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. 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：

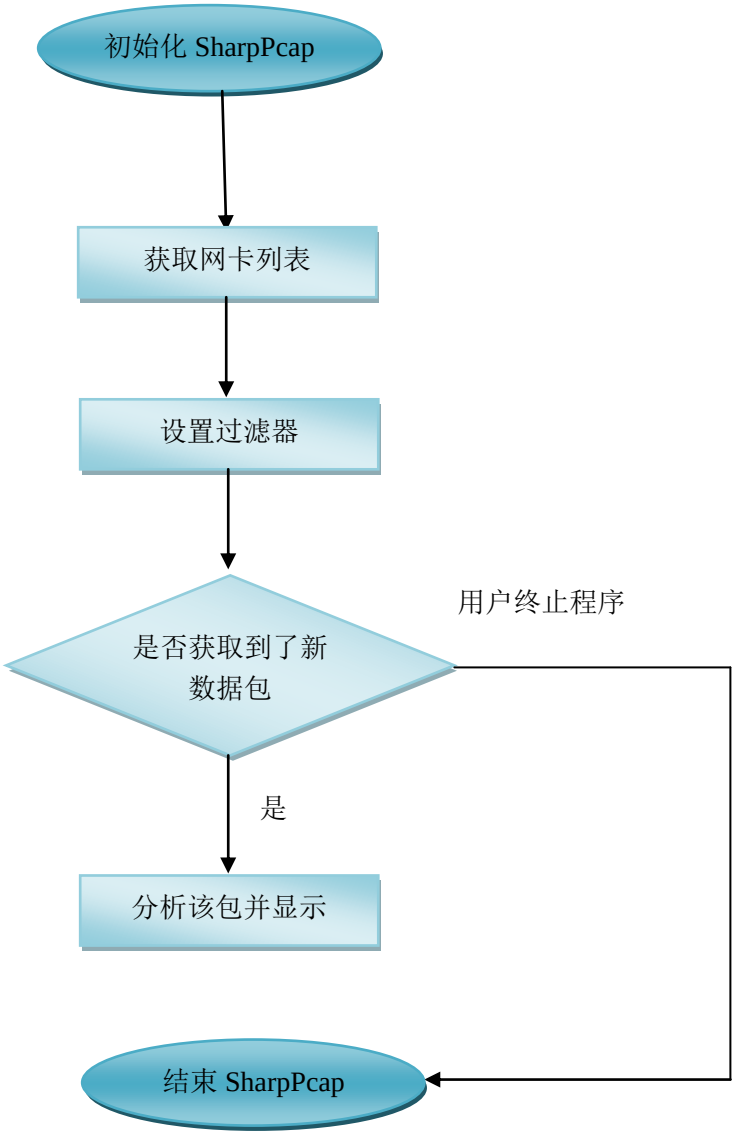
```
private int ConvertString2Int(string str)
{
    int flagvalue = 0;
    for (int index = 0; index < str.Length; ++index)
    {
        flagvalue *= 10;
        flagvalue += (str[index] - '0');
    }

    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


初始化 SharpPcap
获取网卡列表
设置过滤器
是否获取到了新
数据包
用户终止程序
是
分析该包并显示
结束 SharpPcap

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;
}
```

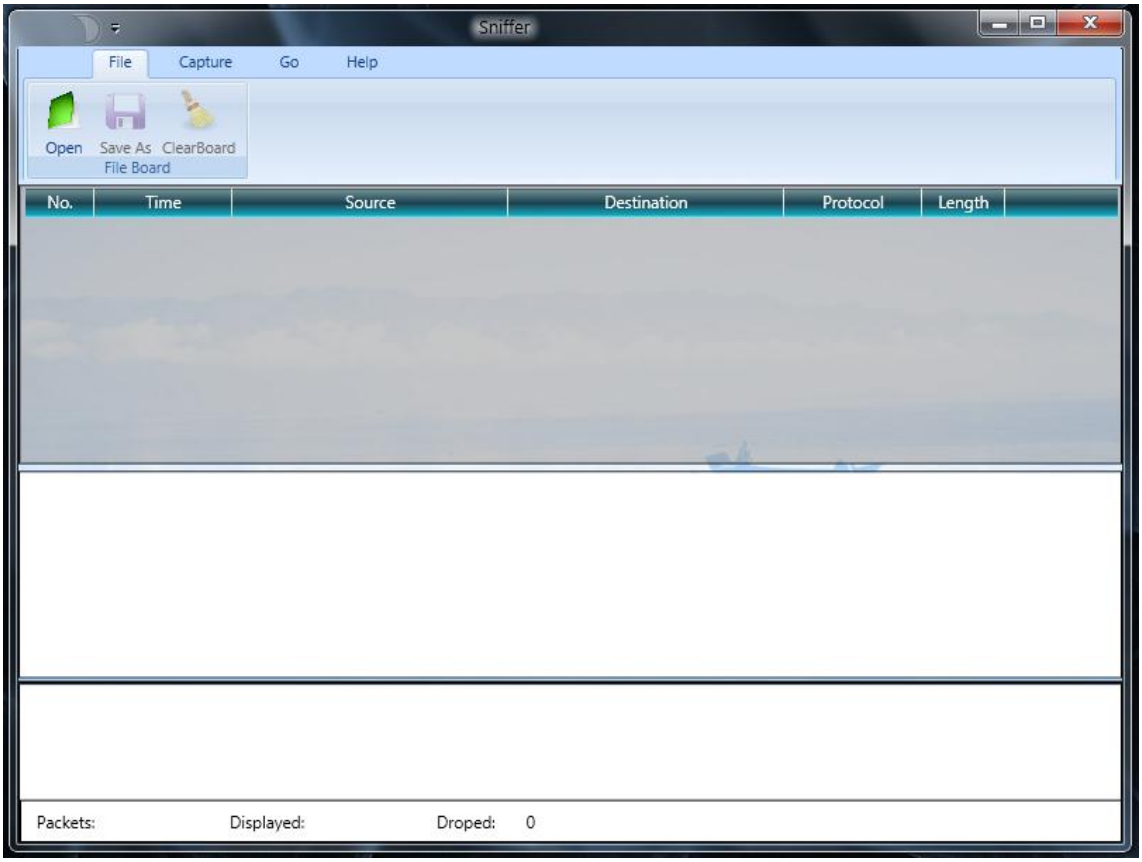
问题11：如何判断一个数字的二进制表示的某一位是1还是0
解决方法：用位运算，如(flagvalue & 128)可以判断最高位是否唯一，用(flagvalue & 64)判断最高第二位是否为1等，依次类推
问题12： 控件Scoller不响应鼠标滑轮的滚动事件：
解决方法：发现是WPF 路由事件的问题，鼠标滑轮的滚动事件被前面的控件先处理了，我就用On方法来提前捕获该鼠标滑轮的滚动事件，人工让Scoller滚动条移动。

第四章　　收获与体会

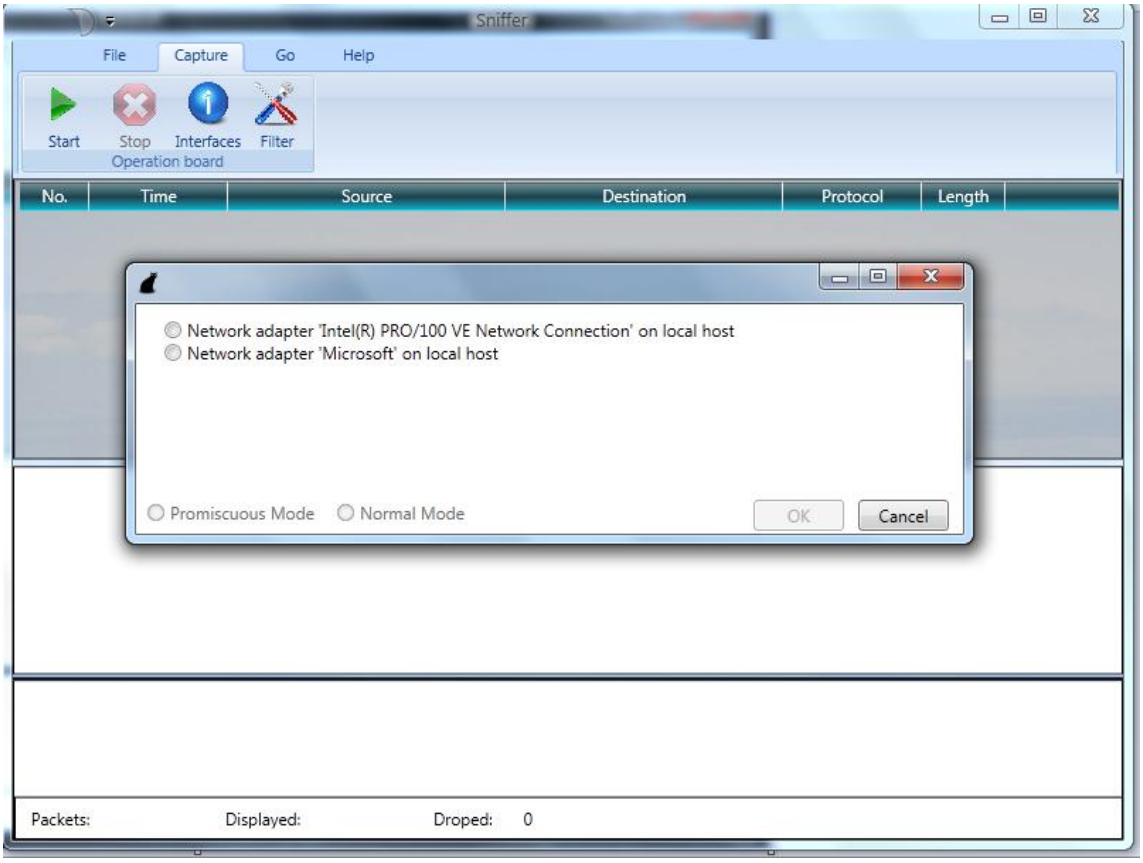
一定程度上加深了对 TCP/IP 协议的理解，对课程学习打下了一个比较好的基础；学习了 MVVM 设计模式，在一定程度提高了开发效能，降低开发的复杂度；

第五章　　程序界面和运行结果

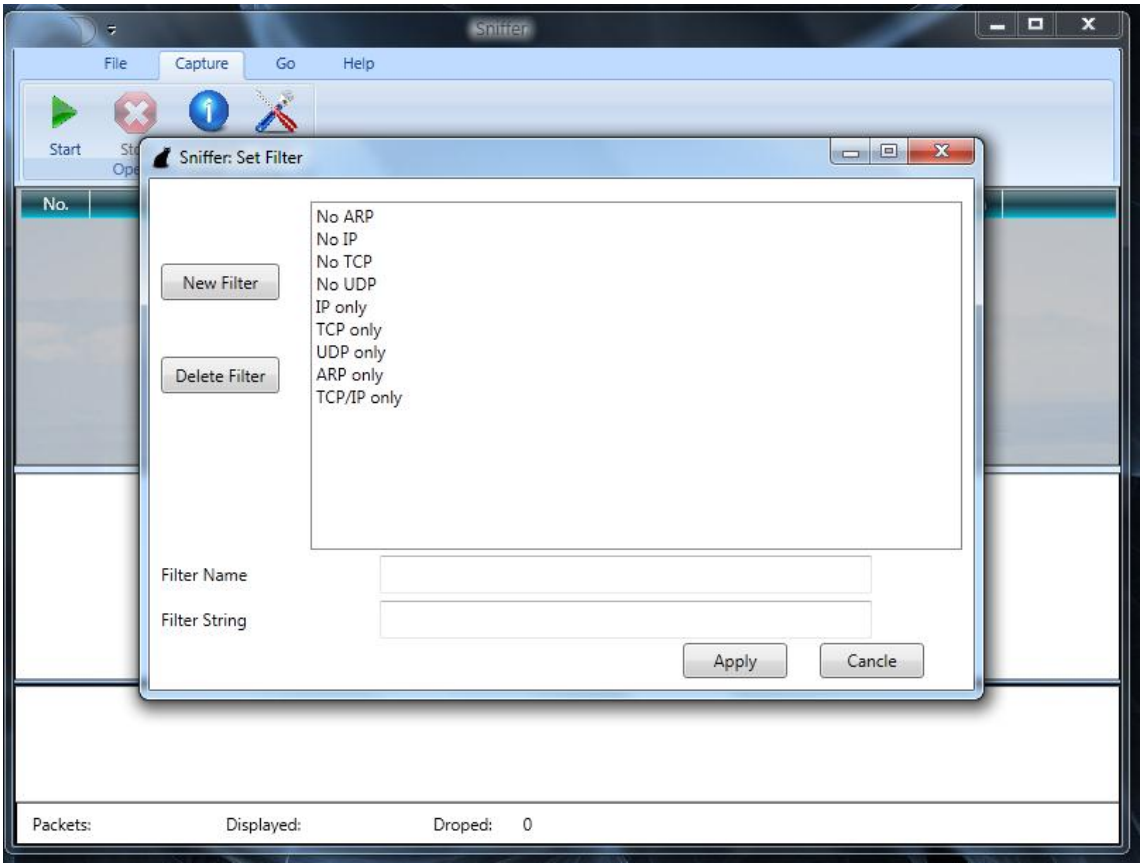
1.程序启动界面：



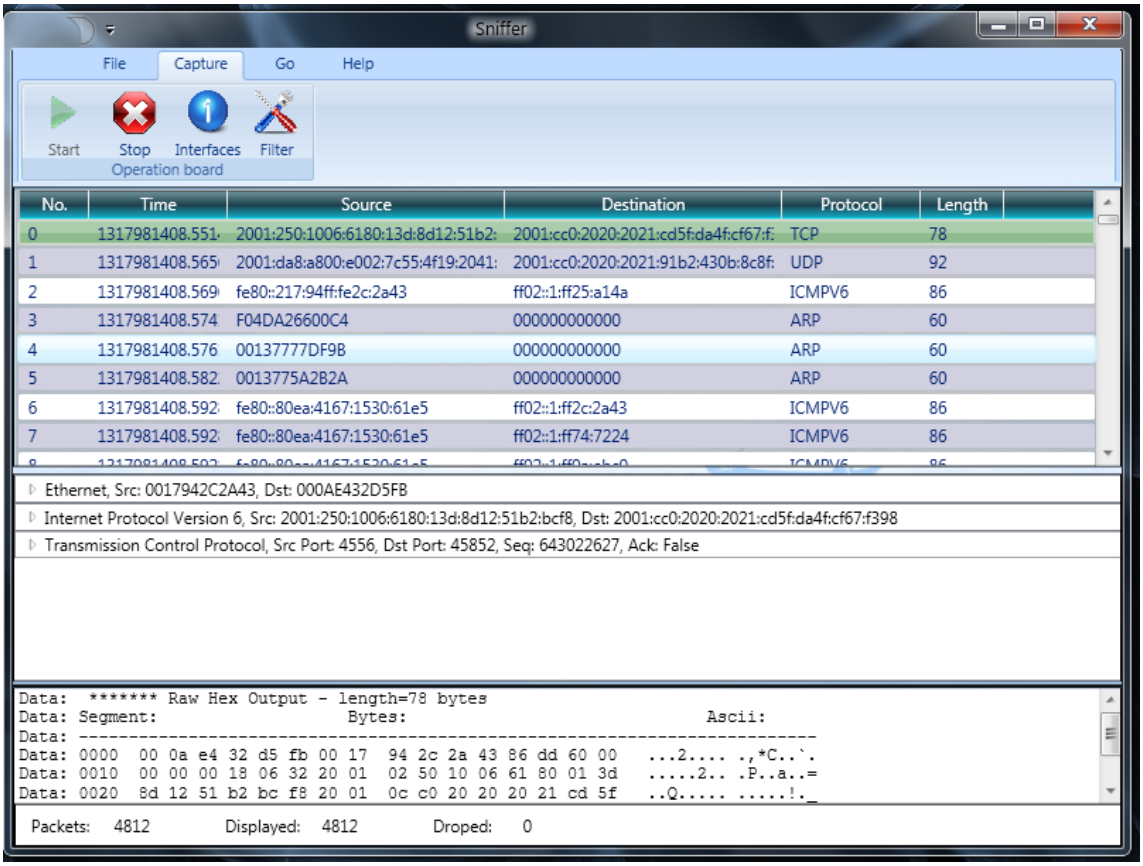
2. 列出本地所有网卡以及选择网卡的界面



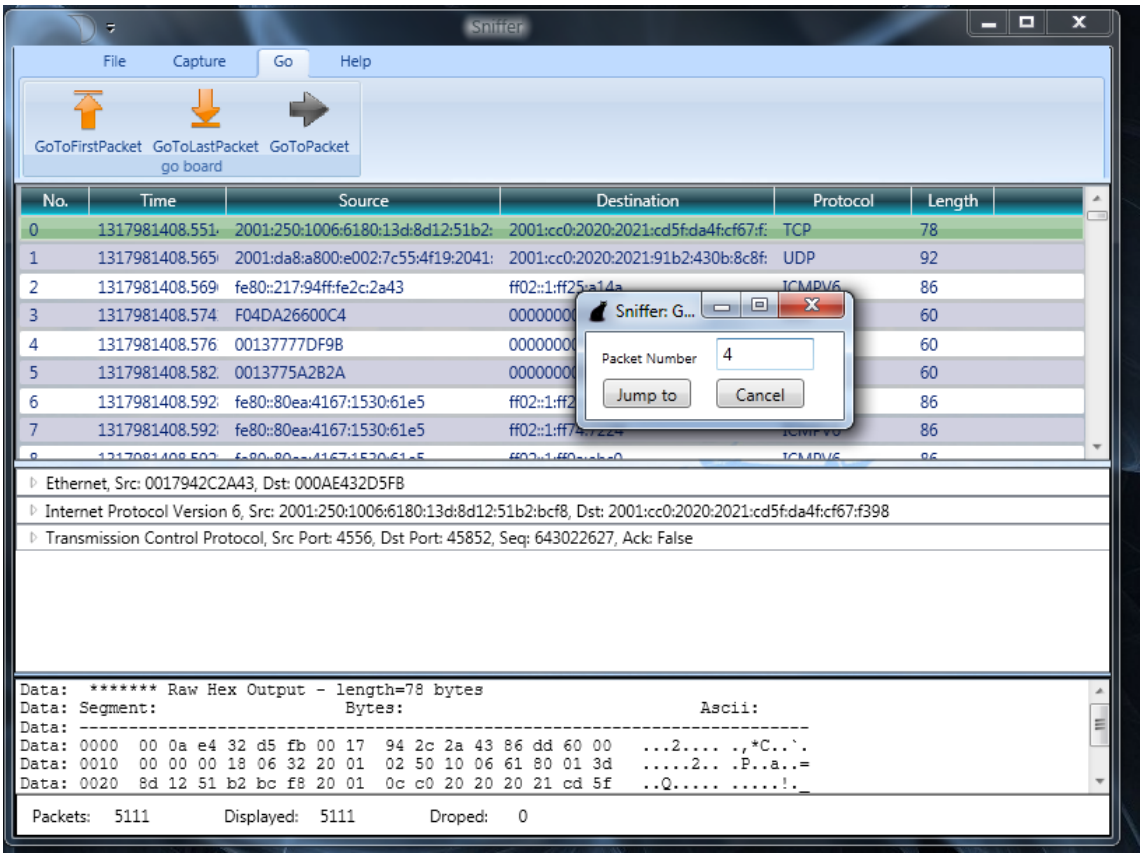
3. 设置过滤规则的界面


Sniffer
File
Capture
Go
Help
Start
No.
Sniffer: Set Filter
No ARP
No IP
No TCP
No UDP
IP only
TCP only
UDP only
ARP only
TCP/IP only
New Filter
Delete Filter
Filter Name
Filter String
Apply
Cancle
Packets:
Displayed:
Droped:
0

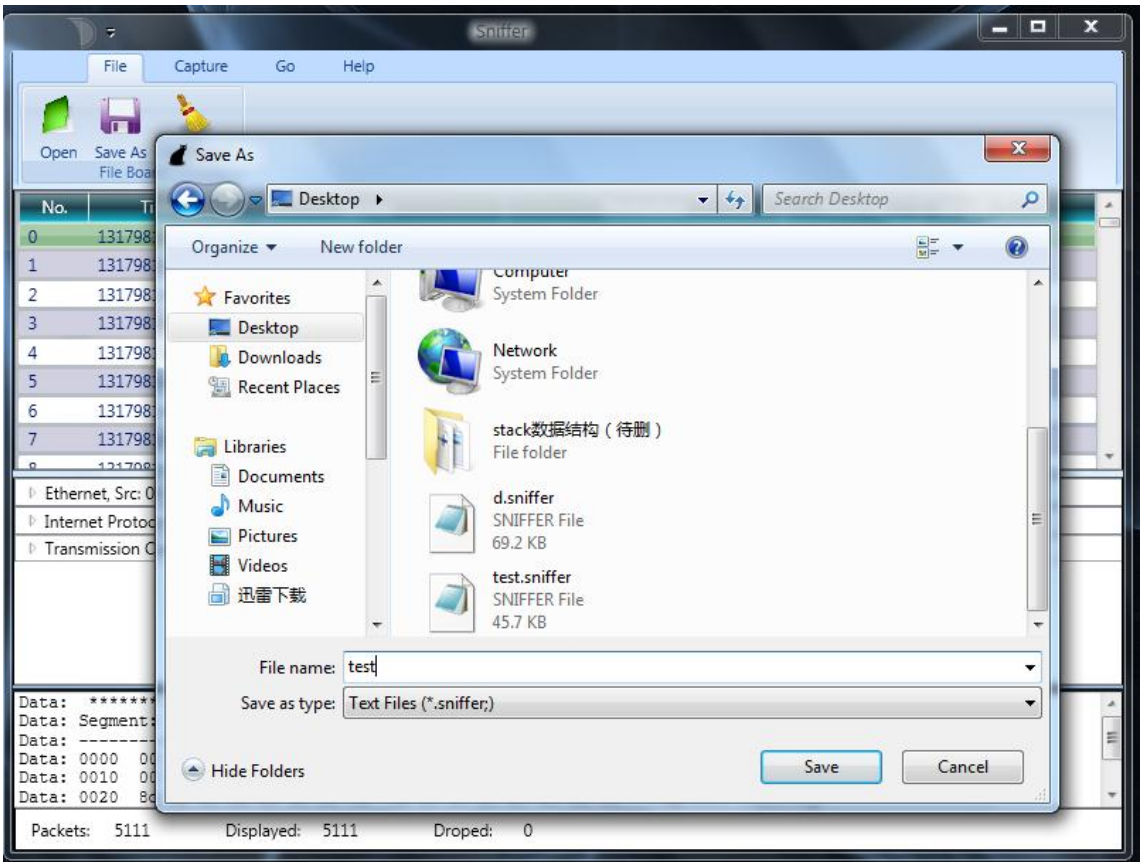
4.抓包过程界面：


Sniffer
File
Capture
Go
Help
Start
Stop
Interfaces
Filter
Operation board
No. Time Source Destination Protocol Length
0 1317981408.551 2001:250:1006:6180:13d:8d12:51b2: 2001:cc0:2020:2021:cd5f:da4f:cf67:f: TCP 78
1 1317981408.565 2001:da8:a800:e002:7c55:4f19:2041: 2001:cc0:2020:2021:91b2:430b:8c8f: UDP 92
2 1317981408.569 fe80::217:94ff:fe2c:2a43 ff02::1:ff25:a14a ICMPV6 86
3 1317981408.574 F04DA26600C4 000000000000 ARP 60
4 1317981408.576 00137777DF9B 000000000000 ARP 60
5 1317981408.582 0013775A2B2A 000000000000 ARP 60
6 1317981408.592 fe80::80ea:4167:1530:61e5 ff02::1:ff2c:2a43 ICMPV6 86
7 1317981408.592 fe80::80ea:4167:1530:61e5 ff02::1:ff74:7224 ICMPV6 86
Ethernet, Src: 0017942C2A43, Dst: 000AE432D5FB
Internet Protocol Version 6, Src: 2001:250:1006:6180:13d:8d12:51b2:bcf8, Dst: 2001:cc0:2020:2021:cd5f:da4f:cf67:f398
Transmission Control Protocol, Src Port: 4556, Dst Port: 45852, Seq: 643022627, Ack: False
Data: ******* Raw Hex Output - length=78 bytes
Data: Segment: Bytes: Ascii:
Data: ----------
Data: 0000 00 0a e4 32 d5 fb 00 17 94 2c 2a 43 86 dd 60 00 ...2.... .,*C..`.
Data: 0010 00 00 00 18 06 32 20 01 02 50 10 06 61 80 01 3d .....2.. .P..a..=
Data: 0020 8d 12 51 b2 bc f8 20 01 0c c0 20 20 20 21 cd 5f ..Q..... .....!._
Packets: 4812 Displayed: 4812 Droped: 0

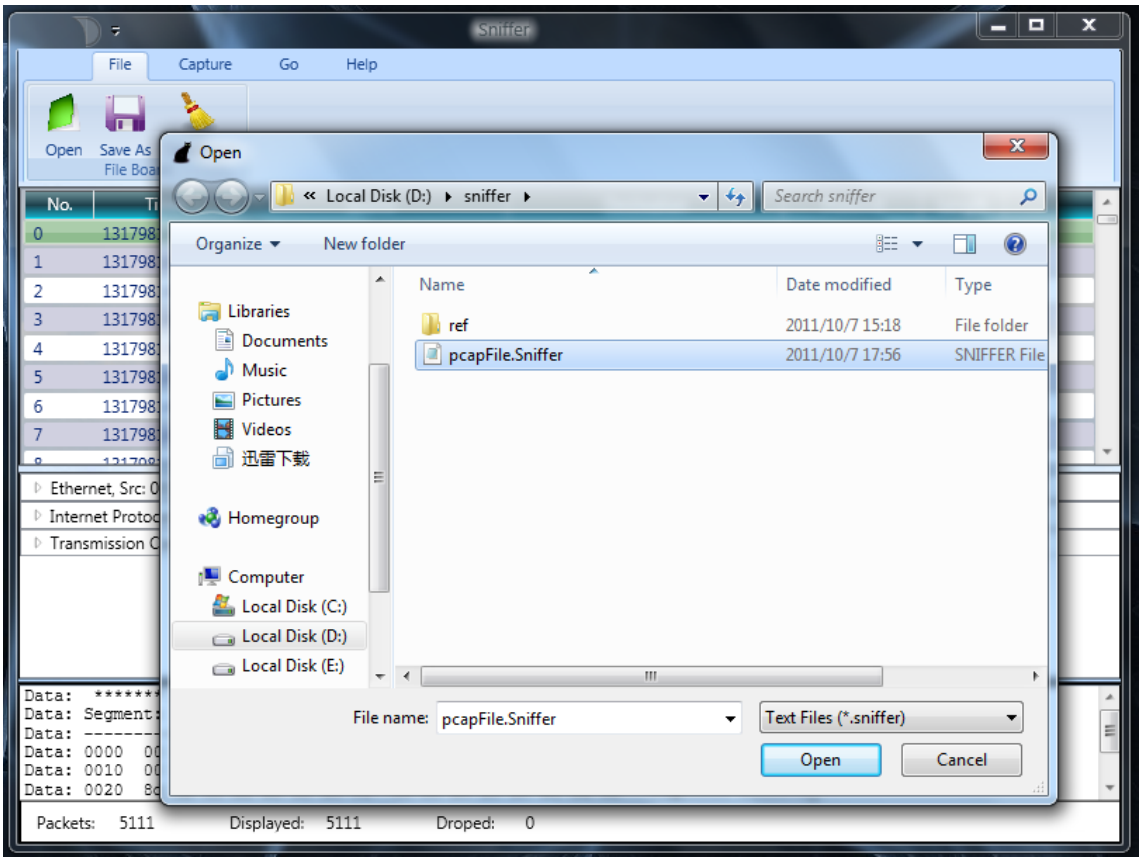
5.定位到某个特定的数据包或者第一个或最后一个数据包的界面：



6.另存文件到本地的界面



7.从本地加载离线文件的界面



第六章　　程序安装说明以及压缩包文件清单说明

6.1 程序安装说明：

1：安装此安装包的系统必须先安装.NET framework 3.5
因为windows 7默认安装.NEt framework3.5,所以在Windows 7下可以直接双击该安装包安装。Windows Xp必须先安装.net framework 3.5,双击改安装包时会提示安装。因为太大，我就没下载，下面是网址：

1. Microsoft .NET Framework 4（独立安装程序）：
   （http://www.microsoft.com/downloads/zh-cn/details.aspx?FamilyID=0a391abd-25c1-4fc0-919f-b21f31ab88b7）
2. Microsoft .NET Framework 3.5：
   （http://www.microsoft.com/downloads/zh-cn/details.aspx?FamilyID=333325fd-ae52-4e35-b531-508d977d32a6）

6.2 压缩包文件清单说明

该压缩包内包括 4 个文件，1. Sniffer 工程所有源代码，2. Sniffer 安装包，3. Sniffer Setp 工程,以及该 4. 实验报告(两份，分别是 word03, pdf 两种格式)。