 1994

2 Srisuresh P, Holdrege M. IP Network Address Translator(NAT) Terminology and Consideration. RFC2663, 1999

3 Hain T. Architectural Implications of NAT. RFC2993, 2000

4 Srisuresh P, Egevang K B. Traditional IP Network Address Translator (Traditional NAT). RFC3022, 2001

5 Gary R, Wright W, Stevens R. TCP/IP 详解与实现（卷 2），北京：机械工业出版社，2000