文章编号:1001 - 9081(2005)08 - 1865 - 02

基于 ActiveX 技术的通用网络检测控件

张 楠[1,2],李志蜀[1],张建华[1,2]
(1. 四川大学 计算机学院,四川 成都 610065; 2. 西南民族大学 计算机科学与技术学院,四川 成都 610041)
(panda_zn@163.com)

摘 要:使用 ActiveX 技术设计并实现了通用的网络数据包检测控件,利用该控件可以在各种编程环境下,根据用户需求捕获及处理网络底层数据包,为各种需要进行数据包获取的网络应用程序的开发提供了统一的接口,极大地方便了应用程序的开发人员。

关键词:ActiveX;Winpcap;回调函数
中图分类号: TP311.52 **文献标识码**:A

Network detecting control based on ActiveX technique

ZHANG Nan[1,2], LI Zhi-shu[1], ZHANG Jian-hua[1,2]
(1. *College of Computer Science, Sichuan University, Chengdu Sichuan* 610065, *China*;
2. *College of Computer Science and Technology, Southwest University for Nationalities, Chengdu Sichuan* 610041, *China*)

Abstract: A network packet detecting control was designed and realized based on ActiveX technique. By using this control, applications can capture and deal with network packet based on user's command in all kinds of programming environments. The control provides programmer a unified interface to develop network applications which need capturing packet. It brings users great convenience.

Key words: ActiveX; Winpcap; call-back function

0 引言

随着计算机网络复杂性和规模的不断增长,网络的安全性面临着严峻的考验。网络设计和维护人员经常需要通过检测网络上的数据包来分析、诊断和测试网络。基于网络操作系统之上的编程接口(如 Winsock)一般无法访问到底层的网络协议,基于物理设备或数据链路层的编程接口需要编程人员对网络协议、各协议层的 PDU 结构及 NIC 接口有一定的了解,加大了程序设计的难度。为了降低网络编程的复杂度,缩短网络应用程序开发周期,有必要设计一个通用的 ActiveX 控件,提供统一、便捷的编程接口进行底层数据包的检测,并可在不同的编程环境中重复使用。本控件是在VC++6.0 下利用 Winpcap3.0 编程开发的,具有高度封装、底层硬件支持良好、上层应用接口友好等特点。

1 网络检测控件的实现

1.1 基于 Winpcap 的数据包捕获技术

Winpcap 是 Win32 环境下用于数据包捕获的 C 函数库,与 UNIX/Linux 系统环境中的 BPF-libpcap 的捕获机制是兼容的。其内核部分的基本任务是从网络中取得链路层数据包,在 Windows NT/2000 系统中,其实现为一个内核驱动程序(packet.sys),在 Windows 95/98 系统中是一个虚拟设备驱动程序(packet.vxd),应用程序通过 packet.dll 动态链接库与之进行通信。

由于不同的编程语言对数据结构和数据类型的定义与 C 语言存在不同,再加上有些语言(如 PowerBuilder)不支持函数指针的传递,也不能使用回调函数。这些都给利用 Winpcap 进行程序开发造成了障碍,因此需要在VC++环境下开发一个通用的 ActiveX 网络检测控件,提供基本数据类型接口,以方便在各类编程环境下使用。

在VC++中利用 Winpcap 进行控件的开发,首先安装 Winpcap 网络驱动程序,然后在源文件中包含 pcap.h,并在编程环境链接器中包含 wpcap.lib 和 wsock32.lib。

检测控件工作在网络接口和应用程序之间,与网络接口进行通信,检测网络数据包,并向高层应用程序提供读写来自数据链路层数据的接口。在网络数据包检测过程中,网络接口工作在混杂模式,接收网段中所有数据包。图 1 为检测控件与应用程序的关系。

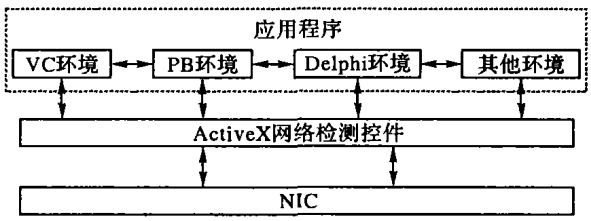

图 1 控件与应用程序的关系

1.2 检测控件框架结构设计

ActiveX 控件基于构件对象模型 COM,其基本原则是一个对象的接口和实现能够而且应该分开对待。在网络检测控件中,提供了简单的外部接口方便应用程序使用,其实现则在类成员函数中完成。

1.2.1 数据包捕获接口

检测控件的数据包捕获接口定义为:

```
[id(1)] short GetPacket(long pktlen, double times,
  BSTR filename);
```

收稿日期:2005 - 03 - 07

作者简介:张楠(1973 -),女,四川眉山人,讲师,博士研究生,主要研究方向:计算机网络; 李志蜀(1947 -),男,四川成都人,教授,博士生导师,主要研究方向:计算机网络与信息系统、智能控制; 张建华(1971 -),男,四川武胜人,副教授,主要研究方向:计算机网络与安全.

用户只需要提供三个基本参数,即可实现数据包的捕获,返回 0 表示捕获成功,返回 -1 表示捕获失败。

其中,参数 pktlen 来指明捕获数据包的最大长度,在 Win32 系统中,可以只捕获一个数据包的某一部分,这样可以减少复制数据的数量,提高系统的效率,也可以设置一个大于通常网络 MTU 的值,以保证接收到完整的数据包;times 表示捕获数据的时间,函数将在时间到后返回,如果设置为 0,表示直到有数据包到达就返回,如果设置为 -1 表示立即返回;filename 指明保存数据包的文件名。

接口定义好后,其实现是在控件的类成员函数中完成的:

```
afx_msg short GetPacket(long pktlen, double times,
  LPCTSTR filename);
```

在该成员函数中,使用 pcap_open_live( ) 来打开捕获设备:

```
pcap_t * pcap_open_live(char * device, int snaplen, int promisc,
  int to_ms, char * ebuf);
```

device 表示已经获得的网络设备的字符名;snaplen 则为第一个接口参数 pktlen 所指定的表示数据包大小;参数 promisc 一般设置为 1,表示把网络接口设置为混杂模式;to_ms 为第二个接口参数 times 所定义的超时;ebuf 用来返回错误信息,只有在函数调用失败并返回 NULL 时才设置。函数使用 pcap_dump_open( ) 来打开一个保存数据的文件,并往文件中写信息:

```
pcap_dumper_t * pcap_dump_open(pcap_t * p,
  const char * fname)
```

其中参数 p 是由 pcap_open_live( ) 返回的一个 pcap 结构;参数 fname 为接口参数 filename 所定义打开的文件名。

所捕获的数据包以二进制形式保存在文件中,这种格式是通用的,许多网络工具(如 Windump、Snort 等)都采用这种格式。

1.2.2 数据包读取接口

在检测控件中,提供了从二进制格式的文件中读取数据包的接口:

```
[id(2)] BSTR ReadPacket(BSTR filename);
```

接口要求用户提供文件名,并将其中内容解析成标准字符串类型返回给用户。接口的实现在控件的类成员函数中完成:

```
afx_msg BSTR ReadPacket(LPCTSTR filename)
```

函数将二进制格式的数据包按包结构中的顺序逐一转换成字符串,也可以通过选项设置接口,来指定所需要的数据包内容。

1.2.3 选项设置接口

检测控件提供选项设置接口来对数据包捕获和读取的内容进行设置:

```
[id(3)] short SetOption(BSTR filter, long readin);
```

该接口要求用户提供两个参数,其中 filter 为一字符串,用来设置要求捕获的数据包类型,其取值见表 1 所示。

表 1 参数 filter 的取值及含义

取值	包类型	取值	包类型
ip	IP 包	icmp	ICMP 包
tcp	TCP 包	udp	UDP 包
arp	ARP 包	rarp	RARP 包
ospf	OSPF 包	all	所有包

参数 readin 用来对读数据包的内容进行设置,其取值见表 2 所示。

表 2 参数 readin 的取值及含义

取值	读数据包内容	取值	读数据包内容
1	链路层帧头	3	1 +2
2	IP 包头	6	2 +4
4	传输层头	7	1 +2 +4
8	应用层数据	其他	1、2、4、8 组合

通过选项设置,可以根据用户需要捕获及读取数据包,提高数据包检测效率。

1.3 用回调函数对捕获的数据包进行处理

使用回调函数的编程模式,实现多种回调事件处理,可以使程序的控制灵活多变,也是一种高效率的、清晰的程序模块之间的耦合方式。

利用回调技术,可以对每个捕获到的数据包进行相应的处理。回调函数原型:

```
typedef void ( * pcap_handler)( u_char * user, const struct
  pcap_pkthdr * pkt_header, const u_char * pkt_data);
```

其中参数 user 是用户定义的包含捕获会话状态的参数;pkt_header 包含了捕获数据包的包头,这个包头并不是网络数据包的协议头;pkt_data 指向数据包中的数据,其中包含网络协议包头信息。

在控件的类成员函数中,对所处理的数据包及回调函数以进行设置。例如,在函数 GetPacket( ) 中,设置回调函数处理数据包:

```
pcap_loop(adhandle, 0, packet_handler, fname);
```

在函数 ReadPacket( ) 中设置回调函数读取并显示文件中的数据包:

```
pcap_loop(fp, 0, dispatcher_handler, NULL);
```

pcap_loop( ) 的第三个参数即为相关的回调函数。在不同的回调函数中实现不同的数据包处理。

2 检测控件在不同语言编程中的应用

由于检测控件是基于 ActiveX 技术,因此它允许在不同语言编写的程序中被重复使用。检测控件和应用程序的连接是通过其接口实现的。下面分别说明在 VC、VB、Delphi 及 PowerBuilder 中使用该控件的应用。

2.1 在VC++6.0 中使用该控件

在VC++ 中首先插入该控件,并创建一个类成员变量 m_a,调用控件的接口函数捕获数据包如下所示:

```
void CTestDlg: : OnButton1( )
 {
  int n;
  …
  if ((n = m_a. GetPacket(65535, 1000, "d: \\tt")) = -1)
   {
    AfxMessageBox("Error in Get Packet!");
    return;
   }
  …
 }
```

2.2 在 VB6.0 中使用该控件

在 VB 中,首先通过 project 下的"components"导入该控件,然后在应用程序中插入控件,调用控件的接口函数捕获数据包如下所示:

(下转第 1869 页)

是买卖双方在 14 次互协调匹配过程中 R' 中的目标与自身匹配意图相似度的平均值变化过程。

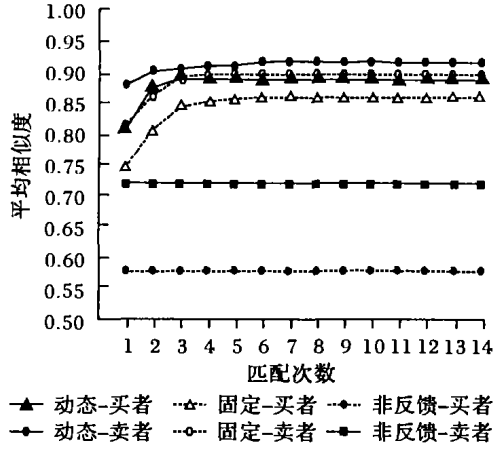

图 2　平均相似度变化曲线

表 1　第 3 号卖者匹配规格的变化情况

匹配次数	$a1$	$a2$	$a3$	$a4$	$a5$	相似度
意图	0.691	0.320	0.612	0.133	0.561	--
初值	0.330	0.650	0.000	0.990	0.750	0.605
1	0.369	0.261	0.102	0.146	0.844	0.751
2	0.376	0.258	0.151	0.285	0.699	0.853
3	0.391	0.269	0.187	0.338	0.602	0.898
4	0.397	0.284	0.213	0.353	0.538	0.919
5	0.395	0.285	0.238	0.366	0.488	0.926
6	0.396	0.287	0.257	0.371	0.452	0.928
7	0.396	0.291	0.270	0.374	0.426	0.927
8	0.397	0.295	0.279	0.377	0.408	0.925
9	0.398	0.298	0.285	0.379	0.395	0.923
10	0.399	0.301	0.289	0.380	0.386	0.922
11	0.399	0.304	0.292	0.380	0.380	0.922
12	0.399	0.305	0.294	0.381	0.376	0.921
13	0.400	0.306	0.295	0.381	0.372	0.921
14	0.400	0.307	0.296	0.381	0.370	0.921

由图2可以看出，采用互适应方法后买卖双方的平均相似度比不采用互适应方法有显著提高，并且在有限次交互后，平均相似度有收敛趋势。另外，采用动态方法确定锁定阈值能够得到更大的平均相似度，这是因为 μ 的动态计算过程本身就是向着大相似度方向逼近的。

随着 R' 中平均相似度的提高，产生的匹配规格也将逐步得到改善。表 1 是 3 号卖者匹配规格的变化情况。可以看出匹配规格与匹配意图的相似度在逐步提高，在第 6 次匹配后达到最大相似度，随后略有下降，但逐步稳定在 0.921 左右。

2.3　实验结论

除了本文列出的实验结果数据外，实验中还使用不同的数据以及不同的 α、β 值进行了多次实验，得到的实验结果所表现的特征基本相似。由此可以看出，该互适应匹配机制是可以提高匹配成功率的。

3　结语

论文提出了以反馈为基础的互适应匹配机制。利用该机制可以使模糊的匹配需求逐步得到优化，并接近于匹配意图，从而提高匹配相似度。作为一个可行的匹配方案，该机制目前已应用于 eHRM 项目电子中介模块的开发，后期工作主要是将该匹配模块标准化、通用化，以适应不同匹配需求。

参考文献：

[1] VEIT D. Multidimensional Matchmaking for Electronic Markets[A]. Autonomous Agents and Multi - Agent Systems (AAMAS'04)[C]. New York, 2004.

[2] 郭庆. 基于整合效用的多议题协商优化[J]. 软件学报, 2004, 15(5): 706 - 711.

[3] Prithviraj(Raj) Dasgupta, Yoshitsugu. Hashimoto. Multi-attribute Dynamic Pricing for Online Markets using Intelligent Agents[A]. Autonomous Agents and Multi-agent Systems(AAMAS'04)[C]. New York, 2004.

[4] VMaintaining Buyer-Supplier Partnership[J]. International Journal of Purchasing and Materials Management, 1995, 31(3): 311.

[5] SANDHOLM T. eMediator: A NEXT GENERATION ELECTRONIC COMMERCE SERVER[J]. Computational Intelligence, 2002, 18(4): 656 - 676.

(上接第 1866 页)

```
Private Sub Command1_Click()
  ...
  n = PktControl1. GetPacket(65535, 1000, "d: \tt")
  ...
End Sub
```

2.3　在 Delphi6 中使用该控件

在 Delphi 中，首先通过 Component 下的“Import ActiveX Control”导入该控件，然后在应用程序中插入控件，调用控件的接口函数捕获数据包如下下所示：

```
procedure TForm1. Button1Click(Sender: TObject);
VAR
n: integer;
begin
  ...
  n: = PktControl1. GetPacket(65535, 1000, d: \tt);
  ...
end;
end.
```

2.4　在 PowerBuilder9 中使用该控件

首先在 PB 中创建一个窗口 w_1，把 PktControl. ocx 嵌入到 w_1 中，命名为 ole_1，执行下列语句捕获数据包如下所示：

```
...
Ole_1. object. getpacket(65535, 1000, "d: \tt")
...
```

从应用可以看出，检测控件可在任何支持 ActiveX 控件的开发环境中使用。应用程序的开发人员无须了解控件的功能如何实现，而只须创建控件对象与网络检测控件的接口建立连接。

3　结语

网络检测控件封装了底层网络编程的细节，利用 ActiveX 技术使控件可以用于所有拥有 Control Container 功能的应用程序，并且给应用程序开发人员提供友好接口，大大缩短网络应用程序的开发时间。

参考文献：

[1] (美) KRUGLINSKI DJ. VISUAL C++6.0 技术内幕[M]. 北京: 希望电子出版社, 2001.

[2] 殷肖川, 刘志宏, 姬伟辉, 等. 网络编程与开发技术[M]. 西安: 西安交通大学出版社, 2003.

[3] 卿斯汉, 蒋建春. 网络攻防技术原理与实战[M]. 北京: 科学出版社, 2004.