

Wireshark 的基本操作

张元茂
2015年11月

说明：本PPT中的所用的Wireshark图片，部分截自本机新版软件，部分收集自网络的为陈旧版本，但不影响操作。

- 通过本课程的学习，您将能够：
 - 了解Wireshark及其界面组成
 - 熟悉Wireshark的基本操作
 - 熟悉Wireshark两种过滤器的使用
- 适用对象：
 - 计算机网络课程实验学生
 - 测试、开发、网络工程人员

Wireshark简介

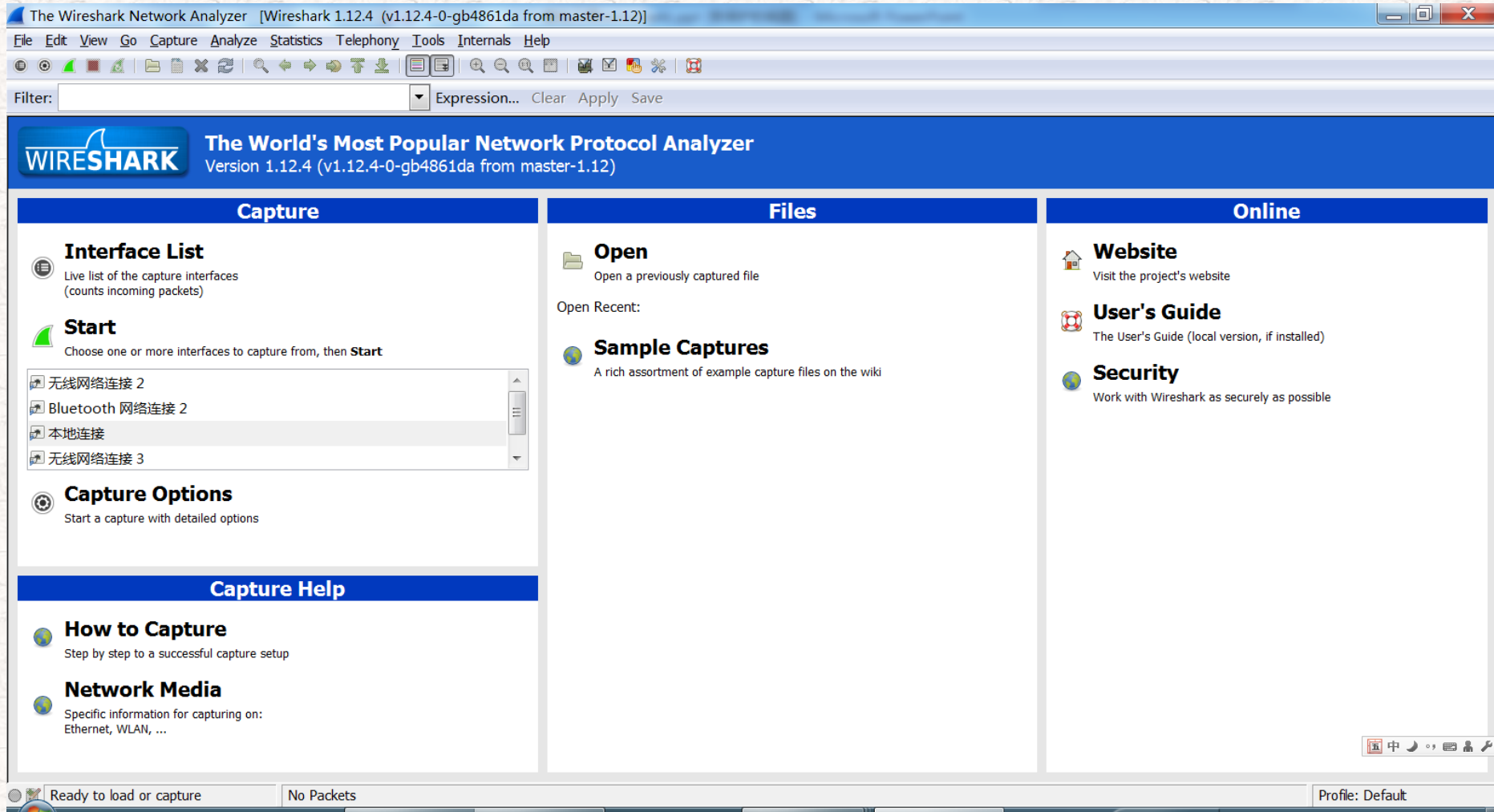
- **Wireshark**是世界上最流行的网络分析工具，很强大。可以捕捉网络中的数据，并为用户提供关于网络 and 上层协议的各种信息。与很多其他网络工具一样，**Wireshark**也使用**pcap network library**来进行封包捕捉。
- **Wireshark**的原名是**Ethereal**，新名字是**2006**年起用的。当时**Ethereal**的主要开发者决定离开他原来供职的公司，并继续开发这个软件。但由于旧名称**Ethereal**已被公司注册，所以改为了新名字**Wireshark**。
- 官方网站：<http://www.wireshark.org/>

- **Wireshark操作的一般步骤:**

- ① 选择网络适配器
- ② 设置捕获过滤器（可省略此步）
- ③ 点击**Start**开始捕获
- ④ 离开Wireshark操作其它网络程序
- ⑤ 返回Wireshark，等待0~N分钟
- ⑥ 点击**Stop**停止捕获
- ⑦ 对捕获到的数据包进行分析（可根据需要设置显示过滤器）
- ⑧ 退出或者执行下一次捕获（可选择保存到文件）

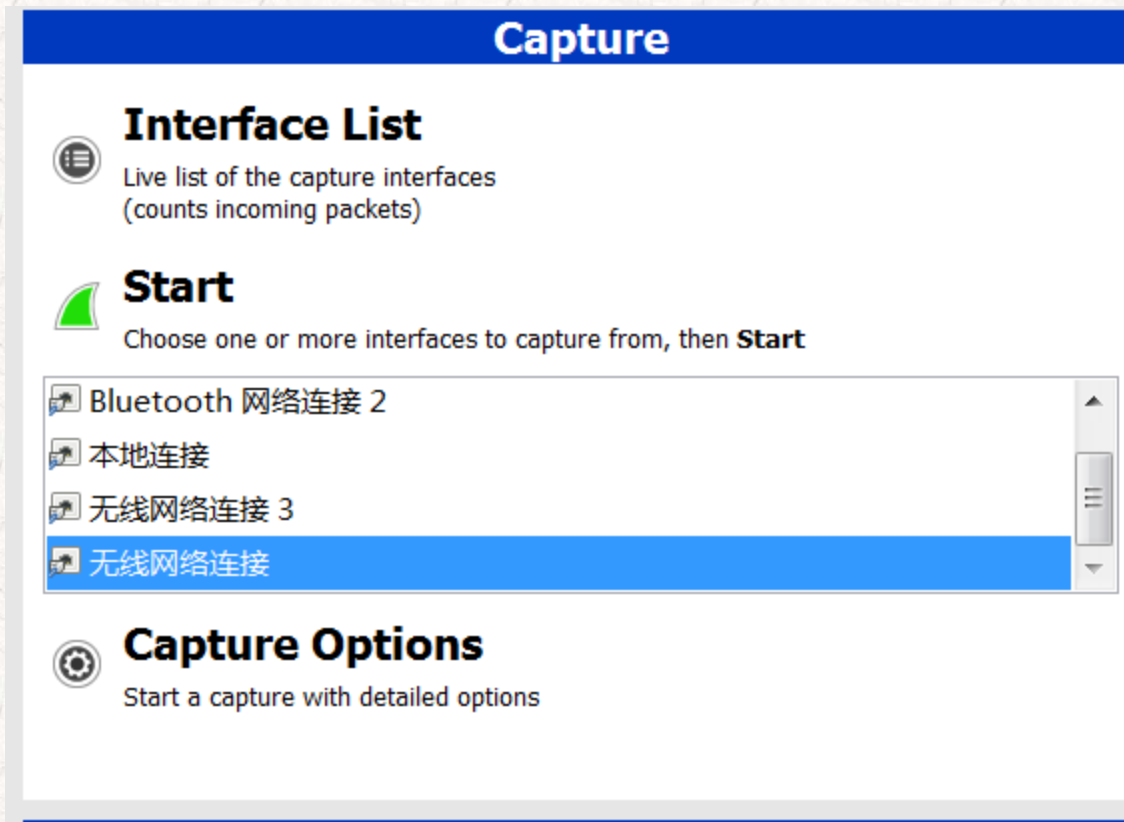
Wireshark的启动主界面（版本 V.1.12.4）

- 本课程实验采用的WireShark版本

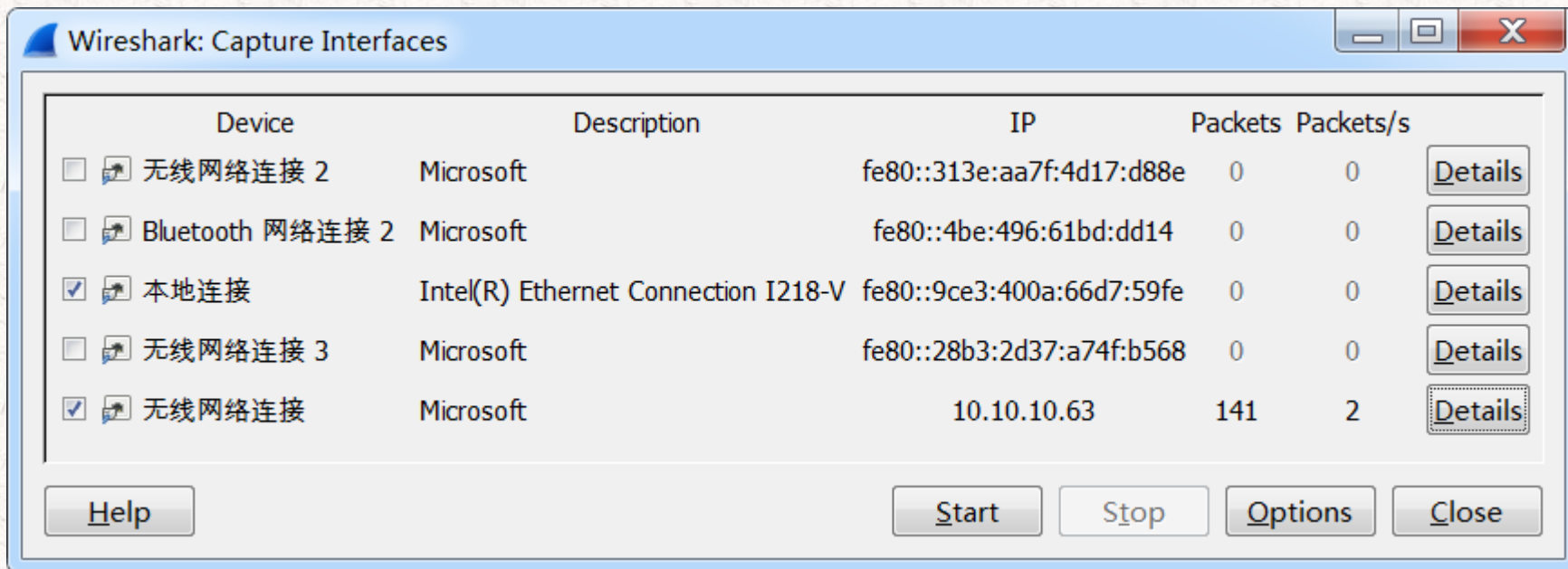


基本操作过程说明:

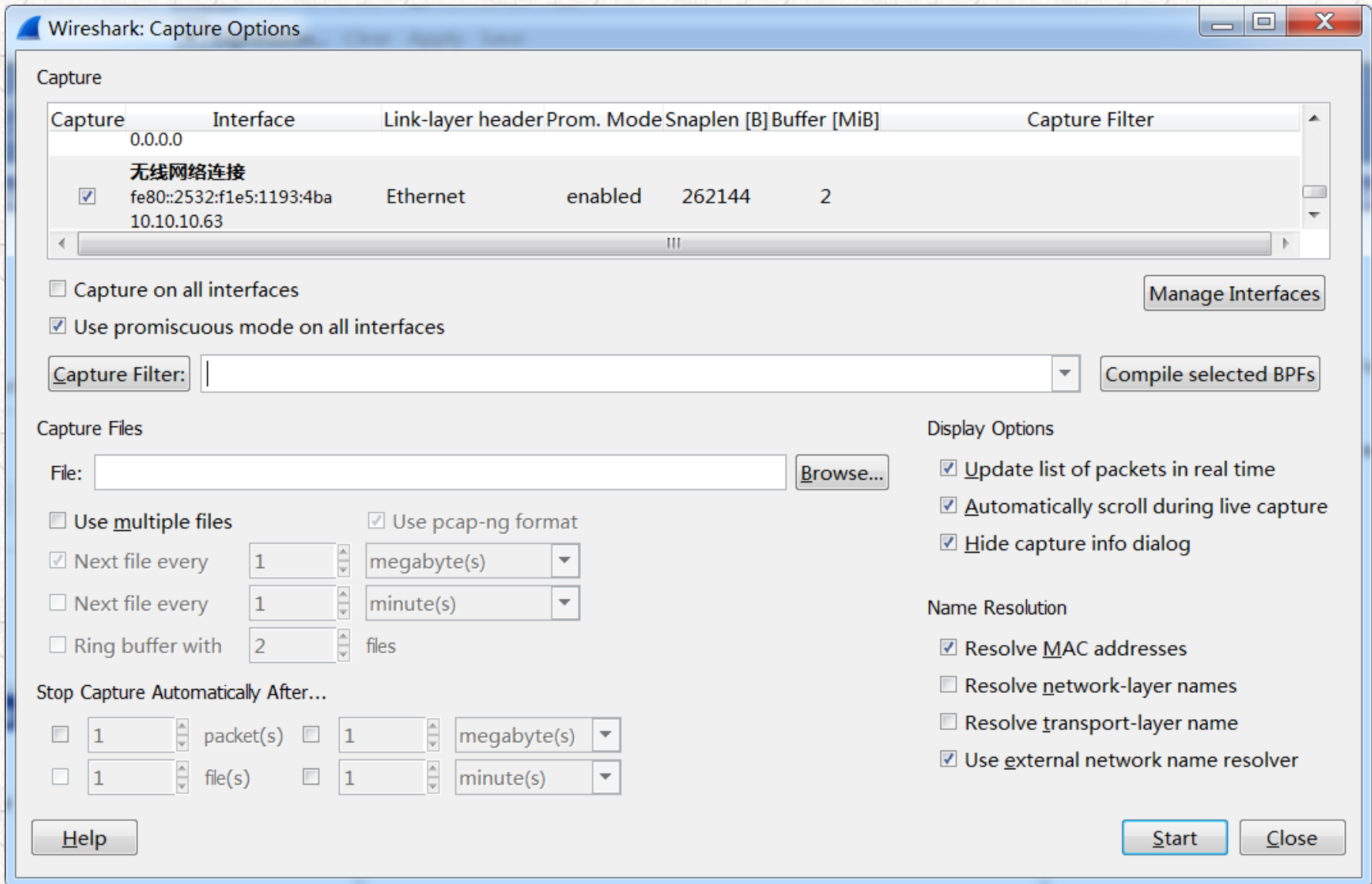
- 主界面中的左上区域如下图所示:



- 点击  **Interface List** 项, 弹出如下界面:



- 勾选要监视的网络接口，如上图中“无线网络连接”是本机对外连接的网络适配器接口，点击后IP地址及Packets信息有变化。请根据实验主机实际情况选择。
- 在点击“Start”启动之前，可以通过“Options”设置数据包的捕获条件。点击“Options”后弹出如下窗口：



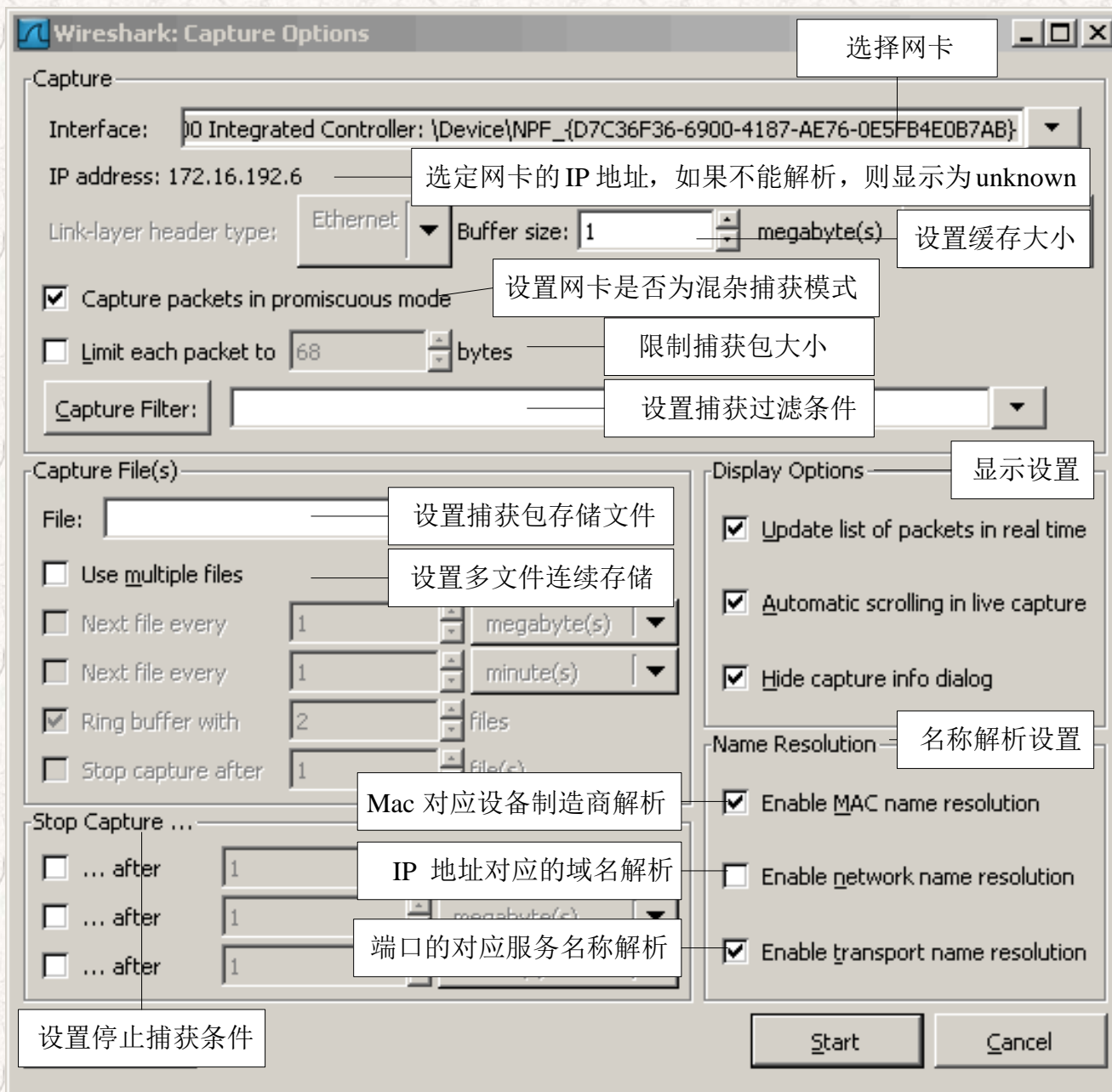
- 在此可以设置捕获过滤器和各种捕获选项。（也可以在主界面中点击  **Capture Options** 弹出本窗口）




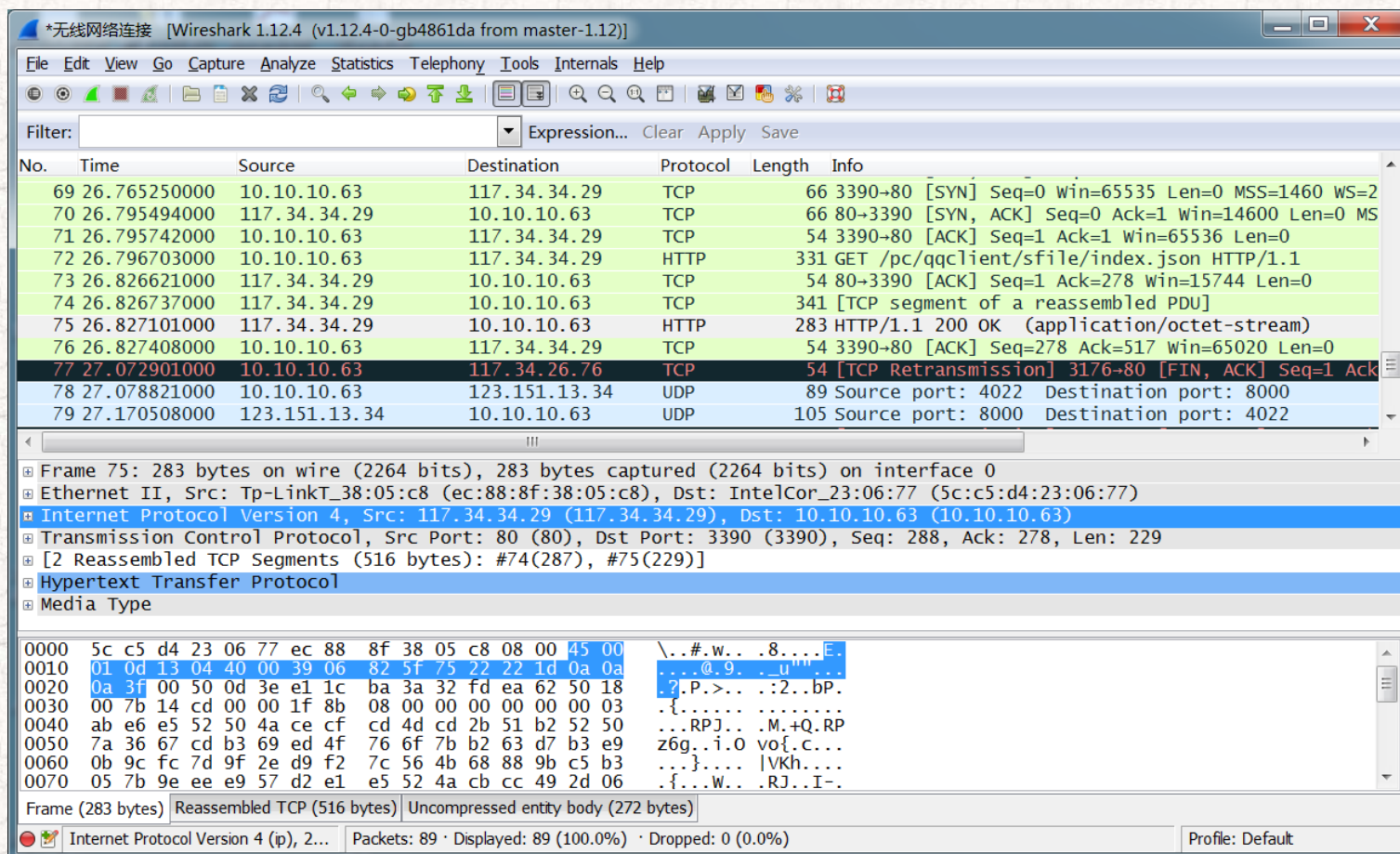
Capture Options

Start a capture with detailed options

旧版本“Capture Options”对话框各项设置含义参考：



- 如果暂时不需要设置捕获过滤条件和其它选项，可以采用默认值。
- 然后点击“**Start**”按钮（启动数据包捕获过程。这时可以执行IE等其它网络程序，当捕获到一定数量的数据包后点击**停止按钮**  停止捕获过程（快捷键**Ctrl+E**）。如下为工作主窗口界面：



Wireshark工作主窗口介绍（旧版，参考）

The diagram illustrates the Wireshark interface with the following components labeled:

- 菜单栏** (Menu Bar): Located at the top left, containing File, Edit, View, Go, Capture, Analyze, Statistics, and Help.
- 工具栏** (Toolbar): Located below the menu bar, containing icons for file operations, capture, and analysis.
- 过滤器** (Filter): Located below the toolbar, containing a text input field for filters and buttons for Expression..., Clear, and Apply.
- 包概况显示窗体** (Packet List Window): A table showing captured packets with columns: No., Time, Source, Destination, Protocol, and Info.
- 协议树显示窗体** (Protocol Tree Window): A tree view showing the hierarchy of the selected packet, including Ethernet II and IP.
- 数据显示窗体** (Packet Details Window): A window showing the raw data of the selected packet, including hexadecimal and ASCII representations.
- 状态栏** (Status Bar): Located at the bottom, showing file path, packet count, and profile information.

包序号	捕获时间	源地址	目的地址	上层协议	包内容提要
1	0.000000	172.16.37.246	172.16.39.32	UDP	Source port: 60050
2	0.004388	172.16.37.246	172.16.39.32	UDP	Source port: 60050
3	0.004392	08:11:32:ff:99:11	Broadcast	ARP	who has 192.168.1.
4	0.007909	172.16.37.246	172.16.39.32	UDP	Source port: 60050
5	0.010839	172.16.37.246	172.16.39.32	UDP	Source port: 60050
6	0.013529	172.16.37.246	172.16.39.32	UDP	Source port: 60050
7	0.016858	172.16.37.246	172.16.39.32	UDP	Source port: 60050
8	0.020865	172.16.37.246	172.16.39.32	UDP	Source port: 60050
9	0.023624	172.16.37.246	172.16.39.32	UDP	Source port: 60050
10	0.026467	172.16.37.246	172.16.39.32	UDP	Source port: 60050
11	0.030372	172.16.37.246	172.16.39.32	UDP	Source port: 60050
12	0.032315	172.16.37.246	172.16.39.32	UDP	Source port: 60050
13	0.037211	172.16.37.246	172.16.39.32	UDP	Source port: 60050
14	0.037218	172.16.37.246	172.16.39.32	UDP	Source port: 60048
15	0.041125	172.16.37.246	172.16.39.32	UDP	Source port: 60050
16	0.044047	172.16.37.246	172.16.39.32	UDP	Source port: 60050
17	0.046976	172.16.37.246	172.16.39.32	UDP	Source port: 60050
18	0.049911	172.16.37.246	172.16.39.32	UDP	Source port: 60050

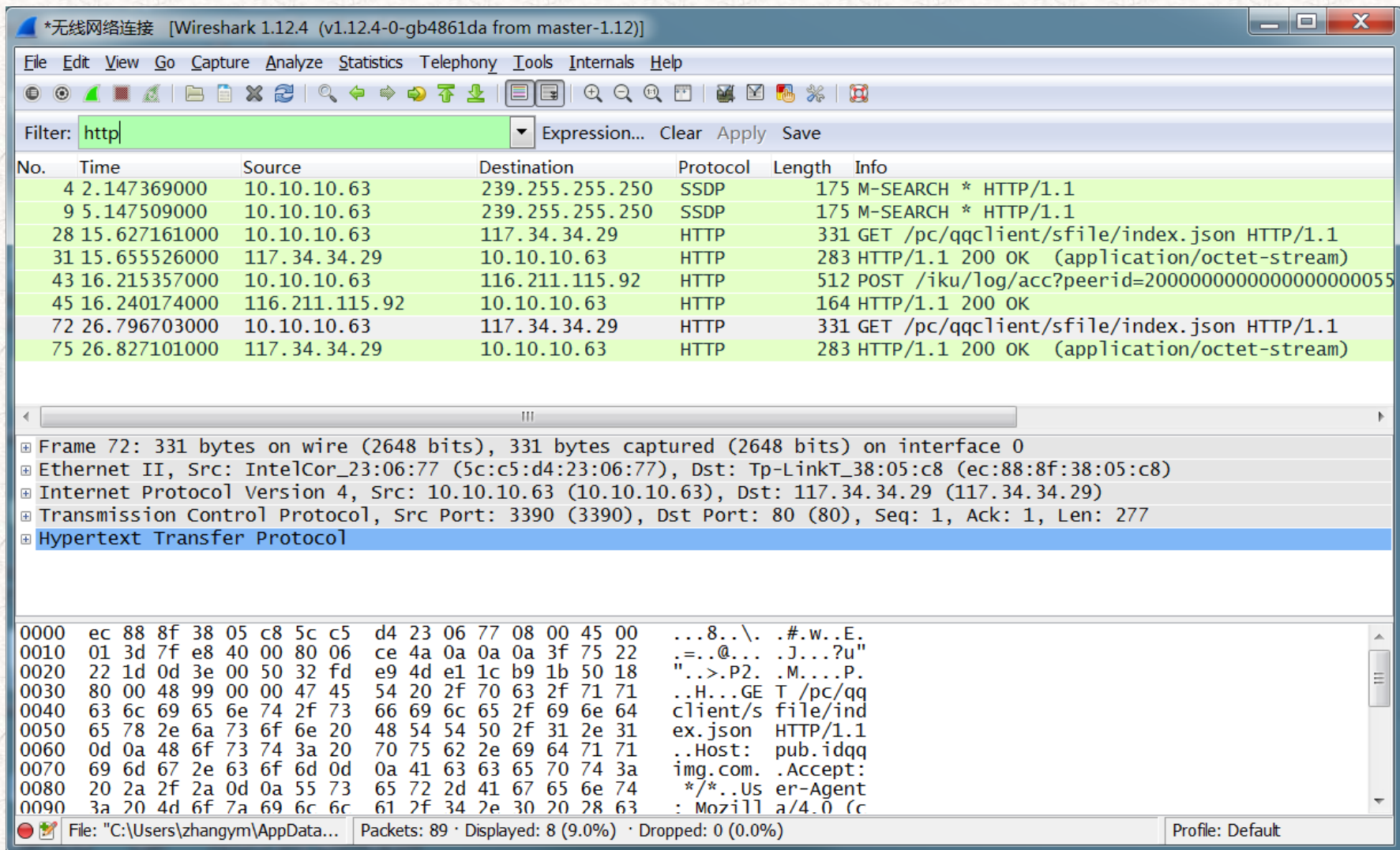
包的16进制代码区

包的ASCII代码区

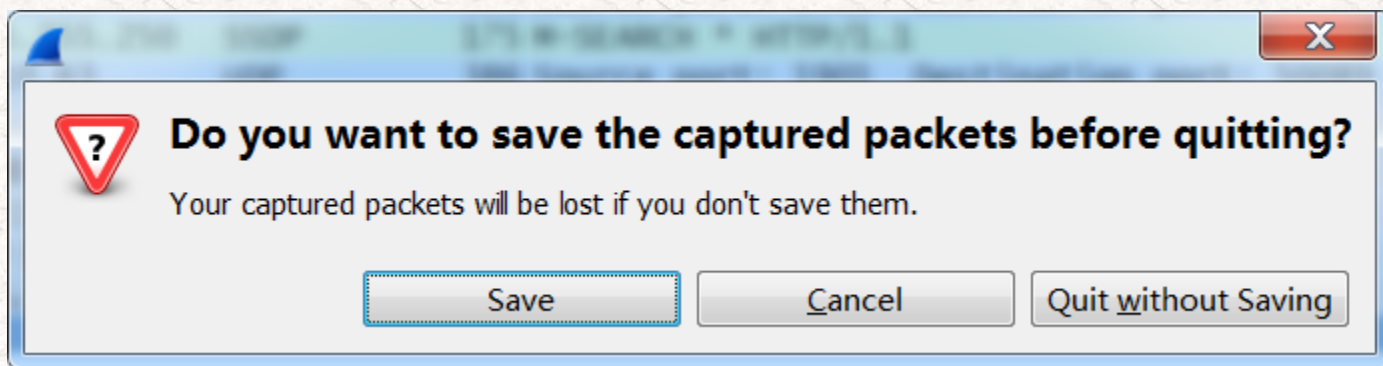
- 在界面上部的Filter框中可以设置显示过滤器条件:



- 如下图, 可以设置只显示http协议数据包:



- 在退出WireShark或进行下一次捕获操作时会提醒保存数据，可根据需要时行选择。如下图：



- 如果保存成文件以后就可以再次打开分析查看。

WireShark 常用菜单

- **1. MENUS**（菜单）
- **2. SHORTCUTS**（快捷按钮）
- **3. DISPLAY FILTER**（显示过滤器）
- **4. PACKET LIST PANE**（封包列表）
- **5. PACKET DETAILS PANE**（封包详细信息）
- **6. PACKET BYTES**（16进制数据）
- **7. Status Bar**（状态条）

1. MENUS（菜单）



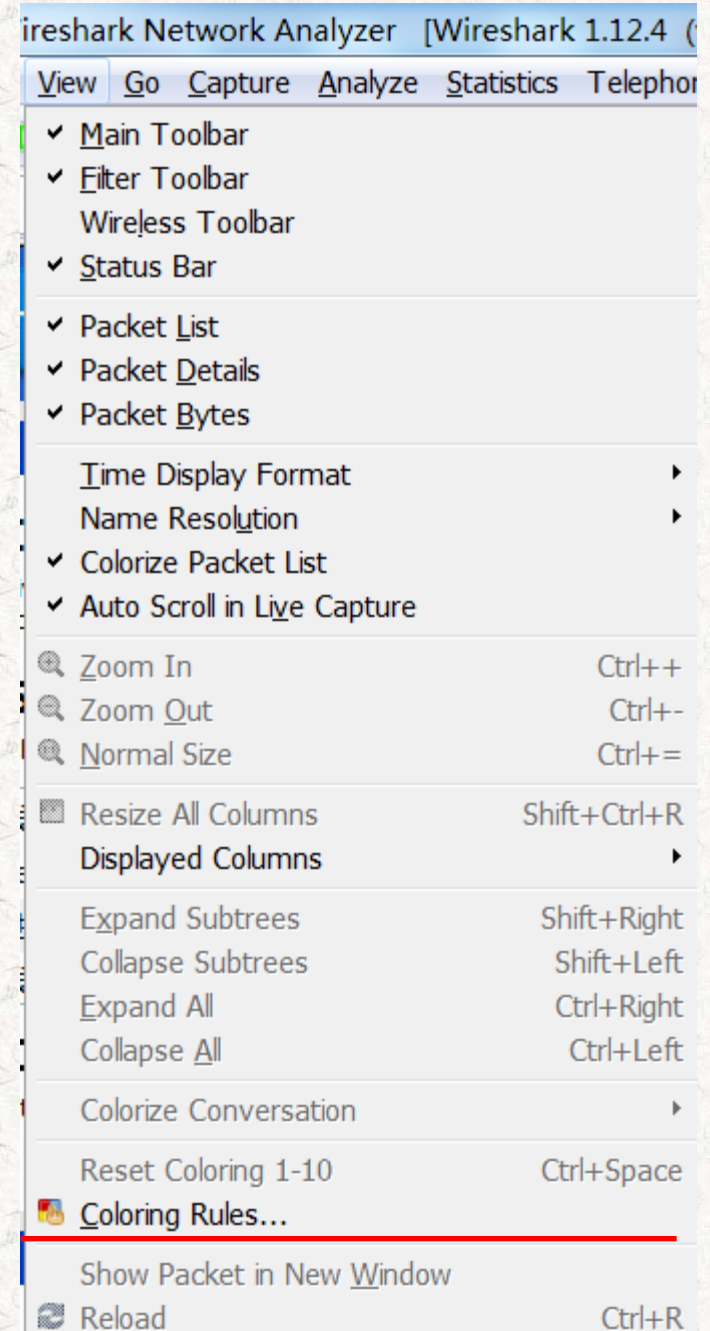
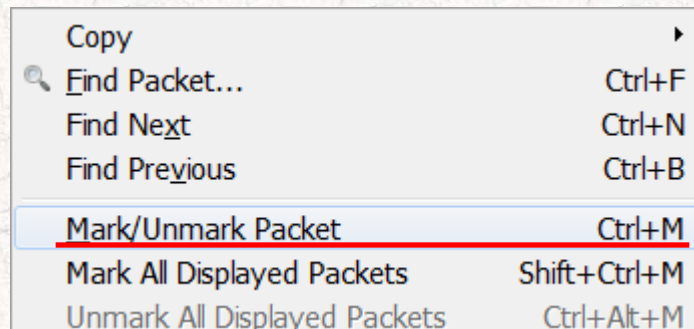
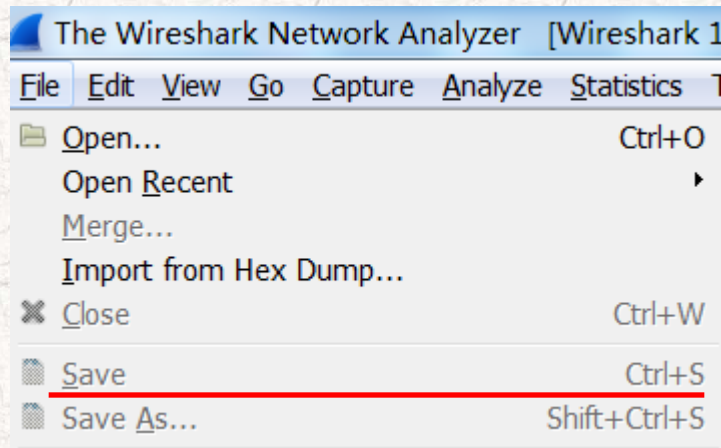
*无线网络连接 [Wireshark 1.12.4 (v1.12.4-0-gb4861da from master-1.12)]

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help

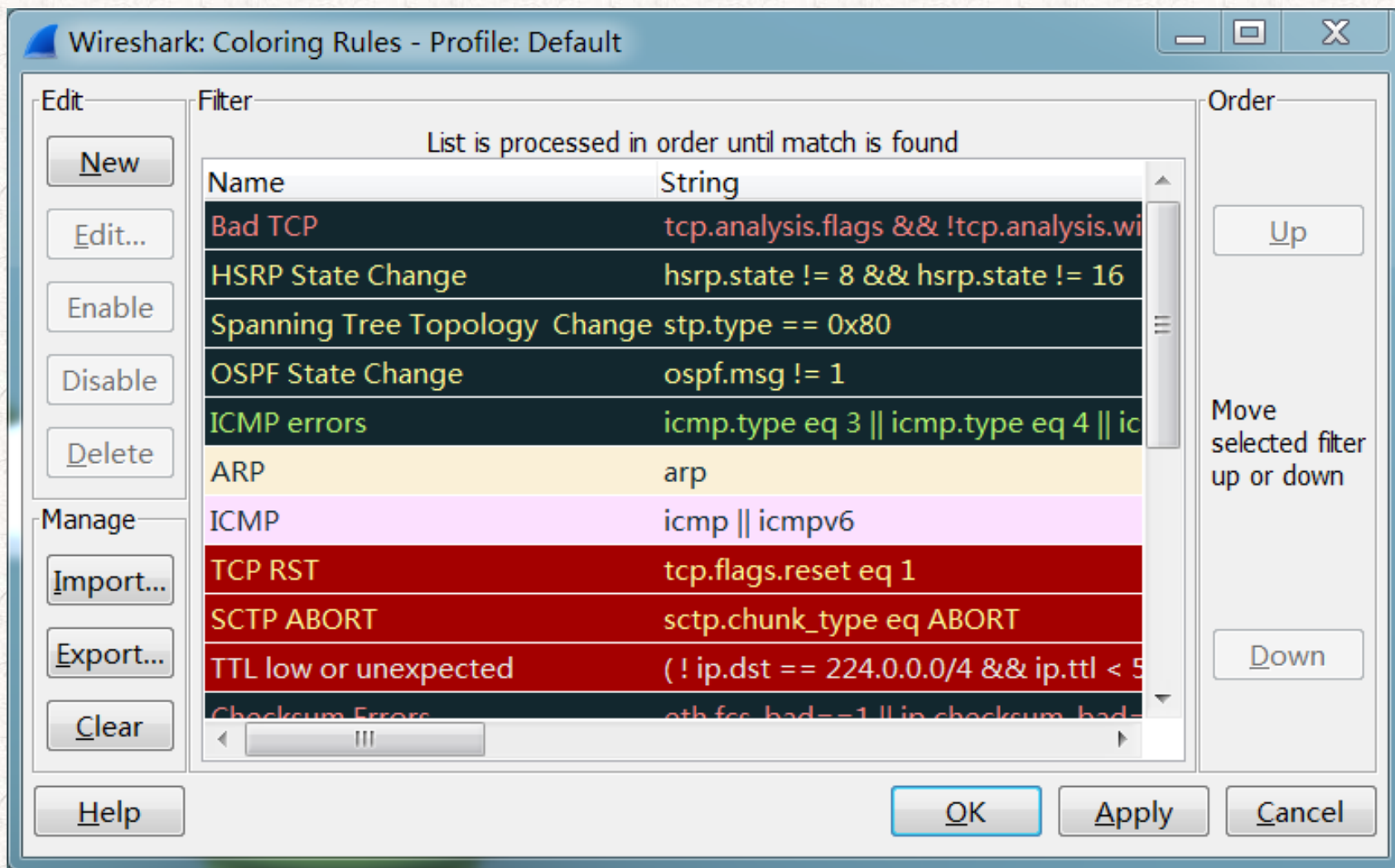
- “File”（文件） 打开或保存捕获的信息。
- “Edit”（编辑） 查找或标记封包。进行全局设置。
- “View”（查看） 设置Wireshark的视图。
- “Go”（转到） 跳转到捕获的数据。
- “Capture”（捕获） 设置捕捉过滤器并开始捕捉。
- “Analyze”（分析） 设置分析选项。
- “Statistics”（统计） 查看Wireshark的统计信息。
- “Help”（帮助） 查看本地或者在线支持。

常用的一些菜单选项:

- File菜单中的Save项
- Edit菜单中的Mark/Unmark Packet
- View菜单中Coloring Rules项

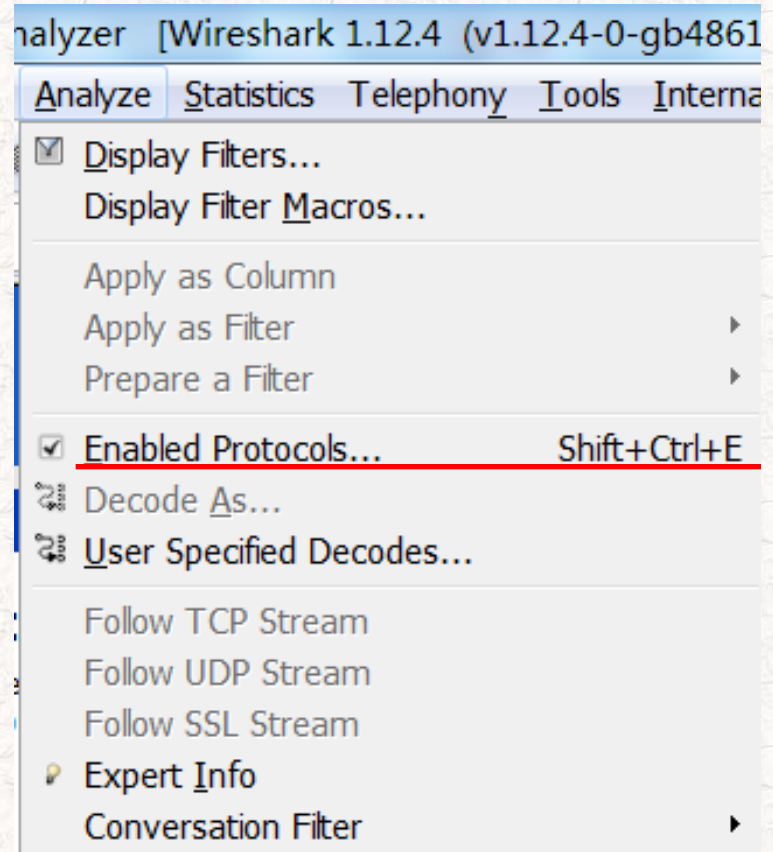
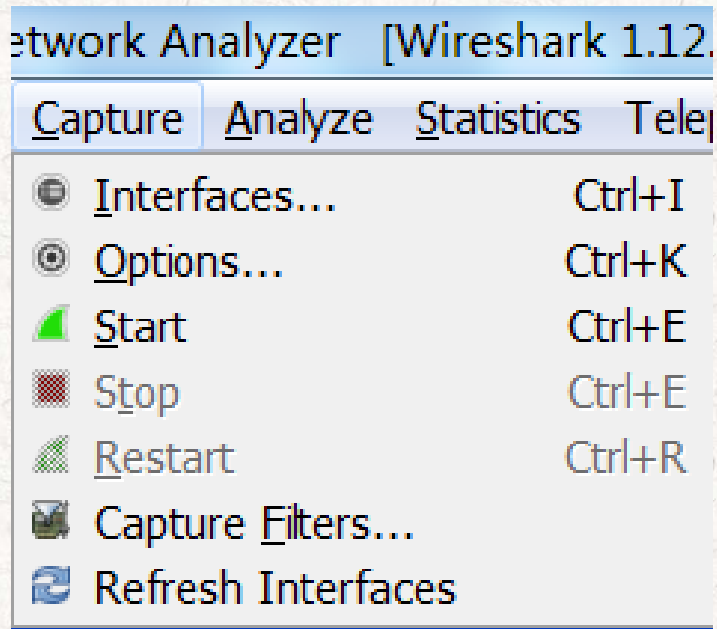


- 通过Coloring Rules可以配置各种协议数据包的显示颜色



常用的一些菜单选项:

- Capture菜单中的Start及Strop等
(快捷建是Ctrl+E)
- Analyze菜单中的Enabled Protocols



2. SHORTCUTS (快捷按钮)



- 在菜单下面，是一些常用的快捷按钮
 - 绿色的启动抓包按钮，一般都是点这个按钮开始抓包；
 - 红色的停止抓包按钮，当你抓完包之后，就是点这个停止了。

3. DISPLAY FILTER（显示过滤器）

Filter: ▼ Expression... Clear Apply Save

- 显示过滤器用于查找捕捉记录中的内容。请不要将捕捉过滤器和显示过滤器的概念相混淆（后页有专门的介绍）。

4. PACKET LIST PANE（封包列表）

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000000	123.151.13.34	10.10.10.63	OICQ	121	OICQ Protocol
2	1.486311000	10.10.10.63	10.10.10.255	NBNS	92	Name query NB {"ERROR"<00>
3	2.247460000	10.10.10.63	10.10.10.255	NBNS	92	Name query NB {"ERROR"<00>
4	3.011756000	10.10.10.63	10.10.10.255	NBNS	92	Name query NB {"ERROR"<00>
5	8.826230000	123.151.13.34	10.10.10.63	OICQ	121	OICQ Protocol
6	10.047967000	10.10.10.63	123.151.13.165	OICQ	81	OICQ Protocol
7	10.098148000	123.151.13.165	10.10.10.63	OICQ	89	OICQ Protocol
8	12.650075000	123.151.13.34	10.10.10.63	OICQ	121	OICQ Protocol
9	13.543574000	10.10.10.63	123.151.13.34	OICQ	81	OICQ Protocol
10	13.595668000	123.151.13.34	10.10.10.63	OICQ	89	OICQ Protocol

- 封包列表中显示所有已经捕获的封包。在这里可以看到发送或接收方的**MAC/IP地址**，**TCP/UDP**端口号，协议或者封包的内容。
- 如果捕获的是一个**OSI layer 2**的封包，在**Source**和**Destination**列中看到的**就是MAC地址**，此时**Port（端口）**列为空。
- 如果捕获的是一个**OSI layer 3**或者更高层的封包，在**Source**和**Destination**列中看到的是**IP地址**。
- 可以在这里添加/删除列，或者改变各列的颜色（**Edit**菜单中的**Preferences**项）

5. PACKET DETAILS PANE (封包详细信息)

Selected Packet	Time	Source	Destination	Port	Protocol	Info
	59.3	192.168.1.2	84.16.81.23	80	HTTP	GET /wireshark_use.php HTTP/1.1
	59.3	192.168.1.2	84.16.81.23	80	HTTP	GET /menu.js HTTP/1.1
	59.4	84.16.81.23	192.168.1.2	1600	HTTP	HTTP/1.1 304 Not Modified
	59.4	192.168.1.2	84.16.81.23	80	HTTP	GET /lookxml.css HTTP/1.1
	59.4	84.16.81.23	192.168.1.2	1600	HTTP	HTTP/1.1 304 Not Modified
Frame 152 (773 bytes on wire, 773 bytes captured)						
OSI Layer 2	Ethernet II, Src: 3com_9b:47:f7 (00:04:75:9b:47:f7), Dst: cisco-Li_2a:fb:9b (00:18:39:2a:fb:9b)					
OSI Layer 3	Internet Protocol, Src: 192.168.1.2 (192.168.1.2), Dst: 84.16.81.23 (84.16.81.23)					
OSI Layer 4	Transmission Control Protocol, Src Port: 1600 (1600), Dst Port: http (80), Seq: 1, Ack: 1, Len: 719					
OSI Layer 7	Hypertext Transfer Protocol					

- 这里显示的是在封包列表中被选中项目的详细信息。信息按照不同的OSI layer进行了分组，可以展开每个项目查看。下面截图中展开的是HTTP信息。

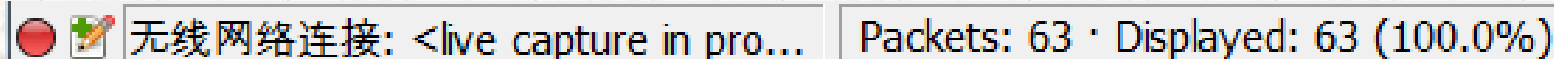
```
Hypertext Transfer Protocol
  POST /register?t=300 HTTP/1.1\r\n
    [Expert Info (Chat/Sequence): POST /register?t=300 HTTP/1.1\r\n]
      Request Method: POST
      Request URI: /register?t=300
      Request Version: HTTP/1.1
      Cache-Control: no-cache\r\n
      Connection: Keep-Alive\r\n
      Pragma: no-cache\r\n
      Content-Type: application/x-protobuf\r\n
      Accept: application/x-protobuf\r\n
      User-Agent: NIS/22.5.5.15/win\r\n
```


6. PACKET BYTES（16进制数据）

+ Frame 7: 122 bytes on wire (976 bits), 122 bytes captured (976 bits) on interface 0																							
+ Ethernet II, Src: IntelCor_23:06:77 (5c:c5:d4:23:06:77), Dst: Tp-LinkT_38:05:c8 (ec:88:8f:38:05:c8)																							
+ Internet Protocol Version 4, Src: 10.10.10.63 (10.10.10.63), Dst: 166.98.7.10 (166.98.7.10)																							
- Transmission Control Protocol, Src Port: 3569 (3569), Dst Port: 80 (80), Seq: 287, Ack: 1, Len: 68																							
Source Port: 3569 (3569)																							
Destination Port: 80 (80)																							
[Stream index: 0]																							
[TCP Segment Len: 68]																							
0000	ec	88	8f	38	05	c8	5c	c5	d4	23	06	77	08	00	45	00	...8...\ .#.w..E.						
0010	00	6c	13	27	40	00	80	06	25	b0	0a	0a	0a	3f	a6	62	.l.'@... %....?.b						
0020	07	0a	0d	f1	00	50	8a	84	cd	a5	bd	bf	f8	8f	50	18	..P..P.						
0030	01	02	74	bb	00	00	0a	0f	0a	09	36	33	30	39	34	30	..t.....630940						
0040	33	33	37	10	04	18	01	0a	0f	0a	09	36	33	30	39	34	337.....63094						
0050	30	33	33	37	10	05	18	01	0a	0f	0a	09	36	33	30	39	0337.....6309						
0060	34	30	33	33	37	10	03	18	01	0a	0f	0a	09	36	33	30	40337... ..630						
0070	39	34	30	33	33	37	10	02	18	01	940337... ..												

- **16进制数据查看面板。**这里显示的内容与“封包详细信息”中相同，只是改为左边以**16进制**的格式显示，右边以**ASCII值**对照显示。
在上图中，我们在“封包详细信息”中选择查看**TCP**的源端口（**3569**），其对应的**16进制**数据就突出显示在下面的面板中（**0d f1**）。

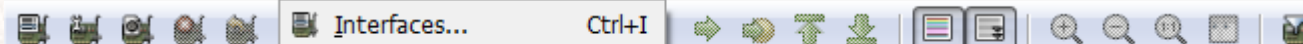
7. Status Bar (状态条)



- 在界面的最下端，可以获得如下信息：
 - 正在进行捕捉的网络设备。
 - 捕捉是否已经开始或已经停止。
 - 捕捉结果的保存位置。
 - 已捕捉的数据量。
 - 丢包信息等。
- 双击相应的项可以进一步查看相关信息。

非混杂模式及混杂模式下抓包

- 非混杂模式指：WireShark只抓取指定网卡上的发出与接收的数据包，与指定网卡无关的数据包将被忽略。
- 混杂模式指：WireShark能够抓取主机所在局域网内的全部网络包，即接收所有经过网卡的数据包，包括不是发给本机的包。



Filter:

WIRESHARK

Capture

Interface List

Live list of the capture interfaces (counts incoming packets)

Start

Choose one or more interfaces to capture from, then

- Intel(R) 82579LM Gigabit Network Connection
- Microsoft: \Device\NPF_{E1576CCD-EDC0-40...}

Capture Options

Start a capture with detailed options

Capture Help

How to Capture

Step by step to a successful capture setup

Network Media

Specific information for capturing on: Ethernet, WLAN, ...

Wireshark: Capture Options

Capture

Capture	Interface	Link-layer header	Prom. Mode	Snapplen [B]	Buffer [MB]
<input type="checkbox"/>	Intel(R) 82579LM Gigabit Net... fe80-b968:8a3a:d0d7:5055 0.0.0.0	Ethernet	enabled	default	1
<input checked="" type="checkbox"/>	Microsoft: \Device\NPF_{E157... fe80-d50e:b08a:26af:2537 192.168.1.101	Ethernet	enabled	default	1

☐ Capture on all interfaces

☒ Capture all in promiscuous mode

Manage Interfaces

Capture File(s)

File: Browse...

☐ Use multiple files ☒ Use pcap-ng format

☒ Next file every 1 megabyte(s)

☐ Next file every 1 minute(s)

☐ Ring buffer with 2 files

☐ Stop capture after 1 file(s)

Display Options

☒ Update list of packets in real time

☒ Automatic scrolling in live capture

☒ Hide capture info dialog

Name Resolution

☒ Enable MAC name resolution

☐ Enable network name resolution

☒ Enable transport name resolution

Stop Capture ...

☐ ... after 1 packet(s)

☐ ... after 1 megabyte(s)

☐ ... after 1 minute(s)

Help Start Close

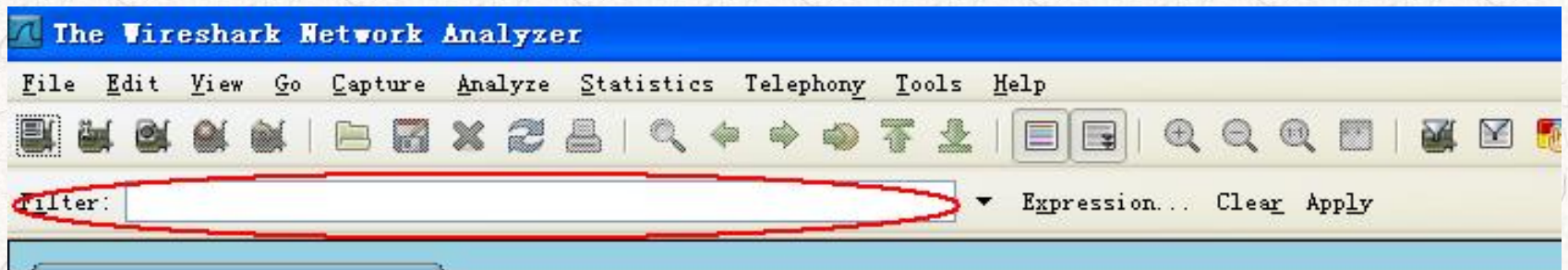
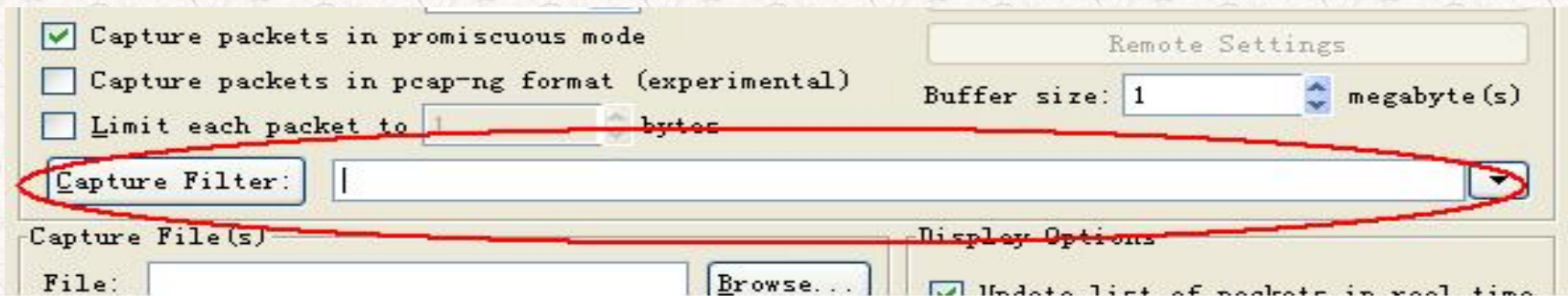
Wireshark Filters

过滤器

- **捕捉过滤器（Capture Filters）**：用于决定将什么样的信息记录在捕捉结果中。需要在开始捕捉前设置。
- **显示过滤器（Display Filters）**：在捕捉结果中进行详细查找。可以在得到捕捉结果后随意修改。
- 捕捉过滤器是数据经过的第一层过滤器，它用于控制捕捉数据的数量，以避免产生过大的日志文件。
- 显示过滤器是一种更为强大（复杂）的过滤器。它允许您在日志文件中迅速准确地找到所需要的记录。
- 两种过滤器使用的语法是完全不同的。

Wireshark Filter –

- > Capture Filter (捕捉过滤器)
- > Filter (显示过滤器)



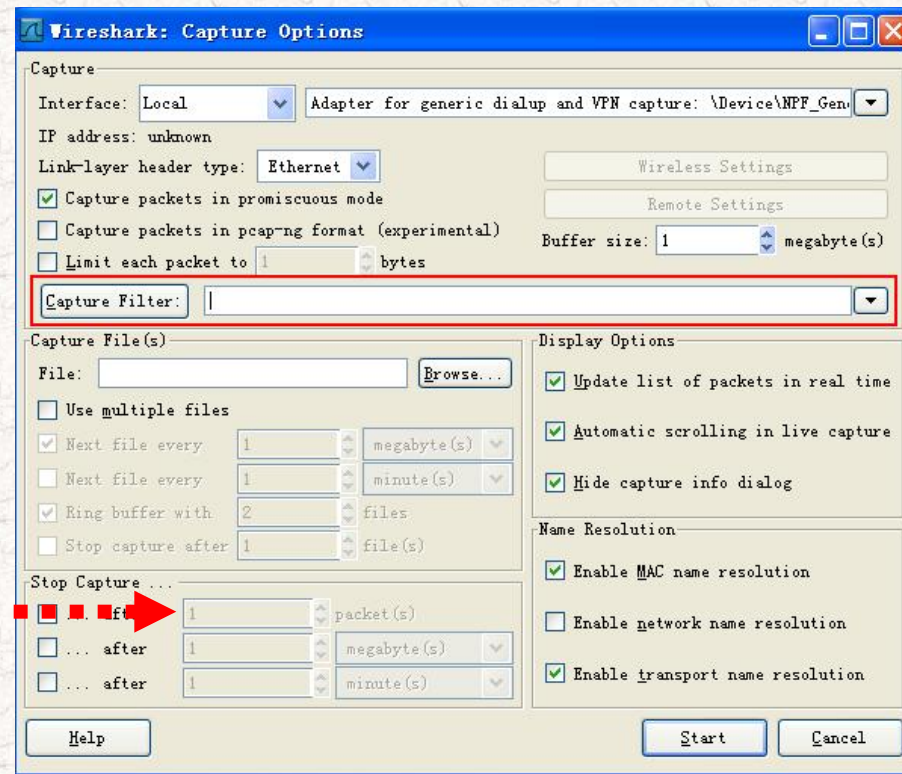
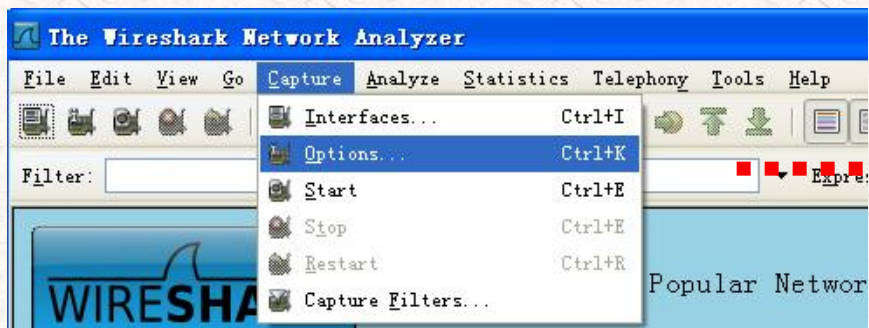
Capture Filter

- 设置Capture filter 步骤:

- 选择 **Capture -> Options...** 或者使用快捷键Ctrl+K

- 填写“Capture Filter”栏或者点击“Capture Filter”按钮为您的过滤器起一个名字并保存，以便在今后的捕捉中继续使用这个过滤器

- 点击开始（Start）进行捕捉。



- **Capture Filter 语法:**

语法:

Protocol	Direction	Host(s)	Value	Logical Operations	Other expression
----------	-----------	---------	-------	--------------------	------------------

例子: **iax2** **dst** **61.154.152.13** **4569** **and** **src 192.168.2.111**

Protocol (协议):

可能的值: ether, fddi, ip, arp, rarp, decnet, lat, sca, moprc, mopdl, tcp and udp.

如果没有特别指明是什么协议, 则默认使用所有支持的协议

Direction (方向):

可能的值: src, dst, src and dst, src or dst

如果没有特别指明来源或目的地, 则默认使用 “src or dst” 作为关键字。

例如, “host 192.168.2.111” 与 “src or dst host 192.168.2.111” 是一样的。

Host(s):

可能的值: net, port, host, portrange.

如果没有指定此值, 则默认使用 “host” 关键字。

例如, “src 192.168.2.111” 与 “src host 192.168.2.111” 相同。

Logical Operations (逻辑运算):

可能的值: not, and, or.

否 (“not”) 具有最高优先级。或 (“or”) 和与 (“and”) 具有相同的优先级, 运算时从左至右进行。

例, “not tcp port 3128 and tcp port 23” 与 “(not tcp port 3128) and tcp port 23” 相同。

“not tcp port 3128 and tcp port 23” 与 “not (tcp port 3128 and tcp port 23)” 不同。

实例：

`udp dst port 4569`

显示目的UDP端口为4569的封包。

`ip src host 192.168.4.7`

显示来源IP地址为192.168.4.7的封包。

`host 192.168.4.7`

显示目的或来源IP地址为192.168.4.7的封包。

`src portrange 2000-5000`

显示来源为TCP或UDP，并且端口在2000~5000范围内的封包。

`not icmp`

显示除icmp以外的封包。

`src host 172.17.12.1 and not dst net 192.168.2.0/24`

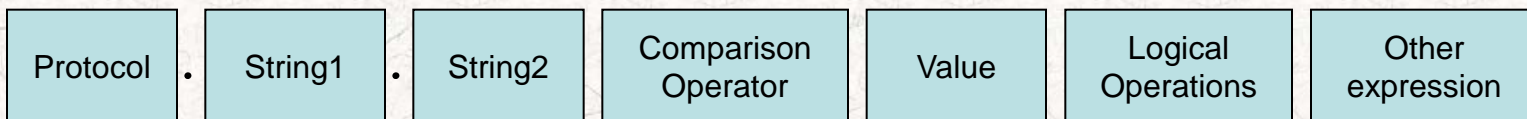
显示来源IP地址为172.17.12.1，但目的地址不是192.168.2.0/24的封包。

`(src host 10.4.1.12 or src net 10.6.0.0/16) and tcp dst portrange 200-10000 and dst net 10.0.0.0/8`

显示来源IP为10.4.1.12或者来源网络为10.6.0.0/16，目的地TCP端口号在200至10000之间，并且目的位于网络10.0.0.0/8内的所有封包。

Filter

语法:

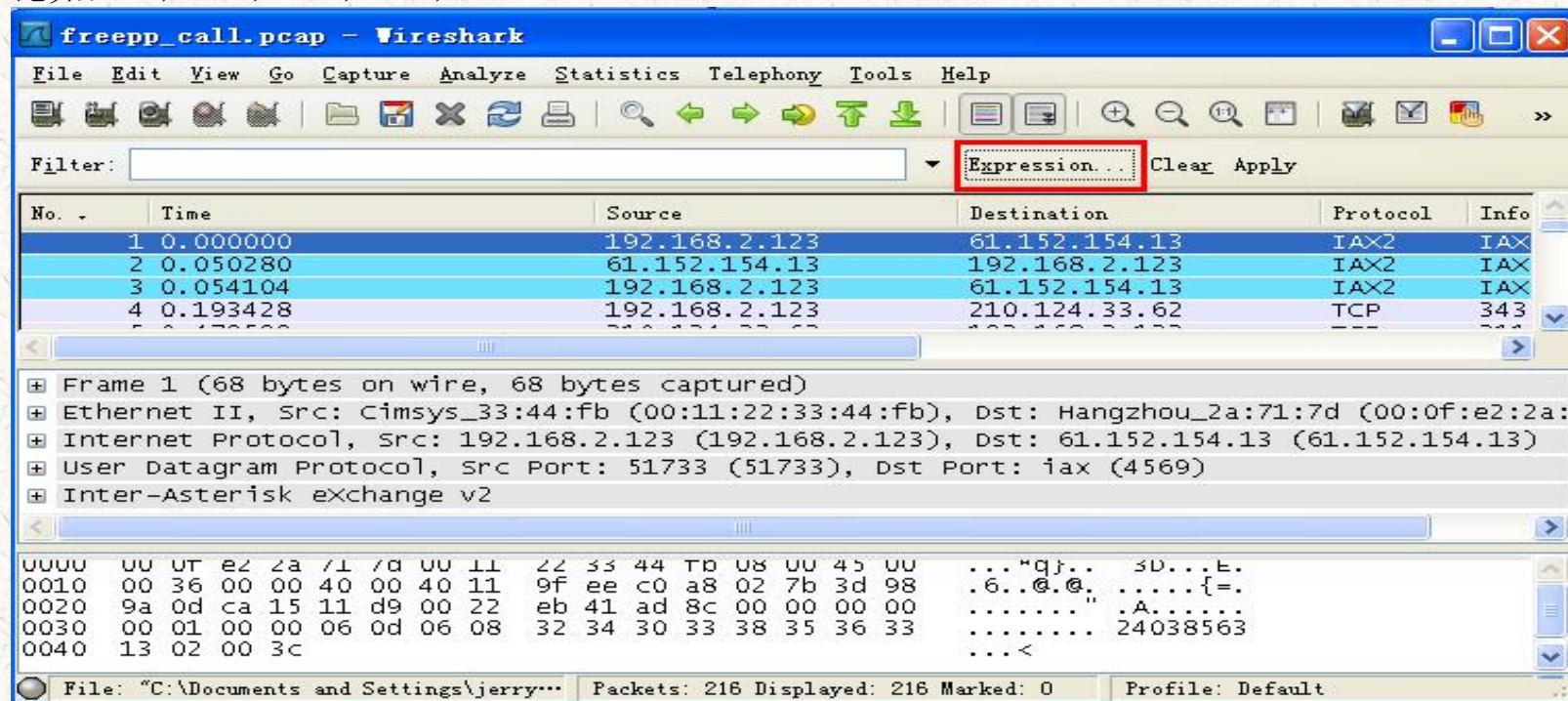


例子: ftp passive ip == 192.168.1.1 or icmp.type

Protocol (协议):

可以使用大量位于OSI模型第2至7层的协议。点击“Expression...”按钮后，您可以看到它们。

比如: IP, TCP, UDP, DNS, SSH



Wireshark: Filter Expression - Profile: Default



Field name

+ MIBS - MIBs

+ Expert - Expert Info

+ 104apci - IEC 60870-5-104-Apci

+ 104asdu - IEC 60870-5-104-Asdu

+ 2dparityfec - Pro-MPEG Code of Practice #3

+ 3COMXNS - 3Com XNS Encapsulation

+ 3GPP2 A11 - 3GPP2 A11

+ 802.11 MGT - IEEE 802.11 wireless LAN manag

+ 802.11 Radiotap - IEEE 802.11 Radiotap Capt

+ 802.3 Slow protocols - Slow Protocols

+ 9P - Plan 9 9P

 AAL1 - ATM AAL1

 AAL3/4 - ATM AAL3/4

+ AARP - Appletalk Address Resolution Protoco

104apci - IEC 60870-5-104-Apci

Relation

is present

==

!=

>

<

>=

<=

contains

matches

Value (protocol)

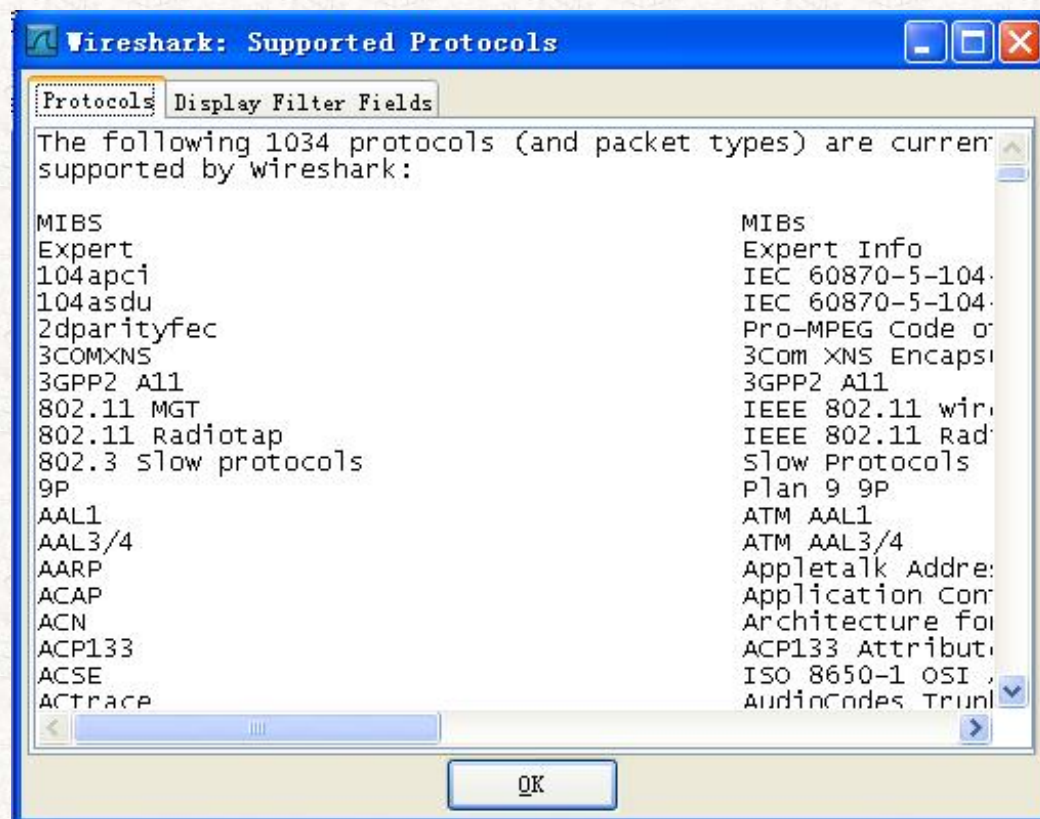
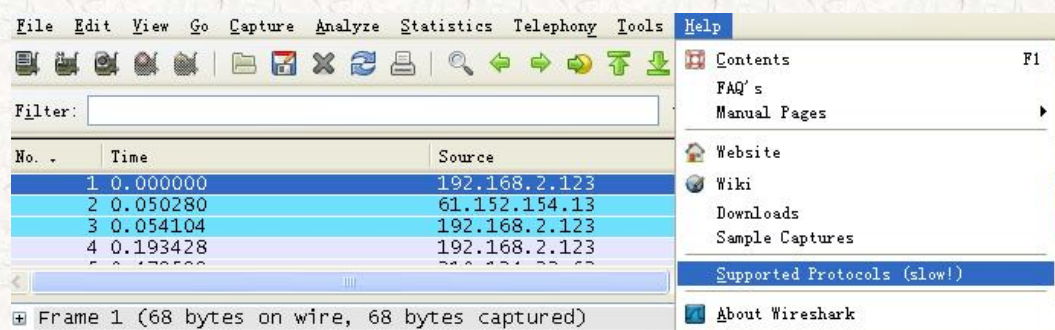
Predefined values:

Range (offset:length)

OK

Cancel

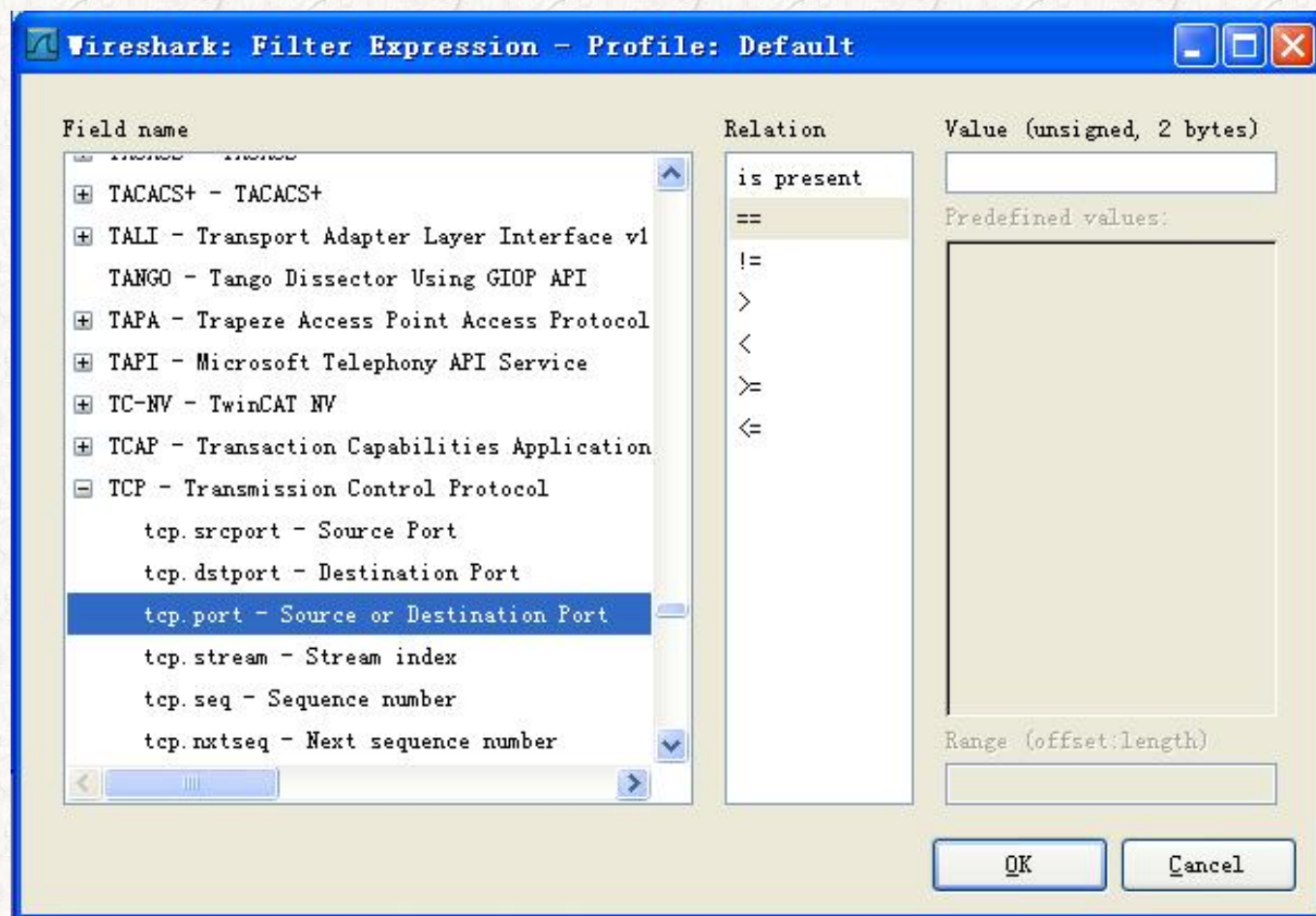
同样可以在以下位置找到所支持的协议



String1, String2（可选项）：

以协议的子类。

点击相关父类旁的“+”号，然后选择其子类。



Comparison Operators（比较运算符）：

可以使用以下几种运算符：

英文写法	C语言写法	含义
eq	==	Equal
ne	!=	Not Equal
gt	>	Greater Than
lt	<	Less Than
ge	>=	Greater than or Equal to
le	<=	Less than or Equal to

Logical Expressions（逻辑运算符）：

英文写法	C语言写法	含义
and	&&	逻辑与
or		逻辑或

注意：

- 1、进行比较运算时，“等于”要使用“**=**”或“eq”，不能使用“=”；
- 2、逻辑运算时，“逻辑与”要使用“**&&**”或“and”，而不能使用“&”；
- 3、逻辑运算时，“逻辑或”要用“**||**”或“or”；
- 4、表达式**区分大小写**，如“udp”不能写成“Udp”或“UDP”，“eq”不能写成“Eq”，等。

语法实例：

1、只显示目的UDP端口为4569或者来源TCP端口为80的封包：

Filter: `udp.dstport eq 4569 or tcp.srcport==80`

✓ 表达式正确

Filter: `udp.dstport eq 4569 || tcp.srcport == 80`

✓ 表达式正确

Filter: `Udp.dstport eq 4569 || tcp.srcport == 80`

✗ 表达式错误

Filter: `UDP.dstport eq 4569 || tcp.srcport == 80`

✗ 表达式错误

Filter: `udp.dstport Eq 4569 || tcp.srcport == 80`

✗ 表达式错误

Filter: `udp.dstport eq 4569 Or tcp.srcport == 80`

✗ 表达式错误

✘ 表达式语法正确，Filter栏背景色为绿色显示，Enter（回车）后，过滤显示符合对应条件的封包，语法错误背景色为红色，Enter（回车）后，会弹出错误提示信息，不会显示出对应条件的封包。

2、一个整数可以用十进制表达，也可以用八进制、十六进制表达，下面的语法意义是相同的：

frame.cap_len >= 10

→ 十进制表达

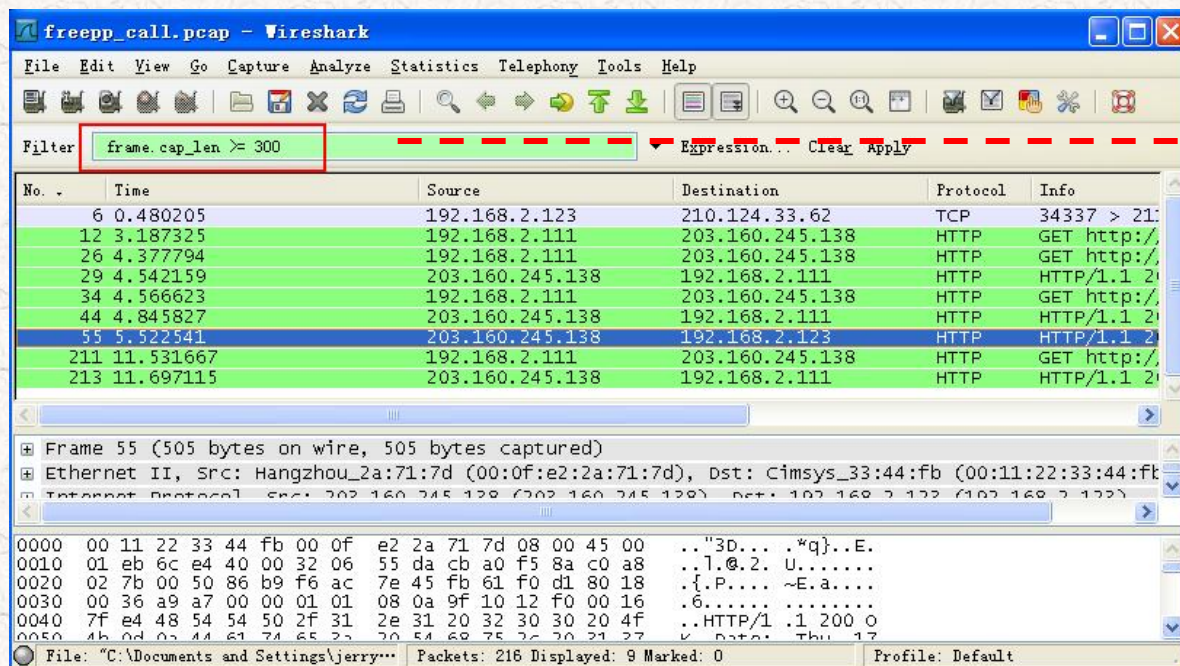
frame.cap_len >= 012

→ 八进制表达

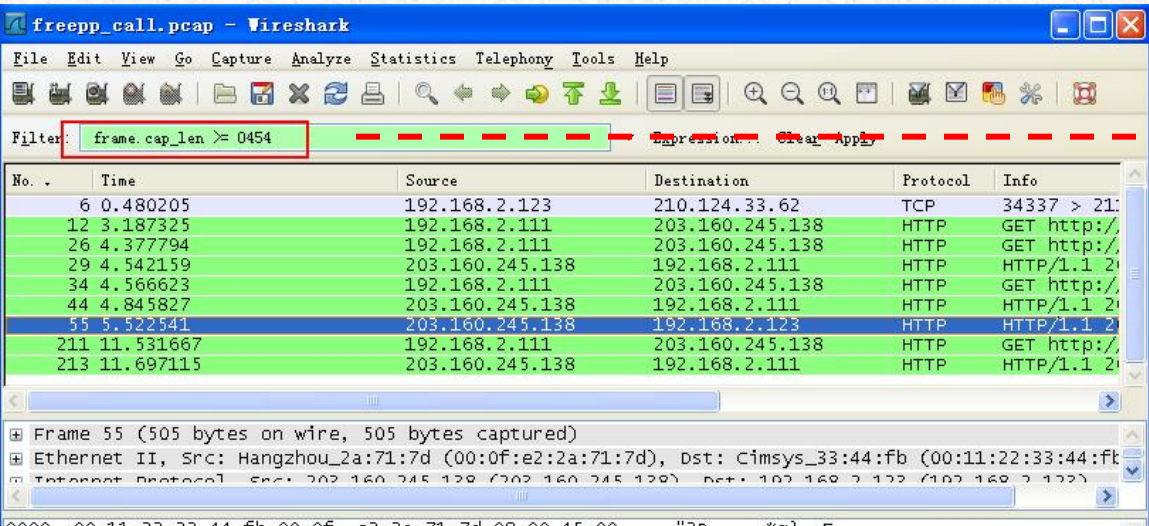
frame.cap_len >= 0xa

→ 十六进制表达

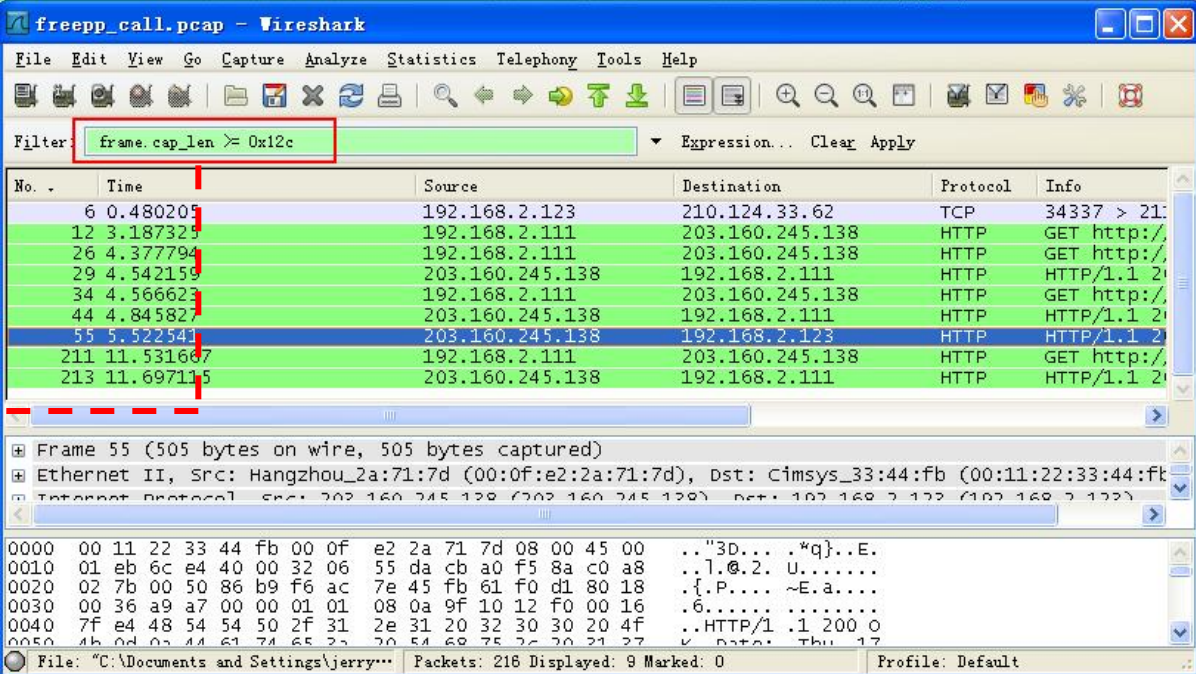
例如：



→ 十进制表达



八进制表达



十六进制表达

3、以太网地址和字节数组使用十六进制表示，十六进制的数字可以被“:”“.”“-”分隔。

例如:

```
eth.dst eq ff:ff:ff:ff:ff:ff
```

```
aim.data == 0.1.0.d
```

```
fddi.src == aa-aa-aa-aa-aa-aa
```

```
echo.data eq 7a
```

4、IPv4地址可以被表示成点分十进制或者使用主机名表示。

例如:

```
ip.dst eq www.freepp.com
```

```
ip.src == 192.168.1.1
```

5、当使用IPv4子网划分的时候，CIDR（Classless InterDomain Routing）表示法也可以使用。

例如：以下的过滤器可以找到所有129.111的数据包：

```
ip.addr == 129.111.0.0/16
```

斜线后面的数字用于表示子网占用的比特数。CIDR表示法也用于查找主机名，例如C类网络中主机“sneezy”的IP地址。

```
ip.addr eq sneezy/24
```

6、双引号封装字符串。

例如:

```
http.request.method == "GET"
```

Filter: http.request.method == "GET" Express				
	Source	Destination	Protocol	Id
	192.168.2.111	202.108.6.243	HTTP	G
	192.168.2.111	202.108.6.243	HTTP	G
	192.168.2.111	202.108.22.04	HTTP	G

常用表达举例：

```
snmp || dns || icmp
```

显示SNMP或DNS或ICMP封包。

```
ip.addr == 192.168.2.1
```

显示来源或目的IP地址为10.1.1.1的封包。

```
ip.src != 10.1.2.3 or ip.dst != 10.4.5.6
```

显示来源不为10.1.2.3或者目的不为10.4.5.6的封包。

```
ip.src != 10.1.2.3 and ip.dst != 10.4.5.6
```

显示来源不为10.1.2.3并且目的不为10.4.5.6的封包。

```
udp.port eq 4569
```

显示来源或目的UDP端口号为4569的封包。

```
tcp.dstport == 25
```

显示目的TCP端口号为25的封包。

```
tcp.flags
```

显示带有TCP标志的封包。

```
tcp.flags.syn eq 0x02
```

显示带有TCP SYN标志的封包。

THE END