DOI:10.16208/j.issn1000-7024.2009.22.050

● 网络与通信技术 ●

分布式网络自动抓包管理系统的设计与实现

刘 捷， 朱程荣， 熊齐邦

(同济大学 计算机科学与技术系，上海 201804)

摘 要：讨论了网络抓包原理及 WinPcap 体系结构，探索了不同于传统客户端被动抓包的方法，对自动抓包进行了研究，提出了由客户端在特定事件触发机制下自动抓包然后推送至服务端的设计思想。在此基础上进行设计与开发实践，给出了一套应用于分布式网络管理环境，包含服务端与客户端的网络自动抓包管理系统的架构及详细设计，并给出了程序实现细节。设计并实现的系统在真实环境中得到成功应用，为网络管理提供了有效的支持。

关键词：网络管理；网络数据包；自动抓包；winpcap；分布式

中图法分类号：TP393.07 **文献标识码**：A **文章编号**：1000-7024 (2009) 22-5091-03

Design and implementation of distributed network automatic packet capture management system

LIU Jie, ZHU Cheng-rong, XIONG Qi-bang

(Department of Computer Science and Technology, Tongji University, Shanghai 201804, China)

Abstract: The theory of network packet capturing and the architecture of WinPcap are discussed. Method other than traditional passive packet capture on the client side is explored. Automatic packet capturing is researched. A design concept of automatically capture packets is proposed by the client itself when some specific pre-defined events triggered and then the packets is sent to the server. Design and implementation are practiced based on that concept. The structure and detail design of the whole set of system is provided including server and clients which can be applied in a distributed network environment. Implementation details are also provided. The designed and implemented system is successfully applied in real scenario and provided useful support to networking management.

Key words: network management; network packet; automatically packet capture; WinPcap; distributed

0 引 言

网络抓包是网络管理中一种比较常用的监测与分析网络节点状况的基本途径和有效手段。由此应运而生出各种相关的抓包工具和软件，如 sniffer、snort 等。然而，目前使用较广泛的点对点抓包工具基本都以服务端发出抓包命令，对于目标主机或其所在子网进行抓包(即服务端从客户端“拉”数据包)，而无法支持当目标主机在特定条件下自动抓包并将数据包传输给服务端的应用情形(即客户端主动把捕获的数据包“推”给服务端)。

本文所设计并实现的分布式网络自动抓包管理系统，不仅支持客户端自动抓包，而且能应用于具有众多客户端的分布式网络管理环境中。该系统还满足当客户端与服务端出现连通性问题后，客户端在其所在子网中自动抓包，待与服务端恢复连通后将数据包传输给服务端以进行分析，从而为网络管理提供有效的参考。

本系统已实际应用于教育局下属某网络运维中心以协助管理辖区内中小学网络节点，使用效果良好。

1 网络抓包原理及 WinPcap 体系结构

网卡完成收发数据包的工作，有 4 种工作模式，要想截获不是给自己数据流，就必须直接通过网卡的混杂模式，使之可以接受目标地址不是自己的 MAC 地址的数据包，直接访问数据链路层，取数据。本系统使用 WinPcap 提供的动态链接库在网卡设置为混杂模式下进行抓包。

WinPcap 的主要思想来源于 Unix 系统中最著名的 BSD 包截获架构，它由内核级的网络组包过滤器(netgroup packet filter，NPF)、用户级的动态链接库 Packet.dll 和高级动态链接库 Wpcap.dll 等 3 个模块组成。使用 WinPcap 既可用包含在 Packet.dll 中的低级函数进入内核级调用，也可用由 Wpcap.dll 提供的高级函数调用[1]。本系统使用 Wpcap.dll 提供的函数，因为其功能更强，使用更方便，且容易移植到 Unix 下。

2 系统设计

本系统分两部分：服务端和客户端。服务端作为一个管理平台，对客户端进行配置和控制，需图形界面；客户端负责

收稿日期：2008-10-24；修订日期：2008-12-22。

作者简介：刘捷(1985－)，男，上海人，硕士研究生，研究方向为信息安全及容错计算；朱程荣(1960－)，男，上海人，硕士，副教授，研究方向为信息安全及容错计算；熊齐邦(1947－)，男，上海人，教授，研究方向为计算机网络与分布式系统。E-mail：liuj85@gmail.com

本地抓包,运行于后台,无需界面。客户端相当于一个网络软探针,侦探本地到服务端的连通情况并根据规则及参数触发自动抓包或响应服务端发出的即时抓包命令,之后将数据包文件发送给服务端;服务端汇总各个客户端捕获的数据包文件及连通性信息,从而能进一步分析网络连通性和连接性能。

2.1 服务端总体功能及架构设计

本系统的应用环境中,服务端作为远程控制台,对部署在各学校的客户端进行管理。管理功能包括:日志管理、数据包管理、客户端触发抓包功能开关、客户端抓包参数配置和实时命令抓包。服务端总体设计架构如图1所示,自顶向下分为3层:用户界面、服务端管理和访问管理。

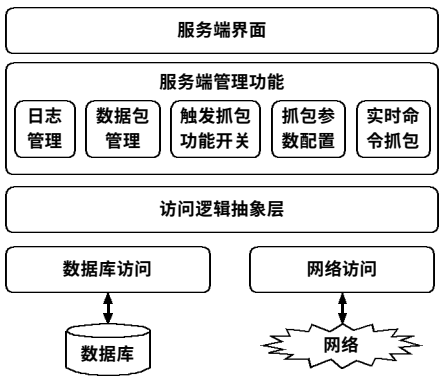

图1 服务端总体架构

用户界面层包括主界面和其它各窗口。通过服务端管理层与数据层和网络层通信。访问逻辑抽象层作为接口层,对基本的数据库操作和网络操作进行抽象。

2.2 客户端总体功能及架构设计

客户端总体功能及架构图如图2所示。逻辑上从上而下分为客户端功能层、访问逻辑抽象层。

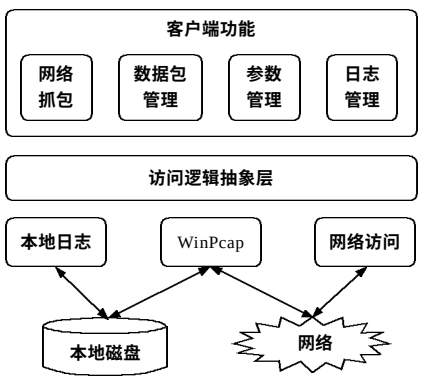

图2 客户端总体架构

(1)网络抓包功能由事件驱动。服务端所触发的抓包事件为服务端发出即时抓包命令。客户端自身触发的抓包事件分为断网事件和延迟事件。客户端在一定周期通过Ping方式进行链路检测,当发现通往服务器链路不通或延迟超过预定义参数所设时间后,则启动抓包功能,按参数抓包。在链路恢复正常前只进行一次抓包,抓获的数据包存储在文件系统中,待链路恢复正常后将数据包文件主动推送给服务端。

(2)客户端数据包管理功能管理捕获的数据包文件在本地文件系统中的存储,在客户端本地安装目录中建立两个文件夹AutoCapped和InstantCapped来存储捕获的数据包。

(3)客户端的抓包参数管理用一个XML文件存放抓包参数。客户端有独立线程监听服务端的参数更新命令。它接收服务端的参数并更新XML参数文件。当该线程更新参数文件时,其它线程必须同步等待该线程结束,然后根据新参数进行抓包和其它操作。

(4)客户端日志记录在LOG格式的日志文件中。

(5)访问逻辑抽象层的目的在于将数据库的访问、网络连接等底层操作进行整合,位于上层的各个功能模块和用户界面的代码只需要调用该层的方法来和底层通信。

2.3 客户端详细模块设计

客户端详细模块分割如图3所示。

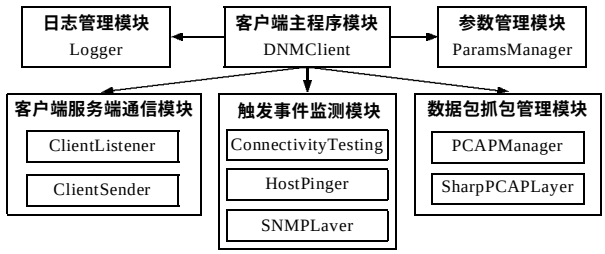

图3 客户端详细模块分割

(1)客户端主程序模块(DNMClient)运行于后台。定义对各模块的触发条件,满足后自动调用相关处理模块。

(2)客户端服务端通信模块由ClientListener和ClientSender组成。ClientListener监听服务端的命令及消息,判断“读取客户端参数”、“修改客户端参数”、“即时抓包”等命令后调用相应功能模块。ClientSender向服务端发送消息及传输捕获的数据包。

(3)触发事件监测模块由ConnectivityTesting进行管理,它对Ping连通性测试及自动抓包、发送数据包等功能进行了事件定义和相应的触发条件,当满足触发条件时,调用数据包抓包管理模块中的抓包及发送数据包功能。HostPinger负责连通性测试的具体实施与监控,运行过程中实时参照客户端参数文件对服务端进行连通性测试,并按参数中的标准来调用相应的自动抓包功能。SNMPLayer主要通过SNMP方式进行连通性测试相关工作。

(4)数据包抓包管理模块由PCAPManager及SharpPCAPLayer组成。PCAPManager主要负责数据包即时抓包与自动抓包。即时抓包时,服务端发送抓包命令,客户端ClientListener捕获命令并调用抓包管理模块中的InstantPacketDump函数触发底层SharpPCAPLayer中的WinPcap抓包API进行即使数据包的捕获。当客户端ConnectivityTesting触发了自动抓包事件,将调用抓包管理模块中的AutoPacketDump函数触发抓包API进行即时数据包的捕获。通过自动抓包功能捕获的数据包文件存放在客户端程序安装路径\DNMClient\AutoCapped中。其中子目录以当天的日期命名(如2008-6-5)。当天捕获的数据包存储在该子目录中,数据包的命名格式为“捕获日期+@+捕获时间+客户端IP”。即时抓包的数据包文件存放在\

InstantCapped 中，其它同自动抓包。SharpPCAPLayer 负责底层核心抓包操作并进行封装。上层抓包功能均调用此层来完成具体抓包工作。

(5)参数管理模块(ParamsManager)负责从客户端本地配置参数文件获取参数、更新客户端 Dictionary paramDict 中的参数、更新配置参数文件。该 XML 配置文件形如：

<param id = "1" name = "LinkCheckCircle" value = "1000"></param>

<param id = "2" name = "LinkCheckMethod" value = "ping"></param>

<param id = "3" name = "TimeOut" value = "5000"></param>

<param id = "4" name = "PCapLasting" value = "10"></param>

(6) 日志管理模块 (logger) 负责向客户端本地日志文件添加新的日志记录。

3　系统实现

3.1　服务端的实现

服务端的实现在 Visual Studio.NET 2005 平台使用 C#语言结合 SQL Server 2005 进行开发。主要为人机交互、数据库存储、基于 TCP/IP 协议的网络数据通信等开发。图 4 为实际运行于教育局下属某网络运维中心的服务器端程序主界面，由菜单、树型拓扑、图形拓扑和日志信息组成；图 5 为查看客户端窗口；图 6 为客户的参数窗口。

图 4　服务端程序主界面

图 5　查看客户端

图 6　客户端参数

3.2　客户端的实现

为保证抓包性能，客户端的底层抓包模块用 C++实现并封装为 DLL。其余模块和功能则在 Visual Studio.NET 2005 平台下用 C#实现。客户端主程序运行流程如图 7 所示。

核心抓包函数 PacketDump 代码节选如下：

```
pcap_t *handle = pcap_open_live(devToUse->name, portion, mode, timeout, errorBuffer);
pcap_dumper_t *dumper = pcap_dump_open(handle, dumpFile);
int res;
struct pcap_pkthdr *header;
const unsigned char *data;
time_t startTime = time(NULL);
while((res = pcap_next_ex(handle, &header, &data)) >= 0)
{if(res == 0){continue;}
 if((current-startTime)>(lasting-5)&&(current-startTime)<(lasting+5))
 {break;}
 pcap_dump((unsigned char *)dumper, header, data);
 time_t current = time(NULL);}
pcap_freealldevs(allDevs);
pcap_close(handle);
```

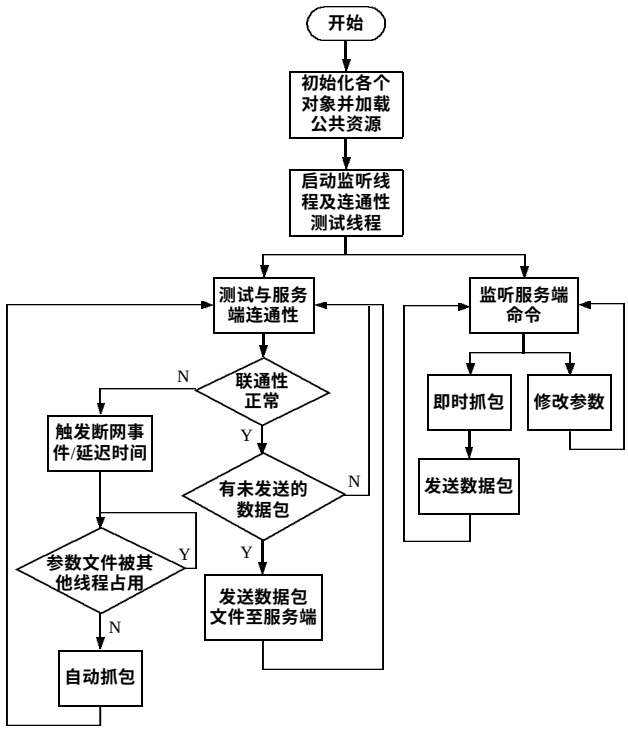

图 7　客户端主程序运行流程

4　与传统抓包方法的对比

本文所述的研究及实践工作是因传统抓包方法在一些网络管理应用中无法较好地满足需求而引发的，设计思想及实际开发的系统也有别与传统，在网络故障诊断应用中具有一定优势，具体对比如表 1 所示。

5　结束语

本文针对目前使用较广泛的点对点抓包无法支持当目标主机在特定条件下自动抓包并将数据包传输给服务端的应用情形这一现状，提出了应用于分布式网络环境的客户端自动抓包并发送给服务端的设计思想，阐述了网络抓包机制及 WinPcap 开发包的原理。并且结合实际应用需求，设计与实现了分布式网络自动抓包管理系统，在系统实际运行过程中验证了自动抓包机制的可行性及有效性，为网络管理与维护提供了一定的参考。

（下转第 5228 页）

表 1 AB 两点间最短路径搜索参数

算法	参数				
	λ(m)	N_1	N_2	L(km)	T(ms)
局部搜索算法	30	29	71	12.80	109
Dijkstra 算法	无		245	12.07	898
中国电子地图 2006 软件	无			12.07	312
本文算法	120	34	118	12.07	278

(1) 局部搜索算法搜索的节点数最少,因而耗时最短。从图 5 中可以看出,该算法得到的最短路径主要分布在起点、终点连线的左右两边,由于出现振荡,故路径最长。

(2) Dijkstra 算法要分析计算每一个节点,故在搜索完成了 247 个节点后,得到最短路径的耗时最长,其耗时是局部搜索算法的 8 倍多。

(3) 中国电子地图 2006 搜索到的路径长度与 Dijkstra 算法和本文算法得到的一致。从图 5 中可以看出,作为商业软件,由中国电子地图 2006 得到的路径主要为直路段,拐弯数目少,较为符合人们出行的选路习惯。

(4) 本文算法与局部搜索算法相比,为了避免振荡的出现,λ扩大了 3 倍,搜索的节点数随之增加了 47 个,时间花费增加了近两倍,但找了 A 到 B 的最短路径;与 Dijkstra 算法相比,搜索的节点数减少了一倍,求解时间减少了 3 倍左右;与中国电子地图 2006 软件相比,求解时间减少了 34ms。

综上所述,本文算法效率高、速度快。

4 结束语

本文研究了基于 MapX 最短路径搜索算法问题,依据最短路径沿起点、终点连线方向可能性最大的特征,采用一种改进的 Dijkstra 搜索算法来求解最短路径。与 Dijkstra 算法相比,该算法速度快、效率高。其效率取决于网络结构的复杂程度和弧段间的连通程度。网络结构、弧段间的连通程度越复杂,算法的效率就越高。因此,该算法具有很强的实用性。

参考文献:

[1] 周培德. 算法设计与分析 [M]. 北京: 机械工业出版社, 2004: 147-150.

[2] Li Zhen-ping,Wu Ling-yun,Zhao Yu-ying,et al.A dynamic programming algorithm for the κ-Haplotyping problem [J]. Acta Mathematicae Applicatae Sinica, English Series, 2006,22 (3): 405-412.

[3] 顾晓东,余道衡,张立明.时延 PCNN 及其用于求解最短路径[J].电子学报,2004(9):1441-1443.

[4] 张涛,柳重堪,张军.卫星时变拓扑网络最短路径算法研究[J].计算机学报,2006,29(3):371-377.

[5] 陈煜,吴力合.最短路径算法的研究[J].中国科教博览,2005(2):72-73.

[6] 严寒冰,刘迎春.基于 GIS 的城市道路网最短路径算法探讨[J].计算机学报,2000,23(2):210-215.

[7] 杨中宝,李朝艳,吕伟.基于 MapX 的局部最短路径算法[J].计算机系统应用,2006,16(3):83-86.

[8] 朱晓青,周涛,张海堂.Mapinfo 中道路拓扑与最优路径的研究[J].信息工程大学测绘学院学报,2001,18(2):133-138.

[9] 王卫红,顾国民.基于栅格法的矢量路径规划算法[J].计算机应用研究,2006,23(3):57-59.

[10] 唐文武,施晓东.GIS 中使用改进的 Dijkstra 算法实现最短路径的计算[J].中国图象图形学报 A 辑,2000,12:1019-1023.

[11] 倪凯,叶雷,鲁铭,等.基于数据库中间件与 GIS 实现的最短路径算法[J].计算机工程,2005,31(7):78-80.

[12] 齐锐,屈韶琳,阳琳斌.用 MapX 开发地理信息系统[M].北京:清华大学出版社,2003.

[13] 吴兴军,胡汉春. AutoCAD 二次开发相交线群自断链功能 [J]. 机电产品开发与创新,2005,18(3):58-60.

(上接第 5093 页)

表 1 与传统抓包方法的比较

对比项	传统抓包方法	本文提出的自动抓包方法
抓包触发模式	被动,由服务端发出抓包命令	主动,客户端满足触发条件后自动抓包
数据包文件的获得与传输	远程从客户端获得数据包,边抓包边在服务端生产数据包文件	客户端在本地完成抓包并存储数据包文件,完成抓包后通过 FTP 方式将数据包文件上传至服务端
在网络故障诊断中的应用	通过定时抓包或人为随机操作对客户端抓包来检测客户端网络状况并为故障诊断提供参考,较难及时捕获网络故障发生时刻客户端的数据	通过预定义的触发条件和参数,可以在客户端网络状况不良或者出现网络故障时及时地进行自动抓包,为网络故障诊断提供更有效的参考

参考文献:

[1] 胡晓元,史浩山.WinPcap 包截获系统的分析及其应用[J].计算机工程,2005,31(2):96-98.

[2] 赵新辉, 李祥. 捕获网络数据包的方法 [J]. 计算机应用研究, 2004,21(8):242-244.

[3] Andrew Parsons,Nick Randolph.Visual Studio 2005 高级编程 [M].北京:清华大学出版社,2008.

[4] The WinPcap Team.The WinPcap manual and tutorial for WinPcap 4.0.2 [EB/OL] .http://www.winpcap.org/docs/docs_40_2/html/main.html.

[5] Risso F,Degioanni L.An architecture for high performance network analysis[C].Proc 6th IEEE Symposium on Computers and Communications,2001:686-693.

[6] Anagnostakis K G,Ioannidis S,Miltchev S,et al.Efficient packet monitoring for network management [C]. Network Operations and Management Symposium,2002:423-436.

[7] Luca Deri.Improving passive packet capture:Beyond device polling[C].15th NMRG,2004.

[8] Morandi O,Risso F,Valenti S,et al.Design and implementation of a framework for creating portable and efficient packet processing applications [C]. International Conference on Embedded Software(EMSOFT),2008.