第34卷 第22期 计 算 机 工 程 2008年11月
Vol.34 No.22 Computer Engineering November 2008
·网络与通信· 文章编号：1000—3428(2008)22—0094—03 文献标识码：A 中图分类号：TP393

基于包截获技术的局域网内主机数的探测

余华君，姜开达
(上海交通大学网络信息中心，上海 200030)

摘 要：针对大型网络运营管理中通常需要了解网络中是否存在局域网的问题，提出一种探测大型网络中局域网的有无，并统计局域网内活动主机数的方法。采用网络包截获技术，并通过对截获包的序列号、源地址、目的地址、源端口、目的端口进行分析来实现主机的识别算法。在实际网络环境下长期运行的结果表明，以该算法开发的软件能够较准确地探测出大型网络内局域网的数目以及各局域网内的活动主机数目。

关键词：局域网；网络地址转换；TCP/IP协议；主机探测

Detection of Hosts Number in LAN Based on Packets Capture Technique

SHE Hua-jun, JIANG Kai-da
(Network & Information Center, Shanghai Jiaotong University, Shanghai 200030)

【Abstract】There are many needs to know the existence of LAN in the management of big networks. Facing this problem, a technique to detect the existence of LAN in a large network is described, and the number of active hosts in the LAN is counted. The technique is based on the technology of packets capture. By processing the IP_SEQ, SRC_IP, SRC_PORT, DEST_IP, and DEST_PORT of the captured packets, those packets send from individual machines can be discriminated, and the number of machines can be determined. The implementation, tested on real network environment, can properly count the number of LAN in the network and the number of hosts behind a LAN, which demonstrates the feasibility of the algorithm.

【Key words】LAN; Network Address Translation(NAT); TCP/IP; hosts detection

1 概述

IPv4协议对IP地址长度的限制导致因特网面临地址匮乏的困境。越来越多的单位需要访问因特网，由此网络地址转换(Network Address Translation, NAT)技术应运而生。NAT将有限的IP地址动态或静态地与内部的IP地址对应起来，极大地缓解了地址空间短缺的问题。同时它还隐藏了内部网络地址信息，使外界无法直接访问内部网络设备，保护了局域网内部的信息安全。NAT技术对局域网内部主机信息的屏蔽导致网管人员很难得到一个准确的目前在线访问因特网用户数的统计。为了解决这个问题，有人提出用操作系统特征识别[1-2]的方法进行探测，这种方法起到了一定的效果，但是只是对小规模、低流量的局域网探测效果较好。本文在他们的工作基础上提出了一种基于网络包截获技术的局域网内主机数的侦测方法，并且在较大规模、大流量、多连接的局域网环境以及无线局域网中作了实际检验，结果显示本方法可以较准确地判断出上述局域网内主机的数目。

2 NAT技术基本原理

在采用NAT路由器的网络拓扑中(如图1所示)，当内部网络的包到达NAT路由器时，路由器根据其地址转换表将包的首部信息改变：替换它的源IP地址或源端口号，或者既替换它的源IP地址又替换其源端口号，然后将这个变化以后的包发送出去；同时将这个对应关系加入到其地址转换表中。当外部网络的包到达NAT路由器时，检查这个包首部的目的IP地址和端口号，在地址转换表中查找与之对应的表项。若找到，则用地址转换表中保存的地址和端口号来替换目的地址和端口号，然后将这个包发送到内部网络；否则就丢掉这个包。对于外部网络来说，无法看到这个发送包的源计算机，从而实现了地址的隐藏和伪装。采用NAT以后，NAT路由器只允许从局域网内发起的连接。外部的计算机不能直接连接到内部的计算机，除非内部计算机已经发起了这个连接。因此内部计算机可以连接到因特网，而外部网络不能得知内部主机IP地址，也就不可能凭IP地址简单地得知局域网内部的主机数目。

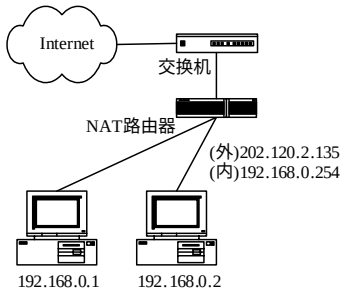

图1 NAT路由网络拓扑

3 算法设计

3.1 算法实现基础

本算法的实现基于以下发现：

(1)当一个主机连续地发送数据包时，IP包首部[3]的16位

作者简介：余华君(1978－)，男，工程师，主研方向：网络安全，信号处理；姜开达，工程师

收稿日期：2008-03-22 **E-mail：**shehuajun@sjtu.edu.cn

标识ID值是递增的，连续的ID序列反映了某一台主机发送的包。通过统计从某个IP地址发送的包的ID序列的数目就可以统计出该IP地址后面主机的数目。

(2)TCP包以及UDP包的首部都会有一个源端口和目的端口[4]，根据人们上网的行为模式，在上述观察基础上加入一个端口对应来加强确认某个包是来自某个主机。基于这样一个假设，即在一段时间里一个连接通常会有好几个包发出和接收，即在某个小的时间间隔里(源地址：源端口-目的地址：目的端口)这一四元组的对应关系是不变的。

(3)同样根据人们上网的行为模式以及 NAT 路由器的工作模式，假定在某个小的时间间隔里从 NAT 路由器的同一个端口发出来的包很有可能由同一台主机发出。

(4)在包截获过程中，发现上述第 1 个观察结果中提到的某台主机发出的 ID 有时会发生较大的跳变，但跳变之后的 ID 序列仍然是连续的。

(5)另外，有时从同一台主机发出的 ID 比较小的包会比 ID 比较大的包迟到，即包的顺序出现了混乱。

(6)在主机数目比较多的情况下，不同主机发出的包的 ID 序列可能产生重叠。

(7)因为IP包的ID只是 16 位整数，所以当一台主机的ID逐渐增加到 2^{15}-1 时，ID又会从 0 开始计数。

3.2 算法实现

定义主机链为主机的一个集合，该集合包括笔者辨别出来的主机。而每个主机又包含有一个由该主机发起的不同的网络连接的集合，即一个(源地址、源端口、目的地址、目的端口)四元组的集合。每个网络连接链又记录了所有的从这个网络连接发出的 IP 数据包的 ID、源地址、源端口、目的地址、目的端口、截获时间。

定义几个参数：

timeout：最大延时，暂定为 300 s；

maxstep：步长门限值，暂定为 5；

maxleap：跳跃门限值，暂定为 100；

minpkgnum：最小发包数，暂定为 5；

maxoverlap：最大允许重叠数，暂定为 10。

算法步骤如下：

(1)镜像并侦听要探测的局域网所在端口的数据包，通过 WinPcap 库截获得到新的 IP 包，取出 IP 包首部中的 16 位标识 ID、源地址、目的地址、截获时间。另外如果判断是 TCP 或者 UDP 包，则取出其源端口、目的端口。

(2)如果新数据包的截获时间与上一个数据包的截获时间差超过 timeout，跳转到(1)，继续侦听。否则跳转到(3)继续判断。

(3)如果新数据包的源地址、源端口、目的地址、目的端口与某个已识别主机的网络连接相同，则新的数据包完全符合该主机的这一网络连接，在匹配的那个主机的网络连接里添加该 ID、源地址、源端口、目的地址、目的端口、截获时间，跳转到(1)，继续侦听。否则跳转到(4)继续判断。

(4)如果新数据包的 ID 值减去某个已识别主机的上次最后收到的数据包的 ID 的差小于 maxstep，则认为新的数据包完全符合该主机，只是新开了一个网络连接，在匹配的那个主机的网络连接集合里新添加一个网络连接，并在新加的网络连接里添加截获的数据包的 ID、源地址、源端口、目的地址、目的端口、截获时间，跳转到(1)，继续侦听。否则跳转到(5)继续判断。

(5)如果新数据包的 ID 值减去某个已识别主机的最后收到的数据包的 ID 的差小于 maxleap，则有 2 种可能：1)新的数据包可能符合该主机，只是该主机的 ID 值出现跳跃，并且新开了一个网络连接；2)新的数据包是由另外一台主机发出的，并且这 2 台主机的 IP 包的 ID 值互相重叠。这时需要更多的信息来区分这 2 种情况，所以在该主机下面新建一个网络连接，添加该 ID、源地址、源端口、目的地址、目的端口、截获时间，并将该包做一个延迟判断标识，待将来有更多的数据包时再作判断，跳转到(1)，继续侦听。否则，如果新数据包的 ID 值减去某个已识别主机的最后收到的数据包的 ID 的差小于 0 且大于负的 maxoverlap，则认为是该包的次序发生了颠倒，新的数据包符合该主机。这种情况下在该主机下面新建一个网络连接，添加该 ID、源地址、源端口、目的地址、目的端口、截获时间，并将该包做一个重叠标识，待将来有更多的数据包时再作判断，跳转到(1)，继续侦听。否则跳转到(6)。

(6)若以上情况都不符合，则认为这是一台以前未识别出的主机发出的 IP 数据包。在主机集合里新添加一个主机，在该主机下面新建一个网络连接，新建的网络连接里添加该 IP 包的 ID、源地址、源端口、目的地址、目的端口、截获时间，跳转到(1)，继续侦听。

通过上述计算，可以得到一个主机集合。最后忽略那些收到的总包数小于 minpkgnum 的主机得到的数目，就是估计的 NAT 后面主机的数量。

上述算法是一个相对保守的值，在某些场合下这个值可能相对实际值偏小。比如，第(5)步中提到的可能出现的 2 台主机发出的 IP 数据包的 ID 重叠的情况，这时就可以用算法中提到的重叠标识校正估计值。另外，有的主机只发出了很少几个 IP 包，这种情况的主机也予以忽略。

4 局域网内主机数侦测

WinPcap 是基于 Windows 平台的开放源代码网络数据包截获和分析系统。该系统性能稳定而且效率高，利用它提供的功能强大的网络数据包处理函数，可以满足对数据包处理有严格性能要求的大多数应用。应用程序是在 WindowsXP 环境下用 Visual C++ 6.0 开发的。使用 WinPcap 需要引入 2 个头文件：#include "pcap.h", #include "remote-ext.h"。并且在 Project/Setting/Link 菜单下的 library modules 中加入 wpcap.lib 和 Packet.lib 2 个库文件。这 2 个库文件包括了所有使用 WinPcap 进行网络包截获的库函数。

本系统的主要程序代码如下：

```
void Detector::Detect(){
  pcap_if_t *alldevs;
  pcap_if_t *d;
  u_int netmask;
  struct bpf_program fcode;
  int i=0;
  char errbuf[PCAP_ERRBUF_SIZE];
  struct pcap_pkthdr *header;
  const u_char *data;
  //获取本机的网络适配器列表
  pcap_findalldevs_ex(PCAP_SRC_IF_STRING, NULL,&alldevs,
errbuf);
  //用户输入网络适配器序号
  int number;
  cin>>number;
```

```
for(d=alldevs;i<number;){d=d->next; i++; }
//打开网络适配器
m_adptHandler=pcap_open(
    d->name,//网络适配器名称
    65536, //每个包保存的字节数
    PCAP_OPENFLAG_PROMISCUOUS,//混杂模式
    1000, //读超时 1000 毫秒
    NULL, //不需认证
     errbuf //错误缓冲区);
if(d->addresses != NULL) //得到网络适配器的地址掩码
     netmask=((struct  sockaddr_in*)(d->addresses->netmask))->
sin_addr.S_un.S_addr;
else
//否则默认为扫描 C 类网络地址
     netmask=0xffffff;
char* filter = "ip";
//编译过滤器
pcap_compile(m_adptHandler, &fcode, filter, 1, netmask);
//设置过滤器
pcap_setfilter(m_adptHandler, &fcode);
while(USER_NOT_BREAK){pcap_next_ex(d, &header, &data);
Process(data); }
//不再需要设备列表，予以释放
pcap_freealldevs(alldevs);
return;  }
```

其中，库函数 pcap_findalldevs_ex()获取本机的网络适配器列表，写入 alldevs 变量；库函数 pcap_open()打开用户指定的网络适配器；库函数 pcap_compile()编译过滤器。库函数 pcap_setfilter()设置过滤器；库函数 pcap_next_ex()侦听网络适配器，如果有数据包到来就读到 data 缓冲区中；主处理函数 Process()利用上述算法对接收到的包进行分析处理。

5 程序运行结果及分析

本程序的某次运行结果如表 1 所示。

表 1 运行结果

host	srcip	srcpt	minid	maxid	len
Host1	202.120.34.47	60023	1802	1834	32
	202.120.34.47	54201	1826	1826	1
Host2	202.120.34.47	59931	3683	4613	931
Host3	202.120.34.47	60445	1079	1092	14
Host4	202.120.34.47	60450	252	252	1
Host5	202.120.34.47	60432	5889	5896	8
Host6	202.120.34.47	60451	5455	5455	1
Host7	202.120.34.47	52685	45206	45206	1
Host8	202.120.34.47	59362	2031	2036	6

这次试验是笔者在上海交通大学校内的一个教学局域网内做的，当时这个局域网内上网的学生有 8 人。程序的运行结果中分别显示了发现的各台主机，及主机下面的各条网络连接对应的源地址、源端口，以及每条网络连接收到的 IP 包的数目以及最小 ID 值和最大 ID 值。在这次采样过程中，程序发现了 8 个怀疑主机(如表 1 所示)，但是有 3 台主机在测试过程中发送的数据包比较少(分别只有 1 个数据包)，小于前面定义的 minpkgnum 最小发包数 5，所以在最后的统计中将它们忽略。

从运行结果可以看出，程序较为准确地识别出了局域网内的各台主机。由于人们上网行为模式的不确定性，一次采样不可避免地带有波动性。因此，在此基础上提出用统计方法对多次采样结果进行处理，得到一天或者一个月里局域网内活动主机数的最高值、最低值、平均值以及方差。这样就可以比较准确地估计出一个局域网内的主机数目，为校园网的发展规划提供参考依据。

6 结束语

笔者在实际包截获过程中发现，有些操作系统的 ID 值并不是递增的而是随机的，例如某些版本的 FreeBSD，这时引入另外的一些特征值可以在某种程度上提高程序的正确性，如文献[3]提到的操作系统指纹识别法等。但使用这些操作系统的用户相对较少。对于常用的 Windows XP, Windows 2000, 大多数 Linux 操作系统而言，该 ID 值都是递增的，所以本文介绍的算法可以应用于绝大多数局域网的主机数目探测当中。本文所阐述的方法已经在上海交通大学校内的很多局域网中经过测试，其中也包括无线局域网。运行结果比较准确，抗干扰能力较强，效果良好。

在主机的识别上还可以做进一步的研究，比如可以尝试使用模式识别理论、小波分析、神经网络、时间序列分析等现代分析手段更好更准确地分析出主机的数目。另外，更精确的识别有赖于更丰富的信息，对于某些特殊的网络或网络协议，如 IPsec 协议等，有更多的特征点可以准确地识别出主机的特征，则算法的准确性将得到进一步的提高。

参考文献

[1] Phaal P. Detecting NAT Devices Using sFlow[EB/OL]. (2003-02-01). http://www.sflow.org/detectNAT/.

[2] Miller T. Passive OS Fingerprinting: Details and Techniques[EB/OL]. (2002-02-01). http://www.ouah.org/incosfingerp.htm.

[3] Stevens W R. TCP/IP 详解(卷一)[M]. 北京: 机械工业出版社, 2000.

[4] Tanenbaum A S. 计算机网络[M]. 3 版. 北京: 清华大学出版社, 1998.

(上接第 85 页)

参考文献

[1] Hennessy J L, David A. Patterson. Computer Architecture: A Quantitative Approach[M]. 北京: 机械工业出版社, 2002-09.

[2] Farley M. SAN 存储区域网络[M]. 北京: 机械工业出版社, 2002-04.

[3] Itakura K. MDPS: The New Mass Data Processing Storage System for the Earth Simulator[Z]. Yokohama, Japan: The Earth Simulator Center, 2006.

[4] Rauscher T. Keep up with RAID[Z]. Digi-data Corporation. 2003-05.

[5] 孟玉柯. 排队论基础及应用[M]. 上海: 同济大学出版社, 1989.