# 1.Omnipeek使用总结

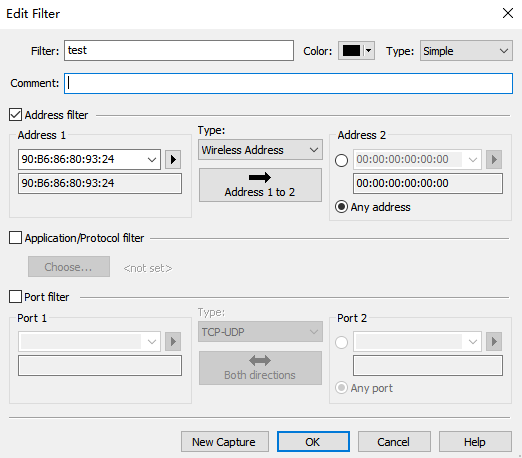
## 过滤器

### 新建过滤器

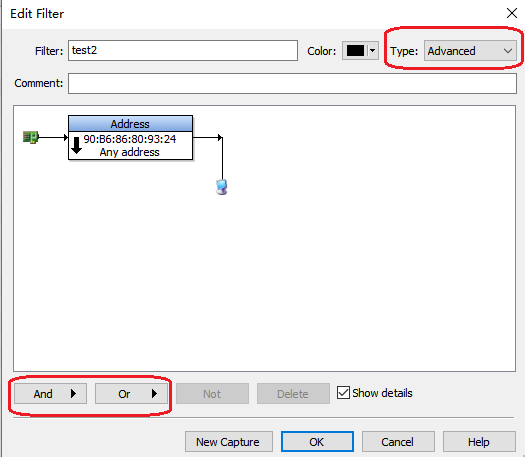
Omnipeek自带过滤器，如Beacon/Control/DATA/DHCP/ARP等。

也可以建立自定义的过滤器，有simple/ advanced两种方式。

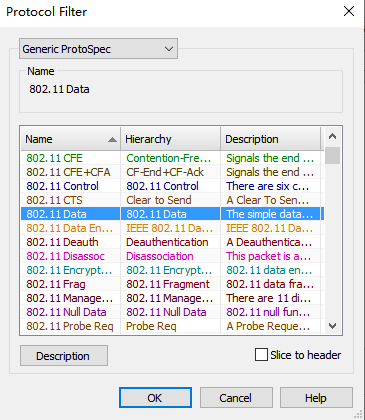
1.新建一个simple的filter，名字为test。Type默认为Simple，



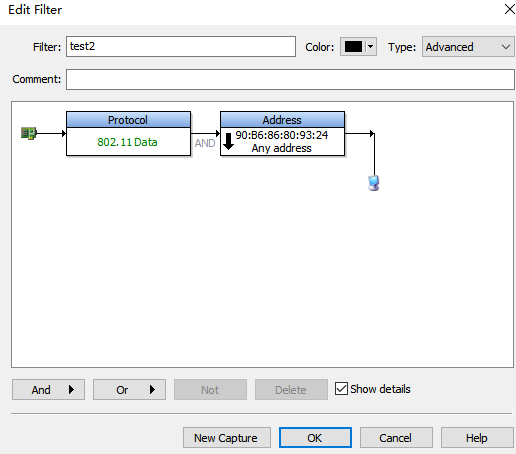
2.新建一个advanced的filter，名字为test2。Type选择Advanced，



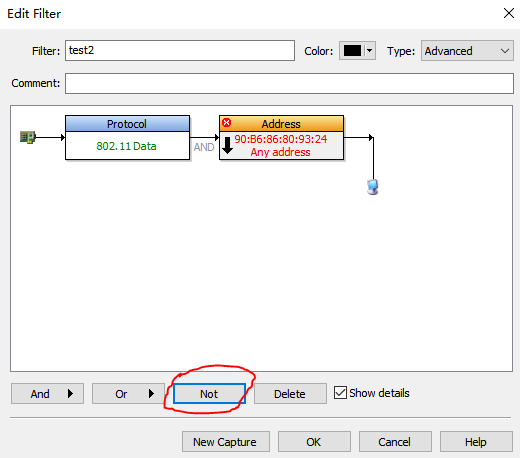
点击And或Or，创建第二个过滤条件，弹出对话框如下：



选择Application/Protocol filter，选择Generic ProtoSpec，选择802.11 Data类型。得到如下过滤器，



还可以设置否定条件，点击第二个条件，再点击Not按钮，得到第二个条件的否定条件，



### 过滤器使用

1.抓包的时候过滤

抓包的时候选择某个过滤器，则只抓取符合过滤器条件的包；也可以不选择任何过滤器，则抓取空中的所有包。

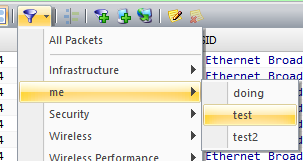
抓包的时候可以选择多个过滤器吗？待验证

2.显示的时候过滤

显示的时候选择某一个过滤器，则只显示符合过滤器条件的包。



选择一个自己建立的过滤器，例如之前建立的名字为test的过滤器。也可以选择Omnipeek自带的过滤器。



注意，如果想切换过滤器，比如当前显示的是management类型包，现在想显示control类型包，则必须先在过滤器上选择All Packets，之后再选择control过滤器。如果没有先在过滤器上选择All Packets这个步骤，则过滤出来的包为空，也就是说之后选择的过滤器是应用在先前已选过滤之上，是二次过滤。

### 过滤器条使用

过滤条，其实就是将之前使用**鼠标点击**使用过滤器的操作，转化为**纯文本**建立和应用过滤条件的操作了。

**过滤条功能更强大：**

1.过滤条可以使用& (And), | (Or), ! (Not), () (Group)等运算符，可以实现比鼠标建立过滤器更复杂的过滤条件。

2.过滤条可以选择显示符合条件的包、或者隐藏符合条件的包，鼠标点击使用过滤器只能显示符合条件的包。

过滤条语法，参考帮助文档：Creating filters with the filter bar。

#### 1.过滤条的使用

过滤条输入框输入filter('test')，回车，选择显示或者隐藏符合过滤条件的帧。

**过滤条件：filter('test')是一个过滤表达式，filter()是过滤函数，test已有的过滤器的名字，过滤器名字加单引号**。过滤器可以是自建的，也可以说omnipeek软件自带的。

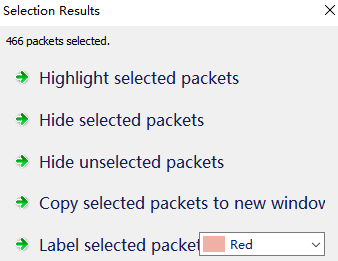
过滤器名字**大小写不敏感**，filter('802.11 control')、filter('802.11 CONTROL')都是正确的。

使用自建的过滤器，如filter('test')；

也可以使用Omnipeek软件自带的过滤器，如filter('802.11 control')、filter('802.11 beacons)、filter('dhcp')。

过滤条输入过滤条件，按回车键，或者鼠标点击右边三角形按钮，应用过滤条件。效果和之前鼠标选择过滤条件是一样的。

按钮，应用过滤条件，可以有多种选择，如下：



比鼠标选择过滤条件功能更强大。

#### 1.过滤条件的编写

**过滤条件：filter('test')是一个过滤表达式，filter()是过滤函数，test已有的过滤器的名字**。

**常用过滤器函数，filter('test')、****pspec('802.11 data')**。

filter('802.11 control') 和pspec('802.11 control')的结果是一样的，但是pspec('802.11 rts')是合法的、而filter('802.11 rts')是非法的。不知道有什么规则？

结论：

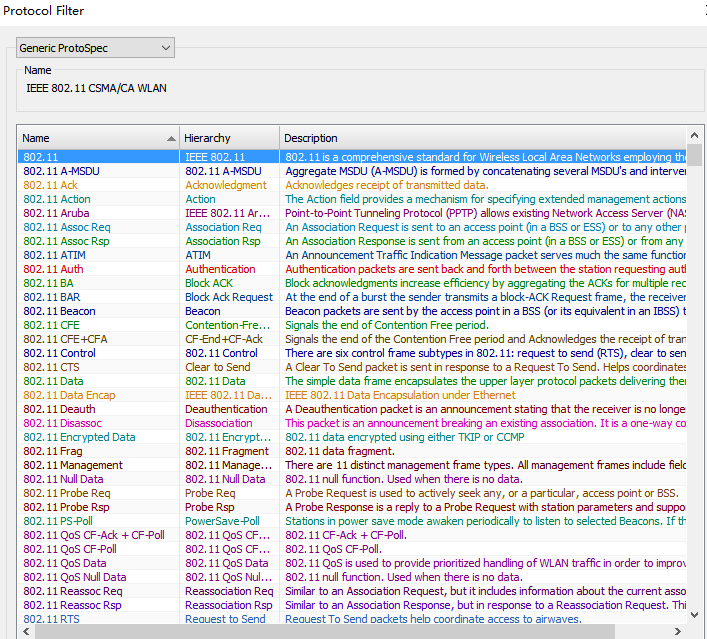
**①.自定义过滤器的过滤条件使用filter('test')，**

**②.802.11协议帧的过滤器条件使用pspec('802.11 data')**。

#### 1.过滤器的名字

**pspec('过滤器名字')，这里的过滤器名怎么写？**

在新建过滤器的时候，选择Application/Protocol filter，选择Generic ProtoSpec，看到如下过滤器列表，**第一栏Name就是过滤器的名字**。备注：这些是omnipeek软件自带过滤器。



##### 1.常用的过滤器名字

802.11 control、802.11 data、802.11 management

802.11 rts、802.11 cts、802.11 ack、802.11 ba、802.11 bar、……

802.11 null data、802.11 qos data、802.11 a-msdu、802.1x、arp、arp request、arp response、……

802.11 beacon、802.11 probe req、802.11 probe rsp、802.11 auth、802.11 deauth、802.11 assoc req、802.11 assoc rsp、802.11 disassoc、802.11 reassoc req、802.11 reassoc rsp、802.11 addba req、802.11 addba rsp、……

备注：

802.1x -- eapol四次握手帧

#### 1.过滤条综合使用

①.sta建立连接的过程：

pspec('802.11 auth') | pspec('802.11 assoc req') | pspec('802.11 assoc rsp') | pspec('802.1x')

2.dhcp、arp过程

pspec('dhcp') | pspec('arp')

pspec('802.11 auth') | pspec('802.11 assoc req') | pspec('802.11 assoc rsp') | pspec('802.1x') | pspec('dhcp') | pspec('arp')

#### 1.过滤条语法

**1.简单过滤条件**

如：filter('test') 、filter('802.11 control')。

**2.组合过滤条件**

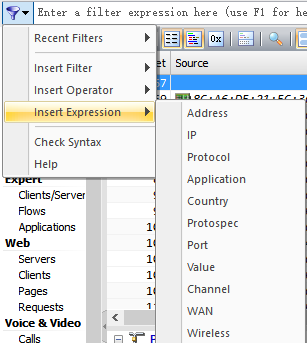
如：取反，在过滤条件的前面加!符号，

!filter('802.11 control')

!filter('802.11 control') & filter('test')

可以使用& (And), | (Or), ! (Not), () (Group)等运算符。

如果对过滤条语法不熟悉，可以使用过滤条下拉列表选择需要的过滤条件，如下：



**3.通配符：**

addr(type:wireless,addr1:'90.B6.86.80.93.\*',addr2:40.45.DA.AA.99.FE) //有通配符\*时需要加单引号，没有通配符时单引号可加可不加。

addr(type:wireless,addr1:'90.B6.86.\*.\*.\*',addr2:40.45.DA.AA.99.FE)

addr(type:wireless,addr1:'90.B6.\*.\*.\*.\*',addr2:40.45.DA.AA.99.FE)

4.或者，直接编写过滤条件，如下：

mac地址过滤，

addr(type:wireless,addr1:90.B6.86.80.93.24,addr2:40.45.DA.AA.99.FE) //注意mac地址中间用点号,不能用冒号(语法冲突)

addr(type:wireless,addr1:90B686809324,addr2:4045DAAA99FE) //或者，直接写十六进制mac地址，中间不加任何符号

addr(type:wireless,addr1:90.B6.86.80.93.24,addr2:40.45.DA.AA.99.FE,dir:1to2) //还可以设定addr1-addr2之间的方向，1to2、2to1、both(默认)，默认both。

备注:2to1不能用，软件bug，只能将addr2、addr1位置对调，仍然使用1to2，如下：

addr(type:wireless,addr2:90.B6.86.80.93.24,addr1:40.45.DA.AA.99.FE,dir:1to2)

5.更多过滤条件

bssid过滤：

wireless(bssid:90.B6.86.80.93.24)

bssid过滤 & mac地址过滤：

addr(type:wireless,addr1:90.B6.86.80.93.24,addr2:40.45.DA.AA.99.FE) & wireless(bssid:90.B6.86.80.93.24)

广播mac地址：

addr(type:wireless,addr1:90.B6.86.80.93.24,addr2:ff.ff.ff.ff.ff.ff)

addr(type:wireless,addr1:90B686809324,addr2:ffffffffffff)

ip地址过滤：

addr(ip:255.255.255.255)或者addr(ip:'255.255.255.\*') //有通配符\*时需要加单引号，没有通配符时单引号可加可不加。

ip(255.255.255.255)或者ip('255.255.255.\*') //直接使用ip函数就可以

6.其他：

addr(ethernet:'3com:\*.\*.\*')、addr(type: ip, addr1: 10.4.3.1, addr2: 10.5.1.1, dir: 1to2)

**filter bar help说明：**

Filter button: Click to display Filter Bar menu options.

• Recent Filters: Select a recently defined filter from this list.

• Insert Filter: Select a filter from this list.

• Insert Operator: Select an operator from this list: **& (And), | (Or), ! (Not), () (Group)**

• Insert Expression: Select a filter type expression from this list.

• Check Syntax: Select this option for a tooltip describing the syntax of your filter. For example, a correctly defined filter will display Filter OK in the tooltip.

• Help: Select this option to display information about how to use the filter bar.

Filter Bar text box: The filters, operators, and expressions chosen from the Filter button menu appear in this text box as you select them. The filter bar text box background changes color as you type into it to indicate whether a valid filter is entered:

• Valid filter=gree

• Invalid filter=red

• Empty=white

Apply Filter button: Click to apply your filter to the packets in the capture buffer of this capture window.

# Omnipeek（wildpackets）抓包：过滤器设置和数据包分析

<http://blog.csdn.net/ceo158/article/details/11133551>

在C/S或B/S[**架构**](http://lib.csdn.net/base/architecture)的系统级[**测试**](http://lib.csdn.net/base/softwaretest)中，有时需要验证客户端是否发送了某种特定的数据包。

可以通过Omnipeek抓包工具分析验证，过滤出符合要求的数据包。关键是设置过滤条件，使Omnepeek能够保存符合条件的数据包。设置过滤条件的过程是这样的：

## 1、简单过滤器

点击菜单栏view\filter打开过滤器视图，双击需要编辑的过滤打开Edit Filter对话框，如下：

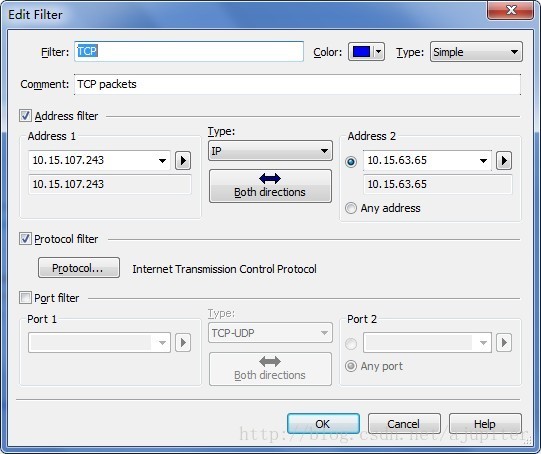


图1  编辑过滤器-简单设置

图1是一个简单的设置页面，在这里可以设置需要过滤的IP地址（Address filter），type，数据通信方向（可选单向，双向），通信端口。

## 2、进阶设置

但是图1所示简单的设置不能满足比如过滤数据包长度，数据包内容等条件。点击图1右上角Type，选择advanced，打开如图2所示设置页面：

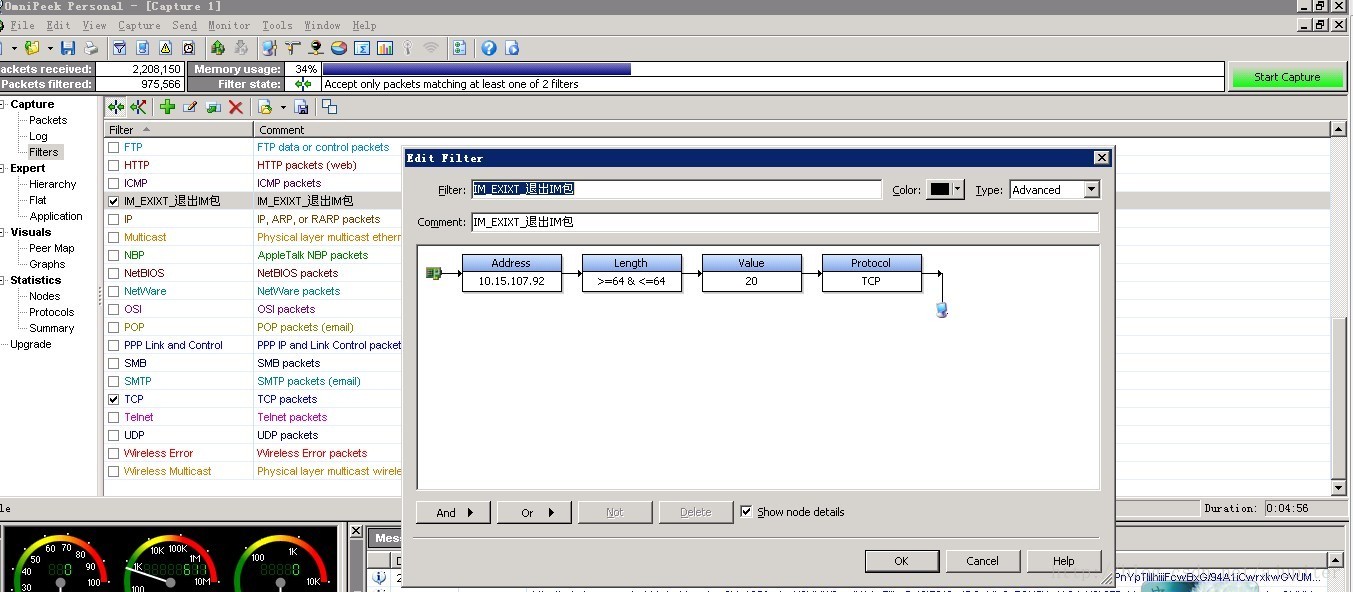


图2  过滤器设置-进阶设置

在上述图2中，可以看到Edit Filter中有四个条件串在一起（and）。双击每一个条件，可详细进行设置。

其中，Address过滤器可设置需过滤的源地址、目的地址、端口等，这个在图1中同样可以设置。

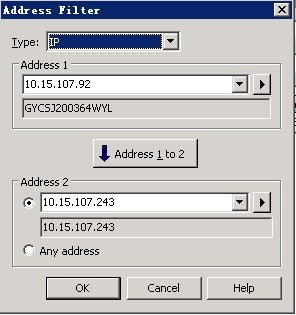


图3  Address过滤器

Protocol过滤器可设置需需要过滤的协议。这个过滤同样在图1中可以设置。

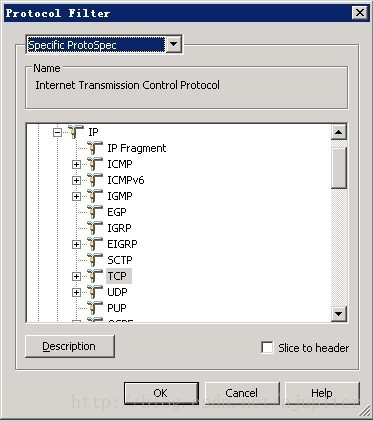


图4  Protocol过滤器

在图2中，可以添加and（必须同时满足）、or（满足其一即可）类型的过滤器。点击and或or，弹出如下菜单：

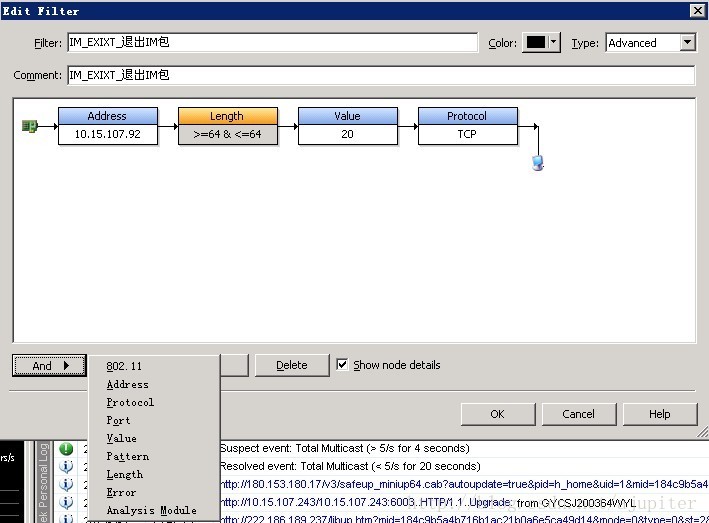


图5  添加过滤条件

在弹出菜单中，有多种过滤器可选择，常见的是使用Value、Pattern、Length方式的过滤器，可分别用来过滤满足指定值、模式、长度的数据包。下面一一说明。

Value过滤器设置：

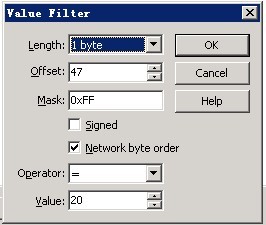


图6  value过滤器设置

Length指明需要过滤的值的长度（按字节），offset指明该值在数据包中的偏移位置（按字节），Mask是掩码，Signed说明值是否是带符号的，Network byte order为缺省选项，可不修改。Operator指明需要过滤的值与value之间的大小关系，一般就是”等于“了。Value值就是需要过滤的数值。

图7是一一个断开连接tcp的数据包。我们需要分析改TCP协议数据包中的tcp flags值，来判断该数据包是用于建立tcp连接，还是断开tcp连接。在本例中，我们的目的是过滤出所有断开tcp连接的数据包。通过分析，得知当tcp flags值是20时，表明它一个断开连接的数据包。tcp flags在数据包中的偏移量是47，长度是1字节。根据这些信息，我们在图6所示编辑框中填写相关选项值。

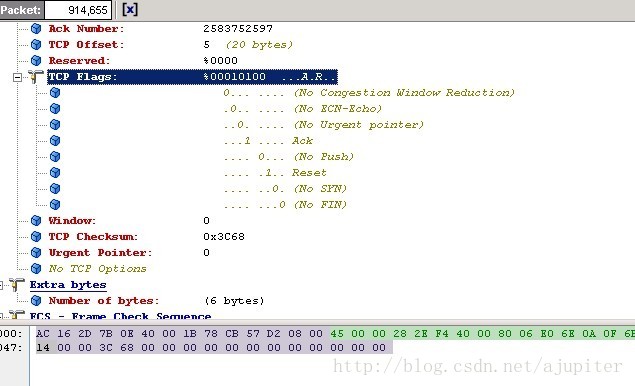


图7  TCP数据包

Length过滤器可用来过滤指定长度的数据包，如图8所示：

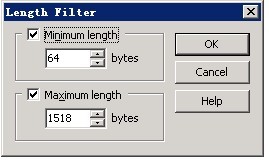


图8  Length过滤器

上图指定了需要过滤的数据包长度是介于64字节到1518字节之间。如果想指定长度为64字节的数据包，则Minmum length和Maximum length均为64。

如果想指明过滤的数据包不能满足某些条件，则可以在进阶设置里，选择某个过滤条件然后点击not按钮。如下图。

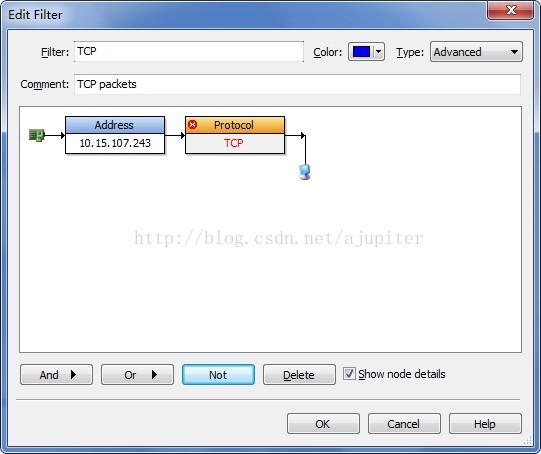


图9  not条件

完成设计后，点击OK按钮。

## 3、应用过滤器

设置完成过滤器后，点击display filter按钮，打开，如下图10：



图10  显示过滤器

点击图10所示display filter按钮后，弹出如图11所示的菜单。点击指定过滤器，则Omnipeek按照该过滤器指定的条件，开始对当前捕获的所有数据包进行过滤。



图11 选择需要应用的过滤器

## 4、分析过滤后的数据包

到现在可以对过滤后的数据包进行一些分析了。Omnipeek本身提供了很多统计功能。

下面12和13是一个统计满足过滤条件的数据包总数的示例。

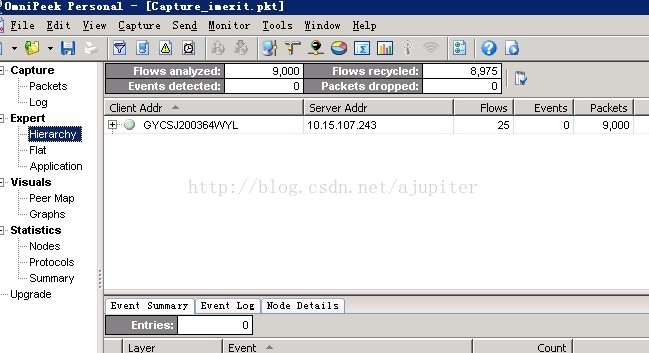


图12  数据包统计

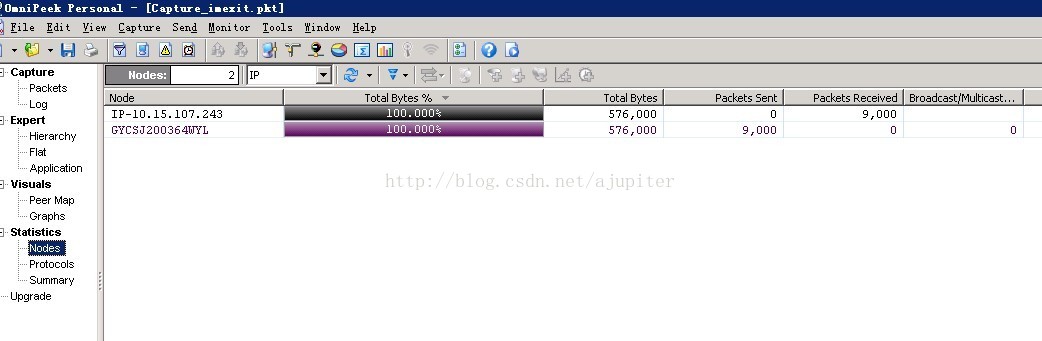


图13  数据包统计

参考：

<http://www.docin.com/p-3608344.html>

<http://wenku.baidu.com/view/3bd23f0290c69ec3d5bb7572.html>