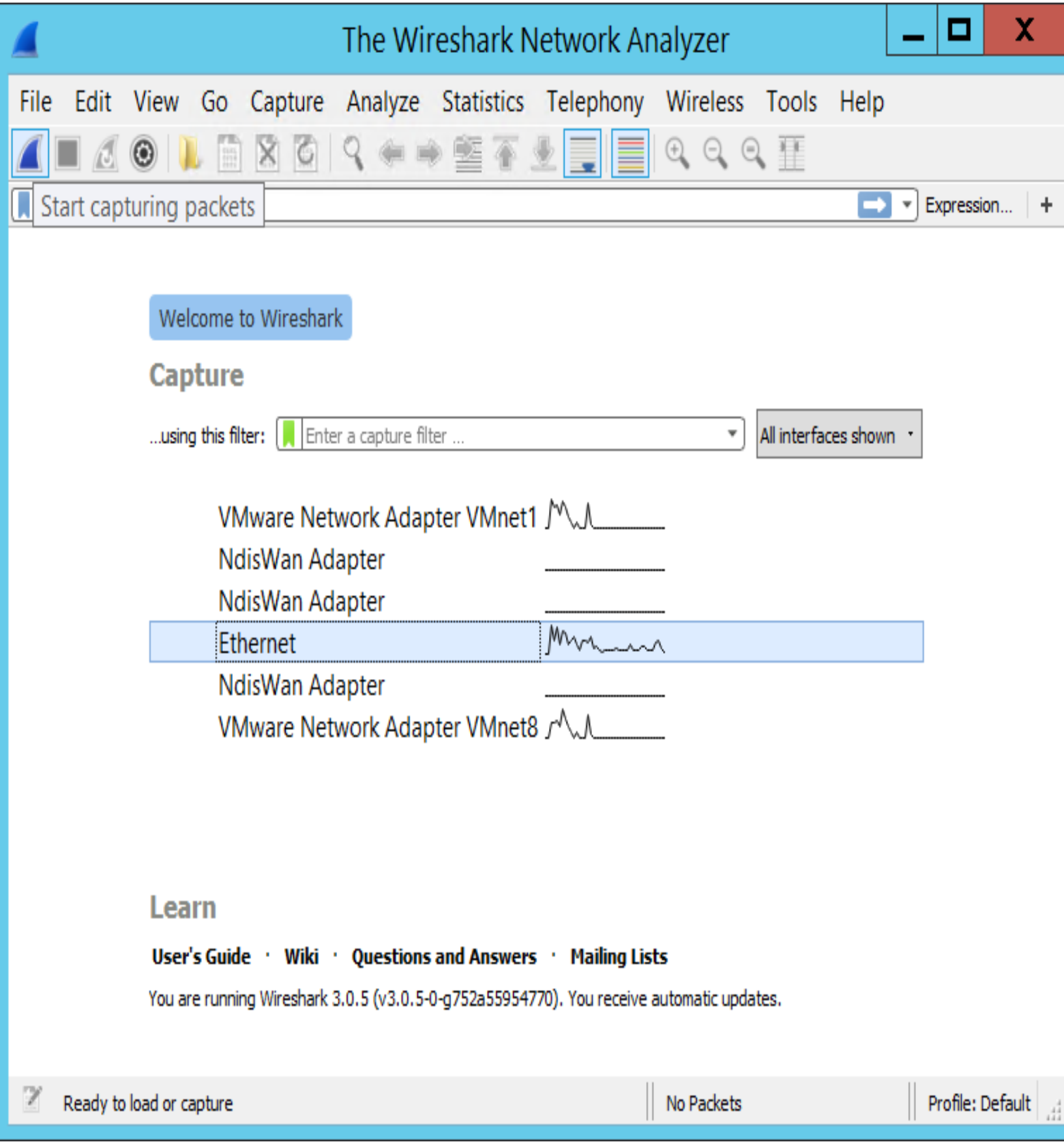
Wireshark

KUHMANN

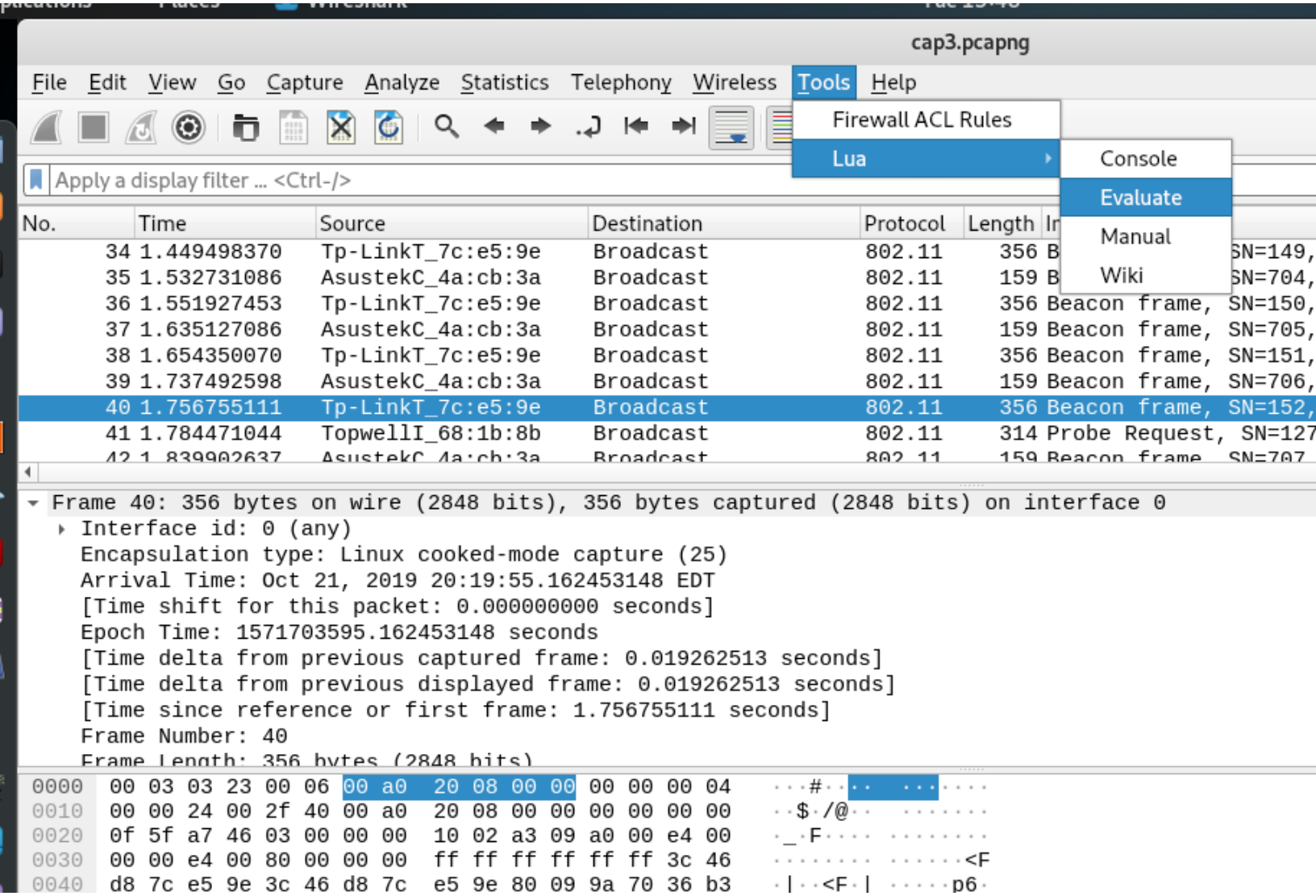
Start Capture

Select Any Interface and Start capturing packets…



Development with LUA

- Open Evaluate Console

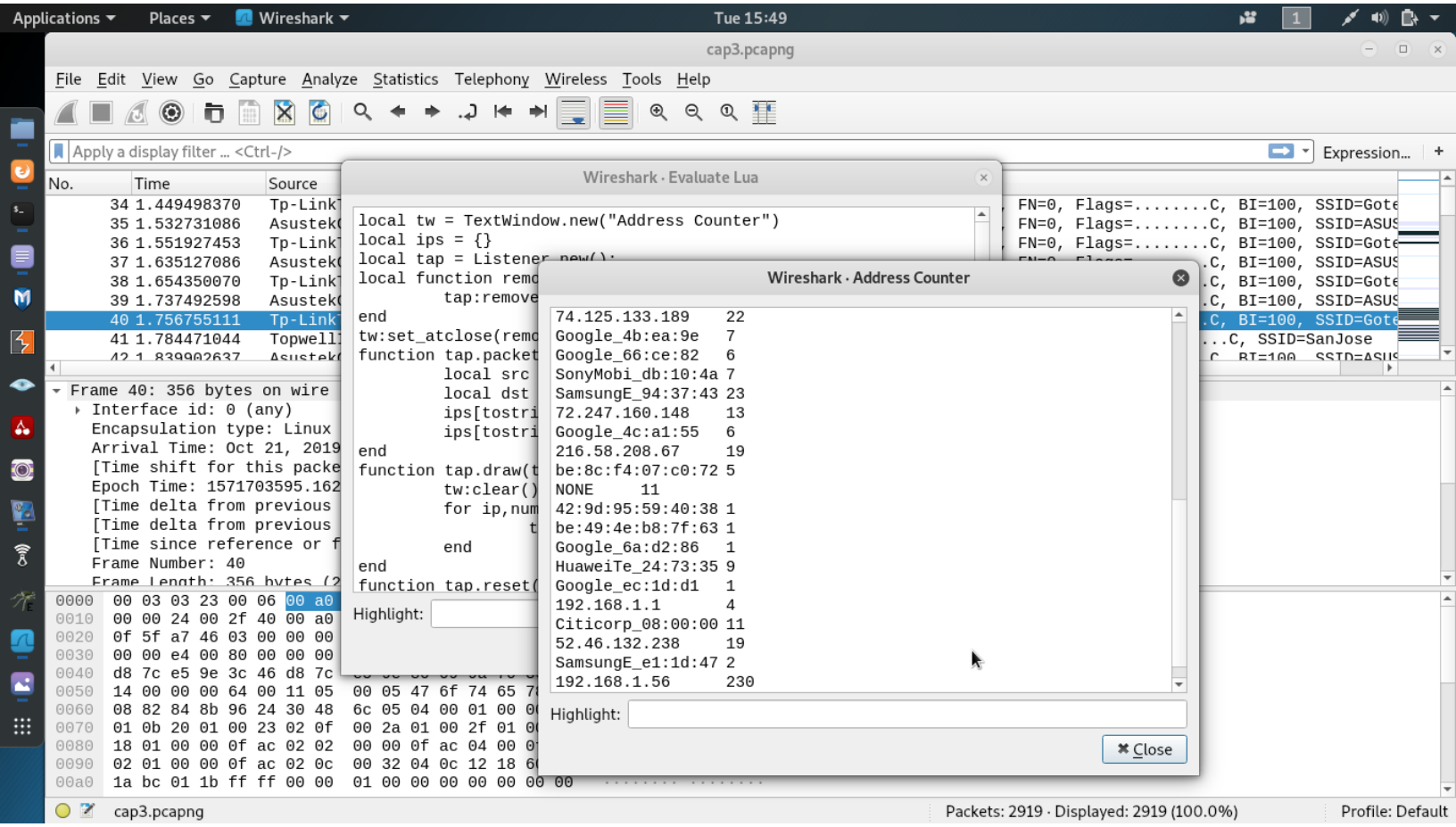


- Paste this code

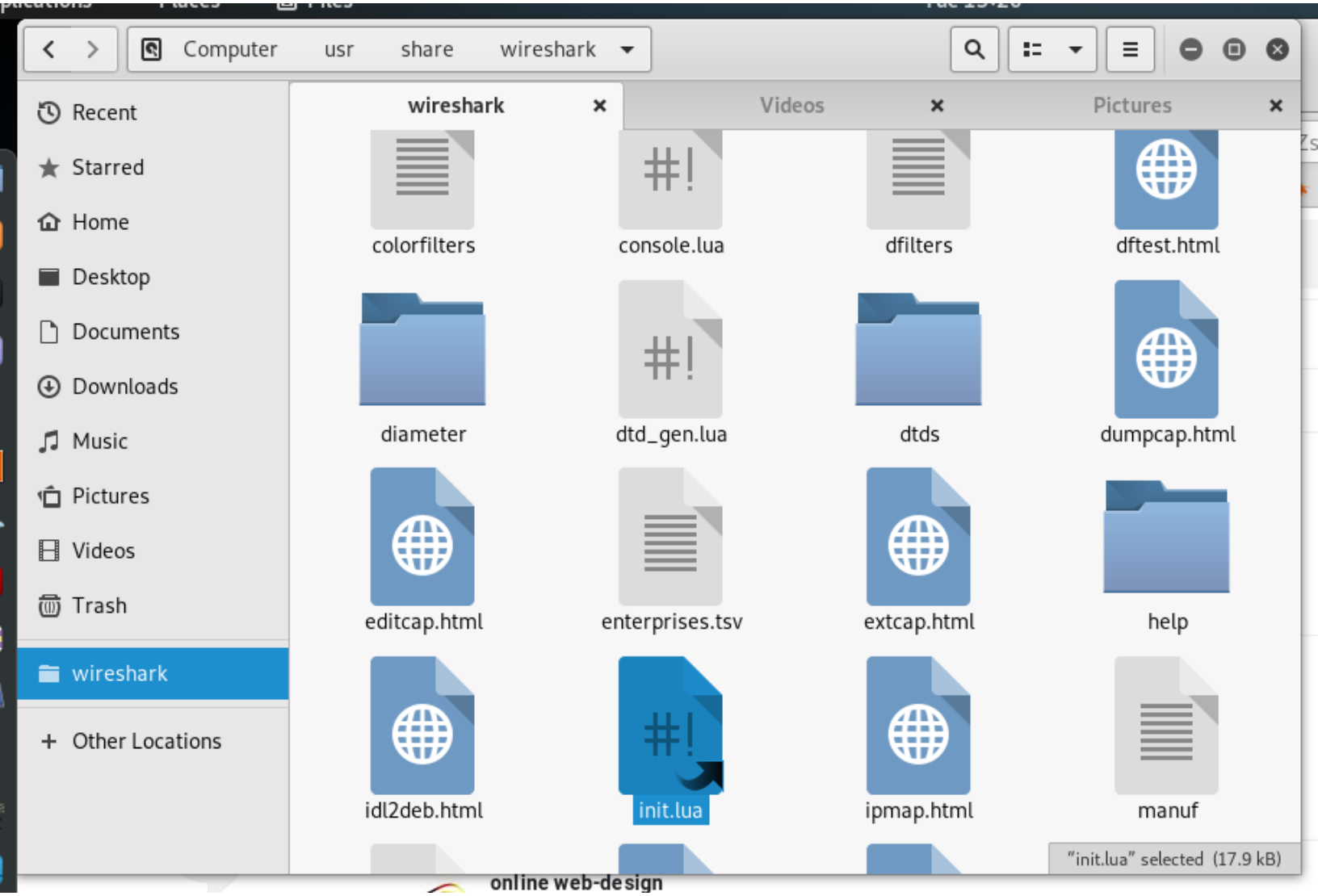
```
local tw = TextWindow.new("Address Counter")
local ips = {}
local tap = Listener.new();
local function remove()
        tap:remove();
end
tw:set_atclose(remove)
function tap.packet(pinfo,tvb)
        local src = ips[tostring(pinfo.src)] or 0
        local dst = ips[tostring(pinfo.dst)] or 0
```

```
        ips[tostring(pinfo.src)] = src + 1
        ips[tostring(pinfo.dst)] = dst + 1
end
function tap.draw(t)
        tw:clear()
        for ip,num in pairs(ips) do
                tw:append(ip .. "\t" .. num .. "\n");
        end
end
function tap.reset()
        tw:clear()
        ips = {}
end
retap_packets()
```

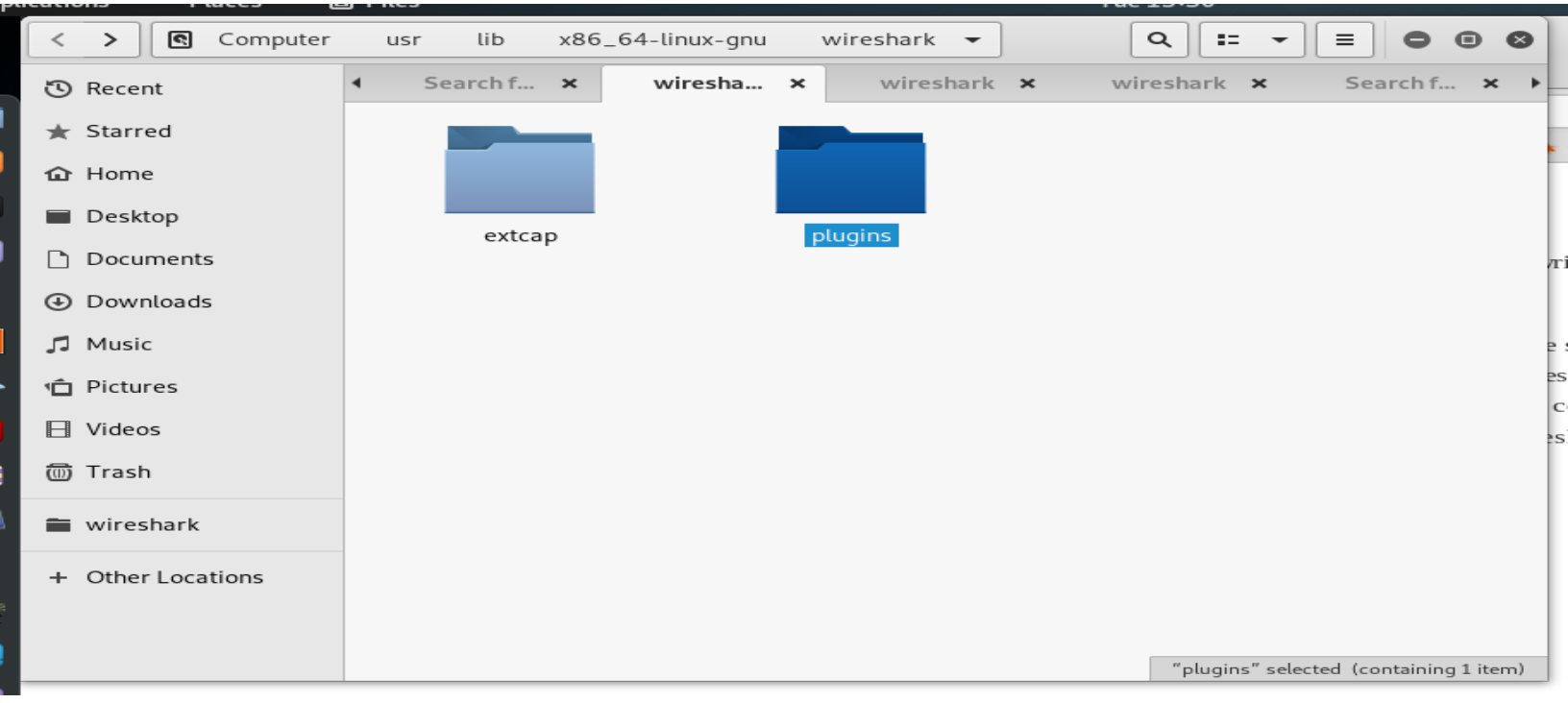
- Click Evaluate button



- Extend LUA



- Automatic Run LUA Scripts in Plugins folder, on Wireshark start



FAST Tutorials

Table 3.13. Main toolbar items

Toolbar Icon	Toolbar Item	Corresponding Menu Item	Description
	Interfaces...	Capture/ Interfaces...	This item brings up the Capture Interfaces List dialog box (discussed further in Section 4.3, "Start Capturing").
	Options...	Capture/Options...	This item brings up the Capture Options dialog box (discussed further in Section 4.3, "Start Capturing") and allows you to start capturing packets.
	Start	Capture/Start	This item starts capturing packets with the options form the last time.
	Stop	Capture/Stop	This item stops the currently running live capture process Section 4.3, "Start Capturing").
	Restart	Capture/Restart	This item stops the currently running live capture process and restarts it again, for convenience.

	Open...	File/Open...	This item brings up the file open dialog box that allows you to load a capture file for viewing. It is discussed in more detail in Section 5.2.1, "The "Open Capture File" dialog box".
	Save As...	File/Save As...	This item allows you to save the current capture file to whatever file you would like. It pops up the Save Capture File As dialog box (which is discussed further in Section 5.3.1, "The "Save Capture File As" dialog box"). **Note!** If you currently have a temporary capture file, the Save icon will be shown instead.

	**Reload**	View/Reload	This item allows you to reload the current ca file.
	**Print...**	File/Print...	This item allows you to print all (or som the packets in the capture file. It pops u Wireshark Print dialog box (which is disc further in Section 5.8, "Printing packets").

	**Find Packet...**	Edit/Find Packet...	This item brings up a dialog box that allow to find a packet. There is further informati finding packets in Section 6.8, "Finding pac
	**Go Back**	Go/Go Back	This item jumps back in the packet history.
	**Go Forward**	Go/Go Forward	This item jumps forward in the packet histor
	**Go to Packet...**	Go/Go to Packet...	This item brings up a dialog box that allow to specify a packet number to go to that pac
	**Go To First Packet**	Go/First Packet	This item jumps to the first packet of the ca file.
	**Go To Last Packet**	Go/Last Packet	This item jumps to the last packet of the ca file.

	**Colorize**	View/Colorize	Colorize the packet list (or not).
	**Auto Scroll in Live Capture**	View/Auto Scroll in Live Capture	Auto scroll packet list while doing a live ca (or not).

	**Zoom In**	View/Zoom In	Zoom into the packet data (increase the font
	**Zoom Out**	View/Zoom Out	Zoom out of the packet data (decrease the size).
	**Normal Size**	View/Normal Size	Set zoom level back to 100%.
	**Resize Columns**	View/Resize Columns	Resize columns, so the content fits into them
---			
	**Capture Filters...**	Capture/Capture Filters...	This item brings up a dialog box that allow to create and edit capture filters. You can

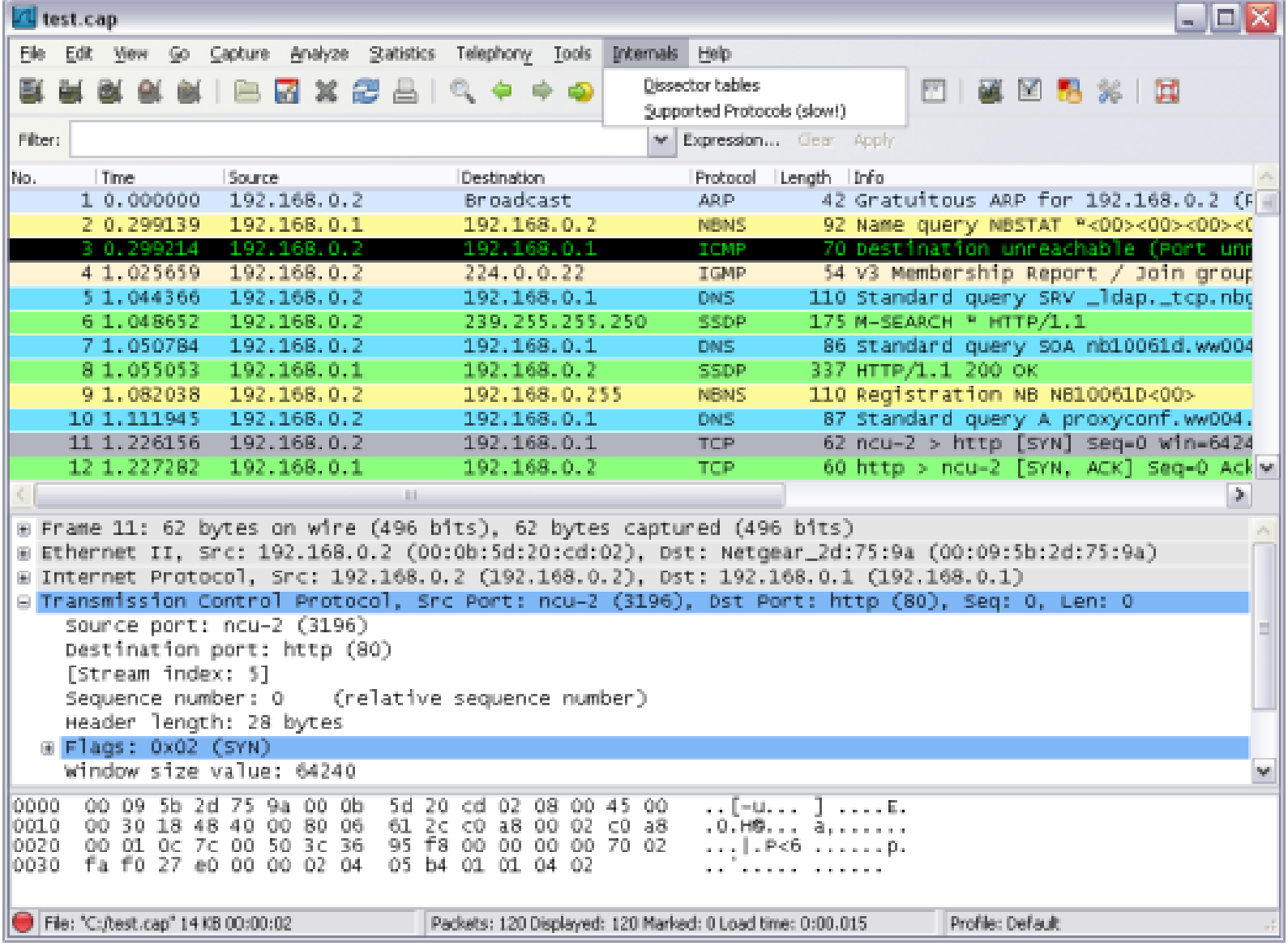


Table 3.11. Help menu items

Menu Item	Accelerator	Description
Dissector tables		This menu item brings up a dialog box showing the tables with subdissector relationships.
Supported Protocols (slow!)		This menu item brings up a dialog box showing the supported protocols and protocol fields.

. The "Help" menu

The Wireshark Help menu contains the fields shown in Table 3.12, "Help menu items".

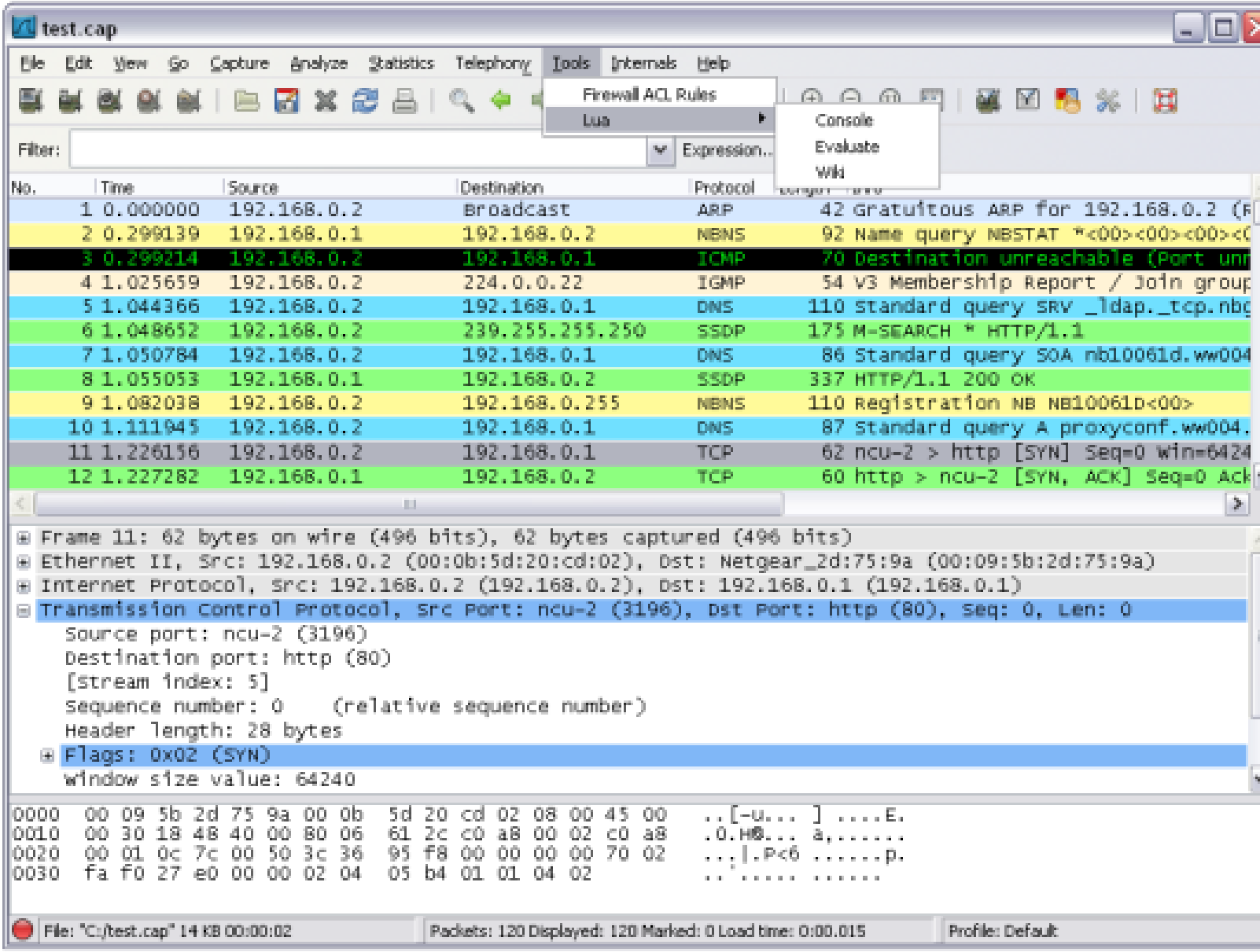
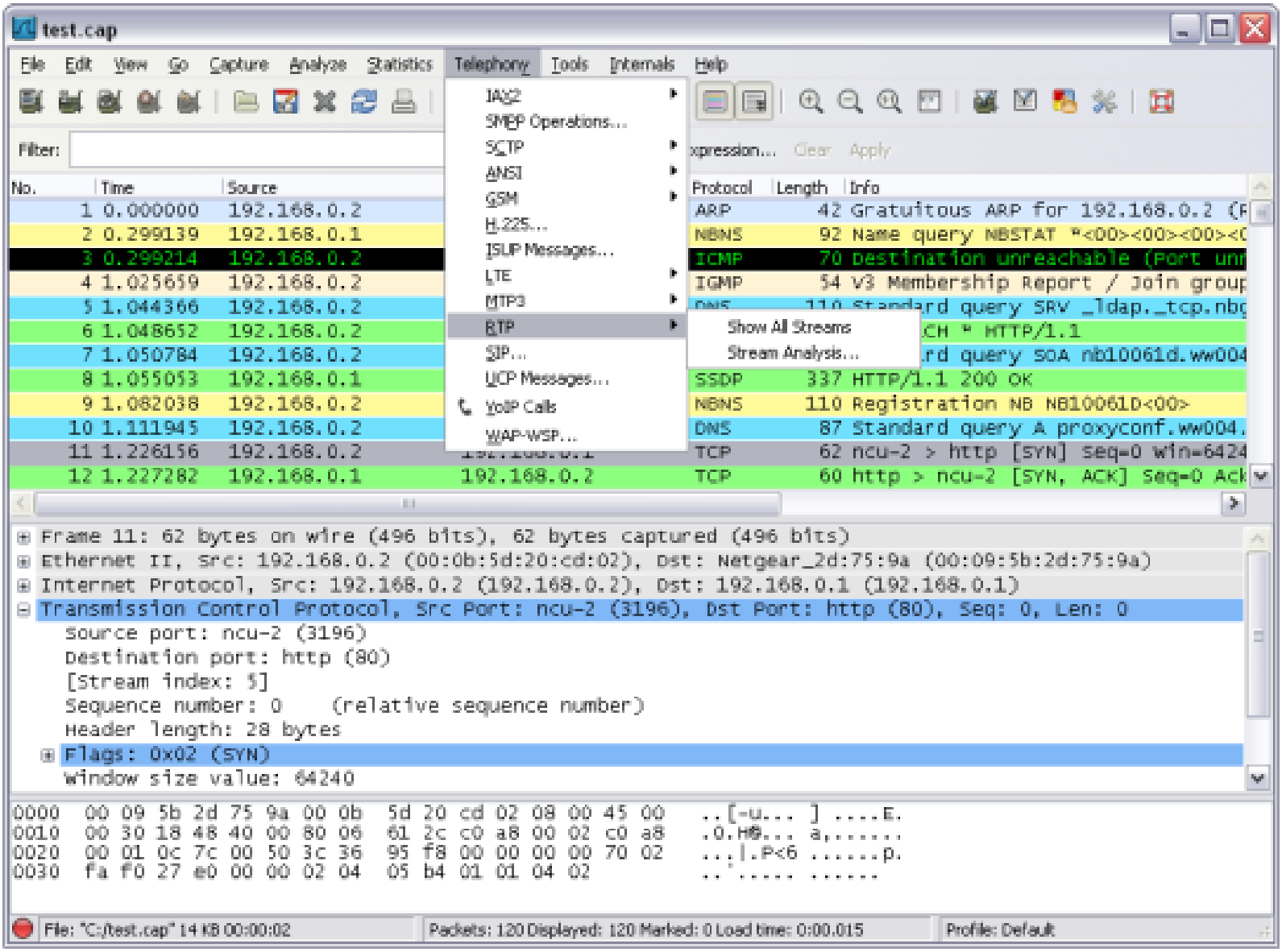


Table 3.10. Tools menu items

Menu Item	Accelerator	Description
Firewall ACL Rules		This allows you to create command-line ACL rules fo many different firewall products, including Cisco IOS, Linu Netfilter (iptables), OpenBSD pf and Windows Firewall (vi netsh). Rules for MAC addresses, IPv4 addresses, TCP an UDP ports, and IPv4+port combinations are supported. It is assumed that the rules will be applied to an outsid interface.
Lua		These options allow you to work with the Lu interpreter optionally build into Wireshark, see Section 11.1 "Introduction".

Figure 3.10. The "Telephony" Menu

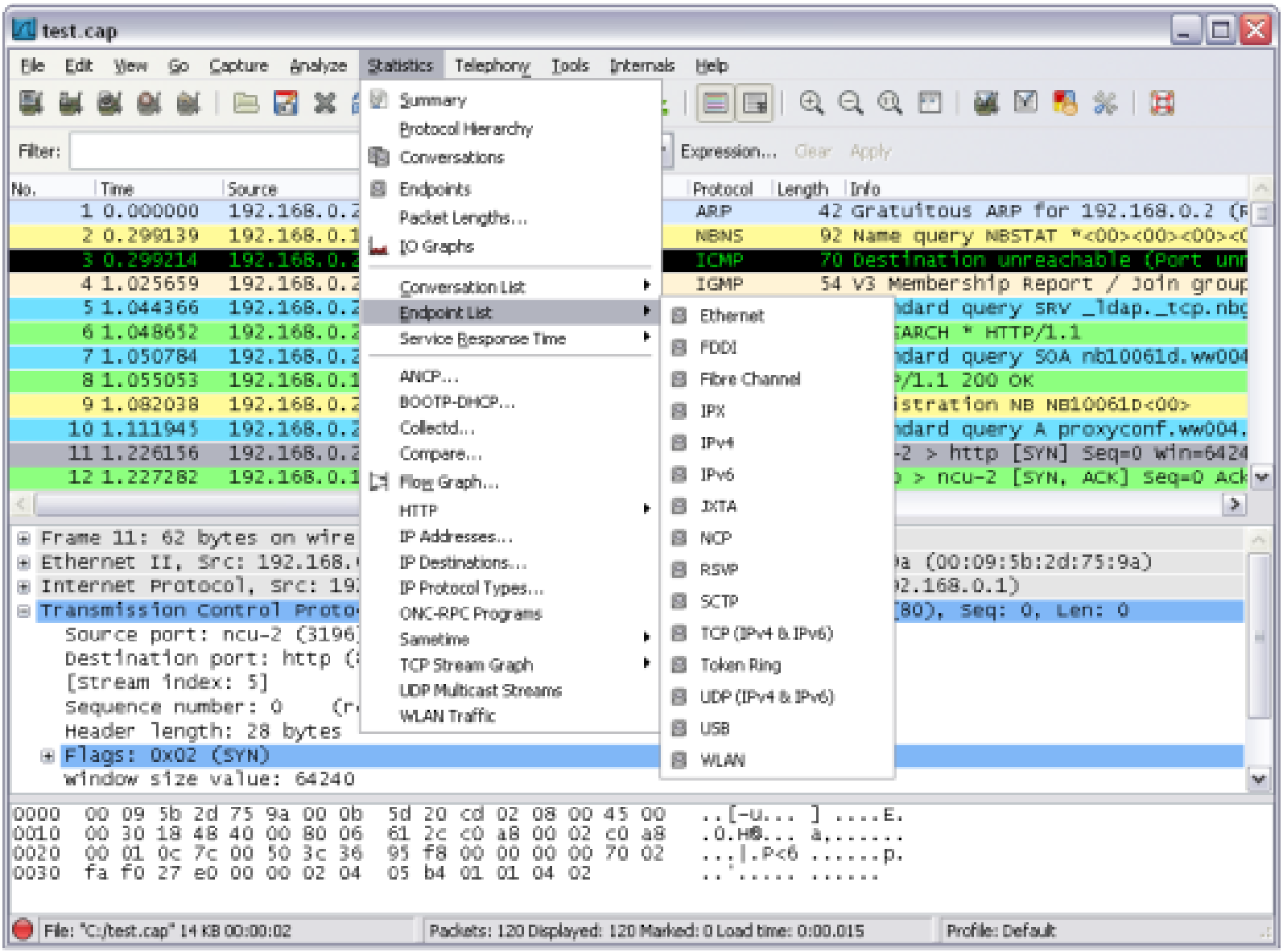


All menu items will bring up a new window showing specific telephony related statistical information.

Table 3.9. Telephony menu items

Menu Item	Accelerator	Description
IAX2		See Section 9.6, "The protocol specific statistics windows"
SMPP Operations...		See Section 9.6, "The protocol specific statistics windows"
SCTP		See Section 9.6, "The protocol specific statistics windows"
ANSI		See Section 9.6, "The protocol specific statistics windows"

Figure 3.9. The "Statistics" Menu



All menu items will bring up a new window showing specific statistical information.

Table 3.8. Statistics menu items

Menu Item	Accelerator	Description
Summary		Show information about the data captured, see Section 8.2, "The "Summary" window".
Protocol Hierarchy		Display a hierarchical tree of protocol statistics, see Section 8.3, "The "Protocol Hierarchy" window".
Conversations		Display a list of conversations (traffic between two endpoints), see Section 8.4.2, "The "Conversations" window".

Figure 3.8. The "Analyze" Menu

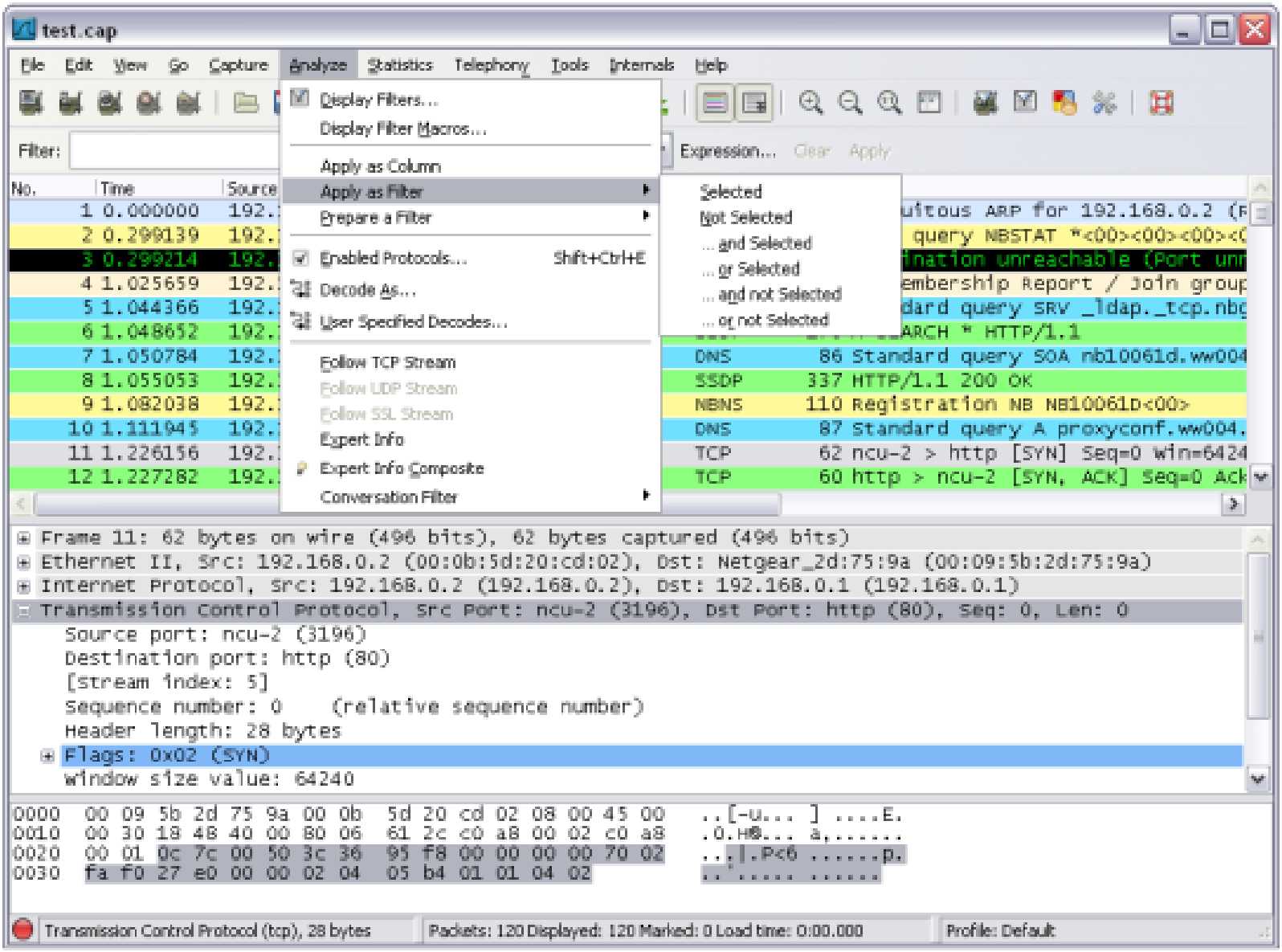


Table 3.7. Analyze menu items

Menu Item	Accelerator	Description
Display Filters...		This menu item brings up a dialog box that allows you to create and edit display filters. You can name filters, and you can save them for future use. More detail on this subject is provided in Section 6.6, "Defining and saving filters"
Display Filter Macros...		This menu item brings up a dialog box that allows you to create and edit display filter macros. You can name filter macros, and you can save them for future use. More detail on this subject is provided in Section 6.7, "Defining and saving filter macros"

Figure 3.7. The "Capture" Menu

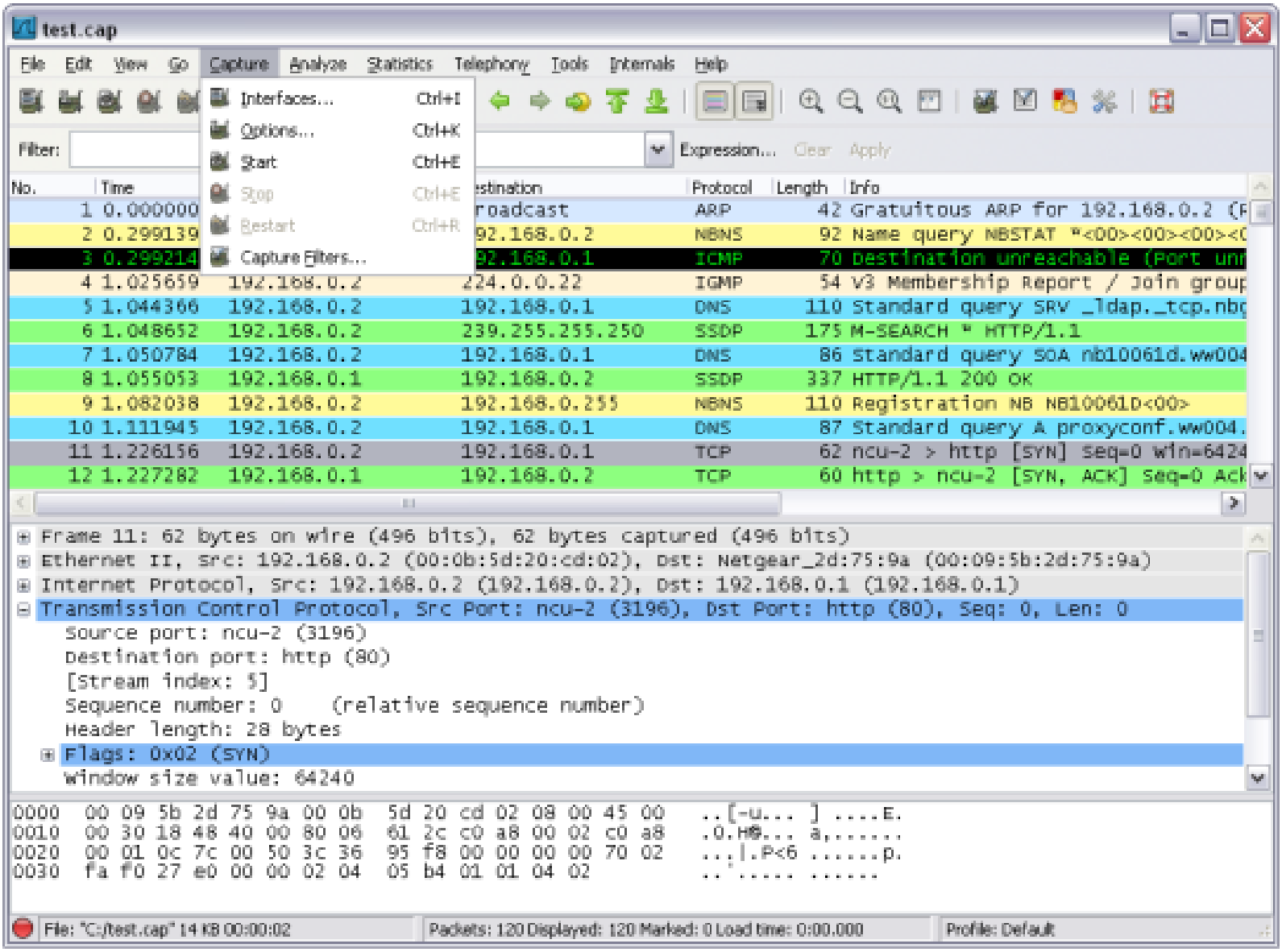


Table 3.6. Capture menu items

Menu Item	Accelerator	Description
Interfaces...	Ctrl+I	This menu item brings up a dialog box that shows what's going on at the network interfaces Wireshark knows of, see Section 4.4, "The "Capture Interfaces" dialog box").
Options...	Ctrl+K	This menu item brings up the Capture Options dialog box (discussed further in Section 4.5, "The "Capture Options" dialog box") and allows you to start capturing packets.
Start	Ctrl+E	Immediately start capturing packets with the same settings than the last time.
Stop	Ctrl+E	This menu item stops the currently running capture, see

Figure 3.6. The "Go" Menu

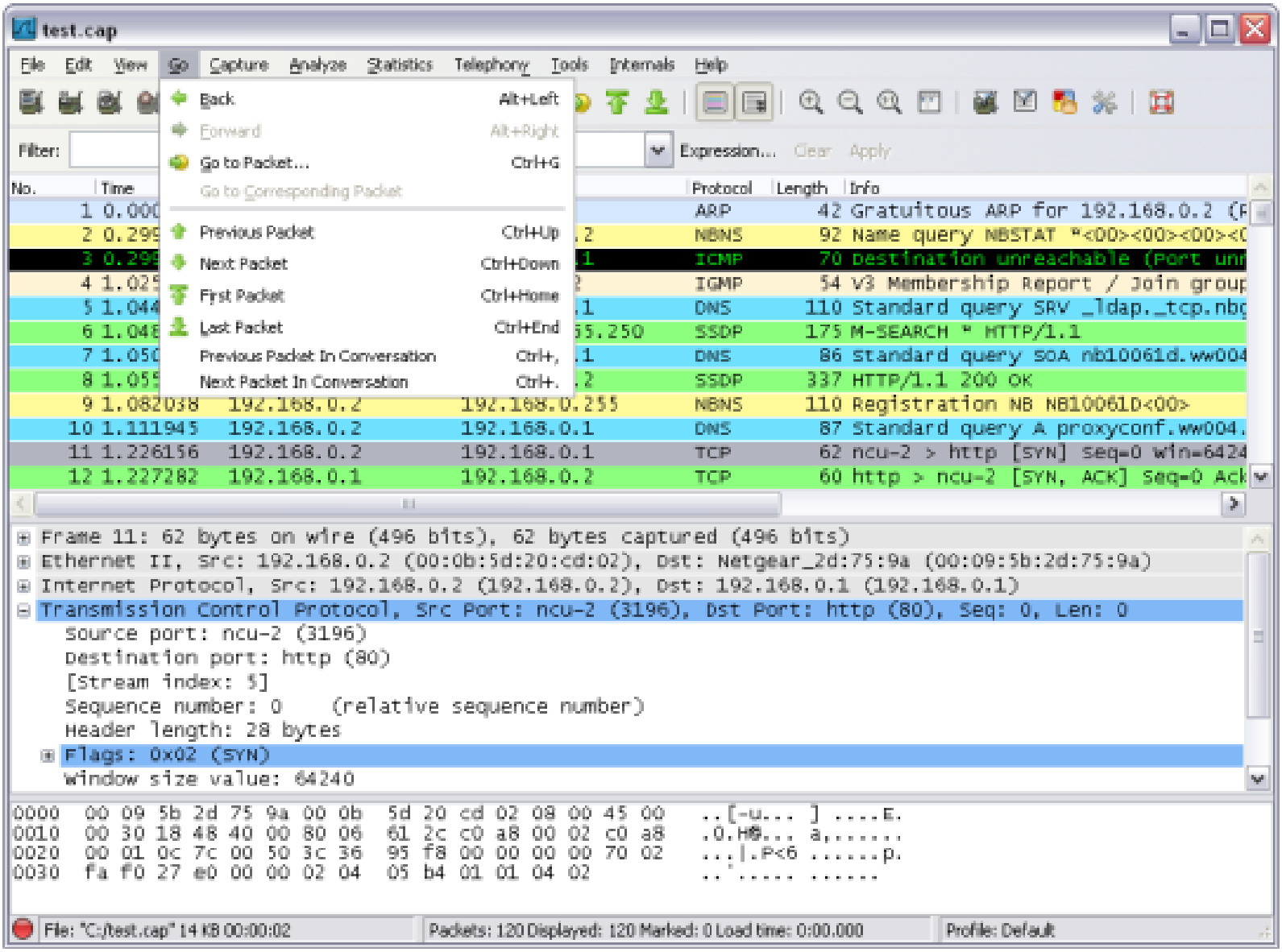


Table 3.5. Go menu items

Menu Item	Accelerator	Description
Back	Alt+Left	Jump to the recently visited packet in the packet history, much like the page history in a web browser.
Forward	Alt+Right	Jump to the next visited packet in the packet history, much like the page history in a web browser.
Go to Packet...	Ctrl+G	Bring up a dialog box that allows you to specify a packet number, and then goes to that packet. See Section 6.9, "Go to a specific packet" for details.

Menu Item	Accelerator	Description
Copy > Description	Shift+Ctrl+D	This menu item will copy the description of the selected item in the detail view to the clipboard.
Copy > Fieldname	Shift+Ctrl+F	This menu item will copy the fieldname of the selected item in the detail view to the clipboard.
Copy > Value	Shift+Ctrl+V	This menu item will copy the value of the selected item in the detail view to the clipboard.
Copy > As Filter	Shift+Ctrl+C	This menu item will use the selected item in the detail view to create a display filter. This display filter is then copied to the clipboard.

Find Packet...	Ctrl+F	This menu item brings up a dialog box that allows you to find a packet by many criteria. There is further information on finding packets in Section 6.8, "Finding packets".

Menu Item	Accelerator	Description
Find Next	Ctrl+N	This menu item tries to find the next packet matching the settings from "Find Packet...".
Find Previous	Ctrl+B	This menu item tries to find the previous packet matching the settings from "Find Packet...".

Mark Packet (toggle)	Ctrl+M	This menu item "marks" the currently selected packet. See Section 6.10, "Marking packets" for details.
Toggle Marking Of	Shift+Ctrl+Alt+M	This menu item toggles the mark on all displayed packets.

The "Filter" toolbar

The filter toolbar lets you quickly edit and apply display filters. More information on display filters is available in Section 6.3, "Filtering packets while viewing".

Figure 3.15. The "Filter" toolbar

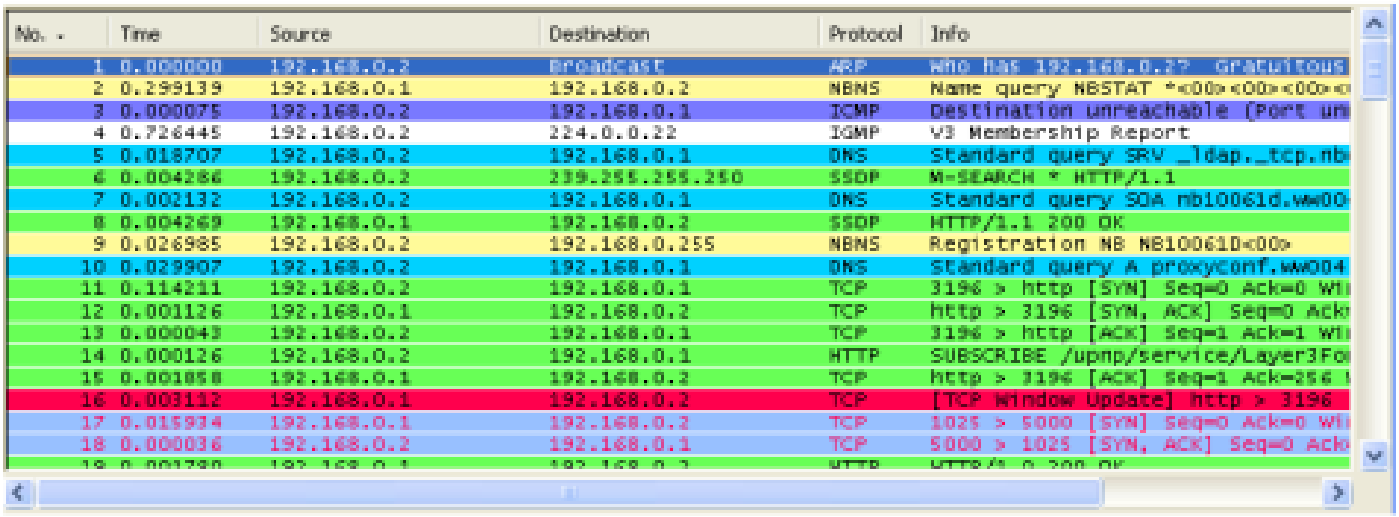
Table 3.14. Filter toolbar items

Toolbar Icon	Toolbar Item	Description
	**Filter:**	Brings up the filter construction dialog, described in Figure 6.8, "The "Capture Filters" and "Display Filters" dialog boxes".
	Filter input	The area to enter or edit a display filter string, see Section 6.4, "Building display filter expressions" . A syntax check of your filter string is done while you are typing. The background will turn red if you enter an incomplete or invalid string, and will become green when you enter a valid string. You can click on the pull down arrow to select a previously-entered filter string from a list. The entries in the pull down list will remain available even after a program restart.  **Note!** After you've changed something in this field, don't forget to press the Apply button (or the Enter/Return key), to apply this filter string to the display.  **Note!** This field is also where the current filter in effect is displayed.
	**Expression...**	The middle button labeled "Add Expression..." opens a dialog box that lets you edit a display filter from a list of protocol fields, described in Section 6.5, "The "Filter Expression" dialog box"
	**Clear**	Reset the current display filter and clears the edit area.
	**Apply**	Apply the current value in the edit area as the new display filter.

3. The "Packet List" pane

The packet list pane displays all the packets in the current capture file.

Figure 3.16. The "Packet List" pane



Each line in the packet list corresponds to one packet in the capture file. If you select a line in this pane, more details will be displayed in the "Packet Details" and "Packet Bytes" panes.

While dissecting a packet, Wireshark will place information from the protocol dissectors into the columns. As higher level protocols might overwrite information from lower levels, you will typically see the information from the highest possible level only.

For example, let's look at a packet containing TCP inside IP inside an Ethernet packet. The Ethernet dissector will write its data (such as the Ethernet addresses), the IP dissector will overwrite this by its own (such as the IP addresses), the TCP dissector will overwrite the IP information, and so on.

There are a lot of different columns available. Which columns are displayed can be selected by preference settings, see Section 10.5, "Preferences".

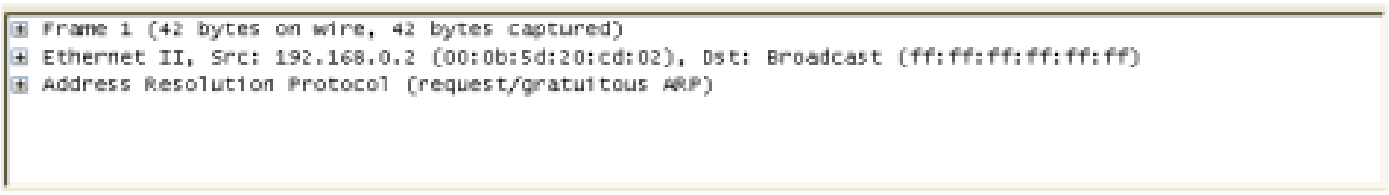
The default columns will show:

- **No.** The number of the packet in the capture file. This number won't change, even if a display filter is used.
- **Time** The timestamp of the packet. The presentation format of this timestamp can be changed, see Section 6.12, "Time display formats and time references".
- **Source** The address where this packet is coming from.

. The "Packet Details" pane

The packet details pane shows the current packet (selected in the "Packet List" pane) in a more detailed form.

Figure 3.17. The "Packet Details" pane



This pane shows the protocols and protocol fields of the packet selected in the "Packet List" pane. The protocols and fields of the packet are displayed using a tree, which can be expanded and collapsed.

There is a context menu (right mouse click) available, see details in Figure 6.5, “Pop-up menu of the "Packet Details" pane”.

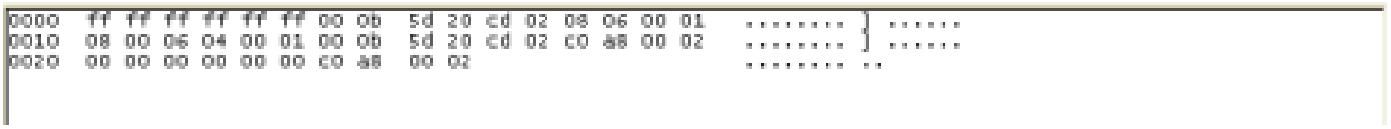
Some protocol fields are specially displayed.

- **Generated fields** Wireshark itself will generate additional protocol fields which are surrounded by brackets. The information in these fields is derived from the known context to other packets in the capture file. For example, Wireshark is doing a sequence/acknowledge analysis of each TCP stream, which is displayed in the [SEQ/ACK analysis] fields of the TCP protocol.
- **Links** If Wireshark detected a relationship to another packet in the capture file, it will generate a link to that packet. Links are underlined and displayed in blue. If double-clicked, Wireshark jumps to the corresponding packet.

. The "Packet Bytes" pane

The packet bytes pane shows the data of the current packet (selected in the "Packet List" pane) in a hexdump style.

Figure 3.18. The "Packet Bytes" pane

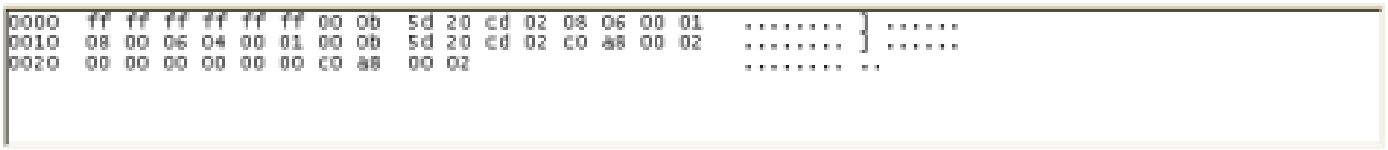


to the corresponding packet.

. The "Packet Bytes" pane

The packet bytes pane shows the data of the current packet (selected in the "Packet List" pane) in a hexdump style.

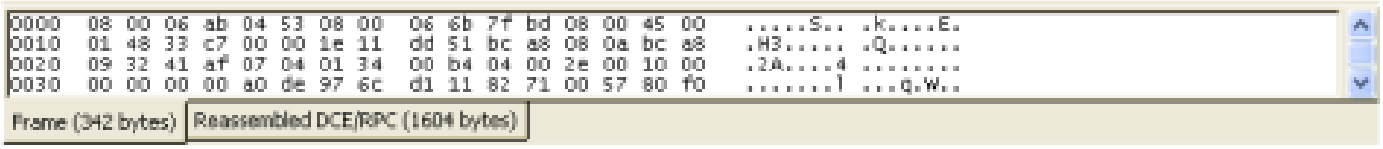
Figure 3.18. The "Packet Bytes" pane



As usual for a hexdump, the left side shows the offset in the packet data, in the middle the packet data is shown in a hexadecimal representation and on the right the corresponding ASCII characters (or . if not appropriate) are displayed.

Depending on the packet data, sometimes more than one page is available, e.g. when Wireshark has reassembled some packets into a single chunk of data, see Section 7.6, "Packet Reassembling". In this case there are some additional tabs shown at the bottom of the pane to let you select the page you want to see.

Figure 3.19. The "Packet Bytes" pane with tabs



Note!

The additional pages might contain data picked from multiple packets.

The context menu (right mouse click) of the tab labels will show a list of all available pages. This can be helpful if the size in the pane is too small for all the tab labels.

. The Statusbar

The statusbar displays informational messages.

In general, the left side will show context related information, the middle part will show the current number of packets, and the right side will show the selected configuration profile. Drag the handles between the text areas to change the size.

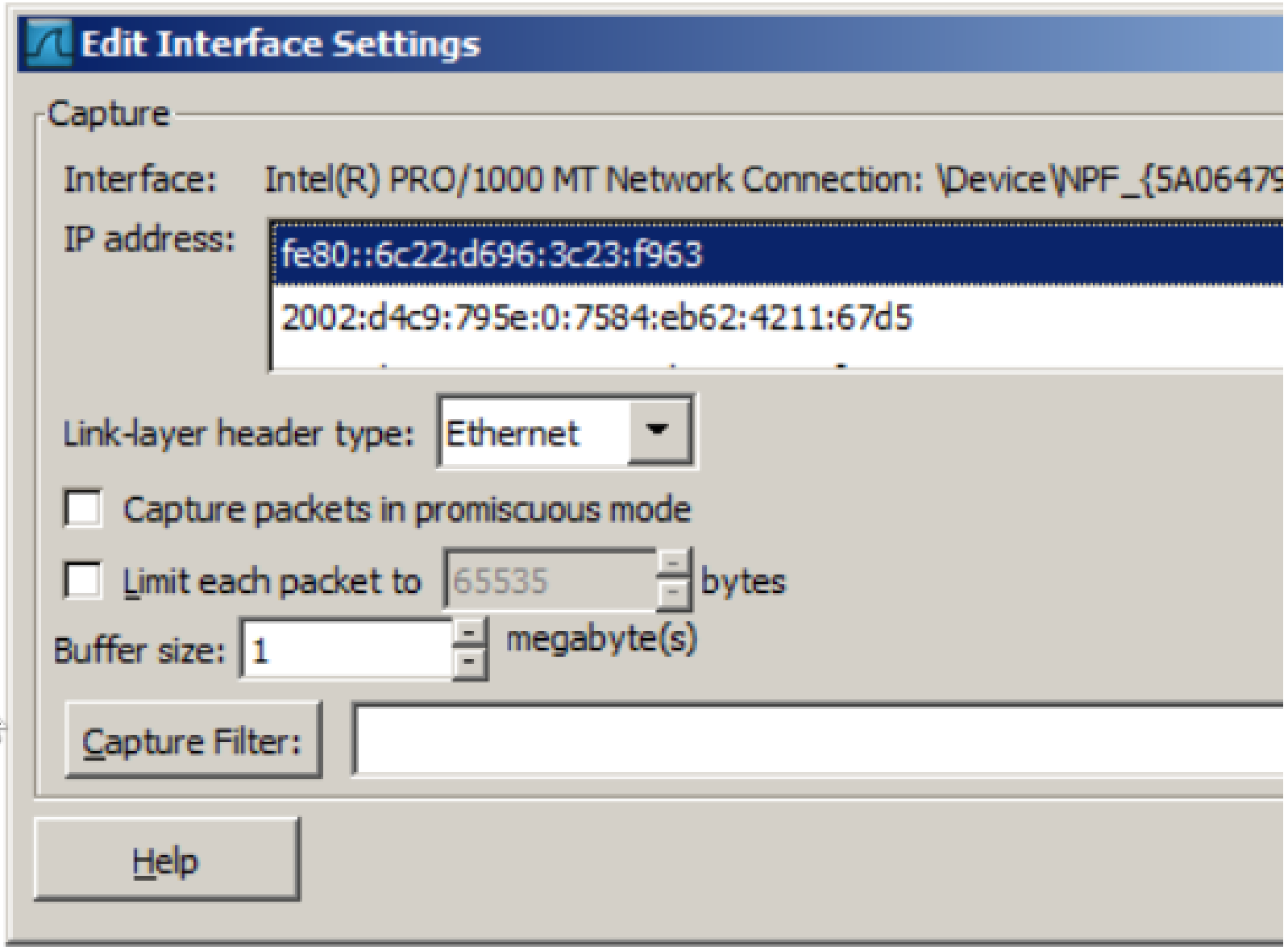
Figure 3.20. The initial Statusbar

This statusbar is shown while no capture file is loaded, e.g. when Wireshark is started.

Figure 3.21. The Statusbar with a loaded capture file

- **The colorized bullet** on the left shows the highest expert info level found in the currently loaded capture file. Hovering the mouse over this icon will show a textual description of the expert info level, and clicking the icon will bring up the Expert Infos dialog box. For a detailed description of expert info, see Section 7.3, "Expert Infos".
- **The left side** shows information about the capture file, its name, its size and the elapsed time while it was being captured.
- **The middle part** shows the current number of packets in the capture file. The following values are displayed:
  - *Packets:* the number of captured packets
  - *Displayed:* the number of packets currently being displayed
  - *Marked:* the number of marked packets
  - *Dropped:* the number of dropped packets (only displayed if Wireshark was unable to capture all packets)
  - *Ignored:* the number of ignored packets (only displayed if packets are ignored)

Figure 4.4. The "Edit Interface Settings" dialog box



You can set the following fields in this dialog box:

IP address — The IP address(es) of the selected interface. If no address could be resolved from the system, "none" will be shown.

Link-layer header type — Unless you are in the rare situation that you need this, just keep the default. For a detailed description, see Section 4.12, "Link-layer header type"

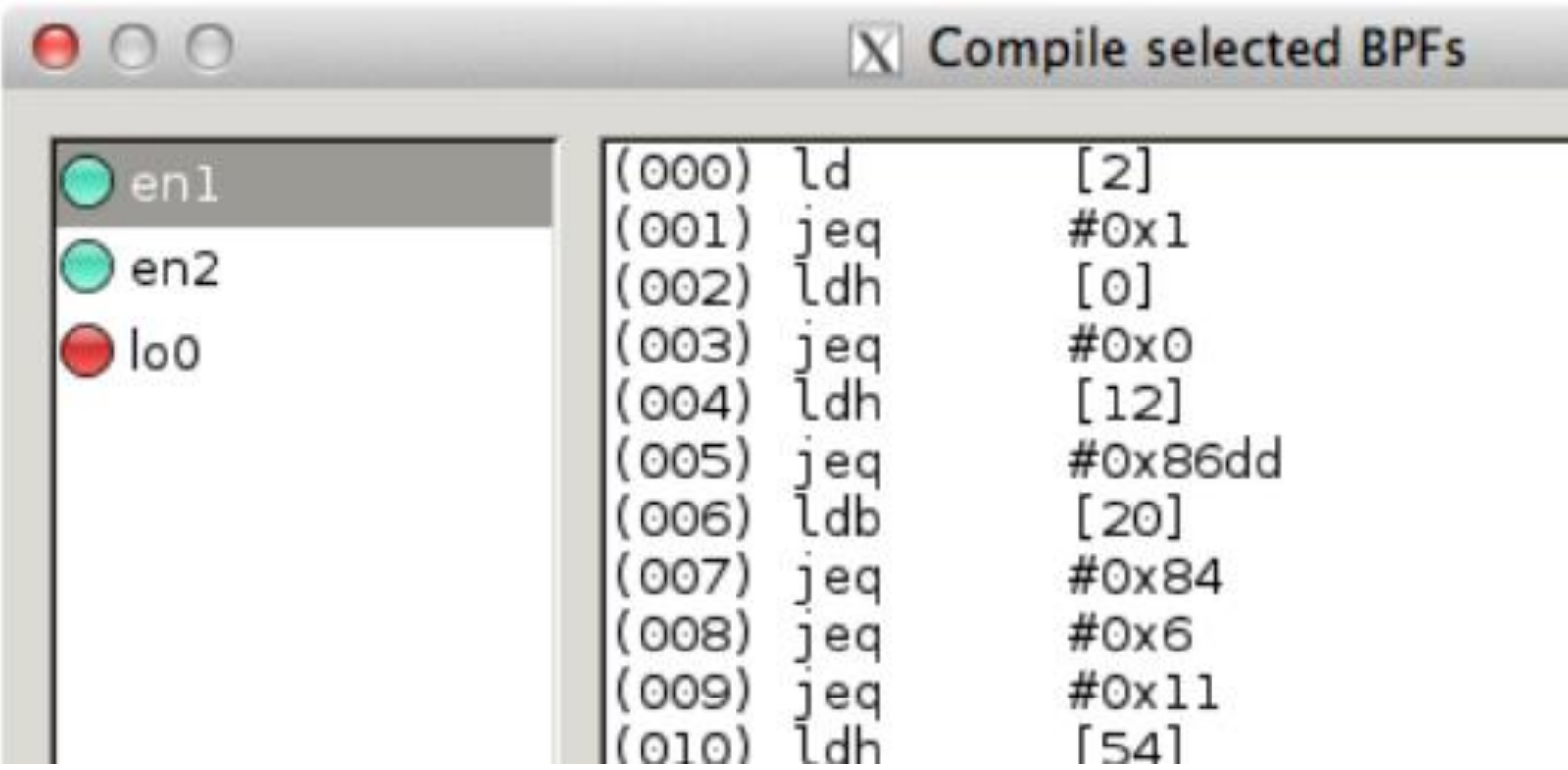
Wireless settings (Windows only) — Here you can set the settings for wireless capture using the

Compile BPF	This button allows you to compile the capture filter into BPF code and pop up a window showing you the resulting pseudo code. This can help in understanding the working of the capture filter you created.

The "Compile Results" dialog box

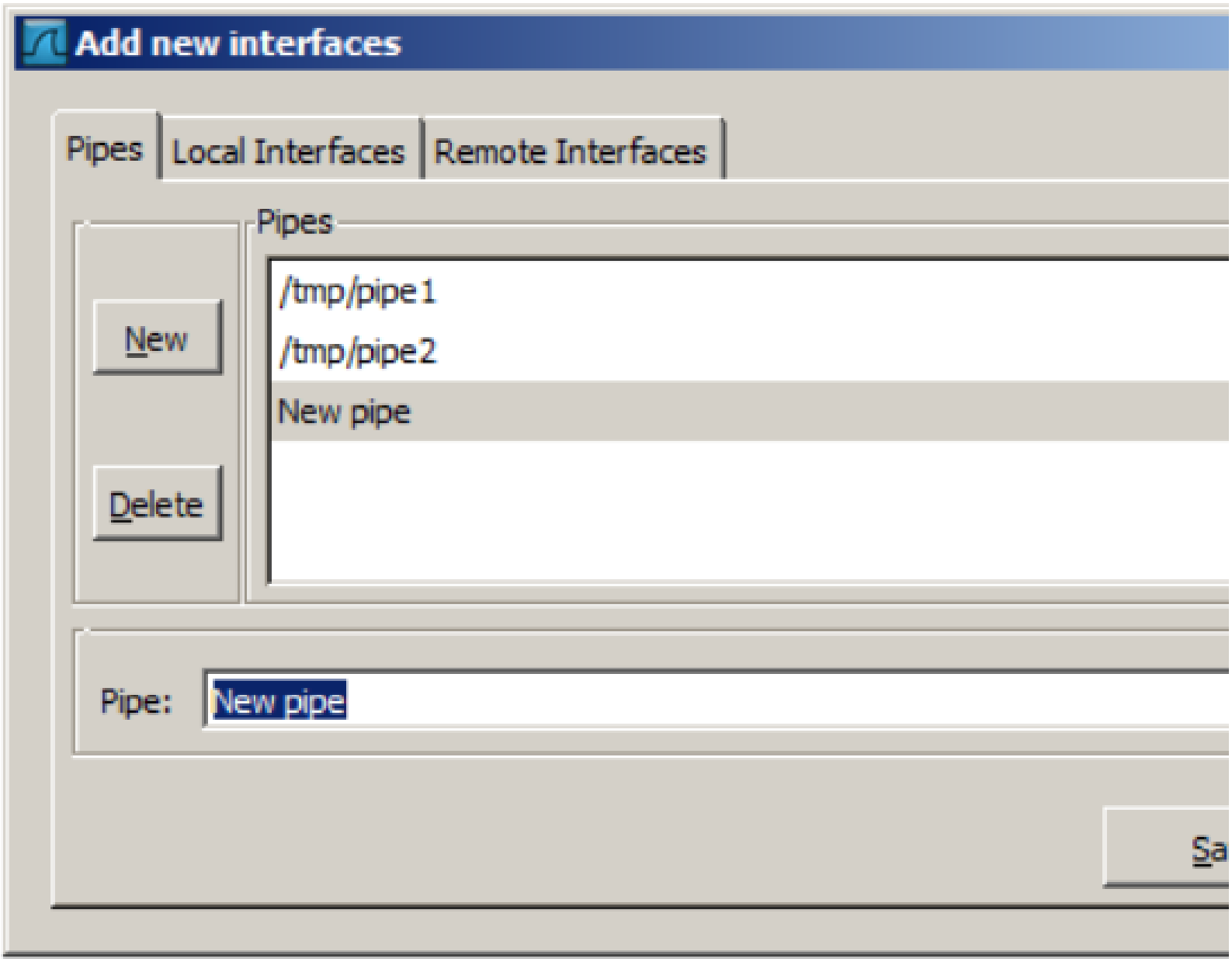
This figure shows the compile results of the selected interfaces.

Figure 4.5. The "Compile Results" dialog box



. Add or remove pipes

Figure 4.7. The "Add New Interfaces - Pipes" dialog box

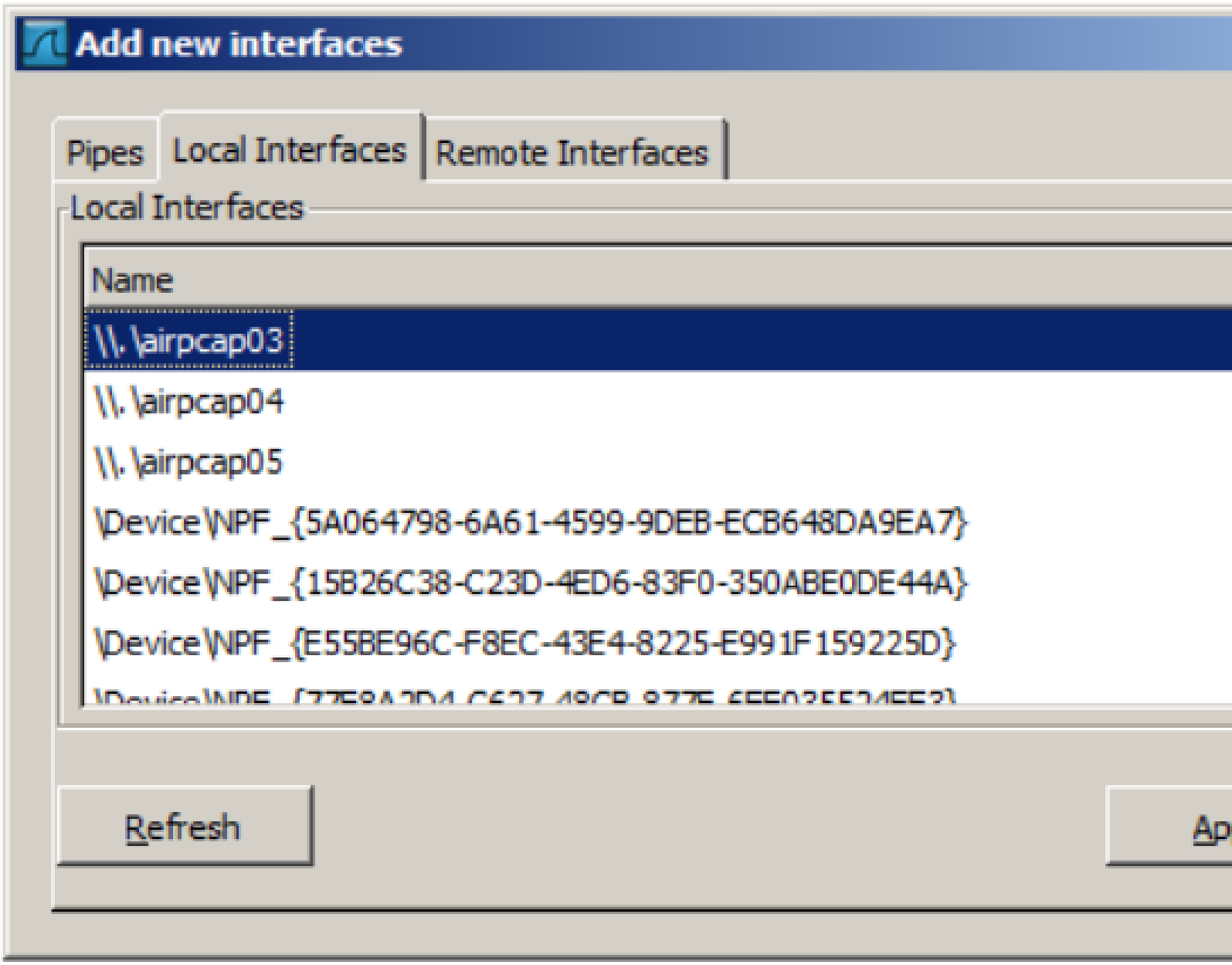


To successfully add a pipe, this pipe must have already been created. Click the "New" button and type the name of the pipe including its path. Alternatively, the "Browse" button can be used to locate the pipe. With the "Save" button the pipe is added to the list of available interfaces. Afterwards, other pipes can be added.

To remove a pipe from the list of interfaces it first has to be selected. Then click the "Delete" button.

. Add or hide local interfaces

Figure 4.8. The "Add New Interfaces - Local Interfaces" dialog box



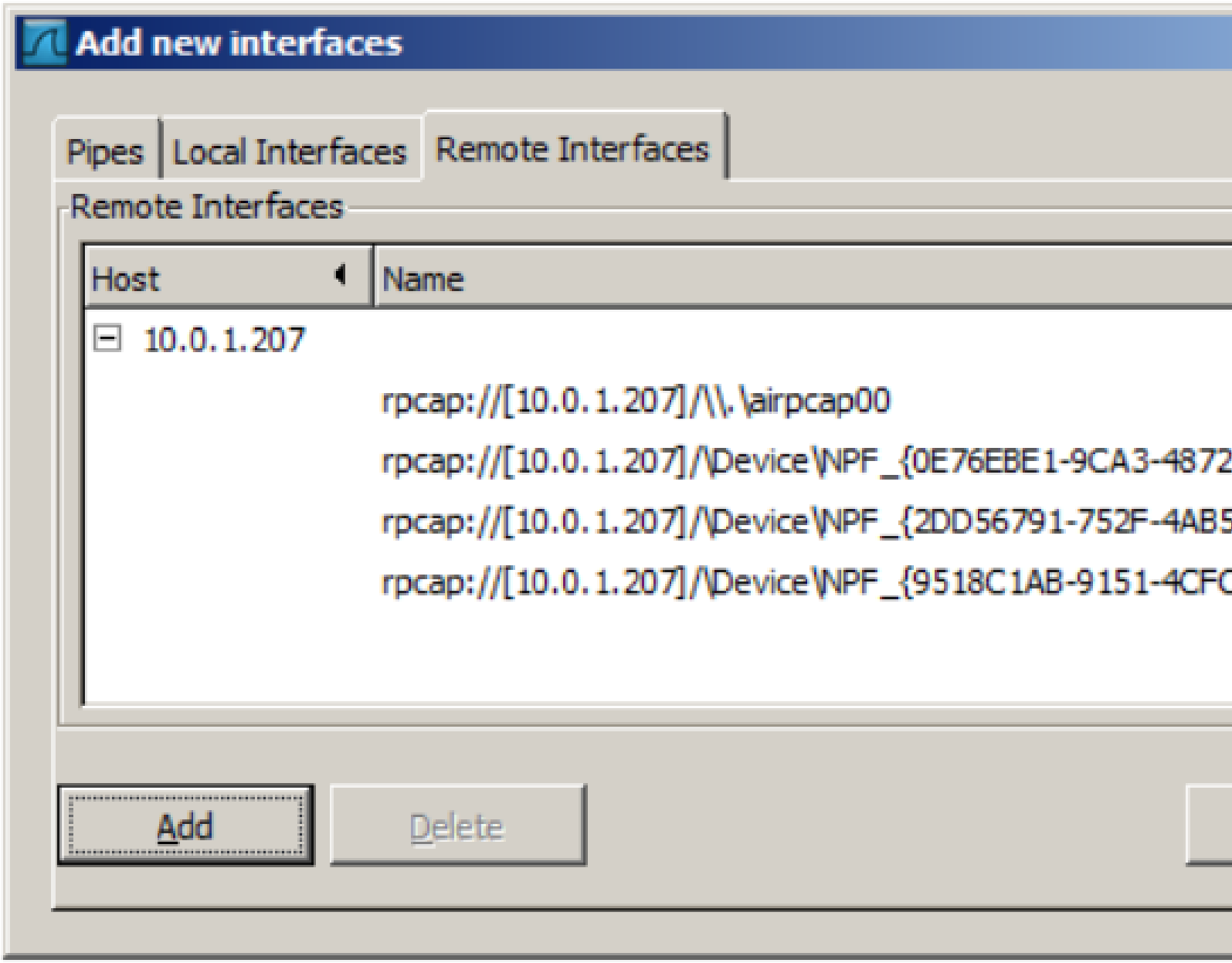
The tab "Local Interfaces" contains a list of available local interfaces, including the hidden ones, which are not shown in the other lists.

If a new local interface is added, for example, a wireless interface has been activated, it is not automatically added to the list to prevent the constant scanning for a change in the list of available interfaces. To renew the list a rescan can be done.

One way to hide an interface is to change the preferences. If the "Hide" checkbox is activated and the "Apply" button clicked, the interface will not be seen in the lists of the "Capture Options" or "Capture

. Add or hide remote interfaces

Figure 4.9. The "Add New Interfaces - Remote Interfaces" dialog box



In this tab interfaces on remote hosts can be added. One or more of these interfaces can be hidden. In contrast to the local interfaces they are not saved in the "Preferences" file.

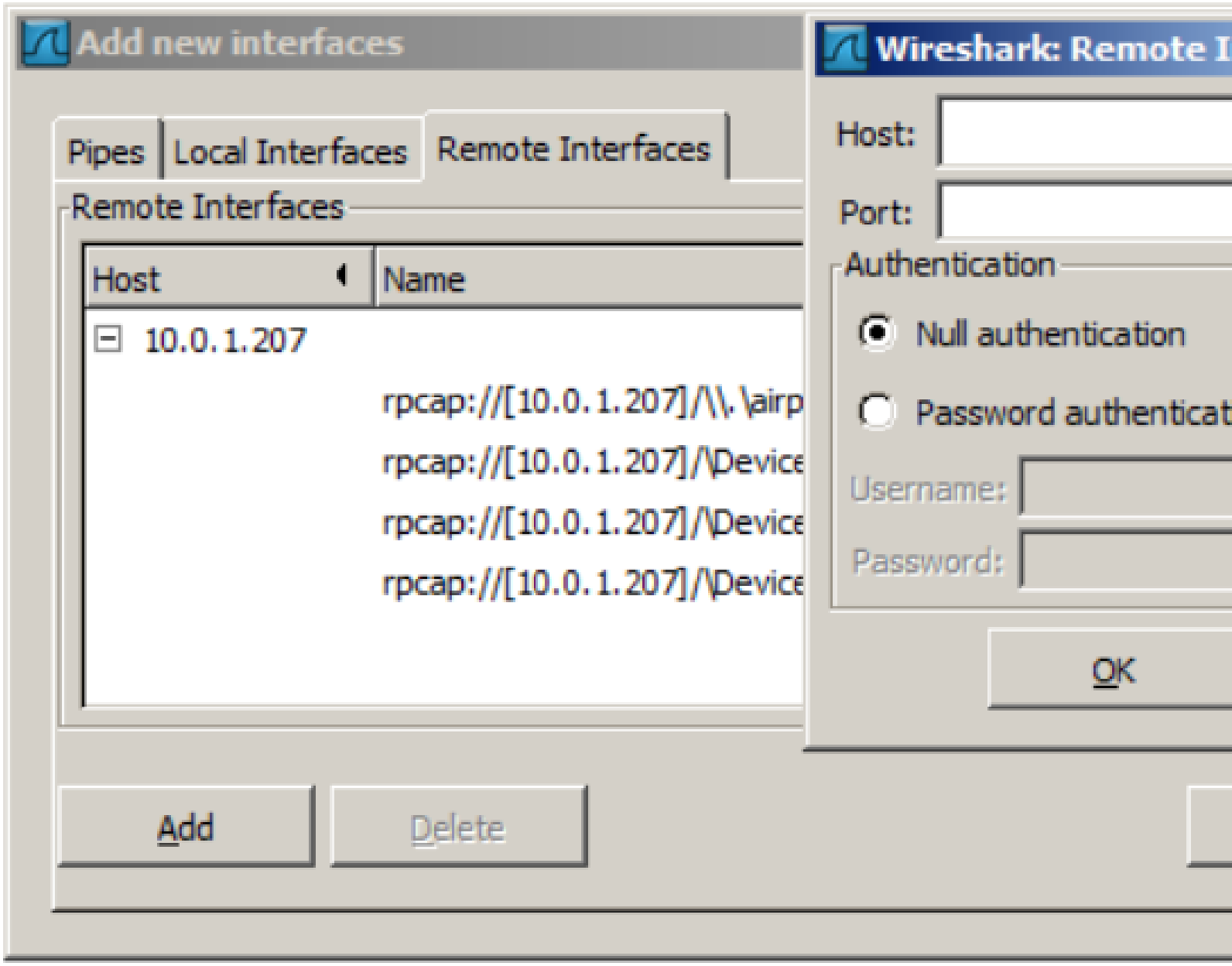
To remove a host including all its interfaces from the list, it has to be selected. Then click the "Delete" button.

For a detailed description, see Section 4.9, "The "Remote Capture Interfaces" dialog box"

To access the Remote Capture Interfaces dialog use the "Add New Interfaces - Remote" dialog, see Figure 4.9, "The "Add New Interfaces - Remote Interfaces" dialog box", and select "Add".

. Remote Capture Interfaces

Figure 4.10. The "Remote Capture Interfaces" dialog box



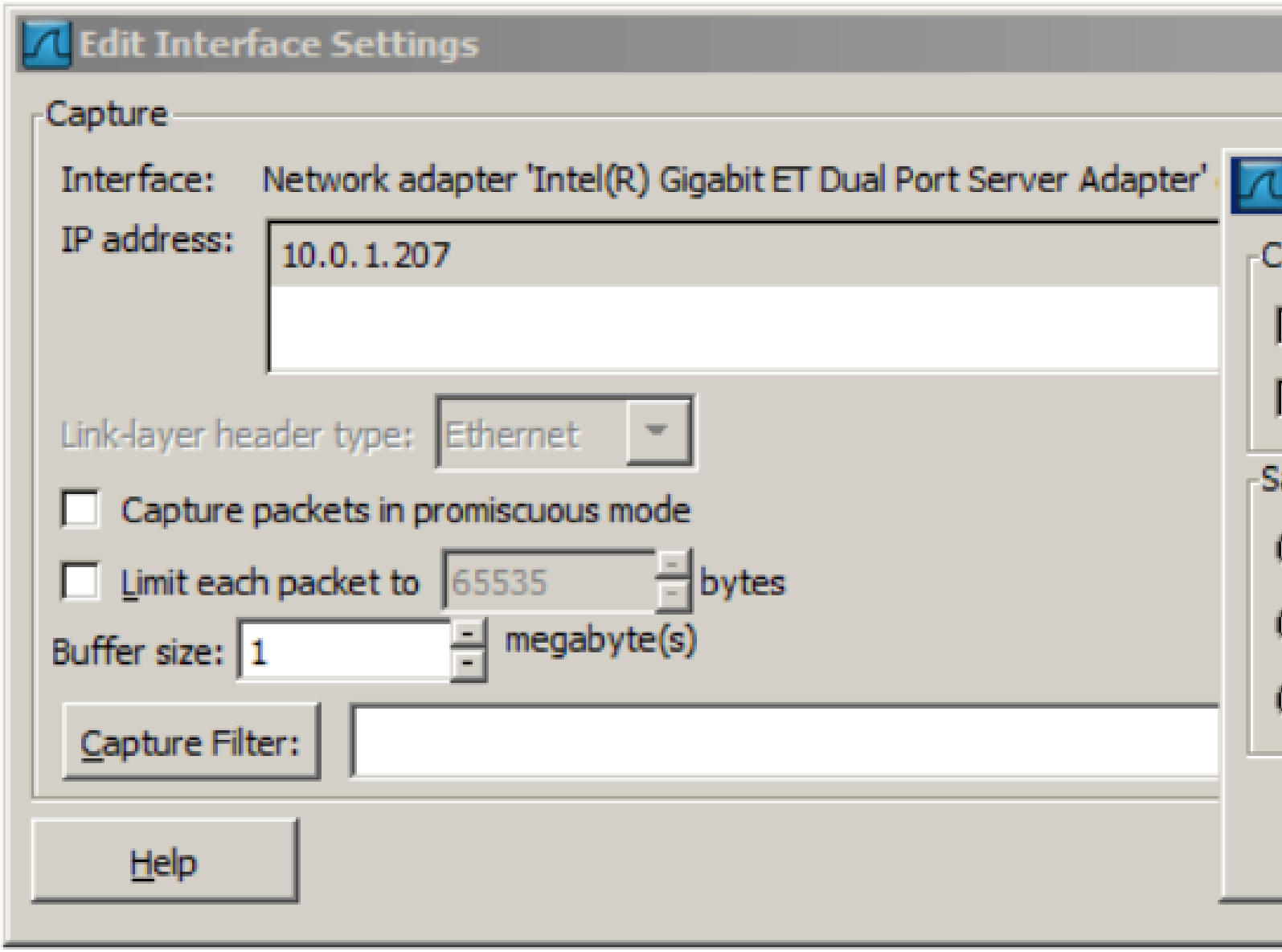
You have to set the following parameter in this dialog:

Host — Enter the IP address or host name of the target platform where the Remote Packet Capture Protocol service is listening. The drop down list contains the hosts that have previously been successfully contacted. The list can be emptied by choosing "Clear list" from the drop down list.

. Remote Capture Settings

The remote capture can be further fine tuned to match your situation. The **Remote Settings** button in Figure 4.4, “The "Edit Interface Settings" dialog box” gives you this option. It pops up the dialog shown in Figure 4.11, “The "Remote Capture Settings" dialog box”.

Figure 4.11. The "Remote Capture Settings" dialog box



You can set the following parameters in this dialog:

Do not capture own RPCAP traffic This option sets a capture filter so that the traffic flowing back from the Remote Packet Capture Protocol service to Wireshark

milliseconds

to send only a sub sampling of the captured data, in terms of time. This allows capture over a narrow band capture session of a higher bandwidth interface.

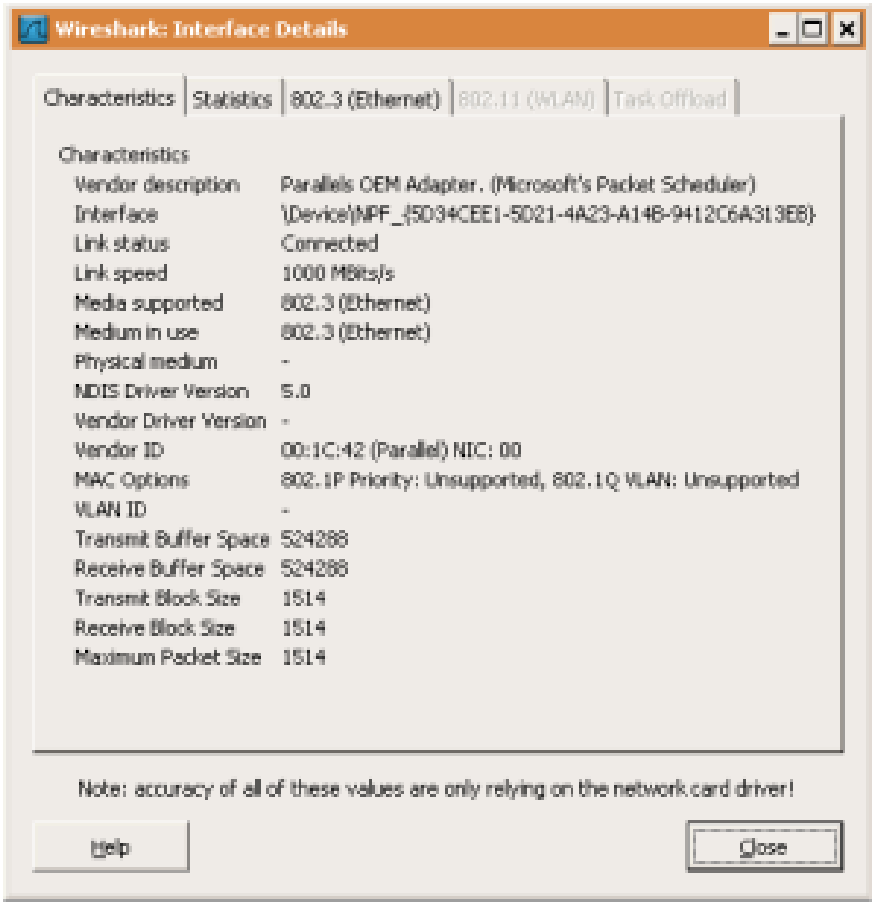
. The "Interface Details" dialog box

When you select Details from the Capture Interface menu, Wireshark pops up the "Interface Details" dialog box as shown in Figure 4.12, "The "Interface Details" dialog box". This dialog shows various characteristics and statistics for the selected interface.

Microsoft Windows only

This dialog is only available on Microsoft Windows

Figure 4.12. The "Interface Details" dialog box



. Capture files and file modes

While capturing, the underlying libpcap capturing engine will grab the packets from the network card

Information about the folders used for the capture file(s), can be found in Appendix A, *Files and Folders*.

Table 4.1. Capture file mode selected by capture options

"File" option	"Use multiple files" option	"Ring buffer with n files" option	Mode	Resulting filename(s) used
-	-	-	**Single temporary file**	wiresharkXXXXXX (where XXXXXX is a unique number)
foo.cap	-	-	**Single named file**	foo.cap
foo.cap	x	-	**Multiple files, continuous**	foo_00001_20100205 foo_00002_20100205
foo.cap	x	x	**Multiple files, ring buffer**	foo_00001_20100205 foo_00002_20100205

Single temporary file

A temporary file will be created and used (this is the default). After the capturing is stopped, this file can be saved later under a user specified name.

Single named file

A single capture file will be used. If you want to place the new capture file to a specific folder, choose this mode.

Multiple files, continuous

Like the "Single named file" mode, but a new file is created and used, after reaching one of the multiple file switch conditions (one of the "Next file every ..." values).

Multiple files, ring buffer

Much like "Multiple files continuous", reaching one of the multiple files switch conditions (one of the "Next file every ..." values) will switch to the next file. This will be a newly created file if value of "Ring buffer with n files" is not reached, otherwise it will replace the oldest of the formerly used files (thus forming a "ring").

This mode will limit the maximum disk usage, even for an unlimited amount of capture input data, keeping the latest captured data.

. Link-layer header type

http://www.tcpdump.org/tcpdump_man.html.

A capture filter takes the form of a series of primitive expressions connected by conjunctions (**and/or**) and optionally preceded by **not**:

```
[not] primitive [and|or [not] primitive ...]
```

An example is shown in Example 4.1, " A capture filter for telnet that captures traffic to and from a particular host ".

Example 4.1. A capture filter for telnet that captures traffic to and from a particular host

```
tcp port 23 and host 10.0.0.5
```

This example captures telnet traffic to and from the host 10.0.0.5, and shows how to use two primitives and the **and** conjunction. Another example is shown in Example 4.2, " Capturing all telnet traffic not from 10.0.0.5", and shows how to capture all telnet traffic except that from 10.0.0.5.

Example 4.2. Capturing all telnet traffic not from 10.0.0.5

```
tcp port 23 and not src host 10.0.0.5
```

XXX - add examples to the following list.

A primitive is simply one of the following:

[src|dst] host <host> This primitive allows you to filter on a host IP address or name. You can optionally precede the primitive with the keyword **src|dst** to specify that you are only interested in

DISPLAY (x11) [remote name]:<display num>

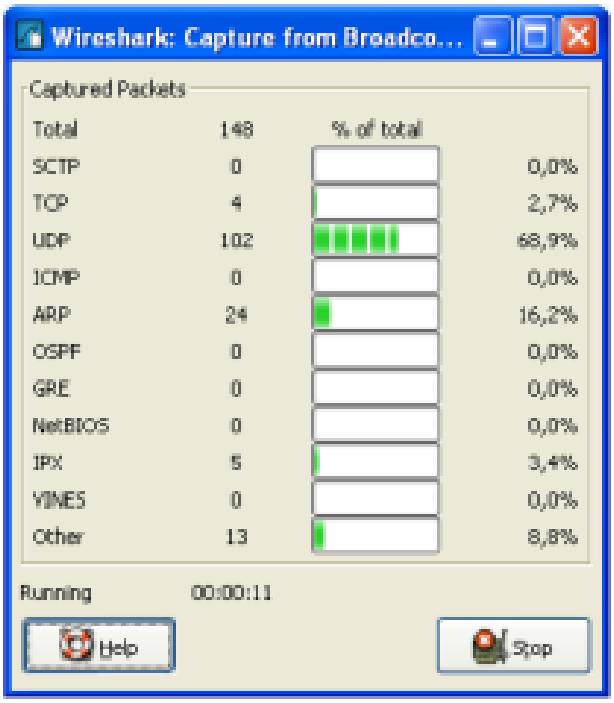
SESSIONNAME (terminal server) <remote name>

On Windows it asks the operating system if it's running in a Remote Desktop Services environment.

. While a Capture is running ...

While a capture is running, the following dialog box is shown:

Figure 4.13. The "Capture Info" dialog box



This dialog box will inform you about the number of captured packets and the time since the capture was started. The selection of which protocols are counted cannot be changed.

Tip!

This Capture Info dialog box can be hidden, using the "Hide capture info dialog" option in the Capture Options dialog box.

1. Stop the running capture

A running capture session will be stopped in one of the following ways:

1. Using the "Stop" button from the Capture Info dialog box

apter 5. File Input / Output and nting

Introduction

This chapter will describe input and output of capture data.

- Open capture files in various capture file formats
- Save/Export capture files in various capture file formats
- Merge capture files together
- Import text files containing hex dumps of packets
- Print packets

Open capture files

Wireshark can read in previously saved capture files. To read them, simply select the menu or toolbar item: "File/ **Open**". Wireshark will then pop up the File Open dialog box, which is discussed in more detail in Section 5.2.1, "The "Open Capture File" dialog box".

It's convenient to use drag-and-drop!

... to open a file, by simply dragging the desired file from your file manager and dropping it onto Wireshark's main window. However, drag-and-drop is not available/won't work in all desktop environments.

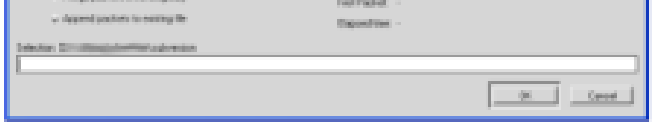
If you haven't previously saved the current capture file, you will be asked to do so, to prevent data loss (this behaviour can be disabled in the preferences).

In addition to its native file format (libpcap format, also used by tcpdump/WinDump and other libpcap/WinPcap-based programs), Wireshark can read capture files from a large number of other packet

. Input File Formats

The following file formats from other capture tools can be opened by Wireshark:

- libpcap - captures from *Wireshark/TShark/dumpcap*, *tcpdump*, and various other tools using libpcap's/tcpdump's capture format
- pcap-ng - "next-generation" successor to libpcap format
- Sun snoop and atmsnoop
- Shomiti/Finisar *Surveyor* captures
- Novell *LANalyzer* captures
- Microsoft Network Monitor captures
- AIX's iptrace captures
- Cinco Networks NetXray captures
- Network Associates Windows-based Sniffer and Sniffer Pro captures
- Network General/Network Associates DOS-based Sniffer (compressed or uncompressed) captures
- AG Group/WildPackets EtherPeek/TokenPeek/AiroPeek/EtherHelp/PacketGrabber captures
- RADCOM's WAN/LAN Analyzer captures
- Network Instruments Observer version 9 captures
- Lucent/Ascend router debug output
- HP-UX's nettl
- Toshiba's ISDN routers dump output
- ISDN4BSD *i4btrace* utility
- traces from the EyeSDN USB S0
- IPLog format from the Cisco Secure Intrusion Detection System
- pppd logs (pppdump format)



Import hex dump

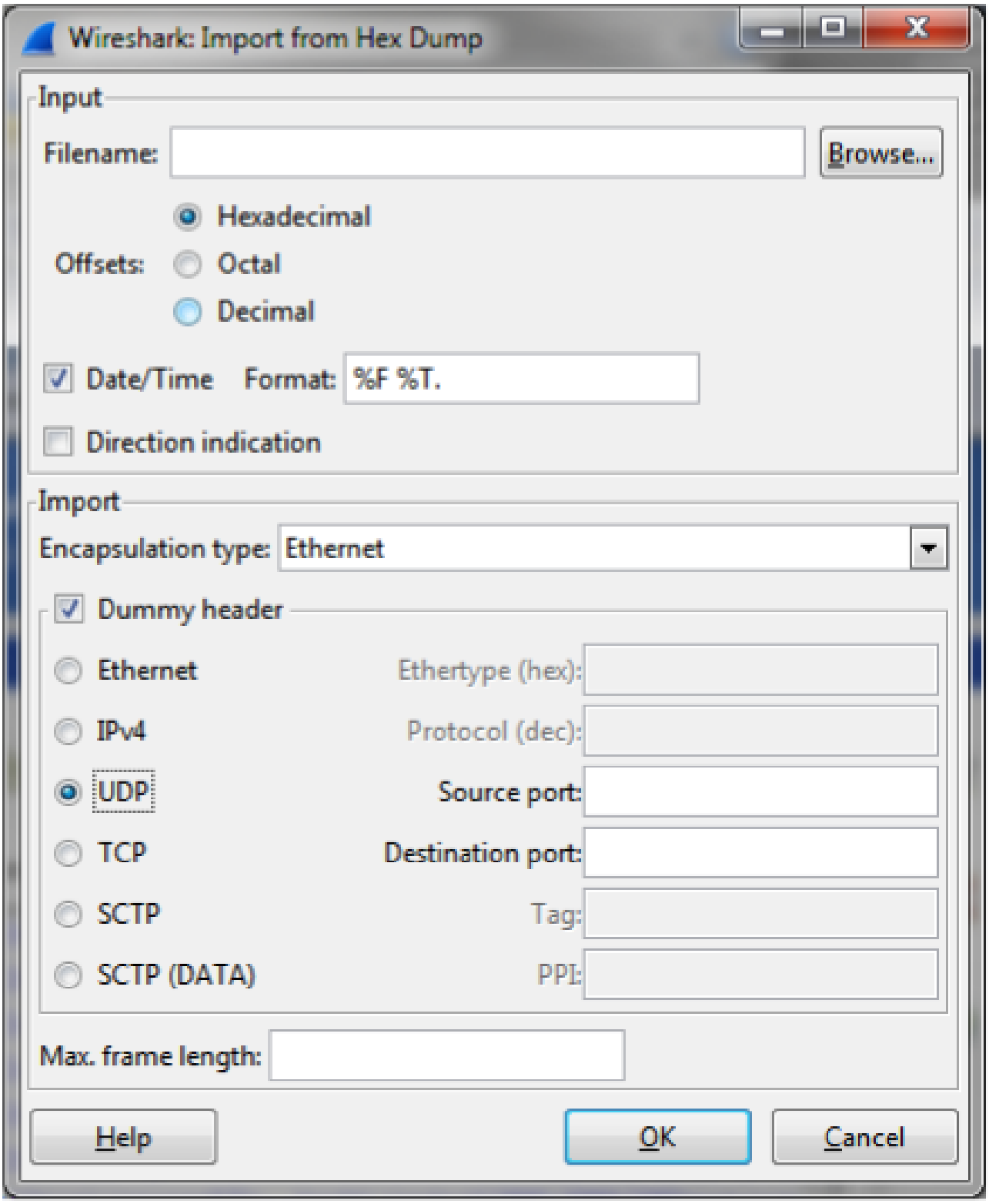
Wireshark can read in an ASCII hex dump and write the data described into a temporary libpcap capture file. It can read hex dumps with multiple packets in them, and build a capture file of multiple packets. It is also capable of generating dummy Ethernet, IP and UDP, TCP, or SCTP headers, in order to build fully processable packet dumps from hexdumps of application-level data only.

Wireshark understands a hexdump of the form generated by **od -Ax -tx1 -v**. In other words, each byte is individually displayed and surrounded with a space. Each line begins with an offset describing the position in the file. The offset is a hex number (can also be octal or decimal), of more than two hex digits. Here is a sample dump that can be imported:


```
000000 00 e0 1e a7 05 6f 00 10 ........
000008 5a a0 b9 12 08 00 46 00 ........
000010 03 68 00 00 00 00 0a 2e ........
000018 ee 33 0f 19 08 7f 0f 19 ........
000020 03 80 94 04 00 00 10 01 ........
000028 16 a2 0a 00 03 50 00 0c ........
000030 01 01 0f 19 03 80 11 01 ........
```

There is no limit on the width or number of bytes per line. Also the text dump at the end of the line is ignored. Bytes/hex numbers can be uppercase or lowercase. Any text before the offset is ignored, including email forwarding characters '>'. Any lines of text between the bytestring lines are ignored. The offsets are used to track the bytes, so offsets must be correct. Any line which has only bytes without a leading offset is ignored. An offset is recognized as being a hex number longer than two

Figure 5.10. The "Import from Hex Dump" dialog



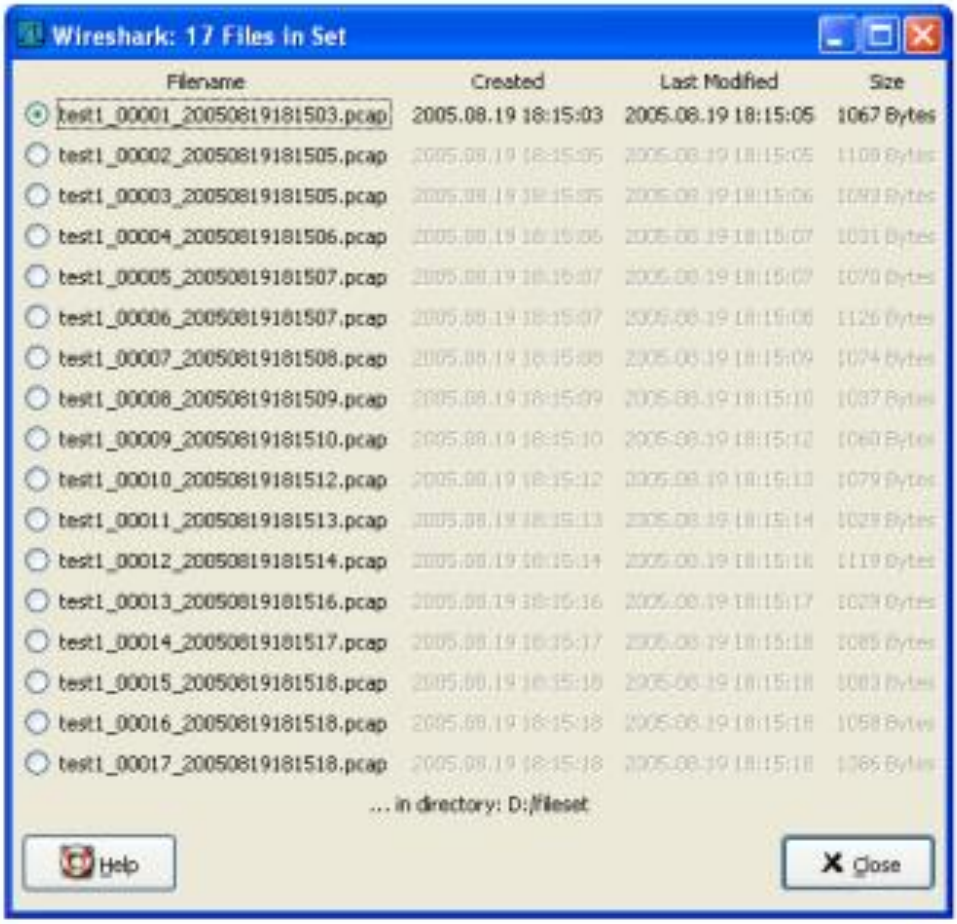
Specific controls of this import dialog are split in two sections:

Input Determine which input file has to be imported and how it is to be interpreted.

- **Previous File** closes the current and opens the previous file in the file set.

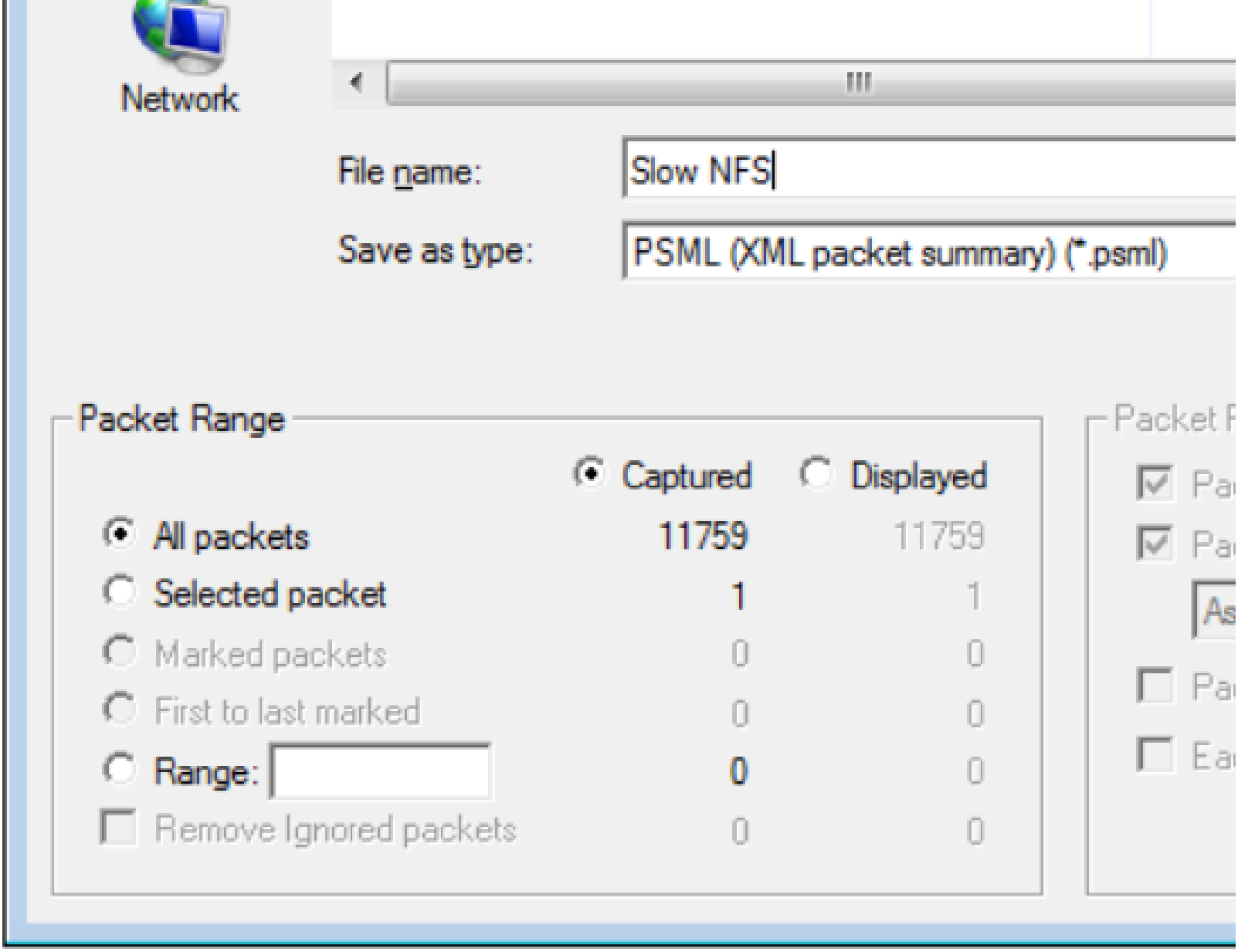
. The "List Files" dialog box

Figure 5.11. The "List Files" dialog box



Each line contains information about a file of the file set:

- **Filename** the name of the file. If you click on the filename (or the radio button left to it), the current file will be closed and the corresponding capture file will be opened.
- **Created** the creation time of the file
- **Last Modified** the last time the file was modified
- **Size** the size of the file



- **Export to file:** frame chooses the file to export the packet data to.
- The **Packet Range** frame is described in Section 5.9, "The Packet Range frame".

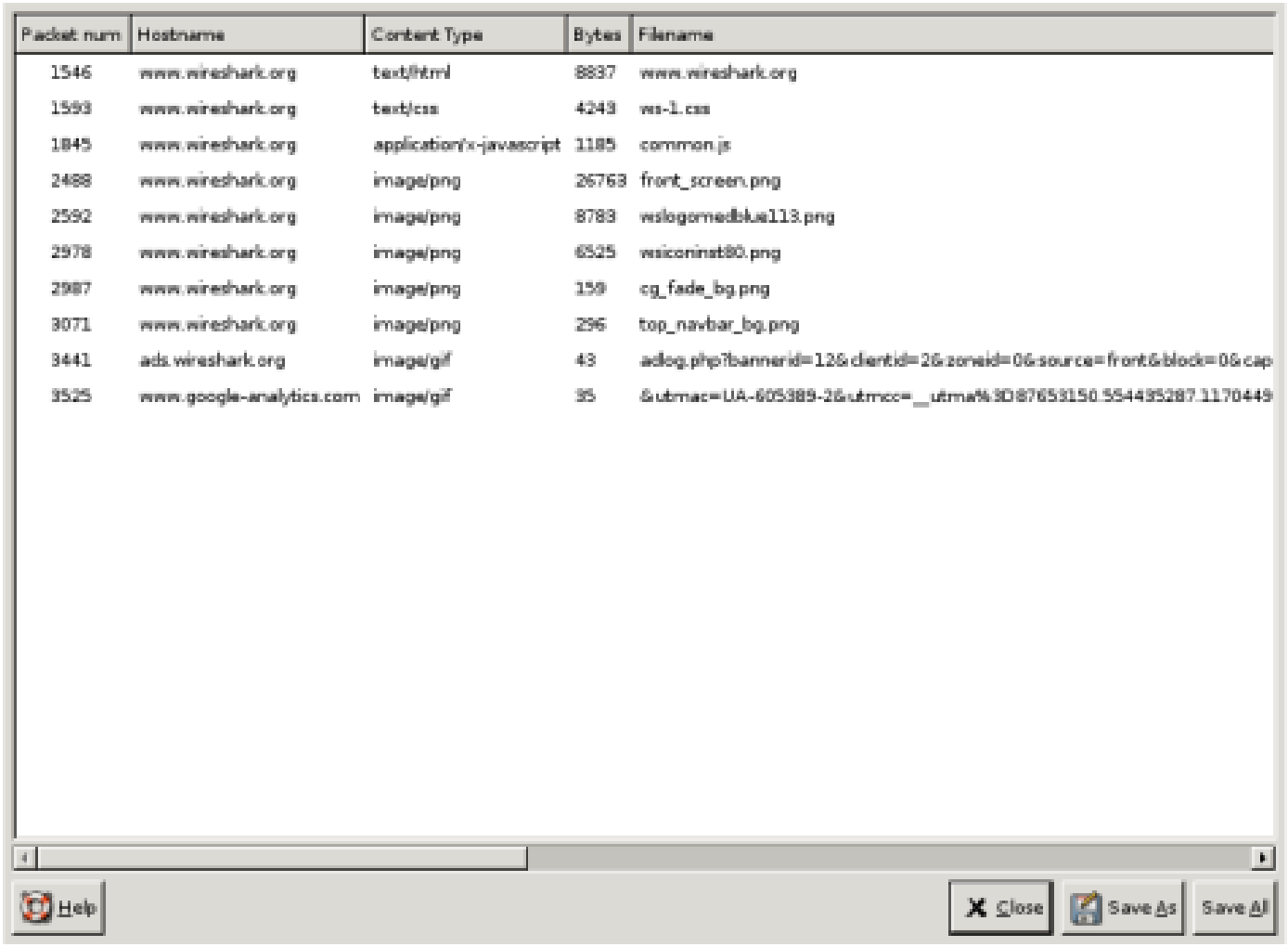
There's no such thing as a packet details frame for PSML export, as the packet format is defined by the PSML specification.

- **Browse for other folders** provides a flexible way to choose a folder.

. The "Export Objects" dialog box

This feature scans through HTTP streams in the currently open capture file or running capture and takes reassembled objects such as HTML documents, image files, executables and anything else that can be transferred over HTTP and lets you save them to disk. If you have a capture running, this list is automatically updated every few seconds with any new objects seen. The saved objects can then be opened with the proper viewer or executed in the case of executables (if it is for the same platform you are running Wireshark on) without any further work on your part. This feature is not available when using GTK2 versions below 2.4.

Figure 5.17. The "Export Objects" dialog box



Columns:

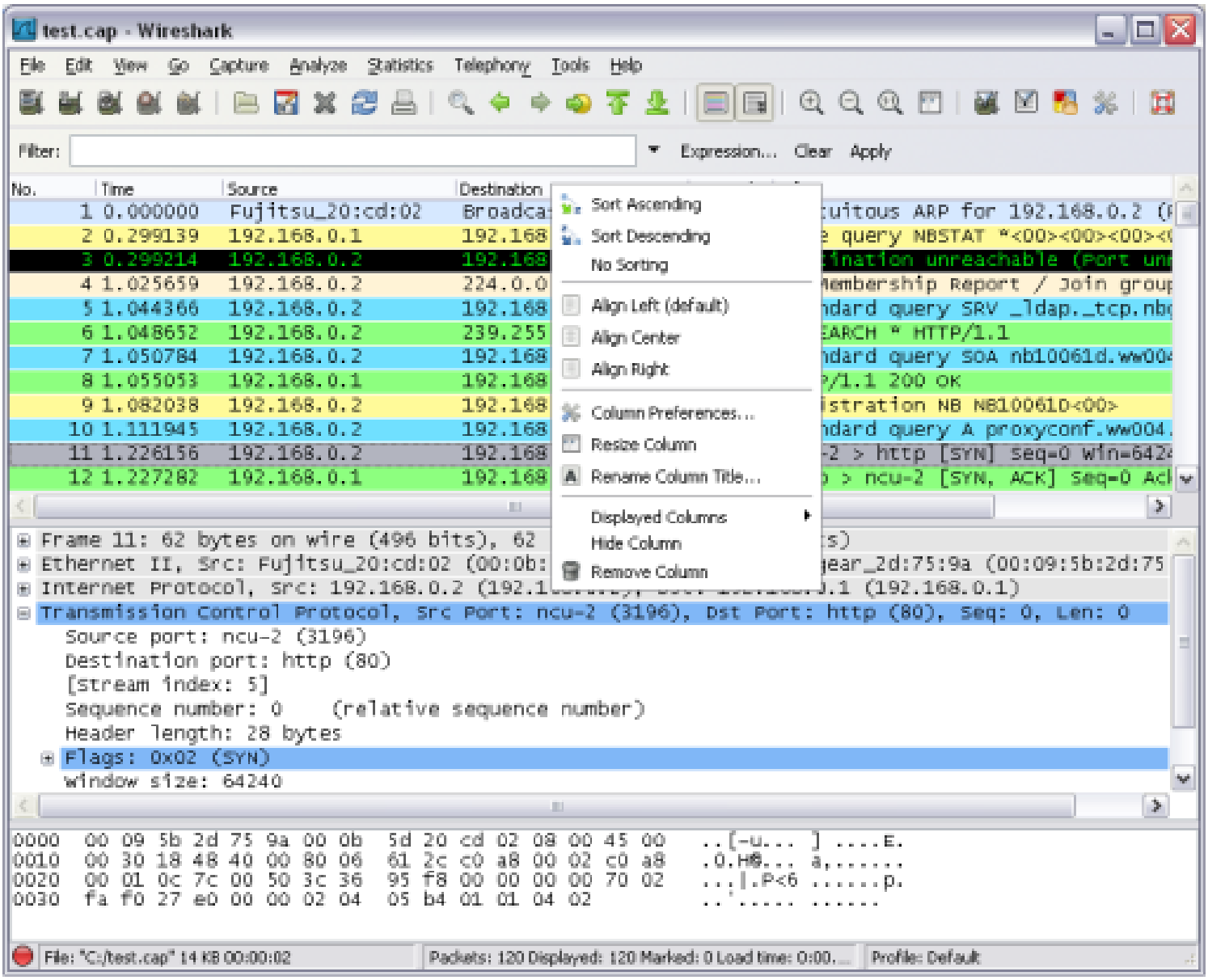
- **Packet num:** The packet number in which this object was found. In some cases, there can be multiple objects in the same packet.
- **Hostname:** The hostname of the server that sent the object as a response to an HTTP request.

Pop-up menus

You can bring up a pop-up menu over either the "Packet List", its column header, or "Packet Details" pane by clicking your right mouse button at the corresponding pane.

. Pop-up menu of the "Packet List" column header

Figure 6.3. Pop-up menu of the "Packet List" column header



The following table gives an overview of which functions are available in this header, where to find the corresponding function in the main menu, and a short description of each item.

Table 6.1. The menu items of the "Packet List" column header pop-up menu

Item	Identical to main menu's item:	Description
No Sort		Remove sorting order based on this column.

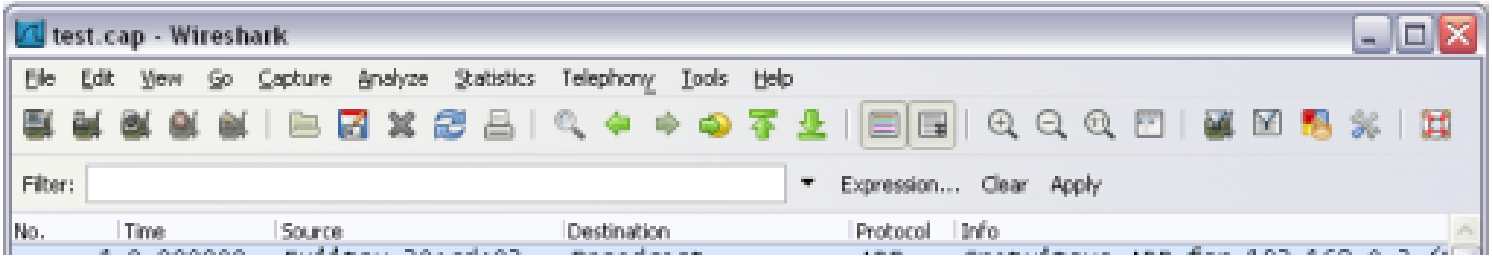
Align Left		Set left alignment of the values in this column.
Align Center		Set center alignment of the values in this column.
Align Right		Set right alignment of the values in this column.

Column Preferences...		Open the Preferences dialog box on the column tab.
Resize Column		Resize the column to fit the values.
Rename Column Title		Allows you to change the title of the column header.

Displayed Column	View	This menu items folds out with a list of all configured columns. These columns can now be shown or hidden in the packet list.
Hide Column		Allows you to hide the column from the packet list.
Remove Column		Allows you to remove the column from the packet list.

. Pop-up menu of the "Packet List" pane

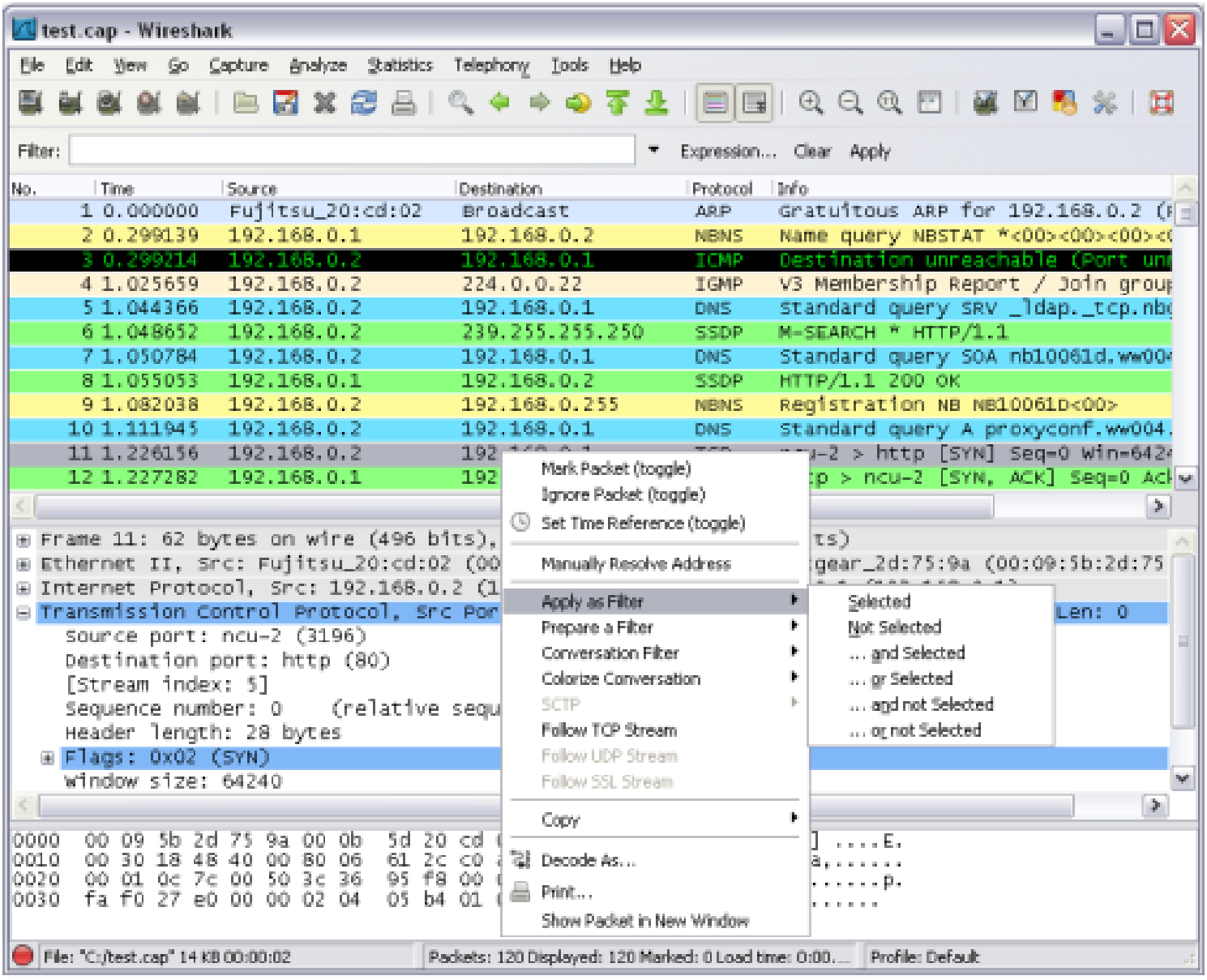
Figure 6.4. Pop-up menu of the "Packet List" pane



Remove Column		Allows you to remove the column from the packet list.

. Pop-up menu of the "Packet List" pane

Figure 6.4. Pop-up menu of the "Packet List" pane



The following table gives an overview of which functions are available in this pane, where to find the corresponding function in the main menu, and a short description of each item.

(toggle)		
Ignore Packet (toggle)	Edit	Ignore or inspect this packet while dissecting the capture file.
Set Time Reference (toggle)	Edit	Set/reset a time reference.
Manually Resolve Address		Allows you to enter a name to resolve for the selected address.

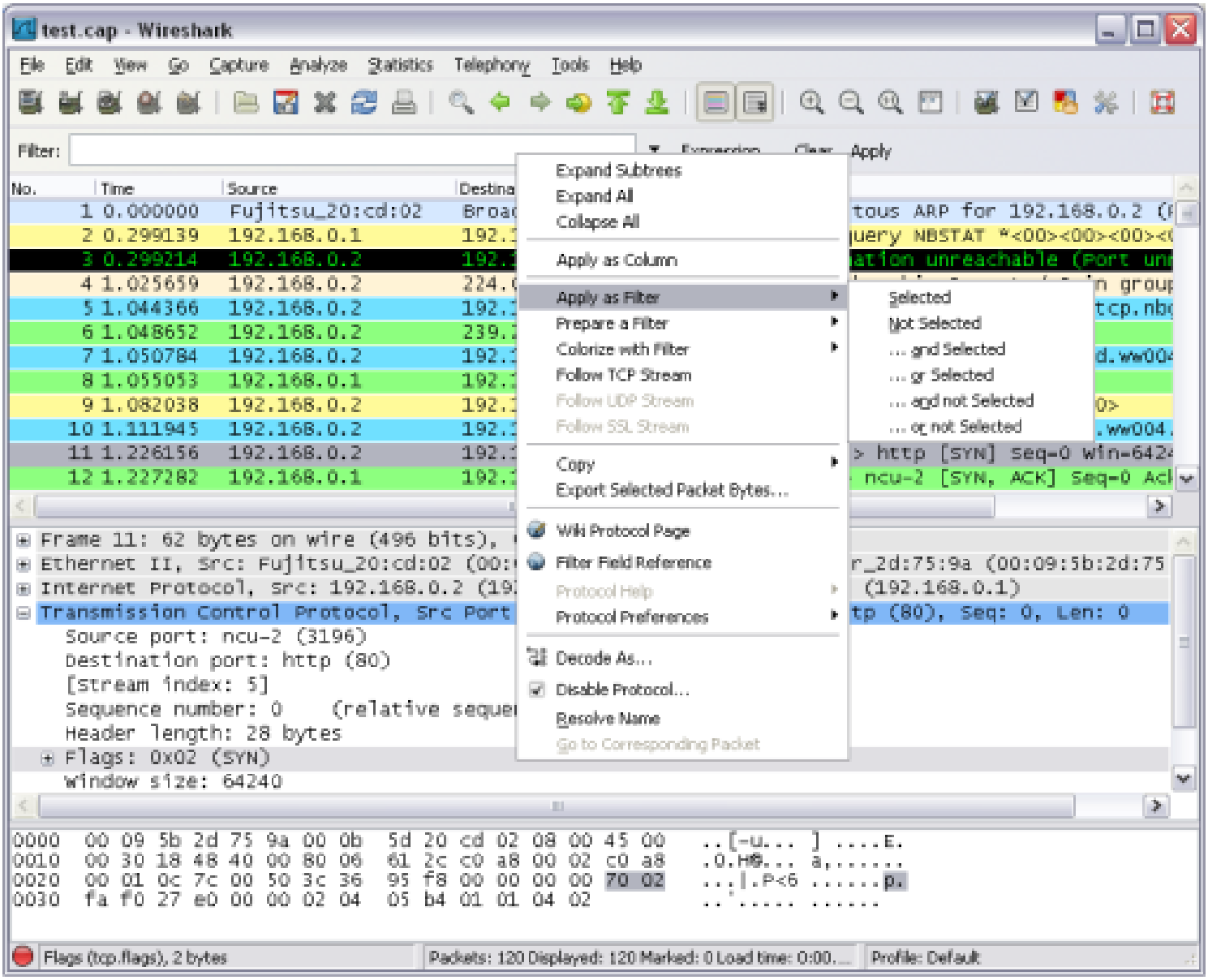
Apply as Filter	Analyze	Prepare and apply a display filter based on the currently selected item.
Prepare a Filter	Analyze	Prepare a display filter based on the currently selected item.
Conversation Filter	-	This menu item applies a display filter with the address information from the selected packet. E.g. the IP menu entry will set a filter to show the traffic between the two IP addresses of the current packet. XXX - add a new section describing this better.
Colorize Conversation	-	This menu item uses a display filter with the address information from the selected packet to build a new colorizing rule.
SCTP	-	Allows you to analyze and prepare a filter for this SCTP association.
Follow TCP Stream	Analyze	Allows you to view all the data on a TCP stream between a pair of nodes.
Follow UDP Stream	Analyze	Allows you to view all the data on a UDP datagram stream between a pair of nodes.
Follow SSL Stream	Analyze	Same as "Follow TCP Stream" but for SSL. XXX - add a new section describing this better.

Copy/ Summary (Text)	-	Copy the summary fields as displayed to the clipboard, as tab-separated text.
Copy/ Summary (CSV)	-	Copy the summary fields as displayed to the clipboard, as comma-separated text.
Copy/ As Filter		Prepare a display filter based on the currently selected item and copy that filter to the clipboard.
Copy/ Bytes		Copy the packet bytes to the clipboard in hexdump like

Print...	File	Print packets.
Show Packet in New Window	View	Display the selected packet in a new window.

. Pop-up menu of the "Packet Details" pane

Figure 6.5. Pop-up menu of the "Packet Details" pane



The following table gives an overview of which functions are available in this pane, where to find the corresponding function in the main menu, and a short description of each item.

Table 6.3. The menu items of the "Packet Details" pop-up menu

Item	Identical to main	Description

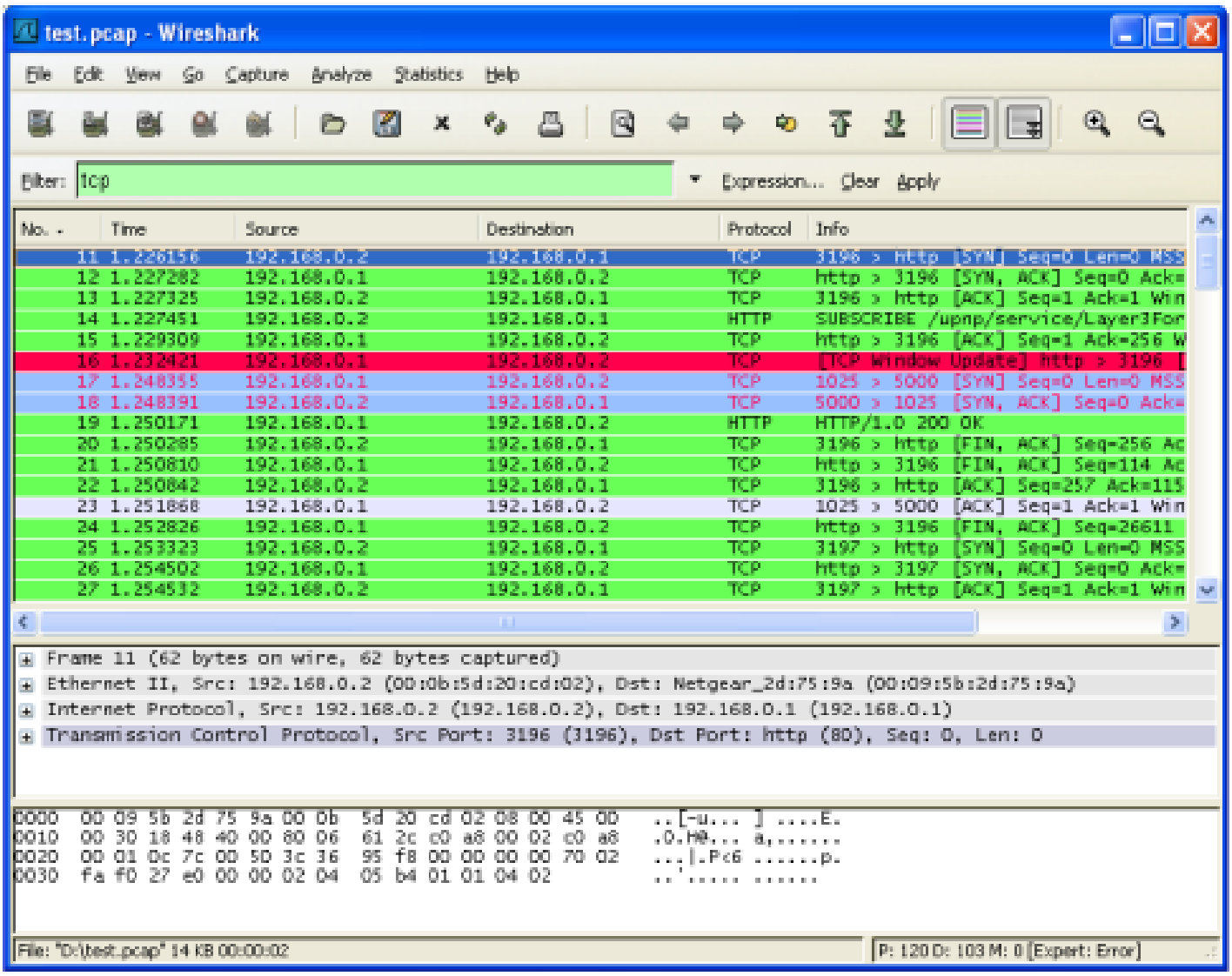
	menu's item:	
		are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.

Apply as Column		Use the selected protocol item to create a new column in the packet list.

Apply as Filter	Analyze	Prepare and apply a display filter based on the currently selected item.
Prepare a Filter	Analyze	Prepare a display filter based on the currently selected item.
Colorize with Filter	-	This menu item uses a display filter with the information from the selected protocol item to build a new colorizing rule.
Follow TCP Stream	Analyze	Allows you to view all the data on a TCP stream between a pair of nodes.
Follow UDP Stream	Analyze	Allows you to view all the data on a UDP datagram stream between a pair of nodes.
Follow SSL Stream	Analyze	Same as "Follow TCP Stream" but for SSL. XXX - add a new section describing this better.

Copy/ Description	Edit	Copy the displayed text of the selected field to the system clipboard.
Copy/ Fieldname	Edit	Copy the name of the selected field to the system clipboard.
Copy/ Value	Edit	Copy the value of the selected field to the system clipboard.
Copy/ As Filter	Edit	Prepare a display filter based on the currently selected item and copy it to the clipboard.
Copy/ Bytes (Offset Hex Text)	-	Copy the packet bytes to the clipboard in hexdump-like format; similar to the Packet List Pane command, but copies only the bytes relevant to the selected part of the tree (the bytes selected in the Packet Bytes Pane).
Copy/ Bytes (Offset Hex)	-	Copy the packet bytes to the clipboard in hexdump-like format, but without the text portion; similar to the Packet List Pane command, but copies only the bytes relevant to the selected part of the tree (the bytes selected in the Packet Bytes Pane).

Figure 6.6. Filtering on the TCP protocol



As you might have noticed, only packets of the TCP protocol are displayed now (e.g. packets 1-10 are hidden). The packet numbering will remain as before, so the first packet shown is now packet number 11.

Note!

When using a display filter, all packets remain in the capture file. The display filter only changes the display of the capture file but not its content!

You can filter on any protocol that Wireshark understands. You can also filter on any field that a dissector adds to the tree view, but only if the dissector has added an abbreviation for the field. A list of such fields is available in Wireshark in the **Add Expression...** dialog box. You can find more

They are shown in Table 6.4, "Display Filter comparison operators".

Tip!

You can use English and C-like terms in the same way, they can even be mixed in a filter string!

Table 6.4. Display Filter comparison operators

English	C-like	Description and example
eq	==	**Equal** `ip.src==10.0.0.5`
ne	!=	**Not equal** `ip.src!=10.0.0.5`
gt	>	**Greater than** `frame.len > 10`
lt	<	**Less than** `frame.len < 128`
ge	>=	**Greater than or equal to** `frame.len ge 0x100`
le	<=	**Less than or equal to** `frame.len <= 0x20`

In addition, all protocol fields are typed. Table 6.5, "Display Filter Field Types" provides a list of the types and example of how to express them.

Table 6.5. Display Filter Field Types

Type	Example

Unsigned integer (8-bit, 16-bit, 24-bit, 32-bit)	You can express integers in decimal, octal, or hexadecimal. The following display filters are equivalent:
	`ip.len le 1500`
	`ip.len le 02734`
	`ip.len le 0x436`

Type	Example
Boolean	A boolean field is present in the protocol decode only if its value is true. For example, **tcp.flags.syn** is present, and thus true, only if the SYN flag is present in a TCP segment header. Thus the filter expression **tcp.flags.syn** will select only those packets for which this flag exists, that is, TCP segments where the segment header contains the SYN flag. Similarly, to find source-routed token ring packets, use a filter expression of **tr.sr**.

Ethernet address (6 bytes)	Separators can be a colon (:), dot (.) or dash (-) and can have one or two bytes between separators:
	
	`eth.dst == ff:ff:ff:ff:ff:ff`
	`eth.dst == ff-ff-ff-ff-ff-ff`
	`eth.dst == ffff.ffff.ffff`

IPv4 address	ip.addr == 192.168.0.1 Classless InterDomain Routing (CIDR) notation can be used to test if an IPv4 address is in a certain subnet. For example, this display filter will find all packets in the 129.111 Class-B network: ip.addr == 129.111.0.0/16
IPv6 address	ipv6.addr == ::1
IPX address	ipx.addr == 00000000.ffffffffffff
String (text)	http.request.uri == "http://www.wireshark.org/"

. Combining expressions

You can combine filter expressions in Wireshark using the logical operators shown in Table 6.6, "Display Filter Logical Operations"

Table 6.6. Display Filter Logical Operations

IPv6 address	ipv6.addr == ::1
IPX address	ipx.addr == 00000000.ffffffffffff
String (text)	http.request.uri == "http://www.wireshark.org/"

. Combining expressions

You can combine filter expressions in Wireshark using the logical operators shown in Table 6.6, "Display Filter Logical Operations"

Table 6.6. Display Filter Logical Operations

English	C-like	Description and example
and	&&	**Logical AND** `ip.src==10.0.0.5 and tcp.flags.fin`
or	||	**Logical OR** `ip.scr==10.0.0.5 or ip.src==192.1.1.1`
xor	^^	**Logical XOR** `tr.dst[0:3] == 0.6.29 xor tr.src[0:3] == 0.6.29`
not	!	**Logical NOT** `not llc`
[...]		**Substring Operator** Wireshark allows you to select subsequences of a sequence in rather elaborate ways. After a label you can place a pair of brackets [] containing a comma separated list of range specifiers.

English	C-like	Description and example

		`eth.src[0:3] == 00:00:83`
		
		The example above uses the n:m format to specify a single range. In this case n is the beginning offset and m is the length of the range being specified.
		
		`eth.src[1-2] == 00:83`
		
		The example above uses the n-m format to specify a single range. In this case n is the beginning offset and m is the ending offset.
		
		`eth.src[:4] == 00:00:83:00`
		
		The example above uses the :m format, which takes everything from the beginning of a sequence to offset m. It is equivalent to 0:m
		
		`eth.src[4:] == 20:20`
		
		The example above uses the n: format, which takes everything from offset n to the end of the sequence.
		
		`eth.src[2] == 83`
		
		The example above uses the n format to specify a single range. In this case the element in the sequence at offset n is selected. This is equivalent to n:1.
		
		`eth.src[0:3,1-2,:4,4:,2] ==`
		`00:00:83:00:83:00:00:83:00:20:20:83`
		
		Wireshark allows you to string together single ranges in a comma separated list to form compound ranges as shown above.

. A common mistake

Warning!

Using the [illegible] operators on combined expressions like: eth.addr in addr.tcp.port udp.port

		list to form compound ranges as shown above.

. A common mistake

Warning!

Using the != operator on combined expressions like: eth.addr, ip.addr, tcp.port, udp.port and alike will probably not work as expected!

Often people use a filter string to display something like **ip.addr == 1.2.3.4** which will display all packets containing the IP address 1.2.3.4.

Then they use **ip.addr != 1.2.3.4** to see all packets not containing the IP address 1.2.3.4 in it. Unfortunately, this does **not** do the expected.

Instead, that expression will even be true for packets where either source or destination IP address equals 1.2.3.4. The reason for this, is that the expression **ip.addr != 1.2.3.4** must be read as "the packet contains a field named ip.addr with a value different from 1.2.3.4". As an IP datagram contains both a source and a destination address, the expression will evaluate to true whenever at least one of the two addresses differs from 1.2.3.4.

If you want to filter out all packets containing IP datagrams to or from IP address 1.2.3.4, then the correct filter is **!(ip.addr == 1.2.3.4)** as it reads "show me all the packets for which it is not true that a field named ip.addr exists with a value of 1.2.3.4", or in other words, "filter out all packets for which there are no occurrences of a field named ip.addr with the value 1.2.3.4".

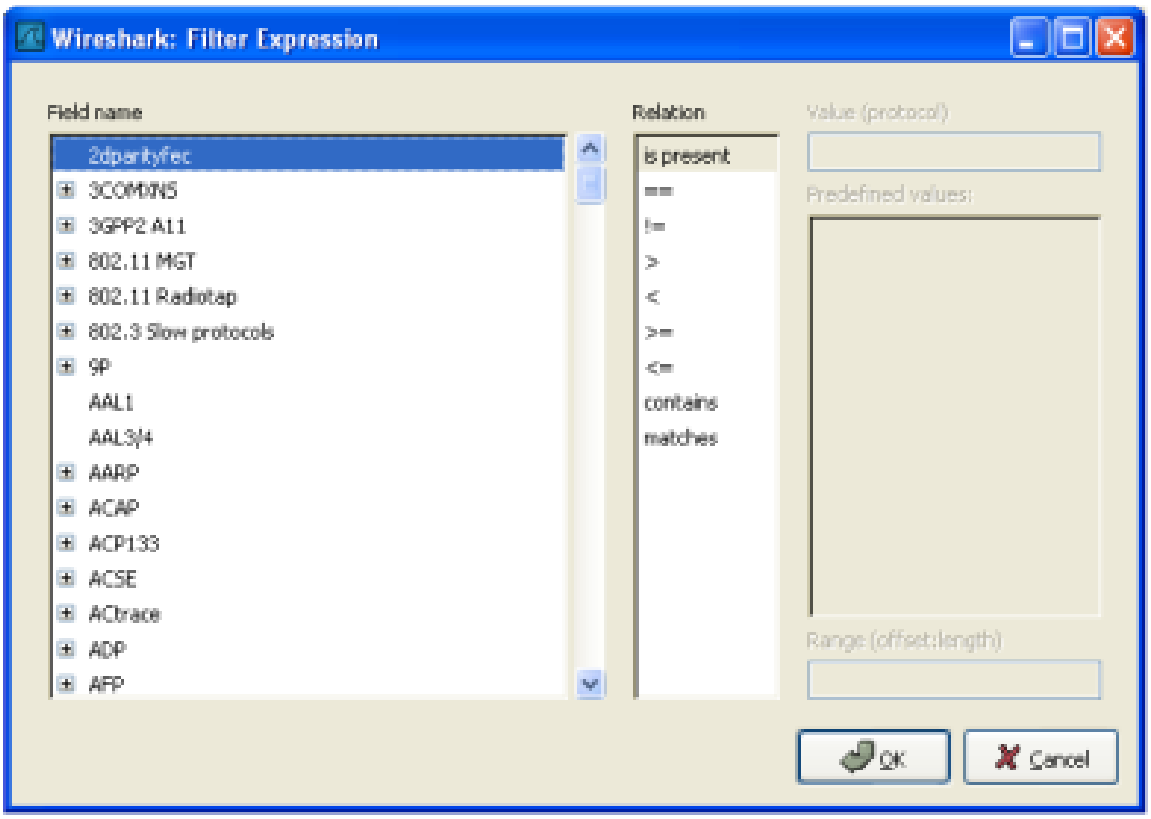
The "Filter Expression" dialog box

When you are accustomed to Wireshark's filtering system and know what labels you wish to use in your filters it can be very quick to simply type a filter string. However if you are new to Wireshark or are working with a slightly unfamiliar protocol it can be very confusing to try to figure out what to type. The Filter Expression dialog box helps with this.

Tip!

The "Filter Expression" dialog box is an excellent way to learn how to write Wireshark display filter strings.

Figure 6.7. The "Filter Expression" dialog box



When you first bring up the Filter Expression dialog box you are shown a tree list of field names, organized by protocol, and a box for selecting a relation.

Field Name Select a protocol field from the protocol field tree. Every protocol with filterable fields is listed at the top level. (You can search for a particular protocol entry by entering the first few letters of the protocol name). By clicking on the "+" next to

Defining and saving filters

You can define filters with Wireshark and give them labels for later use. This can save time in remembering and retyping some of the more complex filters you use.

To define a new filter or edit an existing one, select the **Capture Filters...** menu item from the Capture menu or the **Display Filters...** menu item from the Analyze menu. Wireshark will then pop up the Filters dialog as shown in Figure 6.8, “The "Capture Filters" and "Display Filters" dialog boxes".

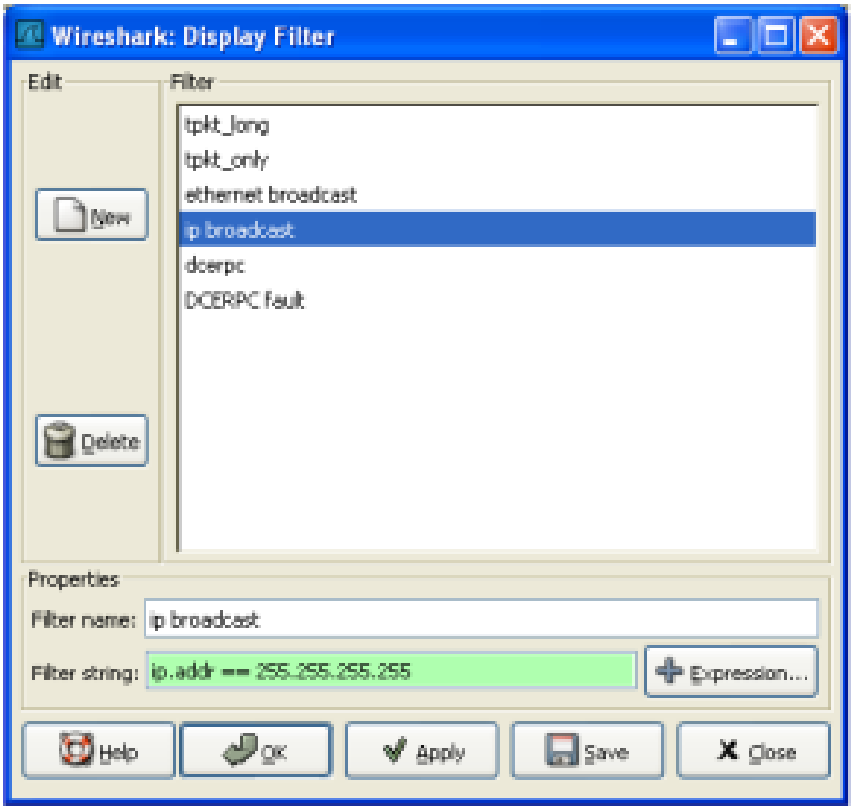
Note!

The mechanisms for defining and saving capture filters and display filters are almost identical. So both will be described here, differences between these two will be marked as such.

Warning!

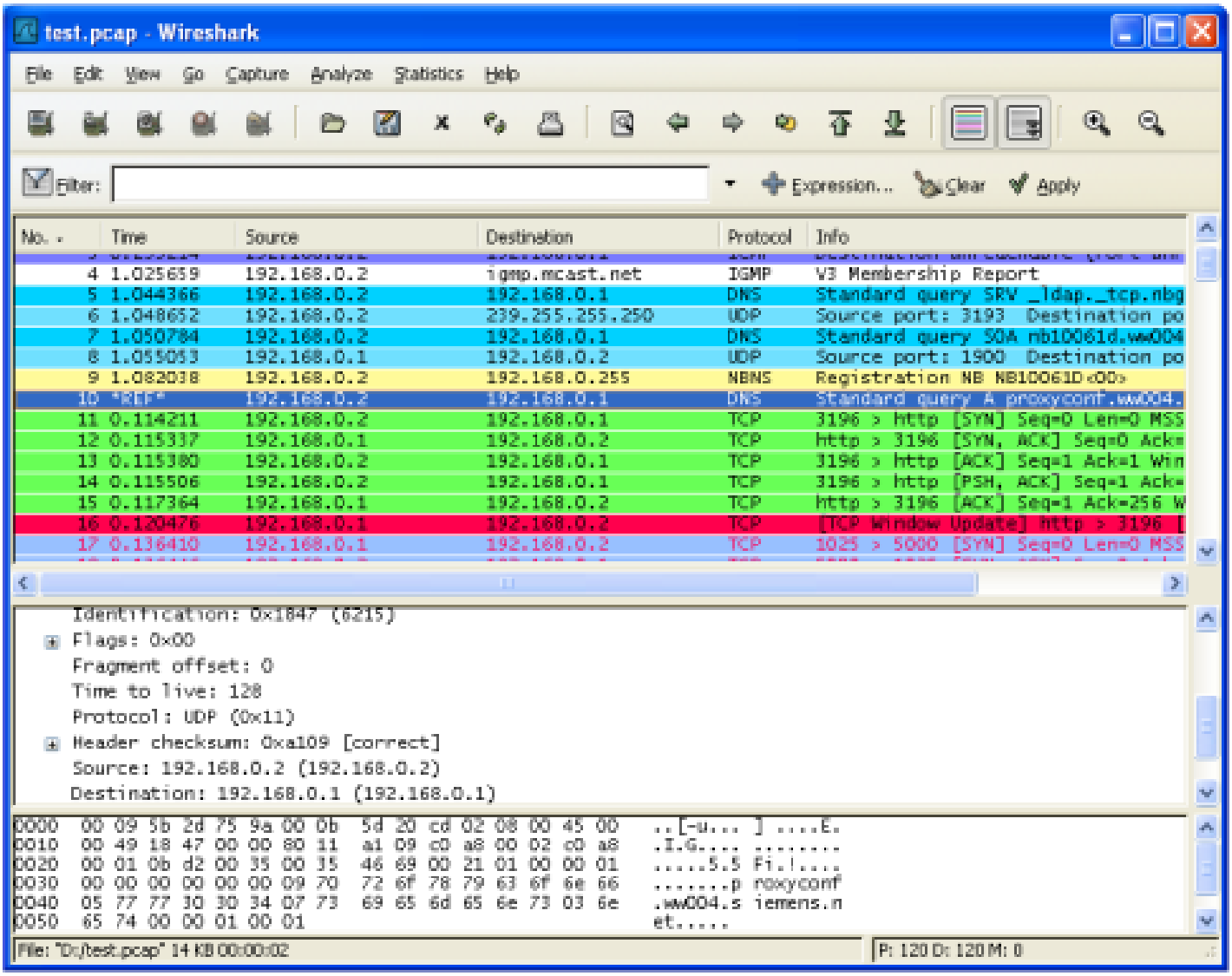
You must use **Save** to save your filters permanently. **Ok** or **Apply** will not save the filters, so they will be lost when you close Wireshark.

Figure 6.8. The "Capture Filters" and "Display Filters" dialog boxes



- **Set Time Reference (toggle)** Toggles the time reference state of the currently selected packet to on or off.
- **Find Next** Find the next time referenced packet in the "Packet List" pane.
- **Find Previous** Find the previous time referenced packet in the "Packet List" pane.

Figure 6.11. Wireshark showing a time referenced packet



A time referenced packet will be marked with the string *REF* in the Time column (see packet number 10). All subsequent packets will show the time since the last time reference.

In this chapter some of the advanced features of Wireshark will be described.

Following TCP streams

If you are working with TCP based protocols it can be very helpful to see the data from a TCP stream in the way that the application layer sees it. Perhaps you are looking for passwords in a Telnet stream, or you are trying to make sense of a data stream. Maybe you just need a display filter to show only the packets of that TCP stream. If so, Wireshark's ability to follow a TCP stream will be useful to you.

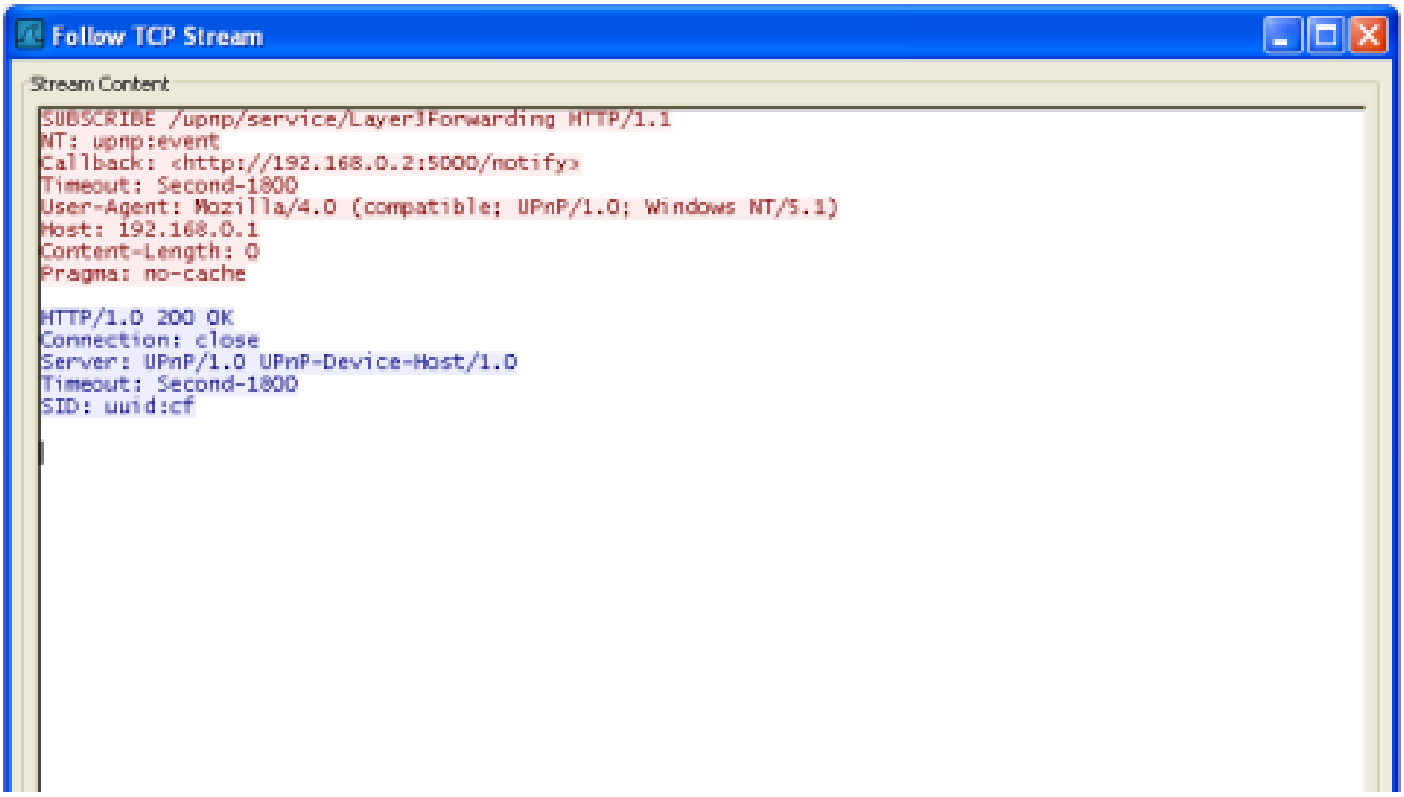
Simply select a TCP packet in the packet list of the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Wireshark Tools menu (or use the context menu in the packet list). Wireshark will set an appropriate display filter and pop up a dialog box with all the data from the TCP stream laid out in order, as shown in Figure 7.1, “The "Follow TCP Stream" dialog box”.

Note!

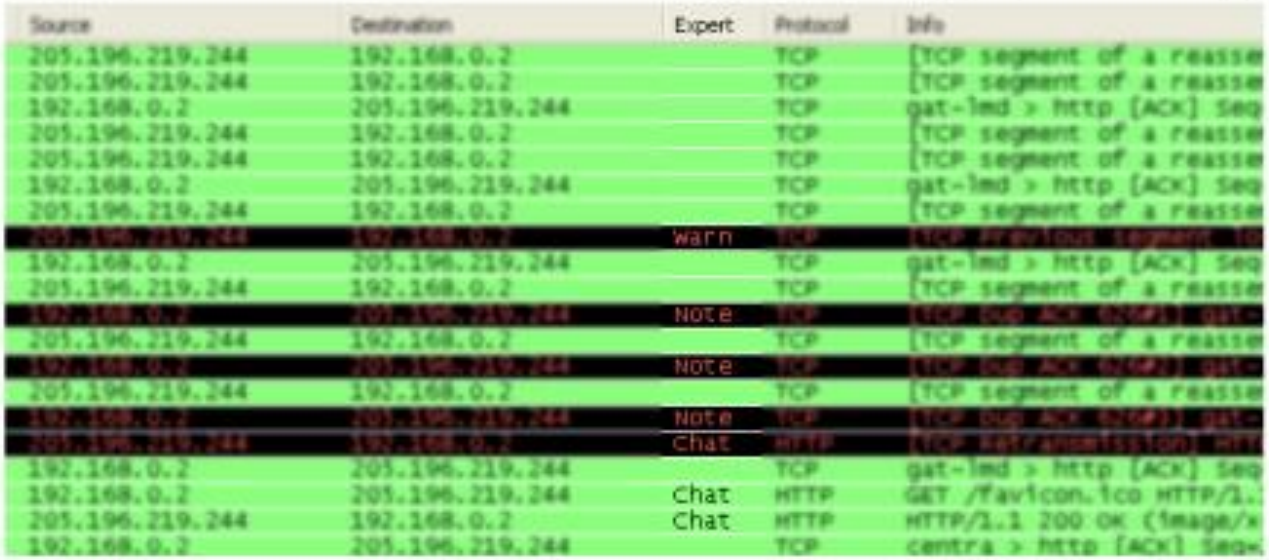
It is worthwhile noting that Follow TCP Stream installs a display filter to select all the packets in the TCP stream you have selected.

. The "Follow TCP Stream" dialog box

Figure 7.1. The "Follow TCP Stream" dialog box



. "Expert" Packet List Column (optional)



An optional "Expert Info Severity" packet list column is available (since SVN 22387 → 0.99.7), that displays the most significant severity of a packet, or stays empty if everything seems ok. This column is not displayed by default, but can be easily added using the Preferences Columns page described in Section 10.5, "Preferences".

Time Stamps

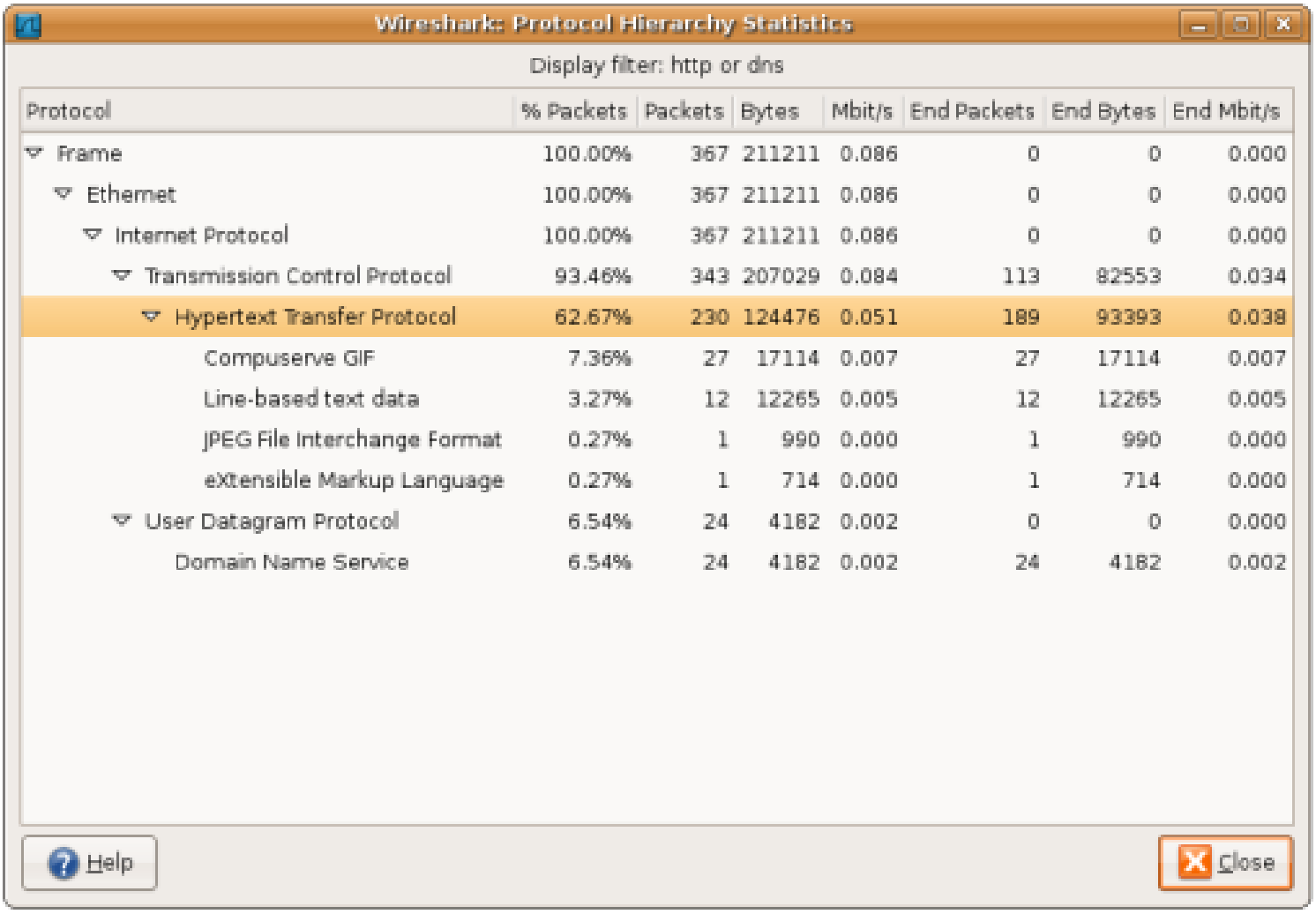
Time stamps, their precisions and all that can be quite confusing. This section will provide you with information about what's going on while Wireshark processes time stamps.

While packets are captured, each packet is time stamped as it comes in. These time stamps will be saved to the capture file, so they also will be available for (later) analysis.

So where do these time stamps come from? While capturing, Wireshark gets the time stamps from the libpcap (WinPcap) library, which in turn gets them from the operating system kernel. If the capture data is loaded from a capture file, Wireshark obviously gets the data from that file.

. Wireshark internals

Figure 8.2. The "Protocol Hierarchy" window



This is a tree of all the protocols in the capture. You can collapse or expand subtrees, by clicking on the plus / minus icons. By default, all trees are expanded.

Each row contains the statistical values of one protocol. The **Display filter** will show the current display filter.

The following columns containing the statistical values are available:

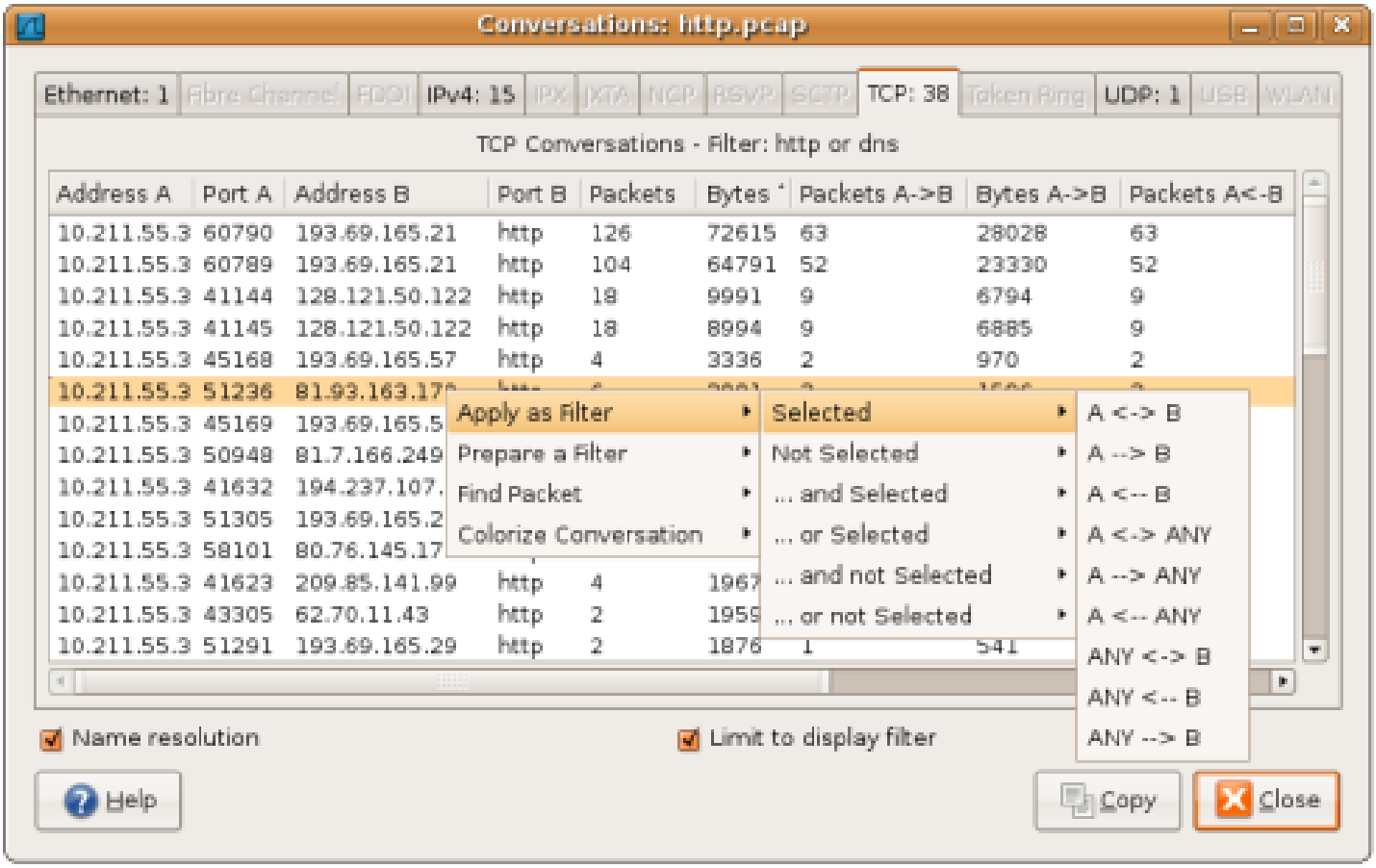
- **Protocol**: this protocol's name
- **% Packets**: the percentage of protocol packets, relative to all packets in the capture
- **Packets**: the absolute number of packets of this protocol
- **Bytes**: the absolute number of bytes of this protocol
- **MBit/s**: the bandwidth of this protocol relative to the capture time

in Section 8.5.1, "What is an Endpoint?".

. The "Conversations" window

The conversations window is similar to the endpoint Window; see Section 8.5.2, "The "Endpoints" window" for a description of their common features. Along with addresses, packet counters, and byte counters the conversation window adds four columns: the time in seconds between the start of the capture and the start of the conversation ("Rel Start"), the duration of the conversation in seconds, and the average bits (not bytes) per second in each direction.

Figure 8.3. The "Conversations" window



Each row in the list shows the statistical values for exactly one conversation.

Name resolution will be done if selected in the window and if it is active for the specific protocol layer (MAC layer for the selected Ethernet endpoints page).

Limit to display filter will only show conversations matching the current display filter.

The **copy** button will copy the list values to the clipboard in CSV (Comma Separated Values) format.

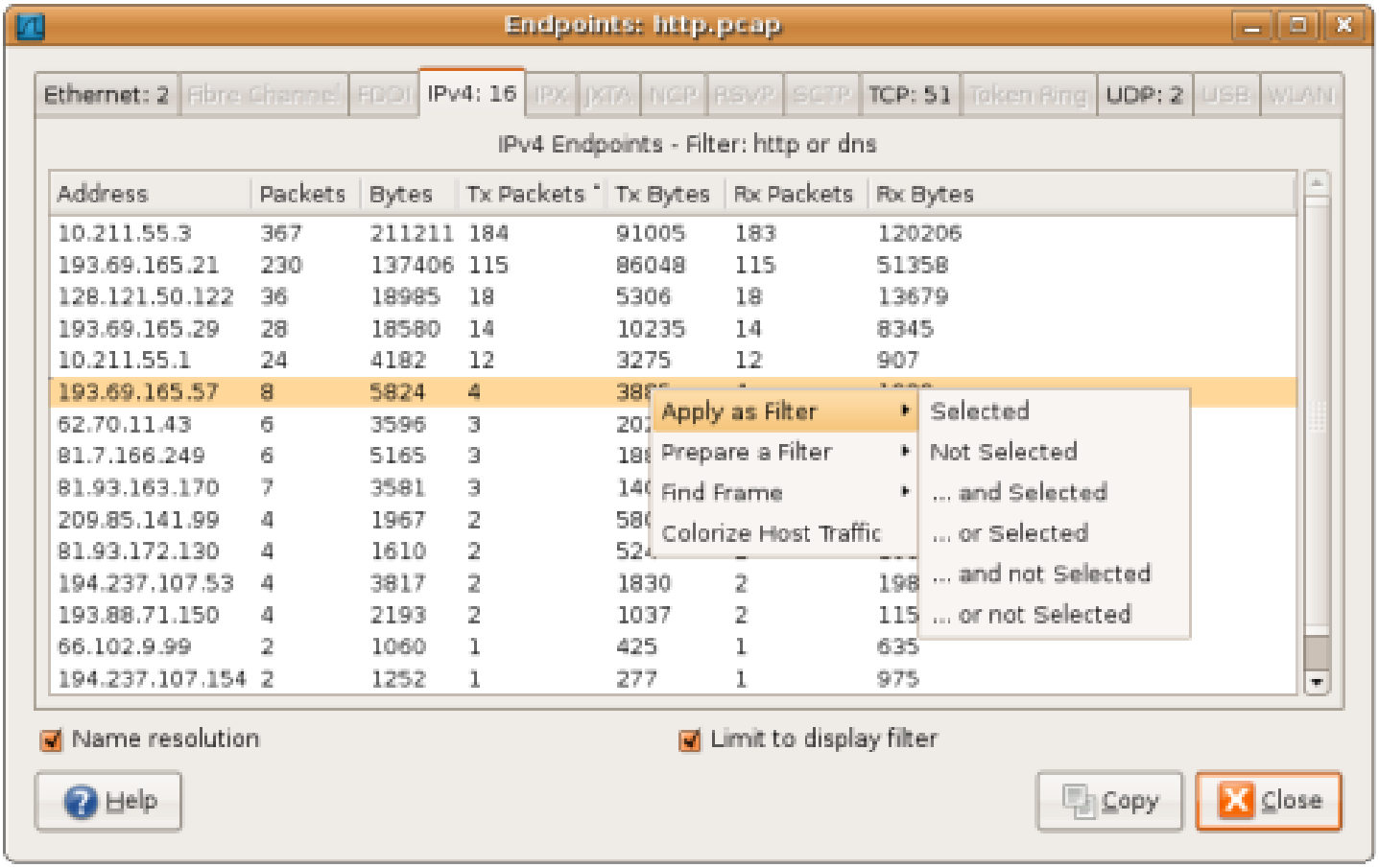
Broadcast / multicast endpoints

Broadcast / multicast traffic will be shown separately as additional endpoints. Of course, as these endpoints are virtual endpoints, the real traffic will be received by all (multicast: some) of the listed unicast endpoints.

. The "Endpoints" window

This window shows statistics about the endpoints captured.

Figure 8.4. The "Endpoints" window



For each supported protocol, a tab is shown in this window. Each tab label shows the number of endpoints captured (e.g. the tab label "Ethernet: 5" tells you that five ethernet endpoints have been captured). If no endpoints of a specific protocol were captured, the tab label will be greyed out (although the related page can still be selected).

Each row in the list shows the statistical values for exactly one endpoint.

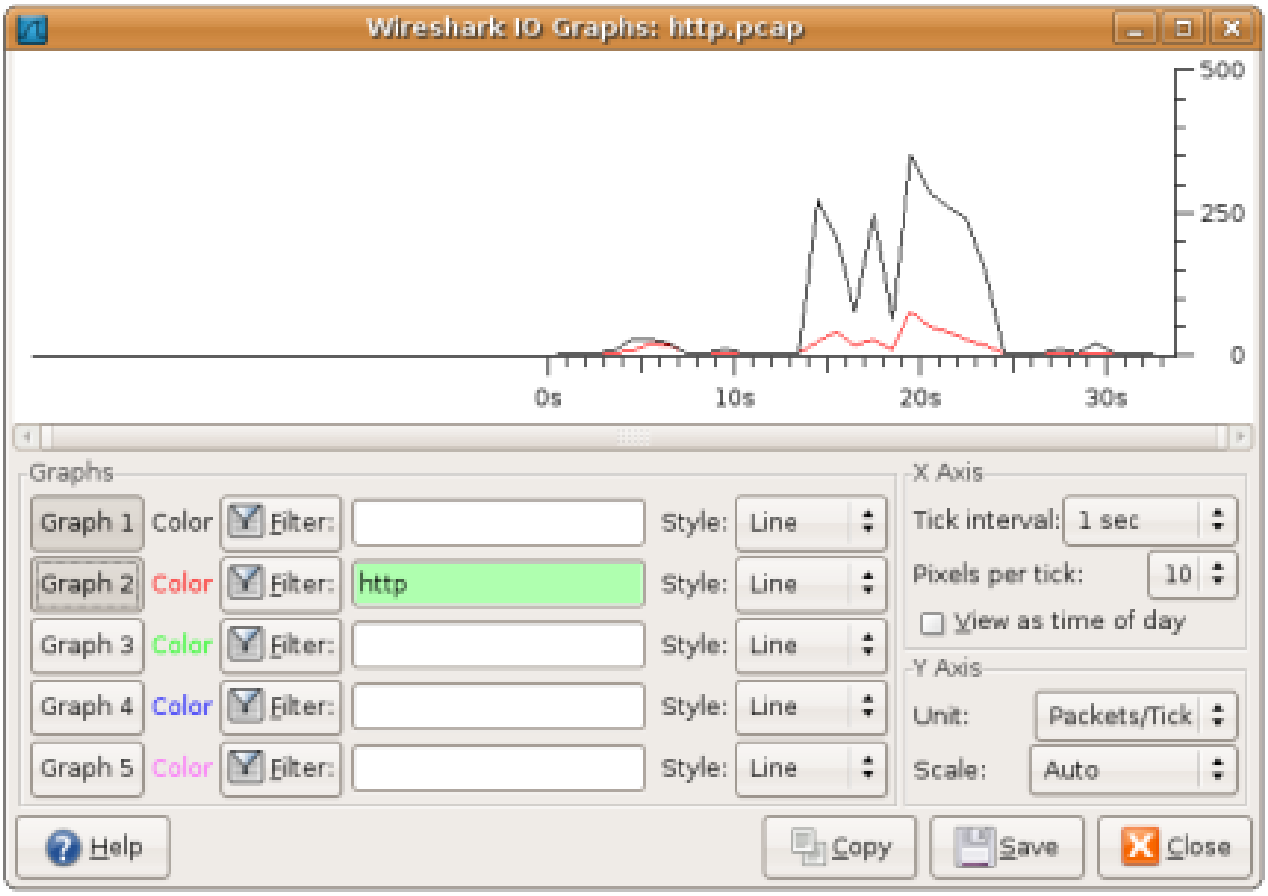
Name resolution will be done if selected in the window and if it is active for the specific protocol layer (MAC layer for the selected Ethernet endpoints page). As you might have noticed, the first row has a name resolution of the first three bytes "Netgear", the second row's address was resolved to

The "IO Graphs" window

User configurable graph of the captured network packets.

You can define up to five differently colored graphs.

Figure 8.5. The "IO Graphs" window



The user can configure the following things:

- **Graphs**
  - **Graph 1-5**: enable the specific graph 1-5 (only graph 1 is enabled by default)
  - **Color**: the color of the graph (cannot be changed)
  - **Filter**: a display filter for this graph (only the packets that pass this filter will be taken into account for this graph)
  - **Style**: the style of the graph (Line/Impulse/FBar/Dot)

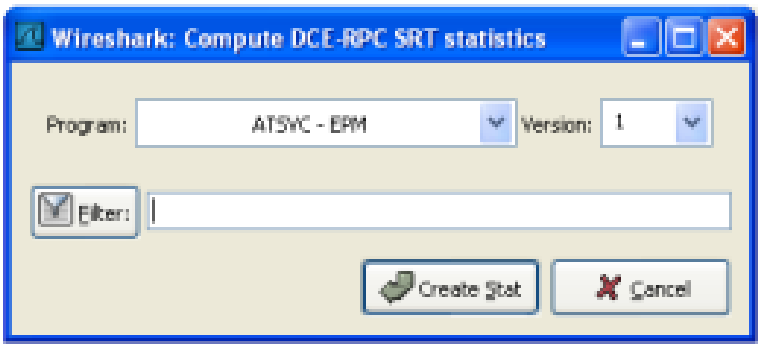
The other Service Response Time windows will work the same way (or only slightly different) compared to the following description.

. The "Service Response Time DCE-RPC" window

The service response time of DCE-RPC is the time between the request and the corresponding response.

First of all, you have to select the DCE-RPC interface:

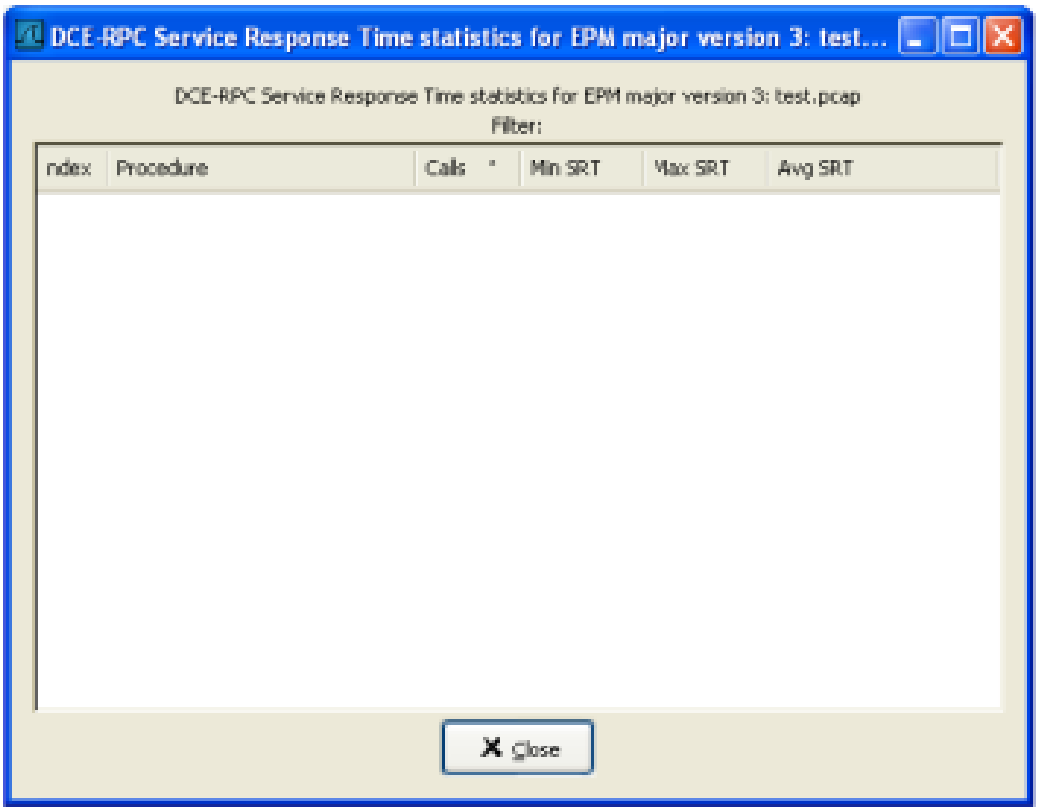
Figure 8.6. The "Compute DCE-RPC statistics" window



You can optionally set a display filter, to reduce the amount of packets.

Figure 8.7. The "DCE-RPC Statistic for ..." window

Figure 8.7. The "DCE-RPC Statistic for ..." window



Each row corresponds to a method of the interface selected (so the EPM interface in version 3 has 7 methods). For each method the number of calls, and the statistics of the SRT time is calculated.

Compare two capture files

Compare two capture files.

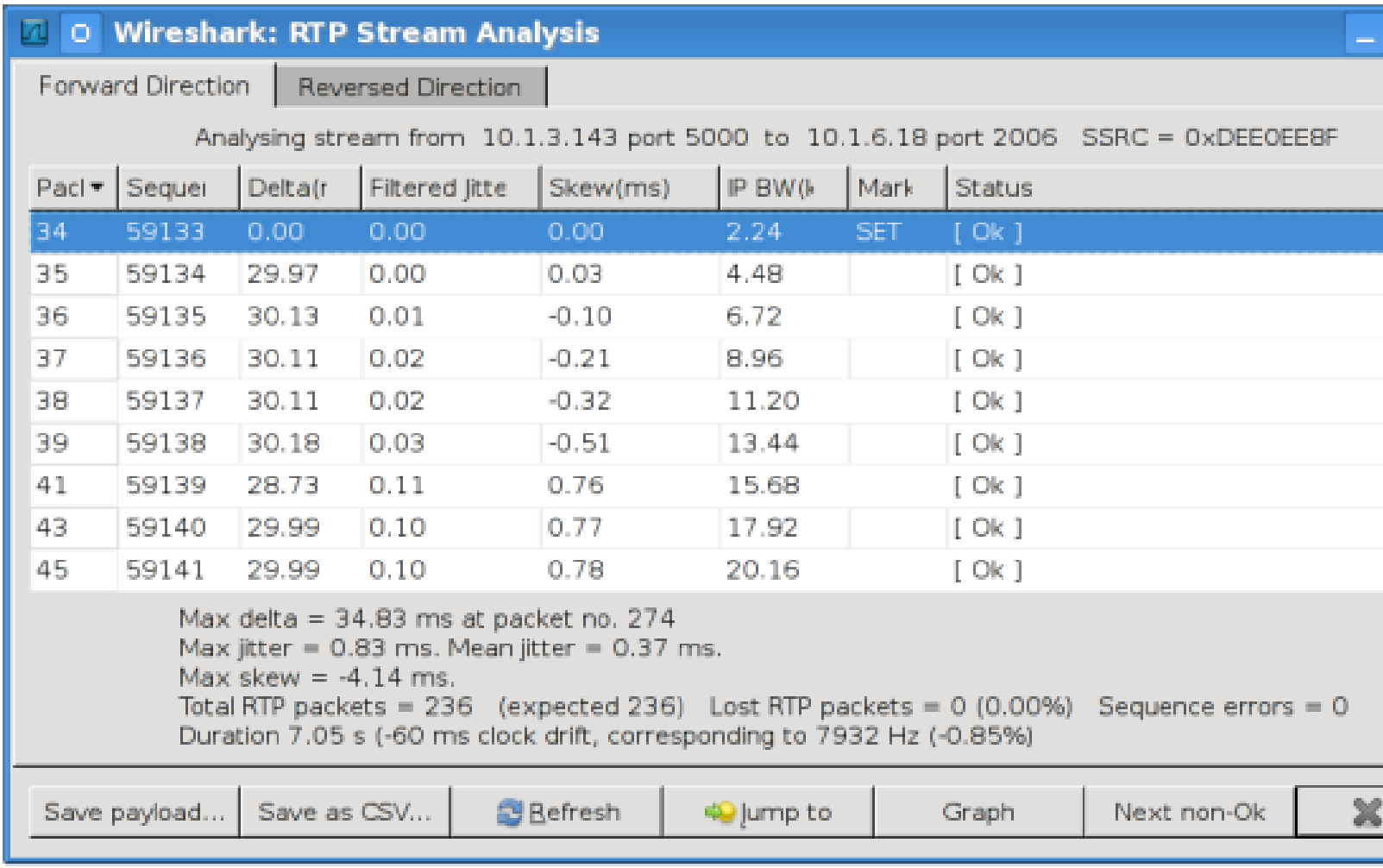
This feature works best when you have merged two capture files chronologically, one from each side of a client/server connection.

The merged capture data is checked for missing packets. If a matching connection is found it is checked for:

RTP Analysis

The RTP analysis function takes the selected RTP stream (and the reverse stream, if possible) and generates a list of statistics on it.

Figure 9.1. The "RTP Stream Analysis" window



Starting with basic data as packet number and sequence number, further statistics are created based on arrival time, delay, jitter, packet size, etc.

Besides the per packet statistics, the lower pane shows the overall statistics, with minimums and maximums for delta, jitter and clock skew. Also an indication of lost packets is included.

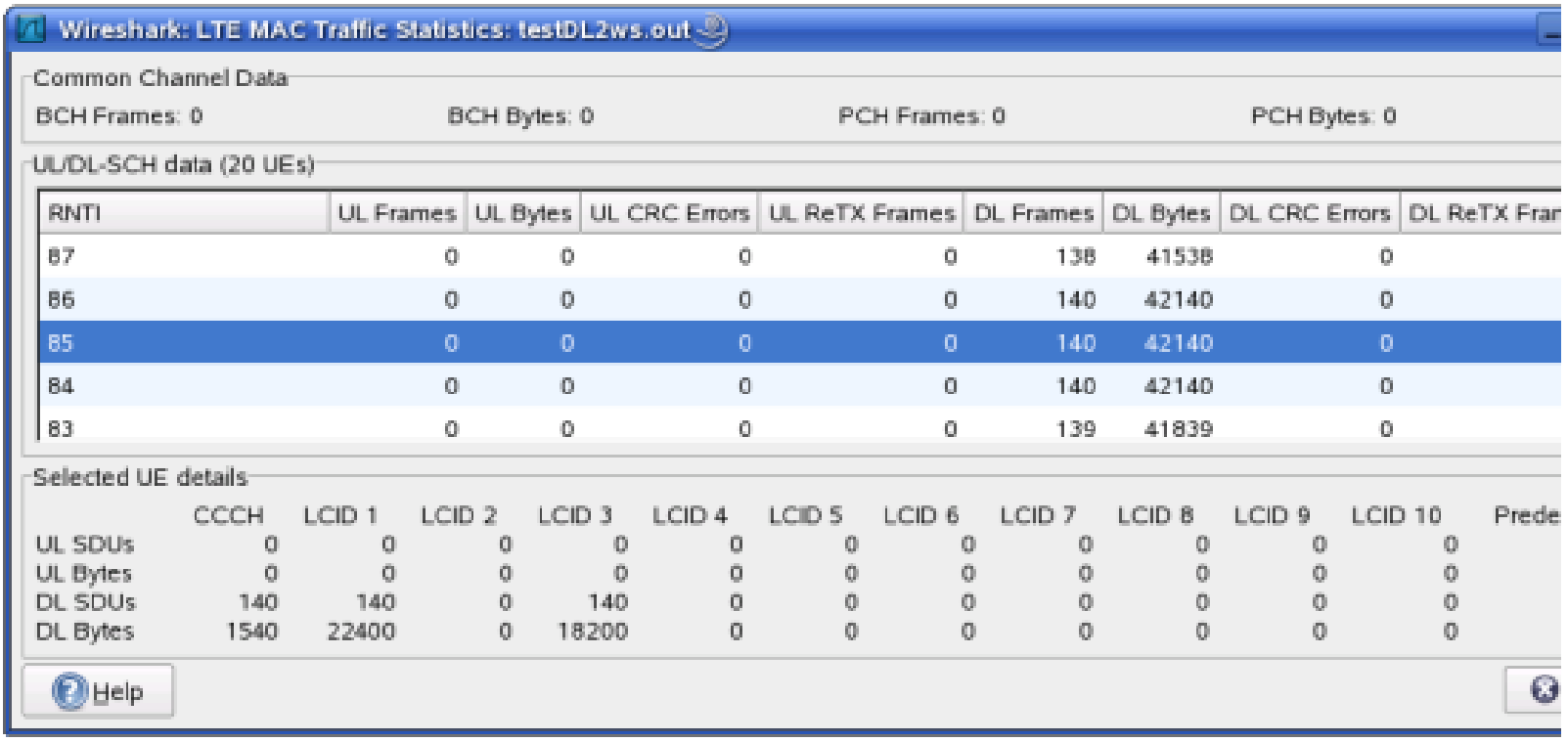
The RTP Stream Analysis window further provides the option to save the RTP payload (as raw data or, if in a PCM encoding, in an Audio file). Other options a to export and plot various statistics on the RTP streams.

VoIP Calls

LTE MAC Traffic Statistics

Statistics of the captured LTE MAC traffic. This window will summarize the LTE MAC traffic found in the capture.

Figure 9.2. The "LTE MAC Traffic Statistics" window



The top pane shows statistics for common channels. Each row in the middle pane shows statistical highlights for exactly one UE/C-RNTI. In the lower pane, you can see the for the currently selected UE/C-RNTI the traffic broken down by individual channel.

LTE RLC Traffic Statistics

Statistics of the captured LTE RLC traffic. This window will summarize the LTE RLC traffic found in the capture.

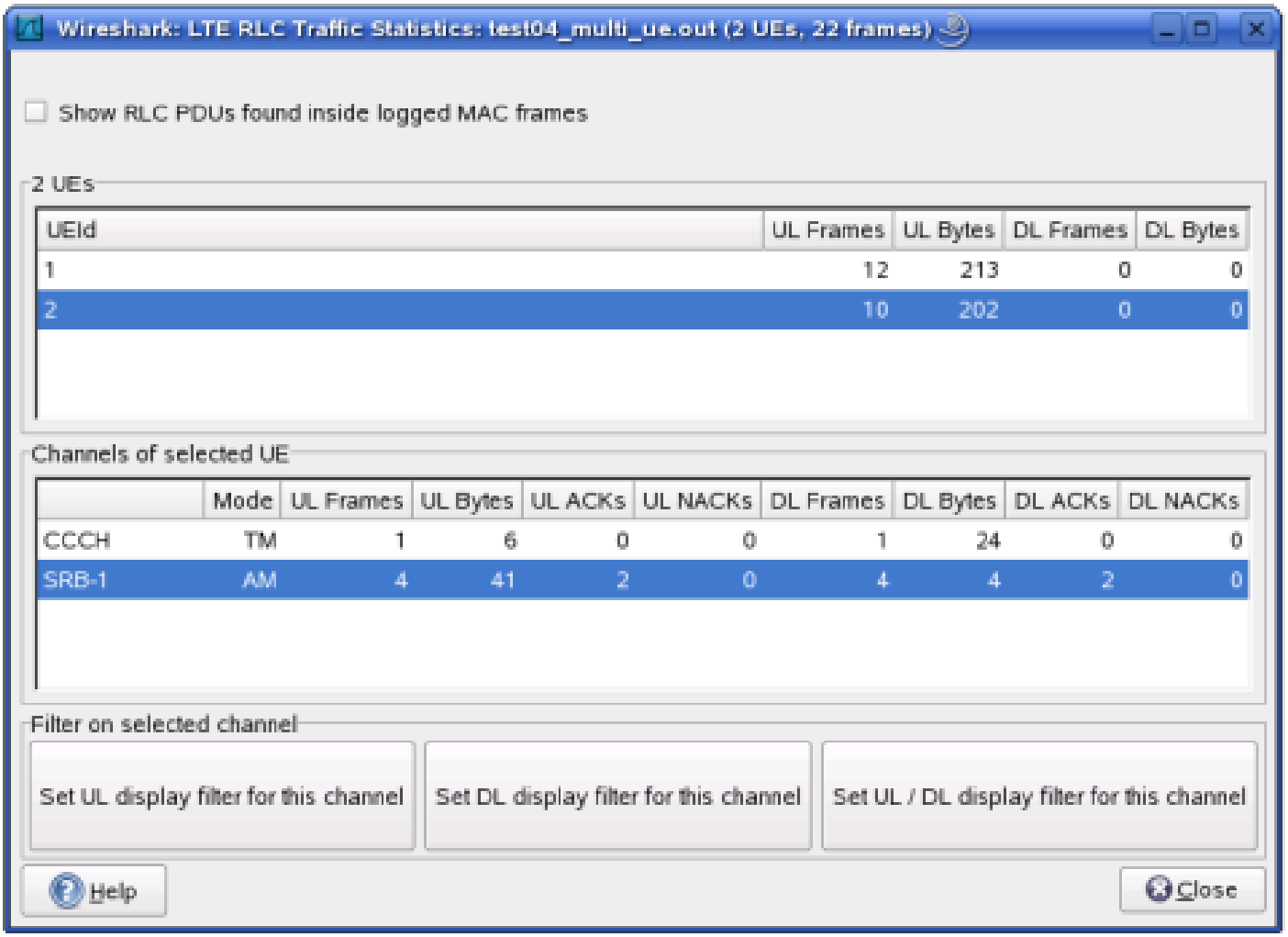
Figure 9.3. The "LTE RLC Traffic Statistics" window

The top pane shows statistics for common channels. Each row in the middle pane shows statistical highlights for exactly one UE/C-RNTI. In the lower pane, you can see the for the currently selected UE/C-RNTI the traffic broken down by individual channel.

LTE RLC Traffic Statistics

Statistics of the captured LTE RLC traffic. This window will summarize the LTE RLC traffic found in the capture.

Figure 9.3. The "LTE RLC Traffic Statistics" window



- How to control protocol dissection
- How to use the various preference settings

. Start Wireshark from the command line

You can start Wireshark from the command line, but it can also be started from most Window managers as well. In this section we will look at starting it from the command line.

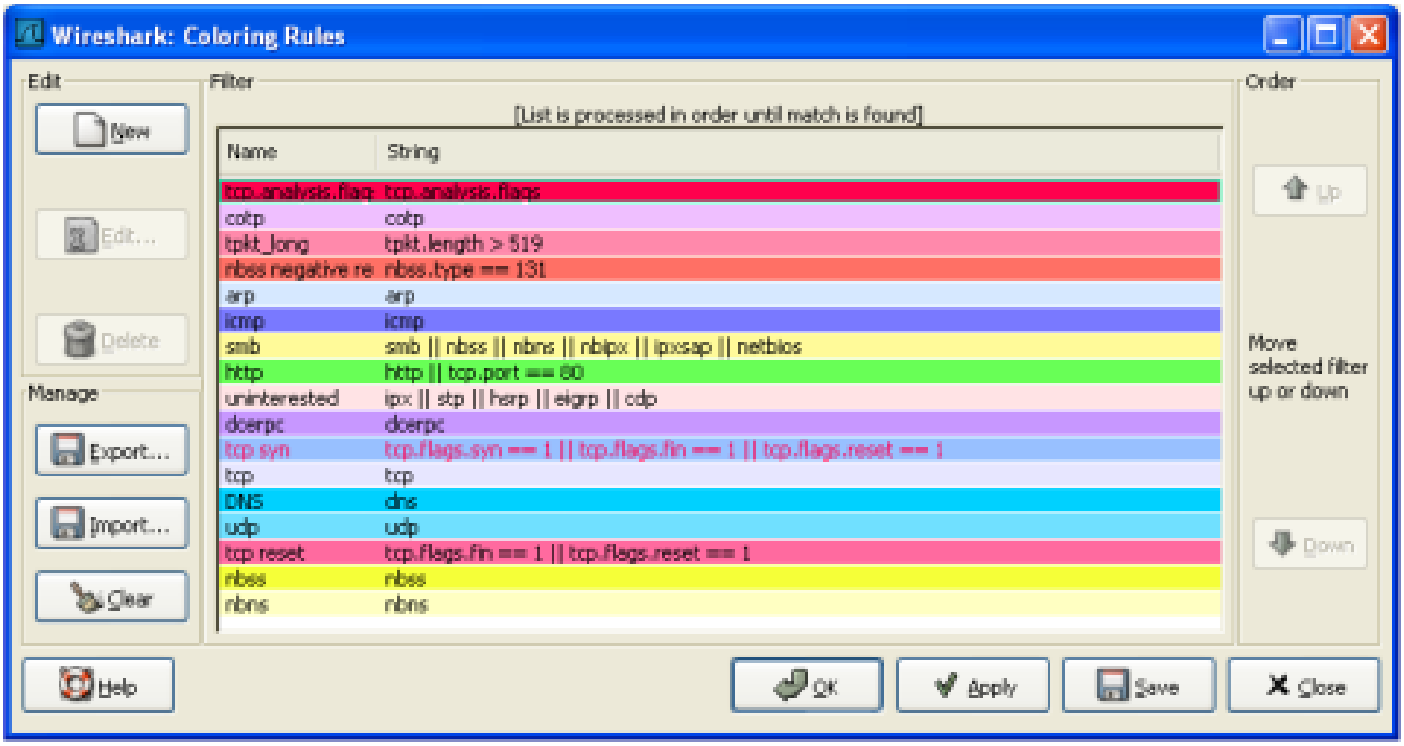
Wireshark supports a large number of command line parameters. To see what they are, simply enter the command **wireshark -h** and the help information shown in Example 10.1, "Help information available from Wireshark" (or something similar) should be printed.

Example 10.1. Help information available from Wireshark

```
Wireshark 1.9.0 (SVN Rev 47047 from /trunk)
Interactively dump and analyze network traffic.
See http://www.wireshark.org for more information.

Copyright 1998-2013 Gerald Combs <gerald@wireshark.org> and contributors.
This is free software; see the source for copying conditions. There is NO
warranty; not even for MERCHANTABILITY or FITNESS FOR A PARTICULAR PURPOSE.

Usage: wireshark [options] ... [ <infile> ]

Capture interface:
  -i <interface>           name or idx of interface (def: first non-loopback)
  -f <capture filter>      packet filter in libpcap filter syntax
  -s <snaplen>             packet snapshot length (def: 65535)
  -p                       don't capture in promiscuous mode
  -k                       start capturing immediately (def: do nothing)
  -S                       update packet display when new packets are captured
  -l                       turn on automatic scrolling while -S is in use
  -I                       capture in monitor mode, if available
  -B <buffer size>         size of kernel buffer (def: 1MB)
  -y <link type>           link layer type (def: first appropriate)
  -D                       print list of interfaces and exit
  -L                       print list of link-layer types of iface and exit

Capture stop conditions:
  -c <packet count>        stop after n packets (def: infinite)
  -a <autostop cond.> ...  duration:NUM - stop after NUM seconds
                           filesize:NUM - stop this file after NUM KB
                              files:NUM - stop after NUM files
Capture output:
  -b <ringbuffer opt.> ... duration:NUM - switch to next file after NUM secs
                           filesize:NUM - switch to next file after NUM KB
                              files:NUM - ringbuffer: replace after NUM files
```

To permanently colorize packets, select the **Coloring Rules...** menu item from the **View** menu; Wireshark will pop up the "Coloring Rules" dialog box as shown in Figure 10.1, "The "Coloring Rules" dialog box".

Figure 10.1. The "Coloring Rules" dialog box



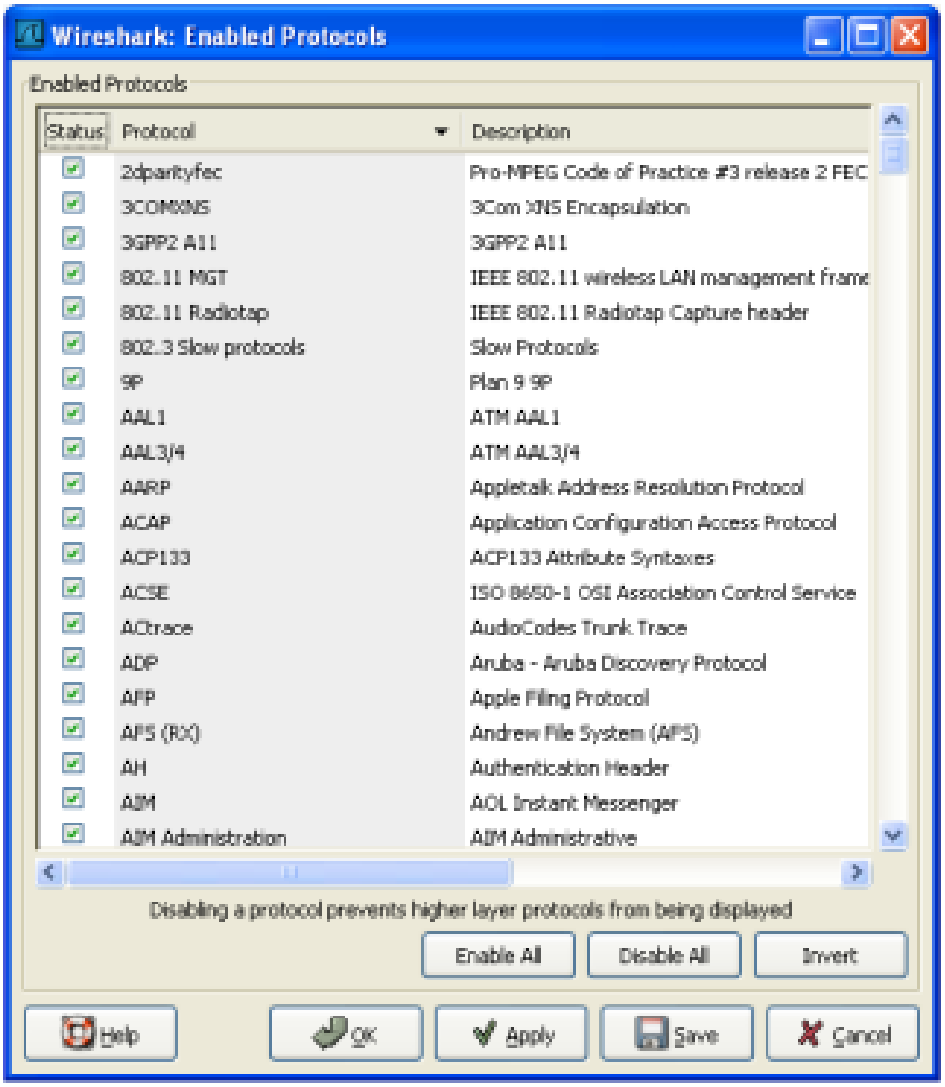
Once the Coloring Rules dialog box is up, there are a number of buttons you can use, depending on whether or not you have any color filters installed already.

Note!

You will need to carefully select the order the coloring rules are listed as they are applied in order from top to bottom. So, more specific rules need to be listed before more general rules. For example, if you have a color rule for UDP before the one for DNS, the color rule for DNS will never be applied (as DNS uses UDP, so the UDP rule will match first).

If this is the first time you have used Coloring Rules, click on the New button which will bring up the Edit color filter dialog box as shown in Figure 10.2, "The "Edit Color Filter" dialog box".

Figure 10.5. The "Enabled Protocols" dialog box

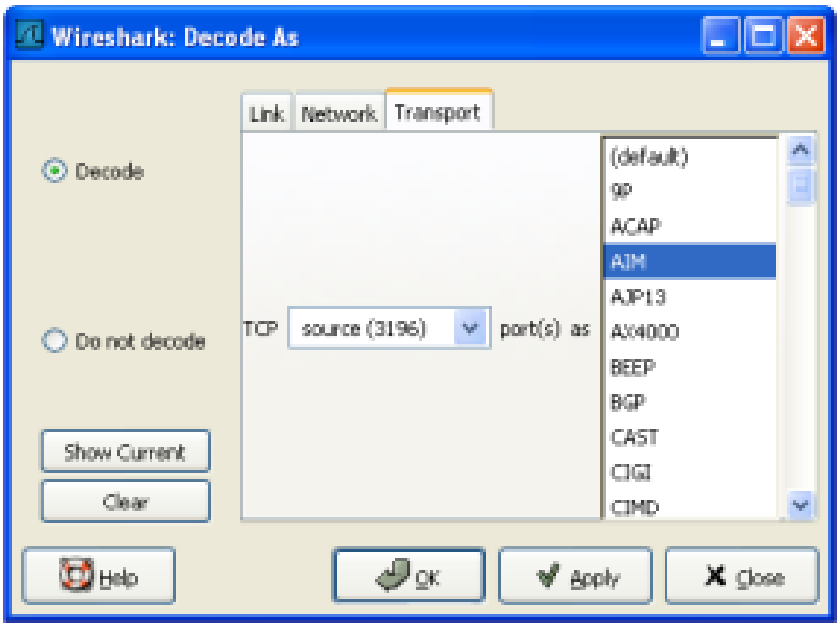


To disable or enable a protocol, simply click on it using the mouse or press the space bar when the protocol is highlighted. Note that typing the first few letters of the protocol name when the Enabled Protocols dialog box is active will temporarily open a search text box and automatically select the first matching protocol name (if it exists).

Warning!

You have to use the Save button to save your settings. The OK or Apply buttons will not save your changes permanently, so they will be lost when Wireshark is closed.

Figure 10.6. The "Decode As" dialog box



The content of this dialog box depends on the selected packet when it was opened.

Warning!

These settings will be lost if you quit Wireshark or change profile, unless you save the entries in the **Show User Specified Decodes...** windows (Section 10.4.3, "Show User Specified Decodes").

1. **Decode**: Decode packets the selected way.

2. **Do not decode**: Do not decode packets the selected way.

3. **Link/Network/Transport**: Specify the network layer at which "Decode As" should take place. Which of these pages are available depends on the content of the selected packet when this dialog box is opened.

4. **Show Current**: Open a dialog box showing the current list of user specified decodes.

5. **OK**: Apply the currently selected decode and close the dialog box.

6. **Apply**: Apply the currently selected decode and keep the dialog box open

Value The value (Label And Cert Value) representing the Category.

Name The textual representation for the value.

0. GeoIP Database Paths

If your copy of Wireshark supports MaxMind's GeoIP library, you can use their databases to match IP addresses to countries, cites, autonomous system numbers, ISPs, and other bits of information. Some

databases are available at no cost, while others require a licensing fee. See the MaxMind web site for more information.

This table is handled by an Section 10.7, "User Table" with the following fields.

Database pathname This specifies a directory containing GeoIP data files. Any files beginning with `Geo` and ending with `.dat` will be automatically loaded. A total of 8 files can be loaded.

The locations for your data files are up to you, but `/usr/share/GeoIP` (Linux), `C:\GeoIP` (Windows), `C:\Program Files\Wireshark\GeoIP` (Windows) might be good choices.

1. IKEv2 decryption table

Wireshark can decrypt Encrypted Payloads of IKEv2 (Internet Key Exchange version 2) packets if necessary information is provided. Note that you can decrypt only IKEv2 packets with this feature. If you want to decrypt IKEv1 packets or ESP packets, use Log Filename setting under ISAKMP protocol preference or settings under ESP protocol preference respectively.

The locations for your data files are up to you, but `/usr/share/GeoIP` (Linux), `C:\GeoIP` (Windows), `C:\Program Files\Wireshark\GeoIP` (Windows) might be good choices.

1. IKEv2 decryption table

Wireshark can decrypt Encrypted Payloads of IKEv2 (Internet Key Exchange version 2) packets if necessary information is provided. Note that you can decrypt only IKEv2 packets with this feature. If you want to decrypt IKEv1 packets or ESP packets, use Log Filename setting under ISAKMP protocol preference or settings under ESP protocol preference respectively.

This table is handled by an Section 10.7, "User Table" with the following fields.

Initiator's SPI — Initiator's SPI of the IKE_SA. This field takes hexadecimal string without "0x" prefix and the length must be 16 hex chars (represents 8 octets).

Responder's SPI — Responder's SPI of the IKE_SA. This field takes hexadecimal string without "0x" prefix and the length must be 16 hex chars (represents 8 octets).

SK_ei — Key used to encrypt/decrypt IKEv2 packets from initiator to responder. This field takes hexadecimal string without "0x" prefix and its length must meet the requirement of the encryption algorithm selected.

SK_er — Key used to encrypt/decrypt IKEv2 packets from responder to initiator. This field takes hexadecimal string without "0x" prefix and its length must meet the requirement of the encryption algorithm selected.

Encryption Algorithm — Encryption algorithm of the IKE_SA.

SK_ai — Key used to calculate Integrity Checksum Data for IKEv2 packets from responder to initiator. This field takes hexadecimal string without "0x" prefix and its length must meet the requirement of the integrity algorithm selected.

SK_ar — Key used to calculate Integrity Checksum Data for IKEv2 packets from initiator to responder. This field takes hexadecimal string without "0x" prefix and its length must meet the requirement of the integrity algorithm selected.