**网络嗅探器实验报告**

201128013229083 虞航仲 计算所网络技术研究中心

**2011/10/7**

**目 录**

[第一章 程序关键算法和流程图 1](#_Toc263955547)

[1.1两个核心过程的基本算法 1](#_Toc263955548)

[1.2流程图 2](#_Toc263955551)

[第二章 用SharpPcap写嗅探器的程序框架的总结 3](#_Toc263955555)

[第三章 遇到的问题以及解决方法 4](#_Toc263955559)

[第四章 收获与体会 28](#_Toc263955572)

[第五章 程序界面和运行结果 38](#_Toc263955580)

[第六章 程序安装说明以及压缩包文件清单说明 38](#_Toc263955580)

# 第一章 程序关键算法和流程图

## 1.1两个核心过程的基本算法

1.1.1 抓包算法

第一：初始化SharpPcap开发库

第二：获得当前的网卡列表，同时要求用户指定要操作的网卡

第三：设置当前的过滤规则，可为空

第四：开一个thread在系统idle的时候调用库函数GetNextPacket（），同时并指定数据包分析过程，显示界面同步的显示抓得到的包被分析后的结果。

第五：结束算法。

1.1.2 数据包分析算法

第一：得到数据包，存到内存。

第二：分析当前数据包，分析过程如下：

1.有些SharpPcap库函数提供的属性，就直接调用库函数获得（如IP数据包的version属性，可直接调用IPPacket.Versioin()获得）。

2.那些SharpPcap没有提供库函数可获取的属性，就分析其byte获得。以IP数据包的Flags为例：获取ipPacket.Header[6].toString(),此时是字符串类型，本程序先用一下算法将字符串转化为数字：

private int ConvertString2Int(string str)

{

int flagvalue = 0;

for (int index = 0; index < str.Length; ++index)

{

flagvalue \*= 10;

flagvalue += (str[index] - '0');

}

return flagvalue;

}

此时，得到的数字就是Flags字段的10进制表示，然后用位运算获取相应属性，如(flagvalue & 128)可以判断最高位是否唯一，用(flagvalue & 64)判断最高第二位是否为1等，依次类推，可快速的获取个属性。

第三：结束分析，将分析结果交给ViewModel层处理，最后又View层显示（本程序采用MVVM 设计模式，即Model- View -ViewModel）。

## 1.2流程图

获取网卡列表

设置过滤器

用户终止程序

是否获取到了新数据包

是

分析该包并显示

# 第二章 SharpPcap的程序框架总结

# 1. 加载PacketDotNet.dll和SharpPcap.dll两个dll,引用一下命名空间：SharpPcap 2. 获取当前的网卡列表，SharpPcap库函数： CaptureDeviceList devices = CaptureDeviceList.Instance; 3. 设置过滤规则，库函数： 以只抓TCP或者IP包为例。 ICaptureDevice device = devices[index]; string filter = “ip and tcp”; device.Filter = filter; 4. 打开一个网卡，设置模式，开始抓包，库函数： device.Open(DeviceMode, ReadTimeoutMilliseconds); void GetNextPacket() { packet = device.GetNextPacket(); } 我开了一个单独的线程在系统idle的时候开始抓包， this.Dispatcher.BeginInvoke(DispatcherPriority.SystemIdle, New GetNextPacketDelegate(GetP)); 5. 关闭网卡： device.Close();

# 

# 第三章 遇到的问题以及解决方法

问题1：如何实现一边抓包一边显示抓到包的数据

解决方法：多线程(c# 中用dispatcher)

问题 2: 多线程有时会使显示界面响应的时间变得很长

解决方法，抓包的线程优先级设的最低，只有在系统idle的时候才执行。

问题3：数据包的离线存储和读取，尤其是数据包的存储格式的设计  
解决方法：

找到了CaptureFileReaderDevice和CaptureFileWriterDevice 两个库函数，它们已经很好的设计了数据包的存储格式。

问题4：分析http协议。

解决方法：获取TCP或者UDP数据包的字节，然后转化为ASCII码。

问题5：把字节转化为 ASCII

解决方法，调用System.Text.ASCIIEncoding.ASCII.GetString(Bytes);

问题6：treeview绑定数据包的分析数据

解决方法：用了Model-View-ViewModel设计模式。

问题7：列出抓到的数据包，

解决方法：用ListView控件

问题8： 显示数据包的16进制信息的时候，因为很多字体数字和字母的显示大小不同，导致显示的时候列对不起

解决方法：用了字体Courier New，是字母和数字显示尺寸相同。

问题9： 如果定位到用户想去的那个数据包

解决方法： 查看了MSDN,发现listview支持自己选中随便一个ITEM。

问题10： 如何快速的把一个表示数字的字符串转化为数字。

解决方法：自己实现了该算法：

private int ConvertString2Int(string str)

{

int flagvalue = 0;

for (int index = 0; index < str.Length; ++index)

{

flagvalue \*= 10;

flagvalue += (str[index] - '0');

}

return flagvalue;

}

问题11：如何判断一个数字的二进制表示的某一位是1还是0

解决方法：用位运算，如(flagvalue & 128)可以判断最高位是否唯一，用(flagvalue & 64)判断最高第二位是否为1等，依次类推

# 第四章 收获与体会

# 第五章 程序界面和运行结果