Ethereal User's Guide

V2.00 for Ethereal 0.10.5

Richard Sharpe, NS Computer Software and Services P/L
Ed Warnicke,
Ulf Lamping,

Ethereal User's Guide: V2.00 for Ethereal 0.10.5

by Richard Sharpe, Ed Warnicke, and Ulf Lamping

Table of Contents

List of Figures

List of Tables

List of Examples

Preface

1. Foreword

Ethereal is one of those programs that many network managers would love to be able to use, but they are often prevented from getting what they would like from Ethereal because of the lack of documentation.

This document is part of an effort by the Ethereal team to improve the usability of Ethereal.

We hope that you find it useful, and look forward to your comments.

2. Who should read this document?

The intended audience of this book is anyone using Ethereal.

This book will explain all the basics and also some of the advanced features that Ethereal provides. As Ethereal has become a very complex program since the early days, not every feature of Ethereal might be explained in this book.

This book is not intended to explain network sniffing in general and it will not provide details about specific network protocols. However, as this book evolves in time (like Ethereal itself), this might change in the future.

By reading this book, you will learn how to install Ethereal, how to use the basic elements of the graphical user interface (like the menu) and what's behind some of the advanced features that are maybe not that obvious at first sight. It will hopefully guide you around some common problems that frequently appears for new (and sometimes even advanced) users of Ethereal.

3. Acknowledgements

The authors would like to thank the whole Ethereal team for their assistance. In particular, the authors would like to thank:

- Gerald Combs, for initiating the Ethereal project and funding to do this documentation.
- Guy Harris, for many helpful hints and a great deal of patience in reviewing this document.
- Gilbert Ramirez, for general encouragement and helpful hints along the way.

The authors would also like to thank the following people for their helpful feedback on this document:

- Pat Eyler, for his suggestions on improving the example on generating a backtrace.
- Martin Regner, for his various suggestions and corrections.
- Graeme Hewson, for a lot of grammatical corrections.

The authors would like to acknowledge those man page and README authors for the ethereal project from who sections of this document borrow heavily:

- Scott Renfro from whose **mergecap** man page Section C.5, “mergecap: Merging multiple capture files into one with mergecap ” is derived.
- Ashok Narayanan from whose **text2pcap** man page Section C.6, “text2pcap: Converting ASCII hexdumps to network captures with text2pcap ” is derived.
- Frank Singleton from whose `README.idl2eth` Section C.7, “idl2eth: Creating dissectors from Corba IDL files with idl2eth ” is derived.

4. About this document

This book was originally developed by Richard Sharpe with funds provided from the Ethereal Fund. It was updated by Ed Warnicke and more recently redesigned and updated by Ulf Lamping.

It is written in DocBook/XML.

You will find some specially marked parts in this book:

This is a warning!

You should pay attention to a warning, as otherwise data loss might occur.

This is a note!

A note will point you to common mistakes and things that might not be obvious.

This is a tip!

Tips will be helpful for your everyday work using Ethereal.

5. Where to get the latest copy of this document?

The latest copy of this documentation can always be found at: http://www.ethereal.com/docs/user-guide/ .

6. Providing feedback about this document

Should you have any feedback about this document, please send them to the authors through ethereal-dev[AT]ethereal.com.

Chapter 1. Introduction

1.1. What is Ethereal?

Ethereal is a network packet analyzer. A network packet analyzer will try to capture network packets and tries to display that packet data as detailed as possible.

You could think of a network packet analyzer as a measuring device used to examine what's going on inside a network cable, just like a voltmeter is used by an electrician to examine what's going on inside an electric cable (but at a higher level, of course).

In the past, such tools were either very expensive, proprietary, or both. However, with the advent of Ethereal, all that has changed.

Ethereal is perhaps one of the best open source packet analyzers available today.

1.1.1. Some intended purposes

Here are some examples people use Ethereal for:

- network administrators use it to **troubleshoot network problems**
- network security engineers use it to **examine security problems**
- developers use it to **debug protocol implementations**
- people use it to **learn network protocol** internals

Beside these examples, Ethereal can be helpful in many other situations too.

1.1.2. Features

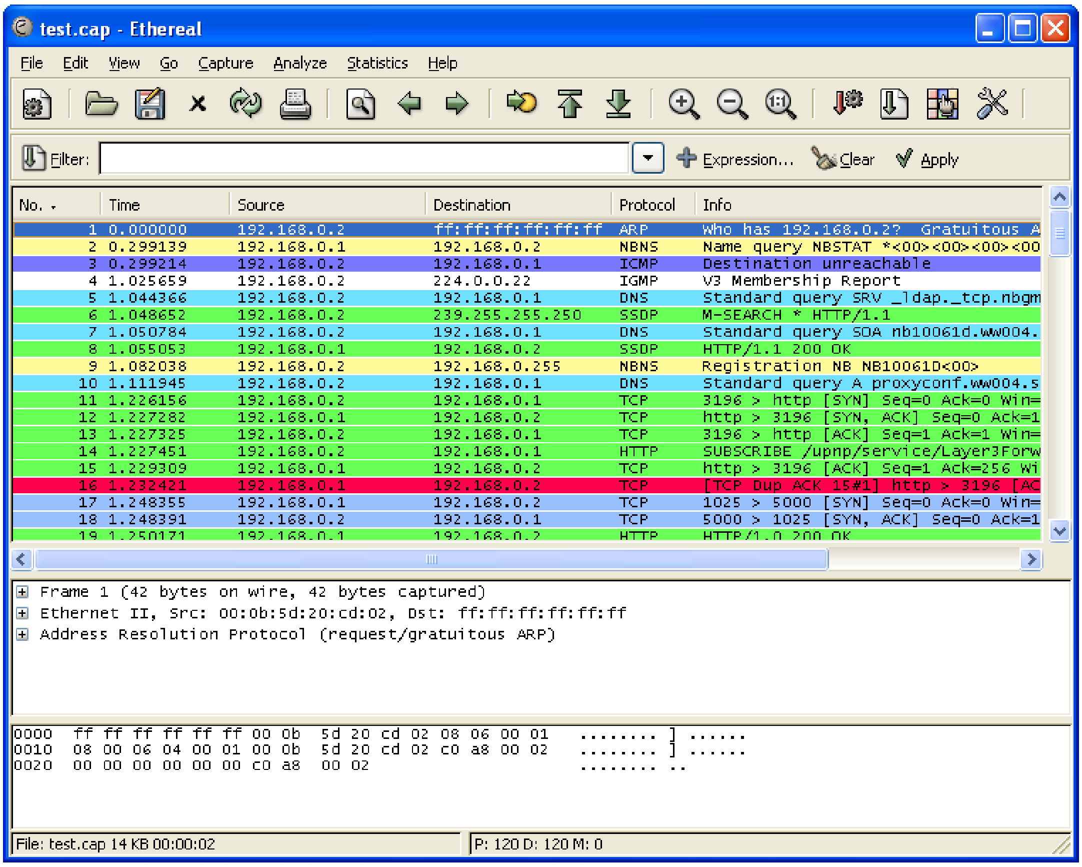
The following are some of the many features Ethereal provides:

- Available for **UNIX** and **Windows**.
- **Capture** live packet data from a network interface.
- Display packets with **very detailed protocol information**.
- **Open and Save** packet data captured.
- **Import and Export** packet data from and to a lot of other capture programs.
- **Filter packets** on many criteria.
- **Search** for packets on many criteria.
- **Colorize** packet display based on filters.
- Create various **statistics**.
- ... and **a lot more!**

However, to really appreciate its power, you have to start using it.

Figure 1.1, “ Ethereal captures packets and allows you to examine their content. ” shows Ethereal having captured some packets and waiting for you to examine them.

Figure 1.1. Ethereal captures packets and allows you to examine their content.


test.cap - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.2? Gratuitous A
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT *<00><00><00><00
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbgm
6	1.048652	192.168.0.2	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004.
8	1.055053	192.168.0.1	192.168.0.2	SSDP	HTTP/1.1 200 OK
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.ww004.s
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Ack=0 Win=
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=1
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win=
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/Layer3Forw
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 Wi
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Dup ACK 15#1] http > 3196 [AC
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Ack=0 Win=
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] Seq=0 Ack=1
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK

```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request/gratuitous ARP)
```

```
0000  ff ff ff ff ff ff 00 0b  5d 20 cd 02 08 06 00 01   ........ ] ......
0010  08 00 06 04 00 01 00 0b  5d 20 cd 02 c0 a8 00 02   ........ ] ......
0020  00 00 00 00 00 00 c0 a8  00 02                     ........ ..
```

File: test.cap 14 KB 00:00:02 | P: 120 D: 120 M: 0

1.1.3. Live capture from many different network media

Despite its name, Ethereal can capture traffic from network media other than Ethernet. Which media types are supported, depends on many things like the operating system you are using. An overview of the supported media types can be found at: http://www.ethereal.com/media.html.

1.1.4. Import files from many other capture programs

Ethereal can open packets captured from a large number of other capture programs. For a list of input formats see Section 5.2.2, “Input File Formats”.

1.1.5. Export files for many other capture programs

Ethereal can save packets captured in a large number of formats of other capture programs. For a list of output formats see Section 5.3.2, “Output File Formats”.

1.1.6. Many protocol decoders

There are protocol decoders (or dissectors, as they are known in Ethereal) for a great many protocols: see Appendix B, *Protocols and Protocol Fields*.

1.1.7. Open Source Software

Ethereal is an open source software project, and is released under the GNU General Public Licence (GPL). You can freely use Ethereal on any number of computers you like, without worrying about license keys or fees or such. In addition, all source code is freely available under the GPL. Because of that, it is very easy for people to add new protocols to Ethereal, either as plugins, or built into the source, and they often do!

1.1.8. What Ethereal is not

Here are some things Ethereal does not provide:

- Ethereal isn't an intrusion detection system. It will not warn you when someone does strange things on your network that he/she isn't allowed to do. However, if strange things happen, Ethereal might help you figure out what is really going on.

- Ethereal will not manipulate things on the network, it will only "measure" things from it. Ethereal doesn't send packets on the network or do other active things (except for name resolutions, but even that can be disabled).

1.2. Platforms Ethereal runs on

Ethereal currently runs on most UNIX platforms and various Windows platforms. It requires GTK+, GLib, libpcap and some other libraries in order to run.

If a binary package is not available for your platform, you should download the source and try to build it. Please report your experiences to ethereal-dev[AT]ethereal.com.

Binary packages are available for at least the following platforms:

1.2.1. Unix

- Apple Mac OS X
- BeOS
- FreeBSD
- HP-UX
- IBM AIX
- NetBSD
- OpenBSD
- SCO UnixWare/OpenUnix
- SGI Irix
- Sun Solaris/Intel
- Sun Solaris/Sparc
- Tru64 UNIX (formerly Digital UNIX)

1.2.2. Linux

- Debian GNU/Linux
- Gentoo Linux
- IBM S/390 Linux (Red Hat)
- Mandrake Linux
- PLD Linux
- Red Hat Linux
- Rock Linux
- Slackware Linux
- Suse Linux

1.2.3. Microsoft Windows

- Windows Me / 98 / 95
- Windows Server 2003 / XP / 2000 / NT 4.0

1.3. Where to get Ethereal?

You can get the latest copy of the program from the Ethereal website: http://www.ethereal.com/download.html. The website allows you to choose from among several mirrors for downloading.

A new Ethereal version will typically become available every 4-8 weeks.

1.4. A rose by any other name

William Shakespeare wrote: *"A rose by any other name would smell as sweet."* And so it is with Ethereal, as there appears to be two different ways that people pronounce the name.

Some people pronounce it ether-real, while others pronounce it e-the-real, as in ghostly, insubstantial, etc.

You are welcome to call it what you like, as long as you find it useful. The FAQ gives the official pronounciation as "e-the-real".

1.5. A brief history of Ethereal

In late 1997, Gerald Combs needed a tool for tracking down networking problems and wanted to learn more about networking, so he started writing Ethereal as a way to solve both problems.

Ethereal was initially released, after several pauses in development, in July 1998 as version 0.2.0. Within days, patches, bug reports, and words of encouragement started arriving, so Ethereal was on its way to success.

Not long after that Gilbert Ramirez saw its potential and contributed a low-level dissector to it.

In October, 1998, Guy Harris of Network Appliance was looking for something better than tcpview, so he started applying patches and contributing dissectors to Ethereal.

In late 1998, Richard Sharpe, who was giving TCP/IP courses, saw its potential on such courses, and started looking at it to see if it supported the protocols he needed. While it didn't at that point, new protocols could be easily added. So he started contributing dissectors and contributing patches.

The list of people who have contributed to Ethereal has become very long since then, and almost all of them started with a protocol that they needed that Ethereal did not already handle. So they copied an existing dissector and contributed the code back to the team.

1.6. Development and maintenance of Ethereal

Ethereal was initially developed by Gerald Combs. Ongoing development and maintenance of Ethereal is handled by the Ethereal team, a loose group of individuals who fix bugs and provide new functionality.

There have also been a large number of people who have contributed protocol dissectors to Ethereal, and it is expected that this will continue. You can find a list of the people who have contributed code to Ethereal by checking the about dialog box of Ethereal, or at the authors page on the Ethereal web site.

Ethereal is an open source software project, and is released under the GNU General Public Licence (GPL). All source code is freely available under the GPL. You are welcome to modify Ethereal to suit your own needs, and it would be appreciated if you contribute your improvements back to the Ethereal team.

You gain three benefits by contributing your improvements back to the community:

- Other people who find your contributions useful will appreciate them, and you will know that you have helped people in the same way that the developers of Ethereal have helped people.
- The developers of Ethereal might improve your changes even more, as there's always room for improvements. Or they may implement some advanced things on top of your code, which can be useful for yourself too.
- The maintainers and developers of Ethereal will maintain your code as well, fixing it when API changes or other changes are made, and generally keeping it in tune with what is happening with Ethereal. So if Ethereal is updated (which is done often), you can get a new Ethereal version from the website and your changes will already be included without any effort for you.

The Ethereal source code and binary kits for some platforms are all available on the download page of the Ethereal website: http://www.ethereal.com/download.html.

1.7. Reporting problems and getting help

If you have problems, or need help with Ethereal, there are several places that may be of interest to you (well, beside this guide of course).

1.7.1. FAQ

The "Frequently Asked Questions" will list often asked questions and the corresponding answers.

Read the FAQ!

Before sending any mail to the mailing lists below, be sure to read the FAQ, as it will often answer the question(s) you might have. This will save yourself and others a lot of time (keep in mind that a lot of people are subscribed to the mailing lists).

You will find the FAQ inside Ethereal by clicking the menu item Help/Contents and selecting the FAQ page in the upcoming dialog.

An online version is available at the ethereal website: http://www.ethereal.com/faq.html. You might prefer this online version, as it's typically more up to date and the HTML format is easier to use.

1.7.2. Mailing Lists

There are several mailing lists of specific Ethereal topics available:

ethereal-announce This mailing list will inform you about new program releases, which usually appear about every 4-8 weeks.

ethereal-users This list is for users of Ethereal. People post questions about building and using Ethereal, others (hopefully) provide answers.

ethereal-dev This list is for Ethereal developers. If you want to start developing a protocol dissector, join this list.

You can subscribe to each of these lists from the Ethereal web site: http://www.ethereal.com. Simply select the **mailing lists** link on the left hand side of the site. The lists are archived at the Ethereal web site as well.

Tip!

You can search in the list archives to see if someone asked the same question some time before and maybe already got an answer. That way you don't have to wait until someone answers your question.

1.7.3. Reporting Problems

Note!

Before reporting any problems, please make sure you have installed the latest version of Ethereal.

When reporting problems with Ethereal, it is helpful if you supply the following information:

1. The version number of Ethereal and the dependent libraries linked with it, eg GTK+, etc. You can obtain this with the command **ethereal -v**.

2. Information about the platform you run Ethereal on.
3. A detailed description of your problem.

Don't send large files!

Do not send large files (>100KB) to the mailing lists, just place a note that further data is available on request. Large files will only annoy a lot of people on the list who are not interested in your specific problem. If required, you will be asked for further data by the persons who really can help you.

Don't send confidential information!

If you send captured data to the mailing lists, be sure they don't contain any sensitive or confidential information like passwords or such.

1.7.4. Reporting Crashes on UNIX/Linux platforms

When reporting crashes with Ethereal, it is helpful if you supply the traceback information (besides the information mentioned in "Reporting Problems").

You can obtain this traceback information with the following commands:

```
$ gdb `whereis ethereal | cut -f2 -d: | cut -d' ' -f2` core >& bt.txt
backtrace
^D
$
```

Note

Type the characters in the first line verbatim! Those are back-tics there!

Note

backtrace is a **gdb** command. You should enter it verbatim after the first line shown above, but it will not be echoed. The ^D (Control-D, that is, press the Control key and the D key together) will cause **gdb** to exit. This will leave you with a file called `bt.txt` in the current directory. Include the file with your bug report.

Note

If you do not have **gdb** available, you will have to check out your operating system's debugger.

You should mail the traceback to the ethereal-dev[AT]ethereal.com mailing list.

1.7.5. Reporting Crashes on Windows platforms

The Windows distributions don't contain the symbol files (.pdb), because they are very large. For this reason it's not possible to create a meaningful backtrace file from it. You should report your crash just like other problems, using the mechanism described above.

Chapter 2. Building and Installing Ethereal

2.1. Introduction

As with all things, there must be a beginning, and so it is with Ethereal. To use Ethereal, you must:

- Obtain a binary package for your operating system, or
- Obtain the source and build Ethereal for your operating system.

Currently, only two or three Linux distributions ship Ethereal, and they are commonly shipping an out-of-date version. No other versions of UNIX ship Ethereal so far, and Microsoft does not ship it with any version of Windows. For that reason, you will need to know where to get the latest version of Ethereal and how to install it.

This chapter shows you how to obtain source and binary packages, and how to build Ethereal from source, should you choose to do so.

The following are the general steps you would use:

1. Download the relevant package for your needs, e.g. source or binary distribution.
2. Build the source into a binary, if you have downloaded the source.

   This may involve building and/or installing other necessary packages.
3. Install the binaries into their final destinations.

2.2. Obtaining the source and binary distributions

You can obtain both source and binary distributions from the Ethereal web site: http://www.ethereal.com. Simply select the download link, and then select either the source package or binary package of your choice from the mirror site closest to you.

Download all the needed files

In general, unless you have already downloaded Ethereal before, you will most likely need to download several source packages if you are building Ethereal from source. This is covered in more detail below.

Once you have downloaded the relevant files, you can go on to the next step.

Note!

While you will find a number of binary packages available on the Ethereal web site, you might not find one for your platform, and they often tend to be several versions behind the current released version, as they are contributed by people who have the platforms they are built for.

For this reason, you might want to pull down the source distribution and build it, as the process is relatively simple.

2.3. Before you build Ethereal

Before you build Ethereal from sources, or install a binary package, you must ensure that you have the following other packages installed:

- GTK+, The GIMP Tool Kit.

  You will also need Glib. Both can be obtained from www.gtk.org

- libpcap, the packet capture software that Ethereal uses.

  You can obtain libpcap from www.tcpdump.org

Depending on your system, you may be able to install these from binaries, e.g. RPMs, or you may need to obtain them in source code form and build them.

If you have downloaded the source for GTK+, the instructions shown in Example 2.1, “Building GTK+ from source” may provide some help in building it:

Example 2.1. Building GTK+ from source

```
gzip -dc gtk+-1.2.10.tar.gz | tar xvf -
<much output removed>
cd gtk+-1.2.10
./configure
<much output removed>
make
<much output removed>
make install
<much output removed>
```

Note!

You may need to change the version number of gtk+ in Example 2.1, “Building GTK+ from source” to match the version of GTK+ you have downloaded. The directory you change to will change if the version of GTK+ changes, and in all cases, **tar xvf -** will show you the name of the directory you should change to.

Note!

If you use Linux, or have GNU **tar** installed, you can use **tar zxvf gtk+-1.2.10.tar.gz**. It is also possible to use **gunzip -c** or **gzcat** rather than **gzip -dc** on many UNIX systems.

Note!

If you downloaded gtk+ or any other tar file using Windows, you may find your file called gtk+-1_2_8_tar.gz.

You should consult the GTK+ web site if any errors occur in carrying out the instructions in Example 2.1, “Building GTK+ from source”.

If you have downloaded the source to libpcap, the general instructions shown in Example 2.2, “Building and installing libpcap” will assist in building it. Also, if your operating system does not support **tcpdump**, you might also want to download it from the tcpdump web site and install it.

Example 2.2. Building and installing libpcap

```
gzip -dc libpcap-0.8.3.tar.Z | tar xvf -
<much output removed>
cd libpcap_0_8_3
./configure
<much output removed>
make
<much output removed>
make install
<much output removed>
make install-incl
<much output removed>
```

Note!

The directory you should change to will depend on the version of libpcap you have downloaded. In all cases, **tar xvf -** will show you the name of the directory that has been unpacked.

When installing the include files, you might get the error shown in Example 2.3, “Errors while installing the libpcap include files” when you submit the command **make install-incl**.

Example 2.3. Errors while installing the libpcap include files

```
/usr/local/include/pcap.h
/usr/bin/install -c -m 444 -o bin -g bin ./pcap-namedb.h \
/usr/local/include/pcap-namedb.h
/usr/bin/install -c -m 444 -o bin -g bin ./net/bpf.h \
/usr/local/include/net/bpf.h
/usr/bin/install: cannot create regular file \
`/usr/local/include/net/bpf.h': No such file or directory
make: *** [install-incl] Error 1
```

If you do, simply create the missing directory with the following command:

```
mkdir /usr/local/include/net
```

and rerun the command **make install-incl**.

Under RedHat 6.x and beyond (and distributions based on it, like Mandrake) you can simply install each of the packages you need from RPMs. Most Linux systems will install GTK+ and Glib in anycase, however, you will probably need to install the devel versions of each of these packages. The commands shown in Example 2.4, “ Installing required RPMs under RedHat Linux 6.2 and beyond ” will install all the needed RPMs if they are not already installed.

Example 2.4. Installing required RPMs under RedHat Linux 6.2 and beyond

```
cd /mnt/cdrom/RedHat/RPMS
rpm -ivh glib-1.2.6-3.i386.rpm
rpm -ivh glib-devel-1.2.6-3.i386.rpm
rpm -ivh gtk+-1.2.6-7.i386.rpm
rpm -ivh gtk+-devel-1.2.6-7.i386.rpm
rpm -ivh libpcap-0.4-19.i386.rpm
```

Note

If you are using a version of RedHat later than 6.2, the required RPMs have most likely changed. Simply use the correct RPMs from your distribution.

Under Debian you can install Ethereal using apt-get. apt-get will handle any dependency issues for you. Example 2.5, “Installing debs under Debian” shows how to do this.

Example 2.5. Installing debs under Debian

```
apt-get install ethereal
```

2.4. Building Ethereal from source under UNIX

Use the following general steps if you are building Ethereal from source under a UNIX operating system:

1. Unpack the source from its **gzip**'d **tar** file. If you are using Linux, or your version of UNIX uses GNU **tar**, you can use the following command:

   ```
   tar zxvf ethereal-0.10.5-tar.gz
   ```

   For other versions of UNIX, You will want to use the following commands:

   ```
   gzip -d ethereal-0.10.5-tar.gz
   tar xvf ethereal-0.10.5-tar
   ```

   ### Note!

   The pipeline **gzip -dc ethereal-0.10.5-tar.gz | tar xvf -** will work here as well.

   ### Note!

   If you have downloaded the Ethereal tarball under Windows, you may find that your browser has created a file with underscores rather than periods in its file name.

2. Change directory to the Ethereal source directory.

3. Configure your source so it will build correctly for your version of UNIX. You can do this with the following command:

   ```
   ./configure
   ```

   If this step fails, you will have to rectify the problems and rerun **configure**. Troubleshooting hints are provided in Section 2.6, “Troubleshooting during the install on Unix”.

4. Build the sources into a binary, with the **make** command. For example:

   ```
   make
   ```

5. Install the software in its final destination, using the command:

   ```
   make install
   ```

Once you have installed Ethereal with **make install** above, you should be able to run it by entering **ethereal**.

2.5. Installing the binaries under UNIX

In general, installing the binary under your version of UNIX will be specific to the installation methods used with your version of UNIX. For example, under AIX, you would use **smit** to install the Ethereal binary package, while under Tru64 UNIX (formerly Digital UNIX) you would use **setld**.

2.5.1. Installing from rpm's under RedHat and alike

Use the following command to install the Ethereal RPM that you have downloaded from the Ethereal web site:

```
rpm -ivh ethereal-0.10.5-0.2.2.i386.rpm
```

If the above step fails because of missing dependencies, install the dependencies first, and then retry the step above. See Example 2.4, “ Installing required RPMs under RedHat Linux 6.2 and beyond ” for information on what RPMs you will need to have installed.

2.5.2. Installing from deb's under Debian

Use the following command to install Ethereal under Debian:

```
apt-get install ethereal
```

apt-get should take care of all of the dependency issues for you.

2.6. Troubleshooting during the install on Unix

A number of errors can occur during the installation process. Some hints on solving these are provided here.

If the **configure** stage fails, you will need to find out why. You can check the file `config.log` in the source directory to find out what failed. The last few lines of this file should help in determining the problem.

The standard problems are that you do not have GTK+ on your system, or you do not have a recent enough version of GTK+. The **configure** will also fail if you do not have libpcap (at least the required include files) on your system.

Another common problem is for the final compile and link stage to terminate with a complaint of: Output too long. This is likely to be caused by an antiquated **sed** (such as the one shipped with Solaris). Since **sed** is used by the **libtool** script to construct the final link command, this leads to mysterious problems. This can be resolved by downloading a recent version of sed from http://www.gnu.org/directory/sed.html .

If you cannot determine what the problems are, send mail to the **ethereal-dev** mailing list explaining your problem, and including the output from `config.log` and anything else you think is relevant, like a trace of the **make** stage.

2.7. Building from source under Windows

It is recommended to use the binary installer for Windows, until you want to start developing Ethereal on this platform.

For further information how to build Ethereal for Windows from the sources, have a look at the file Readme.win32, which can be found in the doc directory of the sources.

2.8. Installing Ethereal under Windows

In this section we explore installing Ethereal under Windows from the binary packages. You must follow two steps:

1. Install WinPcap. You will find a single installer exe called something like "auto-installer", which can be installed under various Windows systems, including 9x/Me/NT4.0/2000/XP. This installer is located at: http://winpcap.polito.it/install/Default.htm. You should download the latest released version (the latest one not marked "beta") and execute it.

2. Install Ethereal. You may acquire a binary installable of Ethereal at http://www.ethereal.com/download.html#binaries. Download the installer and execute it.

Both steps are extremely simply, as you only have to download and install the two exe files.

2.8.1. Update

From time to time you may want to update your installed Ethereal to a more recent version. If you join Ethereal's announce mailing list, you will be informed about new Ethereal versions, see Section 1.7.2, “Mailing Lists” for details how to subscribe to this list.

- **Update Ethereal**. New versions of Ethereal usually become available every 4-8 weeks. Updating Ethereal is done the same way as installing it, you simply download and start the installer exe. A reboot is usually not required and all your personal settings remain unchanged.

- **Update WinPcap**. New versions of WinPcap are less frequently available, maybe only once a year. You will find WinPcap update instructions where you can download new versions. Usually you have to reboot the machine after installing a new WinPcap version.

2.8.2. Uninstall Ethereal

You can uninstall Ethereal the usual way, using the Software option inside the Control Panel. You will find two entries, one for Ethereal itself and one for WinPcap.

Chapter 3. User Interface

3.1. Introduction

By now you have installed Ethereal and are most likely keen to get started capturing your first packets. In the next chapters we will explore:

- How the Ethereal user interface works
- How to capture packets in Ethereal
- How to view packets in Ethereal
- How to filter packets in Ethereal
- ... and many other things!

3.2. Start Ethereal

You can start Ethereal from your shell or window manager.

Tip!

When starting Ethereal it's possible to specify optional settings using the command line. See Section 9.2, “Start Ethereal from the command line” for details.

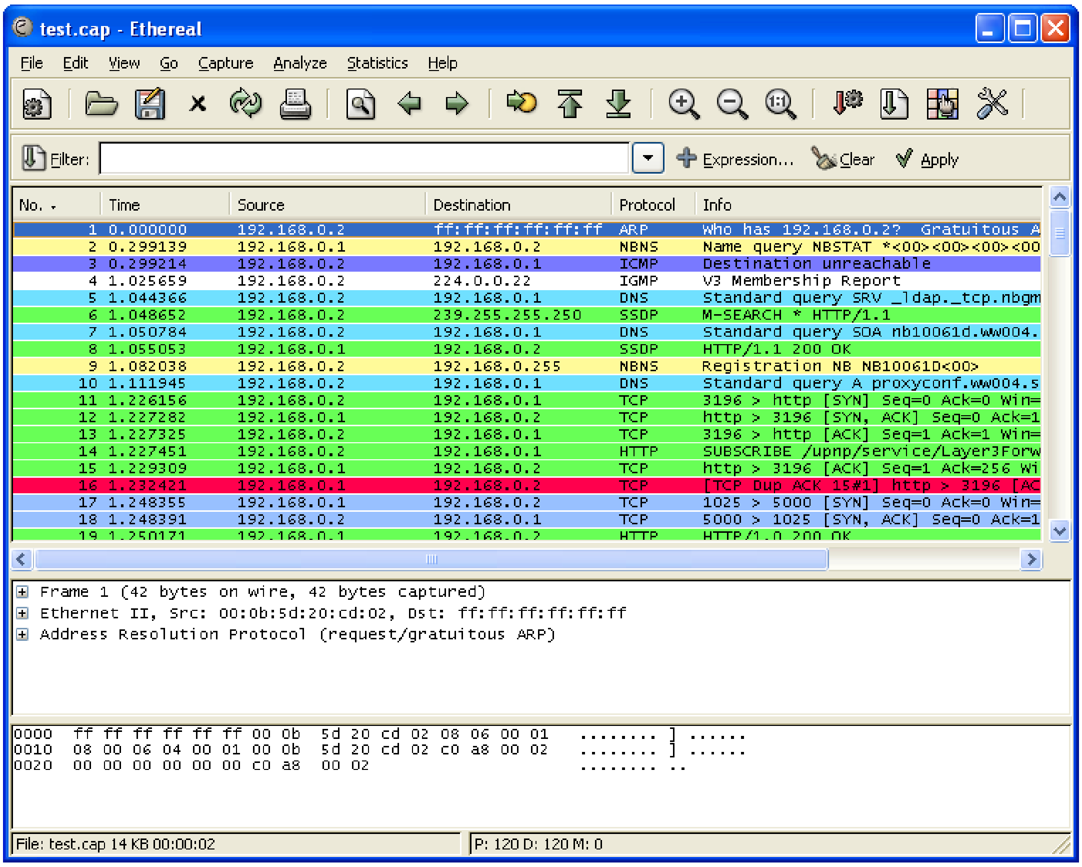
Note!

In the following chapters, a lot of screenshots from Ethereal will be shown. As Ethereal runs on many different platforms and there are different versions of the underlying GUI toolkit (GTK 1.x / 2.x) used, your screen might look different from the provided screenshots. But as there are no real differences in functionality, these screenshots should still be understandable.

3.3. The Main window

Lets look at Ethereal's user interface. Figure 3.1, “The Main window” shows Ethereal as you would usually see it after some packets captured or loaded (how to do this will be described later).

Figure 3.1. The Main window



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.2? Gratuitous A
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT *<00><00><00><00
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbgm
6	1.048652	192.168.0.2	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004.
8	1.055053	192.168.0.1	192.168.0.2	SSDP	HTTP/1.1 200 OK
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.ww004.s
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Ack=0 Win=
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=1
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win=
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/Layer3Forw
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 Wi
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Dup ACK 15#1] http > 3196 [AC
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Ack=0 Win=
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] Seq=0 Ack=1
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK

```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request/gratuitous ARP)
```

```
0000  ff ff ff ff ff ff 00 0b  5d 20 cd 02 08 06 00 01   ........ ] ......
0010  08 00 06 04 00 01 00 0b  5d 20 cd 02 c0 a8 00 02   ........ ] ......
0020  00 00 00 00 00 00 c0 a8  00 02                     ........ ..
```

File: test.cap 14 KB 00:00:02 | P: 120 D: 120 M: 0

Ethereal's main window consist of parts that are commonly known from many other GUI programs.

1. The *menu* (see Section 3.4, “The Menu”) is used to start actions.

2. The *main toolbar* (see Section 3.13, “The "Main" toolbar”) provides quick access to frequently used items from the menu.

3. The *filter toolbar* (see Section 3.14, “The "Filter" toolbar”) provides a way to directly manipulate the currently used display filter (see Section 6.2, “Filtering packets while viewing”).

4. The *packet list pane* (see Section 3.15, “The "Packet List" pane”) displays a summary of each packet captured. By clicking on packets in this pane you control what is displayed in the other two panes.

5. The *packet details pane* (see Section 3.16, “The "Packet Details" pane”) displays the packet selected in the packet list pane in more detail.

6. The *packet bytes pane* (see Section 3.17, “The "Packet Bytes" pane”) displays the data from the packet selected in the packet list pane, and highlights the field selected in the packet details pane.

7. The *statusbar* (see Section 3.18, “The Statusbar”) shows some detailed information about the

current program state and the captured data.

Tip!

The layout of the main window can be customized by changing preference settings. See Figure 9.9, “The "User Interface: Layout" preferences page” for details!

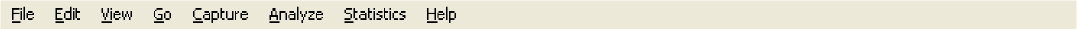
3.4. The Menu

The Ethereal menu sits on top of the Ethereal window. An example is shown in Figure 3.2, “The Menu”.

Note!

Menu items will be greyed out if the corresponding feature isn't available. For example, you cannot save a capture file if you didn't capture or load any data before.

Figure 3.2. The Menu



It contains the following items:

File This menu contains tems to open and merge capture files, save / print / export capture files in whole or in part, and to quit from Ethereal. See Section 3.5, “The "File" menu”.

Edit This menu contains items to find a packet, time reference or mark one or more packets, set your preferences, (cut, copy, and paste are not presently implemented). See Section 3.6, “The "Edit" menu”.

View This menu controls the display of the captured data, including the colorization of packets, zooming the font, show a packet in a separate window, expand and collapse trees in packet details, ... See Section 3.7, “The "View" menu”.

Go This menu contains items to go to a specific packet. See Section 3.8, “The "Go" menu”.

Capture This menu allows you to start and stop captures and to edit capture filters. See Section 3.9, “The "Capture" menu”.

Analyze This menu contains items to manipulate display filters, enable or disable the dissection of protocols, configure user specified decodes and follow a TCP stream. See Section 3.10, “The "Analyze" menu”.

Statistics This menu contains menu-items to display various statistic windows, including a summary of the packets that have been captured, display protocol hierarchy statistics and much more. See Section 3.11, “The "Statistics" menu”.

Help This menu contains items to help the user, like access to some basic help, a list of the supported protocols, manual pages, online access to some of the webpages, and the usual about dialog. See Section 3.12, “The "Help" menu”.

Each of these menu items is described in more detail in the sections that follow.

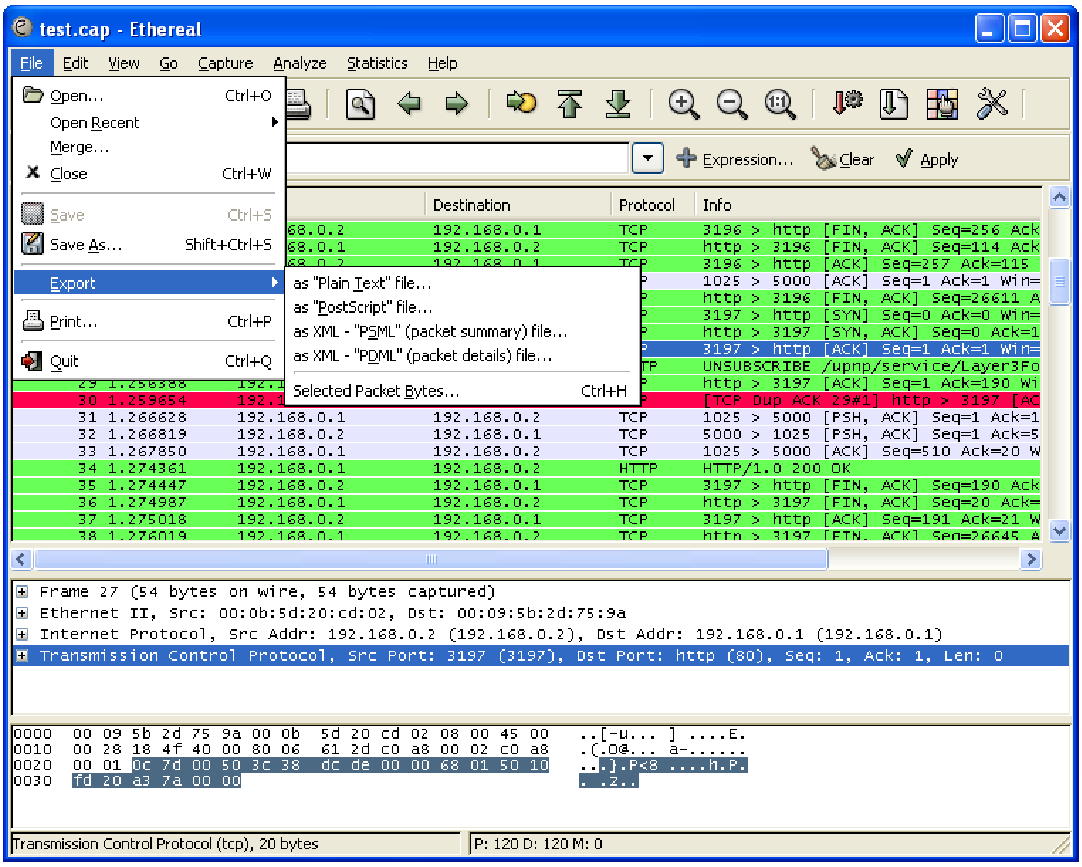
Tip!

You can access menu items directly or by pressing the corresponding accelerator keys, which are shown at the right side of the menu. For example, you can press the Control (or Strg in german) and the K keys together to open the capture dialog.

3.5. The "File" menu

The Ethereal file menu contains the fields shown in Table 3.1, “File menu items”.

Figure 3.3. The "File" Menu



```
68.0.2         192.168.0.1   TCP    3196 > http [FIN, ACK] Seq=256 Ack
68.0.1         192.168.0.2   TCP    http > 3196 [FIN, ACK] Seq=114 Ack
68.0.2         192.168.0.1   TCP    3196 > http [ACK] Seq=257 Ack=115
                                    1025 > 5000 [ACK] Seq=1 Ack=1 Win=
                                    http > 3196 [FIN, ACK] Seq=26611 A
                                    3197 > http [SYN] Seq=0 Ack=0 Win=
                                    http > 3197 [SYN, ACK] Seq=0 Ack=1
                                    3197 > http [ACK] Seq=1 Ack=1 Win=
                                    UNSUBSCRIBE /upnp/service/Layer3Fo
29 1.256388    192.1                http > 3197 [ACK] Seq=1 Ack=190 Wi
30 1.259654    192.1                [TCP Dup ACK 29#1] http > 3197 [AC
31 1.266628    192.168.0.1   192.168.0.2   TCP    1025 > 5000 [PSH, ACK] Seq=1 Ack=1
32 1.266819    192.168.0.2   192.168.0.1   TCP    5000 > 1025 [PSH, ACK] Seq=1 Ack=5
33 1.267850    192.168.0.1   192.168.0.2   TCP    1025 > 5000 [ACK] Seq=510 Ack=20 W
34 1.274361    192.168.0.1   192.168.0.2   HTTP   HTTP/1.0 200 OK
35 1.274447    192.168.0.2   192.168.0.1   TCP    3197 > http [FIN, ACK] Seq=190 Ack
36 1.274987    192.168.0.1   192.168.0.2   TCP    http > 3197 [FIN, ACK] Seq=20 Ack=
37 1.275018    192.168.0.2   192.168.0.1   TCP    3197 > http [ACK] Seq=191 Ack=21 W
38 1.276019    192.168.0.1   192.168.0.2   TCP    http > 3197 [FIN, ACK] Seq=26645 A

⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0

0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..

Transmission Control Protocol (tcp), 20 bytes    P: 120 D: 120 M: 0
```

Table 3.1. File menu items

Menu Item	Accelerator	Description
Open...	Ctrl+O	This menu 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”.
Open Recent		This menu item shows a submenu containing the recently opened capture files. Clicking on one of the submenu items will open the corresponding capture file directly.
Merge...		This menu item brings up the merge file dialog box that allows you to merge a capture file into the currently loaded one. It is discussed in more detail in Section 5.4, “Merging capture files”.
Close	Ctrl+W	This menu item closes the current capture. If you haven't saved the capture, you will be asked to do so first (this can be disabled by a preference setting).

Menu Item	Accelerator	Description

Save	Ctrl+S	This menu item saves the current capture. If you have not set a default capture file name (perhaps with the -w <capfile> option), Ethereal 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 have already saved the current capture, this menu item will be greyed out. **Note!** You cannot save a live capture while it is in progress. You must stop the capture in order to save.
Save As...	Shift+Ctrl+S	This menu 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”).

Export > as "Plain Text" file...		This menu item allows you to export all, or some, of the packets in the capture file to a plain ASCII text file. It pops up the Ethereal Export dialog box (which is discussed further in Section 5.5.1, “The "Export as Plain Text File" dialog box”).
Export > as "PostScript" file...		This menu item allows you to export the (or some) of the packets in the capture file to a PostScript file. It pops up the Ethereal Export dialog box (which is discussed further in Section 5.5.2, “The "Export as PostScript File" dialog box”).
Export > as "PSML" file...		This menu item allows you to export the (or some) of the packets in the capture file to a PSML (packet summary markup language) XML file. It pops up the Ethereal Export dialog box (which is discussed further in Section 5.5.3, “The "Export as PSML File" dialog box”).
Export > as "PDML" file...		This menu item allows you to export the (or some) of the packets in the capture file to a PDML (packet details markup language) XML file. It pops up the Ethereal Export dialog box (which is discussed further in Section 5.5.4, “The "Export as PDML File" dialog box”).
Export > Selected Packet Bytes...	Ctrl+H	This menu item allows you to export the currently selected bytes in the packet bytes pane to a binary file. It pops up the Ethereal Export dialog box (which is discussed further in Section 5.5.5, “The "Export selected packet bytes" dialog box”)

Print...	Ctrl+P	This menu item allows you to print all (or some of) the pack-

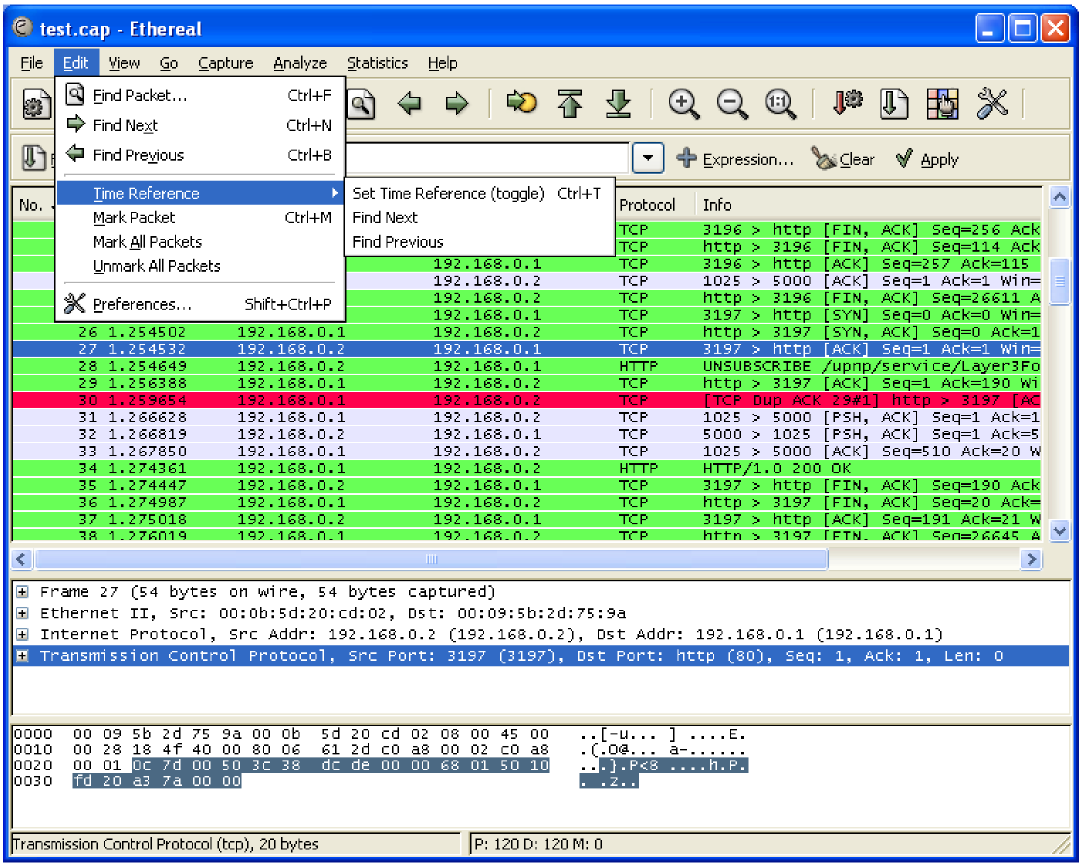
Menu Item	Accelerator	Description
		ets in the capture file. It pops up the Ethereal Print dialog box (which is discussed further in Section 5.6, “Printing packets”).

Quit	Ctrl+Q	This menu item allows you to quit from Ethereal. Ethereal will ask to save your capture file if you haven't saved it before (this can be disabled by a preference setting).

3.6. The "Edit" menu

The Ethereal Edit menu contains the fields shown in Table 3.2, “Edit menu items”.

Figure 3.4. The "Edit" Menu



```
                                                            TCP      3196 > http [FIN, ACK] Seq=256 Ack
                                                            TCP      http > 3196 [FIN, ACK] Seq=114 Ack
                                          192.168.0.1       TCP      3196 > http [ACK] Seq=257 Ack=115
                                          192.168.0.2       TCP      1025 > 5000 [ACK] Seq=1 Ack=1 Win=
                                          192.168.0.2       TCP      http > 3196 [FIN, ACK] Seq=26611 A
                                          192.168.0.1       TCP      3197 > http [SYN] Seq=0 Ack=0 Win=
    26 1.254502     192.168.0.1           192.168.0.2       TCP      http > 3197 [SYN, ACK] Seq=0 Ack=1
    27 1.254532     192.168.0.2           192.168.0.1       TCP      3197 > http [ACK] Seq=1 Ack=1 Win=
    28 1.254649     192.168.0.2           192.168.0.1       HTTP     UNSUBSCRIBE /upnp/service/Layer3Fo
    29 1.256388     192.168.0.1           192.168.0.2       TCP      http > 3197 [ACK] Seq=1 Ack=190 Wi
    30 1.259654     192.168.0.1           192.168.0.2       TCP      [TCP Dup ACK 29#1] http > 3197 [AC
    31 1.266628     192.168.0.1           192.168.0.2       TCP      1025 > 5000 [PSH, ACK] Seq=1 Ack=1
    32 1.266819     192.168.0.2           192.168.0.1       TCP      5000 > 1025 [PSH, ACK] Seq=1 Ack=5
    33 1.267850     192.168.0.1           192.168.0.2       TCP      1025 > 5000 [ACK] Seq=510 Ack=20 W
    34 1.274361     192.168.0.1           192.168.0.2       HTTP     HTTP/1.0 200 OK
    35 1.274447     192.168.0.2           192.168.0.1       TCP      3197 > http [FIN, ACK] Seq=190 Ack
    36 1.274987     192.168.0.1           192.168.0.2       TCP      http > 3197 [FIN, ACK] Seq=20 Ack=
    37 1.275018     192.168.0.2           192.168.0.1       TCP      3197 > http [ACK] Seq=191 Ack=21 W
    38 1.276019     192.168.0.1           192.168.0.2       TCP      http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

Transmission Control Protocol (tcp), 20 bytes

P: 120 D: 120 M: 0

Table 3.2. Edit menu items

Menu Item	Accelerator	Description
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 6.6, “Finding packets”.
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...".

Time Reference > Set Time Reference	Ctrl+T	This menu item set a time reference on the currently selected packet. See Section 6.9.1, “Packet time referencing” for more information about the time referenced packets.
Time Reference > Find Next		This menu item tries to find the next time referenced packet.

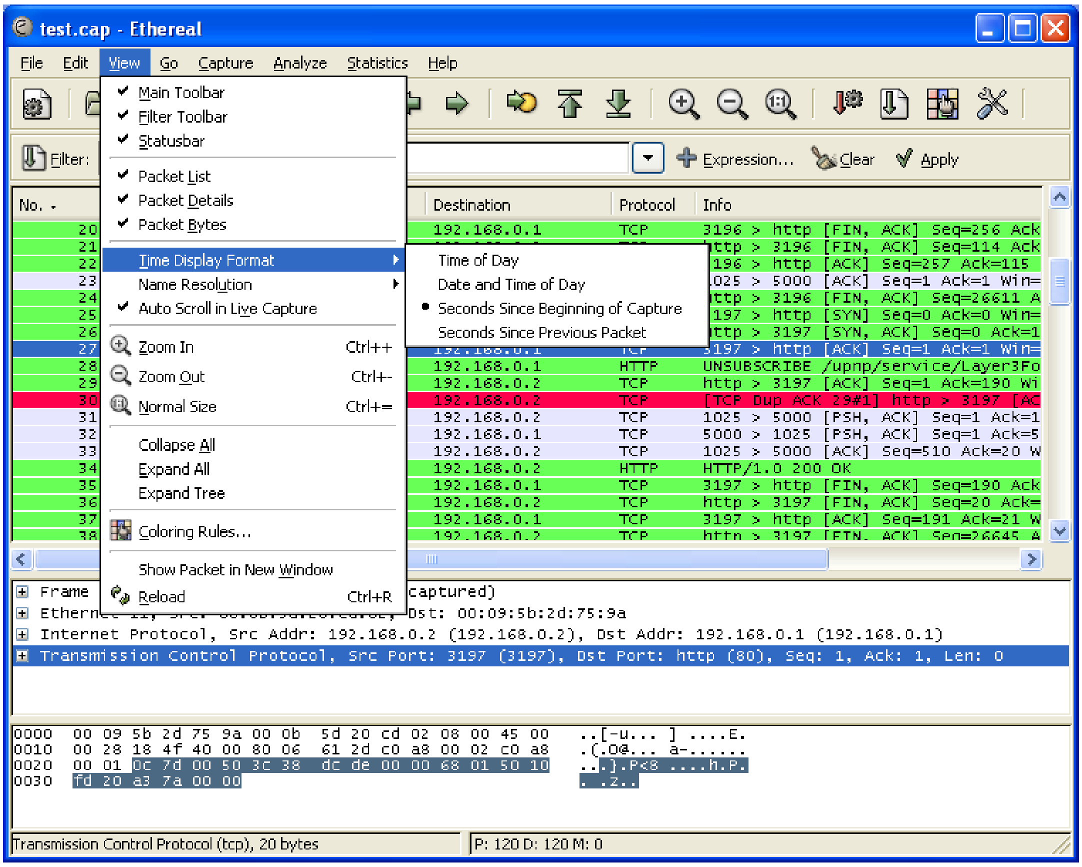
Menu Item	Accelerator	Description
Time Reference > Find Previous		This menu item tries to find the previous time referenced packet.
Mark Packet	Ctrl+M	This menu item "marks" the currently selected packet. See Section 6.8, “Marking packets” for details.
Mark All Packets		This menu item "marks" all packets.
Unmark All Packets		This menu item "unmarks" all marked packets.

Preferences...	Shift+Ctrl+P	This menu item brings up a dialog box that allows you to set preferences for many parameters that control Ethereal. You can also save your preferences so Ethereal will use them the next time you start it. More detail is provided in Section 9.5, “Preferences”.

3.7. The "View" menu

The Ethereal View menu contains the fields shown in Table 3.3, “View menu items”.

Figure 3.5. The "View" Menu



```
No.  Destination    Protocol  Info
20   192.168.0.1    TCP       3196 > http [FIN, ACK] Seq=256 Ack
21                            http > 3196 [FIN, ACK] Seq=114 Ack
22                            196 > http [ACK] Seq=257 Ack=115
23                            025 > 5000 [ACK] Seq=1 Ack=1 Win=
24                            ttp > 3196 [FIN, ACK] Seq=26611 A
25                            197 > http [SYN] Seq=0 Ack=0 Win=
26                            ttp > 3197 [SYN, ACK] Seq=0 Ack=1
27                            3197 > http [ACK] Seq=1 Ack=1 Win=
28   192.168.0.1    HTTP      UNSUBSCRIBE /upnp/service/Layer3Fo
29   192.168.0.2    TCP       http > 3197 [ACK] Seq=1 Ack=190 Wi
30   192.168.0.2    TCP       [TCP Dup ACK 29#1] http > 3197 [AC
31   192.168.0.2    TCP       1025 > 5000 [PSH, ACK] Seq=1 Ack=1
32   192.168.0.1    TCP       5000 > 1025 [PSH, ACK] Seq=1 Ack=5
33   192.168.0.2    TCP       1025 > 5000 [ACK] Seq=510 Ack=20 W
34   192.168.0.2    HTTP      HTTP/1.0 200 OK
35   192.168.0.1    TCP       3197 > http [FIN, ACK] Seq=190 Ack
36   192.168.0.2    TCP       http > 3197 [FIN, ACK] Seq=20 Ack=
37   192.168.0.1    TCP       3197 > http [ACK] Seq=191 Ack=21 W
38   192.168.0.2    TCP       http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame                  captured)
⊞ Ethern                 Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

Transmission Control Protocol (tcp), 20 bytes | P: 120 D: 120 M: 0

Table 3.3. View menu items

Menu Item	Accelerator	Description
Main Toolbar		This menu item hides or shows the main toolbar, see Section 3.13, “The "Main" toolbar”.
Filter Toolbar		This menu item hides or shows the filter toolbar, see Section 3.14, “The "Filter" toolbar”.
Statusbar		This menu item hides or shows the statusbar, see Section 3.18, “The Statusbar”.

Packet List		This menu item hides or shows the packet list pane, see Section 3.15, “The "Packet List" pane”.
Packet Details		This menu item hides or shows the packet details pane, see Section 3.16, “The "Packet Details" pane”.

Menu Item	Accelerator	Description
Packet Bytes		This menu item hides or shows the packet bytes pane, see Section 3.17, “The "Packet Bytes" pane”.

Time Display Format > Time of Day		Selecting this tells Ethereal to display time stamps in time of day format, see Section 6.9, “Time display formats and time references”. **Note!** The fields "Time of Day", "Date and Time of Day", "Seconds Since Beginning of Capture" and "Seconds Since Previous Packet" are mutually exclusive.
Time Display Format > Date and Time of Day		Selecting this tells Ethereal to display the time stamps in date and time of day format, see Section 6.9, “Time display formats and time references”.
Time Display Format > Seconds Since Beginning of Capture		Selecting this tells Ethereal to display time stamps in seconds since beginning of capture format, see Section 6.9, “Time display formats and time references”.
Time Display Format > Seconds Since Previous Packet		Selecting this tells Ethereal to display time stamps in seconds since previous packet format, see Section 6.9, “Time display formats and time references”.
Name Resolution > Resolve Name		This item allows you to trigger a name resolve of the current packet only, see Section 7.4, “Name Resolution”.
Name Resolution > Enable for MAC Layer		This item allows you to control whether or not Ethereal translates MAC addresses into names, see Section 7.4, “Name Resolution”.
Name Resolution > Enable for Network Layer		This item allows you to control whether or not Ethereal translates network addresses into names, see Section 7.4, “Name Resolution”.
Name Resolution > Enable for Transport Layer		This item allows you to control whether or not Ethereal translates transport addresses into names, see Section 7.4, “Name Resolution”.
Auto Scroll in Live Capture		This item allows you to specify that Ethereal should scroll the packet list pane as new packets come in, so you are always looking at the last packet. If you do not specify this, Ethereal simply adds new packets onto the end of the list, but does not scroll the packet list pane.

Zoom In	Ctrl++	Zoom into the packet data (increase the font size).
Zoom Out	Ctrl+-	Zoom out of the packet data (decrease the font size).

Menu Item	Accelerator	Description
Normal Size	Ctrl+=	Set zoom level back to 100% (set font size back to normal).

Collapse All		Ethereal keeps a list of all the protocol subtrees that are expanded, and uses it to ensure that the correct subtrees are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.
Expand All		This menu item expands all subtrees in all packets in the capture.
Expand Tree		This menu item expands the currently selected subtree in the packet details tree.

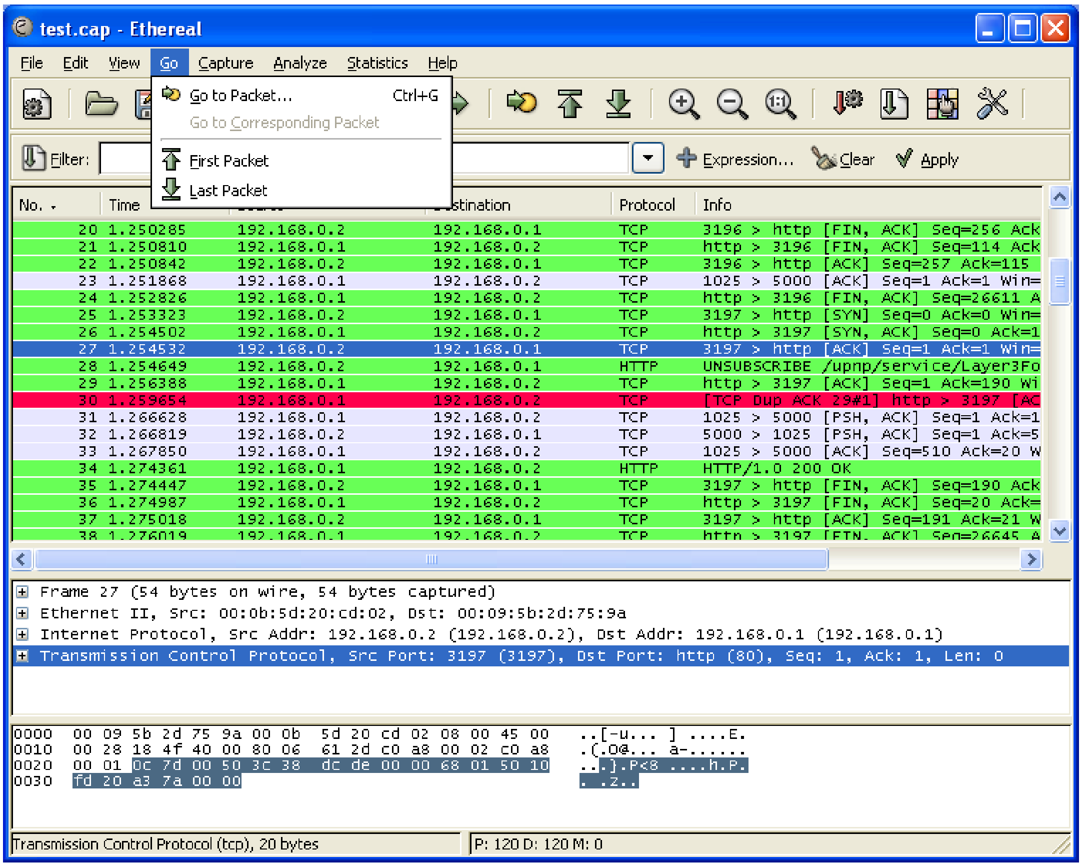
Coloring Rules...		This menu item brings up a dialog box that allows you to color packets in the packet list pane according to filter expressions you choose. It can be very useful for spotting certain types of packets, see Section 9.3, "Packet colorization".

Show Packet in New Window		This menu item brings up the selected packet in a separate window. The separate window shows only the tree view and byte view panes.
Reload	Ctrl-R	This menu item allows you to reload the current capture file.

3.8. The "Go" menu

The Ethereal Go menu contains the fields shown in Table 3.4, “Go menu items”.

Figure 3.6. The "Go" Menu



```
No. ▾  Time                               stination      Protocol  Info
   20 1.250285   192.168.0.2          192.168.0.1     TCP     3196 > http [FIN, ACK] Seq=256 Ack
   21 1.250810   192.168.0.1          192.168.0.2     TCP     http > 3196 [FIN, ACK] Seq=114 Ack
   22 1.250842   192.168.0.2          192.168.0.1     TCP     3196 > http [ACK] Seq=257 Ack=115
   23 1.251868   192.168.0.1          192.168.0.2     TCP     1025 > 5000 [ACK] Seq=1 Ack=1 Win=
   24 1.252826   192.168.0.1          192.168.0.2     TCP     http > 3196 [FIN, ACK] Seq=26611 A
   25 1.253323   192.168.0.2          192.168.0.1     TCP     3197 > http [SYN] Seq=0 Ack=0 Win=
   26 1.254502   192.168.0.1          192.168.0.2     TCP     http > 3197 [SYN, ACK] Seq=0 Ack=1
   27 1.254532   192.168.0.2          192.168.0.1     TCP     3197 > http [ACK] Seq=1 Ack=1 Win=
   28 1.254649   192.168.0.2          192.168.0.1     HTTP    UNSUBSCRIBE /upnp/service/Layer3Fo
   29 1.256388   192.168.0.1          192.168.0.2     TCP     http > 3197 [ACK] Seq=1 Ack=190 Wi
   30 1.259654   192.168.0.1          192.168.0.2     TCP     [TCP Dup ACK 29#1] http > 3197 [AC
   31 1.266628   192.168.0.1          192.168.0.2     TCP     1025 > 5000 [PSH, ACK] Seq=1 Ack=1
   32 1.266819   192.168.0.2          192.168.0.1     TCP     5000 > 1025 [PSH, ACK] Seq=1 Ack=5
   33 1.267850   192.168.0.1          192.168.0.2     TCP     1025 > 5000 [ACK] Seq=510 Ack=20 W
   34 1.274361   192.168.0.1          192.168.0.2     HTTP    HTTP/1.0 200 OK
   35 1.274447   192.168.0.2          192.168.0.1     TCP     3197 > http [FIN, ACK] Seq=190 Ack
   36 1.274987   192.168.0.1          192.168.0.2     TCP     http > 3197 [FIN, ACK] Seq=20 Ack=
   37 1.275018   192.168.0.2          192.168.0.1     TCP     3197 > http [ACK] Seq=191 Ack=21 W
   38 1.276019   192.168.0.1          192.168.0.2     TCP     http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

```
Transmission Control Protocol (tcp), 20 bytes      P: 120 D: 120 M: 0
```

Table 3.4. Go menu items

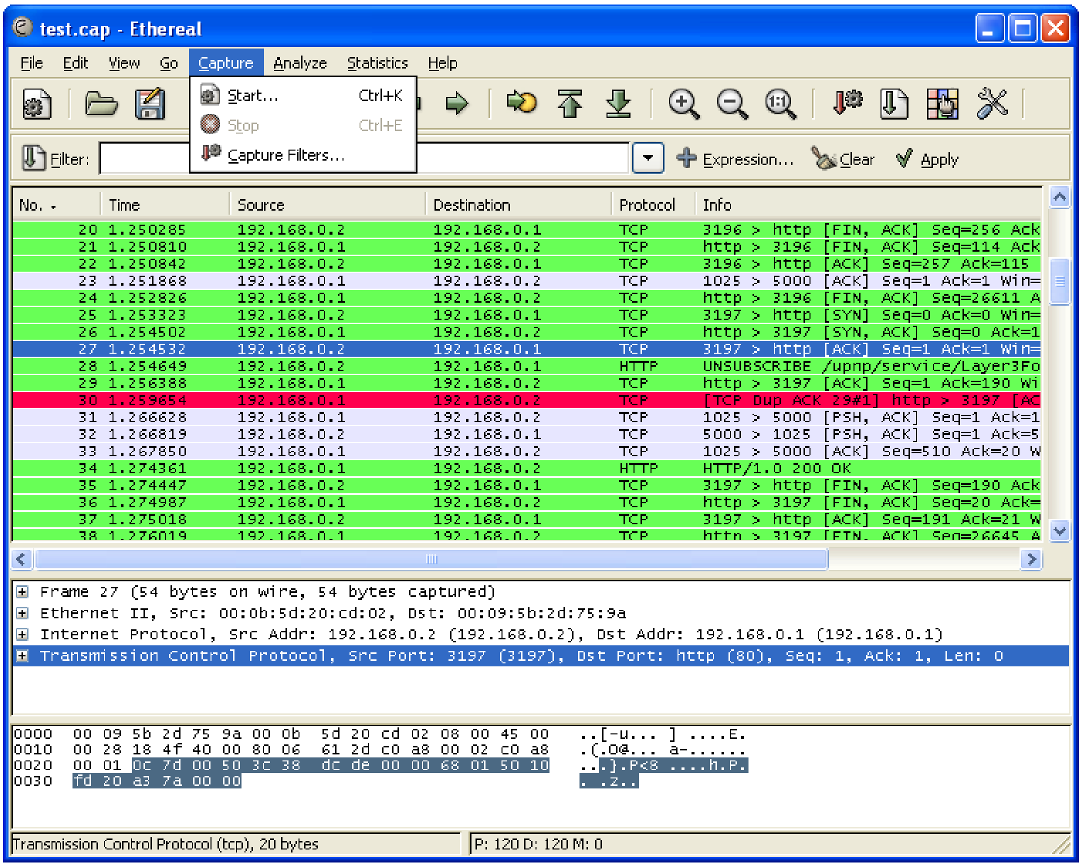
Menu Item	Accelerator	Description
Go to Packet...	Ctrl-G	This menu item brings up a dialog box that allows you to specify a packet number, and then goes to that packet. See Section 6.7, “Go to a specific packet” for details.
Go to Corresponding Packet		This menu item goes to the corresponding packet of the currently selected protocol field. If the selected field doesn't correspond to a packet, this item is greyed out.

First Packet		This menu item jumps to the first packet of the capture file.
Last Packet		This menu item jumps to the last packet of the capture file.

3.9. The "Capture" menu

The Ethereal Capture menu contains the fields shown in Table 3.5, “Capture menu items”.

Figure 3.7. The "Capture" Menu



```
No. .  Time      Source         Destination    Protocol  Info
    20 1.250285  192.168.0.2    192.168.0.1    TCP       3196 > http [FIN, ACK] Seq=256 Ack
    21 1.250810  192.168.0.1    192.168.0.2    TCP       http > 3196 [FIN, ACK] Seq=114 Ack
    22 1.250842  192.168.0.2    192.168.0.1    TCP       3196 > http [ACK] Seq=257 Ack=115
    23 1.251868  192.168.0.1    192.168.0.2    TCP       1025 > 5000 [ACK] Seq=1 Ack=1 Win=
    24 1.252826  192.168.0.1    192.168.0.2    TCP       http > 3196 [FIN, ACK] Seq=26611 A
    25 1.253323  192.168.0.2    192.168.0.1    TCP       3197 > http [SYN] Seq=0 Ack=0 Win=
    26 1.254502  192.168.0.1    192.168.0.2    TCP       http > 3197 [SYN, ACK] Seq=0 Ack=1
    27 1.254532  192.168.0.2    192.168.0.1    TCP       3197 > http [ACK] Seq=1 Ack=1 Win=
    28 1.254649  192.168.0.2    192.168.0.1    HTTP      UNSUBSCRIBE /upnp/service/Layer3Fo
    29 1.256388  192.168.0.1    192.168.0.2    TCP       http > 3197 [ACK] Seq=1 Ack=190 Wi
    30 1.259654  192.168.0.1    192.168.0.2    TCP       [TCP Dup ACK 29#1] http > 3197 [AC
    31 1.266628  192.168.0.1    192.168.0.2    TCP       1025 > 5000 [PSH, ACK] Seq=1 Ack=1
    32 1.266819  192.168.0.2    192.168.0.1    TCP       5000 > 1025 [PSH, ACK] Seq=1 Ack=5
    33 1.267850  192.168.0.1    192.168.0.2    TCP       1025 > 5000 [ACK] Seq=510 Ack=20 W
    34 1.274361  192.168.0.1    192.168.0.2    HTTP      HTTP/1.0 200 OK
    35 1.274447  192.168.0.2    192.168.0.1    TCP       3197 > http [FIN, ACK] Seq=190 Ack
    36 1.274987  192.168.0.1    192.168.0.2    TCP       http > 3197 [FIN, ACK] Seq=20 Ack=
    37 1.275018  192.168.0.2    192.168.0.1    TCP       3197 > http [ACK] Seq=191 Ack=21 W
    38 1.276019  192.168.0.1    192.168.0.2    TCP       http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

```
Transmission Control Protocol (tcp), 20 bytes    P: 120 D: 120 M: 0
```

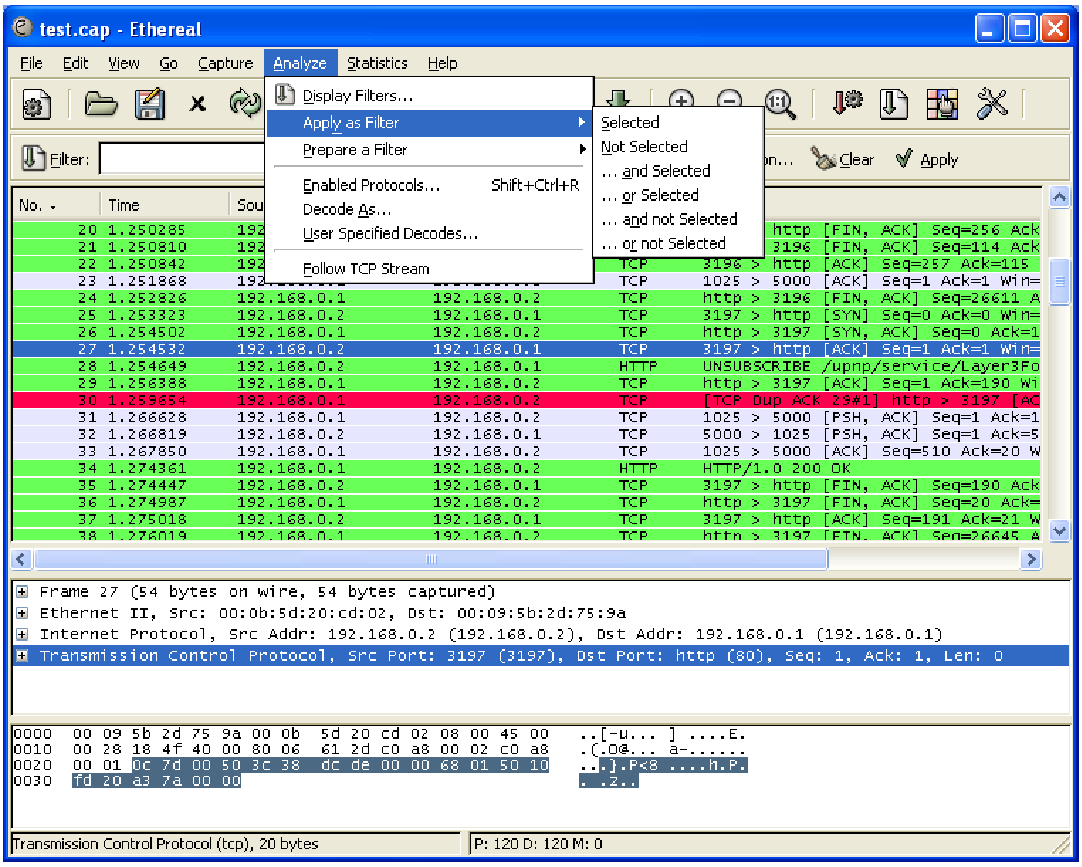
Table 3.5. Capture menu items

Menu Item	Accelerator	Description
Start...	Ctrl+K	This menu item brings up the Capture Options dialog box (discussed further in Section 4.2, “The "Capture Options" dialog box”) and allows you to start capturing packets.
Stop	Ctrl+E	This menu item stops the currently running capture, see Section 4.6.1, “Stop the running capture”) .
Capture Filters...		This menu item brings up a dialog box that allows you to create and edit capture filters. You can name filters, and you can save them for future use. More detail on this subject is provided in Section 6.5, “Defining and saving filters”

3.10. The "Analyze" menu

The Ethereal Analyze menu contains the fields shown in Table 3.6, “Analyze menu items”.

Figure 3.8. The "Analyze" Menu



```
   20 1.250285   192                                                http [FIN, ACK] Seq=256 Ack
   21 1.250810   192                                                3196 [FIN, ACK] Seq=114 Ack
   22 1.250842   192                                  TCP      3196 > http [ACK] Seq=257 Ack=115
   23 1.251868   192                                  TCP      1025 > 5000 [ACK] Seq=1 Ack=1 Win=
   24 1.252826   192.168.0.1         192.168.0.2      TCP      http > 3196 [FIN, ACK] Seq=26611 A
   25 1.253323   192.168.0.2         192.168.0.1      TCP      3197 > http [SYN] Seq=0 Ack=0 Win=
   26 1.254502   192.168.0.1         192.168.0.2      TCP      http > 3197 [SYN, ACK] Seq=0 Ack=1
   27 1.254532   192.168.0.2         192.168.0.1      TCP      3197 > http [ACK] Seq=1 Ack=1 Win=
   28 1.254649   192.168.0.2         192.168.0.1      HTTP     UNSUBSCRIBE /upnp/service/Layer3Fo
   29 1.256388   192.168.0.1         192.168.0.2      TCP      http > 3197 [ACK] Seq=1 Ack=190 Wi
   30 1.259654   192.168.0.1         192.168.0.2      TCP      [TCP Dup ACK 29#1] http > 3197 [AC
   31 1.266628   192.168.0.1         192.168.0.2      TCP      1025 > 5000 [PSH, ACK] Seq=1 Ack=1
   32 1.266819   192.168.0.2         192.168.0.1      TCP      5000 > 1025 [PSH, ACK] Seq=1 Ack=5
   33 1.267850   192.168.0.1         192.168.0.2      TCP      1025 > 5000 [ACK] Seq=510 Ack=20 W
   34 1.274361   192.168.0.1         192.168.0.2      HTTP     HTTP/1.0 200 OK
   35 1.274447   192.168.0.2         192.168.0.1      TCP      3197 > http [FIN, ACK] Seq=190 Ack
   36 1.274987   192.168.0.1         192.168.0.2      TCP      http > 3197 [FIN, ACK] Seq=20 Ack=
   37 1.275018   192.168.0.2         192.168.0.1      TCP      3197 > http [ACK] Seq=191 Ack=21 W
   38 1.276019   192.168.0.1         192.168.0.2      TCP      http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

Transmission Control Protocol (tcp), 20 bytes | P: 120 D: 120 M: 0

Table 3.6. 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 6.5, “Defining and saving filters”
Apply as Filter > ...		These menu items will change the current display filter and apply the changed filter immediately. Depending on the chosen menu item, the current display filter string will be replaced or appended to by the selected protocol field in the packet details pane.
Prepare a Filter > ...		These menu items will change the current display filter but won't apply the changed filter. Depending on the chosen menu item, the current display filter string will be replaced or appended to by the selected protocol field in the packet details pane.

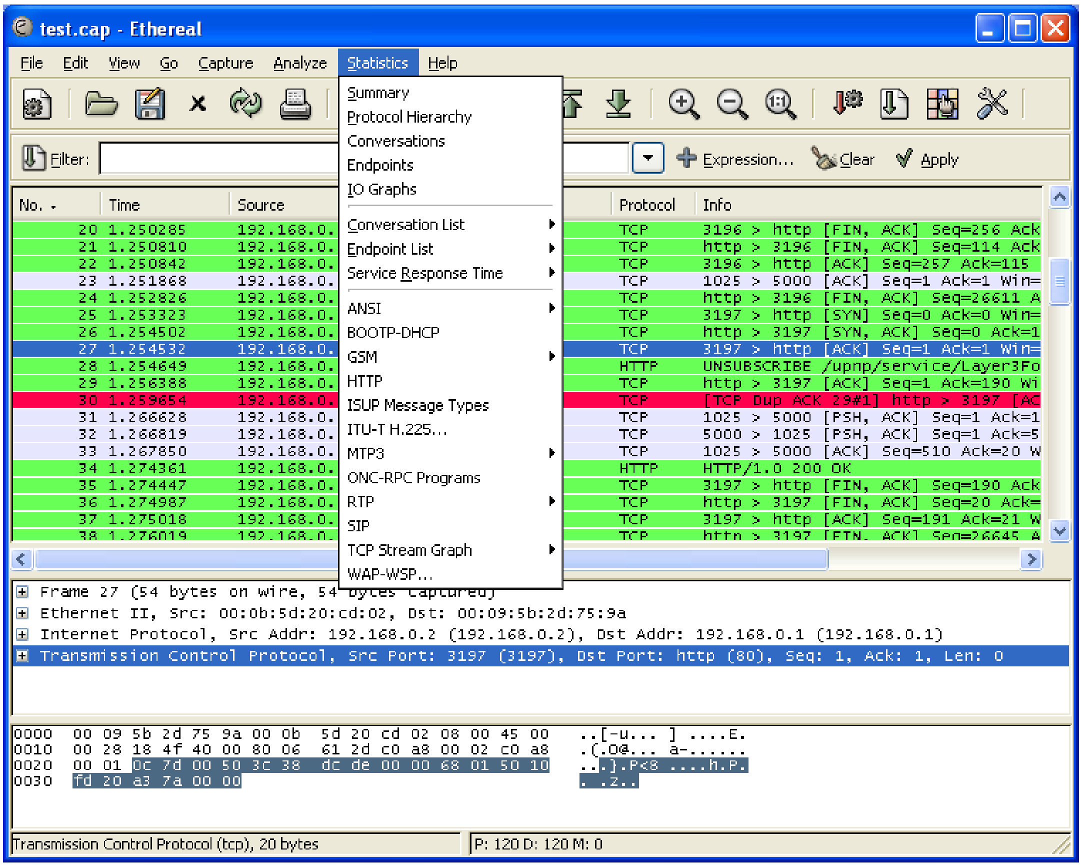
Menu Item	Accelerator	Description
Enabled Protocols...	Shift+Ctrl+R	This menu item allows the user to enable/disable protocol dissectors, see Section 9.4.1, “The "Enabled Protocols" dialog box”
Decode As...		This menu item allows the user to force Ethereal to decode certain packets as a particular protocol, see Section 9.4.2, “User Specified Decodes”
User Specified Decodes...		This menu item allows the user to force Ethereal to decode certain packets as a particular protocol, see Section 9.4.3, “Show User Specified Decodes”

Follow TCP Stream		This menu item brings up a separate window and displays all the TCP segments captured that are on the same TCP connection as a selected packet, see Section 7.2, “Following TCP streams”

3.11. The "Statistics" menu

The Ethereal Statistics menu contains the fields shown in Table 3.7, “Statistics menu items”.

Figure 3.9. The "Statistics" Menu



No.	Time	Source	Protocol	Info
20	1.250285	192.168.0.	TCP	3196 > http [FIN, ACK] Seq=256 Ack
21	1.250810	192.168.0.	TCP	http > 3196 [FIN, ACK] Seq=114 Ack
22	1.250842	192.168.0.	TCP	3196 > http [ACK] Seq=257 Ack=115
23	1.251868	192.168.0.	TCP	1025 > 5000 [ACK] Seq=1 Ack=1 Win=
24	1.252826	192.168.0.	TCP	http > 3196 [FIN, ACK] Seq=26611 A
25	1.253323	192.168.0.	TCP	3197 > http [SYN] Seq=0 Ack=0 Win=
26	1.254502	192.168.0.	TCP	http > 3197 [SYN, ACK] Seq=0 Ack=1
27	1.254532	192.168.0.	TCP	3197 > http [ACK] Seq=1 Ack=1 Win=
28	1.254649	192.168.0.	HTTP	UNSUBSCRIBE /upnp/service/Layer3Fo
29	1.256388	192.168.0.	TCP	http > 3197 [ACK] Seq=1 Ack=190 Wi
30	1.259654	192.168.0.	TCP	[TCP Dup ACK 29#1] http > 3197 [AC
31	1.266628	192.168.0.	TCP	1025 > 5000 [PSH, ACK] Seq=1 Ack=1
32	1.266819	192.168.0.	TCP	5000 > 1025 [PSH, ACK] Seq=1 Ack=5
33	1.267850	192.168.0.	TCP	1025 > 5000 [ACK] Seq=510 Ack=20 W
34	1.274361	192.168.0.	HTTP	HTTP/1.0 200 OK
35	1.274447	192.168.0.	TCP	3197 > http [FIN, ACK] Seq=190 Ack
36	1.274987	192.168.0.	TCP	http > 3197 [FIN, ACK] Seq=20 Ack=
37	1.275018	192.168.0.	TCP	3197 > http [ACK] Seq=191 Ack=21 W
38	1.276019	192.168.0.	TCP	http > 3197 [FIN, ACK] Seq=26645 A

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

Transmission Control Protocol (tcp), 20 bytes P: 120 D: 120 M: 0

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

Table 3.7. 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 8.5.2, “The "Conversations" window”.
Endpoints		Display a list of endpoints (traffic to/from an address), see Section 8.4.2, “The "Endpoints" window”.
IO Graphs		Display user specified graphs (e.g. the number of packets in the course of time), see Section 8.6, “The "IO Graphs" window”.

Menu Item	Accelerator	Description

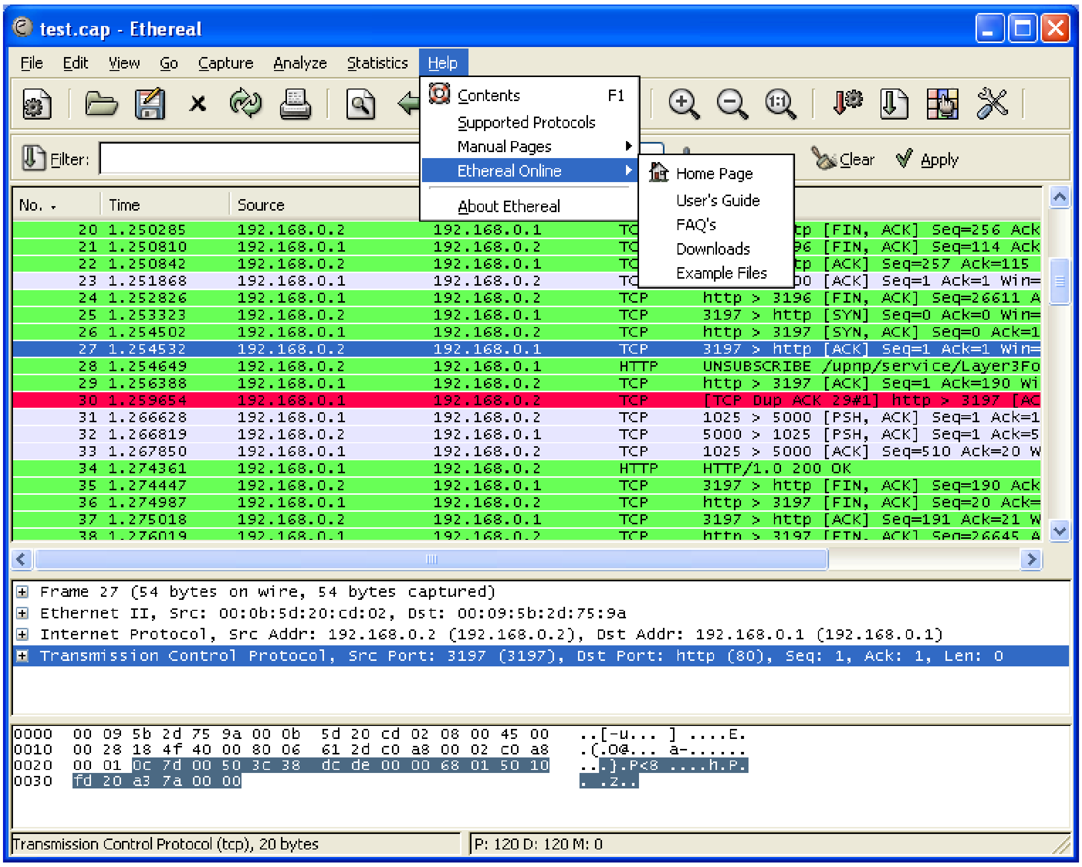
Conversation List		Display a list of conversations, obsoleted by the combined window of Conversations above, see Section 8.5.3, “The protocol specific "Conversation List" windows”.
Endpoint List		Display a list of endpoints, obsoleted by the combined window of Endpoints above, see Section 8.4.3, “The protocol specific "Endpoint List" windows”.
Service Response Time		Display the time between a request and the corresponding response, see Section 8.7, “Service Response Time”.

ANSI		See Section 8.8, “The protocol specific statistics windows”
BOOTP-DHCP		See Section 8.8, “The protocol specific statistics windows”
GSM		See Section 8.8, “The protocol specific statistics windows”
HTTP		HTTP request/response statistics, see Section 8.8, “The protocol specific statistics windows”
ISUP Message Types		See Section 8.8, “The protocol specific statistics windows”
ITU-T H.225		See Section 8.8, “The protocol specific statistics windows”
MTP3		See Section 8.8, “The protocol specific statistics windows”
ONC-RPC Programs		See Section 8.8, “The protocol specific statistics windows”
RTP		See Section 8.8, “The protocol specific statistics windows”
SIP		See Section 8.8, “The protocol specific statistics windows”
TCP Stream Graph		See Section 8.8, “The protocol specific statistics windows”
WAP-WSP		See Section 8.8, “The protocol specific statistics windows”

3.12. The "Help" menu

The Ethereal Help menu contains the fields shown in Table 3.8, “Help menu items”.

Figure 3.10. The "Help" Menu



```
No. -    Time          Source              
    20 1.250285      192.168.0.2          192.168.0.1          TC               tp [FIN, ACK] Seq=256 Ack
    21 1.250810      192.168.0.1          192.168.0.2          TC               96 [FIN, ACK] Seq=114 Ack
    22 1.250842      192.168.0.2          192.168.0.1          TC               tp [ACK] Seq=257 Ack=115
    23 1.251868      192.168.0.1          192.168.0.2          TC               00 [ACK] Seq=1 Ack=1 Win=
    24 1.252826      192.168.0.1          192.168.0.2          TCP       http > 3196 [FIN, ACK] Seq=26611 A
    25 1.253323      192.168.0.2          192.168.0.1          TCP       3197 > http [SYN] Seq=0 Ack=0 Win=
    26 1.254502      192.168.0.1          192.168.0.2          TCP       http > 3197 [SYN, ACK] Seq=0 Ack=1
    27 1.254532      192.168.0.2          192.168.0.1          TCP       3197 > http [ACK] Seq=1 Ack=1 Win=
    28 1.254649      192.168.0.2          192.168.0.1          HTTP      UNSUBSCRIBE /upnp/service/Layer3Fo
    29 1.256388      192.168.0.1          192.168.0.2          TCP       http > 3197 [ACK] Seq=1 Ack=190 Wi
    30 1.259654      192.168.0.1          192.168.0.2          TCP       [TCP Dup ACK 29#1] http > 3197 [AC
    31 1.266628      192.168.0.1          192.168.0.2          TCP       1025 > 5000 [PSH, ACK] Seq=1 Ack=1
    32 1.266819      192.168.0.2          192.168.0.1          TCP       5000 > 1025 [PSH, ACK] Seq=1 Ack=5
    33 1.267850      192.168.0.1          192.168.0.2          TCP       1025 > 5000 [ACK] Seq=510 Ack=20 W
    34 1.274361      192.168.0.1          192.168.0.2          HTTP      HTTP/1.0 200 OK
    35 1.274447      192.168.0.2          192.168.0.1          TCP       3197 > http [FIN, ACK] Seq=190 Ack
    36 1.274987      192.168.0.1          192.168.0.2          TCP       http > 3197 [FIN, ACK] Seq=20 Ack=
    37 1.275018      192.168.0.2          192.168.0.1          TCP       3197 > http [ACK] Seq=191 Ack=21 W
    38 1.276019      192.168.0.1          192.168.0.2          TCP       http > 3197 [FIN, ACK] Seq=26645 A
```

```
⊞ Frame 27 (54 bytes on wire, 54 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ Transmission Control Protocol, Src Port: 3197 (3197), Dst Port: http (80), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 28 18 4f 40 00 80 06  61 2d c0 a8 00 02 c0 a8   .(.O@... a-......
0020  00 01 0c 7d 00 50 3c 38  dc de 00 00 68 01 50 10   ...}.P<8 ....h.P.
0030  fd 20 a3 7a 00 00                                  . .z..
```

Transmission Control Protocol (tcp), 20 bytes | P: 120 D: 120 M: 0

Table 3.8. Help menu items

Menu Item	Accelerator	Description
Contents	F1	This menu item brings up a basic help system.
Supported Protocols		This menu item brings up a dialog box showing the supported protocols and protocol fields.
Manual Pages > ...		This menu item starts a Web browser showing one of the locally installed html manual pages.
Ethereal Online > ...		This menu item tries to start a Web browser showing a specific webpage from: http://www.ethereal.com.

About Ethereal		This menu item brings up an information window that provides some information on Ethereal, such as the plugins, the used folders, ...

3.13. The "Main" toolbar

The main toolbar provides quick access to frequently used items from the menu. This toolbar cannot be customized by the user, but it can be hidden using the View menu, if the space on the screen is needed to show even more packet data.

As in the menu, only the items useful in the current program state will be available. The others will be greyed out (e.g. you cannot save a capture file if you haven't loaded one).

Figure 3.11. The "Main" toolbar

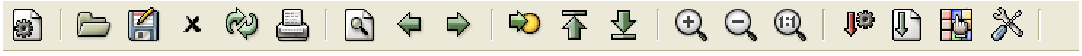


Table 3.9. Main toolbar items

Toolbar Icon	Toolbar Item	Corresponding Menu Item	Description
	Start Capture...	Capture/Start...	This item brings up the Capture Options dialog box (discussed further in Section 4.1, “Start Capturing”) and allows you to start capturing packets. **Note!** If a live capture is in progress, and you are using "Update List of Packets in Realtime", this icon will be replaced by the Stop Capture icon .
	Stop Capture	Capture/Stop	This item stops the currently running live capture process Section 4.1, “Start Capturing”). **Note!** This icon is shown if a live capture is in progress, and you are using "Update List of Packets in Realtime", otherwise the Start Capture icon is shown.

	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

Toolbar Icon	Toolbar Item	Corresponding Menu Item	Description
			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.
	Close	File/Close	This item closes the current capture. If you have not saved the capture, you will be asked to save it first.
	Reload	View/Reload	This item allows you to reload the current capture file.
	Print...	File/Print...	This item allows you to print all (or some of) the packets in the capture file. It pops up the Ethereal Print dialog box (which is discussed further in Section 5.6, “Printing packets”).

	Find Packet...	Edit/Find Packet...	This item brings up a dialog box that allows you to find a packet. There is further information on finding packets in Section 6.6, “Finding packets”.
	Find Previous	Edit/Find Previous	This item tries to find the previous packet, matching the settings from "Find Packet...".
	Find Next	Edit/Find Next	This item tries to find the next packet, matching the settings from "Find Packet...".

	Go to Packet...	Go/Go to Packet...	This item brings up a dialog box that allows you to specify a packet number to go to that packet.
	Go To First Packet	Go/First Packet	This item jumps to the first packet of the capture file.
	Go To Last Packet	Go/Last Packet	This item jumps to the last packet of the capture file.

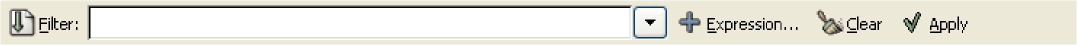
	Zoom In	View/Zoom In	Zoom into the packet data (increase the font size).
	Zoom Out	View/Zoom Out	Zoom out of the packet data (decrease the font size).
	Normal Size	View/Normal Size	Set zoom level back to 100%.

Toolbar Icon	Toolbar Item	Corresponding Menu Item	Description
	Capture Filters...	Capture/Capture Filters...	This item brings up a dialog box that allows you to create and edit capture filters. You can name filters, and you can save them for future use. More detail on this subject is provided in Section 6.5, “Defining and saving filters”.
	Display Filters...	Analyze/Display Filters...	This 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 6.5, “Defining and saving filters”.
	Coloring Rules...	View/Coloring Rules...	This item brings up a dialog box that allows you color packets in the packet list pane according to filter expressions you choose. It can be very useful for spotting certain types of packets. More detail on this subject is provided in Section 9.3, “Packet colorization”.
	Preferences...	Edit/Preferences	This item brings up a dialog box that allows you to set preferences for many parameters that control Ethereal. You can also save your preferences so Ethereal will use them the next time you start it. More detail is provided in Section 9.5, “Preferences”

3.14. The "Filter" toolbar

The filter toolbar lets you quickly edit and apply display filters. More information on display filters is available in Section 6.2, “Filtering packets while viewing”.

Figure 3.12. The "Filter" toolbar



- The leftmost button labeled "Filter:" can be clicked to bring up the filter construction dialog, described in Figure 6.8, “The "Capture Filters" and "Display Filters" dialog boxes”.
- The left middle text box provides an area to enter or edit display filter strings, see Section 6.3, “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.

- 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 6.4, “The "Filter Expression" dialog box”
- The right middle button labeled "Clear" resets the current display filter and clears the edit area.
- The rightmost button labeled "Apply" applies the current value in the edit area as the new display filter.

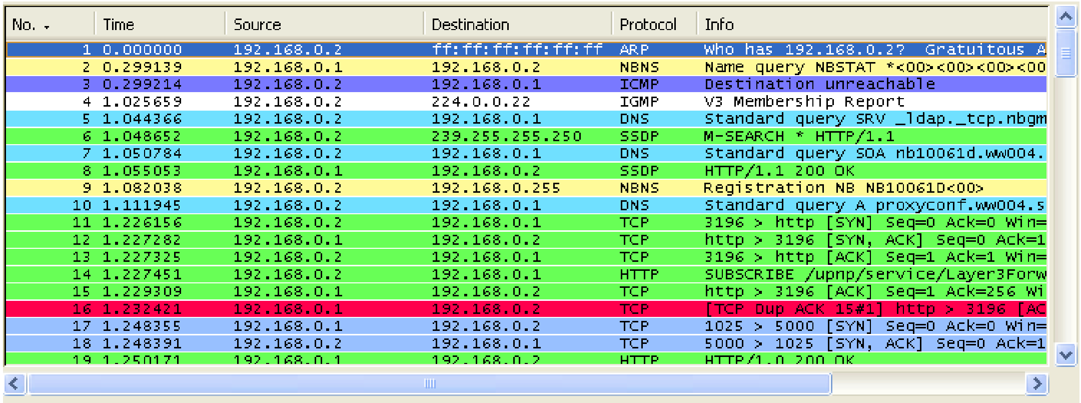
Note!

Applying a display filter on large capture files might take quite a long time!

3.15. The "Packet List" pane

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

Figure 3.13. The "Packet List" pane



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.2? Gratuitous A
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT *<00><00><00><00
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap._tcp.nbgm
6	1.048652	192.168.0.2	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb10061d.ww004.
8	1.055053	192.168.0.1	192.168.0.2	SSDP	HTTP/1.1 200 OK
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D<00>
10	1.111945	192.168.0.2	192.168.0.1	DNS	Standard query A proxyconf.ww004.s
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0 Ack=0 Win=
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] Seq=0 Ack=1
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1 Ack=1 Win=
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/Layer3Forw
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1 Ack=256 Wi
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Dup ACK 15#1] http > 3196 [AC
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0 Ack=0 Win=
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] Seq=0 Ack=1
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK

Each line in the packet list corrresponds 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, Ethereal 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 Figure 9.10, “The "User Interface: Columns" preferences page”.

The default columns will show:

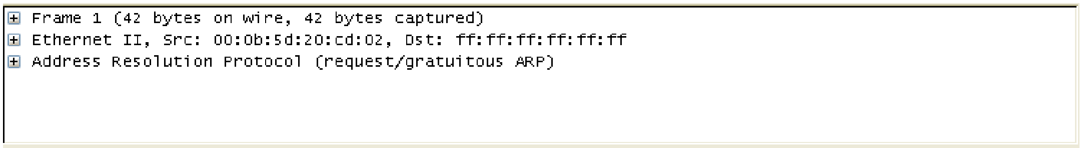
- **No.** The number of the packet in the capture file. This number won't change, even if a display filter is used.
- **Time** The timestamp of the packet. The presentation format of this timestamp can be changed, see Section 6.9, “Time display formats and time references”.
- **Source** The address where this packet is coming from.
- **Destination** The address where this packet is going to.
- **Protocol** The protocol name in a short (perhaps abbreviated) version.
- **Info** Additional information about the packet content.

There is a context menu (right mouse click) available, see details in Figure 6.3, “Pop-up menu of "Packet List" pane”.

3.16. The "Packet Details" pane

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

Figure 3.14. The "Packet Details" pane



```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request/gratuitous ARP)
```

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 6.4, “Pop-up menu of "Packet Details" pane”.

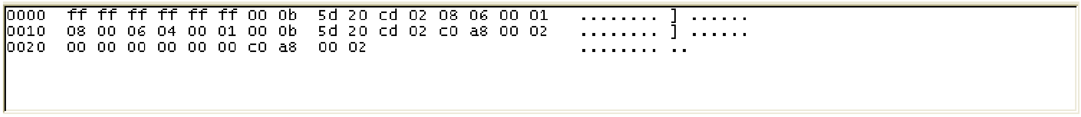
Some protocol fields are specially displayed.

- **Generated fields** Ethereal 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, Ethereal 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 Ethereal 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, Ethereal jumps to the corresponding packet.

3.17. 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 3.15. The "Packet Bytes" pane



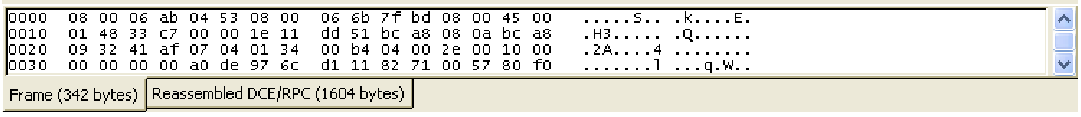
```
0000  ff ff ff ff ff ff 00 0b  5d 20 cd 02 08 06 00 01   ........ ] ......
0010  08 00 06 04 00 01 00 0b  5d 20 cd 02 c0 a8 00 02   ........ ] ......
0020  00 00 00 00 00 00 c0 a8  00 02                     ........ ..
```

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.

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

Depending on the packet data, sometimes more than one page is available, e.g. when Ethereal has reassembled some packets into a single chunk of data, see Section 7.3, “Packet Reassembling/De-segmenting”. 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 3.16. The "Packet Bytes" pane with tabs



```
0000  08 00 06 ab 04 53 08 00  06 6b 7f bd 08 00 45 00   .....S.. .k....E.
0010  01 48 33 c7 00 00 1e 11  dd 51 bc a8 08 0a bc a8   .H3..... .Q......
0020  09 32 41 af 07 04 01 34  00 b4 04 00 2e 00 10 00   .2A....4 ........
0030  00 00 00 00 a0 de 97 6c  d1 11 82 71 00 57 80 f0   .......l ...q.W..
```

Note!

The additional pages might contain data picked from multiple packets.

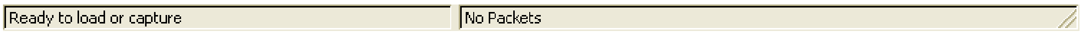
The context menu (right mouse click) of the tab labels will show a list of all available pages. This can be helpful if the size in the pane is too small for all the tab labels.

3.18. The Statusbar

The statusbar displays informational messages.

In general, the left side will show context related information, while the right side will show the current number of packets.

Figure 3.17. The initial Statusbar

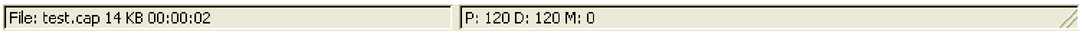


Ready to load or capture

No Packets

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

Figure 3.18. The Statusbar with a loaded capture file



File: test.cap 14 KB 00:00:02

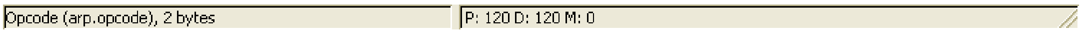
P: 120 D: 120 M: 0

The left side shows information about the capture file, its name, its size and the elapsed time while it was being captured.

The right side shows the current number of packets in the capture file. The following values are displayed:

- *P:* the number of captured packets
- *D:* the number of packets currently being displayed
- *M:* the number of marked packets

Figure 3.19. The Statusbar with a selected protocol field



Opcode (arp.opcode), 2 bytes

P: 120 D: 120 M: 0

This is displayed if you have selected a protocol field from the "Packet Details" pane.

Tip!

The value between the brackets (in this example **arp.opcode**) can be used as a display filter string, representing the selected protocol field.

Chapter 4. Capturing Live Network Data

4.1. Start Capturing

There are two methods you can use to start capturing packets with Ethereal:

1. From the command line using the following:

   ```
   ethereal -i eth0 -k
   ```

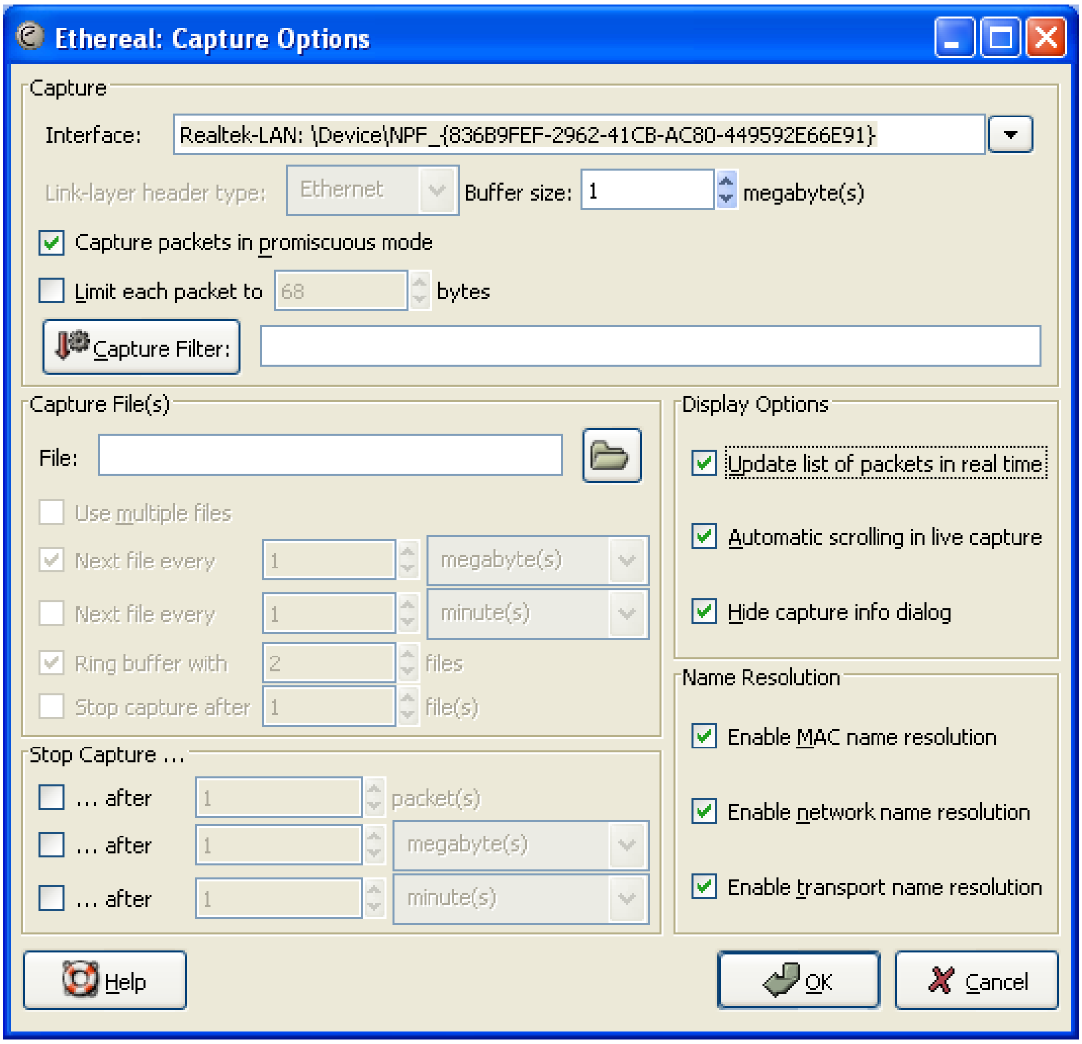
   This will start Ethereal capturing on interface eth0.

2. By starting Ethereal and then selecting Start... from the Capture menu (or use the corresponding item in the "Main" toolbar), this brings up the Capture Options dialog box.

4.2. The "Capture Options" dialog box

When you select Start... from the Capture menu, Ethereal pops up the "Capture Options" dialog box as shown in Figure 4.1, “The "Capture Options" dialog box”.

Figure 4.1. The "Capture Options" dialog box



Ethereal: Capture Options

Capture

Interface: Realtek-LAN: \Device\NPF_{836B9FEF-2962-41CB-AC80-449592E66E91}

Link-layer header type: Ethernet

Buffer size: 1 megabyte(s)

☑ Capture packets in promiscuous mode

□ Limit each packet to 68 bytes

Capture Filter:

Capture File(s)

File:

□ Use multiple files

☑ Next file every 1 megabyte(s)

□ Next file every 1 minute(s)

☑ Ring buffer with 2 files

□ Stop capture after 1 file(s)

Stop Capture ...

□ ... after 1 packet(s)

□ ... after 1 megabyte(s)

□ ... after 1 minute(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

Tip!

If you are unsure which options to choose in this dialog box, just try keeping the defaults as this should work well in many cases.

You can set the following fields in this dialog box:

4.2.1. Capture frame

Interface

This field specifies the interface you want to capture on. You can only capture on one interface, and you can only capture on interfaces that Ethereal has found on the system. It is a drop-down list, so simply click on the button on the right hand side and select the interface you want. It defaults to the first non-loopback interface that supports capturing, and if there are none, the first loopback interface. On some systems, loopback interfaces cannot be used for capturing (loopback interfaces are not available on Windows platforms).

This field performs the same function as the **-i <interface>** command line option.

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 4.4, “Link-layer header type”

Buffer size: n megabyte(s)

Enter the buffer size to be used while capturing. This is the size of the kernel buffer which will keep the captured packets, until they are written to disk. If you encounter packet drops, try increasing this value.

Note

This option is only available on Windows platforms.

Capture packets in promiscuous mode

This checkbox allows you to specify that Ethereal should put the interface in promiscuous mode when capturing. If you do not specify this, Ethereal will only capture the packets going to or from your computer (not all packets on your LAN segment).

Note

If some other process has put the interface in promiscuous mode you may be capturing in promiscuous mode even if you turn off this option

Note

Even in promiscuous mode you still won't necessarily see all packets on your LAN segment, see http://www.ethereal.com/faq#promiscsniff for some more explanations.

Limit each packet to n bytes

This field allows you to specify the maximum amount of data that will be captured for each packet, and is sometimes referred to as the **snaplen**. If disabled, the default is 65535, which will be sufficient for most protocols. Some rules of thumb:

- If you are unsure, just keep the default value.
- If you don't need all of the data in a packet - for example, if you only need the link-layer, IP, and TCP headers - you might want to choose a small snapshot length, as less CPU time is required for copying packets, less buffer space is required for packets, and thus perhaps fewer packets will be dropped if traffic is very heavy.

- If you don't capture all of the data in a packet, you might find that the packet data you want is in the part that's dropped, or that reassembly isn't possible as the data required for reassembly is missing.

Capture Filter

This field allows you to specify a capture filter. Capture filters are discussed in more details in Section 4.5, “Filtering while capturing”. It defaults to empty, or no filter.

You can also click on the button labelled Capture Filter, and Ethereal will bring up the Capture Filters dialog box and allow you to create and/or select a filter. Please see Section 6.5, “Defining and saving filters”

4.2.2. Capture File(s) frame

An explanation about capture file usage can be found in Section 4.3, “Capture files and file modes”.

File

This field allows you to specify the file name that will be used for the capture file. This field is left blank by default. If the field is left blank, the capture data will be stored in a temporary file, see Section 4.3, “Capture files and file modes” for details.

You can also click on the button to the right of this field to browse through the filesystem.

Use multiple files

Instead of using a single file, Ethereal will automatically switch to a new one, if a specific trigger condition is reached.

Next file every n megabyte(s)

Multiple files only: Switch to the next file after the given number of byte(s)/kilobyte(s)/megabyte(s)/gigabyte(s) have been captured.

Next file every n minute(s)

Multiple files only: Switch to the next file after the given number of second(s)/minutes(s)/hours(s)/days(s) have elapsed.

Ring buffer with n files

Multiple files only: Form a ring buffer of the capture files, with the given number of files.

Stop capture after n file(s)

Multiple files only: Stop capturing after switching to the next file the given number of times.

4.2.3. Stop Capture... frame

... after n packet(s)

Stop capturing after the given number of packets have been captured.

... after n megabytes(s)

Stop capturing after the given number of byte(s)/kilobyte(s)/megabyte(s)/gigabyte(s) have been captured. This option is greyed out, if "Use multiple files" is selected.

... after n minute(s)

Stop capturing after the given number of second(s)/minutes(s)/hours(s)/days(s) have elapsed.

4.2.4. Display Options frame

Update list of packets in real time This option allows you to specify that Ethereal should update the packet list pane in real time. If you do not specify this, Ethereal does not display any packets until you stop the capture. When you check this, Ethereal captures in a separate process and feeds the captures to the display process.

> **Note**
>
> If this option is checked, it will disable the "Use multiple files" option.

Automatic scrolling in live capture This option allows you to specify that Ethereal should scroll the packet list pane as new packets come in, so you are always looking at the last packet. If you do not specify this, Ethereal simply adds new packets onto the end of the list, but does not scroll the packet list pane. This option is greyed out if "Update list of packets in real time" is disabled.

Hide capture info dialog If this option is checked, the following capture info dialog will be hidden. This option is greyed out, if "Update list of packets in real time" is disabled.

4.2.5. Name Resolution frame

Enable MAC name resolution This option allows you to control whether or not Ethereal translates MAC addresses into names, see Section 7.4, “Name Resolution”.

Enable network name resolution This option allows you to control whether or not Ethereal translates network addresses into names, see Section 7.4, “Name Resolution”.

Enable transport name resolution This option allows you to control whether or not Ethereal translates transport addresses into protocols, see Section 7.4, “Name Resolution”.

4.2.6. Buttons

Once you have set the values you desire and have selected the options you need, simply click on **OK** to commence the capture, or **Cancel** to cancel the capture.

If you start a capture, Ethereal pops up a dialog box that shows you the progress of the capture and allows you to stop capturing when you have enough packets captured, see Section 4.6, “Running Capture”.

4.3. Capture files and file modes

While capturing, the underlying libpcap capturing engine will grab the packets from the network card and keep the packet data in a (relatively) small kernel buffer. This data is read by Ethereal and saved into the capture file(s) the user specified.

Different modes of operation are available when saving this packet data to the capture file(s).

Tip!

Working with large files (several 100 MB's) can be quite slow. If you plan to do a long term capture or capturing from a high traffic network, think about using one of the "Multiple files" options. This will spread the captured packets over several smaller files which can be much more pleasant to work with.

Note!

Using Multiple files may cut context related information. Ethereal keeps context information of the loaded packet data, so it can report context related problems (like a stream error) and keeps information about context related protocols (e.g. where data is exchanged at the establishing phase and only referred to in later packets). As it keeps this information only for the loaded file, using one of the multiple file modes may cut these contexts, If the establishing phase is saved in one file and the things you would like to see is in another, you might not see some of the valuable context related information.

Tip!

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

Table 4.1. Capture file mode selected by capture options

Mode	"File" option	"Use multiple files" option	"Ring buffer with n files" option
Single temporary file	-	-	-
Single named file	foo.cap	-	-
Multiple files, continuous	foo.cap	x	-
Multiple files, ring buffer	foo.cap	x	x

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 condi-

tions (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.

4.4. Link-layer header type

In the usual case, you won't have to choose this link-layer header type. The following paragraphs describe the exceptional cases, where selecting this type is possible, so you will have a guide what to do:

If you are capturing on an 802.11 device on some versions of BSD, this might offer a choice of "Ethernet" or "802.11". "Ethernet" will cause the captured packets to have fake Ethernet headers; "802.11" will cause them to have IEEE 802.11 headers. Unless the capture needs to be read by an application that doesn't support 802.11 headers, you should select "802.11".

If you are capturing on an Endace DAG card connected to a synchronous serial line, this might offer a choice of "PPP over serial" or "Cisco HDLC"; if the protocol on the serial line is PPP, select "PPP over serial", and if the protocol on the serial line is Cisco HDLC, select "Cisco HDLC".

If you are capturing on an Endace DAG card connected to an ATM network, this might offer a choice of "RFC 1483 IP-over-ATM" or "Sun raw ATM". If the only traffic being captured is RFC 1483 LLC-encapsulated IP, or if the capture needs to be read by an application that doesn't support SunATM headers, select "RFC 1483 IP-over-ATM", otherwise select "Sun raw ATM".

If you are capturing on an Ethernet device, this might offer a choice of "Ethernet" or "DOCSIS". If you are capturing traffic from a Cisco Cable Modem Termination System that is putting DOCSIS traffic onto the Ethernet to be captured, select "DOCSIS", otherwise select "Ethernet".

4.5. Filtering while capturing

Ethereal uses the libpcap filter language for capture filters. This is explained in the tcpdump man page, which can be hard to understand, so it's explained here to some extent.

You enter the capture filter into the Filter field of the Ethereal Capture Options dialog box, as shown in Figure 4.1, “The "Capture Options" dialog box”. The following is an outline of the syntax of the **tcpdump** capture filter language.

A capture filter takes the form of a series of primitive expressions connected by conjuctions (**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 than captures traffic to and from a particular host ”.

Example 4.1. A capture filter for telnet than 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 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 source or destination addresses. If these are not present, packets where the specified address appears as either the source or the destination address will be selected.

ether [src|dst] host <ehost>

This primitive allows you to filter on Ethernet host addresses. You can optionally include the keyword **src|dst** between the keywords **ether** and **host** to specify that you are only interested in source or destination addresses. If these are not present, packets where the specified address appears in either the source or destination address will be selected.

gateway host <host>

This primitive allows you to filter on packets that used **host** as a gateway. That is, where the Ethernet source or destination was **host** but neither the source nor destination IP address was **host**.

[src|dst] net <net> [{mask <mask>}|{len <len>}]

This primitive allows you to filter on network numbers. You can optionally precede this primitive with the keyword **src|dst** to specify that you are only interested in a source or destination network. If neither of these are present, packets will be selected that have the specified network in either the source or destination address. In addition, you can specify either the netmask or the CIDR prefix for the network if they are different from your own.

[tcp|udp] [src|dst] port <port>

This primitive allows you to filter on TCP and UDP port numbers. You can optionally precede this primitive with the keywords **src|dst** and **tcp|udp** which allow you to specify that you are only interested in source or destination ports and TCP or UDP packets respectively. The keywords **tcp|udp** must appear before **src|dst**.

If these are not specified, packets will be selected for both the TCP and UDP protocols and when the specified address appears in either the source or destination port field.

less|greater <length>

This primitive allows you to filter on packets whose length was less than or equal to the specified length, or greater than or equal to the specified length, respectively.

ip|ether proto <protocol>

This primitive allows you to filter on the specified protocol at either the Ethernet layer or the IP layer.

ether|ip broadcast|multicast

This primitive allows you to filter on either Ethernet or IP broadcasts or multicasts.

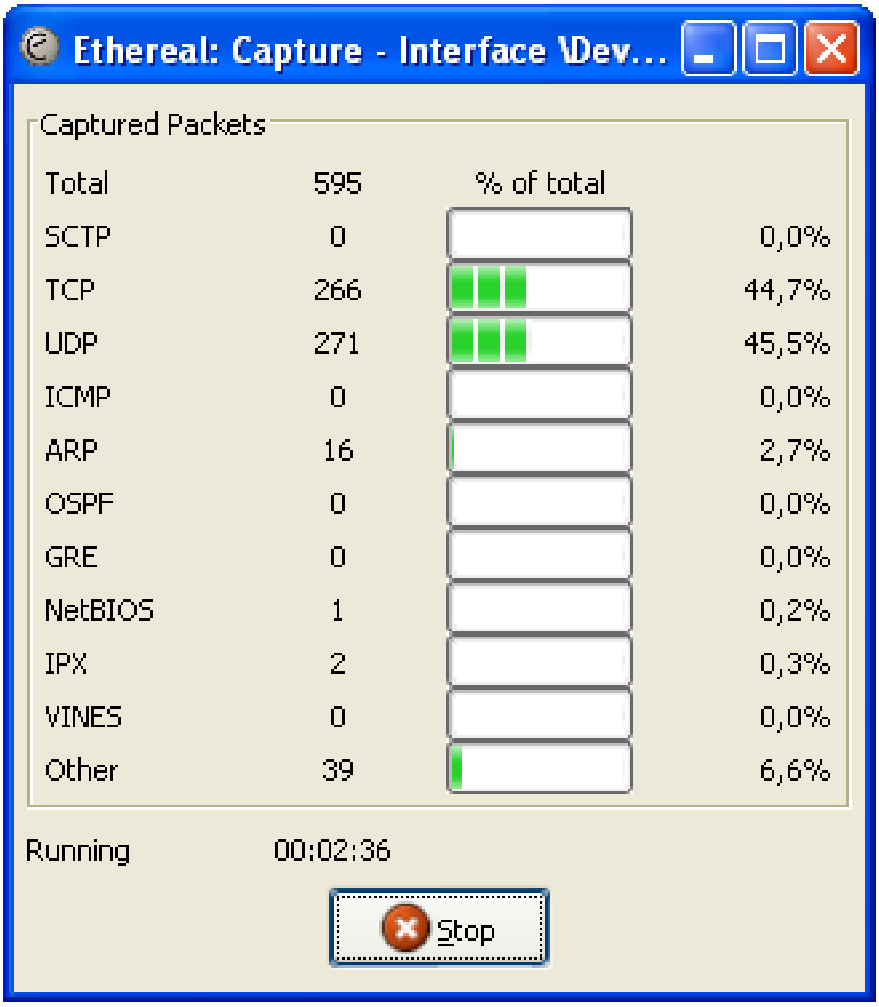
<expr> relop <expr>

This primitive allows you to create complex filter expressions that select bytes or ranges of bytes in packets. Please see the tcpdump man pages for more details.

4.6. Running Capture

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

Figure 4.2. The "Capture Info" dialog box



Captured Packets

Total	595	% of total	
SCTP	0		0,0%
TCP	266		44,7%
UDP	271		45,5%
ICMP	0		0,0%
ARP	16		2,7%
OSPF	0		0,0%
GRE	0		0,0%
NetBIOS	1		0,2%
IPX	2		0,3%
VINES	0		0,0%
Other	39		6,6%

Running 00:02:36

This dialog box will inform you about the number of captured packets and the time since the capture was started. The selection 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.

4.6.1. 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** .

Note!

The Capture Info dialog box might be hidden, if the option "Hide capture info dialog" is used.

2. Using the **menu item** "Capture/Stop Capture" or the corresponding Stop Capture **toolbar icon** ✖.

   ### Note!

   These are only available, if the option "Update list of packets in real time" is used.

3. Pressing the accelerator keys: **Ctrl+E**.

4. The capture will be automatically stopped, if one of the **Stop Conditions** is exceeded, e.g. the maximum amount of data was captured.

Chapter 5. File Input / Output and Printing

5.1. Introduction

This chapter will describe input and output of capture data.

- Open/Import capture files in various capture file formats
- Save/Export capture files in various capture file formats
- Merge capture files together
- Print packets

5.2. Open capture files

Ethereal can read in previously saved capture files. To read them, simply select the **Open** menu item from the **File** menu. Ethereal 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”.

Note!

You can also use **drag-and-drop** to open a file, by simply dropping the desired file from your file manager onto Ethereal's main window. However, drag-and-drop is not available/won't work in all desktop environments.

If you didn't save the current capture file before, 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), Ethereal can read capture files from a large number of other packet capture programs as well. See Section 5.2.2, “Input File Formats” for the list of capture formats Ethereal understands.

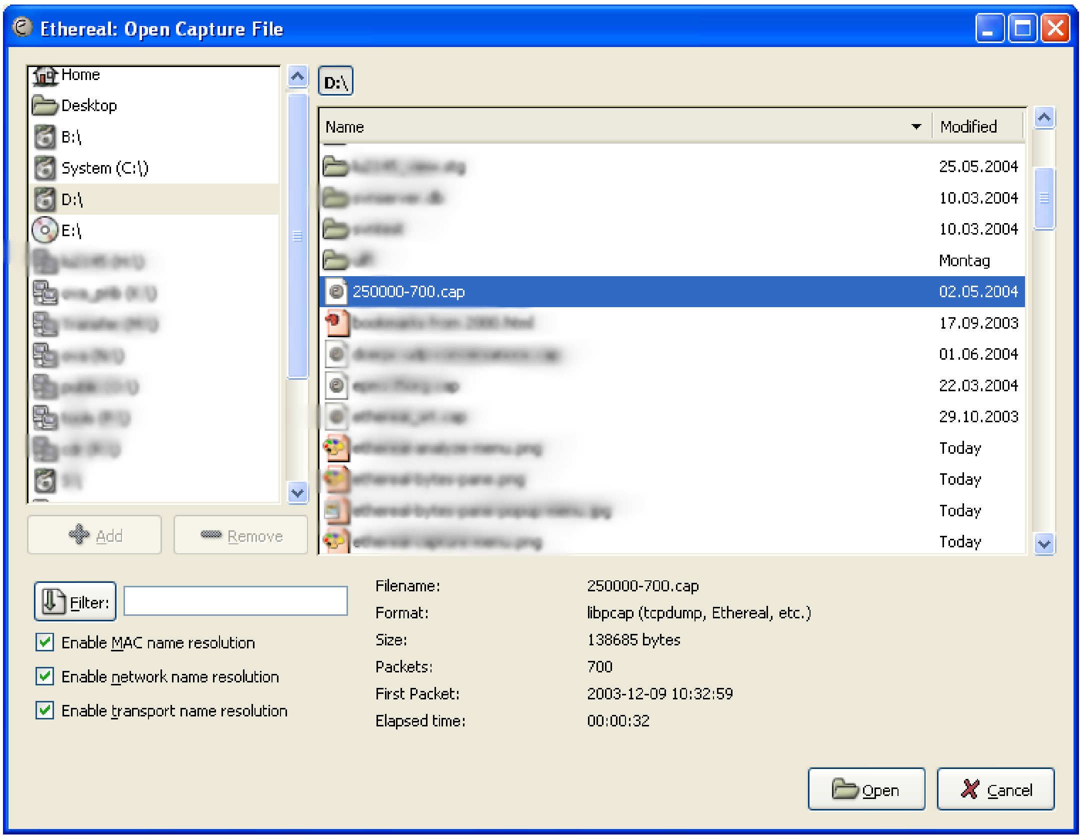
5.2.1. The "Open Capture File" dialog box

The "Open Capture File" dialog box allows you to search for a capture file containing previously captured packets for display in Ethereal. Figure 5.1, “The "Open Capture File" Dialog box” shows an example of the Ethereal Open File Dialog box.

Note

Ethereal uses the open dialog box from the version of the GTK+ toolkit that it's using. This dialog was completely redesigned in GTK version 2.4. Depending on the installed GTK version, your dialog box might look different. However, as the functionality remains almost the same, much of this description will work with your version of Ethereal.

Figure 5.1. The "Open Capture File" Dialog box



☑ Enable MAC name resolution
☑ Enable network name resolution
☑ Enable transport name resolution

Filename:	250000-700.cap
Format:	libpcap (tcpdump, Ethereal, etc.)
Size:	138685 bytes
Packets:	700
First Packet:	2003-12-09 10:32:59
Elapsed time:	00:00:32

With this dialog box, you can perform the following actions:

1. The "+ Add" button allows you to add a directory, selected in the right-hand pane, to the favorites (bookmarks?) list. Those changes are persistent.

2. The "- Remove" button allows you to remove a selected directory from that list again (the items like: "Home", "Desktop", and "Filesystem" cannot be removed).

3. Select files and directories with the list boxes.

4. View file preview information (like the filesize, the number of packets, ...), while browsing the filesystem.

5. Specify a display filter with the Filter button and filter field. This filter will be used when opening the new file. Clicking on the Filter button causes Ethereal to pop up the Filters dialog box (which is discussed further in Section 6.2, “Filtering packets while viewing”).

6. Specify which name resolution is to be performed for all packets by clicking on one of the "Enable name resolution" check buttons. Details about name resolution can be found in Section 7.4, “Name Resolution”.

7. Click the Open button to accept your selected file and open it. If Ethereal doesn't recognize the capture format, it will grey out this button.

8. Click the Cancel button to go back to Ethereal and not load a capture file.

You can change the display filter and name resolution settings later while viewing the packets. However, for very large capture files it can take a significant amount of time changing these settings, so it might be a good idea to set them in advance here.

5.2.2. Input File Formats

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

- libpcap, tcpdump and various other tools using tcpdump's capture 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 unconpressed) 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)
- the output from VMS's TCPIPtrace/TCPtrace/UCX$TRACE utilities
- the text output from the DBS Etherwatch VMS utility
- Visual Networks' Visual UpTime traffic capture
- the output from CoSine L2 debug
- the output from Accellent's 5Views LAN agents
- Endace Measurement Systems' ERF format captures
- Linux Bluez Bluetooth stack hcidump -w traces

Note!

It may not be possible to read some formats dependent on the packet types captured. Ethernet captures are usually supported for most file formats, but other packet types (e.g. token ring packets) may not be possible to read from all file formats.

5.3. Saving captured packets

You can save captured packets simply by using the Save As... menu item from the File menu under Ethereal. You can choose which packets to save and which file format to be used.

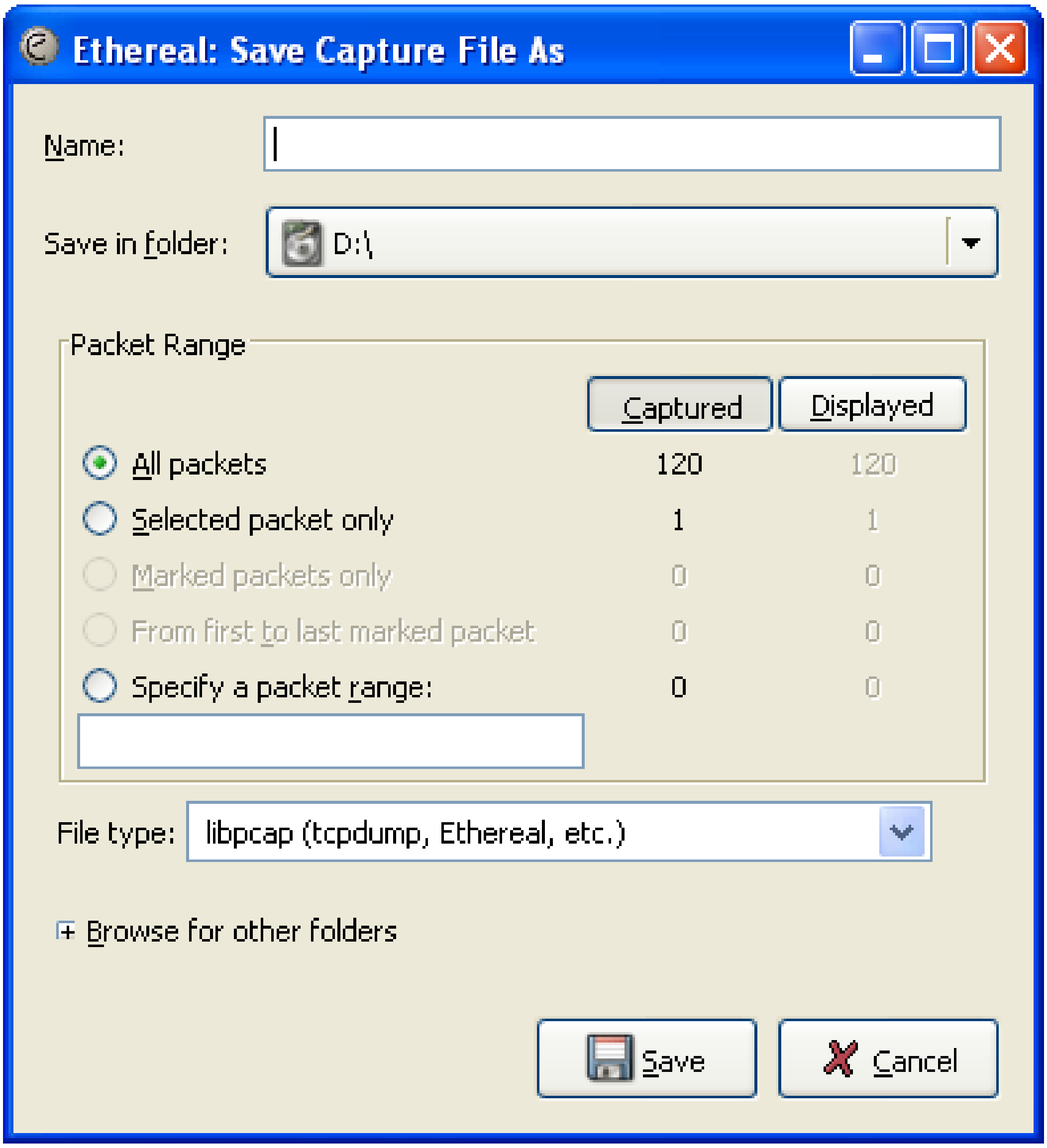
5.3.1. The "Save Capture File As" dialog box

The "Save Capture File As" dialog box allows you to save the current capture to a file. Figure 5.2, “The "Save Capture File As" dialog box” shows an example of this dialog box.

Note

Ethereal uses the open dialog box from the version of the GTK+ toolkit that it's using. This dialog was completely redesigned in the GTK version 2.4. Depending on the installed GTK version, your dialog box might look different. However, as the functionality remains almost the same, much of this description will work with your version of Ethereal.

Figure 5.2. The "Save Capture File As" dialog box



Name:

Save in folder: D:\

Packet Range

	Captured	Displayed
● All packets	120	120
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

File type: libpcap (tcpdump, Ethereal, etc.)

With this dialog box, you can perform the following actions:

1. Type in the name of the file you wish to save the captured packets in, as a standard file name in your file system.

2. Select the directory to save the file into.

3. Select the range of the packets to be saved, see Section 5.7, “The Packet Range frame”

4. Specify the format of the saved capture file by clicking on the File type drop down box. You can choose from the types, described in Section 5.2.2, “Input File Formats”.

Note!

Some capture formats may not be available, depending on the packet types captured.

Tip!

You can convert capture files from one format to another by reading in a capture file and writing it out using a different format.

5. Use "Browse for other folders" to browse files and folders in your file system.

6. Click on the Save button to accept your selected file and save to it. If Ethereal has a problem saving the captured packets to the file you specified, it will display an error dialog box. After clicking OK on this error dialog box, you can try again.

7. Click on the Cancel button to go back to Ethereal and not save the captured packets.

5.3.2. Output File Formats

The following file formats can be saved by Ethereal, so other capture tools can read the capture data from:

- libpcap (tcpdump)
- Novell LANalyzer
- Network Associates Sniffer
- Sun snoop
- Microsoft Network Monitor
- Visual Networks Visual UpTime traffic
- Accellent 5Views
- Networks Instruments Observer version 9

Other protocol analyzers may require that the file has a certain suffix in order to read the files you generate with Ethereal, e.g.:

".DMP" for Tcpdump/libpcap

".CAP" for Network Assosciates Sniffer Windows

5.4. Merging capture files

Sometimes you need to merge several capture files into one. For example this can be useful, if you have captured simultaneously from multiple interfaces at once (e.g. using multiple instances of Ethereal).

Merging capture files can be done in three ways:

- Use the **menu item "Merge"** from the "File" menu, to open the merge dialog, see Figure 5.3, “The "Merge with Capture File" dialog box”.
- Use **drag-and-drop** to drop multiple files on the main window. Ethereal will try to merge the packets in chronological order from the dropped files into a newly created temporary file.
- Use the **mergecap** tool, which is a command line tool to merge capture files. This tool provides the most options to merge capture files, see Section C.5, “mergecap: Merging multiple capture files into one with mergecap ”.

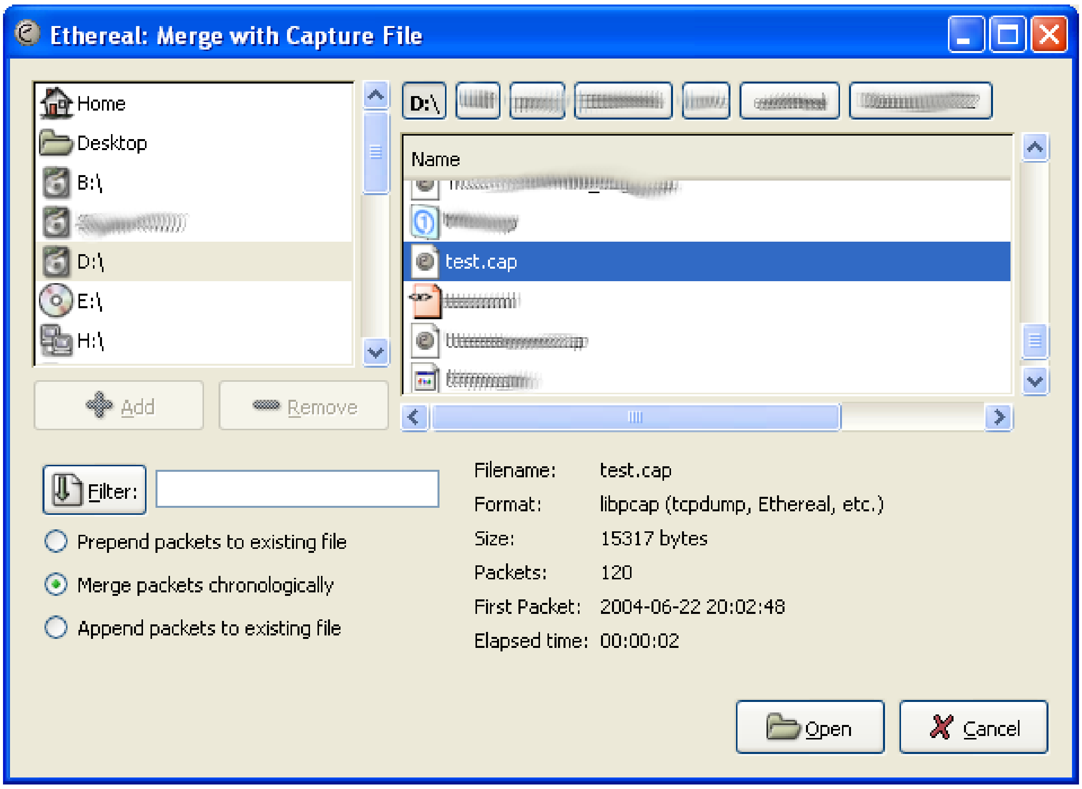
5.4.1. The "Merge with Capture File" dialog box

This dialog box let you select a file to be merged into the currently loaded file.

Note!

If your current data wasn't saved before, you will be asked to save it first, before this dialog box is shown.

Figure 5.3. The "Merge with Capture File" dialog box



Filter:

○ Prepend packets to existing file

● Merge packets chronologically

○ Append packets to existing file

Filename: test.cap

Format: libpcap (tcpdump, Ethereal, etc.)

Size: 15317 bytes

Packets: 120

First Packet: 2004-06-22 20:02:48

Elapsed time: 00:00:02

Prepend packets to existing file Prepend the packets from the selected file before the currently loaded packets.

Merge packets chronologically Merge both the packets from the selected and currently loaded file in chronological order.

Append packets to existing file Append the packets from the selected file after the currently loaded packets.

All other controls will work the same way as in the "Open Capture File" dialog box, see Section 5.2.1, “The "Open Capture File" dialog box”.

5.5. Exporting data

Ethereal provides several ways and formats to export packet data. This section describes general ways to export data from Ethereal.

Note!

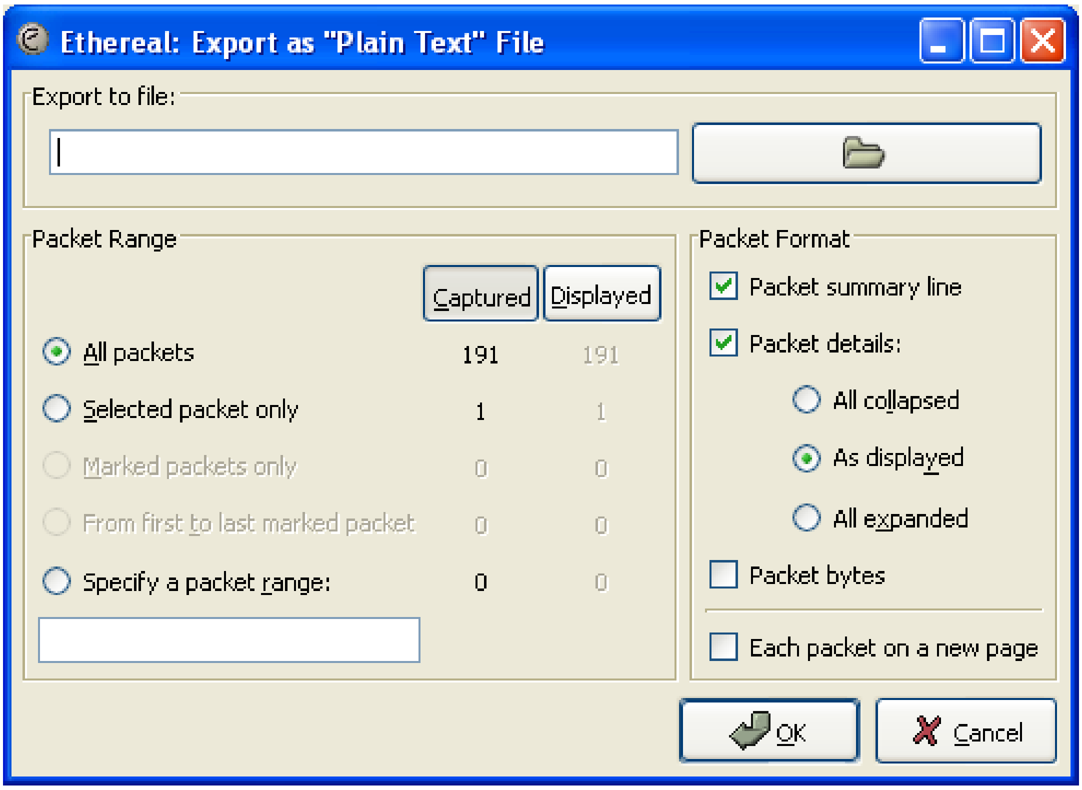
There are more specialized functions to export specific data, which will be described at the appropriate places.

XXX - add detailed descriptions of the output formats and some sample output, too.

5.5.1. The "Export as Plain Text File" dialog box

Export packet data into a plain ASCII text file, much like the format used to print packets.

Figure 5.4. The "Export as Plain Text File" dialog box



Ethereal: Export as "Plain Text" File

Export to file:

Packet Range

	Captured	Displayed
● All packets	191	191
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

Packet Format

☑ Packet summary line

☑ Packet details:

○ All collapsed

● As displayed

○ All expanded

□ Packet bytes

□ Each packet on a new page

- **Export to file:** frame chooses the file to export the packet data to.
- The **Packet Range** frame is described in Section 5.7, “The Packet Range frame”.
- The **Packet Details** frame is described in Section 5.8, “The Packet Format frame”.

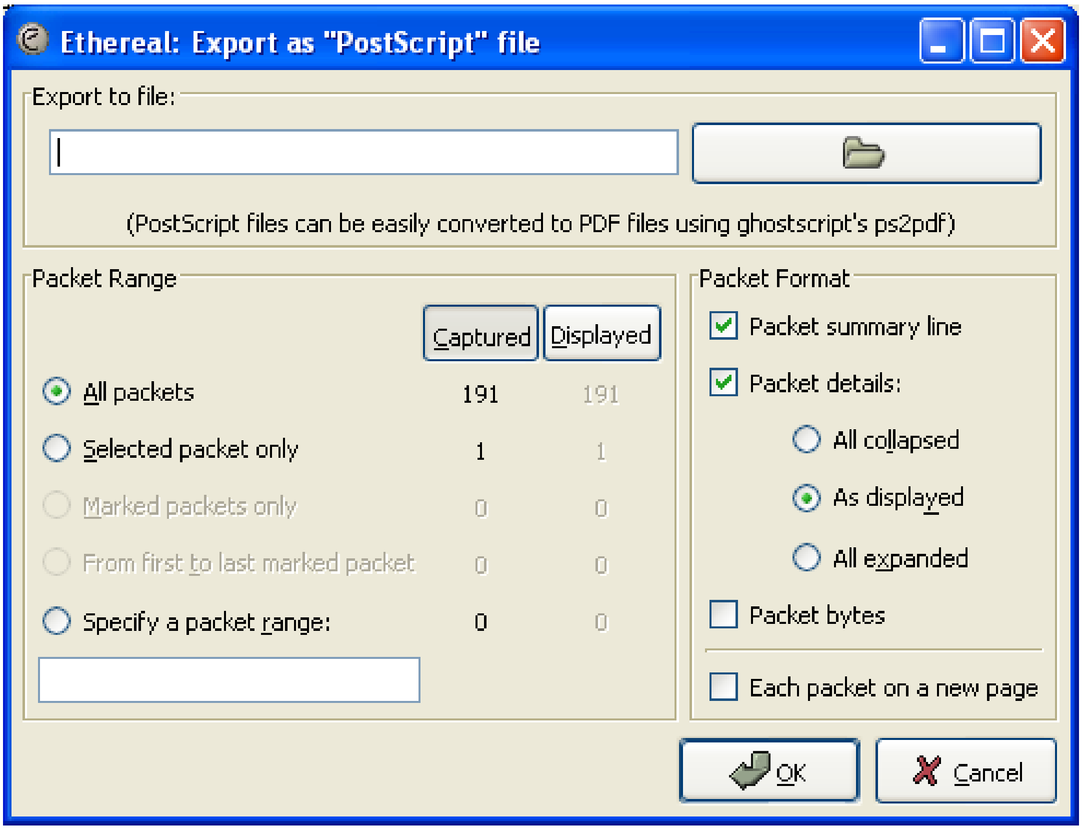
5.5.2. The "Export as PostScript File" dialog box

Export packet data into PostScript, much like the format used to print packets.

Tip!

You can easily convert PostScript files to PDF files using ghostscript. For example: export to a file named foo.ps and then call: **ps2pdf foo.ps**

Figure 5.5. The "Export as PostScript File" dialog box



Ethereal: Export as "PostScript" file

Export to file:

(PostScript files can be easily converted to PDF files using ghostscript's ps2pdf)

Packet Range

	Captured	Displayed
● All packets	191	191
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

Packet Format

☑ Packet summary line

☑ Packet details:

○ All collapsed

● As displayed

○ All expanded

□ Packet bytes

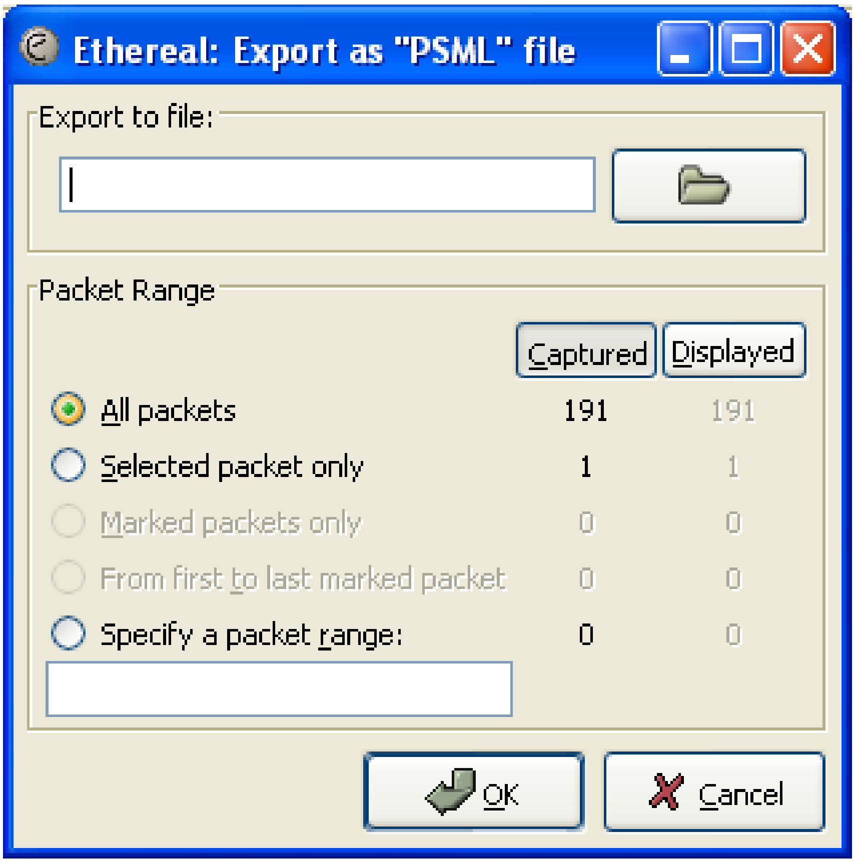
□ Each packet on a new page

- **Export to file:** frame chooses the file to export the packet data to.
- The **Packet Range** frame is described in Section 5.7, “The Packet Range frame”.
- The **Packet Details** frame is described in Section 5.8, “The Packet Format frame”.

5.5.3. The "Export as PSML File" dialog box

Export packet data into PSML. This is an XML based format including only the packet summary.

Figure 5.6. The "Export as PSML File" dialog box



Ethereal: Export as "PSML" file

Export to file:

Packet Range

	Captured	Displayed
● All packets	191	191
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

- **Export to file:** frame chooses the file to export the packet data to.
- The **Packet Range** frame is described in Section 5.7, “The Packet Range frame”.

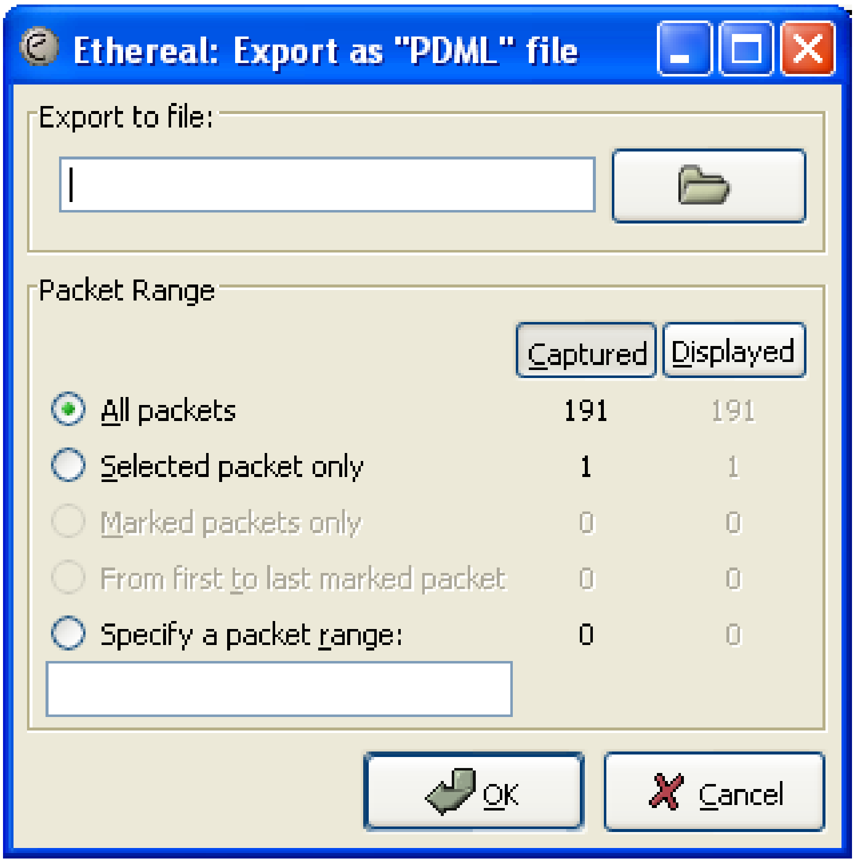
There's no such thing as a packet details frame for PSML export, as the packet format is defined by the PSML specification.

5.5.4. The "Export as PDML File" dialog box

Export packet data into PDML. This is an XML based format including the packet details. The PDML file specification is available at: PDML specification.

The PDML specification is not officially released and Ethereal's implementation of it is still in an early beta state, so please expect changes in future Ethereal versions.

Figure 5.7. The "Export as PDML File" dialog box


Ethereal: Export as "PDML" file

Export to file:

Packet Range

	Captured	Displayed
● All packets	191	191
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

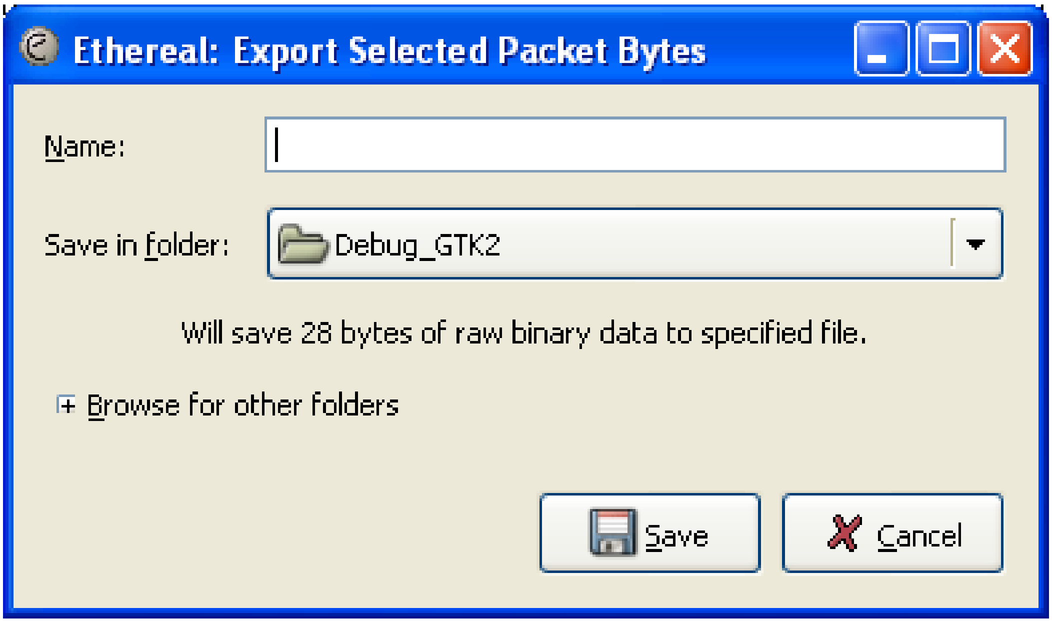
- **Export to file:** frame chooses the file to export the packet data to.
- The **Packet Range** frame is described in Section 5.7, “The Packet Range frame”.

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

5.5.5. The "Export selected packet bytes" dialog box

Export the bytes selected in the "Packet Bytes" pane into a raw binary file.

Figure 5.8. The "Export Selected Packet Bytes" dialog box



Will save 28 bytes of raw binary data to specified file.

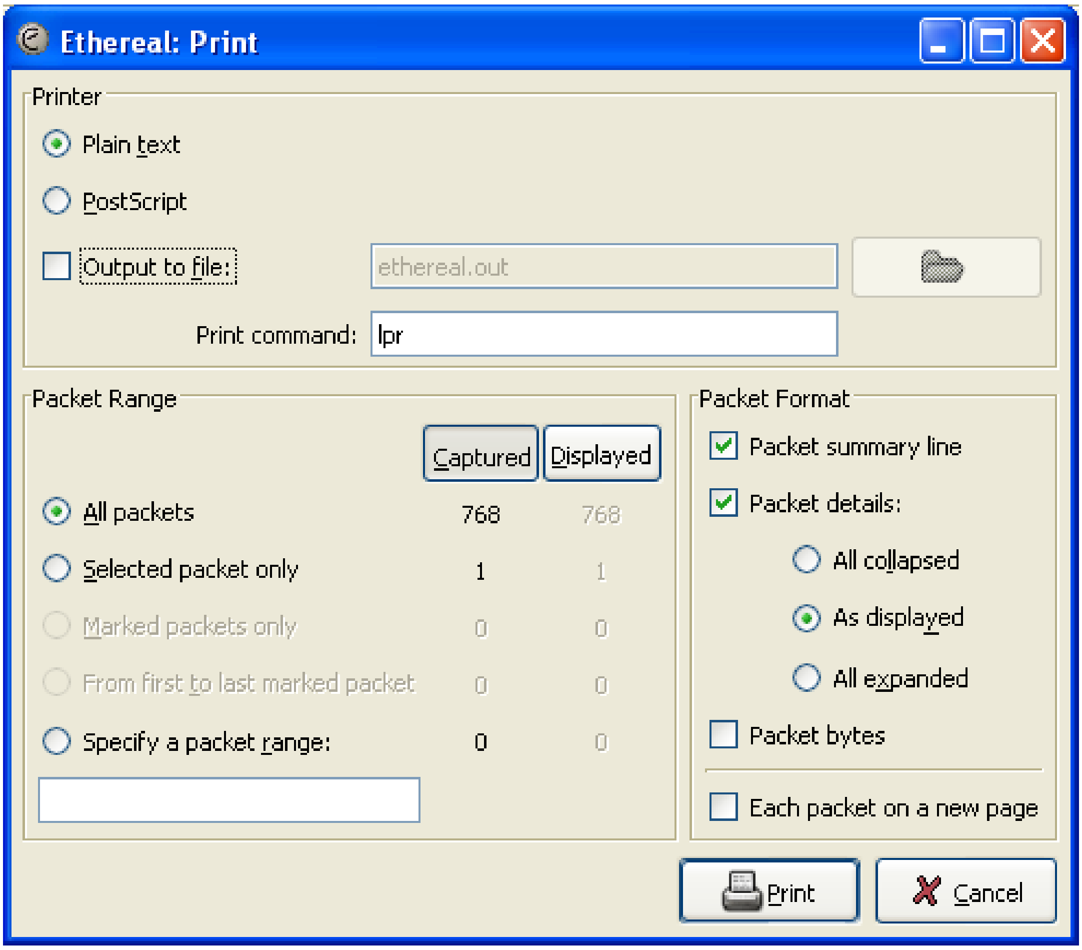
- **Name:** the filename to export the packet data to.
- The **Save in folder:** field lets you select the folder to save to (from some predefined folders).
- **Browse for other folders** provides a flexible way to choose a folder.

5.6. Printing packets

To print packets, select the "Print..." menu item from the File menu. When you do this, Ethereal pops up the Print dialog box as shown in Figure 5.9, “The "Print" dialog box”.

5.6.1. The "Print" dialog box

Figure 5.9. The "Print" dialog box



Ethereal: Print

Printer

● Plain text

○ PostScript

□ Output to file: ethereal.out

Print command: lpr

Packet Range

	Captured	Displayed
● All packets	768	768
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

Packet Format

☑ Packet summary line

☑ Packet details:

○ All collapsed

● As displayed

○ All expanded

□ Packet bytes

□ Each packet on a new page

The following fields are available in the Print dialog box:

Printer This field contains a pair of mutually exclusive radio buttons:

- **Plain Text** specifies that the packet print should be in plain text.
- **PostScipt** specifies that the packet print process should use PostScript to generate a better print output on PostScript aware printers.
- **Output to file:** specifies that printing be done to a file, which name is entered in the field or selected using the browse button.

  This field is where you enter the **file** to print to if you have selected Print to a file, or you can click the button to browse the filesystem. It is greyed out if Print to a file is not selected.

- **Print command** specifies that a command be used for printing.

  ### Note!

  These **Print command** fields are not available on windows platforms.

  This field specifies the command to use for printing. It is typically **lpr**. You would change it to specify a particular queue if you need to print to a queue other than the default. An example might be:

  ```
  lpr -Pmypostscript
  ```

  This field is greyed out if **Output to file:** is checked above.

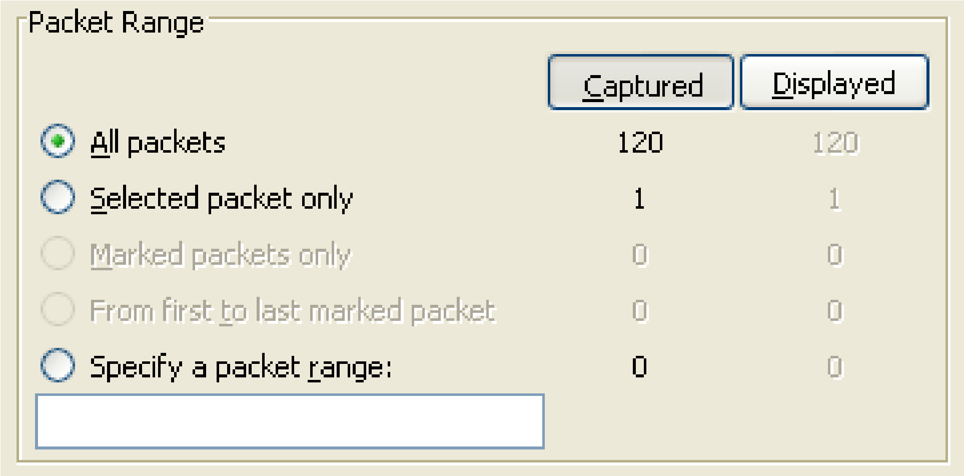
Packet Range Select the packets to be printed, see Section 5.7, “The Packet Range frame”

Packet Format Select the output format of the packets to be printed. You can choose, how each packet is printed, see Figure 5.11, “The "Packet Format" frame”

5.7. The Packet Range frame

The packet range frame is a part of various output related dialog boxes. It provides options to select which packets should be processed for the output function.

Figure 5.10. The "Packet Range" frame



Packet Range

	Captured	Displayed
● All packets	120	120
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
○ Specify a packet range:	0	0

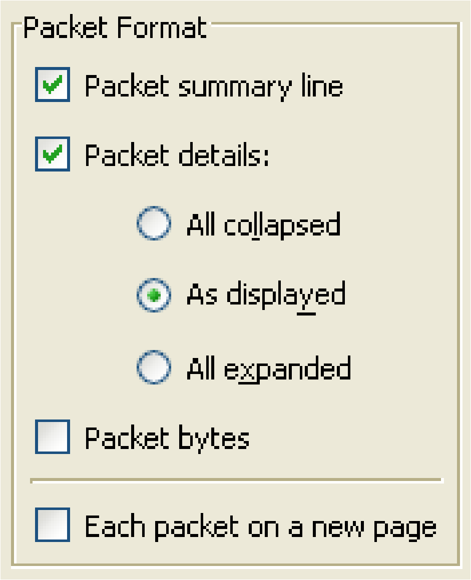
If the **Captured** button is set (default), all packets from the selected rule will be processed. If the **Displayed** button is set, only the currently displayed packets are taken into account to the selected rule.

- **All packets** will process all packets.
- **Selected packet only** process only the selected packet.
- **Marked packets only** process only the marked packets.
- **From first to last marked packet** process the packets from the first to the last marked one.
- **Specify a packet range** process a user specified range of packets, e.g. specifying **5,10-15,20-** will process the packet number five, the packets from packet number ten to fifteen (inclusive) and every packet from number twenty to the end of the capture.

5.8. The Packet Format frame

The packet format frame is a part of various output related dialog boxes. It provides options to select which parts of a packet should be used for the output function.

Figure 5.11. The "Packet Format" frame



Packet Format

☑ Packet summary line

☑ Packet details:

○ All collapsed

● As displayed

○ All expanded

□ Packet bytes

□ Each packet on a new page

- **Packet summary line** enable the output of the summary line, just as in the "Packet List" pane.
- **Packet details** enable the output of the packet details tree.
  - **All collapsed** the info from the "Packet Details" pane in "all collapsed" state.
  - **As displayed** the info from the "Packet Details" pane in the current state.
  - **All expanded** the info from the "Packet Details" pane in "all expanded" state.
- **Packet bytes** enable the output of the packet bytes, just as in the "Packet Bytes" pane.
- **Each packet on a new page** put each packet on a separate page (e.g. when saving/printing to a text file, this will put a form feed character between the packets).

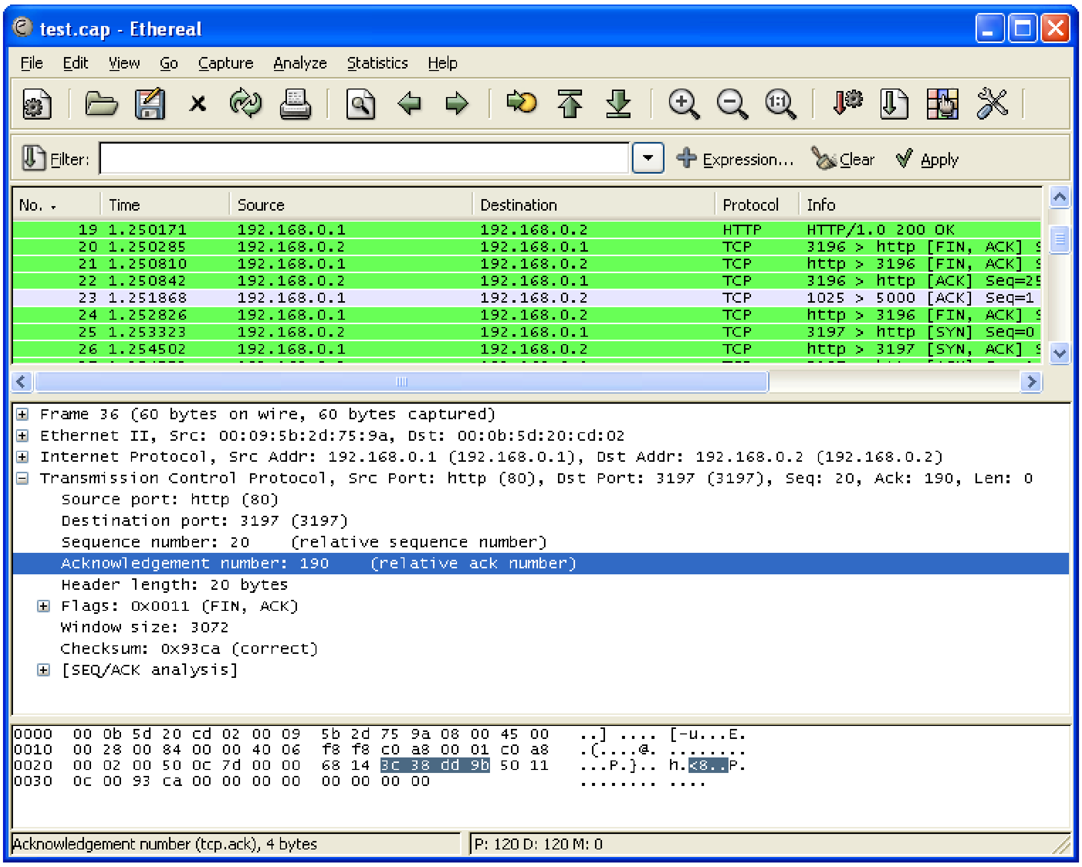
Chapter 6. Working with captured packets

6.1. Viewing packets you have captured

Once you have captured some packets, or you have opened a previously saved capture file, you can view the packets that are displayed in the packet list pane by simply clicking on that packet in the packet list pane, which will bring up the selected packet in the tree view and byte view panes.

You can then expand any part of the tree view by clicking on the **plus** sign (the symbol itself may vary) to the left of that part of the payload, and you can select individual fields by clicking on them in the tree view pane. An example with a TCP packet selected is shown in Figure 6.1, “Ethereal with a TCP packet selected for viewing”. It also has the Acknowledgment number in the TCP header selected, which shows up in the byte view as the selected bytes.

Figure 6.1. Ethereal with a TCP packet selected for viewing



```
No. .   Time        Source          Destination     Protocol  Info
     19 1.250171    192.168.0.1     192.168.0.2     HTTP      HTTP/1.0 200 OK
     20 1.250285    192.168.0.2     192.168.0.1     TCP       3196 > http [FIN, ACK] S
     21 1.250810    192.168.0.1     192.168.0.2     TCP       http > 3196 [FIN, ACK] S
     22 1.250842    192.168.0.2     192.168.0.1     TCP       3196 > http [ACK] Seq=25
     23 1.251868    192.168.0.1     192.168.0.2     TCP       1025 > 5000 [ACK] Seq=1
     24 1.252826    192.168.0.1     192.168.0.2     TCP       http > 3196 [FIN, ACK] S
     25 1.253323    192.168.0.2     192.168.0.1     TCP       3197 > http [SYN] Seq=0
     26 1.254502    192.168.0.1     192.168.0.2     TCP       http > 3197 [SYN, ACK] S
```

```
⊞ Frame 36 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:09:5b:2d:75:9a, Dst: 00:0b:5d:20:cd:02
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 192.168.0.2 (192.168.0.2)
⊟ Transmission Control Protocol, Src Port: http (80), Dst Port: 3197 (3197), Seq: 20, Ack: 190, Len: 0
    Source port: http (80)
    Destination port: 3197 (3197)
    Sequence number: 20    (relative sequence number)
    Acknowledgement number: 190    (relative ack number)
    Header length: 20 bytes
  ⊞ Flags: 0x0011 (FIN, ACK)
    Window size: 3072
    Checksum: 0x93ca (correct)
  ⊞ [SEQ/ACK analysis]
```

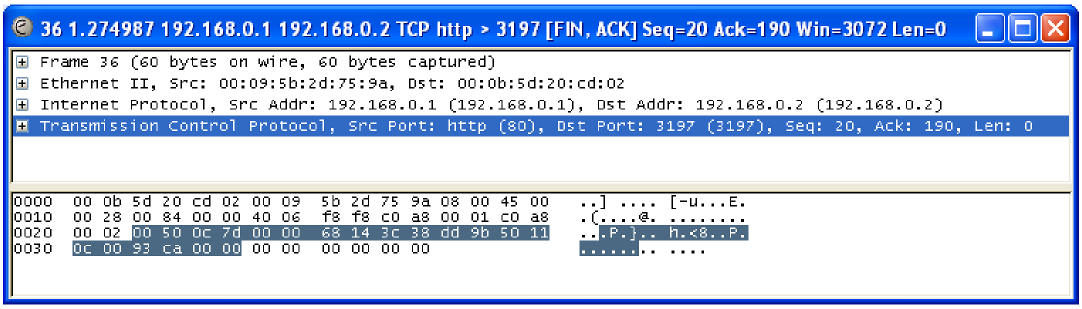
```
0000  00 0b 5d 20 cd 02 00 09  5b 2d 75 9a 08 00 45 00   ..] .... [-u...E.
0010  00 28 00 84 00 00 40 06  f8 f8 c0 a8 00 01 c0 a8   .(....@. ........
0020  00 02 00 50 0c 7d 00 00  68 14 3c 38 dd 9b 50 11   ...P.}.. h.<8..P.
0030  0c 00 93 ca 00 00 00 00  00 00 00 00               ........ ....
```

Acknowledgement number (tcp.ack), 4 bytes | P: 120 D: 120 M: 0

You can also select and view packets the same way, while Ethereal is capturing, if you selected "Update list of packets in real time" in the Ethereal Capture Preferences dialog box.

In addition, you can view individual packets in a separate window as shown in Figure 6.2, “Viewing a packet in a separate window”. Do this by selecting the packet you are interested in in the packet list pane, and then select "Show Packet in New Windows" from the Display menu. This allows you to easily compare two or more packets.

Figure 6.2. Viewing a packet in a separate window



36 1.274987 192.168.0.1 192.168.0.2 TCP http > 3197 [FIN, ACK] Seq=20 Ack=190 Win=3072 Len=0

```
⊞ Frame 36 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:09:5b:2d:75:9a, Dst: 00:0b:5d:20:cd:02
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 192.168.0.2 (192.168.0.2)
⊞ Transmission Control Protocol, Src Port: http (80), Dst Port: 3197 (3197), Seq: 20, Ack: 190, Len: 0
```

```
0000  00 0b 5d 20 cd 02 00 09  5b 2d 75 9a 08 00 45 00   ..] .... [-u...E.
0010  00 28 00 84 00 00 40 06  f8 f8 c0 a8 00 01 c0 a8   .(....@. ........
0020  00 02 00 50 0c 7d 00 00  68 14 3c 38 dd 9b 50 11   ...P.}.. h.<8..P.
0030  0c 00 93 ca 00 00 00 00  00 00 00 00               ........ ....
```

Finally, you can bring up a pop-up menu over either the "Packet List", "Packet Details" or "Packet Bytes" pane by clicking your right mouse button.

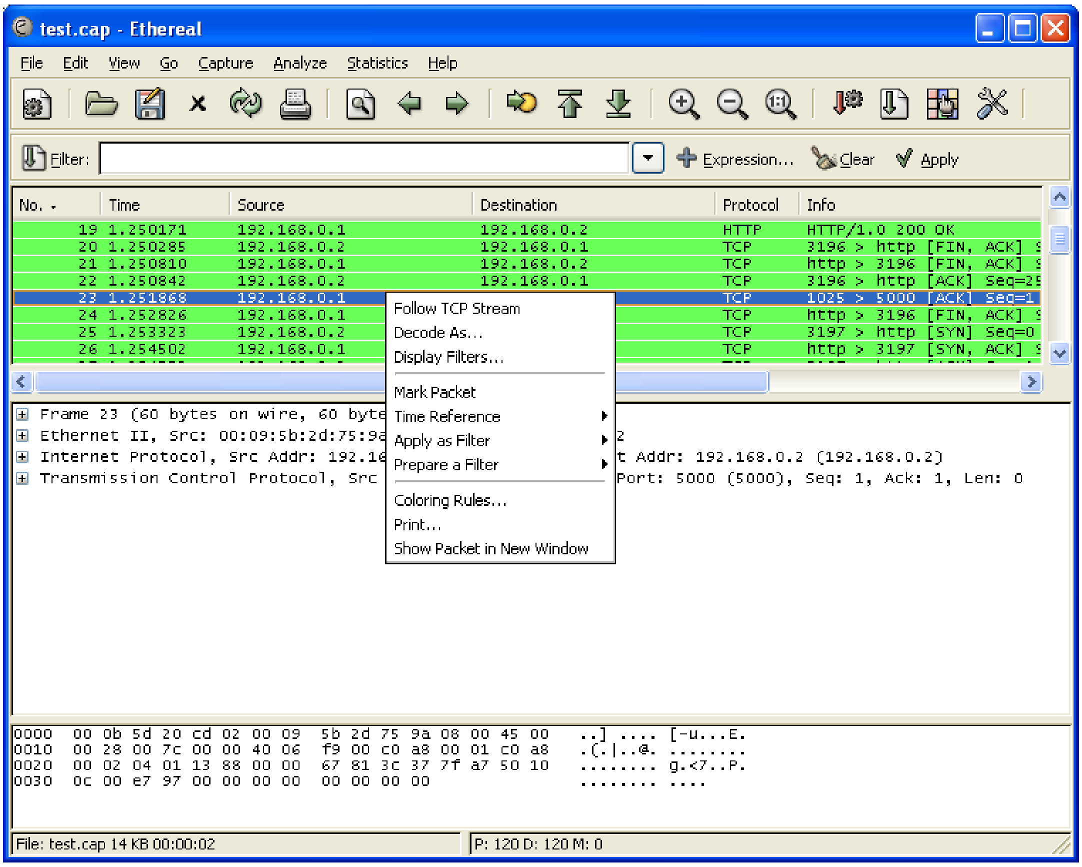
The following table gives an overview of which functions are available in the panes, where to find the corresponding function in the menu, and a short description of each item.

Table 6.1. Function overview of the pop-up menus

Item	List	Details	Bytes	Menu	Description
Follow TCP stream	X	X	X	Analyze	View all the data on a TCP stream between a pair of nodes.
Decode As...	X	X	X	Analyze	.
Display Filters...	X	X	X	Analyze	Specify and manage filters.
Mark Packet	X	-	-	Edit	Mark a packet.
Time Reference	X	-	-	Edit	Set/reset and find time references.
Apply as Filter	X	X	-	Analyze	.
Prepare a Filter	X	X	-	Analyze	.
Coloring Rules...	X	-	-	View	Colorize packets in the "Packet List" pane.
Print...	X	-	-	File	Print packets.
Show Packet in New Window	X	-	-	View	Display the selected packet in another window.
Resolve name	-	X	-	-	Cause a name resolution to be performed for the selected packet, but NOT for every packet in the capture.
Go to Corresponding Packet	-	X	-	Go	If the selected field has a packet number in it, go to it. The corresponding packet will often be a response which is requested by this packet, or the request for which this packet is a response.

Item	List	Details	Bytes	Menu	Description
Export Selected Packet Bytes...	-	X	X	File->Export	Export raw packet bytes to a binary file.
Protocol Preferences...	-	X	-	Edit	The menu item takes you to the preferences dialog and selects the page corresponding to the protocol if there are settings associated with the highlighted field. More information on preferences can be found in Section 9.5.9, “The "Protocols" pages”.
Collapse All	-	X	-	View	Ethereal keeps a list of all the protocol subtrees that are expanded, and uses it to ensure that the correct subtrees are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.
Expand All	-	X	-	View	Expand all subtrees in all packets in the capture.
Expand Tree	-	X	-	View	Expand the currently selected subtree.

Figure 6.3. Pop-up menu of "Packet List" pane



```
No. .     Time         Source          Destination     Protocol  Info
   19 1.250171     192.168.0.1     192.168.0.2     HTTP      HTTP/1.0 200 OK
   20 1.250285     192.168.0.2     192.168.0.1     TCP       3196 > http [FIN, ACK] S
   21 1.250810     192.168.0.1     192.168.0.2     TCP       http > 3196 [FIN, ACK] S
   22 1.250842     192.168.0.2     192.168.0.1     TCP       3196 > http [ACK] Seq=25
   23 1.251868     192.168.0.1                     TCP       1025 > 5000 [ACK] Seq=1
   24 1.252826     192.168.0.1                     TCP       http > 3196 [FIN, ACK] S
   25 1.253323     192.168.0.2                     TCP       3197 > http [SYN] Seq=0
   26 1.254502     192.168.0.1                     TCP       http > 3197 [SYN, ACK] S
```

```
⊞ Frame 23 (60 bytes on wire, 60 byte
⊞ Ethernet II, Src: 00:09:5b:2d:75:9a                   2
⊞ Internet Protocol, Src Addr: 192.16                   t Addr: 192.168.0.2 (192.168.0.2)
⊞ Transmission Control Protocol, Src                    Port: 5000 (5000), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 0b 5d 20 cd 02 00 09  5b 2d 75 9a 08 00 45 00   ..] .... [-u...E.
0010  00 28 00 7c 00 00 40 06  f9 00 c0 a8 00 01 c0 a8   .(.|..@. ........
0020  00 02 04 01 13 88 00 00  67 81 3c 37 7f a7 50 10   ........ g.<7..P.
0030  0c 00 e7 97 00 00 00 00  00 00 00 00               ........ ....
```

File: test.cap 14 KB 00:00:02 | P: 120 D: 120 M: 0

Follow TCP Stream This menu item is the same as the Analyze menu item of the same name. It allows you to view all the data on a TCP stream between a pair of nodes.

Decode As... This menu item is the same as the Analyze menu item of the same name.

Display Filters... This menu item is the same as the Analyze menu item of the same name. It allows you to specify and manage display filters.

Mark Packet This menu item is the same as the Edit menu item of the same name. It allows you to mark a packet.

Time Reference This menu item is the same as the Edit menu items of the same name. It allows you to set and work with time references.

Apply as Filter This menu item is the same as the Analyze menu items of the same name.

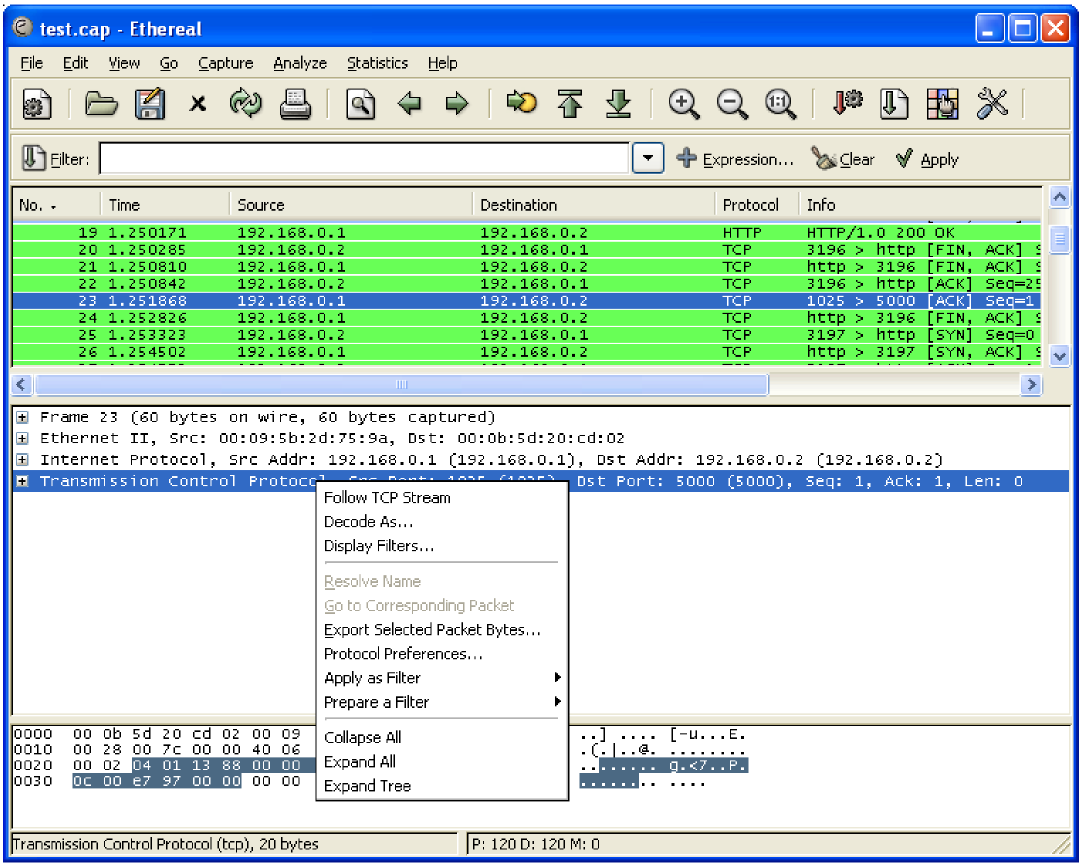
Prepare a Filter This menu item is the same as the Analyze menu items of the same name.

Coloring Rules... This menu item is the same as the View menu item of the same name. It allows you to colorize packets in the packet list pane.

Print... This menu item is the same as the File menu item of the same name. It allows you to print packets.

Show Packet in New Window This menu item is the same as the View menu item of the same name. It allows you to display the selected packet in another window.

Figure 6.4. Pop-up menu of "Packet Details" pane



```
No. -   Time       Source          Destination     Protocol  Info
     19 1.250171   192.168.0.1     192.168.0.2     HTTP      HTTP/1.0 200 OK
     20 1.250285   192.168.0.2     192.168.0.1     TCP       3196 > http [FIN, ACK] S
     21 1.250810   192.168.0.1     192.168.0.2     TCP       http > 3196 [FIN, ACK] S
     22 1.250842   192.168.0.2     192.168.0.1     TCP       3196 > http [ACK] Seq=25
     23 1.251868   192.168.0.1     192.168.0.2     TCP       1025 > 5000 [ACK] Seq=1
     24 1.252826   192.168.0.1     192.168.0.2     TCP       http > 3196 [FIN, ACK] S
     25 1.253323   192.168.0.2     192.168.0.1     TCP       3197 > http [SYN] Seq=0
     26 1.254502   192.168.0.1     192.168.0.2     TCP       http > 3197 [SYN, ACK] S
```

```
⊞ Frame 23 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:09:5b:2d:75:9a, Dst: 00:0b:5d:20:cd:02
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 192.168.0.2 (192.168.0.2)
⊞ Transmission Control Protocol, Src Port: 1025 (1025), Dst Port: 5000 (5000), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 0b 5d 20 cd 02 00 09     ..] .... [-u...E.
0010  00 28 00 7c 00 00 40 06     .(.|..@. ........
0020  00 02 04 01 13 88 00 00     ........ g.<7..P.
0030  0c 00 e7 97 00 00 00 00     ........ ....
```

Transmission Control Protocol (tcp), 20 bytes

P: 120 D: 120 M: 0

Follow TCP Stream
This menu item is the same as the Analyze menu item of the same name. It allows you to view all the data on a TCP stream between a pair of nodes.

Decode As...
This menu item is the same as the Analyze menu item of the same name.

Display Filters...
This menu item is the same as the Analyze menu item of the same name. It allows you to specify and manage filters.

Resolve Name
This menu item causes name resolution to be performed for the selected packet, but NOT every packet in the capture.

Go to Corresponding Packet
If the selected field has a corresponding packet, go to it. Corresponding packets will usually be a request/response packet pair or such.

Export Selected Packet Bytes...
This menu item is the same as the File menu item of the same name. It allows you to export raw packet bytes to a binary file.

Protocol Properties...
The menu item takes you to the properties dialog and selects the page corresponding to the protocol if there are properties associated with the highlighted field. More information on preferences can be found in Figure 9.8, “The "User Interface" preferences page”.

Apply as Filter
This menu item is the same as the Analyze menu items of the same name.

Prepare a Filter
This menu item is the same as the Analyze menu items of the

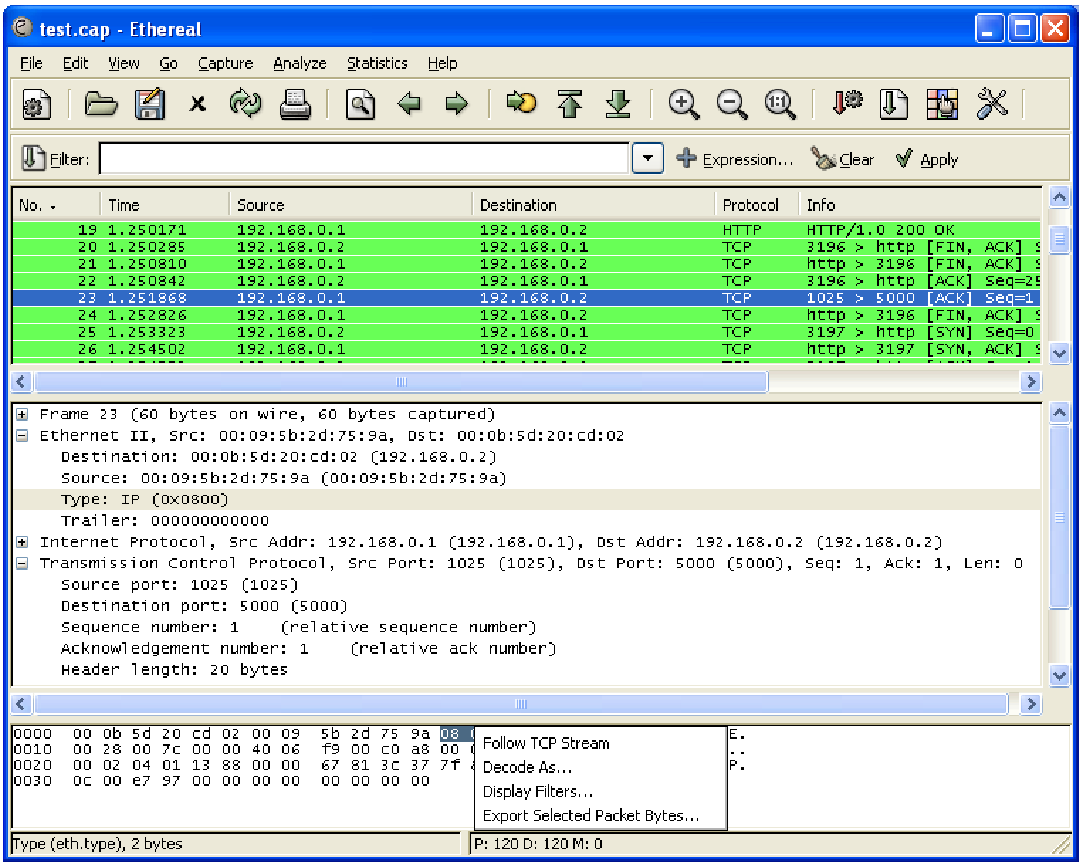
same name.

Collapse All Ethereal keeps a list of all the protocol subtrees that are expanded, and uses it to ensure that the correct subtrees are expanded when you display a packet. This menu item collapses the tree view of all packets in the capture list.

Expand All This menu item expands all subtrees in all packets in the capture.

Expand Tree This menu item expands the currently selected subtree.

Figure 6.5. Pop-up menu of "Packet Bytes" pane



```
No. -    Time       Source           Destination      Protocol  Info
      19 1.250171   192.168.0.1      192.168.0.2      HTTP      HTTP/1.0 200 OK
      20 1.250285   192.168.0.2      192.168.0.1      TCP       3196 > http [FIN, ACK] S
      21 1.250810   192.168.0.1      192.168.0.2      TCP       http > 3196 [FIN, ACK] S
      22 1.250842   192.168.0.2      192.168.0.1      TCP       3196 > http [ACK] Seq=25
      23 1.251868   192.168.0.1      192.168.0.2      TCP       1025 > 5000 [ACK] Seq=1
      24 1.252826   192.168.0.1      192.168.0.2      TCP       http > 3196 [FIN, ACK] S
      25 1.253323   192.168.0.2      192.168.0.1      TCP       3197 > http [SYN] Seq=0
      26 1.254502   192.168.0.1      192.168.0.2      TCP       http > 3197 [SYN, ACK] S
```

```
⊞ Frame 23 (60 bytes on wire, 60 bytes captured)
⊟ Ethernet II, Src: 00:09:5b:2d:75:9a, Dst: 00:0b:5d:20:cd:02
    Destination: 00:0b:5d:20:cd:02 (192.168.0.2)
    Source: 00:09:5b:2d:75:9a (00:09:5b:2d:75:9a)
    Type: IP (0x0800)
    Trailer: 000000000000
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 192.168.0.2 (192.168.0.2)
⊟ Transmission Control Protocol, Src Port: 1025 (1025), Dst Port: 5000 (5000), Seq: 1, Ack: 1, Len: 0
    Source port: 1025 (1025)
    Destination port: 5000 (5000)
    Sequence number: 1    (relative sequence number)
    Acknowledgement number: 1    (relative ack number)
    Header length: 20 bytes
```

```
0000  00 0b 5d 20 cd 02 00 09  5b 2d 75 9a 08 (     E.
0010  00 28 00 7c 00 00 40 06  f9 00 c0 a8 00 (     ..
0020  00 02 04 01 13 88 00 00  67 81 3c 37 7f a     P.
0030  0c 00 e7 97 00 00 00 00  00 00 00 00
```

Type (eth.type), 2 bytes   P: 120 D: 120 M: 0

Follow TCP Stream This menu item is the same as the Analyze menu item of the same name. It allows you to view all the data on a TCP stream between a pair of nodes.

Decode As... This menu item is the same as the Analyze menu item of the same name.

Display Filters... This menu item is the same as the Analyze menu item of the same name. It allows you to specify and manage filters.

Export Selected Packet Bytes... This menu item is the same as the File menu item of the same name. It allows you to export raw packet bytes to a binary file.

6.2. Filtering packets while viewing

Ethereal has two filtering languages: One used when capturing packets, and one used when displaying packets. In this section we explore that second type of filter: Display filters. The first one has already been dealt with in Section 4.5, “Filtering while capturing”.

Display filters allow you to concentrate on the packets you are interested in. They allow you to select packets by:

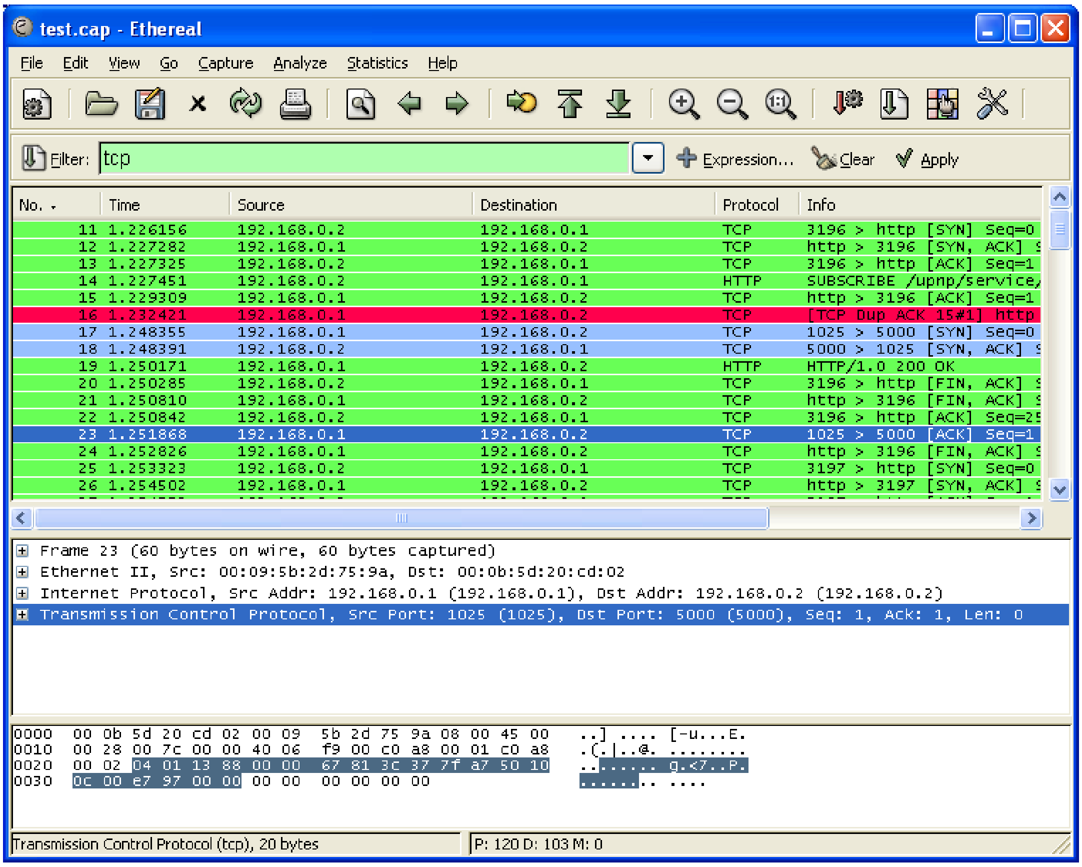
- Protocol
- The presence of a field
- The values of fields
- A comparison between fields
- ... and a lot more!

To select packets based on protocol type, simply type the protocol you are interested in in the **Filter:** field in the filter toolbar of the Ethereal window and press enter to initiate the filter. Figure 6.6, “Filtering on the TCP protocol” shows an example of what happens when you type **tcp** in the filter field.

Note!

All protocol and field names are entered in lowercase. Also, don't forget to press enter after entering the filter expression.

Figure 6.6. Filtering on the TCP protocol


test.cap - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter: tcp
Expression... Clear Apply

No.	Time	Source	Destination	Protocol	Info
11	1.226156	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0
12	1.227282	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] S
13	1.227325	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1
14	1.227451	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/
15	1.229309	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1
16	1.232421	192.168.0.1	192.168.0.2	TCP	[TCP Dup ACK 15#1] http
17	1.248355	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0
18	1.248391	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] S
19	1.250171	192.168.0.1	192.168.0.2	HTTP	HTTP/1.0 200 OK
20	1.250285	192.168.0.2	192.168.0.1	TCP	3196 > http [FIN, ACK] S
21	1.250810	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] S
22	1.250842	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=25
23	1.251868	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [ACK] Seq=1
24	1.252826	192.168.0.1	192.168.0.2	TCP	http > 3196 [FIN, ACK] S
25	1.253323	192.168.0.2	192.168.0.1	TCP	3197 > http [SYN] Seq=0
26	1.254502	192.168.0.1	192.168.0.2	TCP	http > 3197 [SYN, ACK] S

```
⊞ Frame 23 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:09:5b:2d:75:9a, Dst: 00:0b:5d:20:cd:02
⊞ Internet Protocol, Src Addr: 192.168.0.1 (192.168.0.1), Dst Addr: 192.168.0.2 (192.168.0.2)
⊞ Transmission Control Protocol, Src Port: 1025 (1025), Dst Port: 5000 (5000), Seq: 1, Ack: 1, Len: 0
```

```
0000  00 0b 5d 20 cd 02 00 09  5b 2d 75 9a 08 00 45 00   ..] .... [-u...E.
0010  00 28 00 7c 00 00 40 06  f9 00 c0 a8 00 01 c0 a8   .(.|..@. ........
0020  00 02 04 01 13 88 00 00  67 81 3c 37 7f a7 50 10   ........ g.<7..P.
0030  0c 00 e7 97 00 00 00 00  00 00 00 00               ........ ....
```

Transmission Control Protocol (tcp), 20 bytes | P: 120 D: 103 M: 0

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 and not its content!

You can filter on any protocol that Ethereal 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 the Ethereal in the **Add Expression...** dialog box. You can find more information on the **Add Expression...** dialog box in Section 6.4, “The "Filter Expression" dialog box”.

For example, to narrow the packet list pane down to only those packets to or from the IP address 192.168.0.1, use **ip.addr==192.168.0.1**.

Note!

To remove the filter, click on the **Clear** button to the right of the filter field.

6.3. Building display filter expressions

Ethereal provides a simple but powerful display filter language that you can build quite complex filter expressions with. You can compare values in packets as well as combine expressions into more specific expressions. The following sections provide more information on doing this.

6.3.1. Display filter fields

Every field in the packet details pane can be used as a filter string, this will result in showing only the packets where this field exists. For example: the filter string: **tcp** will show all packets containing the tcp protocol.

There is a complete list of all filter fields available through the menu item "Help/Supported Protocols" in the page "Display Filter Fields" of the upcoming dialog.

XXX - add some more info here and a link to the statusbar info.

6.3.2. Comparing values

You can build display filters that compare values using a number of different comparison operators. They are shown in Table 6.2, "Display Filter comparison operators".

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

Table 6.2. Display Filter comparison operators

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

English	C-like	Description and example
le	<=	**Less than or equal to** `frame.pkt_len <= 0x20`

In addition, all protocol fields are typed. Table 6.3, “Display Filter Field Types” provides a list of the types and example of how to express them.

Table 6.3. 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`

Signed integer (8-bit, 16-bit, 24-bit, 32-bit)	
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)	eth.addr == ff:ff:ff:ff:ff:ff
IPv4 address	ip.addr == 192.168.0.1
IPv6 address	
IPX network number	
String (text)	
Double-precision floating point number	

6.3.3. Combining expressions

You can combine filter expressions in Ethereal using the logical operators shown in Table 6.4, “Display Filter Logical Operations”

Table 6.4. Display Filter Logical Operations

English	C-like	Description and example
and	&&	**Logical AND**

English	C-like	Description and example
		ip.addr==10.0.0.5 and tcp.flags.fin
or	||	**Logical OR** ip.addr==10.0.0.5 or ip.addr==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** Ethereal allows you to select subsequences of a sequence in rather elaborate ways. After a label you can place a pair of brackes [] containing a comma separated list of range specifiers. 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

English	C-like	Description and example
		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` Ethereal allows you to string together single ranges in a comma separated list to form compound ranges as shown above.

6.3.4. A common mistake

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".

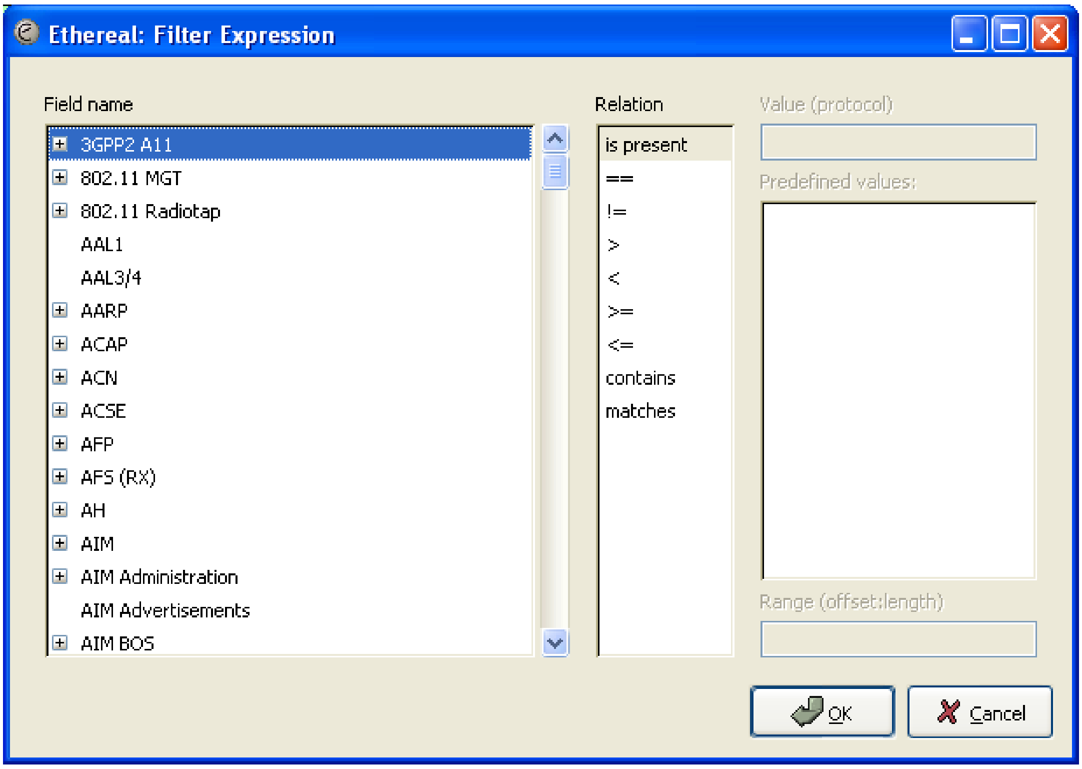
6.4. The "Filter Expression" dialog box

When you are accustomed to Ethereal'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 Ethereal 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 Ethereal 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. By clicking on the "+" next to a protocol name you can get a list of the field names available for filtering for that protocol.

Relation Select a relation from the list of available relation. The **is present** is a unary relation which is true if the selected field is present in a packet. All other listed relations are binary relations which require additional data (e.g. a **Value** to match) to complete.

When you select a field from the field name list and select a binary relation (such as the equality relation ==) you will be given the opportunity to enter a value, and possibly some range information.

Value

You may enter an appropriate value in the **Value** text box. The **Value** will also indicate the type of value for the **field name** you have selected (like character string).

Predefined values

Some of the protocol fields have predefined values available, much like enum's in C. If the selected protocol field has such values defined, you can choose it here.

Range

XXX - add an explanation here!

OK

When you have built a satisfactory expression click **OK** and a filter string will be built for you.

Cancel

You can leave the **Add Expression...** dialog box without any effect by clicking the **Cancel**

6.5. Defining and saving filters

You can define filters with Ethereal 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 filter, select the **Capture Filters...** menu item from the Capture menu or the **Display Filters...** menu item from the Analyze menu. Ethereal will then pop up the Filters dialog as shown in Figure 6.8, “The "Capture Filters" and "Display Filters" dialog boxes”.

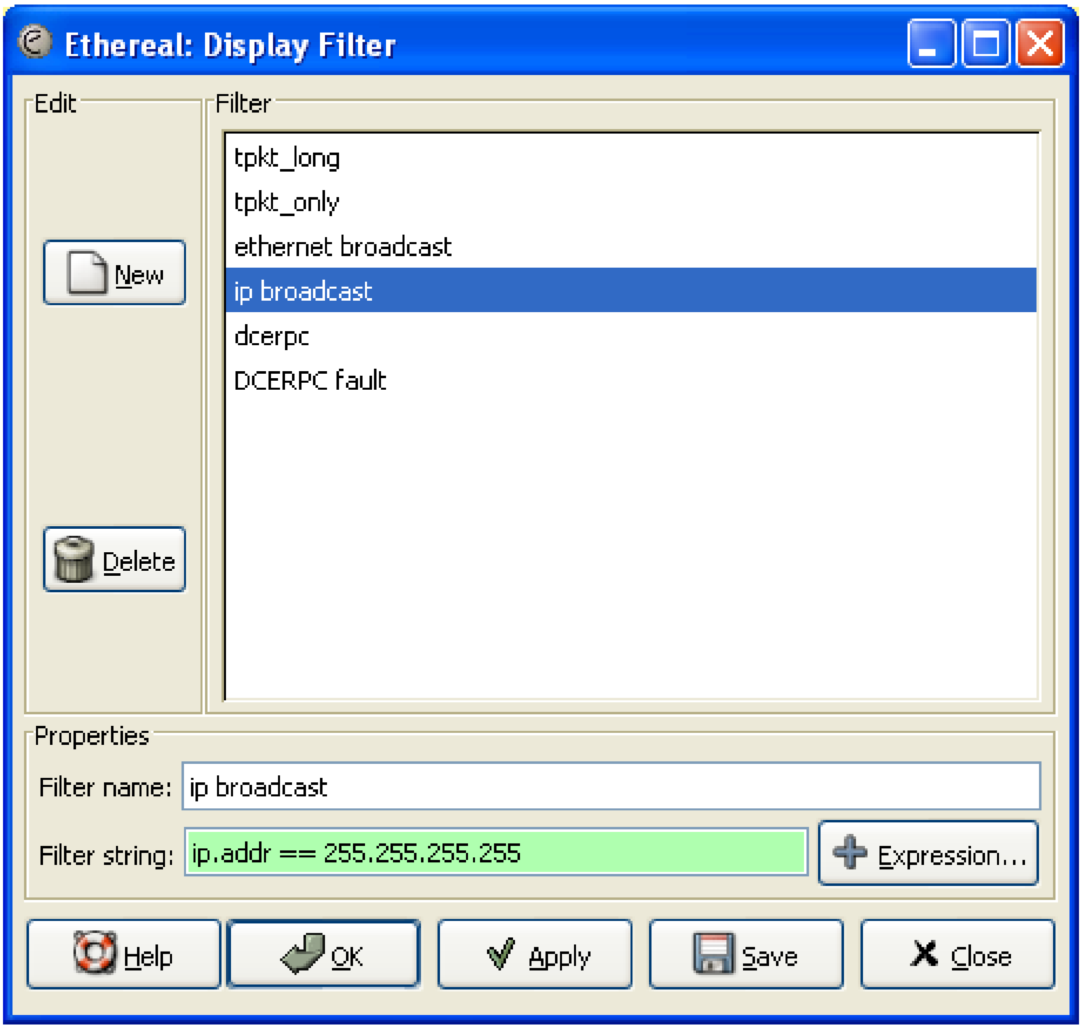
Note!

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 Ethereal.

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



Filter name: ip broadcast

Filter string: ip.addr == 255.255.255.255

New
This button adds a new filter to the list of filters. The currently entered values from Filter name and Filter string will be used. If any of these fields are empty, it will be set to "new".

Delete
This button deletes the selected filter. It will be greyed out, if no filter is selected.

Filter
You can select a filter from this list (which will fill in the filter name and filter string in the fields down the bottom of the dialog box).

Filter name:
You can change the name of the currently selected filter here.

Note!

The filter name will only be used in this dialog to identify the filter for your convenience, it will not be used elsewhere. You can add multiple filters with the same name, but this is not very useful.

Filter string:
You can change the filter string of the currently selected filter here. Display Filter only: the string will be syntax checked while you are typing.

Add Expression...
Display Filter only: This button brings up the Add Expression dialog box which assists in building filter strings. You can find more information about the Add Expression dialog in Section 6.4, “The "Filter Expression" dialog box”

OK
Display Filter only: This button applies the selected filter to the current display and closes the dialog.

Apply
Display Filter only: This button applies the selected filter to the current display, and keeps the dialog open.

Save
Save the current settings in this dialog. The file location and format is explained in Appendix A, *Configuration Files and Folders*.

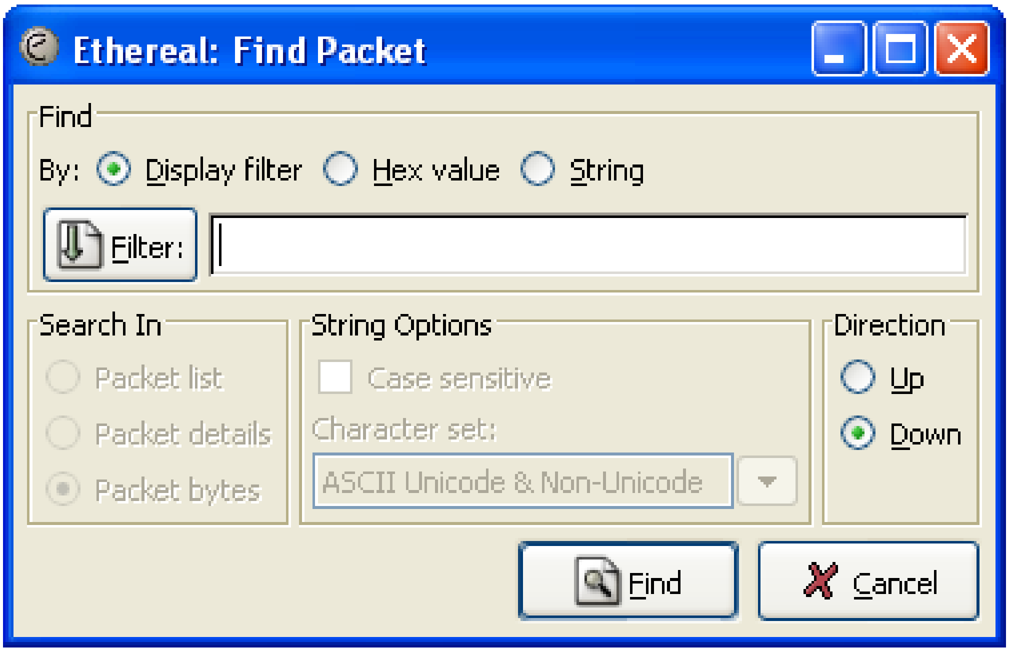
Close
Close this dialog. This will discard unsaved settings.

6.6. Finding packets

You can easily find packets once you have captured some packets or have read in a previously saved capture file. Simply select the **Find Packet...** menu item from the **Edit** menu. Ethereal will pop up the dialog box shown in Figure 6.9, “The "Find Packet" dialog box”.

6.6.1. The "Find Packet" dialog box

Figure 6.9. The "Find Packet" dialog box



Ethereal: Find Packet

Find

By: ● Display filter ○ Hex value ○ String

Filter:

Search In

○ Packet list

○ Packet details

● Packet bytes

String Options

□ Case sensitive

Character set: ASCII Unicode & Non-Unicode

Direction

○ Up

● Down

You might first select the kind of thing to search for:

- **Display filter**

  Simply enter a display filter string into the **Filter:** field, select a direction, and click on OK.

  For example, to find the three way handshake for a connection from host 192.168.0.1, use the following filter string:

  ```
  ip.addr==192.168.0.1 and tcp.flags.syn
  ```

  For more details on display filters, see Section 6.2, “Filtering packets while viewing”

- **Hex Value**

  Search for a specific byte sequence in the packet data.

  For example, use "00:00" to find the next packet including two null bytes in the packet data.

- **String**

  Find a string in the packet data, with various options.

The value to be found will by syntax checked while you type it in. If the syntax check of your value succeeded, the background of the entry field will turn green, if it fails, it will turn red.

You can choose the direction to be searched for:

- **Up**

  Search upwards in the packet list (decreasing packet numbers).

- **Down**

  Search downwards in the packet list (increasing packet numbers).

6.6.2. The "Find Next" command

"Find Next" will continue searching with the same options like in the last "Find Packet" run.

6.6.3. The "Find Previous" command

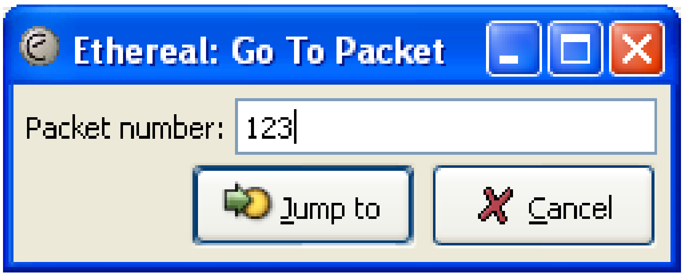
"Find Previous" will do the same thing as "Find Next", but with reverse search direction.

6.7. Go to a specific packet

You can easily jump to specific packets with one of the menu items in the Go menu.

6.7.1. The "Go to Packet" dialog box

Figure 6.10. The "Go To Packet" dialog box



Ethereal: Go To Packet

Packet number: 123

This dialog box will let you enter a packet number. When you press **OK**, Ethereal will jump to that packet.

6.7.2. The "Go to Corresponding Packet" command

If a protocol field is selected, which points to another packet in the capture file, this command will jump to that packet.

Note!

As these protocol fields now work like links (just as in your Web browser), it's easier simply to double-click on the field to jump to the corresponding field.

6.7.3. The "Go to First Packet" command

This command will simply jump to the first packet displayed.

6.7.4. The "Go to Last Packet" command

This command will simply jump to the last packet displayed.

6.8. Marking packets

You can mark packets in the "Packet List" pane. A marked packet will be shown with black background, regardless of the coloring rules set. Marking a packet can be useful to find it later while analyzing in a large capture file.

Warning!

The packet marks are not stored in the capture file or anywhere else, so all packet marks will be lost if you close the capture file.

You can use packet marking to control the output of packets when saving/exporting/printing. To do so, an option in the packet range is available, see Section 5.7, “The Packet Range frame”.

There are three functions to manipulate the marked state of a packet:

- **Mark packet** toggle the marked state of a single packet.
- **Mark all packets** set the mark state of all packets.
- **Unmark all packets** reset the mark state of all packets.

These mark function are available from the "Edit" menu, and the "Mark packet" function is also available from the popup menu of the "Packet List" pane.

6.9. Time display formats and time references

While packets are captured, each packet is timestamped. These timestamps will be saved to the capture file, so they will be available for later analysis.

When the packets are displayed, the presentation of these timestamps can be chosen by the user. There are four presentation formats available:

- **Time of Day**, e.g. 20:02:48.863096 The absolute time of the day when the packet was captured.
- **Date and Time of Day**, e.g. 2004-06-22 20:02:48.863096 The absolute date and time of the day when the packet was captured.
- **Seconds Since Beginning of Capture**, e.g. 123.299139 The time relative to the start of the capture file or the first "Time Reference" before this packet (see Section 6.9.1, “Packet time referencing”).
- **Seconds Since Previous Packet**, e.g. 1.162423 The time relative to the previous packet.

The time format can be selected from the View menu, see Figure 3.5, “The "View" Menu”.

XXX - how is the GMT / localtime thing handled.

6.9.1. Packet time referencing

The user can set time references to packets. A time reference is the starting point for all subsequent packet time calculations. It will be useful, if you want to see the time values relative to a special packet, e.g. the start of a new request. It's possible to set multiple time references in the capture file.

Warning!

The time references will not be saved permanently and will be lost when you close the capture file.

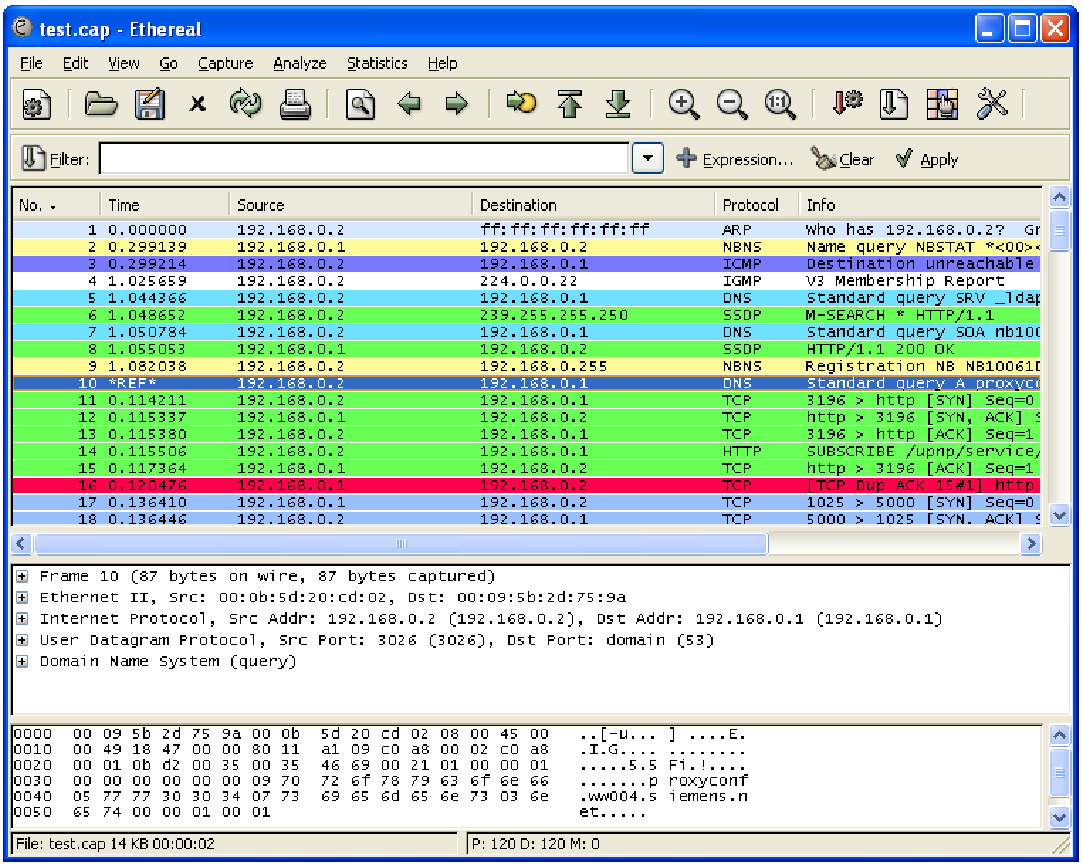
Note!

Time referencing will only be useful, if the time display format is set to "Seconds Since Beginning of Capture". If one of the other time display formats are used, time referencing will have no effect (and will make no sense either).

To work with time references, choose one of the "Time Reference" items in the "Edit" menu , see Section 3.6, “The "Edit" menu”, or from the popup menu of the "Packet List" pane.

- **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. Ethereal showing a time referenced packet



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.2	ff:ff:ff:ff:ff:ff	ARP	Who has 192.168.0.2? Gr
2	0.299139	192.168.0.1	192.168.0.2	NBNS	Name query NBSTAT *<00><
3	0.299214	192.168.0.2	192.168.0.1	ICMP	Destination unreachable
4	1.025659	192.168.0.2	224.0.0.22	IGMP	V3 Membership Report
5	1.044366	192.168.0.2	192.168.0.1	DNS	Standard query SRV _ldap
6	1.048652	192.168.0.2	239.255.255.250	SSDP	M-SEARCH * HTTP/1.1
7	1.050784	192.168.0.2	192.168.0.1	DNS	Standard query SOA nb100
8	1.055053	192.168.0.1	192.168.0.2	SSDP	HTTP/1.1 200 OK
9	1.082038	192.168.0.2	192.168.0.255	NBNS	Registration NB NB10061D
10	*REF*	192.168.0.2	192.168.0.1	DNS	Standard query A proxyco
11	0.114211	192.168.0.2	192.168.0.1	TCP	3196 > http [SYN] Seq=0
12	0.115337	192.168.0.1	192.168.0.2	TCP	http > 3196 [SYN, ACK] S
13	0.115380	192.168.0.2	192.168.0.1	TCP	3196 > http [ACK] Seq=1
14	0.115506	192.168.0.2	192.168.0.1	HTTP	SUBSCRIBE /upnp/service/
15	0.117364	192.168.0.1	192.168.0.2	TCP	http > 3196 [ACK] Seq=1
16	0.120476	192.168.0.1	192.168.0.2	TCP	[TCP Dup ACK 15#1] http
17	0.136410	192.168.0.1	192.168.0.2	TCP	1025 > 5000 [SYN] Seq=0
18	0.136446	192.168.0.2	192.168.0.1	TCP	5000 > 1025 [SYN, ACK] S

```
⊞ Frame 10 (87 bytes on wire, 87 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: 00:09:5b:2d:75:9a
⊞ Internet Protocol, Src Addr: 192.168.0.2 (192.168.0.2), Dst Addr: 192.168.0.1 (192.168.0.1)
⊞ User Datagram Protocol, Src Port: 3026 (3026), Dst Port: domain (53)
⊞ Domain Name System (query)
```

```
0000  00 09 5b 2d 75 9a 00 0b  5d 20 cd 02 08 00 45 00   ..[-u... ] ....E.
0010  00 49 18 47 00 00 80 11  a1 09 c0 a8 00 02 c0 a8   .I.G.... ........
0020  00 01 0b d2 00 35 00 35  46 69 00 21 01 00 00 01   .....5.5 Fi.!....
0030  00 00 00 00 00 00 09 70  72 6f 78 79 63 6f 6e 66   .......p roxyconf
0040  05 77 77 30 30 34 07 73  69 65 6d 65 6e 73 03 6e   .ww004.s iemens.n
0050  65 74 00 00 01 00 01                               et.....
```

File: test.cap 14 KB 00:00:02 | P: 120 D: 120 M: 0

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.

Chapter 7. Advanced Features

7.1. Introduction

In this chapter some advanced features of Ethereal will be described.

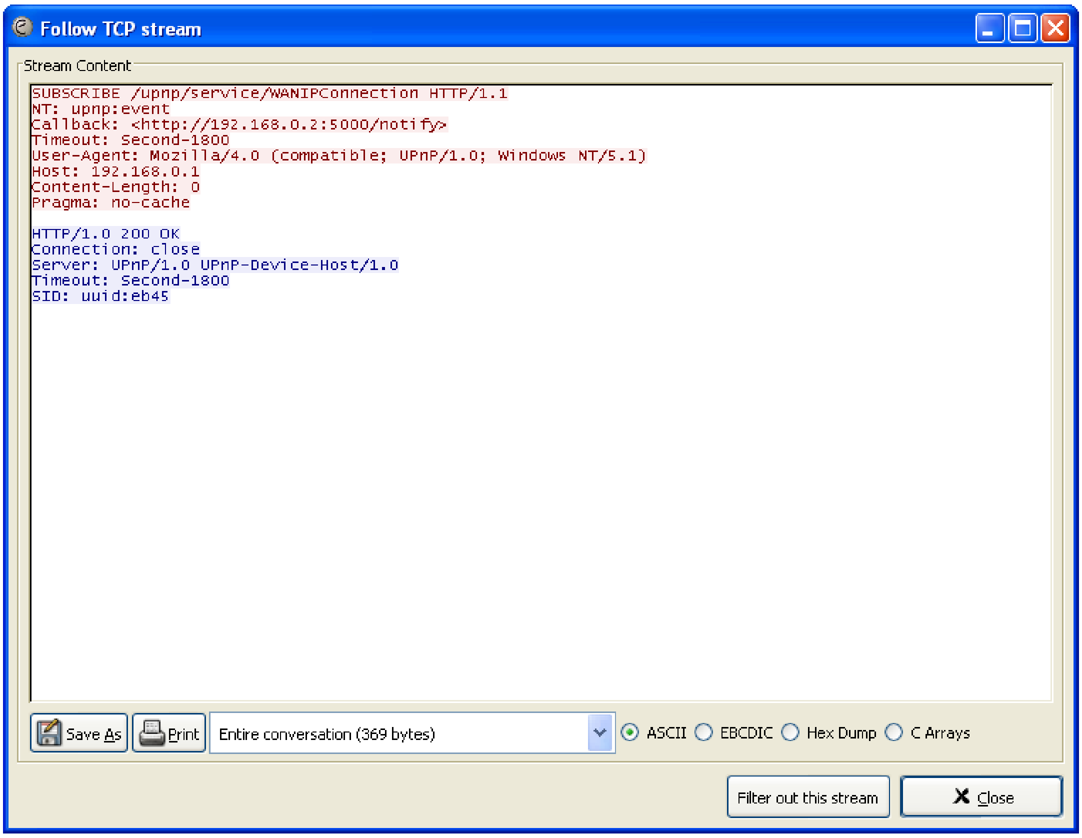
7.2. Following TCP streams

There will be occasions when you would like to see the data from a TCP session in the order 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. If so, Ethereal's ability to follow a TCP stream will be useful to you.

Simply select a TCP packet in the stream/connection you are interested in and then select the Follow TCP Stream menu item from the Ethereal Tools menu. Ethereal will pop up a separate window with all the data from the TCP stream laid out in order, as shown in Figure 7.1, “The "Follow TCP Stream" dialog box”.

7.2.1. The "Follow TCP stream" dialog box

Figure 7.1. The "Follow TCP Stream" dialog box



```
SUBSCRIBE /upnp/service/WANIPConnection HTTP/1.1
NT: upnp:event
Callback: <http://192.168.0.2:5000/notify>
Timeout: Second-1800
User-Agent: Mozilla/4.0 (compatible; UPnP/1.0; Windows NT/5.1)
Host: 192.168.0.1
Content-Length: 0
Pragma: no-cache

HTTP/1.0 200 OK
Connection: close
Server: UPnP/1.0 UPnP-Device-Host/1.0
Timeout: Second-1800
SID: uuid:eb45
```

You can choose from the following actions:

1. **Save As** Save the stream data in the currently selected format.
2. **Print** Print the stream data in the currently selected format.
3. **Direction** Choose the stream direction to be displayed ("Entire conversation", "data from A to B only" or "data from B to A only").
4. **Filter out this stream** Apply a display filter removing the current TCP stream data from the display.
5. **Close** Close this dialog box.

You can then choose to view the data in one of four formats:

1. **ASCII**. In this view you see the data from each end in ASCII, but alternating according to when each end sent data. Unfortunately, non-printing characters do not print.
2. **EBCDIC**. For the big-iron freaks out there.
3. **HEX Dump**. This allows you to see all the data, but you lose the ability to read it in ASCII.
4. **C Arrays**. This allows you to import the stream data into your own C program.

Note!

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

7.3. Packet Reassembling/Desegmenting

XXX - rework this chapter, as it's still a bit confusing.

7.3.1. What is it?

Often network protocols needs to transport large chunks of data, which are complete in itself, e.g. when transferring a file. The underlying protocol might not be able to handle that chunk size (e.g. limitation of the network packet size), or is stream-based like TCP, which doesn't know data chunks at all.

In that case the network protocol has to handle that chunks itself and (if required) spreading the data over multiple packets. It also needs a mechanism to find back the chunk boundaries on the receiving side.

Reassembling vs. Desegmenting!

Desegmenting is a slightly different mechanism compared to reassembling, but doing the same thing. Both mechanisms combine traffic back together, in this chapter only the term reassembling will be used.

7.3.2. How Ethereal handles it

For some of the network protocols Ethereal knows of, a mechanism is implemented to find, decode and display this chunks of data. Ethereal will try to find the corresponding packets of this chunk, and will show the combined data as additional pages in the "Packet Bytes" pane, see Section 3.17, “The "Packet Bytes" pane”.

Note!

Reassembling might take place in several protocol layers, so it's possible that multiple tabs in the "Packet Bytes" pane appear.

Note!

You will find the reassembled data in the last packet of the chunk.

Some examples:

- In a **HTTP** GET response, the requested data (e.g. a HTML page) is returned. Ethereal will show the hex dump of the data in a new tab "Uncompressed entity body" in the "Packet Bytes" pane.
- A **DCE-RPC** (Remote Procedure Call) client send a request to the server and expects a response back from it. Both the request and the response is a complete chunk of data and will be shown as a new tab "Reassembled DCE/RPC" in the "Packet Bytes" pane.

7.3.3. Reassembling is disabled!

Reassembling is usually disabled in the preferences by default, as it slows down packet processing a bit.

Enabling reassembling of a protocol typically requires two things:

1. the lower level protocol (e.g., TCP) must support reassembly. Often this reassembly can be enabled or disabled at will via the protocol preferences.

2. the higher level protocol (e.g., HTTP) must use the reassembly mechanism to reassemble fragmented protocol data. This too can often be enabled or disabled via the protocol preferences.

As a result, if reassembly of protocol Y on top of protocol X must be enabled, it is wise to take a look at the protocol preferences for both protocols. Check whether protocol X allows subdissectors to reassemble, and check whether protocol Y supports reassembly and has it enabled.

For example: if you have HTTP on top of TCP, you have to enable the TCP preference "Allow subdissectors to reassemble" and enable the HTTP preference "Reassemble".

7.4. Name Resolution

Name resolution tries to resolve some of the address values to human readable names. This conversion might fail. For example, the name might be unknown. Some of the lookups are done with data from your local machine, while others asking network services such as DNS.

XXX - add ipxnets name resolution explanation.

Note!

You might see packets to/from your machine in your capture file, which are caused by name resolution network services (e.g. DNS packets).

Note!

The resolved names are not stored in the capture file or somewhere else, so the resolved names might not be available if you open the capture file later or on another machine.

The name resolution feature can be en-/disabled separately for the following protocol layers:

7.4.1. MAC Layer

ARP name resolution Convert an ethernet address to the corresponding IP address (e.g. 00:09:5b:01:02:03 -> 192.168.0.1).

Ethernet manufacturer codes If the ARP name resolution failed, Ethereal tries to convert the first 3 bytes of an ethernet address to an abbreviated manufacturer name, which has been assigned by the IETF (e.g. 00:09:5b:01:02:03 -> Netgear_01:02:03).

7.4.2. Network Layer

DNS name resolution Convert an IP address to the hostname associated with it (e.g. 65.208.228.223 -> www.ethereal.com).

Warning!

Enabling network name resolution when your name server is unavailable may significantly slow Ethereal while it waits for all of the name server requests to time out. Use ADNS in that case.

7.4.3. Transport Layer

TCP/UDP port conversion Convert a TCP or UDP port to its well known name (e.g. 80 -> http).

7.4.4. ADNS

As noted, DNS lookups can significantly slow down Ethereal and make it appear frozen, which can be very annoying. To solve this, Ethereal can use the ADNS library, which handles DNS calls asynchronously.

Chapter 8. Statistics

8.1. Introduction

Ethereal provides a wide range of network statistics.

These statistics range from general information about the loaded capture file (like the number of captured packets), to statistics about specific protocols (e.g. statistics about the number of HTTP requests and responses captured).

- General statistics:
  - **Summary** about the capture file.
  - **Protocol Hierarchy** of the captured packets.
  - **Endpoints** e.g. traffic to and from an IP addresses.
  - **Conversations** e.g. traffic between specific IP addresses.
  - **IO Graphs** visualizing the number of packets (or similar) in time.
- Protocol specific statistics:
  - **Service Response Time** between request and response of some protocols.
  - **Various other** protocol specific statistics.

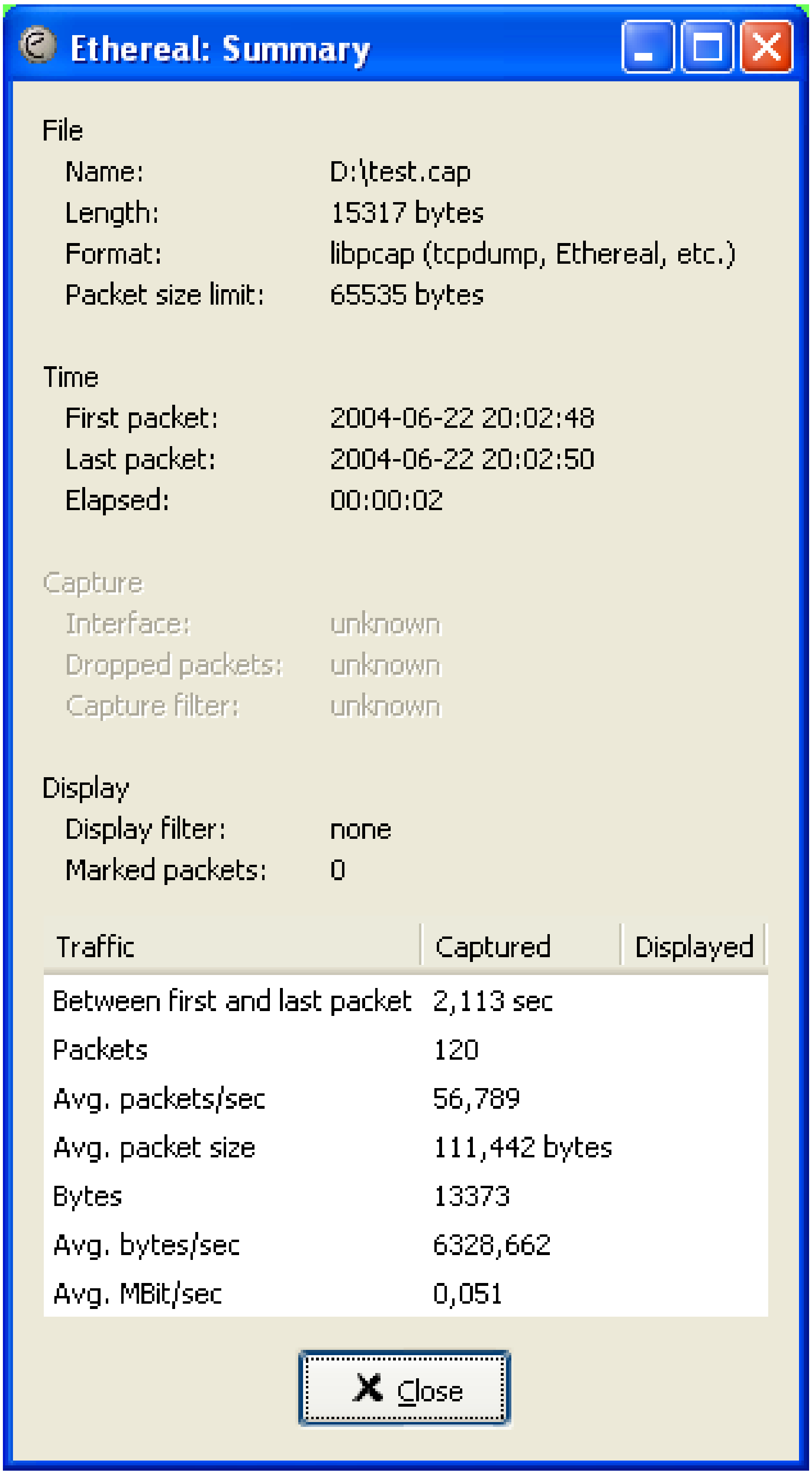
Tip!

The protocol specific statistics requires detailed knowledge about the specific protocol. Unless you are familiar with that protocol, statistics about it will be pretty hard to understand.

8.2. The "Summary" window

General statistics about the current capture file.

Figure 8.1. The "Summary" window



File

Name:	D:\test.cap
Length:	15317 bytes
Format:	libpcap (tcpdump, Ethereal, etc.)
Packet size limit:	65535 bytes

Time

First packet:	2004-06-22 20:02:48
Last packet:	2004-06-22 20:02:50
Elapsed:	00:00:02

Capture

Interface:	unknown
Dropped packets:	unknown
Capture filter:	unknown

Display

Display filter:	none
Marked packets:	0

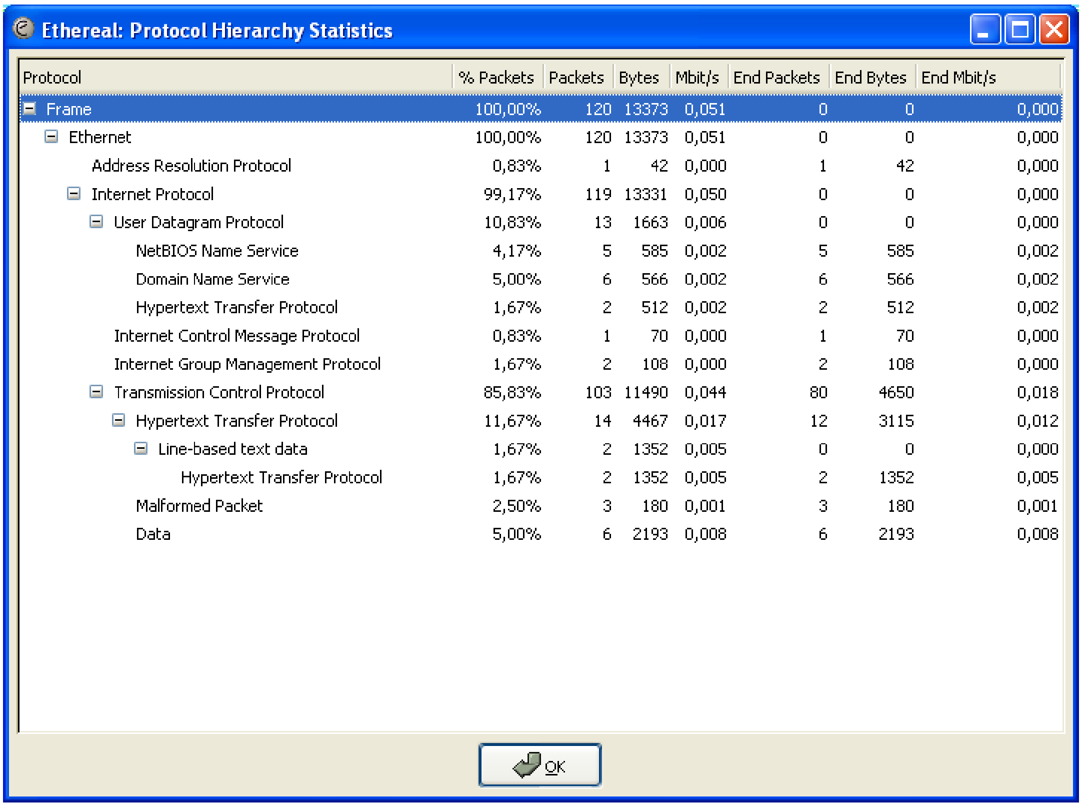
Traffic	Captured	Displayed
Between first and last packet	2,113 sec	
Packets	120	
Avg. packets/sec	56,789	
Avg. packet size	111,442 bytes	
Bytes	13373	
Avg. bytes/sec	6328,662	
Avg. MBit/sec	0,051	

- **File** general information about the capture file.
- **Time** the timestamps when the first and the last packet were capturing (and the time between them).
- **Capture** information from the time when the capture was done (only available if the packet data was captured from the network and not loaded from a file).
- **Display** some display related information.
- **Traffic** some statistics of the network traffic seen. If a display filter is set, you will see values in both columns. The values in the **Captured** column will remain the same as before, while the values in the **Displayed** column will reflect the values corresponding to the packets shown in the display.

8.3. The "Protocol Hierarchy" window

The protocol hierarchy of the captured packets.

Figure 8.2. The "Protocol Hierarchy" window



Protocol	% Packets	Packets	Bytes	Mbit/s	End Packets	End Bytes	End Mbit/s
Frame	100,00%	120	13373	0,051	0	0	0,000
Ethernet	100,00%	120	13373	0,051	0	0	0,000
Address Resolution Protocol	0,83%	1	42	0,000	1	42	0,000
Internet Protocol	99,17%	119	13331	0,050	0	0	0,000
User Datagram Protocol	10,83%	13	1663	0,006	0	0	0,000
NetBIOS Name Service	4,17%	5	585	0,002	5	585	0,002
Domain Name Service	5,00%	6	566	0,002	6	566	0,002
Hypertext Transfer Protocol	1,67%	2	512	0,002	2	512	0,002
Internet Control Message Protocol	0,83%	1	70	0,000	1	70	0,000
Internet Group Management Protocol	1,67%	2	108	0,000	2	108	0,000
Transmission Control Protocol	85,83%	103	11490	0,044	80	4650	0,018
Hypertext Transfer Protocol	11,67%	14	4467	0,017	12	3115	0,012
Line-based text data	1,67%	2	1352	0,005	0	0	0,000
Hypertext Transfer Protocol	1,67%	2	1352	0,005	2	1352	0,005
Malformed Packet	2,50%	3	180	0,001	3	180	0,001
Data	5,00%	6	2193	0,008	6	2193	0,008

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 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
- **End Packets** the absolute number of packets of this protocol (where this protocol were the highest protocol to decode)
- **End Bytes** the absolute number of bytes of this protocol (where this protocol were the highest protocol to decode)
- **End MBit/s** the bandwidth of this protocol, relative to the capture time (where this protocol were the highest protocol to decode)

Note!

Packets will usually contain multiple protocols, so more than one protocol will be counted for each packet. Example: In the screenshot IP has 99,17% and TCP 85,83% (which is together much more than 100%).

8.4. Endpoints

Statistics of the endpoints captured.

Tip!

If you are looking for a feature other network tools call a **hostlist**, here is the right place to look. The list of Ethernet or IP endpoints is usually what you're looking for.

8.4.1. What is an Endpoint?

A network endpoint is the logical endpoint of separate protocol traffic of a specific protocol layer. The endpoint statistics of Ethereal will take the following endpoints into account:

- **Ethernet** an Ethernet endpoint is identical to the Ethernet's MAC address.
- **Fibre Channel** XXX - insert info here.
- **FDDI** a FDDI endpoint is identical to the FDDI MAC address.
- **IPv4** an IP endpoint is identical to its IP address.
- **IPX** XXX - insert info here.
- **TCP** a TCP endpoint is a combination of the IP address and the TCP port used, so different TCP ports on the same IP address are different TCP endpoints.
- **Token Ring** a Token Ring endpoint is identical to the Token Ring MAC address.
- **UDP** a UDP endpoint is a combination of the IP address and the UDP port used, so different UDP ports on the same IP address are different UDP endpoints.

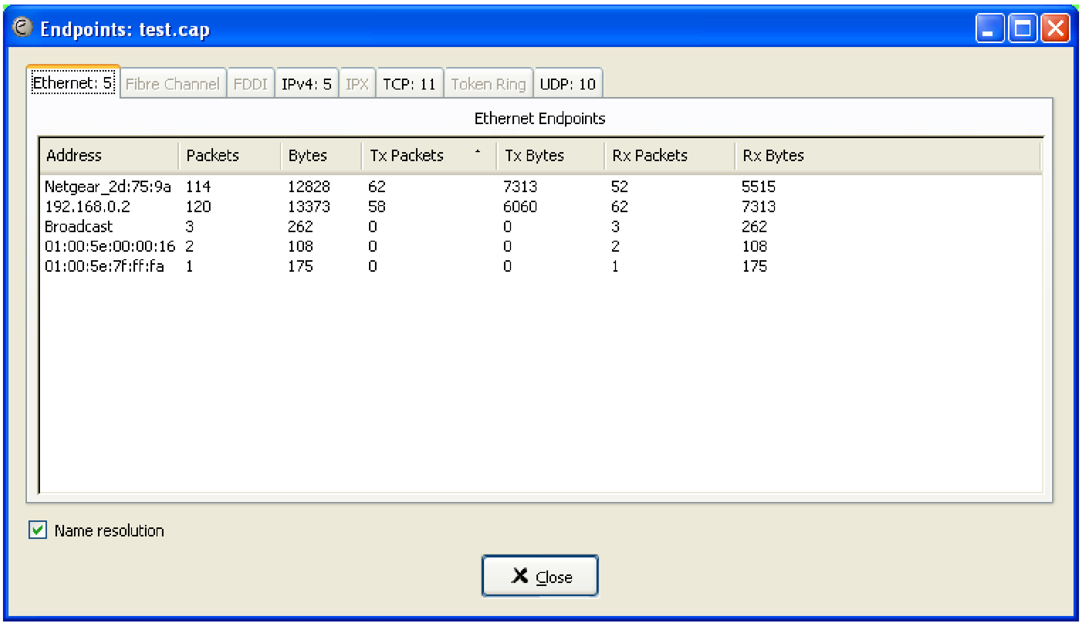
Broadcast / multicast endpoints

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.

8.4.2. The "Endpoints" window

This window shows statistics about the endpoints captured.

Figure 8.3. The "Endpoints" window



Ethernet Endpoints

Address	Packets	Bytes	Tx Packets	Tx Bytes	Rx Packets	Rx Bytes
Netgear_2d:75:9a	114	12828	62	7313	52	5515
192.168.0.2	120	13373	58	6060	62	7313
Broadcast	3	262	0	0	3	262
01:00:5e:00:00:16	2	108	0	0	2	108
01:00:5e:7f:ff:fa	1	175	0	0	1	175

☑ Name resolution

For each supported protocol, a tab is shown in this window. The tab labels 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 grayed 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 an IP address (using ARP) and the third was resolved to a broadcast (unresolved this would still be: ff:ff:ff:ff:ff:ff), the last two Ethernet addresses remain unresolved.

Tip!

This window will be updated frequently, so it will be useful, even if you open it before (or while) you are doing a live capture.

8.4.3. The protocol specific "Endpoint List" windows

Before the combined window described above was available, each of its pages were shown as separate windows. Even though the combined window is much more convenient to use, these separate windows are still available. The main reason is, they might process faster for very large capture files. However, as the functionality is exactly the same as in the combined window, they won't be discussed in detail here.

8.5. Conversations

Statistics of the captured conversations.

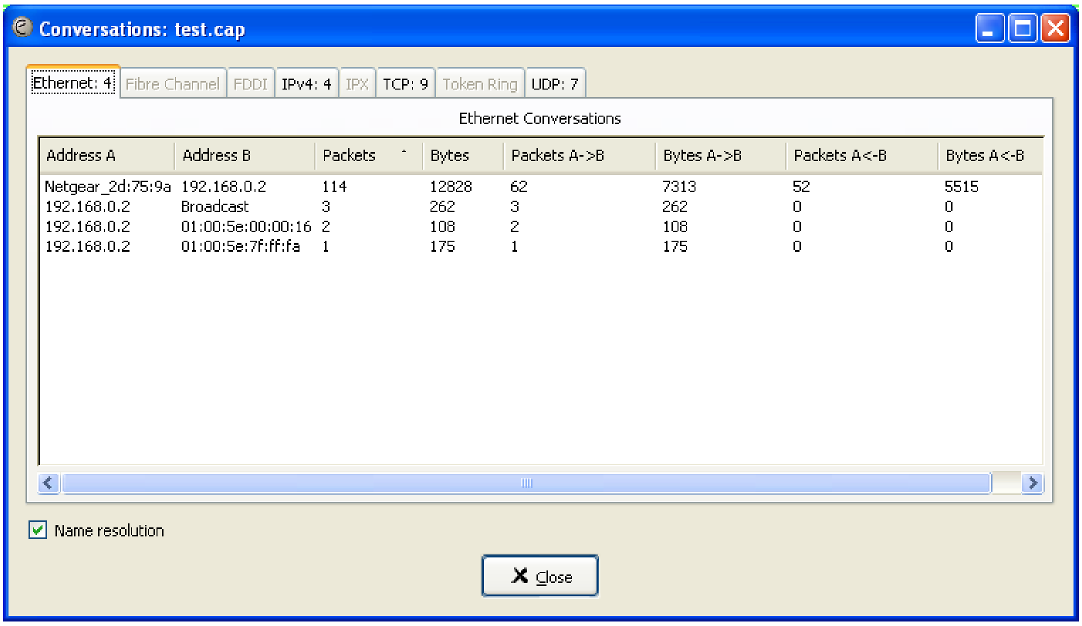
8.5.1. What is a Conversation?

A network conversation is the traffic between two specific endpoints. For example, an IP conversation is all the traffic between two IP addresses. The description of the known endpoint types can be found in Section 8.4.1, “What is an Endpoint?”.

8.5.2. The "Conversations" window

Beside the list content, the conversations window work the same way as the endpoint ones, see Section 8.4.2, “The "Endpoints" window” for a description how it works.

Figure 8.4. The "Conversations" window



Ethernet Conversations

Address A	Address B	Packets	Bytes	Packets A->B	Bytes A->B	Packets A<-B	Bytes A<-B
Netgear_2d:75:9a	192.168.0.2	114	12828	62	7313	52	5515
192.168.0.2	Broadcast	3	262	3	262	0	0
192.168.0.2	01:00:5e:00:00:16	2	108	2	108	0	0
192.168.0.2	01:00:5e:7f:ff:fa	1	175	1	175	0	0

☑ Name resolution

8.5.3. The protocol specific "Conversation List" windows

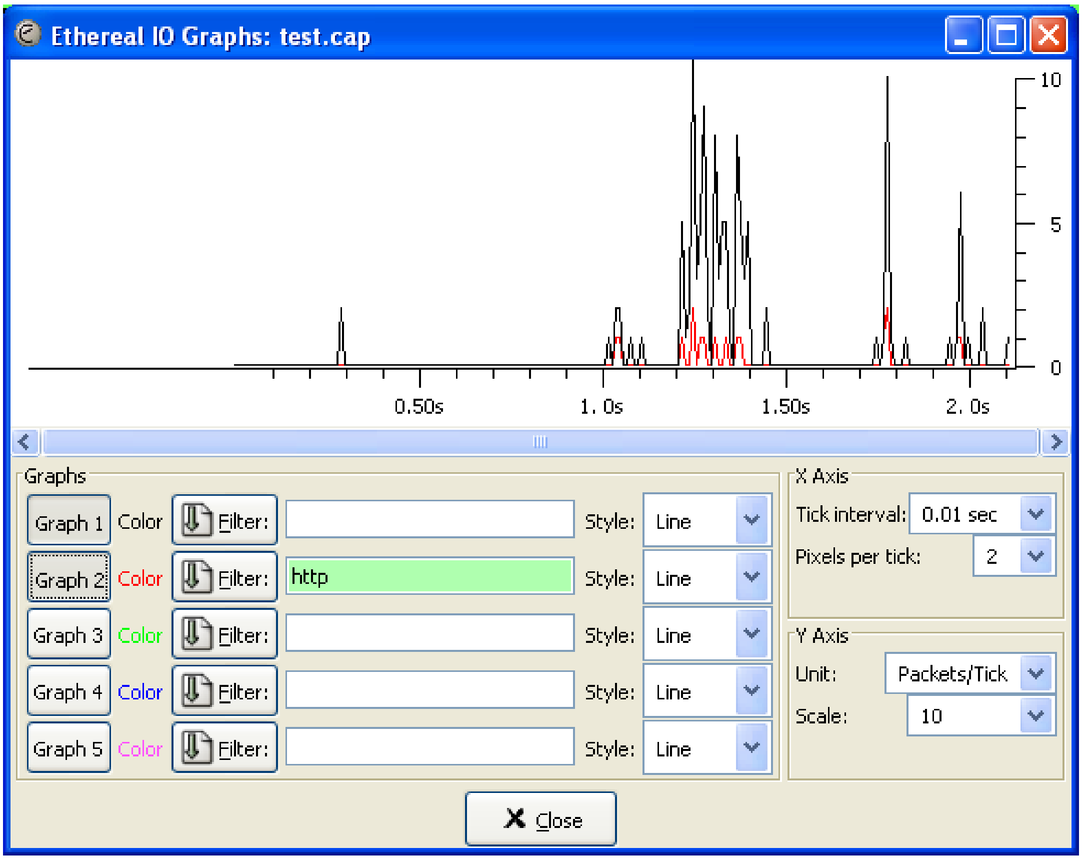
Before the combined window described above was available, each of its pages were shown as separate windows. Even though the combined window is much more convenient to use, these separate windows are still available. The main reason is, they might process faster for very large capture files. However, as the functionality is exactly the same as in the combined window, they won't be discussed in detail here.

8.6. 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



Graphs

Graph 1 Color Filter: Style: Line

Graph 2 Color Filter: http Style: Line

Graph 3 Color Filter: Style: Line

Graph 4 Color Filter: Style: Line

Graph 5 Color Filter: Style: Line

X Axis

Tick interval: 0.01 sec

Pixels per tick: 2

Y Axis

Unit: Packets/Tick

Scale: 10

The user can configure the following things:

- **Graphs**

  - **Graph 1-5** enable the 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 that graph)
  - **Style:** the style of the graph (Line/Impulse/FBar)

- **X Axis**

  - **Tick interval** an interval in x direction lasts (10/1/0.1/0.01/0.001 seconds)
  - **Pixels per tick** use 10/5/2/1 pixels per tick interval

- **Y Axis**
    - **Unit** the unit for the y direction (Packets/Tick, Bytes/Tick, Advanced...)
    - **Scale** the scale for the y unit (10,20,50,100,200,500,...)

XXX - describe the Advanced feature.

8.7. Service Response Time

The service response time is the time between a request and the corresponding response. This information is available for many protocols.

Service response time statistics are currently available for the following protocols:

- **DCE-RPC**
- **Fibre Channel**
- **ITU-T H.225 RAS**
- **LDAP**
- **MGCP**
- **ONC-RPC**
- **SMB**

As an example, the DCE-RPC service response time is described in more detail.

Note!

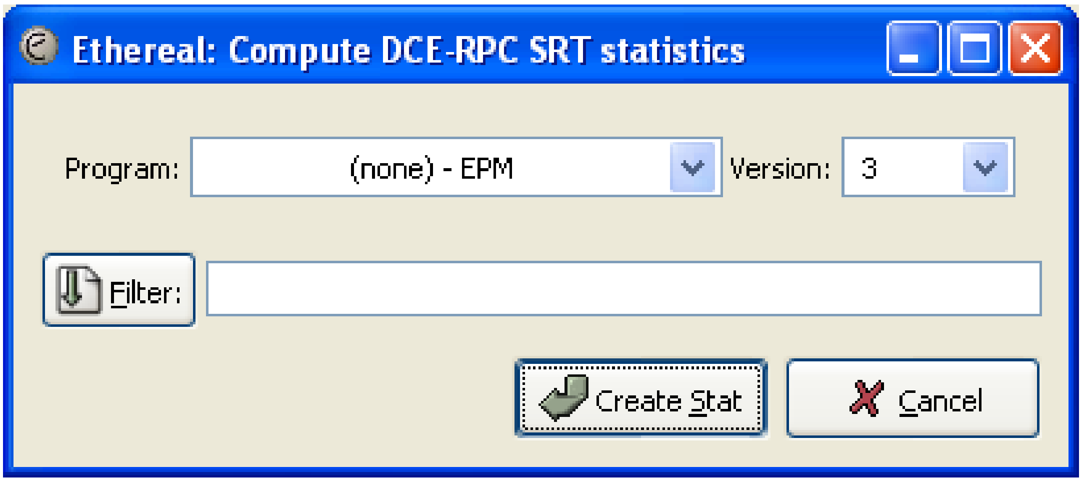
The other Service Response Time windows will work the same way (or only sligthly different) compared to the following description.

8.7.1. 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



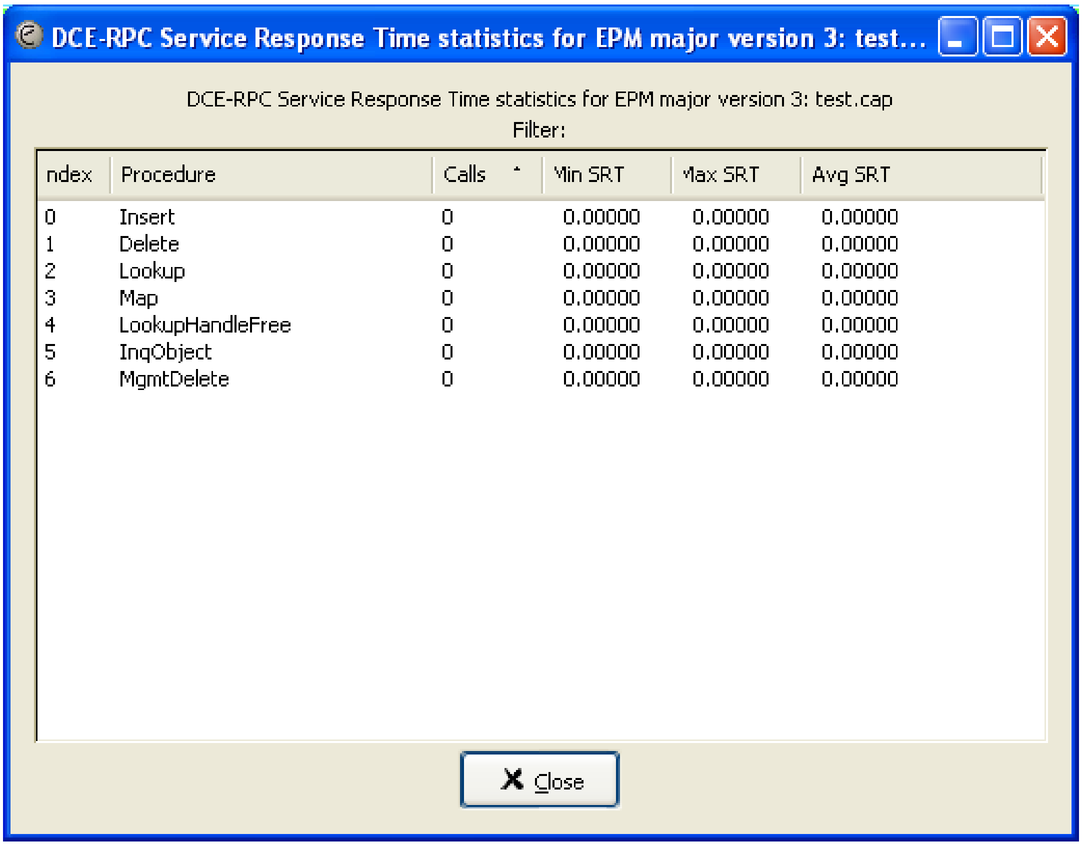
Ethereal: Compute DCE-RPC SRT statistics

Program: (none) - EPM

Version: 3

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

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


DCE-RPC Service Response Time statistics for EPM major version 3: test...

DCE-RPC Service Response Time statistics for EPM major version 3: test.cap

Filter:

ndex	Procedure	Calls	Min SRT	Max SRT	Avg SRT
0	Insert	0	0.00000	0.00000	0.00000
1	Delete	0	0.00000	0.00000	0.00000
2	Lookup	0	0.00000	0.00000	0.00000
3	Map	0	0.00000	0.00000	0.00000
4	LookupHandleFree	0	0.00000	0.00000	0.00000
5	InqObject	0	0.00000	0.00000	0.00000
6	MgmtDelete	0	0.00000	0.00000	0.00000

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.

8.8. The protocol specific statistics windows

The protocol specific statistics windows display detailed information of specific protocols and might be described in a later version of this document.

Chapter 9. Customizing Ethereal

9.1. Introduction

Ethereal's default behaviour will usually suit your needs pretty well. However, as you become more familiar with Ethereal, it can be customized in various ways to suit your needs even better. In this chapter we explore:

- How to start Ethereal with command line parameters
- How to colorize the Ethereal display
- How to use the various preference settings

9.2. Start Ethereal from the command line

You can start Ethereal 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.

Ethereal supports a large number of command line parameters. To see what they are, simply enter the command **ethereal -h** and the help information shown in Example 9.1, “Help information available from Ethereal” (or something similar) should be printed.

Example 9.1. Help information available from Ethereal

```
This is GNU ethereal 0.10.5
Compiled with GTK+ 2.4.3, with GLib 2.4.2, with WinPcap (version unknown),
with libz 1.2.1, with libpcre 4.4, with Net-SNMP 5.1, with ADNS.

Running with WinPcap version 3.0 (packet.dll version 3, 1, 0, 20), based
on libpcap version 0.8 on Windows XP Service Pack 1, build 2600.

ethereal [ -vh ] [ -klLnpQS ] [ -a <capture autostop condition> ] ...
        [ -b <number of ringbuffer files>[:<duration>] ]
        [ -B <byte view height> ] [ -c <count> ] [ -f <capture filter> ]
        [ -i <interface> ] [ -m <medium font> ] [ -N <resolving> ]
        [ -o <preference setting> ] ... [ -P <packet list height> ]
        [ -r <infile> ] [ -R <read filter> ] [ -s <snaplen> ]
        [ -t <time stamp format> ] [ -T <tree view height> ]
        [ -w <savefile> ] [ -y <link type> ] [ -z <statistics string> ]
        [ <infile> ]
```

We will examine each of the command line options in turn.

The first thing to notice is that issuing the command **ethereal** by itself will bring up Ethereal. However, you can include as many of the command line parameters as you like. Their meanings are as follows (in alphabetical order): XXX - is the alphabetical order a good choice? Maybe better task based?

-a <capture autostop condition>

Specify a criterion that specifies when Ethereal is to stop writing to a capture file. The criterion is of the form test:value, where test is one of:

- **duration** Stop writing to a capture file after value of seconds have elapsed.
- **filesize** Stop writing to a capture file after it reaches a size of value kilobytes (where a kilobyte is 1000 bytes, not 1024 bytes).

-b <number of ringbuffer files>

If a maximum capture file size was specified, cause Ethereal to run in "ring buffer" mode, with the specified number of files. In "ring buffer" mode, Ethereal will write to several capture files. Their name is based on the number of the file and on the creation date and time.

When the first capture file fills up, Ethereal will switch to writing to the next file, until it fills up the last file, at which point it'll discard the data in the first file (unless 0 is specified, in which case, the number of files is unlimited) and start writing to that file and so on.

If the optional duration is specified, Ethereal will switch also to the next file when the specified number of seconds has elapsed even if the current file is not completely fills up.

-B <byte view height>

This option sets the initial height of the "Packet Bytes" pane. This pane is usually the bottom pane in the Ethereal display.

-c <count>

This option specifies the maximum number of packets to capture when capturing live data. It would be used in conjunction with the **-k** option.

-f <capture filter>

This option sets the initial capture filter expression to be used when capturing packets.

-h

The **-h** option requests Ethereal to print its version and usage instructions (as shown above) and exit.

-i <interface>

The **-i** option allows you to specify, from the command line, which interface packet capture should occur on if capturing packets.

An example would be: **ethereal -i eth0**.

To get a listing of all the interfaces you can capture on, use the command **ifconfig -a** or **netstat -i**. Unfortunately, some versions of UNIX do not support **ifconfig -a**, so you will have to use **netstat -i** in these cases.

-k

The **-k** option specifies that Ethereal should start capturing packets immediately. This option requires the use of the **-i** parameter to specify the interface that packet capture will occur from.

-l

This option turns on automatic scrolling if the packet list pane is being updated automatically as packets arrive during a capture (as specified by the **-S** flag).

-L

List the data link types supported by the interface and exit.

-m <medium font>

This option sets the name of the font used for most text displayed by Ethereal. XXX - add an example!

-n

Disable network object name resolution (such as hostname, TCP and UDP port names).

-N <resolving>

Turns on name resolving for particular types of addresses and port numbers; the argument is a string that may contain the letters **m** to enable MAC address resolution, **n** to enable network address resolution, and **t** to enable transport-layer port number resolution. This overrides **-n** if both **-N** and **-n** are present. The letter C enables concurrent (asynchronous) DNS lookups.

-o <preference settings>

Sets a preference value, overriding the default value and any value read from a preference file. The argument to the flag is a string of the form prefname:value, where prefname is the name of the preference (which is the same name that would appear in the preference file), and value is the value to which it should be set. Multiple instances of **-o <preference settings>** can be given on a single command line.

An example of setting a single preference would be:

ethereal -o mgcp.display_dissect_tree:TRUE

An example of setting multiple preferences would be:

ethereal -o mgcp.display_dissect_tree:TRUE -o mgcp.udp.callagent_port:2627

Tip!

You can get a list of all available preference strings from the preferences file, see Appendix A, *Configuration Files and Folders*.

-p

Don't put the interface into promiscuous mode. Note that the interface might be in promiscuous mode for some other reason; hence, -p cannot be used to ensure that the only traffic that is captured is traffic sent to or from the machine on which Ethereal is running, broadcast traffic, and multicast traffic to addresses received by that machine.

-P <packet list height>

This option sets the initial height of the "Packet List" pane, ie, the top pane.

-Q

This option forces Ethereal to exit when capturing is complete. It can be used with the **-c** option. It must be used in conjunction with the **-i** and **-w** options.

-r <infile>

This option provides the name of a capture file for Ethereal to read and display. This capture file can be in one of the formats Ethereal understands.

-R <read filter>

This option specifies a display filter to be applied when reading packets from a capture file. The syntax of this filter is that of the display filters discussed in Section 6.2, “Filtering packets while viewing”. Packets not matching the filter are discarded.

-s <snaplen>

This option specifies the snapshot length to use when capturing packets. Ethereal will only capture **<snaplen>** bytes of data for each packet.

-S

This option specifies that Ethereal will display packets as it captures them. This is done by capturing in one process and displaying them in a separate process. This is the same as "Update list of packets in real time" in the Capture Options dialog box.

-t <time stamp format>

This option sets the format of packet timestamps that are displayed in the packet list window. The format can be one of:

- **r** relative, which specifies timestamps are displayed relative to the first packet captured.
- **a** absolute, which specifies that actual times be displayed for all packets.
- **ad** absolute with date, which specifies that actual dates and times be displayed for all packets.

- **d** delta, which specifies that timestamps are relative to the previous packet.

-T <tree view height>
This option sets the initial height of the "Packet Details" pane.

-v
The **-v** option requests Ethereal to print out its version information and exit.

-w <savefile>
This option sets the name of the **savefile** to be used when saving a capture file.

-y <link type>
If a capture is started from the command line with -k, set the data link type to use while capturing packets. The values reported by -L are the values that can be used.

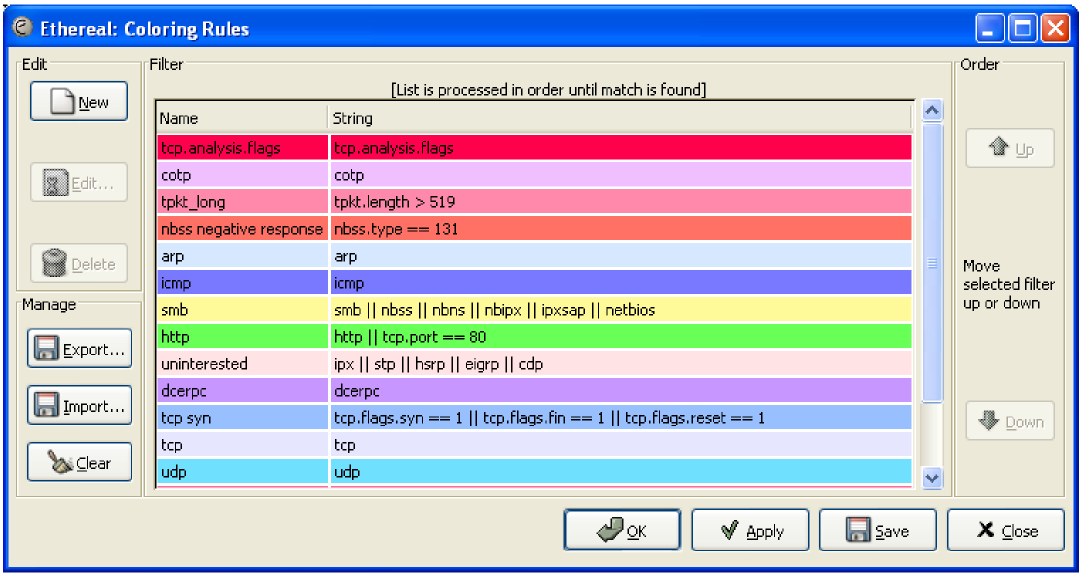
-z <statistics-string>
Get Ethereal to collect various types of statistics and display the result in a window that updates in semi-real time. XXX - add more details here!

9.3. Packet colorization

A very useful mechanism available in Ethereal is packet colorization. You can set Ethereal up so that it colorizes packets according to a filter. This allows you to emphasize the packets you are interested in.

To colorize packets, select the Coloring Rules... menu item from the View menu, and Ethereal will pop up the "Coloring Rules" dialog box as shown in Figure 9.1, “The "Coloring Rules" dialog box”.

Figure 9.1. The "Coloring Rules" dialog box



Name	String
tcp.analysis.flags	tcp.analysis.flags
cotp	cotp
tpkt_long	tpkt.length > 519
nbss negative response	nbss.type == 131
arp	arp
icmp	icmp
smb	smb || nbss || nbns || nbipx || ipxsap || netbios
http	http || tcp.port == 80
uninterested	ipx || stp || hsrp || eigrp || cdp
dcerpc	dcerpc
tcp syn	tcp.flags.syn == 1 || tcp.flags.fin == 1 || tcp.flags.reset == 1
tcp	tcp
udp	udp

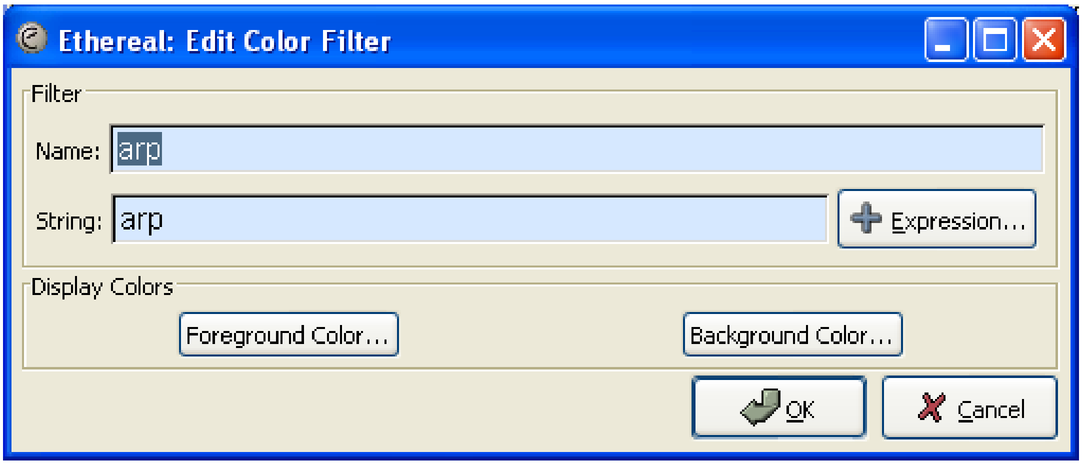
Once the Coloring Rules dialog box is up, there are a number of buttons you can use, depending on whether or not you have any color filters installed already.

Note!

You will need to carefully select the order that rules are listed (and thus applied) 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 be matching 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 9.2, “The "Edit Color Filter" dialog box”.

Figure 9.2. The "Edit Color Filter" dialog box


Ethereal: Edit Color Filter

Filter

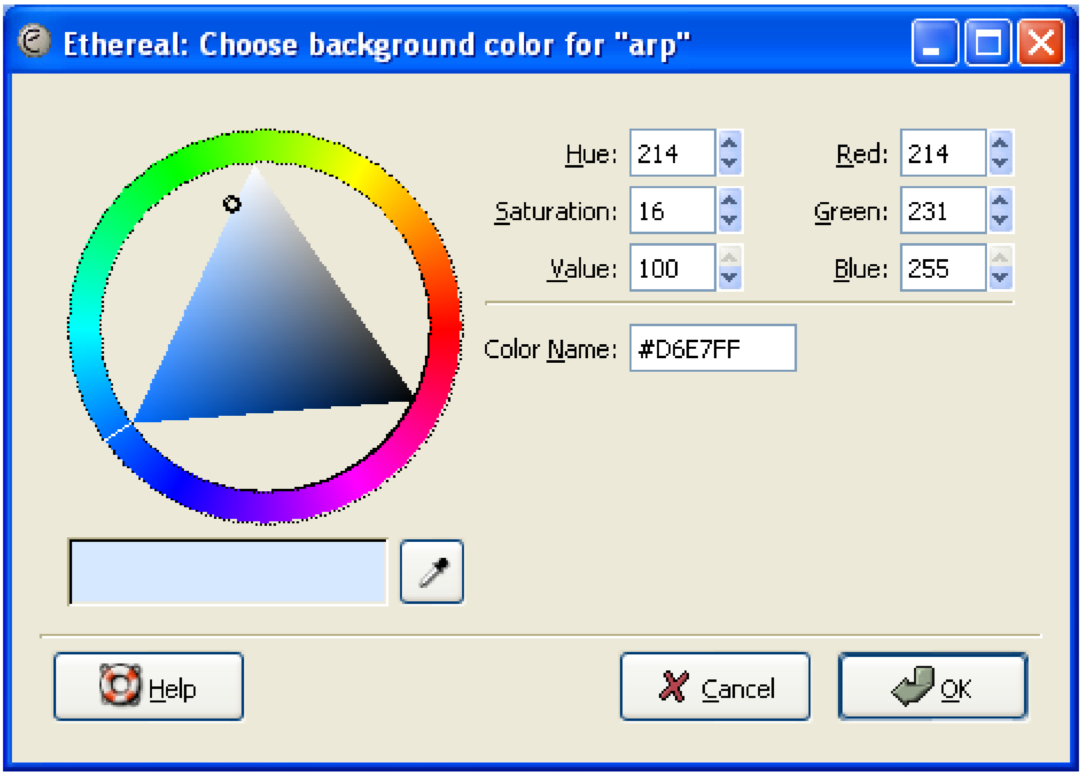
Name: arp

String: arp

Display Colors

In the Edit Color dialog box, simply enter a name for the color filter, and enter a filter string in the Filter text field. Figure 9.2, “The "Edit Color Filter" dialog box” shows the values **arp** and **arp** which means that the name of the color filter is **arp** and the filter will select protocols of type **arp**. Once you have entered these values, you can choose a foreground and background color for packets that match the filter expression. Click on **Foreground color...** or **Background color...** to achieve this and Ethereal will pop up the Choose foreground/background color for protocol dialog box as shown in Figure 9.3, “The "Choose color" dialog box”.

Figure 9.3. The "Choose color" dialog box


Ethereal: Choose background color for "arp"

Hue: 214

Saturation: 16

Value: 100

Red: 214

Green: 231

Blue: 255

Color Name: #D6E7FF

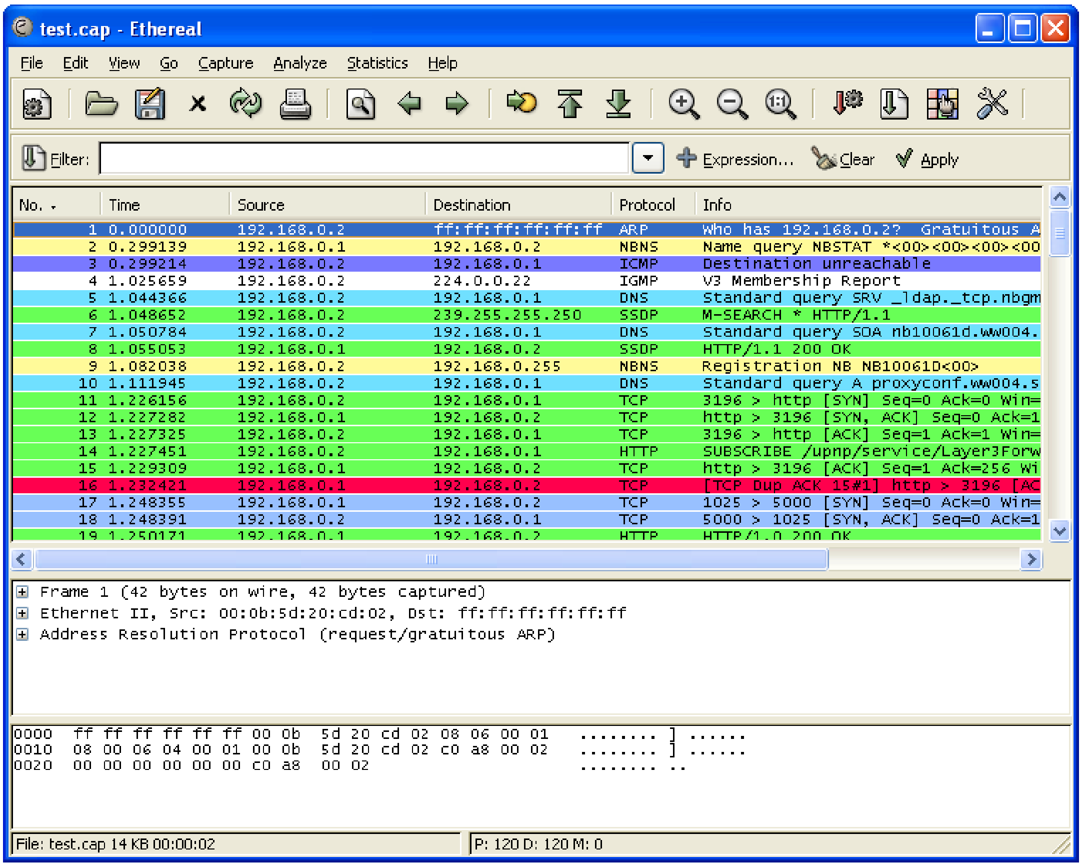
Select the color you desire for the selected packets and click on OK.

Note!

You must select a color in the colorbar next to the colorwheel to load values into the RGB values. Alternatively, you can set the values to select the color you want.

Figure 9.4, “Using color filters with Ethereal” shows an example of several color filters being used in Ethereal. You may not like the color choices, however, feel free to choose your own.

Figure 9.4. Using color filters with Ethereal



```
No. .    Time         Source         Destination        Protocol  Info
    1 0.000000    192.168.0.2    ff:ff:ff:ff:ff:ff  ARP       Who has 192.168.0.2?  Gratuitous A
    2 0.299139    192.168.0.1    192.168.0.2        NBNS      Name query NBSTAT *<00><00><00><00
    3 0.299214    192.168.0.2    192.168.0.1        ICMP      Destination unreachable
    4 1.025659    192.168.0.2    224.0.0.22         IGMP      V3 Membership Report
    5 1.044366    192.168.0.2    192.168.0.1        DNS       Standard query SRV _ldap._tcp.nbgm
    6 1.048652    192.168.0.2    239.255.255.250    SSDP      M-SEARCH * HTTP/1.1
    7 1.050784    192.168.0.2    192.168.0.1        DNS       Standard query SOA nb10061d.ww004.
    8 1.055053    192.168.0.1    192.168.0.2        SSDP      HTTP/1.1 200 OK
    9 1.082038    192.168.0.2    192.168.0.255      NBNS      Registration NB NB10061D<00>
   10 1.111945    192.168.0.2    192.168.0.1        DNS       Standard query A proxyconf.ww004.s
   11 1.226156    192.168.0.2    192.168.0.1        TCP       3196 > http [SYN] Seq=0 Ack=0 Win=
   12 1.227282    192.168.0.1    192.168.0.2        TCP       http > 3196 [SYN, ACK] Seq=0 Ack=1
   13 1.227325    192.168.0.2    192.168.0.1        TCP       3196 > http [ACK] Seq=1 Ack=1 Win=
   14 1.227451    192.168.0.2    192.168.0.1        HTTP      SUBSCRIBE /upnp/service/Layer3Forw
   15 1.229309    192.168.0.1    192.168.0.2        TCP       http > 3196 [ACK] Seq=1 Ack=256 Wi
   16 1.232421    192.168.0.1    192.168.0.2        TCP       [TCP Dup ACK 15#1] http > 3196 [AC
   17 1.248355    192.168.0.1    192.168.0.2        TCP       1025 > 5000 [SYN] Seq=0 Ack=0 Win=
   18 1.248391    192.168.0.2    192.168.0.1        TCP       5000 > 1025 [SYN, ACK] Seq=0 Ack=1
   19 1.250171    192.168.0.1    192.168.0.2        HTTP      HTTP/1.0 200 OK
```

```
⊞ Frame 1 (42 bytes on wire, 42 bytes captured)
⊞ Ethernet II, Src: 00:0b:5d:20:cd:02, Dst: ff:ff:ff:ff:ff:ff
⊞ Address Resolution Protocol (request/gratuitous ARP)
```

```
0000  ff ff ff ff ff ff 00 0b  5d 20 cd 02 08 06 00 01   ........ ] ......
0010  08 00 06 04 00 01 00 0b  5d 20 cd 02 c0 a8 00 02   ........ ] ......
0020  00 00 00 00 00 00 c0 a8  00 02                     ........ ..
```

File: test.cap 14 KB 00:00:02 | P: 120 D: 120 M: 0

9.4. Control Protocol dissection

There are some ways, to let the user control how protocols are dissected.

Each protocol has its own dissector, so dissecting a packet will typically involve several dissectors. As Ethereal tries to find the right dissector for each packet (using static "routes" and heuristics "guessing"), it might choose the wrong dissector in your specific case. For example, Ethereal won't know if you use a common protocol on an uncommon TCP port, e.g. using HTTP on TCP port 800 instead of the standard port 80.

There are two ways to control the relations between protocol dissectors: disable a protocol dissector completely or temporarily divert the way Ethereal calls the dissectors.

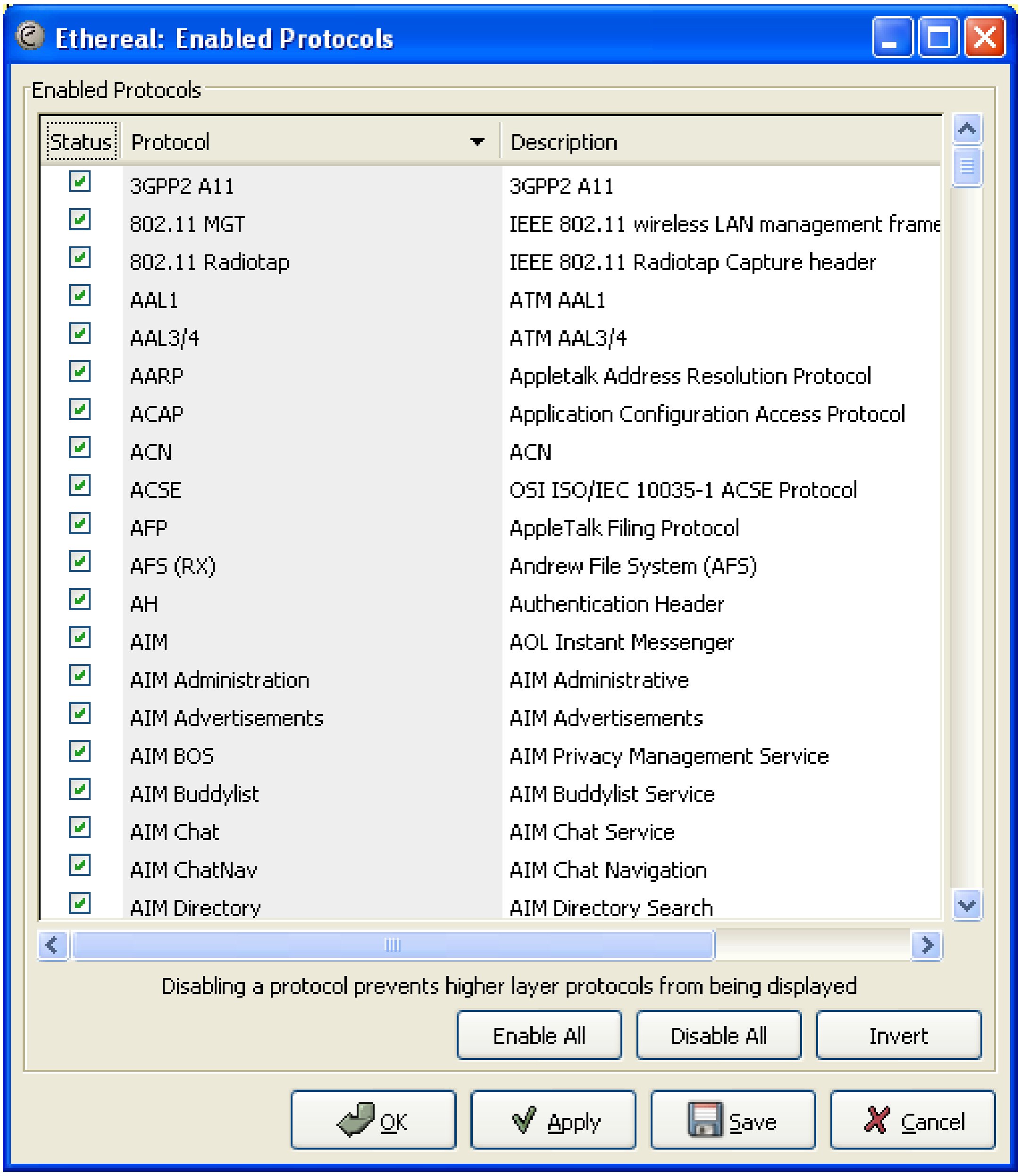
9.4.1. The "Enabled Protocols" dialog box

The Enabled Protocols dialog box lets you enable or disable specific protocols, all protocols are enabled by default. When a protocol is disabled, Ethereal stops processing a packet whenever that protocol is encountered.

Note!

Disabling a protocol will prevent information about higher-layer protocols from being displayed. For example, suppose you disabled the IP protocol and selected a packet containing Ethernet, IP, TCP, and HTTP information. The Ethernet information would be displayed, but the IP, TCP and HTTP information would not - disabling IP would prevent it and the other protocols from being displayed.

Figure 9.5. The "Enabled Protocols" dialog box



Status	Protocol	Description
☑	3GPP2 A11	3GPP2 A11
☑	802.11 MGT	IEEE 802.11 wireless LAN management frame
☑	802.11 Radiotap	IEEE 802.11 Radiotap Capture header
☑	AAL1	ATM AAL1
☑	AAL3/4	ATM AAL3/4
☑	AARP	Appletalk Address Resolution Protocol
☑	ACAP	Application Configuration Access Protocol
☑	ACN	ACN
☑	ACSE	OSI ISO/IEC 10035-1 ACSE Protocol
☑	AFP	AppleTalk Filing Protocol
☑	AFS (RX)	Andrew File System (AFS)
☑	AH	Authentication Header
☑	AIM	AOL Instant Messenger
☑	AIM Administration	AIM Administrative
☑	AIM Advertisements	AIM Advertisements
☑	AIM BOS	AIM Privacy Management Service
☑	AIM Buddylist	AIM Buddylist Service
☑	AIM Chat	AIM Chat Service
☑	AIM ChatNav	AIM Chat Navigation
☑	AIM Directory	AIM Directory Search

Disabling a protocol prevents higher layer protocols from being displayed

To disable or enable a protocol, simply click on it using the mouse or press the space bar when the protocol is highlighted.

Warning!

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

You can choose from the following actions:

1. **Enable All** Enable all protocols in the list.
2. **Disable All** Disable all protocols in the list.
3. **Invert** Toggle the state of all protocols in the list.
4. **OK** Apply the changes and close the dialog box.

5. **Apply** Apply the changes and keep the dialog box open.

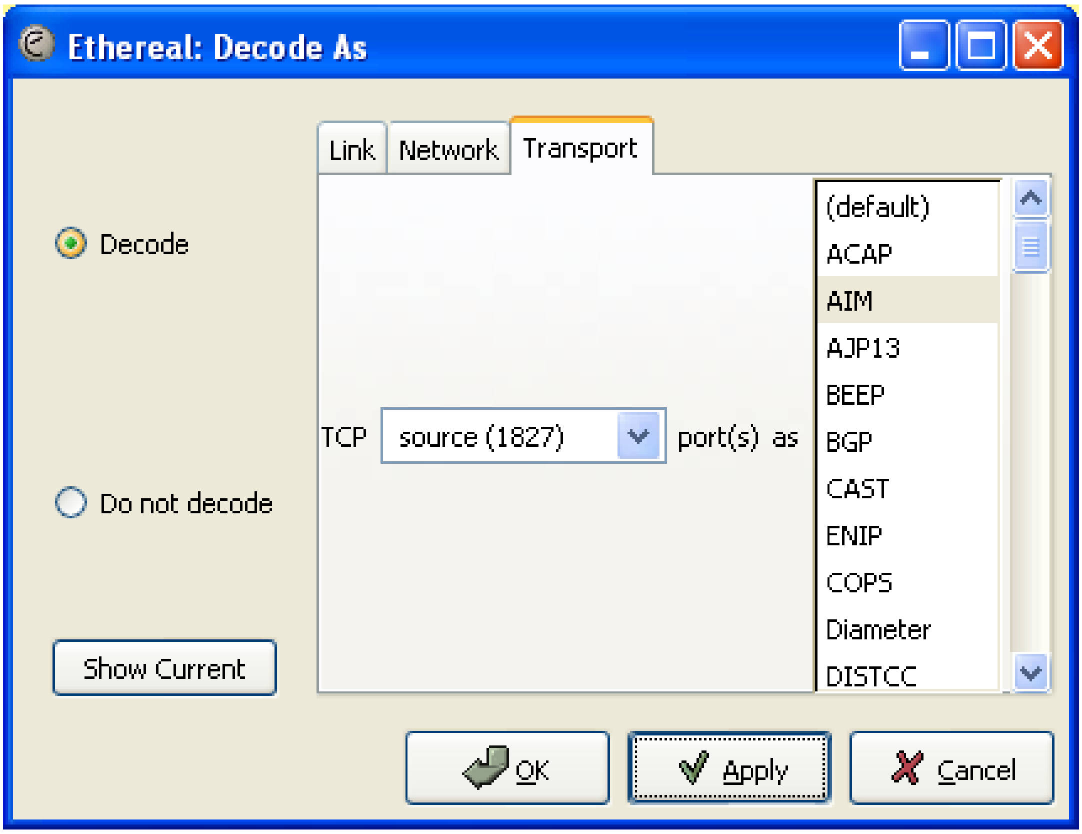
6. **Save** Save the settings to the disabled_protos, see *Appendix A, Configuration Files and Folders* for details.

7. **Cancel** Cancel the changes and close the dialog box.

9.4.2. User Specified Decodes

The "Decode As" functionality let you temporarily divert specific protocol dissections. This might be useful for example, if you do some uncommon things on your network.

Figure 9.6. The "Decode As" dialog box



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

Warning!

The user specified decodes can not be saved. If you quit Ethereal, these settings will be lost.

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

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

3. **Link/Network/Transport** Specify the way to decode packets. Which of these pages are available, depends on the content of the selected packet when this dialog box was 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.
7. **Cancel** Cancel the changes and close the dialog box.

9.4.3. Show User Specified Decodes

This dialog box shows the currently active user specified decodes.

Figure 9.7. The "Decode As: Show" dialog box

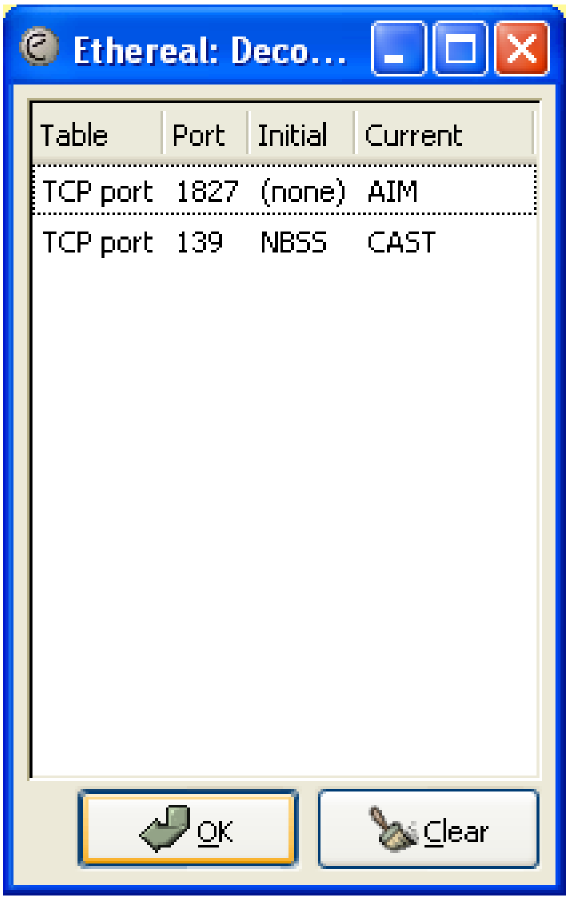


Table	Port	Initial	Current
TCP port	1827	(none)	AIM
TCP port	139	NBSS	CAST

1. **OK** Close this dialog box.
2. **Clear** Removes all user specified decodes.

9.5. Preferences

There are a number of preferences you can set. Simply select the Preferences... menu item from the Edit menu, and Ethereal will pop up the Preferences dialog box as shown in Figure 9.8, “The "User Interface" preferences page”, with the "User Interface" page as default. On the left side is a tree where you can select the page to be shown. XXX - add detailed descriptions of all the preferences pages.

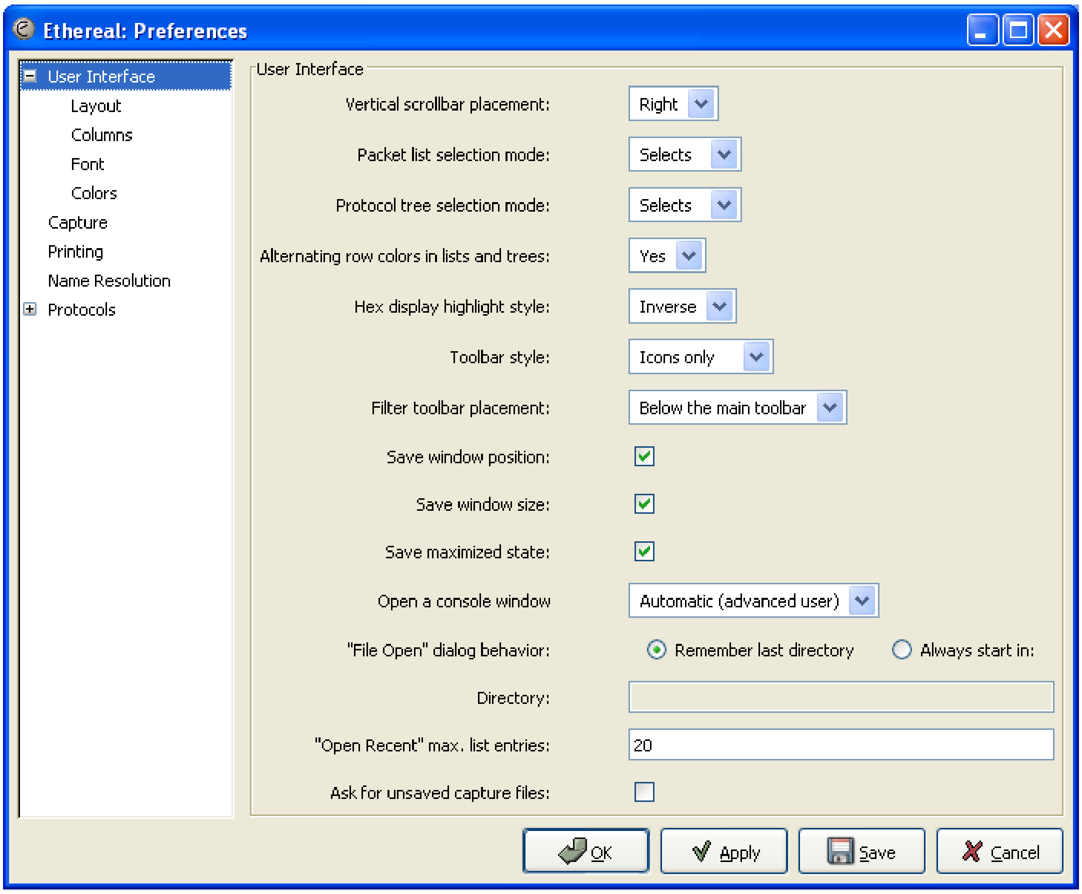
Warning!

The OK or Apply button will not save the preference settings, you'll have to save the settings by clicking the Save button.

- The **OK** button will apply the preferences settings and close the dialog.
- The **Apply** button will apply the preferences settings and keep the dialog open.
- The **Save** button will apply the preferences settings, save the settings on the harddisk and keep the dialog open.
- The **Cancel** button will restore all preferences settings to the last saved state.

9.5.1. The "User Interface" page

Figure 9.8. The "User Interface" preferences page



User Interface

Vertical scrollbar placement: Right

Packet list selection mode: Selects

Protocol tree selection mode: Selects

Alternating row colors in lists and trees: Yes

Hex display highlight style: Inverse

Toolbar style: Icons only

Filter toolbar placement: Below the main toolbar

Save window position: ☑

Save window size: ☑

Save maximized state: ☑

Open a console window Automatic (advanced user)

"File Open" dialog behavior: ● Remember last directory ○ Always start in:

Directory:

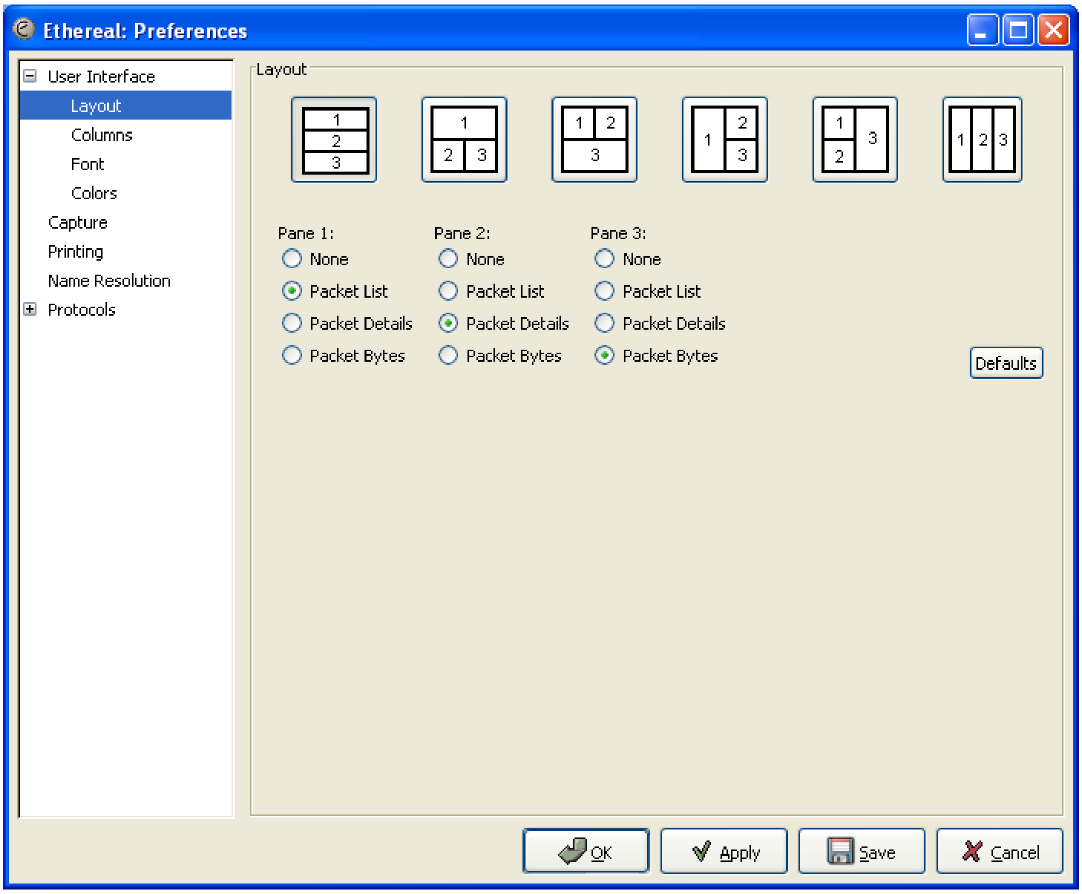
"Open Recent" max. list entries: 20

Ask for unsaved capture files: □

This page allows you to configure various characteristics of the GUI.

9.5.2. The "User Interface: Layout" page

Figure 9.9. The "User Interface: Layout" preferences page

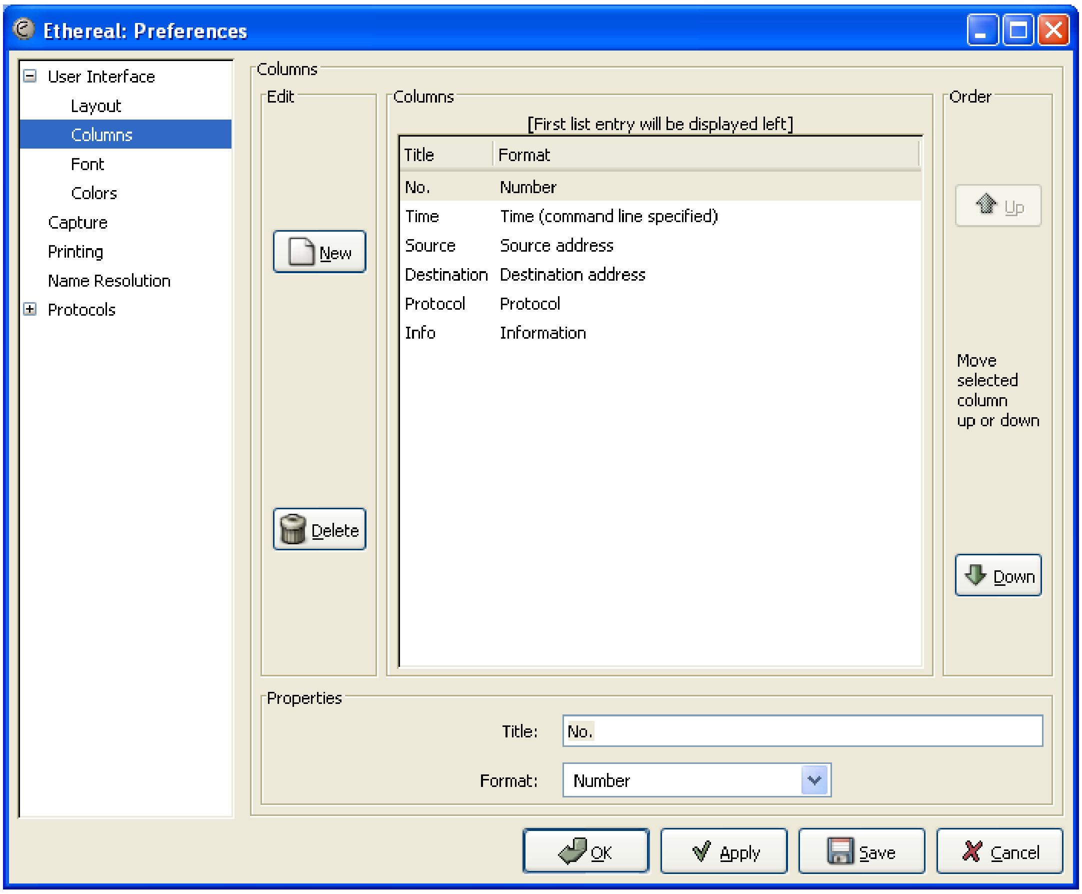


Pane 1:	Pane 2:	Pane 3:
○ None	○ None	○ None
● Packet List	○ Packet List	○ Packet List
○ Packet Details	● Packet Details	○ Packet Details
○ Packet Bytes	○ Packet Bytes	● Packet Bytes

This page selects the GUI layout of the main window.

9.5.3. The "User Interface: Columns" page

Figure 9.10. The "User Interface: Columns" preferences page



[First list entry will be displayed left]

Title	Format
No.	Number
Time	Time (command line specified)
Source	Source address
Destination	Destination address
Protocol	Protocol
Info	Information

Title: No.

Format: Number

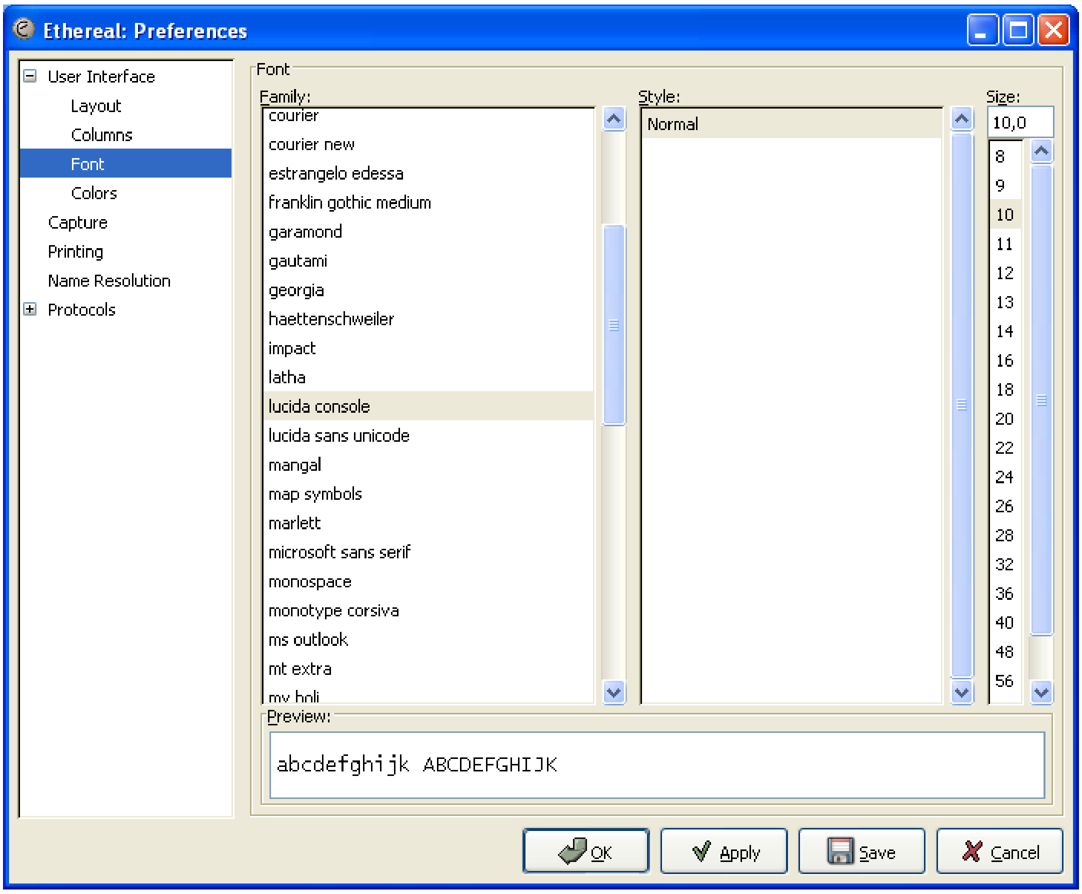
This page allows you to select which columns appear in the "Packet List" Pane.

Note!

Unlike all other preference changes, you will have to save the preferences and restart Ethereal in order for column changes to take effect!

9.5.4. The "User Interface: Font" page

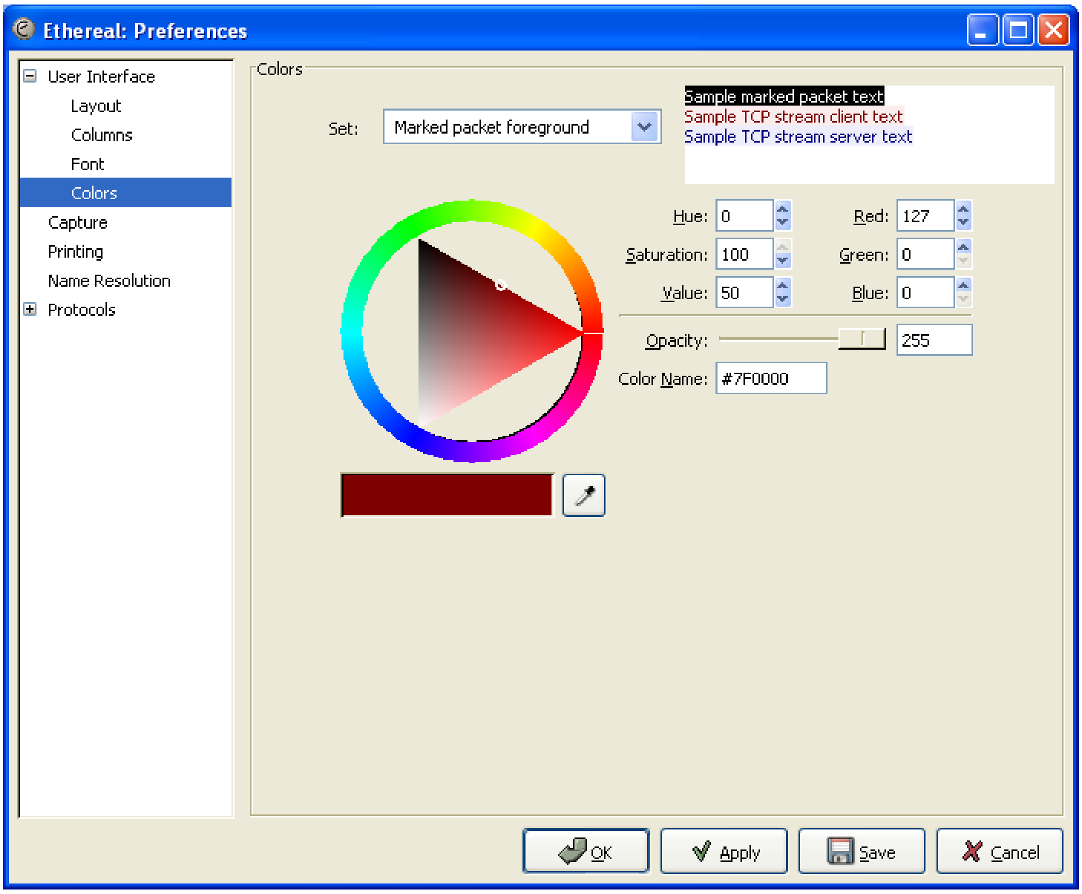
Figure 9.11. The "User Interface: Font" preferences page



This page allows you to select which font to use.

9.5.5. The "User Interface: Colors" page

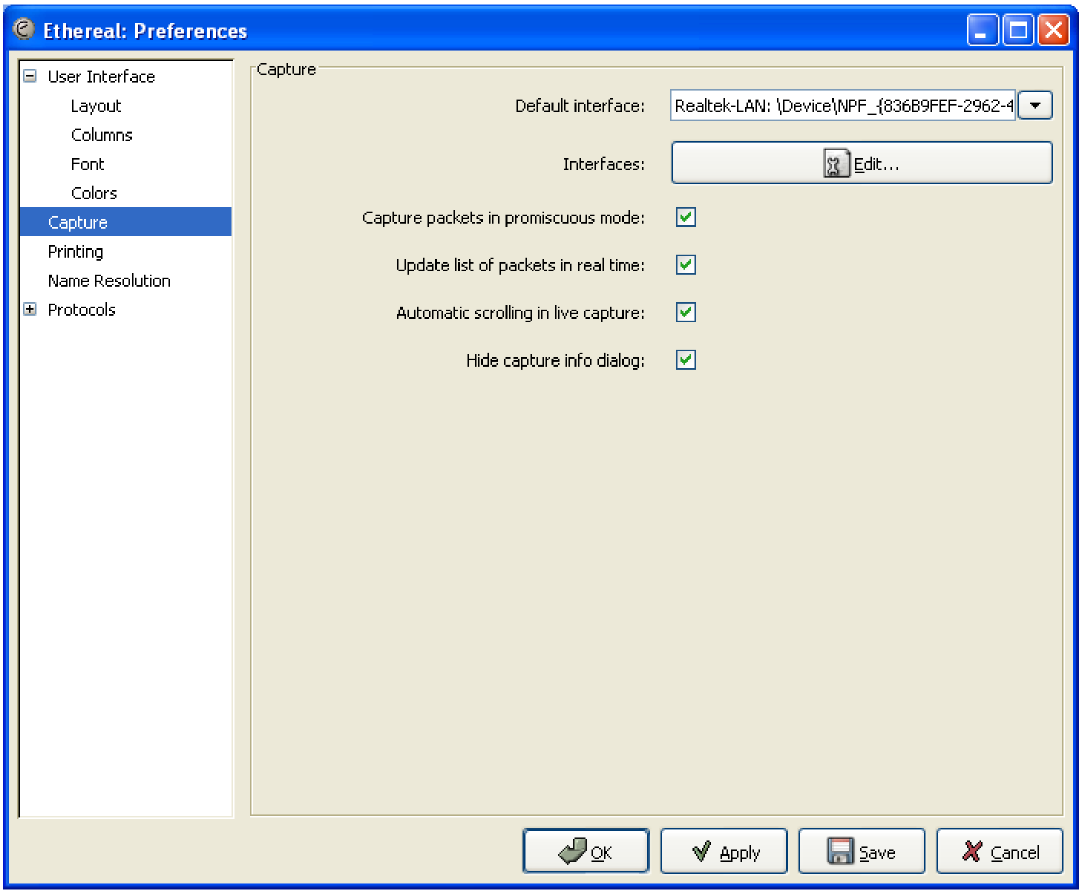
Figure 9.12. The "User Interface: Colors" preferences page



This page allows you to select which colors to use.

9.5.6. The "Capture" page

Figure 9.13. The "Capture" preferences page



Capture

Default interface: Realtek-LAN: \Device\NPF_{836B9FEF-2962-4

Interfaces: Edit...

Capture packets in promiscuous mode: ☑

Update list of packets in real time: ☑

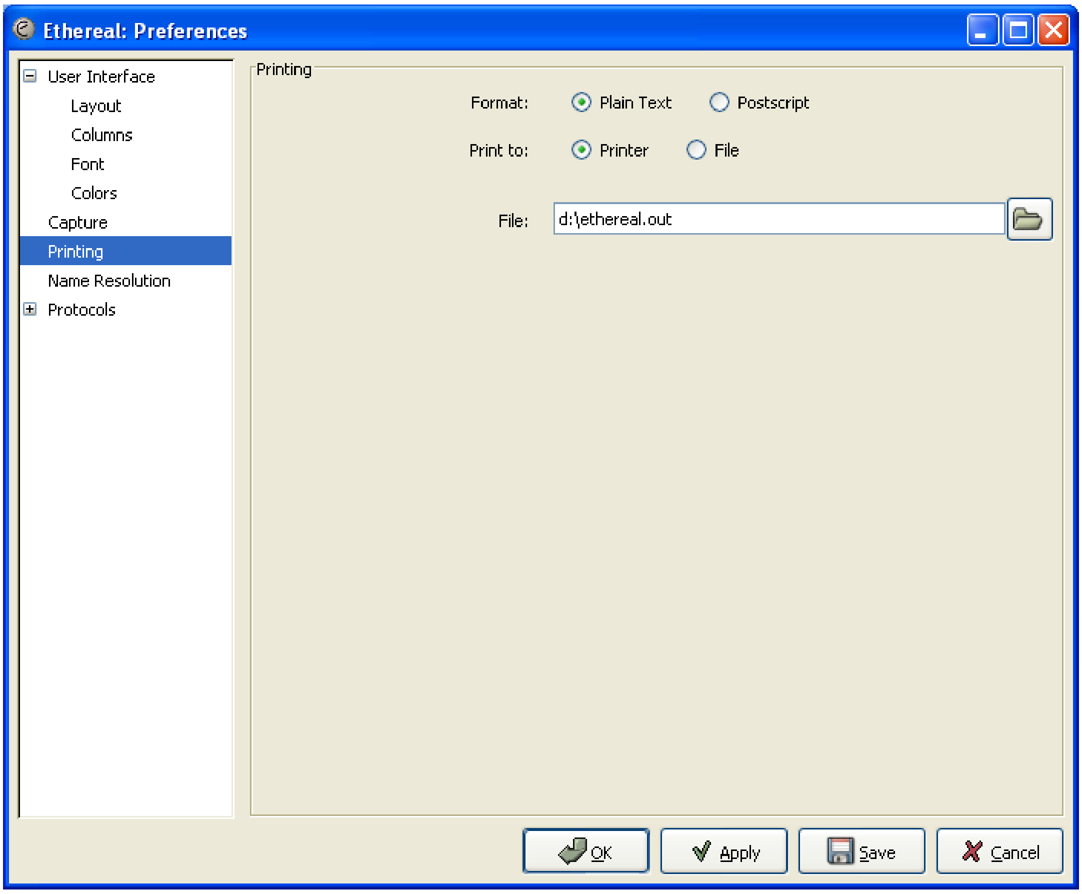
Automatic scrolling in live capture: ☑

Hide capture info dialog: ☑

This page allows you to select some defaults for the capture options dialog.

9.5.7. The "Printing" page

Figure 9.14. The "Printing" preferences page



Printing

Format: ● Plain Text ○ Postscript

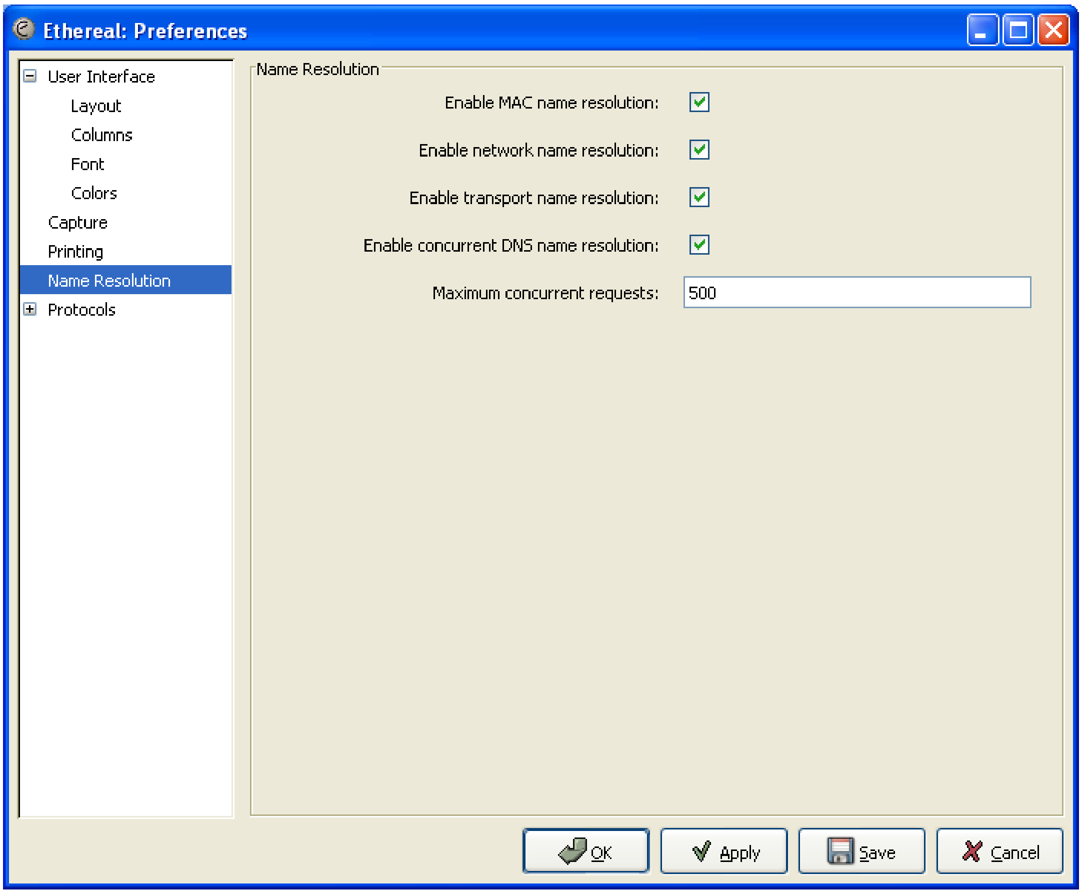
Print to: ● Printer ○ File

File: d:\ethereal.out

This page allows you to select some defaults for the print dialog.

9.5.8. The "Name Resolution" page

Figure 9.15. The "Name Resolution" preferences page



Name Resolution

Enable MAC name resolution: ☑

Enable network name resolution: ☑

Enable transport name resolution: ☑

Enable concurrent DNS name resolution: ☑

Maximum concurrent requests: 500

This page allows you to select some defaults for the name resolution.

9.5.9. The "Protocols" pages

These pages allows you to select settings for various protocols.

Appendix A. Configuration Files and Folders

Ethereal uses a number of files while it is running. Some of these reside in the personal configuration folder and are used to maintain information between runs of Ethereal, while some of them are maintained in system areas.

XXX - Add info about "temporary capture file" folders.

Tip

A list of the folders Ethereal actually uses can be found under the "Folders" tab in the "About" dialog box.

The content format of the configuration files is the same on all platforms. However, to match the different policies for unix and windows platforms, different folders for these files are used.

Table A.1. Configuration files overview

File	Description	Unix folders	Windows folders
preferences	Settings from the Preferences dialog box.	$HOME/.ethereal/preferences	%ETHEREAL%\preferences, %APPDATA%\Ethereal\preferences
recent	Recent GUI settings (e.g. recent files lists).	$HOME/.ethereal/recent	%APPDATA%\Ethereal\recent
cfilters	Capture filters.	$HOME/.ethereal/cfilters	%ETHEREAL%\cfilters, %APPDATA%\Ethereal\cfilters
dfilters	Display filters.	$HOME/.ethereal/dfilters	%ETHEREAL%\dfilters, %APPDATA%\Ethereal\dfilters
colorfilters	Coloring rules.	$HOME/.ethereal/colorfilters	%ETHEREAL%\colorfilters, %APPDATA%\Ethereal\colorfilters
disabled_protos	Disabled protocols.	$HOME/.ethereal/disabled_protos	%APPDATA%\Ethereal\disabled_protos
ethers	Ethernet name resolution.	/etc/ethers, $HOME/.ethereal/ethers	%ETHEREAL%\ethers, %APPDATA%\Ethereal\ethers
manuf	Ethernet name resolution.	/usr/local/etc/manuf	%ETHEREAL%\manuf
ipxnets	IPX name resolution.	$HOME/.ethereal/ipxnets	%ETHEREAL%\ipxnets
plugins	Plugin directories.	/usr/share/ethereal/plugins, /usr/local/share/ethereals/plugins, $HOME/.ethereal/plugins	%ETHEREAL%\plugins\<version>, %APPDATA%\Ethereal\plugins

Windows folders

%APPDATA% points to the personal configuration folder, typically "C:\Documents and Settings\<username>\Application Data", %ETHEREAL% points to the Ethereal program folder, typically "C:\Program Files\Ethereal"

preferences

This file contains your Ethereal preferences, including defaults for capturing and displaying packets. It is a simple text file containing statements of the form:

```
variable: value
```

The settings from this file are read in at program start and written to disk when you press the Save button in the "Preferences" dialog box.

recent

This file contains various GUI related settings like the main window position and size, the recent files list and such. It is a simple text file containing statements of the form:

```
variable: value
```

It is read at program start and written at program exit.

cfilters

This file contains all the capture filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
"<filter name>" <filter string>
```

The settings from this file are read in at program start and written to disk when you press the Save button in the "Capture Filters" dialog box.

dfilters

This file contains all the display filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
"<filter name>" <filter string>
```

The settings from this file are read in at program start and written to disk when you press the Save button in the "Display Filters" dialog box.

colorfilters

This file contains all the color filters that you have defined and saved. It consists of one or more lines, where each line has the following format:

```
@<filter name>@<filter string>
@[<bg RGB(16-bit)>][<fg RGB(16-bit)>]
```

The settings from this file are read in at program start and written to disk when you press the Save button in the "Coloring Rules" dialog box.

disabled_protos

Each line in this file specifies a disabled protocol name. The following are some examples:

```
tcp
udp
```

The settings from this file are read in at program start and written to disk when you press the Save button in the "Enabled Protocols" dialog box.

ethers

When Ethereal is trying to translate Ethernet hardware addresses to names, it consults the files listed in Table A.1, “Configuration files overview”. If an address is not found in /etc/ethers, Ethereal looks in $HOME/.ethereal/ethers

Each line in these files consists of one hardware address and name separated by whitespace. The digits of hardware addressses are separated by colons (:), dashes (-) or periods(.). The following are some examples:

```
ff-ff-ff-ff-ff-ff    Broadcast
c0-00-ff-ff-ff-ff    TR_broadcast
00.2b.08.93.4b.a1    Freds_machine
```

The settings from this file are read in at program start and never written by Ethereal.

manuf

Ethereal uses the files listed in Table A.1, “Configuration files overview” to translate the first three bytes of an Ethernet address into a manufacturers name. This file has the same format as the ethers file, except addresses are three bytes long.

The settings from this file are read in at program start and never written by Ethereal.

ipxnets

Ethereal uses the files listed in Table A.1, “Configuration files overview” to translate IPX network numbers into names.

An example is:

```
C0.A8.2C.00      HR
c0-a8-1c-00      CEO
00:00:BE:EF      IT_Server1
110f             FileServer3
```

The settings from this file are read in at program start and never written by Ethereal.

plugins

Ethereal searches for plugins in the directories listed in Table A.1, “Configuration files overview”. They are searched in the order listed.

Appendix B. Protocols and Protocol Fields

Ethereal distinguishes between protocols (e.g. tcp) and protocol fields (e.g. tcp.port).

A comprehensive list of all protocols and protocol fields can be found at: http://www.ethereal.com/docs/dfref/

XXX - update this protocols list.

For a quick reference, the list of available protocols is following:

- 802.1q Virtual LAN
- 802.1x Authentication
- AFS (4.0) Replication Server call declarations
- AOL Instant Messenger
- ATM
- ATM LAN Emulation
- Ad hoc On-demand Distance Vector Routing Protocol
- Ad hoc On-demand Distance Vector Routing Protocol v6
- Address Resolution Protocol
- Aggregate Server Access Protocol
- Andrew File System (AFS)
- Apache JServ Protocol v1.3
- AppleTalk Filing Protocol
- AppleTalk Session Protocol
- AppleTalk Transaction Protocol packet
- Appletalk Address Resolution Protocol
- Async data over ISDN (V.120)
- Authentication Header
- BACnet Virtual Link Control
- Banyan Vines
- Banyan Vines Fragmentation Protocol
- Banyan Vines SPP
- Blocks Extensible Exchange Protocol
- Boot Parameters
- Bootstrap Protocol

- Border Gateway Protocol
- Building Automation and Control Network APDU
- Building Automation and Control Network NPDU
- CDS Clerk Server Calls
- Check Point High Availability Protocol
- Checkpoint FW-1
- Cisco Auto-RP
- Cisco Discovery Protocol
- Cisco Group Management Protocol
- Cisco HDLC
- Cisco Hot Standby Router Protocol
- Cisco ISL
- Cisco Interior Gateway Routing Protocol
- Cisco NetFlow
- Cisco SLARP
- CoSine IPNOS L2 debug output
- Common Open Policy Service
- Common Unix Printing System (CUPS) Browsing Protocol
- DCE DFS Calls
- DCE Name Service
- DCE RPC
- DCE Security ID Mapper
- DCE/RPC BOS Server
- DCE/RPC CDS Solicitation
- DCE/RPC Conversation Manager
- DCE/RPC Endpoint Mapper
- DCE/RPC FLDB
- DCE/RPC FLDB UBIK TRANSFER
- DCE/RPC Kerberos V
- DCE/RPC RS_ACCT
- DCE/RPC RS_MISC
- DCE/RPC RS_UNIX
- DCE/RPC Remote Management

- DCE/RPC Repserver Calls
- DCE/RPC TokenServer Calls
- DCOM OXID Resolver
- DCOM Remote Activation
- DEC Spanning Tree Protocol
- DHCPv6
- DNS Control Program Server
- Data
- Data Link SWitching
- Data Stream Interface
- Datagram Delivery Protocol
- Diameter Protocol
- Distance Vector Multicast Routing Protocol
- Distributed Checksum Clearinghouse Prototocl
- Domain Name Service
- Dummy Protocol
- Dynamic DNS Tools Protocol
- Encapsulating Security Payload
- Enhanced Interior Gateway Routing Protocol
- Ethernet
- Extensible Authentication Protocol
- FTP Data
- FTServer Operations
- Fiber Distributed Data Interface
- File Transfer Protocol (FTP)
- Financial Information eXchange Protocol
- Frame
- Frame Relay
- GARP Multicast Registration Protocol
- GARP VLAN Registration Protocol
- GPRS Tunneling Protocol
- GPRS Tunnelling Protocol v0
- GPRS Tunnelling Protocol v1

- General Inter-ORB Protocol
- Generic Routing Encapsulation
- Generic Security Service Application Program Interface
- Gnutella Protocol
- Hummingbird NFS Daemon
- Hypertext Transfer Protocol
- ICQ Protocol
- IEEE 802.11 wireless LAN
- IEEE 802.11 wireless LAN management frame
- ILMI
- IP Payload Compression
- IPX Message
- IPX Routing Information Protocol
- ISDN Q.921-User Adaptation Layer
- ISDN User Part
- ISO 10589 ISIS InTRA Domain Routeing Information Exchange Protocol
- ISO 8073 COTP Connection-Oriented Transport Protocol
- ISO 8473 CLNP ConnectionLess Network Protocol
- ISO 8602 CLTP ConnectionLess Transport Protocol
- ISO 9542 ESIS Routeing Information Exchange Protocol
- ITU-T Recommendation H.261
- Inter-Access-Point Protocol
- Interbase
- Internet Cache Protocol
- Internet Content Adaptation Protocol
- Internet Control Message Protocol
- Internet Control Message Protocol v6
- Internet Group Management Protocol
- Internet Message Access Protocol
- Internet Printing Protocol
- Internet Protocol
- Internet Protocol Version 6
- Internet Relay Chat

- Internet Security Association and Key Management Protocol
- Internetwork Packet eXchange
- Java RMI
- Java Serialization
- Kerberos
- Kernel Lock Manager
- Label Distribution Protocol
- Layer 2 Tunneling Protocol
- Lightweight Directory Access Protocol
- Line Printer Daemon Protocol
- Link Access Procedure Balanced (LAPB)
- Link Access Procedure Balanced Ethernet (LAPBETHER)
- Link Access Procedure, Channel D (LAPD)
- Link Aggregation Control Protocol
- Link Management Protocol (LMP)
- Linux cooked-mode capture
- Local Management Interface
- LocalTalk Link Access Protocol
- Logical-Link Control
- Lucent/Ascend debug output
- MMS Message Encapsulation
- MS Proxy Protocol
- MSNIP: Multicast Source Notification of Interest Protocol
- MTP 2 Transparent Proxy
- MTP 2 User Adaptation Layer
- MTP 3 User Adaptation Layer
- MTP2 Peer Adaptation Layer
- Malformed Packet
- Message Transfer Part Level 2
- Message Transfer Part Level 3
- Microsoft Distributed File System
- Microsoft Exchange MAPI
- Microsoft Local Security Architecture

- Microsoft Network Logon
- Microsoft Registry
- Microsoft Security Account Manager
- Microsoft Server Service
- Microsoft Spool Subsystem
- Microsoft Telephony API Service
- Microsoft Windows Browser Protocol
- Microsoft Windows Lanman Remote API Protocol
- Microsoft Windows Logon Protocol
- Microsoft Workstation Service
- Mobile IP
- Modbus/TCP
- Mount Service
- MultiProtocol Label Switching Header
- Multicast Router DISCovery protocol
- Multicast Source Discovery Protocol
- NFSACL
- NFSAUTH
- NIS+
- NIS+ Callback
- NSPI
- NTLM Secure Service Provider
- Name Binding Protocol
- Name Management Protocol over IPX
- NetBIOS
- NetBIOS Datagram Service
- NetBIOS Name Service
- NetBIOS Session Service
- NetBIOS over IPX
- NetWare Core Protocol
- Network Data Management Protocol
- Network File System
- Network Lock Manager Protocol

- Network News Transfer Protocol
- Network Status Monitor CallBack Protocol
- Network Status Monitor Protocol
- Network Time Protocol
- Novell Distributed Print System
- Null/Loopback
- Open Shortest Path First
- OpenBSD Packet Filter log file
- PC NFS
- PPP Bandwidth Allocation Control Protocol
- PPP Bandwidth Allocation Protocol
- PPP CDP Control Protocol
- PPP Callback Control Protocol
- PPP Challenge Handshake Authentication Protocol
- PPP Compressed Datagram
- PPP Compression Control Protocol
- PPP IP Control Protocol
- PPP Link Control Protocol
- PPP MPLS Control Protocol
- PPP Multilink Protocol
- PPP Multiplexing
- PPP Password Authentication Protocol
- PPP VJ Compression
- PPP-over-Ethernet Discovery
- PPP-over-Ethernet Session
- PPPMux Control Protocol
- Point-to-Point Protocol
- Point-to-Point Tunnelling Protocol
- Portmap
- Post Office Protocol
- Pragmatic General Multicast
- Prism
- Privilege Server operations

- Protocol Independent Multicast
- Q.2931
- Q.931
- Quake II Network Protocol
- Quake III Arena Network Protocol
- Quake Network Protocol
- QuakeWorld Network Protocol
- Qualified Logical Link Control
- RFC 2250 MPEG1
- RIPng
- RPC Browser
- RSTAT
- RX Protocol
- Radio Access Network Application Part
- Radius Protocol
- Raw packet data
- Real Time Streaming Protocol
- Real-Time Transport Protocol
- Real-time Transport Control Protocol
- Registry Server Attributes Manipulation Interface
- Registry server administration operations.
- Remote Override interface
- Remote Procedure Call
- Remote Quota
- Remote Shell
- Remote Wall protocol
- Remote sec_login preauth interface.
- Resource ReserVation Protocol (RSVP)
- Rlogin Protocol
- Routing Information Protocol
- Routing Table Maintenance Protocol
- SADMIND
- SCSI

- SMB (Server Message Block Protocol)
- SMB MailSlot Protocol
- SMB Pipe Protocol
- SNA-over-Ethernet
- SNMP Multiplex Protocol
- SPNEGO-KRB5
- SPRAY
- SS7 SCCP-User Adaptation Layer
- SSCOP
- Secure Socket Layer
- Sequenced Packet eXchange
- Service Advertisement Protocol
- Service Location Protocol
- Session Announcement Protocol
- Session Description Protocol
- Session Initiation Protocol
- Short Frame
- Short Message Peer to Peer
- Signalling Connection Control Part
- Signalling Connection Control Part Management
- Simple Mail Transfer Protocol
- Simple Network Management Protocol
- Sinec H1 Protocol
- Skinny Client Control Protocol
- SliMP3 Communication Protocol
- Socks Protocol
- Spanning Tree Protocol
- Spnego
- Stream Control Transmission Protocol
- Syslog message
- Systems Network Architecture
- TACACS
- TACACS+

- TPKT
- Tabular Data Stream
- Telnet
- Time Protocol
- Time Service Provider Interfacer
- Time Synchronization Protocol
- Token-Ring
- Token-Ring Media Access Control
- Transmission Control Protocol
- Transparent Network Substrate Protocol
- Trivial File Transfer Protocol
- Universal Computer Protocol
- Unreassembled Fragmented Packet
- User Datagram Protocol
- Virtual Router Redundancy Protocol
- Virtual Trunking Protocol
- Web Cache Coordination Protocol
- Wellfleet Compression
- Who
- Windows 2000 DNS
- Wireless Session Protocol
- Wireless Transaction Protocol
- Wireless Transport Layer Security
- X Display Manager Control Protocol
- X.25
- X.25 over TCP
- X11
- Xyplex
- Yahoo Messenger Protocol
- Yellow Pages Bind
- Yellow Pages Passwd
- Yellow Pages Service
- Yellow Pages Transfer

- Zebra Protocol
- Zone Information Protocol
- iSCSI

Appendix C. Related command line tools

C.1. Introduction

Beside the Ethereal GUI application, there are some command line tools, which can be helpful for doing some more specialized things. These tools will be described in this chapter.

C.2. tcpdump: Capturing with tcpdump for viewing with Ethereal

There are occasions when you want to capture packets using **tcpdump** rather than **ethereal**, especially when you want to do a remote capture and do not want the network load associated with running Ethereal remotely (not to mention all the X traffic polluting your capture).

However, the default **tcpdump** parameters result in a capture file where each packet is truncated, because **tcpdump**, by default, does only capture the first 68 bytes of each packet.

To ensure that you capture complete packets, use the following command:

```
tcpdump -i <interface> -s 1500 -w <some-file>
```

You will have to specify the correct **interface** and the name of a **file** to save into. In addition, you will have to terminate the capture with ^C when you believe you have captured enough packets.

Note!

tcpdump is not part of the Ethereal distribution. You can get it from: http://www.tcpdump.org for various platforms.

C.3. tethereal: Terminal-based Ethereal

Tethereal is a terminal oriented version of ethereal designed for capturing and displaying packets when an interactive user interface isn't necessary or available. It supports the same options as **ethereal**. For more information on **tethereal**, see the manual pages (**man tethereal**).

C.4. editcap: Edit capture files

Included with Ethereal is a small utility called **editcap**, which is a command-line utility for working with capture files. Its main function is to remove packets from capture files, but it can also be used to convert capture files from one format to another, as well as print information about capture files.

Example C.1. Help information available from editcap

```
$ editcap.exe -h
Usage: editcap [-r] [-h] [-v] [-T <encap type>] [-F <capture type>]
               [-s <snaplen>] [-t <time adjustment>]
               <infile> <outfile> [ <record#>[-<record#>] ... ]
  where -r specifies that the records specified should be kept, not deleted,
                           default is to delete
        -v specifies verbose operation, default is silent
        -h produces this help listing.
        -T <encap type> specifies the encapsulation type to use:
            ether - Ethernet
            tr - Token Ring
            slip - SLIP
            ppp - PPP
            fddi - FDDI
            fddi-swapped - FDDI with bit-swapped MAC addresses
            rawip - Raw IP
            arcnet - ARCNET
            arcnet_linux - Linux ARCNET
            atm-rfc1483 - RFC 1483 ATM
            linux-atm-clip - Linux ATM CLIP
            lapb - LAPB
            atm-pdus - ATM PDUs
            atm-pdus-untruncated - ATM PDUs - untruncated
            null - NULL
            ascend - Lucent/Ascend access equipment
            isdn - ISDN
            ip-over-fc - RFC 2625 IP-over-Fibre Channel
            ppp-with-direction - PPP with Directional Info
            ieee-802-11 - IEEE 802.11 Wireless LAN
            prism - IEEE 802.11 plus Prism II monitor mode header
            ieee-802-11-radio - IEEE 802.11 Wireless LAN with radio information
            ieee-802-11-bsd - IEEE 802.11 plus BSD WLAN header
            ieee-802-11-avs - IEEE 802.11 plus AVS WLAN header
            linux-sll - Linux cooked-mode capture
            frelay - Frame Relay
            frelay-with-direction - Frame Relay with Directional Info
            chdlc - Cisco HDLC
            ios - Cisco IOS internal
            ltalk - Localtalk
            pflog-old - OpenBSD PF Firewall logs, pre-3.4
            hhdlc - HiPath HDLC
            docsis - Data Over Cable Service Interface Specification
            cosine - CoSine L2 debug log
            whdlc - Wellfleet HDLC
            sdlc - SDLC
            tzsp - Tazmen sniffer protocol
            enc - OpenBSD enc(4) encapsulating interface
            pflog - OpenBSD PF Firewall logs
            chdlc-with-direction - Cisco HDLC with Directional Info
            bluetooth-h4 - Bluetooth H4
            mtp2 - SS7 MTP2
            mtp3 - SS7 MTP3
            irda - IrDA
            user0 - USER 0
            user1 - USER 1
            user2 - USER 2
            user3 - USER 3
            user4 - USER 4
            user5 - USER 5
            user6 - USER 6
            user7 - USER 7
            user8 - USER 8
            user9 - USER 9
            user10 - USER 10
            user11 - USER 11
            user12 - USER 12
            user13 - USER 13
            user14 - USER 14
            user15 - USER 15
            symantec - Symantec Enterprise Firewall
            ap1394 - Apple IP-over-IEEE 1394
            bacnet-ms-tp - BACnet MS/TP
            default is the same as the input file
        -F <capture type> specifies the capture file type to write:
            libpcap - libpcap (tcpdump, Ethereal, etc.)
            rh6_1libpcap - RedHat Linux 6.1 libpcap (tcpdump)
            suse6_3libpcap - SuSE Linux 6.3 libpcap (tcpdump)
            modlibpcap - modified libpcap (tcpdump)
            nokialibpcap - Nokia libpcap (tcpdump)
            lanalyzer - Novell LANalyzer
            ngsniffer - Network Associates Sniffer (DOS-based)
            snoop - Sun snoop
            netmon1 - Microsoft Network Monitor 1.x
            netmon2 - Microsoft Network Monitor 2.x
            ngwsniffer_1_1 - Network Associates Sniffer (Windows-based) 1.1
            ngwsniffer_2_0 - Network Associates Sniffer (Windows-based) 2.00x
            visual - Visual Networks traffic capture
            5views - Accellent 5Views capture
            niobserverv9 - Network Instruments Observer version 9
            default is libpcap
        -s <snaplen> specifies that packets should be truncated to
           <snaplen> bytes of data
        -t <time adjustment> specifies the time adjustment
           to be applied to selected packets

            A range of records can be specified as well
```

Where each option has the following meaning:

-r

This option specifies that the frames listed should be kept, not deleted. The default is to delete the listed frames.

-h

This option provides help.

-v

This option specifies verbose operation. The default is silent operation.

-T {encap type}

This option specifies the frame encapsulation type to use.

It is mainly for converting funny captures to something that Ethereal can deal with.

The default frame encapsulation type is the same as the input encapsulation.

-F {capture type}

This option specifies the capture file format to write the output file in.

The default is libpcap format.

-s {snaplen} Specifies that packets should be truncated to {snaplen} bytes of data.

-t {time adjustment} Specifies the time adjustment to be applied to selected packets.

{infile} This parameter specifies the input file to use. It must be present.

{outfile} This parameter specifies the output file to use. It must be present.

[record#[-][record# ...]] This optional parameter specifies the records to include or exclude (depending on the **-r** option. You can specify individual records or a range of records.

C.5. mergecap: Merging multiple capture files into one with mergecap

Mergecap is a program that combines multiple saved capture files into a single output file specified by the -w argument. Mergecap knows how to read libpcap capture files, including those of tcpdump. In addition, Mergecap can read capture files from snoop (including Shomiti) and atmsnoop, LanAlyzer, Sniffer (compressed or uncompressed), Microsoft Network Monitor, AIX's iptrace, NetXray, Sniffer Pro, RADCOM's WAN/LAN analyzer, Lucent/Ascend router debug output, HP-UX's nettl, and the dump output from Toshiba's ISDN routers. There is no need to tell Mergecap what type of file you are reading; it will determine the file type by itself. Mergecap is also capable of reading any of these file formats if they are compressed using gzip. Mergecap recognizes this directly from the file; the '.gz' extension is not required for this purpose.

By default, it writes the capture file in libpcap format, and writes all of the packets in both input capture files to the output file. The -F flag can be used to specify the format in which to write the capture file; it can write the file in libpcap format (standard libpcap format, a modified format used by some patched versions of libpcap, the format used by Red Hat Linux 6.1, or the format used by SuSE Linux 6.3), snoop format, uncompressed Sniffer format, Microsoft Network Monitor 1.x format, and the format used by Windows-based versions of the Sniffer software.

Packets from the input files are merged in chronological order based on each frame's timestamp, unless the -a flag is specified. Mergecap assumes that frames within a single capture file are already stored in chronological order. When the -a flag is specified, packets are copied directly from each input file to the output file, independent of each frame's timestamp.

If the -s flag is used to specify a snapshot length, frames in the input file with more captured data than the specified snapshot length will have only the amount of data specified by the snapshot length written to the output file. This may be useful if the program that is to read the output file cannot handle packets larger than a certain size (for example, the versions of snoop in Solaris 2.5.1 and Solaris 2.6 appear to reject Ethernet frames larger than the standard Ethernet MTU, making them incapable of handling gigabit Ethernet captures if jumbo frames were used).

If the -T flag is used to specify an encapsulation type, the encapsulation type of the output capture file will be forced to the specified type, rather than being the type appropriate to the encapsulation type of the input capture file. Note that this merely forces the encapsulation type of the output file to be the specified type; the packet headers of the packets will not be translated from the encapsulation type of the input capture file to the specified encapsulation type (for example, it will not translate an Ethernet capture to an FDDI capture if an Ethernet capture is read and '-T fddi' is specified).

Example C.2. Help information available from mergecap

```
$ mergecap.exe -h
mergecap version 0.10.5
Usage: mergecap [-hva] [-s <snaplen>] [-T <encap type>]
          [-F <capture type>] -w <outfile> <infile> [...]

  where -h produces this help listing.
        -v verbose operation, default is silent
        -a files should be concatenated, not merged
             Default merges based on frame timestamps
        -s <snaplen>: truncate packets to <snaplen> bytes of data
        -w <outfile>: sets output filename to <outfile>
        -T <encap type> encapsulation type to use:
             ether - Ethernet
             tr - Token Ring
             slip - SLIP
             ppp - PPP
             fddi - FDDI
             fddi-swapped - FDDI with bit-swapped MAC addresses
          rawip - Raw IP
          arcnet - ARCNET
          arcnet_linux - Linux ARCNET
          atm-rfc1483 - RFC 1483 ATM
          linux-atm-clip - Linux ATM CLIP
          lapb - LAPB
          atm-pdus - ATM PDUs
          atm-pdus-untruncated - ATM PDUs - untruncated
          null - NULL
          ascend - Lucent/Ascend access equipment
          isdn - ISDN
          ip-over-fc - RFC 2625 IP-over-Fibre Channel
          ppp-with-direction - PPP with Directional Info
          ieee-802-11 - IEEE 802.11 Wireless LAN
          prism - IEEE 802.11 plus Prism II monitor mode header
          ieee-802-11-radio - IEEE 802.11 Wireless LAN with radio informatio
          ieee-802-11-bsd - IEEE 802.11 plus BSD WLAN header
          ieee-802-11-avs - IEEE 802.11 plus AVS WLAN header
          linux-sll - Linux cooked-mode capture
          frelay - Frame Relay
          frelay-with-direction - Frame Relay with Directional Info
          chdlc - Cisco HDLC
          ios - Cisco IOS internal
          ltalk - Localtalk
          pflog-old - OpenBSD PF Firewall logs, pre-3.4
          hhdlc - HiPath HDLC
          docsis - Data Over Cable Service Interface Specification
          cosine - CoSine L2 debug log
          whdlc - Wellfleet HDLC
          sdlc - SDLC
          tzsp - Tazmen sniffer protocol
          enc - OpenBSD enc(4) encapsulating interface
          pflog - OpenBSD PF Firewall logs
          chdlc-with-direction - Cisco HDLC with Directional Info
          bluetooth-h4 - Bluetooth H4
          mtp2 - SS7 MTP2
          mtp3 - SS7 MTP3
          irda - IrDA
          user0 - USER 0
          user1 - USER 1
          user2 - USER 2
          user3 - USER 3
          user4 - USER 4
          user5 - USER 5
          user6 - USER 6
          user7 - USER 7
          user8 - USER 8
          user9 - USER 9
          user10 - USER 10
          user11 - USER 11
          user12 - USER 12
          user13 - USER 13
          user14 - USER 14
          user15 - USER 15
          symantec - Symantec Enterprise Firewall
          ap1394 - Apple IP-over-IEEE 1394
          bacnet-ms-tp - BACnet MS/TP
          default is the same as the first input file
     -F <capture type> capture file type to write:
          libpcap - libpcap (tcpdump, Ethereal, etc.)
          rh6_1libpcap - RedHat Linux 6.1 libpcap (tcpdump)
          suse6_3libpcap - SuSE Linux 6.3 libpcap (tcpdump)
          modlibpcap - modified libpcap (tcpdump)
          nokialibpcap - Nokia libpcap (tcpdump)
          lanalyzer - Novell LANalyzer
          ngsniffer - Network Associates Sniffer (DOS-based)
          snoop - Sun snoop
          netmon1 - Microsoft Network Monitor 1.x
          netmon2 - Microsoft Network Monitor 2.x
```

```
             ngwsniffer_1_1 - Network Associates Sniffer (Windows-based) 1.1
             ngwsniffer_2_0 - Network Associates Sniffer (Windows-based) 2.00x
             visual - Visual Networks traffic capture
             5views - Accellent 5Views capture
             niobserverv9 - Network Instruments Observer version 9
             default is libpcap
```

-h Prints the version and options and exits.

-v Causes **mergecap** to print a number of messages while it's working.

-a Causes the frame timestamps to be ignored, writing all packets from the first input file followed by all packets from the second input file. By default, when **-a** is not specified, the contents of the input files are merged in chronological order based on each frame's timestamp. Note: when merging, mergecap assumes that packets within a capture file are already in chronological order.

-s Sets the snapshot length to use when writing the data.

-w Sets the output filename.

-T Sets the packet encapsulation type of the output capture file.

-F Sets the file format of the output capture file.

A simple example merging `dhcp-capture.libpcap` and `imap-1.libpcap` into `outfile.libpcap` is shown below.

Example C.3. Simple example of using mergecap

```
$ mergecap -w outfile.libpcap dhcp-capture.libpcap imap-1.libpcap
```

C.6. text2pcap: Converting ASCII hexdumps to network captures with text2pcap

There may be some occasions when you wish to convert a hex dump of some network traffic into a libpcap file.

Text2pcap is a program that reads in an ASCII hex dump and writes the data described into a libpcap-style capture file. text2pcap can read hexdumps withmultiple packets in them, and build a capture file of multiple packets. text2pcap is also capable of generating dummy Ethernet, IP and UDP headers, in order to build fully processable packet dumps from hexdumps of application-level data only.

Text2pcap understands a hexdump of the form generated by od -t x1. 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 - see -o), of more than two hex digits. Here is a sample dump that text2pcap can recognize:

```
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 is 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 characters. Any text after the bytes is ignored (e.g. the character dump). Any hex numbers in this text are also ignored. An offset of zero is indicative of starting a new packet, so a single text file with a series of hexdumps can be converted into a packet capture with multiple packets. Multiple packets are read in with timestamps differing by one second each. In general, short of these restrictions, text2pcap is pretty liberal about reading in hexdumps and has been tested with a variety of mangled outputs (including being forwarded through email multiple times, with limited line wrap etc.)

There are a couple of other special features to note. Any line where the first non-whitespace character is '#' will be ignored as a comment. Any line beginning with #TEXT2PCAP is a directive and options can be inserted after this command to be processed by text2pcap. Currently there are no directives implemented; in the future, these may be used to give more fine grained control on the dump and the way it should be processed e.g. timestamps, encapsulation type etc.

Text2pcap also allows the user to read in dumps of application-level data, by inserting dummy L2, L3 and L4 headers before each packet. The user can elect to insert Ethernet headers, Ethernet and IP, or Ethernet, IP and UDP headers before each packet. This allows Ethereal or any other full-packet decoder to handle these dumps.

Example C.4. Help information available for text2pcap

```
$ text2pcap.exe -h

Usage: text2pcap.exe [-h] [-d] [-q] [-o h|o] [-l typenum] [-e l3pid] [-i proto]
          [-m max-packet] [-u srcp,destp] [-T srcp,destp] [-s srcp,destp,tag]
          [-S srcp,destp,tag] [-t timefmt] <input-filename> <output-filename>
```

```
where <input-filename> specifies input filename (use - for standard input)
      <output-filename> specifies output filename (use - for standard output)

[options] are one or more of the following

 -h              : Display this help message
 -d              : Generate detailed debug of parser states
 -o hex|oct      : Parse offsets as (h)ex or (o)ctal. Default is hex
 -l typenum      : Specify link-layer type number. Default is 1 (Ethernet).
                   See net/bpf.h for list of numbers.
 -q              : Generate no output at all (automatically turns off -d)
 -e l3pid        : Prepend dummy Ethernet II header with specified L3PID (in
                   HEX)
                   Example: -e 0x800
 -i proto        : Prepend dummy IP header with specified IP protocol (in
                   DECIMAL).
                   Automatically prepends Ethernet header as well.
                   Example: -i 46
 -m max-packet   : Max packet length in output, default is 64000
 -u srcp,destp   : Prepend dummy UDP header with specified dest and source port
                   (in DECIMAL).
                   Automatically prepends Ethernet and IP headers as well
                   Example: -u 30,40
 -T srcp,destp   : Prepend dummy TCP header with specified dest and source port
                   (in DECIMAL).
                   Automatically prepends Ethernet and IP headers as well
                   Example: -T 50,60
 -s srcp,dstp,tag: Prepend dummy SCTP header with specified dest/source ports
                   and verification tag (in DECIMAL).
                   Automatically prepends Ethernet and IP headers as well
                   Example: -s 30,40,34
 -S srcp,dstp,ppi: Prepend dummy SCTP header with specified dest/source ports
                   and verification tag 0. It also prepends a dummy SCTP DATA
                   chunk header with payload protocol identifier ppi.
                   Example: -S 30,40,34
 -t timefmt      : Treats the text before the packet as a date/time code; the
                   specified argument is a format string of the sort supported
                   by strptime.
                   Example: The time "10:15:14.5476" has the format code
                   "%H:%M:%S."
                   NOTE:    The subsecond component delimiter must be specified
                            (.) but no pattern is required; the remaining numbe
                            is assumed to be fractions of a second.
```

-w <filename> Write the capture file generated by **text2pcap** to <filename>. The default is to write to standard output.

-h Display the help message

-d Displays debugging information during the process. Can be used multiple times to generate more debugging information.

-q Be completely quiet during the process.

-o hex|oct Specify the radix for the offsets (hex or octal). Defaults to hex. This corresponds to the **-A** option for od.

-l Specify the link-layer type of this packet. Default is Ethernet(1). See net/bpf.h for the complete list of possible encapsulations. Note that this option should be used if your dump is a complete hex dump of an encapsulated packet and you wish to specify the exact type of encapsulation. Example: -l 7 for ARCNet packets.

-e l3pid

Include a dummy Ethernet header before each packet. Specify the L3PID for the Ethernet header in hex. Use this option if your dump has Layer 3 header and payload (e.g. IP header), but no Layer 2 encapsulation. Example: -e 0x806 to specify an ARP packet.

For IP packets, instead of generating a fake Ethernet header you can also use -l 12 to indicate a raw IP packet to Ethereal. Note that -l 12 does not work for any non-IP Layer 3 packet (e.g. ARP), whereas generating a dummy Ethernet header with -e works for any sort of L3 packet.

-u srcport destport

Include dummy UDP headers before each packet. Specify the source and destination UDP ports for the packet in decimal. Use this option if your dump is the UDP payload of a packet but does not include any UDP, IP or Ethernet headers. Note that this automatically includes appropriate Ethernet and IP headers with each packet. Example: -u 1000 69 to make the packets look like TFTP/UDP packets.

C.7. idl2eth: Creating dissectors from Corba IDL files with idl2eth

In an ideal world idl2eth would be mentioned in the users guide in passing and documented in the developers guide. As the developers guide has not yet been completed it will be documented here.

C.7.1. What is it?

As you have probably guessed from the name, **idl2eth** takes a user specified IDL file and attempts to build a dissector that can decode the IDL traffic over GIOP. The resulting file is "C" code, that should compile okay as an ethereal dissector.

idl2eth basically parses the data struct given to it by the omniidl compiler, and using the GIOP API available in packet-giop.[ch], generates get_CDR_xxx calls to decode the CORBA traffic on the wire.

It consists of 4 main files.

`README.idl2eth`	This document
`ethereal_be.py`	The main compiler backend
`ethereal_gen.py`	A helper class, that generates the C code.
`idl2eth`	A simple shell script wrapper that the end user should use to generate the dissector from the IDL file(s).

C.7.2. Why do this?

It is important to understand what CORBA traffic looks like over GIOP/IIOP, and to help build a tool that can assist in troubleshooting CORBA interworking. This was especially the case after seeing a lot of discussions about how particular IDL types are represented inside an octet stream.

I have also had comments/feedback that this tool would be good for say a CORBA class when teaching students what CORBA traffic looks like "on the wire".

It is also COOL to work on a great Open Source project such as the case with "Ethereal" (http://www.ethereal.com)

C.7.3. How to use idl2eth

To use the idl2eth to generate ethereal dissectors, you need the following:

Prerequisites to using idl2eth

1. Python must be installed. See http://python.org/
2. omniidl from the the omniORB package must be available. See http://omniorb.sourceforge.net/
3. Of course you need ethereal installed to compile the code and tweak it if required. idl2eth is part of the standard Ethereal distribution

To use idl2eth to generate an ethereal dissector from an idl file use the following proceedure:

Proceedure for converting a Corba idl file into an ethereal dissector

1. To write the C code to stdout.

   ```
   idl2eth  <your file.idl>
   ```

   eg:

   ```
   idl2eth echo.idl
   ```

2. To write to a file, just redirect the output.

   ```
   idl2eth echo.idl > packet-test-idl.c
   ```

   You may wish to comment out the register_giop_user_module() code and that will leave you with heuristic dissection.

If you dont want to use the shell script wrapper, then try steps 3 or 4 instead.

3. To write the C code to stdout.

   ```
   Usage: omniidl  -p ./ -b ethereal_be <your file.idl>
   ```

   eg:

   ```
   omniidl  -p ./ -b ethereal_be echo.idl
   ```

4. To write to a file, just redirect the output.

   ```
   omniidl  -p ./ -b ethereal_be echo.idl > packet-test-idl.c
   ```

   You may wish to comment out the register_giop_user_module() code and that will leave you with heuristic dissection.

5. Copy the resulting C code to your ethereal src directory, edit the 2 make files to include the packet-test-idl.c

   ```
   cp packet-test-idl.c /dir/where/ethereal/lives/
   edit Makefile.am
   edit Makefile.nmake
   ```

6. Run configure

   ```
   ./configure (or ./autogen.sh)
   ```

7. Compile the code

   ```
   make
   ```

8. Good Luck !!

C.7.4. TODO

1. Exception code not generated (yet), but can be added manually.
2. Enums not converted to symbolic values (yet), but can be added manually.
3. Add command line options etc
4. More I am sure :-)

C.7.5. Limitations

See the TODO list inside `packet-giop.c`

C.7.6. Notes

1. The "-p ./" option passed to omniidl indicates that the ethereal_be.py and ethereal_gen.py are residing in the current directory. This may need tweaking if you place these files somewhere else.
2. If it complains about being unable to find some modules (eg tempfile.py), you may want to check if PYTHONPATH is set correctly. On my Linux box, it is PYTHONPATH=/usr/lib/python1.5/

Appendix D. This Document's License (GFDL)

D.1. The GNU Free Document Public Licence

D.1.1. Copyright

Version 1.1, March 2000

Copyright (C) 2000 Free Software Foundation, Inc. 59 Temple Place, Suite 330, Boston, MA 02111-1307 USA Everyone is permitted to copy and distribute verbatim copies of this license document, but changing it is not allowed.

D.1.2. Preamble

The purpose of this License is to make a manual, textbook, or other written document "free" in the sense of freedom: to assure everyone the effective freedom to copy and redistribute it, with or without modifying it, either commercially or noncommercially. Secondarily, this License preserves for the author and publisher a way to get credit for their work, while not being considered responsible for modifications made by others.

This License is a kind of "copyleft", which means that derivative works of the document must themselves be free in the same sense. It complements the GNU General Public License, which is a copyleft license designed for free software.

We have designed this License in order to use it for manuals for free software, because free software needs free documentation: a free program should come with manuals providing the same freedoms that the software does. But this License is not limited to software manuals; it can be used for any textual work, regardless of subject matter or whether it is published as a printed book. We recommend this License principally for works whose purpose is instruction or reference.

D.1.3. Applicability and Definitions

This License applies to any manual or other work that contains a notice placed by the copyright holder saying it can be distributed under the terms of this License. The "Document", below, refers to any such manual or work. Any member of the public is a licensee, and is addressed as "you".

A "Modified Version" of the Document means any work containing the Document or a portion of it, either copied verbatim, or with modifications and/or translated into another language.

A "Secondary Section" is a named appendix or a front-matter section of the Document that deals exclusively with the relationship of the publishers or authors of the Document to the Document's overall subject (or to related matters) and contains nothing that could fall directly within that overall subject. (For example, if the Document is in part a textbook of mathematics, a Secondary Section may not explain any mathematics.) The relationship could be a matter of historical connection with the subject or with related matters, or of legal, commercial, philosophical, ethical or political position regarding them.

The "Invariant Sections" are certain Secondary Sections whose titles are designated, as being those of Invariant Sections, in the notice that says that the Document is released under this License.

The "Cover Texts" are certain short passages of text that are listed, as Front-Cover Texts or Back-Cover Texts, in the notice that says that the Document is released under this License.

A "Transparent" copy of the Document means a machine-readable copy, represented in a format whose specification is available to the general public, whose contents can be viewed and edited directly and straightforwardly with generic text editors or (for images composed of pixels) generic paint programs or (for drawings) some widely available drawing editor, and that is suitable for input to text formatters or for automatic translation to a variety of formats suitable for input to text formatters. A copy made in an otherwise Transparent file format whose markup has been designed to thwart or discourage subsequent modification by readers is not Transparent. A copy that is not "Transparent" is called "Opaque".

Examples of suitable formats for Transparent copies include plain ASCII without markup, Texinfo input format, LaTeX input format, SGML or XML using a publicly available DTD, and standard-conforming simple HTML designed for human modification. Opaque formats include PostScript, PDF, proprietary formats that can be read and edited only by proprietary word processors, SGML or XML for which the DTD and/or processing tools are not generally available, and the machine-generated HTML produced by some word processors for output purposes only.

The "Title Page" means, for a printed book, the title page itself, plus such following pages as are needed to hold, legibly, the material this License requires to appear in the title page. For works in formats which do not have any title page as such, "Title Page" means the text near the most prominent appearance of the work's title, preceding the beginning of the body of the text.

D.1.4. Verbatim Copying

You may copy and distribute the Document in any medium, either commercially or noncommercially, provided that this License, the copyright notices, and the license notice saying this License applies to the Document are reproduced in all copies, and that you add no other conditions whatsoever to those of this License. You may not use technical measures to obstruct or control the reading or further copying of the copies you make or distribute. However, you may accept compensation in exchange for copies. If you distribute a large enough number of copies you must also follow the conditions in section 3.

You may also lend copies, under the same conditions stated above, and you may publicly display copies.

D.1.5. Copying in Quantity

If you publish printed copies of the Document numbering more than 100, and the Document's license notice requires Cover Texts, you must enclose the copies in covers that carry, clearly and legibly, all these Cover Texts: Front-Cover Texts on the front cover, and Back-Cover Texts on the back cover. Both covers must also clearly and legibly identify you as the publisher of these copies. The front cover must present the full title with all words of the title equally prominent and visible. You may add other material on the covers in addition. Copying with changes limited to the covers, as long as they preserve the title of the Document and satisfy these conditions, can be treated as verbatim copying in other respects.

If the required texts for either cover are too voluminous to fit legibly, you should put the first ones listed (as many as fit reasonably) on the actual cover, and continue the rest onto adjacent pages.

If you publish or distribute Opaque copies of the Document numbering more than 100, you must either include a machine-readable Transparent copy along with each Opaque copy, or state in or with each Opaque copy a publicly-accessible computer-network location containing a complete Transparent copy of the Document, free of added material, which the general network-using public has access to download anonymously at no charge using public-standard network protocols. If you use the latter option, you must take reasonably prudent steps, when you begin distribution of Opaque copies in quantity, to ensure that this Transparent copy will remain thus accessible at the stated location until at least one year after the last time you distribute an Opaque copy (directly or through your agents or retailers) of that edition to the public.

It is requested, but not required, that you contact the authors of the Document well before redistributing any large number of copies, to give them a chance to provide you with an updated version of the Document.

D.1.6. Modifications

You may copy and distribute a Modified Version of the Document under the conditions of sections 2 and 3 above, provided that you release the Modified Version under precisely this License, with the Modified Version filling the role of the Document, thus licensing distribution and modification of the Modified Version to whoever possesses a copy of it. In addition, you must do these things in the Modified Version:

- Use in the Title Page (and on the covers, if any) a title distinct from that of the Document, and from those of previous versions (which should, if there were any, be listed in the History section of the Document). You may use the same title as a previous version if the original publisher of that version gives permission.
- List on the Title Page, as authors, one or more persons or entities responsible for authorship of the modifications in the Modified Version, together with at least five of the principal authors of the Document (all of its principal authors, if it has less than five).
- State on the Title page the name of the publisher of the Modified Version, as the publisher.
- Preserve all the copyright notices of the Document.
- Add an appropriate copyright notice for your modifications adjacent to the other copyright notices.
- Include, immediately after the copyright notices, a license notice giving the public permission to use the Modified Version under the terms of this License, in the form shown in the Addendum below.
- Preserve in that license notice the full lists of Invariant Sections and required Cover Texts given in the Document's license notice.
- Include an unaltered copy of this License.
- Preserve the section entitled "History", and its title, and add to it an item stating at least the title, year, new authors, and publisher of the Modified Version as given on the Title Page. If there is no section entitled "History" in the Document, create one stating the title, year, authors, and publisher of the Document as given on its Title Page, then add an item describing the Modified Version as stated in the previous sentence.
- Preserve the network location, if any, given in the Document for public access to a Transparent copy of the Document, and likewise the network locations given in the Document for previous versions it was based on. These may be placed in the "History" section. You may omit a network location for a work that was published at least four years before the Document itself, or if the original publisher of the version it refers to gives permission.
- In any section entitled "Acknowledgements" or "Dedications", preserve the section's title, and preserve in the section all the substance and tone of each of the contributor acknowledgements and/or dedications given therein.
- Preserve all the Invariant Sections of the Document, unaltered in their text and in their titles. Section numbers or the equivalent are not considered part of the section titles.
- Delete any section entitled "Endorsements". Such a section may not be included in the Modified Version.
- Do not retitle any existing section as "Endorsements" or to conflict in title with any Invariant Section.

If the Modified Version includes new front-matter sections or appendices that qualify as Secondary Sections and contain no material copied from the Document, you may at your option designate some or all of these sections as invariant. To do this, add their titles to the list of Invariant Sections in the Modified Version's license notice. These titles must be distinct from any other section titles.

You may add a section entitled "Endorsements", provided it contains nothing but endorsements of your Modified Version by various parties--for example, statements of peer review or that the text has been approved by an organization as the authoritative definition of a standard.

You may add a passage of up to five words as a Front-Cover Text, and a passage of up to 25 words as a Back-Cover Text, to the end of the list of Cover Texts in the Modified Version. Only one passage of Front-Cover Text and one of Back-Cover Text may be added by (or through arrangements made by) any one entity. If the Document already includes a cover text for the same cover, previously added by you or by arrangement made by the same entity you are acting on behalf of, you may not add another; but you may replace the old one, on explicit permission from the previous publisher that added the old one.

The author(s) and publisher(s) of the Document do not by this License give permission to use their names for publicity for or to assert or imply endorsement of any Modified Version.

D.1.7. Combining Documents

You may combine the Document with other documents released under this License, under the terms defined in section 4 above for modified versions, provided that you include in the combination all of the Invariant Sections of all of the original documents, unmodified, and list them all as Invariant Sections of your combined work in its license notice.

The combined work need only contain one copy of this License, and multiple identical Invariant Sections may be replaced with a single copy. If there are multiple Invariant Sections with the same name but different contents, make the title of each such section unique by adding at the end of it, in parentheses, the name of the original author or publisher of that section if known, or else a unique number. Make the same adjustment to the section titles in the list of Invariant Sections in the license notice of the combined work.

In the combination, you must combine any sections entitled "History" in the various original documents, forming one section entitled "History"; likewise combine any sections entitled "Acknowledgements", and any sections entitled "Dedications". You must delete all sections entitled "Endorsements."

D.1.8. Collections of Documents

You may make a collection consisting of the Document and other documents released under this License, and replace the individual copies of this License in the various documents with a single copy that is included in the collection, provided that you follow the rules of this License for verbatim copying of each of the documents in all other respects.

You may extract a single document from such a collection, and distribute it individually under this License, provided you insert a copy of this License into the extracted document, and follow this License in all other respects regarding verbatim copying of that document.

D.1.9. Aggregation with Independent Works

A compilation of the Document or its derivatives with other separate and independent documents or works, in or on a volume of a storage or distribution medium, does not as a whole count as a Modified Version of the Document, provided no compilation copyright is claimed for the compilation. Such a compilation is called an "aggregate", and this License does not apply to the other self-contained works thus compiled with the Document, on account of their being thus compiled, if they are not themselves derivative works of the Document.

If the Cover Text requirement of section 3 is applicable to these copies of the Document, then if the Document is less than one quarter of the entire aggregate, the Document's Cover Texts may be placed on covers that surround only the Document within the aggregate. Otherwise they must appear on covers around the whole aggregate.

D.1.10. Translation

Translation is considered a kind of modification, so you may distribute translations of the Document under the terms of section 4. Replacing Invariant Sections with translations requires special permission from their copyright holders, but you may include translations of some or all Invariant Sections in addition to the original versions of these Invariant Sections. You may include a translation of this License provided that you also include the original English version of this License. In case of a disagreement between the translation and the original English version of this License, the original English version will prevail.

D.1.11. Termination

You may not copy, modify, sublicense, or distribute the Document except as expressly provided for under this License. Any other attempt to copy, modify, sublicense or distribute the Document is void, and will automatically terminate your rights under this License. However, parties who have received copies, or rights, from you under this License will not have their licenses terminated so long as such parties remain in full compliance.

D.1.12. Future Revisions of this License

The Free Software Foundation may publish new, revised versions of the GNU Free Documentation License from time to time. Such new versions will be similar in spirit to the present version, but may differ in detail to address new problems or concerns. See http://www.gnu.org/copyleft/.

Each version of the License is given a distinguishing version number. If the Document specifies that a particular numbered version of this License "or any later version" applies to it, you have the option of following the terms and conditions either of that specified version or of any later version that has been published (not as a draft) by the Free Software Foundation. If the Document does not specify a version number of this License, you may choose any version ever published (not as a draft) by the Free Software Foundation.