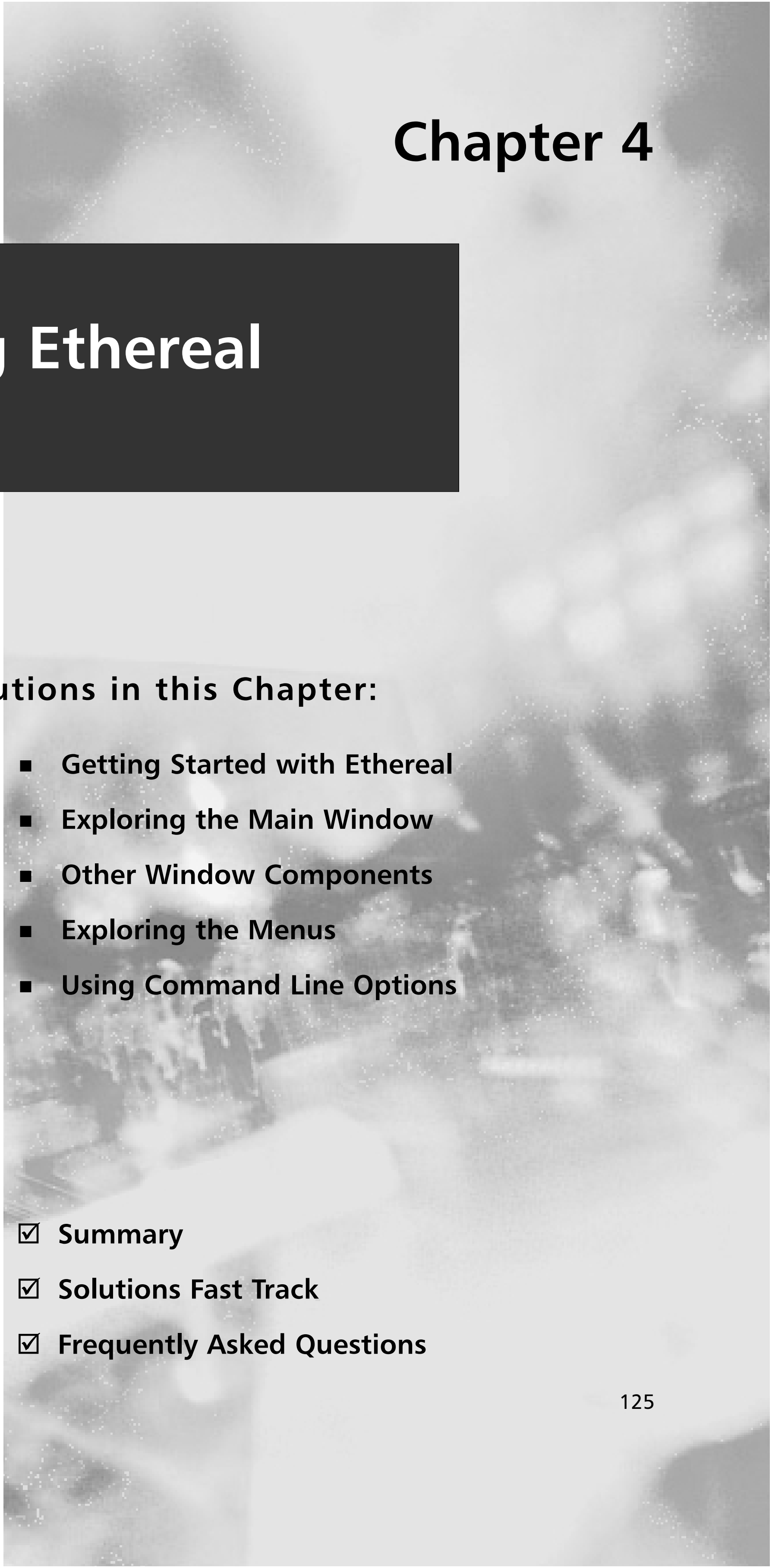


Chapter 4

Using Ethereal

Solutions in this Chapter:

- Getting Started with Ethereal
- Exploring the Main Window
- Other Window Components
- Exploring the Menus
- Using Command Line Options

☑ Summary

☑ Solutions Fast Track

☑ Frequently Asked Questions

Introduction

Ethereal is an invaluable tool for providing insight into what is really happening on your network. This kind of insight is useful when implementing protocols, debugging network applications, testing networks, or trying to debug live networks. In almost any situation that involves interaction with the network at a technical level you can enhance your understanding of your problem by using Ethereal to see what is really going on.

Ethereal is also an excellent pedagogical aid. Being able to see and analyze actual network traffic is extremely instructive. It's one thing to read about the functioning of a protocol, it's quite another to be able to watch it in action and explore how it's really operating

This chapter covers the main components of the Ethereal Graphical User Interface (GUI). These parts are the Main Window and it's subcomponents the Menu Bar, Tool Bar, Summary Window, Protocol Tree Window, Data View Window, Filter Bar, and Information field. This chapter will also cover use of the context-sensitive pop-up windows available in the Summary Window, Protocol Tree Window, and Data View Window, and will also explain the various dialog boxes that are launched by the menus and toolbars.

You will learn how to perform basic tasks in Ethereal such as capturing network traffic, loading and saving capture files, performing basic filtering, printing packets, and using some of the more involved tools provided by Ethereal. Examples have been provided to show you step-by-step how some of the less obvious areas of Ethereal work.

All of the screenshots for this chapter were taken on Linux, and so they may look slightly different in terms of fonts or window decorations than the screenshots in some of the other chapters in this book, which were taken on Windows.

Getting Started with Ethereal

In order to use Ethereal you must first acquire Ethereal and install it. If you are running a Linux distribution it is likely that your distribution shipped with Ethereal. If you are running Windows or some version of UNIX (Solaris, HP-UX, AIX, etc) you will have to download Ethereal and install it.

You can download binary packages for Ethereal from the Ethereal website at www.ethereal.com. If there are no binary packets there for your platform, if they are not up to date, or compiled without options you need, you may download the source code from the Ethereal website and compile Ethereal yourself. Installing Ethereal is covered in detail in Chapter 3.

You may also wish to download the bgp.pcap.gz capture at www.ethereal.com/sample/bgp.pcap.gz, as it is used in many examples in this chapter. You can follow along through the examples with your own copy of Ethereal. Once you have installed Ethereal on the computer you wish to capture with, execute at the command line:

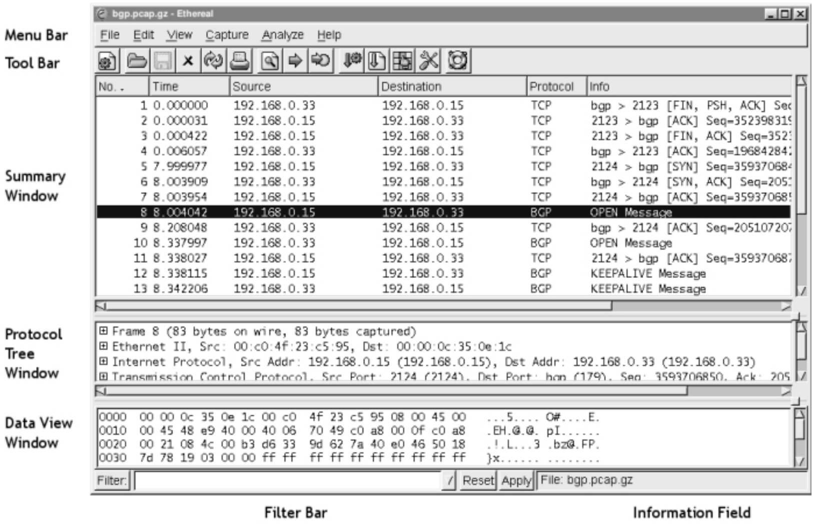
```
ethereal.
```

To launch Ethereal on Windows, select **Start | Programs | Ethereal | Ethereal**. The Main Window of the Ethereal application will now be displayed.

Exploring the Main Window

It is important for us to define a common set of labels for the different components of the Main Window so that we can speak clearly about them. Figure 4.1 shows the Main Window of Ethereal with its major components labeled.

Figure 4.1 Main Window



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.33	192.168.0.15	TCP	bgp > 2123 [FIN, PSH, ACK] Sec
2	0.000031	192.168.0.15	192.168.0.33	TCP	2123 > bgp [ACK] Seq=35239831
3	0.000422	192.168.0.15	192.168.0.33	TCP	2123 > bgp [FIN, ACK] Seq=352
4	0.006057	192.168.0.33	192.168.0.15	TCP	bgp > 2123 [ACK] Seq=1968428
5	7.999977	192.168.0.15	192.168.0.33	TCP	2124 > bgp [SYN] Seq=3593706
6	8.003909	192.168.0.33	192.168.0.15	TCP	bgp > 2124 [SYN, ACK] Seq=205
7	8.003954	192.168.0.15	192.168.0.33	TCP	2124 > bgp [ACK] Seq=3593706
8	8.004042	192.168.0.15	192.168.0.33	BGP	OPEN Message
9	8.208048	192.168.0.33	192.168.0.15	TCP	bgp > 2124 [ACK] Seq=2051072
10	8.337997	192.168.0.33	192.168.0.15	BGP	OPEN Message
11	8.338027	192.168.0.15	192.168.0.33	TCP	2124 > bgp [ACK] Seq=3593706
12	8.338115	192.168.0.15	192.168.0.33	BGP	KEEPALIVE Message
13	8.342206	192.168.0.33	192.168.0.15	BGP	KEEPALIVE Message

```
⊞ Frame 8 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 205
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
```

File: bgp.pcap.gz

The Main Window components are described in Table 4.1:

Table 4.1 Main Window Components

Window Component	Description
Menu Bar	A typical application menu bar containing drop-down menu items.
Tool Bar	A toolbar containing buttons for some commonly used functions of Ethereal. The Tool Bar icons have tool tips that are displayed when you pause the mouse pointer over them.
Summary Window	A window providing a one-line summary for each packet in the capture.
Protocol Tree Window	A window providing a detailed decode of the packet selected in the Summary Window.
Data View Window	A window providing a view of the raw data in the packet selected in the Summary Window.
Filter Bar	A tool for applying filters to the Summary Window to restrict which packets in the capture are displayed, based upon their attributes.
Information Field	A small display area to provide information about the capture or field selected in the Protocol Tree Window.

Summary Window

The Summary Window displays a summary of each packet in the capture, one per line. One or more columns of summary data for each packet will be displayed. Typical columns are shown in Table 4.2.

Table 4.2 Summary Window Columns

Column Name	Description
No.	Frame number within the capture.
Time	The time from beginning of the capture to the time when the packet was captured (in seconds).
Source	Highest level source address. This will frequently be the IP (Internet Protocol) source address, but may also be the Media Access Control (MAC) address for layer 2 (L2) Ethernet protocols (see the Ethereal Name Resolution sidebar for a discussion of MAC addresses), or other address types for other protocols (IPX, AppleTalk, etc).
Destination	Highest level destination address. This will frequently be the IP destination address, but may also be the MAC address for L2 Ethernet protocols, or other address types for other protocols (IPX, AppleTalk, etc).
Protocol	Typically the highest level protocol decoded. Examples include user level protocols such as HyperText Transfer Protocol (HTTP), File Transfer Protocol (FTP), Simple Mail Transfer Protocol (SMTP).
Info	This field contains information that was determined by the highest level decode to be useful or informative as part of a summary for this packet.

You can use the Preferences feature to select which columns are displayed in the Summary Window (select **Edit | Preferences** from the Menu Bar).

The summary information for the packet selected in the Summary Window in Figure 4.1 is shown in Table 4.3.

Table 4.3 Summary Window Column Example

Column Name	Value
No.	8
Time	8.004042 seconds since the capture started
Source	IP number 192.168.0.15
Destination	IP number 192.168.0.33

Continued

Table 4.3 Summary Window Column Example

Column Name	Value
Protocol	BGP (Border Gateway Protocol)
Info	OPEN Message

We can immediately see that this is a packet carrying a message for opening a Border Gateway Protocol session between 192.168.0.15 and 192.168.0.33. More information on Border Gateway Protocol is available in RFC (Request for Comment) 1771 at www.ietf.org/rfc/rfc1771.txt?number=1771. RFCs are the documents that describe the protocols in use on the Internet. Many RFCs define protocol standards (although not all RFCs do).

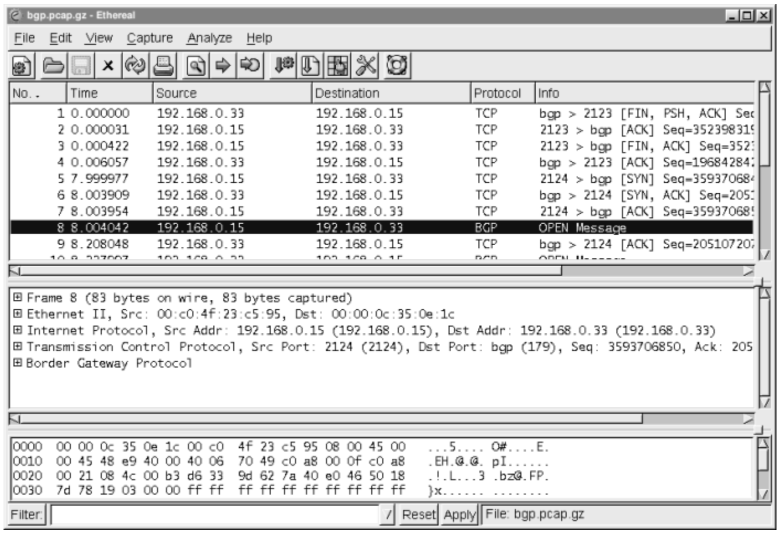
You can select packets in the Summary Window by clicking on the row summarizing a given packet. The information for the selected packet will then be displayed in the Protocol Tree Window and the Data View Window. Once you have selected a packet in the Summary Window you can use the Protocol Tree Window to drill down into it in greater detail.

Protocol Tree Window

We can conceptualize a packet as a tree of fields and subtrees. For each protocol there is a tree node summarizing the protocol, which can be expanded to provide the values in that protocol's fields. Within some protocols there may be tree nodes summarizing more complicated data structures in the protocol. These tree nodes can be expanded to show those data structures. For any given node that has a subtree, we can expand it's subtree to reveal more information, or collapse it to only show the summary. The Protocol Tree Window allows you to examine the tree created by Ethereal from decoding a packet.

We'll take a moment now to drill down through the Protocol Tree Window into the packet we selected in the previous example (Figure 4.2).

Figure 4.2 Protocol Tree Window Collapsed



```
No. .  Time       Source          Destination     Protocol  Info
     1 0.000000   192.168.0.33    192.168.0.15    TCP       bgp > 2123 [FIN, PSH, ACK] Sec
     2 0.000031   192.168.0.15    192.168.0.33    TCP       2123 > bgp [ACK] Seq=35239831
     3 0.000422   192.168.0.15    192.168.0.33    TCP       2123 > bgp [FIN, ACK] Seq=352
     4 0.006057   192.168.0.33    192.168.0.15    TCP       bgp > 2123 [ACK] Seq=19684284
     5 7.999977   192.168.0.15    192.168.0.33    TCP       2124 > bgp [SYN] Seq=35937068
     6 8.003909   192.168.0.33    192.168.0.15    TCP       bgp > 2124 [SYN, ACK] Seq=205
     7 8.003954   192.168.0.15    192.168.0.33    TCP       2124 > bgp [ACK] Seq=35937068
     8 8.004042   192.168.0.15    192.168.0.33    BGP       OPEN Message
     9 8.208048   192.168.0.33    192.168.0.15    TCP       bgp > 2124 [ACK] Seq=20510720
```

```
⊞ Frame 8 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 205
⊞ Border Gateway Protocol
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
```

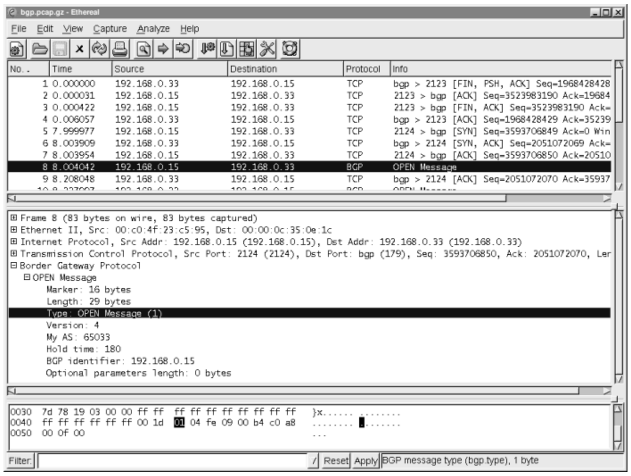
In the Protocol Tree Window, you can see that for each layer in the protocol stack for this packet we have a one-line summary of that layer (see Table 4.4).

Table 4.4 Protocol Layer Example

Layer	Protocol	Description
Packet Meta Data	Frame	83 bytes on wire, 83 bytes captured
Datalink (Layer 2/L2)	Ethernet II	Src Addr: 00:c0:4f:23:c5:95, Dst Addr: 00:00:0c:35:0e:1c
Network (Layer 3 / L3)	Internet Protocol (IP)	Src Addr: 192.168.0.15, Dst Addr: 192.168.0.33
Transport (Layer 4/ L4)	Transmission Control Protocol (TCP)	Src Port: 2124, Dst Port: bgp(179), Seq: 2593706850, Ack …
Application Layer (Layer 7 /L7)	Border Gateway Protocol (BGP)	

Each of these layers have little boxed plus (+) signs next to them indicating that they have a subtree that can be expanded to provide more information about that particular protocol.

Figure 4.3 Protocol Tree Window Expanded



```
No. .  Time      Source         Destination    Protocol Info
    1 0.000000  192.168.0.33   192.168.0.15   TCP      bgp > 2123 [FIN, PSH, ACK] Seq=1968428428
    2 0.000031  192.168.0.15   192.168.0.33   TCP      2123 > bgp [ACK] Seq=3523983190 Ack=19684
    3 0.000422  192.168.0.15   192.168.0.33   TCP      2123 > bgp [FIN, ACK] Seq=3523983190 Ack=
    4 0.006057  192.168.0.33   192.168.0.15   TCP      bgp > 2123 [ACK] Seq=1968428429 Ack=35239
    5 7.999977  192.168.0.15   192.168.0.33   TCP      2124 > bgp [SYN] Seq=3593706849 Ack=0 Win
    6 8.003909  192.168.0.33   192.168.0.15   TCP      bgp > 2124 [SYN, ACK] Seq=2051072069 Ack=
    7 8.003954  192.168.0.15   192.168.0.33   TCP      2124 > bgp [ACK] Seq=3593706850 Ack=20510
    8 8.004042  192.168.0.15   192.168.0.33   BGP      OPEN Message
    9 8.208048  192.168.0.33   192.168.0.15   TCP      bgp > 2124 [ACK] Seq=2051072070 Ack=35937
```

```
⊞ Frame 8 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051072070, Ler
⊟ Border Gateway Protocol
  ⊟ OPEN Message
      Marker: 16 bytes
      Length: 29 bytes
      Type: OPEN Message (1)
      Version: 4
      My AS: 65033
      Hold time: 180
      BGP identifier: 192.168.0.15
      Optional parameters length: 0 bytes
```

```
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                           ...
```

BGP message type (bgp.type), 1 byte

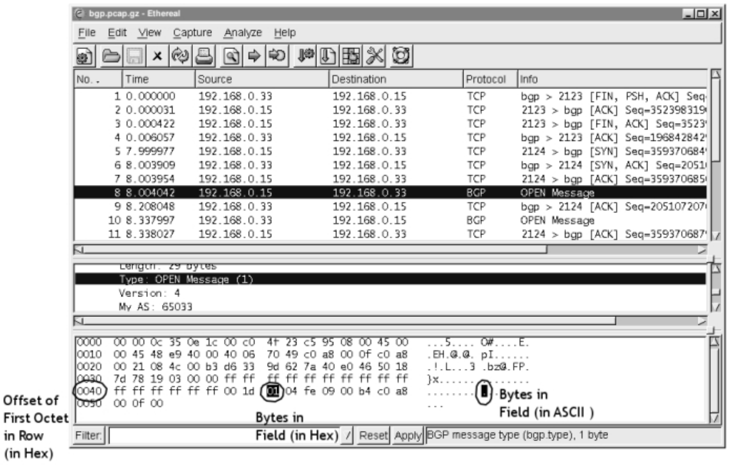
In Figure 4.3, we have expanded the **Border Gateway Protocol** tree to reveal that it contains one **OPEN Message**, and further expanded that **OPEN Message** to reveal the fields contained within it.

By selecting the **Type** field in the Protocol Tree Window, we've caused the **Information** field in the lower right corner to display the message *BGP message type (bgp.type), 1byte*. This indicates the long name of this field (*BGP message type*) and the display filter field name used to identify this field for filtering and colorization (*bgp.type*), as well as the size of this field in the packet (*1 byte*).

Data View Window

The Data View Window contains a series of rows, as show in Figure 4.4. Each row begins with a four-digit number representing the number of bytes the first octet (an octet is eight bits or one byte or two hexadecimal – also known as *hex* - digits) in that row is offset from the beginning of the packet. This offset is then followed by sixteen two-character hexadecimal bytes. The last item in each row is a series of sixteen ASCII (American Standard Code for Information Interchange, a common character encoding) characters representing the same 16 bytes from the packet. Not all bytes are conveniently displayable in ASCII. For those bytes a period (.) is substituted as a placeholder.

Figure 4.4 Data View Window



```
No. . Time      Source        Destination   Protocol Info
    1 0.000000  192.168.0.33  192.168.0.15  TCP      bgp > 2123 [FIN, PSH, ACK] Seq
    2 0.000031  192.168.0.15  192.168.0.33  TCP      2123 > bgp [ACK] Seq=352398319
    3 0.000422  192.168.0.15  192.168.0.33  TCP      2123 > bgp [FIN, ACK] Seq=3523
    4 0.006057  192.168.0.33  192.168.0.15  TCP      bgp > 2123 [ACK] Seq=196842842
    5 7.999977  192.168.0.15  192.168.0.33  TCP      2124 > bgp [SYN] Seq=359370684
    6 8.003909  192.168.0.33  192.168.0.15  TCP      bgp > 2124 [SYN, ACK] Seq=2051
    7 8.003954  192.168.0.15  192.168.0.33  TCP      2124 > bgp [ACK] Seq=359370685
    8 8.004042  192.168.0.15  192.168.0.33  BGP      OPEN Message
    9 8.208048  192.168.0.33  192.168.0.15  TCP      bgp > 2124 [ACK] Seq=205107207
   10 8.337997  192.168.0.33  192.168.0.15  BGP      OPEN Message
   11 8.338027  192.168.0.15  192.168.0.33  TCP      2124 > bgp [ACK] Seq=359370687
```

```
    Length: 29 bytes
    Type: OPEN Message (1)
    Version: 4
    My AS: 65033
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                           ...
```

BGP message type (bgp.type), 1 byte

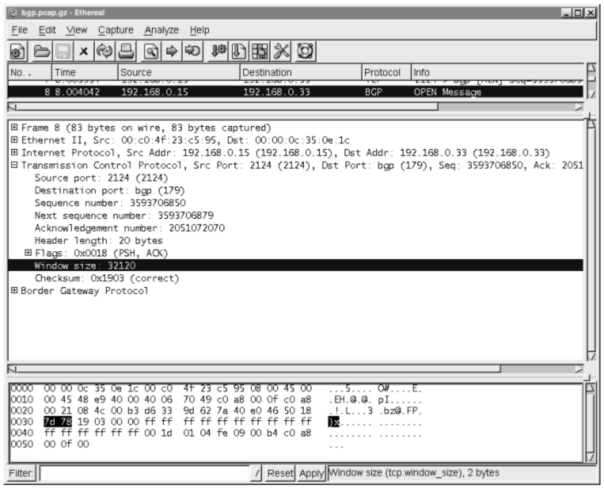
When a field in the Protocol Tree Window is selected, the bytes corresponding to that field are highlighted in the Data View Window. In our example we have selected the **BGP message type** field in the Protocol Tree Window. In the Data View Window we find that byte highlighted in the row with offset 0040 representing 0x40 hex or 64 bytes into the packet. The ninth byte in the row is highlighted, and has a value of 01 hex. In the ASCII representation we have a period (.) because the value 0x01 is not conveniently represented in ASCII.

When you click on a hex byte or ASCII character in the Data View Window**,** Ethereal will highlight the field in the Protocol Tree Window corresponding to the selected byte, and all bytes in the Data View Window associated with that **Protocol** field.

In Figure 4.5 we've clicked on the 7d at the beginning of row 0030. It just so happens that the 48th byte (0030 or hex 0x30) is the first byte of the two-byte Transmission Control Protocol (TCP) **Window Size** field. As a result, the TCP tree was automatically expanded and the **Window size** field was highlighted. Additionally the second byte, with value 78 hex, in the 0030 row, was also selected, since TCP Window Size is a two-byte field.

This feature of Ethereal makes it very easy to use the Protocol Tree Window and the Data View Window together to obtain a solid grasp of the relationships between the fields in a protocol and the actual bits on the wire.

Figure 4.5 Data View Window Byte Selection



No. .	Time	Source	Destination	Protocol	Info
8	8.004042	192.168.0.15	192.168.0.33	BGP	OPEN Message

```
⊞ Frame 8 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊟ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051
    Source port: 2124 (2124)
    Destination port: bgp (179)
    Sequence number: 3593706850
    Next sequence number: 3593706879
    Acknowledgement number: 2051072070
    Header length: 20 bytes
  ⊞ Flags: 0x0018 (PSH, ACK)
    Window size: 32120
    Checksum: 0x1903 (correct)
⊞ Border Gateway Protocol
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                           ...
```

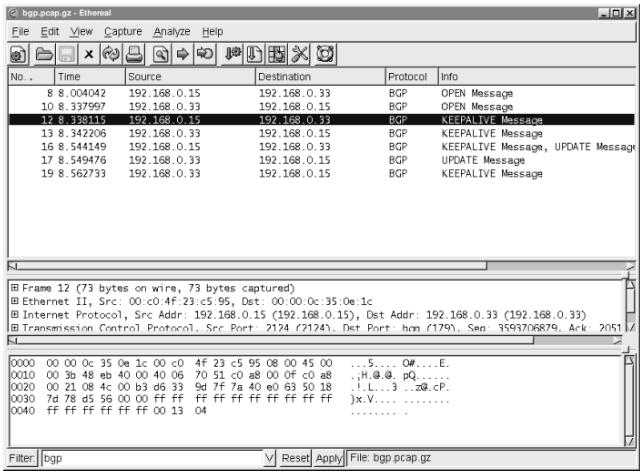
Window size (tcp.window_size), 2 bytes

Other Window Components

Filter Bar

The Filter Bar (Figure 4.6) allows you to enter a filter string restricting which packets are displayed in the Summary Window. Only packets that match the display filter string will be displayed in the Summary Window. A display filter string is a string defining some conditions on a packet that may or may not match a packet. For example, the display filter string *(ip.addr == 10.15.162.1 && bgp)* would match all packets with an IP address (source or destination) of 10.15.162.1 that are BGP protocol packets. Filtering will be explored in much greater depth in Chapter 5, but we will examine the basic functions of the Filter Bar briefly here.

Figure 4.6 Filter Bar



```
No. .  Time        Source          Destination     Protocol  Info
     8 8.004042    192.168.0.15    192.168.0.33    BGP       OPEN Message
    10 8.337997    192.168.0.33    192.168.0.15    BGP       OPEN Message
    12 8.338115    192.168.0.15    192.168.0.33    BGP       KEEPALIVE Message
    13 8.342206    192.168.0.33    192.168.0.15    BGP       KEEPALIVE Message
    16 8.544149    192.168.0.15    192.168.0.33    BGP       KEEPALIVE Message, UPDATE Messag
    17 8.549476    192.168.0.33    192.168.0.15    BGP       UPDATE Message
    19 8.562733    192.168.0.33    192.168.0.15    BGP       KEEPALIVE Message
```

```
⊞ Frame 12 (73 bytes on wire, 73 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706879, Ack: 2051
```

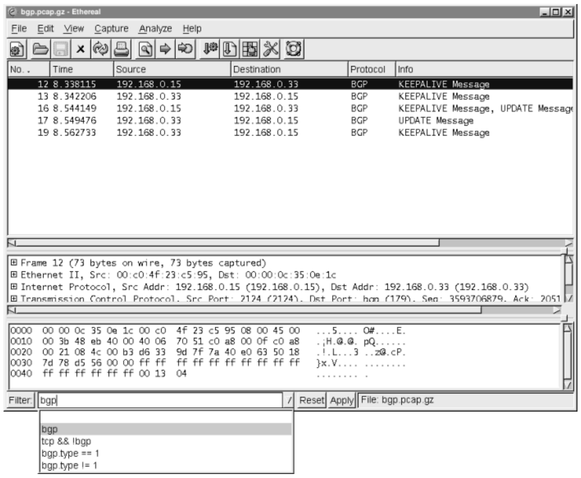
```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 3b 48 eb 40 00 40 06  70 51 c0 a8 00 0f c0 a8   .;H.@.@. pQ......
0020  00 21 08 4c 00 b3 d6 33  9d 7f 7a 40 e0 63 50 18   .!.L...3 ..z@.cP.
0030  7d 78 d5 56 00 00 ff ff  ff ff ff ff ff ff ff ff   }x.V.... ........
0040  ff ff ff ff ff ff 00 13  04                        ........ .
```

Filter: bgp

You can see in Figure 4.6 that we have applied a **bgp** filter. To apply a filter, enter the desired string into the **Filter:** text field and press **Enter** (or click the **Apply** button). Please note that your filter string will not be applied until you hit **Enter** or click the **Apply** button**.** In addition, note that filter strings are case-sensitive and the filter string **BGP** (uppercase) will not work. Traditionally, filter string labels like **bgp** are entirely in lowercase.

Once the display filter string **bgp** is applied, you will note that only BGP packets are displayed in the Summary Window. You should also note that the **No.** column displays jumps between the frame numbers of the displayed packets. This is because there are packets in the capture that are not being displayed, but are rather being suppressed by the **bgp** filter string. Previously used filters can be easily recalled, as shown in Figure 4.7.

Figure 4.7 Filter Bar Drop-down List



```
No. .  Time      Source         Destination    Protocol Info
    12 8.338115  192.168.0.15   192.168.0.33   BGP      KEEPALIVE Message
    13 8.342206  192.168.0.33   192.168.0.15   BGP      KEEPALIVE Message
    16 8.544149  192.168.0.15   192.168.0.33   BGP      KEEPALIVE Message, UPDATE Message
    17 8.549476  192.168.0.33   192.168.0.15   BGP      UPDATE Message
    19 8.562733  192.168.0.33   192.168.0.15   BGP      KEEPALIVE Message
```

```
⊞ Frame 12 (73 bytes on wire, 73 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706879, Ack: 2051
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 3b 48 eb 40 00 40 06  70 51 c0 a8 00 0f c0 a8   .;H.@.@. pQ......
0020  00 21 08 4c 00 b3 d6 33  9d 7f 7a 40 e0 63 50 18   .!.L...3 ..z@.cP.
0030  7d 78 d5 56 00 00 ff ff  ff ff ff ff ff ff ff ff   }x.V.... ........
0040  ff ff ff ff ff ff 00 13  04                        ........ .
```

Filter: bgp

File: bgp.pcap.gz

```
bgp
tcp && !bgp
bgp.type == 1
bgp.type != 1
```

By clicking the drop-down arrow at the right end of the **Filter** field, you can access a list of previously applied filters. To use one of these filters, select it from the list and press **Enter** or click the **Apply** button.

To remove the currently displayed filter string and re-display all packets, click the **Reset** button.

If you click the **Filter:** button itself, the Display Filter dialog box will be displayed. This feature is described in the section entitled "Analyze".

Information Field

The **Information** field displays the name of the capture file or information about the protocol field selected in the Protocol Tree Window.

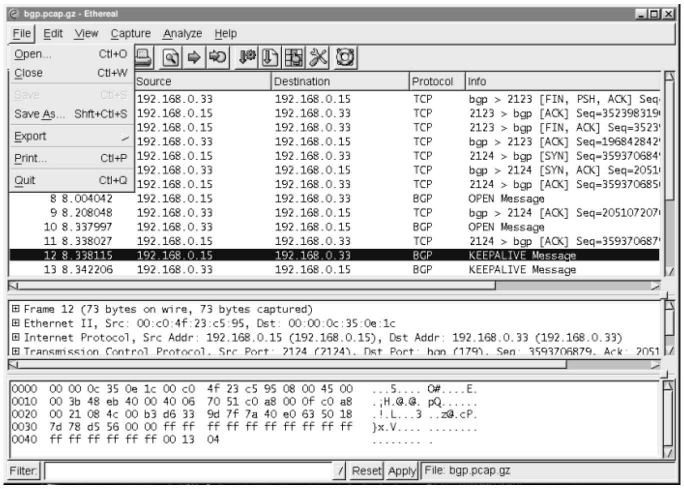
Exploring the Menus

All of the functionality available within Ethereal is accessible from the Menu Bar. In this section we will systematically explore that functionality and provide examples of its use.

File

The **File** menu provides access to loading, saving and printing capture files, as seen in Figure 4.8. **File** menu options are defined in Table 4.5.

Figure 4.8 File Menu



```
                Source              Destination          Protocol  Info
                192.168.0.33        192.168.0.15         TCP       bgp > 2123 [FIN, PSH, ACK] Seq
                192.168.0.15        192.168.0.33         TCP       2123 > bgp [ACK] Seq=352398319
                192.168.0.15        192.168.0.33         TCP       2123 > bgp [FIN, ACK] Seq=3523
                192.168.0.33        192.168.0.15         TCP       bgp > 2123 [ACK] Seq=196842842
                192.168.0.15        192.168.0.33         TCP       2124 > bgp [SYN] Seq=359370684
                192.168.0.33        192.168.0.15         TCP       bgp > 2124 [SYN, ACK] Seq=2051
                192.168.0.15        192.168.0.33         TCP       2124 > bgp [ACK] Seq=359370685
      8 8.004042 192.168.0.15       192.168.0.33         BGP       OPEN Message
      9 8.208048 192.168.0.33       192.168.0.15         TCP       bgp > 2124 [ACK] Seq=205107207
     10 8.337997 192.168.0.33       192.168.0.15         BGP       OPEN Message
     11 8.338027 192.168.0.15       192.168.0.33         TCP       2124 > bgp [ACK] Seq=359370687
     12 8.338115 192.168.0.15       192.168.0.33         BGP       KEEPALIVE Message
     13 8.342206 192.168.0.33       192.168.0.15         BGP       KEEPALIVE Message
```

```
⊞ Frame 12 (73 bytes on wire, 73 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706879, Ack: 2051
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 3b 48 eb 40 00 40 06  70 51 c0 a8 00 0f c0 a8   .;H.@.@. pQ......
0020  00 21 08 4c 00 b3 d6 33  9d 7f 7a 40 e0 63 50 18   .!.L...3 ..z@.cP.
0030  7d 78 d5 56 00 00 ff ff  ff ff ff ff ff ff ff ff   }x.V.... ........
0040  ff ff ff ff ff ff 00 13  04                        ........ .
```

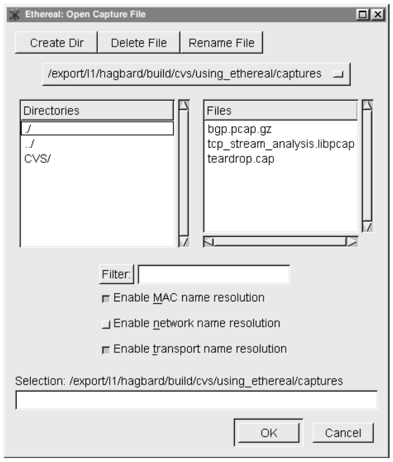
Table 4.5 File Menu Options

Menu Option	Description
Open…	Open a capture file.
Close	Close the current capture file.
Save	Save the current capture file.
Save As…	Save the current capture file with a different filename/format.
Export	Display the **Export** submenu allowing the portion of the packet highlighted in the Data View Window to be exported as a hexadecimal dump.
Print…	Print the current capture file.
Quit	Quit the Ethereal application.

Open

To open a file (Figure 4.9), select **File | Open**.

Figure 4.9 Open Dialog Box



/export/l1/hagbard/build/cvs/using_ethereal/captures

Directories: ./ ../ CVS/

Files: bgp.pcap.gz tcp_stream_analysis.libpcap teardrop.cap

☑ Enable MAC name resolution

□ Enable network name resolution

☑ Enable transport name resolution

Selection: /export/l1/hagbard/build/cvs/using_ethereal/captures

The Open dialog box provides normal mechanisms for navigation in selecting a file. Additionally, it provides a **Filter:** field where an Ethereal display filter string can be entered to filter which packets are read from the capture file. Clicking the **Filter:** button will open the Display Filter dialog box. The Display Filter dialog box is described in the section entitled "Analyze".

The Open dialog box also has checkboxes to enable name resolution for MAC addresses, network names, and transport names. To open a file, browse to the correct location and select the desired file, optionally provide a filter string, and enable or disable the name resolutions you wish to use. Finally, click the **OK** button.

Notes from the Underground…

Ethereal Name Resolution

Ethereal provides three kinds of name resolution to make some of the numbers found in network protocols more comprehensible. You can choose to enable or disable MAC name resolution, network name resolution, and transport name resolution, when opening a file, starting a capture, or while a capture is running. It is useful to understand what the different name resolutions mean.

Every host on a LAN is identified by a unique six-byte MAC address. These addresses are used in Ethernet frames to provide source and destination addresses at the Datalink Layer. MAC addresses are supposed to be globally unique. To achieve this end the IEEE (Institute of Electrical and Electronic Engineers), assigns blocks of MAC addresses to manufacturers. The first three bytes of every MAC address designate the manufacturer who produced the device. Ethereal is shipped with a list of the assigned prefixes and the manufacturers to whom they've been assigned. When you select the **Enable MAC name resolution** checkbox in the Open dialog box, Ethereal will resolve the first three bytes of the MAC address to a manufacturer and display that information for each MAC address. For example the prefix 00:00:0c has been assigned to Cisco Systems. When MAC address resolution is enabled Ethereal will display the MAC address *00:00:0c:35:0e 1c* as *00:00:0c:35:0e:1c (Cisco_35:0e:1c)*.

Every node on an IP network has an IP address. If you select the **Enable network name resolution** checkbox Ethereal will perform a reverse Domain Name System (DNS) lookup when it encounters an IP address to determine its associated domain name (like www.syngress.com). Ethereal will then display this domain name with the IP address. For example, the IP address *66.35.250.150* can be resolved via reverse DNS to the domain name slashdot.org. If network name resolution is enabled, Ethereal will display it as *slashdot.org (66.35.250.150)*.

Transport layer protocols like TCP and UDP (User Datagram Protocol, a connectionless transport protocol over IP defined in RFC 768 and viewable at www.ietf.org/rfc/rfc0768.txt?number=768) typically provide some form of multiplexing by allowing a source and destination

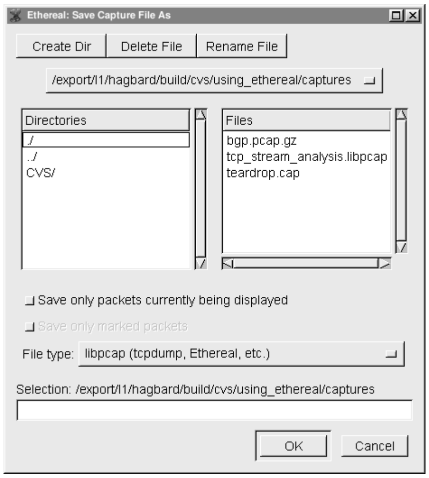
Continued

port to be specified. As a result, two hosts can have multiple clearly delineated conversations between them at the same time, as long as they have unique source port/destination port pairs for each conversation. Many protocols that use TCP or UDP for their transport layer have well-known ports that servers for those protocols traditionally listen on. When you select the **Enable transport name resolution** checkbox, Ethereal will display the name of the service that traditionally runs over each port. This behavior can be seen in many of our examples, where the port *179* has been labeled by the protocol that is well known to run over that port: *bgp*. It's important to note that most ports have no protocols associated with them.

Save As

The Save As dialog box, shown in Figure 4.10, is displayed by selecting **File | Save As**, or by selecting **File | Save** for a capture that has not previously been saved to file.

Figure 4.10 Save As Dialog Box



/export/l1/hagbard/build/cvs/using_ethereal/captures

Directories:
./
../
CVS/

Files:
bgp.pcap.gz
tcp_stream_analysis.libpcap
teardrop.cap

□ Save only packets currently being displayed

□ Save only marked packets

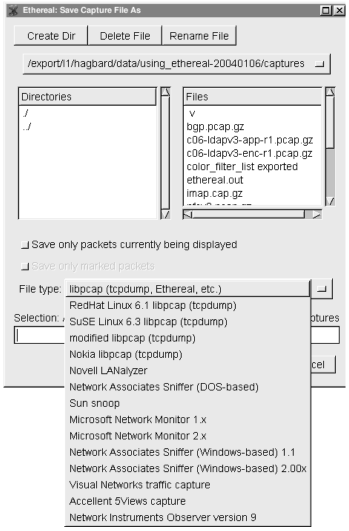
File type: libpcap (tcpdump, Ethereal, etc.)

Selection: /export/l1/hagbard/build/cvs/using_ethereal/captures

The Save As dialog box allows you to perform normal tasks for saving a capture file in the desired place and with the desired name. You can choose to save only the packets that pass the currently active display filter by enabling the **Save only packets currently being displayed** checkbox, or to save only marked packets by enabling the **Save only marked packets** checkbox. Marking packets will be explained later in the "Edit" section. Selecting both checkboxes will save only those marked packets that match the current display filter.

Finally, you can choose to save the file in one of a large number of supported capture file formats (Figure 4.11).

Figure 4.11 Save As Dialog: File Formats



Ethereal: Save Capture File As

/export/l1/hagbard/data/using_ethereal-20040106/captures

Directories:
./
../

Files:
v
bgp.pcap.gz
c06-ldapv3-app-r1.pcap.gz
c06-ldapv3-enc-r1.pcap.gz
color_filter_list exported
ethereal.out
imap.cap.gz

□ Save only packets currently being displayed

□ Save only marked packets

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

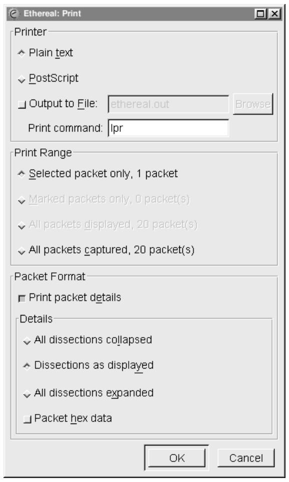
RedHat Linux 6.1 libpcap (tcpdump)
SuSE Linux 6.3 libpcap (tcpdump)
modified libpcap (tcpdump)
Nokia libpcap (tcpdump)
Novell LANalyzer
Network Associates Sniffer (DOS-based)
Sun snoop
Microsoft Network Monitor 1.x
Microsoft Network Monitor 2.x
Network Associates Sniffer (Windows-based) 1.1
Network Associates Sniffer (Windows-based) 2.00x
Visual Networks traffic capture
Accellent 5Views capture
Network Instruments Observer version 9

Selection:

Print

The Print dialog box is displayed by selecting **File** | **Print** (Figure 4.12).

Figure 4.12 Print Dialog Box



Ethereal: Print

Printer
● Plain text
○ PostScript
□ Output to File: ethereal.out
Print command: lpr

Print Range
● Selected packet only, 1 packet
○ Marked packets only, 0 packet(s)
○ All packets displayed, 20 packet(s)
○ All packets captured, 20 packet(s)

Packet Format
☑ Print packet details

Details
○ All dissections collapsed
● Dissections as displayed
○ All dissections expanded
□ Packet hex data

The Print dialog box allows you to provide answers to the three questions relevant to printing in Ethereal:

1. How are you going to print?
2. Which packets are you going to print?
3. What information are you going to print for each packet?

The **Printer** section allows you to choose which packets you are going to print. You can choose as your output format either **Plain Text** or **Postscript**. Once you have selected your output format you may choose to print the output to a file by enabling the **Output to File:** checkbox and providing a filename in the **Output to File:** textbox. If you do not choose to print to file then you may provide a command to be executed to print in the **Print command:** textbox.

The **Print Range** section allows you to choose which packets you are going to print. You may choose to print only the packet currently selected in the Summary Window, only packets that are marked in the Summary Window (marked packets are discussed later in the "Edit" section), all packets displayed in the Summary Window by the currently applied filter, or all packets captured. You can choose to print only the packet currently selected in the Summary Window by selecting the **Selected packet only** radio button. To print only the packets that have been marked in the Summary Window**,** select the **Marked packets only** radio button. And, to print all packets displayed in the Summary Window by the currently applied display filter, selecting the **All packets displayed** radio button. Printing all packets displayed means that all packets that pass the currently applied filter will print, not just the packets that are currently visible in the Summary Window. If you are able to scroll up or down to a packet in the Summary Window**,** it is considered to be "displayed" for the purposes of this print range option. You can print all packets in the capture by selecting the **All packets captured** radio button.

The **Packet Format** section allows you to choose which information you are going to print for each packet. If you do not enable the **Print packet details** checkbox, then for each packet a one-line summary consisting of the columns currently being displayed in the Summary Window will be printed. Consider, for example, the state of Ethereal in Figure 4.3. Packet 8 is selected. If the **Print packet details** checkbox is unselected, the result of printing only the selected packet (packet 8) would be:

```
No. Time       Source         Destination     Protocol Info
  8 8.004042  192.168.0.15   192.168.0.33    BGP      OPEN Message
```

This output had some whitespace removed to contract it to fit the space. If you do enable the **Print packet details** checkbox, then more detailed information will be printed.

The **Details** section allows you to choose which details are printed for a packet when you have enabled the **Print packet details** checkbox. You may choose to print the protocol tree with all subtrees collapsed, the protocol tree with subtrees expanded (but only if those subtrees are expanded in the Protocol Tree Window), or with all subtrees in the protocol tree expanded. If you select the **All dissections collapsed** option, the protocol tree will be printed with all subtrees collapsed. For the situation shown in Figure 4.3, printing only the selected packet, the output would look like:

```
Frame 8 (83 bytes on wire, 83 bytes captured)

Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c

Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr:
192.168.0.33 (192.168.0.33)

Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179),
Seq: 3593706850, Ack: 2051072070, Len: 29

Border Gateway Protocol
```

If you select the **Dissections as displayed** option, the protocol tree will be printed with those subtrees expanded that would be expanded in the Protocol Tree Window if that packet was selected in the Summary Window. Using this option to print only the selected packet from Figure 4.3 would produce output like:

```
Frame 8 (83 bytes on wire, 83 bytes captured)

Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c

Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr:
192.168.0.33 (192.168.0.33)

Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179),
Seq: 3593706850, Ack: 2051072070, Len: 29

Border Gateway Protocol

    OPEN Message

        Marker: 16 bytes

        Length: 29 bytes

        Type: OPEN Message (1)

        Version: 4

        My AS: 65033

        Hold time: 180
        BGP identifier: 192.168.0.15
        Optional parameters length: 0 bytes
```

If you select the **All dissections expanded** option, the protocol tree will be printed with all subtrees expanded. Printing just the selected packet in Figure 4.8 with this option would produce the output:

```
Frame 8 (83 bytes on wire, 83 bytes captured)
    Arrival Time: Mar 29, 2000 23:56:56.957322000
    Time delta from previous packet: 0.000088000 seconds
    Time since reference or first frame: 8.004042000 seconds
    Frame Number: 8
    Packet Length: 83 bytes
    Capture Length: 83 bytes
Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
    Destination: 00:00:0c:35:0e:1c (Cisco_35:0e:1c)
    Source: 00:c0:4f:23:c5:95 (DellComp_23:c5:95)
    Type: IP (0x0800)
Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr:
192.168.0.33 (192.168.0.33)
    Version: 4
    Header length: 20 bytes
    Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
        0000 00.. = Differentiated Services Codepoint: Default (0x00)
        .... ..0. = ECN-Capable Transport (ECT): 0
        .... ...0 = ECN-CE: 0
    Total Length: 69
    Identification: 0x48e9 (18665)
    Flags: 0x04
        .1.. = Don't fragment: Set
        ..0. = More fragments: Not set
    Fragment offset: 0
    Time to live: 64
    Protocol: TCP (0x06)
    Header checksum: 0x7049 (correct)
    Source: 192.168.0.15 (192.168.0.15)
    Destination: 192.168.0.33 (192.168.0.33)
```

```
Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179),
Seq: 3593706850, Ack: 2051072070, Len: 29
    Source port: 2124 (2124)
    Destination port: bgp (179)
    Sequence number: 3593706850
    Next sequence number: 3593706879
    Acknowledgement number: 2051072070
    Header length: 20 bytes
    Flags: 0x0018 (PSH, ACK)
        0... .... = Congestion Window Reduced (CWR): Not set
        .0.. .... = ECN-Echo: Not set
        ..0. .... = Urgent: Not set
        ...1 .... = Acknowledgment: Set
        .... 1... = Push: Set
        .... .0.. = Reset: Not set
        .... ..0. = Syn: Not set
        .... ...0 = Fin: Not set
    Window size: 32120
    Checksum: 0x1903 (correct)
Border Gateway Protocol
    OPEN Message
        Marker: 16 bytes
        Length: 29 bytes
        Type: OPEN Message (1)
        Version: 4
        My AS: 65033
        Hold time: 180
        BGP identifier: 192.168.0.15
        Optional parameters length: 0 bytes
```

Regardless of the option you choose for expanding protocol tree subtrees, if you enable the **Packet hex data** checkbox, following the protocol tree for each packet will be a hex dump of that packet. Printing only the packet selected in Figure 4.3, with the **All dissections collapsed** checkbox enabled and the **Packet hex data** checkbox enabled would produce this output:

```
Frame 8 (83 bytes on wire, 83 bytes captured)
Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
```

```
Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr:
192.168.0.33 (192.168.0.33)
Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179),
Seq: 3593706850, Ack: 2051072070, Len: 29
Border Gateway Protocol

0000  00 00 0c 35 0e 1c 00 c0 4f 23 c5 95 08 00 45 00   ...5....O#....E.
0010  00 45 48 e9 40 00 40 06 70 49 c0 a8 00 0f c0 a8   .EH.@.@.pI......
0020  00 21 08 4c 00 b3 d6 33 9d 62 7a 40 e0 46 50 18   .!.L...3.bz@.FP.
0030  7d 78 19 03 00 00 ff ff ff ff ff ff ff ff ff ff   }x..............
0040  ff ff ff ff ff ff 00 1d 01 04 fe 09 00 b4 c0 a8   ................
0050  00 0f 00                                            ...
```

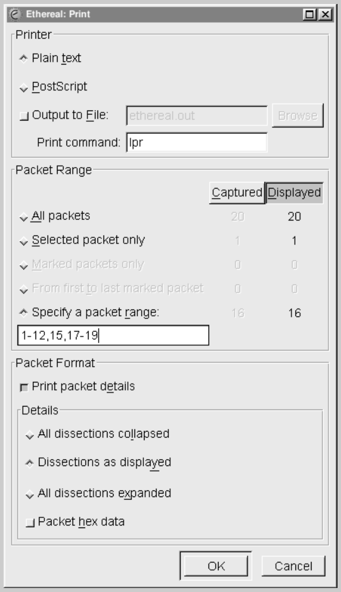
Notes from the Underground…

Ghost of Ethereal Future: Print Dialog Improvements

Ethereal evolves very rapidly. It is not uncommon for there to be a new release of the software every four to six weeks. As a result, some things will change over time. An example of this is the Print dialog box, which has already been changed in CVS (Concurrent Version System, the source code control mechanism used by Ethereal) since the release of the 0.10.0a version of Ethereal (the latest release version as of the writing of this book). The new Print dialog box, which will likely be in Ethereal version 0.10.1, is shown in Figure 4.13

Continued

Figure 4.13 The New Print Dialog Box



Ethereal: Print

Printer

● Plain text

○ PostScript

□ Output to File: ethereal.out Browse

Print command: lpr

Packet Range

	Captured	Displayed
○ All packets	20	20
○ Selected packet only	1	1
○ Marked packets only	0	0
○ From first to last marked packet	0	0
● Specify a packet range:	16	16

1-12,15,17-19

Packet Format

☑ Print packet details

Details

○ All dissections collapsed

● Dissections as displayed

○ All dissections expanded

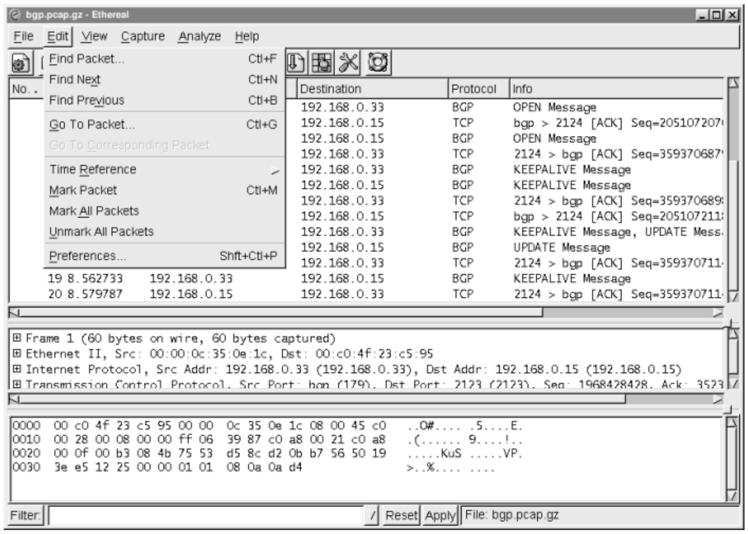
□ Packet hex data

The **Packet Range** section has been rewritten to provide more options when selecting the range of packets to be printed. The **Captured** and **Displayed** buttons are used to determine whether the selection criteria apply to all packets captured or all packets displayed. There are also two new packet selection criteria, **From first to last marked packet**, which selects all packets between the first marked packet and the last marked packet, and **Specify a packet range**, which will allow you to specify a packet range like the one shown in Figure 4.13: *1-12,15,17,17-19*.

Edit

The Edit menu, shown in Figure 4.14, allows you to find and mark packets, as well as set user preferences. Descriptions of the Edit menu options are given in Table 4.6.

Figure 4.14 Edit Menu



```
No. .                                 Destination       Protocol  Info
                                      192.168.0.33      BGP       OPEN Message
                                      192.168.0.15      TCP       bgp > 2124 [ACK] Seq=20510720
                                      192.168.0.15      BGP       OPEN Message
                                      192.168.0.33      TCP       2124 > bgp [ACK] Seq=35937068
                                      192.168.0.33      BGP       KEEPALIVE Message
                                      192.168.0.15      BGP       KEEPALIVE Message
                                      192.168.0.33      TCP       2124 > bgp [ACK] Seq=35937068
                                      192.168.0.15      TCP       bgp > 2124 [ACK] Seq=20510721
                                      192.168.0.33      BGP       KEEPALIVE Message, UPDATE Mess
                                      192.168.0.15      BGP       UPDATE Message
                                      192.168.0.33      TCP       2124 > bgp [ACK] Seq=35937071
19 8.562733     192.168.0.33          192.168.0.15      BGP       KEEPALIVE Message
20 8.579787     192.168.0.15          192.168.0.33      TCP       2124 > bgp [ACK] Seq=35937071
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 2123 (2123), Seq: 1968428428, Ack: 3523
```

```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8   .(...... 9....!..
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19   .....KuS .....VP.
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

Table 4.6 Edit Menu Options

Menu Option	Description
Find Packet…	Search for a packet using a display filter or by searching for a matching hexadecimal string or character string.
Find Next	Find the next packet that matches the search defined in the Find Packet dialog box.
Find Previous	Find the previous packet that matches the search defined in the Find Packet dialog box.
Go To Packet…	Go to a packet by frame number.
Go To Corresponding Packet	When a field that refers to another frame is selected in the Protocol Tree Window, select the packet being referred to in the Summary Window.
Time Reference	A submenu for manipulating time references (for more details, see the section entitled "Time Reference Submenu".
Mark Packet	Mark the packet currently selected in the **Summary Window**. If the selected packet is already marked, then unmark it. Marking provides a mechanism for manually selecting a packet or group of packets to be subsequently printed or saved.

Continued

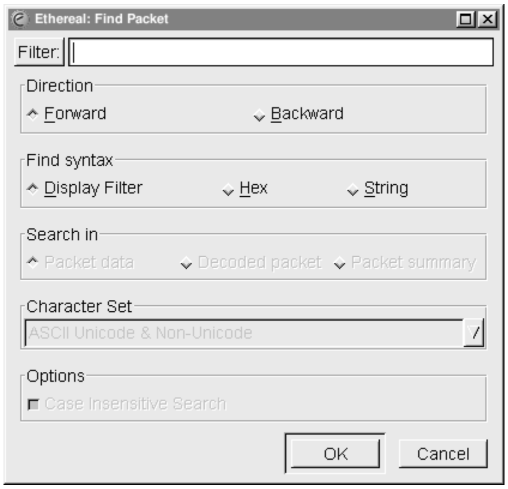
Table 4.6 Edit Menu Options

Menu Option	Description
Mark All Packets	Marks all packets that match the currently applied display filter.
Unmark All Packets	Unmark all packets that match the currently applied display filter.
Preferences…	Change user preferences, including preferences for packet decodes.

Find Packet

The Find Packet dialog box, show in Figure 4.15, is displayed when you select **Edit | Find Packet…**.

Figure 4.15 Find Packet Dialog Box



Ethereal: Find Packet

Filter:

Direction
● Forward ○ Backward

Find syntax
● Display Filter ○ Hex ○ String

Search in
● Packet data ○ Decoded packet ○ Packet summary

Character Set
ASCII Unicode & Non-Unicode

Options
☑ Case Insensitive Search

The Find Packet dialog box allows you to answer the three questions relevant to finding a packet in Ethereal:

1. What are we trying to find?
2. Which direction should we search in?
3. What type of thing are we trying to find?

The **Filter:** textbox allows you to define a search criteria by entering a string such as a display filter string, hex string, or ASCII string. If you need assistance constructing a filter string you can click the **Filter:** button to display the Display Filter dialog box. The Display Filter dialog box is described in more detail in the in the section entitled "Analyze".

The **Direction** section allows you to choose which direction you wish to search in–**Forward** from the packet currently selected in the Summary Window, or **Backward** from the packet currently selected in the Summary Window.

The **Find Syntax** section allows you define your search criteria. You may choose to search for packets that match a display filter string, packets that contain a hex string, or packets that contain a character string. If you select the **Display Filter** option, the string in the **Filter:** textbox will be interpreted as a display filter string and you will search for matches to that display filter string. If you select the **Hex** option, the string in the **Filter:** textbox will be interpreted as a hex string and you will search for packets that contain that hex string.

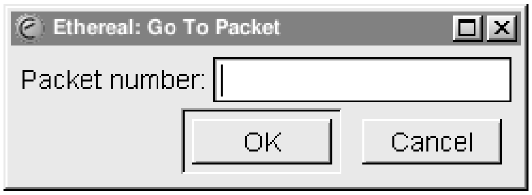
If you select the **String** option, the string in the **Filter:** will be interpreted as a character string and you will search for packets that contain that character string.

The search for character strings is handled differently than the search for hex strings. Hexadecimal string searches attempt to search for a packet containing a particular sequence of bytes anywhere in the raw data of that packet. The search for character strings will not look for a packet that contains a string anywhere in the packet. Instead, you can use the **Search In** section to specify whether to look for the string in the **Packet data** left over after decoding all possible fields, look for the character string in the **Decoded packet** displayed in the Protocol Tree Window, or look for the character string in the one-line **Packet summary** in the Summary Window. If you select the **Packet data** option, Ethereal will search for the character string in the packet data. By packet data, we mean the data in the packet that is left over after decoding the protocol fields. Selecting the **Find Decoded packet** will cause Ethereal to search for the character string in the protocol field strings that are displayed in the Protocol Tree Window. It does not matter if the subtree of the protocol tree containing the character string is collapsed or expanded. If you use the **Decoded packet** option, you must also use the **Character Set** drop-down list to select the character set for the character string you are trying to find. To make your character string search case-insensitive, enable the **Case Insensitive Search** checkbox.

Go To Packet

The Go To Packet dialog box, shown in Figure 4.16, can be displayed by selecting **Edit | Go To Packet Dialog**.

Figure 4.16 Go To Packet Dialog Box

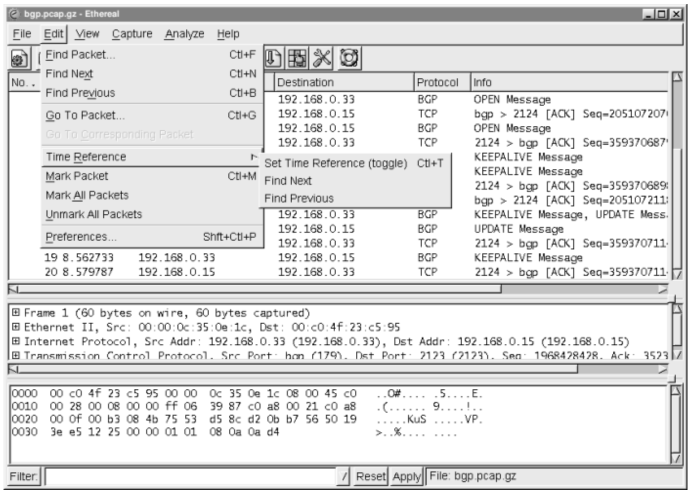


Enter a packet number in the **Packet Number** textbox and click **OK**. The packet with that packet number will be selected in the Summary Window.

Time Reference Submenu

The **Time Reference** submenu, shown in Figure 4.17, is displayed when you select **Edit | Time Reference**. The **Time Reference** submenu options are described in Table 4.7.

Figure 4.17 Time Reference Submenu



```
                         192.168.0.33     BGP       OPEN Message
                         192.168.0.15     TCP       bgp > 2124 [ACK] Seq=205107207
                         192.168.0.15     BGP       OPEN Message
                         192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370687
                                                    KEEPALIVE Message
                                                    KEEPALIVE Message
                                                    2124 > bgp [ACK] Seq=359370689
                                                    bgp > 2124 [ACK] Seq=205107211
                         192.168.0.33     BGP       KEEPALIVE Message, UPDATE Mess
                         192.168.0.15     BGP       UPDATE Message
                         192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370711
19 8.562733  192.168.0.33  192.168.0.15     BGP       KEEPALIVE Message
20 8.579787  192.168.0.15  192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370711
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 2123 (2123), Seq: 1968428428, Ack: 3523
```

```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8   .(...... 9....!..
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19   .....KuS .....VP.
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

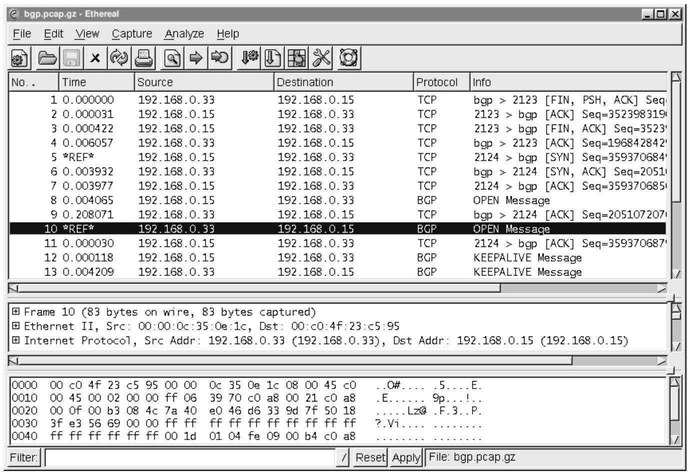
Table 4.7 Time Reference Submenu Options

Menu Option	Description
Set Time Reference (toggle)	Sets the packet currently selected in the Summary Window as a time reference packet.
Find Next	Finds the next time reference packet after the packet currently selected in the Summary Window.
Find Previous	Find the previous time reference packet before the packet currently selected in the Summary Window.

When the **Time** column in the Summary Window is configured to display the time that has elapsed since the beginning of the capture, then the time displayed is the number of seconds since the beginning of the capture or the last time reference packet.

In Figure 4.18 , we have set packets 5 and 10 as time reference packets. This is indicated by their **Time** column value (*REF*). Packets 1-4 are marked with the time since the beginning of the capture in which they were captured. Packets 6-9 are marked with the time since the time referencepacket 5. Packets 11 and greater are marked with the time since the time reference packet 10.

Figure 4.18 Time Reference Submenu Example



No.	Time	Source	Destination	Protocol	Info
1	0.000000	192.168.0.33	192.168.0.15	TCP	bgp > 2123 [FIN, PSH, ACK] Seq
2	0.000031	192.168.0.15	192.168.0.33	TCP	2123 > bgp [ACK] Seq=35239831
3	0.000422	192.168.0.15	192.168.0.33	TCP	2123 > bgp [FIN, ACK] Seq=3523
4	0.006057	192.168.0.33	192.168.0.15	TCP	bgp > 2123 [ACK] Seq=19684284
5	*REF*	192.168.0.15	192.168.0.33	TCP	2124 > bgp [SYN] Seq=35937068
6	0.003932	192.168.0.33	192.168.0.15	TCP	bgp > 2124 [SYN, ACK] Seq=2051
7	0.003977	192.168.0.15	192.168.0.33	TCP	2124 > bgp [ACK] Seq=35937068
8	0.004065	192.168.0.15	192.168.0.33	BGP	OPEN Message
9	0.208071	192.168.0.33	192.168.0.15	TCP	bgp > 2124 [ACK] Seq=20510720
10	*REF*	192.168.0.33	192.168.0.15	BGP	OPEN Message
11	0.000030	192.168.0.15	192.168.0.33	TCP	2124 > bgp [ACK] Seq=35937068
12	0.000118	192.168.0.15	192.168.0.33	BGP	KEEPALIVE Message
13	0.004209	192.168.0.33	192.168.0.15	BGP	KEEPALIVE Message

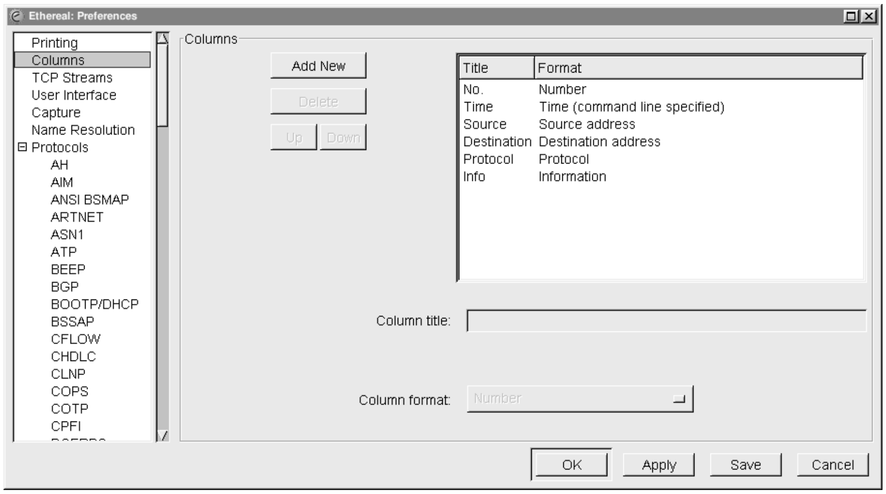
```
⊞ Frame 10 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
```

```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 45 00 02 00 00 ff 06  39 70 c0 a8 00 21 c0 a8   .E...... 9p...!..
0020  00 0f 00 b3 08 4c 7a 40  e0 46 d6 33 9d 7f 50 18   .....Lz@ .F.3..P.
0030  3f e3 56 69 00 00 ff ff  ff ff ff ff ff ff ff ff   ?.Vi.... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
```

Preferences

The Preferences dialog box, shown in Figure 4.19, is displayed when you select **Edit | Preferences…**.

Figure 4.19 Preferences Dialog Box



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

Column title:

Column format: Number

The Preferences dialog box allows you to set preferences for various subsystems of Ethereal, including setting preferences for decodes of various protocols. To edit preferences for an area of Ethereal, like **Columns** in Figure 4.19, select that area from the box on the left and change the settings displayed in the box on the right. It is strongly recommended that you browse through the protocol preferences for any protocol you use frequently, as protocol preferences can change the way a protocol is decoded or displayed.

When you have made your changes to Ethereal's preferences you can choose to apply them without closing the Preference dialog box by clicking the **Apply** button. To apply your settings and close the Preferences dialog box, click the **OK** button. To save your preferences for use in a different Ethereal session, click the **Save** button.

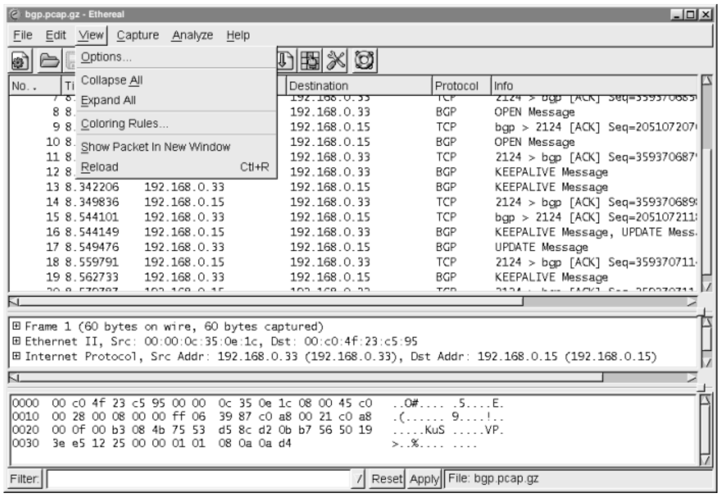
NOTE

The **Columns** preference, selected in Figure 4.19, is subtly broken in Ethereal. You can add, delete, or reorder columns in the Preferences dialog box, but your changes will not take effect unless you save them, then exit and restart Ethereal.

View

The **View** menu, shown in Figure 4.20, allows you to control how packets are displayed in the Summary Window, and the Protocol Tree Window.You can also set up color filters to color the packets in the Summary Window.The **View** menu options are described in Table 4.8.

Figure 4.20 View Menu



```
No. .  Ti                                  Destination      Protocol  Info
   8 8.                                    192.168.0.33     BGP       OPEN Message
   9 8.                                    192.168.0.15     TCP       bgp > 2124 [ACK] Seq=205107207
  10 8.                                    192.168.0.15     BGP       OPEN Message
  11 8.                                    192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370687
  12 8.                                    192.168.0.33     BGP       KEEPALIVE Message
  13 8.342206   192.168.0.33               192.168.0.15     BGP       KEEPALIVE Message
  14 8.349836   192.168.0.15               192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370689
  15 8.544101   192.168.0.33               192.168.0.15     TCP       bgp > 2124 [ACK] Seq=205107211
  16 8.544149   192.168.0.15               192.168.0.33     BGP       KEEPALIVE Message, UPDATE Mess.
  17 8.549476   192.168.0.33               192.168.0.15     BGP       UPDATE Message
  18 8.559791   192.168.0.15               192.168.0.33     TCP       2124 > bgp [ACK] Seq=359370711
  19 8.562733   192.168.0.33               192.168.0.15     BGP       KEEPALIVE Message
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
```

```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8   .(...... 9....!..
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19   .....KuS .....VP.
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

Table 4.8 View Menu Options

View Submenu Options	
Menu Option	**Description**
Options…	Set the View options.
Collapse All	Collapse all subtrees in the Protocol Tree Window
Expand All	Expand all subtrees in the Protocol Tree Window

Continued

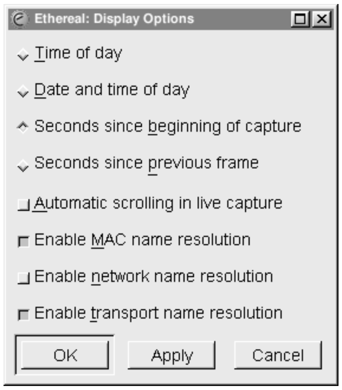
Table 4.8 View Menu Options

View Submenu Options	
Menu Option	**Description**
Coloring Rules…	Create and edit color filters to colorize the packets in the Summary Window that match a given display filter string.
Show Packet In New Window	For the packet currently selected in the Summary Window display it's Protocol Tree Window and Data View Window in a new window.
Reload	Reload the current capture file.

Display Options

The Display Options dialog box, shown in Figure 4.21, is displayed when you select **View | Options…**.

Figure 4.21 Display Options Dialog Box



Ethereal: Display Options

○ Time of day

○ Date and time of day

● Seconds since beginning of capture

○ Seconds since previous frame

☐ Automatic scrolling in live capture

☑ Enable MAC name resolution

☐ Enable network name resolution

☑ Enable transport name resolution

The Display Options dialog box allows you to choose which time value will be displayed in the **Time** column of the Summary Window, whether automatic scrolling will be enabled for live captures, and what type of name resolution will be enabled.

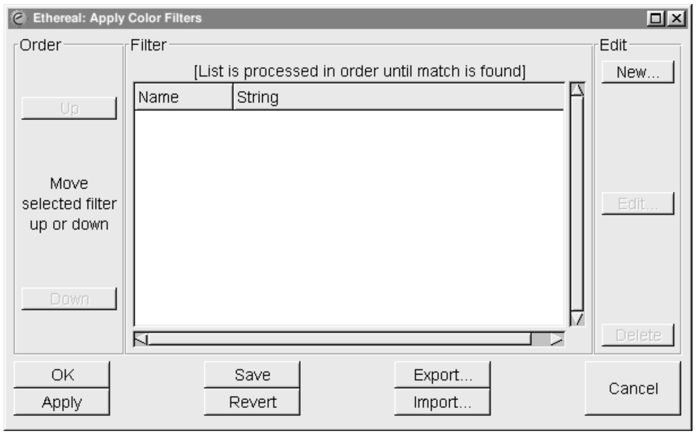
For a given packet, you may choose to have the **Time** column in the Summary Window display the **Time of day** when that packet was captured, **Date and time of day** when that packet was captured, **Seconds since beginning of capture** (or the last time reference packet) that packet was captured, or the **Seconds since the previous frame** that matched the current display filter.

In a live capture, you can choose to have old packets scroll up and out of view as new packets are captured and appended to the end of the Summary Window. To do so, enable the **Automatic scrolling in live capture** checkbox. You may choose your name resolution options in the Display Options dialog box as well. Refer back to the section entitled "Open" for more information about name resolution choices.

Apply Color Filters

The Apply Color Filters dialog box, shown in Figure 4.22, can be displayed by selecting **View | Coloring Rules…**.

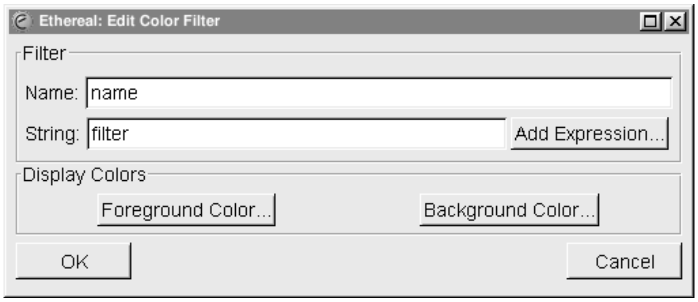
Figure 4.22 Apply Color Filters Dialog Box


Ethereal: Apply Color Filters
Order
Filter
Edit
[List is processed in order until match is found]
Name
String
New...
Up
Move selected filter up or down
Down
Edit...
Delete
OK
Apply
Save
Revert
Export...
Import...
Cancel

Ethereal has the ability to color packets in the Summary Window that match a given display filter string, making patterns in the capture data more visible. This can be immensely useful when trying to follow request response protocols where variations in the order of requests or responses may be interesting. You can color such traffic into as many categories as you'd like and will be able to see at a glance what is going on from the Summary Window instead of having to go through the Protocol Tree Window for each packet.

To create a color filter click the **New** button in the Apply Color Filters dialog box. The Edit Color Filter dialog box will be displayed (Figure 4.23).

Figure 4.23 Edit Color Filter Dialog Box



Ethereal: Edit Color Filter

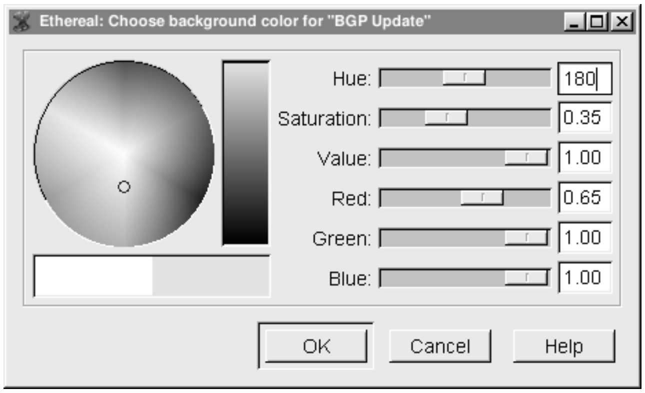
Filter

Name: name

String: filter

When the Edit Color Filter dialog box is first opened, the **Name** textbox will have the string *name* in it, and the **String** field will contain the string *filter*. To create a color filter you should first fill in a name for it in the **Name** textbox. Then, you should enter a filter string in the **String** textbox. You may use the **Add Expression** button to display the Filter Expression dialog box to assist you in constructing a filter string. The Filter Expression dialog box is described in the section entitled "Analyze". Once you have a name and filter string you are happy with, you need to select the foreground and background color to colorize the packets matching your filter string. Click the **Background Color...** button to set the foreground color, as shown in Figure 4.24.

Figure 4.24 Background Color Dialog Box



Ethereal: Choose background color for "BGP Update"

Hue: 180

Saturation: 0.35

Value: 1.00

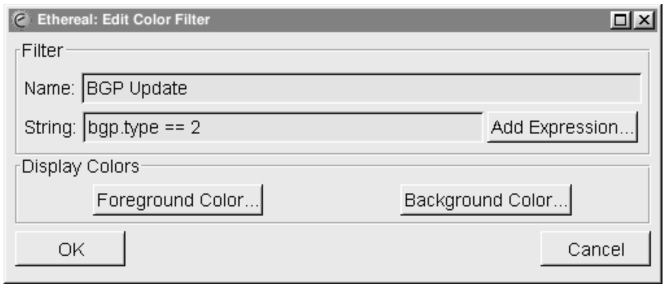
Red: 0.65

Green: 1.00

Blue: 1.00

When you are happy with the color you have selected click the **OK** button. The Edit Color Filter dialog box (Figure 4.25) will be displayed.

Figure 4.25 Edit Color Filter



Name: BGP Update

String: bgp.type == 2

In Figure 4.25 we have created a filter named *BGP Update* with a filter string ***bgp.type == 2***. The name and filter string will be colored to match our background color choice. Click the **Foreground Color…** button to set the foreground color and proceed as you did with the background color. When you are happy with your name, filter string, and text coloring click the **OK** to close the Edit Color Filter dialog box.

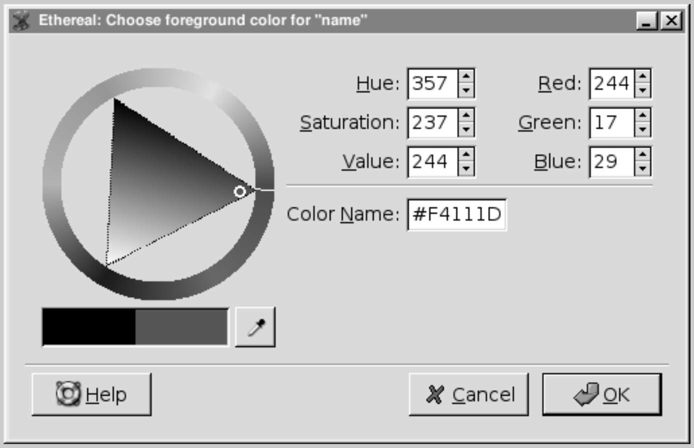
Notes from the Underground…

Ghost of Ethereal Future: GTK+2

Ethereal uses a widget set to provide its GUI elements (buttons, menus, radio buttons, file dialogs, color selection dialogs, etc.) called GTK+ (Gimp Tool Kit, www.gtk.org). By default, Ethereal builds with GTK+1, and so all of the screenshots you've seen in this book use GTK+1. There is already code in place in Ethereal for GTK+2, the next version of the GTK+ library. Many elements look very similar in GTK+2, just slightly cleaner, but a few things look very different. One of them is the color selection widget used in the Foreground Color and Background Color dialog boxes. You can see the new GTK+2 Foreground Color dialog box for Ethereal in Figure 4.26:

Continued

Figure 4.26 GTK+2 Foreground Color Dialog Box

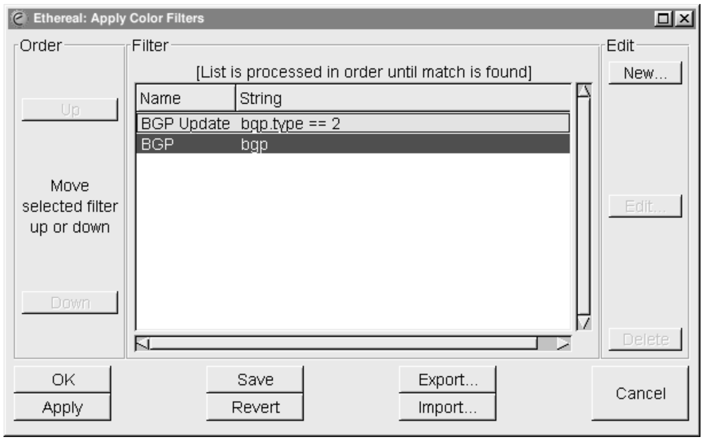


Ethereal: Choose foreground color for "name"

Hue: 357
Saturation: 237
Value: 244
Red: 244
Green: 17
Blue: 29
Color Name: #F4111D

Figure 4.27 shows the Apply Color Filters dialog box now populated with the new *BGP Update* entry and a *BGP* filter.

Figure 4.27 Apply Color Filters Dialog Box



Ethereal: Apply Color Filters

[List is processed in order until match is found]

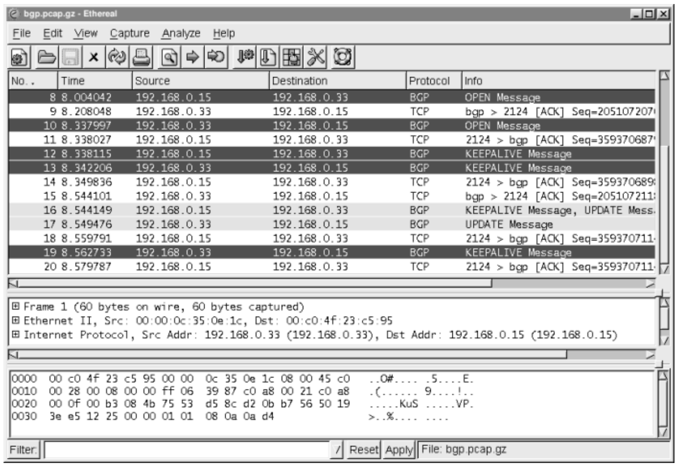
Name	String
BGP Update	bgp.type == 2
BGP	bgp

Move selected filter up or down

Click the **OK** button to apply the changes and close the dialog box. Click **Apply** to apply the changes and leave the dialog box open. If you wish to use your color filters with another Ethereal session, click **Save**.

If you click the **Revert** button, all coloring will be removed from the Summary Window, the color filters will be removed from the **Filter** list, and the saved color file will be deleted. Use the **Export** or **Import** buttons to export your color filters to another file or import the color filters from a file of your choice. This is very useful for sharing color filters with coworkers or between different machines on which you have Ethereal installed.Notice the order of the color filters in the **Filter** list in Figure 4.27. For every packet in the Summary View the color filters strings will be tried in order until one is matched. At that point, its associated color will be applied. The filters in the **Filter** list are applied from the top down, so the *BGP Update* color filter will be tried first. Only if the *BGP Update* color filter does not match a packet will Ethereal proceed to try the *BGP* color filter to that packet. An example of the application of these color filters can be seen in Figure 4.28.

Figure 4.28 Application of Color Filters



```
No. .  Time       Source         Destination    Protocol  Info
    8 8.004042    192.168.0.15   192.168.0.33   BGP       OPEN Message
    9 8.208048    192.168.0.33   192.168.0.15   TCP       bgp > 2124 [ACK] Seq=205107207
   10 8.337997    192.168.0.33   192.168.0.15   BGP       OPEN Message
   11 8.338027    192.168.0.15   192.168.0.33   TCP       2124 > bgp [ACK] Seq=359370687
   12 8.338115    192.168.0.15   192.168.0.33   BGP       KEEPALIVE Message
   13 8.342206    192.168.0.33   192.168.0.15   BGP       KEEPALIVE Message
   14 8.349836    192.168.0.15   192.168.0.33   TCP       2124 > bgp [ACK] Seq=359370689
   15 8.544101    192.168.0.33   192.168.0.15   TCP       bgp > 2124 [ACK] Seq=205107211
   16 8.544149    192.168.0.15   192.168.0.33   BGP       KEEPALIVE Message, UPDATE Mess.
   17 8.549476    192.168.0.33   192.168.0.15   BGP       UPDATE Message
   18 8.559791    192.168.0.15   192.168.0.33   TCP       2124 > bgp [ACK] Seq=359370711
   19 8.562733    192.168.0.33   192.168.0.15   BGP       KEEPALIVE Message
   20 8.579787    192.168.0.15   192.168.0.33   TCP       2124 > bgp [ACK] Seq=359370711
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
```

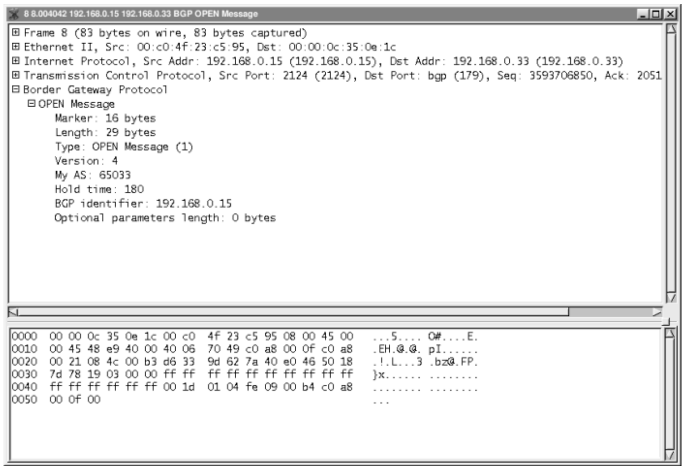
```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8   .(...... 9....!..
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19   .....KuS .....VP.
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

In Figure 4.28, the *BGP Update* messages (lines 16 and 17) are black text on light blue, not white text on dark blue, even though they would also match the white text on dark blue *BGP* color filter. This is because the black text on light blue *BGP Update* filter is applied first, and since it matches, no further color filter is tried.

Show Packet in New Window

You can display a packet's Protocol Tree Window and Data View Window in a new window by selecting a packet in the Summary Window and selecting **View | Show Packet in New Window** (see Figure 4.29). This is useful when you would like to be able to see detailed information about more than one packet at once. Note that the title bar shows the same information as the summary line for this packet in the Summary Window.

Figure 4.29 Show Packet in New Window



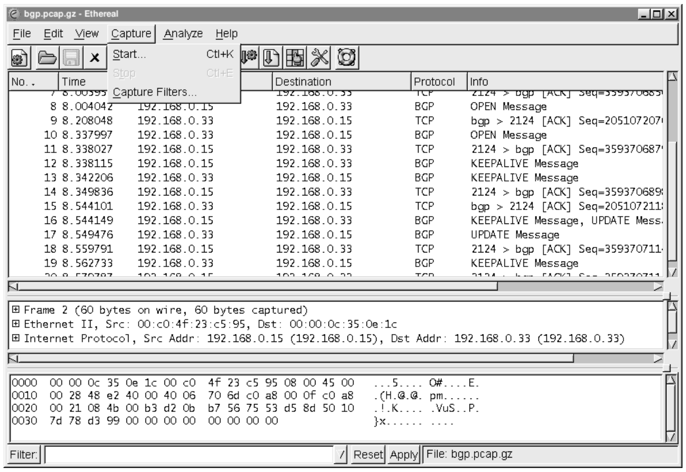
```
⊞ Frame 8 (83 bytes on wire, 83 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051
⊟ Border Gateway Protocol
  ⊟ OPEN Message
       Marker: 16 bytes
       Length: 29 bytes
       Type: OPEN Message (1)
       Version: 4
       My AS: 65033
       Hold time: 180
       BGP identifier: 192.168.0.15
       Optional parameters length: 0 bytes
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                           ...
```

Capture

The **Capture** menu is shown in Figure 4.30, and the menu entries are explained in Table 4.9.

Figure 4.30 Capture Menu



```
No. .  Time                                  Destination        Protocol  Info
     8 8.004042   192.168.0.15     192.168.0.33       BGP       OPEN Message
     9 8.208048   192.168.0.33     192.168.0.15       TCP       bgp > 2124 [ACK] Seq=205107207
    10 8.337997   192.168.0.33     192.168.0.15       BGP       OPEN Message
    11 8.338027   192.168.0.15     192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370687
    12 8.338115   192.168.0.15     192.168.0.33       BGP       KEEPALIVE Message
    13 8.342206   192.168.0.33     192.168.0.15       BGP       KEEPALIVE Message
    14 8.349836   192.168.0.15     192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370689
    15 8.544101   192.168.0.33     192.168.0.15       TCP       bgp > 2124 [ACK] Seq=205107211
    16 8.544149   192.168.0.15     192.168.0.33       BGP       KEEPALIVE Message, UPDATE Mess.
    17 8.549476   192.168.0.33     192.168.0.15       BGP       UPDATE Message
    18 8.559791   192.168.0.15     192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370711
    19 8.562733   192.168.0.33     192.168.0.15       BGP       KEEPALIVE Message
```

```
⊞ Frame 2 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 28 48 e2 40 00 40 06  70 6d c0 a8 00 0f c0 a8   .(H.@.@. pm......
0020  00 21 08 4b 00 b3 d2 0b  b7 56 75 53 d5 8d 50 10   .!.K.... .VuS..P.
0030  7d 78 d3 99 00 00 00 00  00 00 00 00               }x...... ....
```

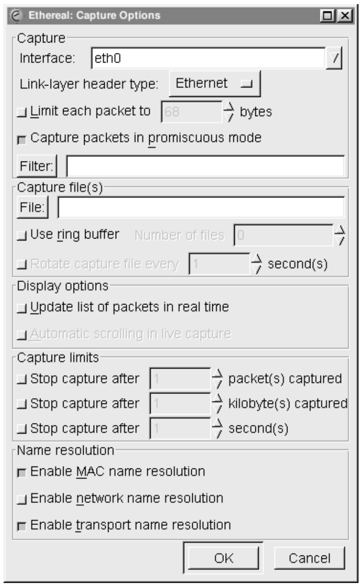
Table 4.9 Capture Menu Options

Menu Option	Description
Start…	Start a capture.
Stop	Stop a running packet capture.
Capture Filters…	Edit the capture filters.

Capture Options

The Capture Options dialog box, shown in Figure 4.31, can be displayed by selecting **Capture | Start…**.

Figure 4.31 Capture Options Dialog



Ethereal: Capture Options

Capture
Interface: eth0
Link-layer header type: Ethernet
☐ Limit each packet to 68 bytes
☑ Capture packets in promiscuous mode
Filter:

Capture file(s)
File:
☐ Use ring buffer Number of files 0
☐ Rotate capture file every 1 second(s)

Display options
☐ Update list of packets in real time
☐ Automatic scrolling in live capture

Capture limits
☐ Stop capture after 1 packet(s) captured
☐ Stop capture after 1 kilobyte(s) captured
☐ Stop capture after 1 second(s)

Name resolution
☑ Enable MAC name resolution
☐ Enable network name resolution
☑ Enable transport name resolution

This dialog box allows us to answer the basic questions about capturing data:

1. What traffic are we capturing?
2. Where are we saving it?
3. How are we displaying it?
4. When do we stop capturing?

The **Capture** section allows us to choose which traffic we are capturing. When choosing what traffic to capture we can ask:

1. Which interface are we capturing from?
2. How much of each packet are we capturing?
3. Which packets arriving at the interface are we capturing?

The **Interface** drop-down list allows us to choose which interface we are going to be capturing from. You can choose from the interfaces listed in the drop–down list, or if the interface you are seeking isn't listed there, you can enter it manually in the textbox. If both libpcap and the interface you select support multiple link layers for that interface, you can choose which link layer header type to capture using the **Link-layer header type:** selector.

The **Limit each packet to** field allows you to choose to capture less than the entire packet. If you enable the **Limit each packet to** checkbox and provide a number in the **Limit each packet to** textbox, then only the first number of bytes you indicate will be captured from each packet. Be aware that if you choose to capture less than the full packet Ethereal may mark many of your packets as fragments. This is because all of the data expected by the dissectors in Ethereal may not be present due to the packets having been truncated.

The **Capture packets in promiscuous mode** checkbox and the **Filter:** textbox allow you to choose which packets arriving at the interface will be captured. If you enable the **Capture packets in promiscuous mode** checkbox, Ethereal will put the interface into promiscuous mode before capturing data. Normally, an interface only passes onto the operating system packets that are addressed to the link layer address assigned to that interface. When an interface is in promiscuous mode it passes on all packets arriving at the interface to the operating system. So, if you choose not to capture in promiscuous mode, you will only capture packets addressed to or being sent by the interface on which you are capturing. If you choose to capture in promiscuous mode you will capture all packets arriving at the interface. Entering a tcpdump-style capture filter in the **Filter** textbox will cause Ethereal to only capture packets matching that capture filter. If you click on the **Filter** button then the Edit Capture Filter List dialog box will be displayed to allow you to choose among previously defined capture filters. See the section entitled "Edit Capture Filter List" for more details.

The **Capture File(s)** section allows you to choose where to save the capture. If you choose to leave this section blank, Ethereal will save the capture to a temporary file, and you can choose to save the capture at some later point by selecting

File | SaveAs. If you enter a filename in the **File** textbox, Ethereal will save the capture to that file. Clicking the **File** button will open the Save As dialog box. If you enable the **Use ring buffer** checkbox, you can save your capture to a ring buffer. Saving to a ring buffer will be dealt with in a separate section.

The **Display options** section allows you to choose how you are going to display packets as they are captured. By default, Ethereal does not update the list of packets in the Summary Window during capture, but only once the capture is stopped. If you enable the **Update list of packets in real time** checkbox, Ethereal will update the Summary Window as soon as a packet is captured and processed. By default, when Ethereal is updating the Summary Window during live capture, new packets are appended to the end of the Summary Window, and the Summary Window does not scroll up old packets to reveal new ones. To have the Summary Window scroll up to display the most recent packets, enable the **Automatic scrolling in live capture** checkbox. If you change your mind about whether you want automatic scrolling once a capture has started, you can select **View | Options** to enable or disable this feature.

The **Capture limits** section allows you to choose when to stop capturing. You can, of course, always manually stop a capture by selecting **Capture | Stop**, but it is sometimes convenient to set conditions under which the capture will automatically stop. There are three types of automatic limits to a capture supported by Ethereal:

1. Capture a specified number of packets.
2. Capture a specified number of kilobytes of traffic.
3. Capture for a specified number of seconds.

Ethereal will allow you to set up any combination of these three limits simultaneously–it is possible to limit the number of packets, the number of kilobytes, and the number of seconds at the same time. Whenever one of the limits is satisfied, the capture will stop.

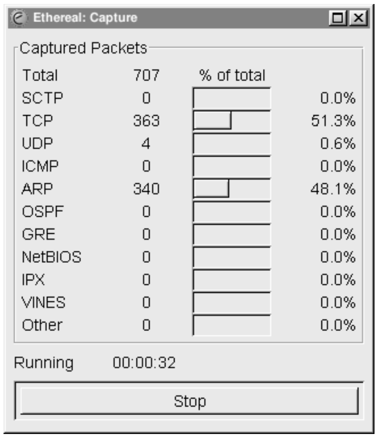
If you enable the **Stop capture after… packet(s) captured** checkbox and enter a number of packets in the **Stop capture after… packet(s) captured** textbox, the capture will stop when it has reached the specified number of packets. If you enable the **Stop capture after… kilobyte(s) captured** checkbox and enter a number of kilobytes in the **Stop capture after… kilobytes(s) captured** textbox, the capture will stop once it has reached the specified number of kilobytes. If you enable the **Stop capture after… seconds(s)** checkbox and enter a number of packets in the **Stop capture after… sec-**

onds(s) textbox, the capture will stop when the specified number of seconds have elapsed since the beginning of the capture.

The **Name resolution** section allows you to choose the name resolution options for the capture. Name resolution options are described earlier in the chapter in the section entitled "Open".

When you have specified your capture choices via the Capture Options dialog box, you can start the capture by clicking the **OK** button. The **Capture Dialog** will then be displayed, as shown in Figure 4.32.

Figure 4.32 Capture Dialog Box



Ethereal: Capture

Captured Packets

Total	707	% of total
SCTP	0	0.0%
TCP	363	51.3%
UDP	4	0.6%
ICMP	0	0.0%
ARP	340	48.1%
OSPF	0	0.0%
GRE	0	0.0%
NetBIOS	0	0.0%
IPX	0	0.0%
VINES	0	0.0%
Other	0	0.0%

Running 00:00:32

The Capture dialog box displays the number of packets of various protocols that have been captured, and the percentage of all captured traffic consisting of those protocols. In Figure 4.32 a total of 707 packets have been captured, of which 363 (51.3%) are TCP packets, 4 (0.6%) are UDP packets, and 340 (48.1%) are ARP (Address Resolution Protocol) packets. You can stop the capture at any time by clicking the **Stop** button.

Ring Buffer Captures

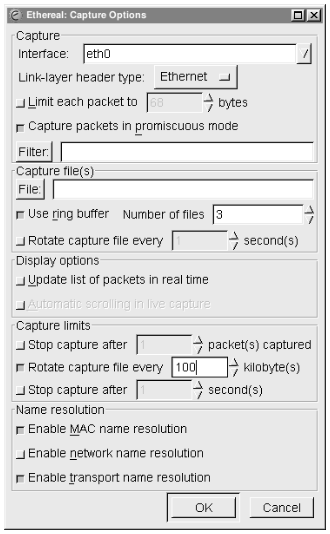
There are applications in which it makes sense to capture network traffic to a series of smaller files. Frequently, you may want to limit the number of such smaller files, and delete the oldest when starting a new one. Such a structure is called a ring buffer, because conceptually the data fills up a buffer and when it reaches the end it loops back to the beginning.

There are certain questions that need to be answered about using ring buffer files:

1. How many capture files in the ring buffer?
2. What are those capture files named?
3. When do we rotate to the next capture file?

To enable ring buffer captures, access the Capture Options dialog box and enable the **Use ring buffer** checkbox. The appearance of the Capture Options dialog box will change, as shown in figure 4.33.

Figure 4.33 Capture Options Dialog Box: Use Ring Buffer Selected



Ethereal: Capture Options

Capture
Interface: eth0
Link-layer header type: Ethernet
☐ Limit each packet to 68 bytes
☑ Capture packets in promiscuous mode
Filter:

Capture file(s)
File:
☑ Use ring buffer Number of files 3
☐ Rotate capture file every 1 second(s)

Display options
☐ Update list of packets in real time
☐ Automatic scrolling in live capture

Capture limits
☐ Stop capture after 1 packet(s) captured
☑ Rotate capture file every 100 kilobyte(s)
☐ Stop capture after 1 second(s)

Name resolution
☑ Enable MAC name resolution
☐ Enable network name resolution
☑ Enable transport name resolution

The **Rotate capture file every… second(s)** checkbox becomes available and the **Stop capture after… kilobytes captured** checkbox is renamed **Rotate capture file very… kilobyte(s)** and becomes unavailable.

The **Number of files** textbox allows you to choose how many files are in the ring buffer. If you choose zero, the number of ring buffer files is assumed to be infinite–no old files will be deleted to make room for new files.

The **File** textbox provides the base name for the filenames in the capture ring buffer. The base name is broken up into a prefix and a suffix. The filename of a ring buffer capture file is prefix_NNNNN_YYYYMMDDhhmmss.suffix. Where NNNNN is a five-digit zero-padded count indicating the sequence number of the ring buffer file, YYYY is a four-digit year, MM is the two-digit zero-padded month, DD is a two-digit zero-padded date, hh is a two-digit zero-padded hour, mm is a two-digit zero-padded minute, and ss is a two-digit zero-padded second. For example, if the file *foo.bar.libpcap* was the fifth capture file in the ring buffer created at 23:21:01 on January 8, 2004, it would be named *foo.bar_00005_20040108232101.libpcap*. It is important to note that the sequence numbers in the filenames increase monotonically. If a ring buffer has three files in it, when the fourth capture file is started it will have the sequence number 00004, and the file with the sequence number 00001 will be deleted. The sequence numbers are not recycled as we loop through the ring.

The **Rotate capture file every… kilobyte(s)** textbox and the optional **Rotate capture file every… second(s)** textbox allow you to choose when the capture files will be rotated. You must provide a kilobyte limit to the size of a capture file in the ring buffer by entering a number (or accepting the default value) in the **Rotate capture file every… kilobyte(s)** textbox. If a capture file reaches the number of kilobytes you have specified, a new capture file will be created to store any new packets captured, and the oldest capture file in the ring buffer may be deleted if the new capture file puts you over the limit specified in the **Number of files** textbox. If you enable the **Rotate capture file every… second(s)** checkbox and enter a number of seconds in the **Rotate capture file every… second(s)** textbox, if a capture file has been open for the number of seconds you specify, a new capture file will be created to store any new packets captured. The oldest capture file in the ring buffer may then be deleted if the new capture file puts you over the limit specified in the **Number of files** textbox.

NOTE

The **Use ring buffer** checkbox is incompatible with the **Update list of packets in real time** checkbox, and Ethereal will not allow you to enable **Use ring buffer** if you have already enabled **Update list of packets in real time**. Unfortunately, the reverse is not true–Ethereal will allow you to select **Update list of packets in real time** if **Use ring buffer** has already been selected. When this occurs, the **Use Ring buffer** checkbox will automatically (and without warning) be disabled. This, in turn, causes the **Rotate capture file every… kilobyte(s)** checkbox to revert to **Stop capture after… kilobyte(s)**, but it is left still selected, and with a value set (1 kilobyte by default). The net result is that not only do you not get a ring buffer capture in such a situation, but you also get your capture stopped after 1 byte.

Tools & Traps…

Handling Large Captures

Into everyone's life eventually falls a problem that involves enormous amounts of network data to analyze. Maybe it's an intermittent problem that happens only every couple of days where you need to see the message exchange that leads up to the problem. Maybe it's a problem on a fairly active network. Whatever the reason, the issue of capturing and analyzing large amounts of network traffic is a common one. As captures become larger, Ethereal consumes more memory, and filtering or finding packets begins to take a very long time.

In these situations it is best to use Tethereal, the console-based version of Ethereal, to do the actual capture and initial processing of the data. To capture to from an interface <interface> to a file <savefile> use this command:

```
tethereal –i <interface> -w <savefile>
```

If you have a limited amount of space and/or want to limit the size of your capture files, you can use the ring buffer functionality with Tethereal to capture from interface <interface> to <num_capture_files> capture files with maximum size each <filesize> and base filename <savefile> by executing the following at the command line:

```
tethereal –i <interface> -w <savefile> -b <num_capture_files> -a
filesize:<filesize>
```

Once you have captured the data you need, you can then use Tethereal to prune down the capture to a more manageable size. To use a display filter string <filter string> to filter a capture file <savefile> and save the results to a new capture file <newsavefile>, you would execute the following at the command line:

```
tethereal –r <savefile> -w <newsavefile> -R <filter string>
```

An example might be if you needed to extract all packets from the capture file that were captured between Jan 8, 2004 22:00 and Jan 8, 2004 23:00. To perform this feat you would execute the following at the command line:

Continued

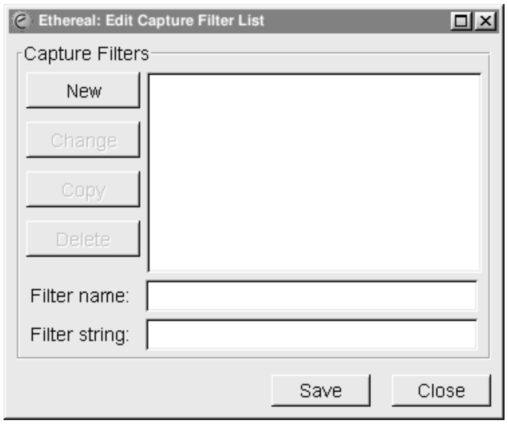
```
tethereal -r <savefile> -w <newsavefile> -R '(frame.time >= "Jan 8,
2004 22:00:00.00" ) && (frame.time <= "Jan 8, 2004 23:00:00.00")'
```

Once you have pruned the data down to a size where Ethereal's performance is workable, open the capture file in Ethereal to perform more involved analysis.

Edit Capture Filter List

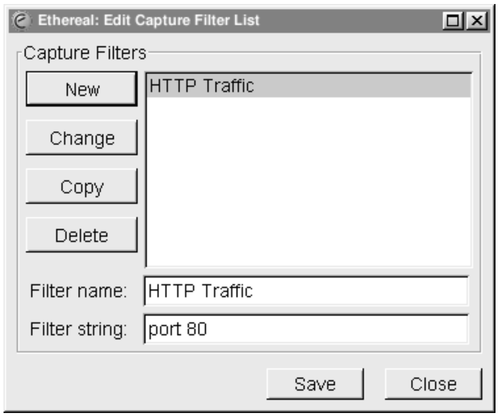
The Edit Capture Filter List dialog box is displayed by selecting **Capture | Capture Filters…** (Figure 4.34).

Figure 4.34 Edit Capture Filter List Dialog Box



This dialog box allows you to create new tcpdump-style capture filters, described in Chapter 5, and to save them for later use. To create a new capture filter, provide a name for your filter in the **Filter name** textbox, provide a tcpdump style capture filter string in the **Filter string** textbox, and click the **New** button. In Figure 4.35 we have created a capture filter named HTTP Traffic that provides filter string port 80.

Figure 4.35 Edit Capture Filter List Dialog Box Example



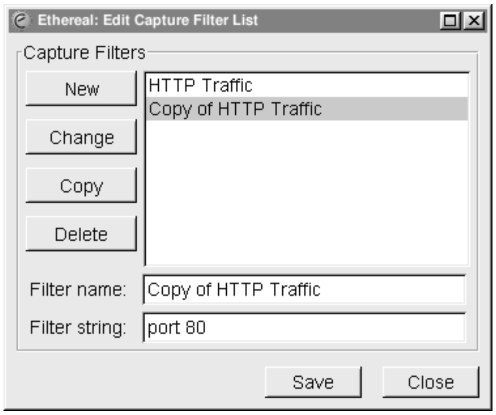
HTTP Traffic

Filter name: HTTP Traffic

Filter string: port 80

You can select an existing capture filter from the **Capture Filters** list and choose to change, delete, or copy it. To change an existing capture filter, select it from the **Capture Filters** list, change its name in the **Filter name** textbox and/or change its tcpdump style capture filter string in the **Filter string** textbox and then click the **Change** button. To copy an existing capture filter, select the capture filter from the **Capture Filters** list and click the **Copy**, as shown in Figure 4.36.

Figure 4.36 Edit Capture Filter List Dialog Box: Copy



HTTP Traffic
Copy of HTTP Traffic

Filter name: Copy of HTTP Traffic

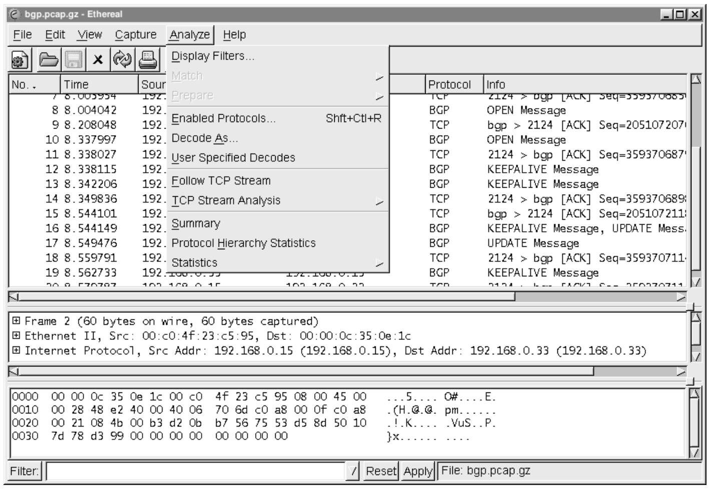
Filter string: port 80

You can delete a capture filter by selecting it from the **Capture Filters** list and clicking the **Delete** button. If you wish to have your list of capture filters available in a subsequent Ethereal session, then you must click the **Save** button to save them to disk.

Analyze

The **Analyze Menu** is shown in Figure 4.37, and it's options are explained in Table 4.10.

Figure 4.37 Analyze Menu



```
No. .  Time       Sour                              Protocol  Info
    8 8.004042   192.                              BGP       OPEN Message
    9 8.208048   192.                              TCP       bgp > 2124 [ACK] Seq=205107207
   10 8.337997   192.                              BGP       OPEN Message
   11 8.338027   192.                              TCP       2124 > bgp [ACK] Seq=359370687
   12 8.338115   192.                              BGP       KEEPALIVE Message
   13 8.342206   192.                              BGP       KEEPALIVE Message
   14 8.349836   192.                              TCP       2124 > bgp [ACK] Seq=359370689
   15 8.544101   192.                              TCP       bgp > 2124 [ACK] Seq=205107211
   16 8.544149   192.                              BGP       KEEPALIVE Message, UPDATE Mess
   17 8.549476   192.                              BGP       UPDATE Message
   18 8.559791   192.                              TCP       2124 > bgp [ACK] Seq=359370711
   19 8.562733   192.                              BGP       KEEPALIVE Message
```

```
⊞ Frame 2 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:c0:4f:23:c5:95, Dst: 00:00:0c:35:0e:1c
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
```

```
0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 28 48 e2 40 00 40 06  70 6d c0 a8 00 0f c0 a8   .(H.@.@. pm......
0020  00 21 08 4b 00 b3 d2 0b  b7 56 75 53 d5 8d 50 10   .!.K.... .VuS..P.
0030  7d 78 d3 99 00 00 00 00  00 00 00 00               }x...... ....
```

Table 4.10 Analyze Menu Options

Menu Option	Description
Display Filters…	Edit the display filters.
Match	Submenu for preparing and applying a display filter based upon the protocol field selected in the Protocol Tree Window.
Prepare	Submenu for preparing a display filter based upon the protocol field selected in the Protocol Tree Window.
Enabled Protocols…	Enable and disable the decoding of individual protocols.
Decode As…	Specify decoding certain packets as being part of a particular protocol.
User Specified Decodes	Report which user specified decodes are currently in force.
Follow TCP Stream	Display an entire TCP stream at once.
TCP Stream Analysis	Submenu for choosing a TCP Stream Analysis tool.

Continued

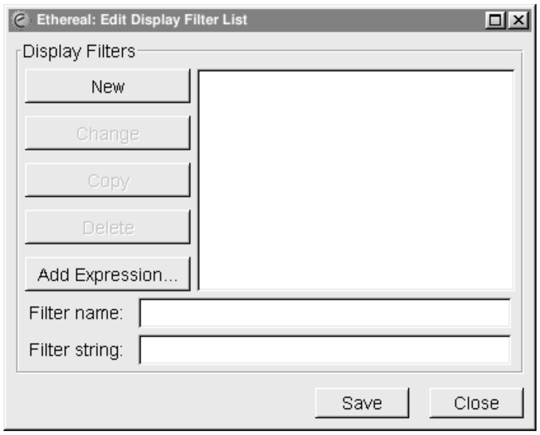
Table 4.10 Analyze Menu Options

Menu Option	Description
Summary	Display a summary of the capture file.
Protocol Hierarchy Statistics	Display statistics in a tree view for the protocols in the capture.
Statistics	Submenu for choosing a Statistics tool.

Edit Display Filter List

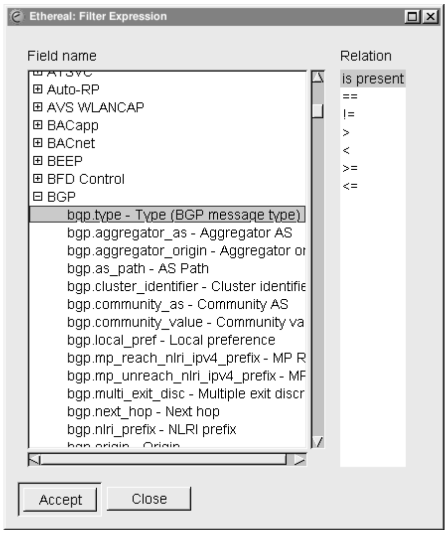
The Edit Display Filter List dialog box, shown in Figure 4.38, can be displayed by selecting **Analyze | Display Filter...**.

Figure 4.38 Edit Display Filter List Dialog Box



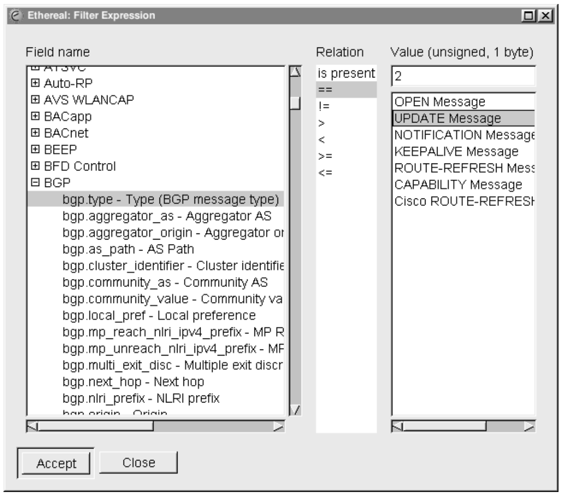
This dialog box is designed to help you construct a filter string. To create a new filter string, click the **Add Expression** button. The Filter Expression dialog box (Figure 4.39) will be displayed.

Figure 4.39 Filter Expression Dialog Box


Ethereal: Filter Expression
Field name
Relation
ATSVC
Auto-RP
AVS WLANCAP
BACapp
BACnet
BEEP
BFD Control
BGP
bgp.type - Type (BGP message type)
bgp.aggregator_as - Aggregator AS
bgp.aggregator_origin - Aggregator or
bgp.as_path - AS Path
bgp.cluster_identifier - Cluster identifie
bgp.community_as - Community AS
bgp.community_value - Community va
bgp.local_pref - Local preference
bgp.mp_reach_nlri_ipv4_prefix - MP R
bgp.mp_unreach_nlri_ipv4_prefix - MF
bgp.multi_exit_disc - Multiple exit discr
bgp.next_hop - Next hop
bgp.nlri_prefix - NLRI prefix
bgp.origin - Origin
is present
==
!=
>
<
>=
<=
Accept
Close

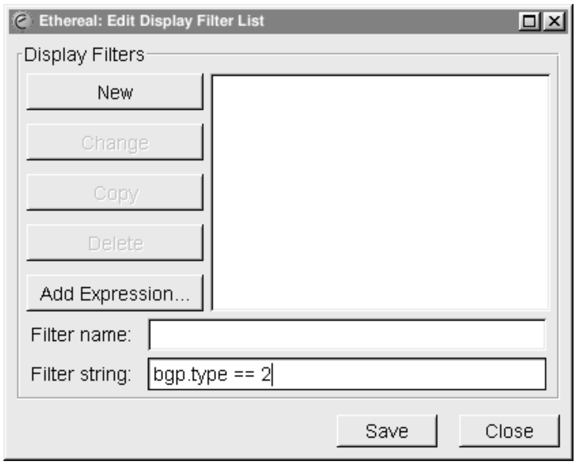
Select the protocol you are interested in for your filter expression and expand it to show which of its fields can be filtered. Select the desired filter field. When you pick a relation other than **is present**, the Filter Expression dialog box will change to show your options for that field, as shown in Figure 4.40.

Figure 4.40 Filter Expression Dialog: Equality


Ethereal: Filter Expression
Field name
Relation
Value (unsigned, 1 byte)
ATSVC
Auto-RP
AVS WLANCAP
BACapp
BACnet
BEEP
BFD Control
BGP
bgp.type - Type (BGP message type)
bgp.aggregator_as - Aggregator AS
bgp.aggregator_origin - Aggregator or
bgp.as_path - AS Path
bgp.cluster_identifier - Cluster identifie
bgp.community_as - Community AS
bgp.community_value - Community va
bgp.local_pref - Local preference
bgp.mp_reach_nlri_ipv4_prefix - MP R
bgp.mp_unreach_nlri_ipv4_prefix - MF
bgp.multi_exit_disc - Multiple exit discr
bgp.next_hop - Next hop
bgp.nlri_prefix - NLRI prefix
bgp.origin - Origin
is present
==
!=
>
<
>=
<=
2
OPEN Message
UPDATE Message
NOTIFICATION Message
KEEPALIVE Message
ROUTE-REFRESH Mess
CAPABILITY Message
Cisco ROUTE-REFRESH
Accept
Close

In this case, we have chosen the equality (==) relation. You can choose the value you wish to match and click the **Accept** button. The result will be to insert the filter expression you just constructed into the **Filter string:** textbox (Figure 4.41)

Figure 4.41 Edit Display Filter List Dialog Box: Filter String

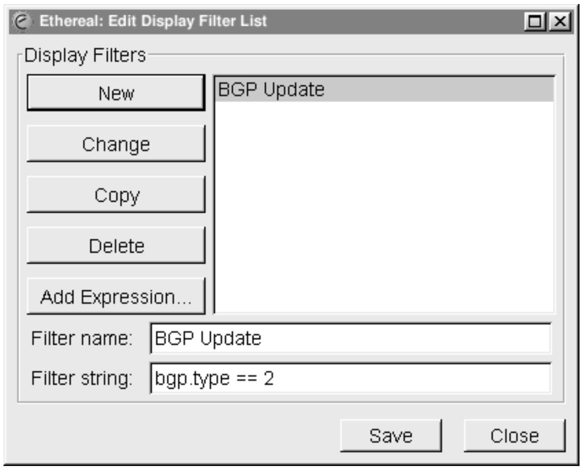


Filter name:

Filter string: bgp.type == 2

If you wish to save the filter string you have just created, type a name in the **Filter name** textbox and click the **New** button. The filter string will be added to the Display Filters List dialog box (Figure 4.42)

Figure 4.42 Edit Display Filter List Dialog Box: Filter Name



BGP Update

Filter name: BGP Update

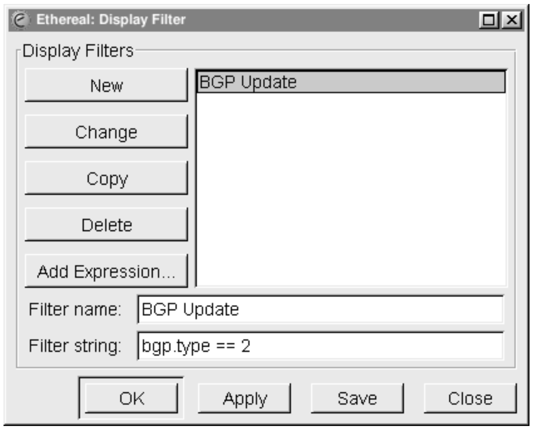
Filter string: bgp.type == 2

You can select an existing display filter from the list and choose to change, delete, or copy it. To change an existing display filter, select it from the list, change its name in the **Filter name** textbox and/or change its display filter string in the **Filter string** textbox and then click the **Change** button. To copy an existing display filter, select it from the list and click the **Copy** button.

You can save the list for use in later Ethereal sessions by clicking the **Save** Button.

If you have accessed the Edit Display Filter List dialog box from the filter bar or some other part of Ethereal from which you can apply a display filter, then an **OK** button will also be available. Use this button to apply the filter and close the dialog box. Use the **Apply** button to apply your filter and leave the dialog box open (see Figure 4.43).

Figure 4.43 Display Filter Dialog Box: OK/Apply Buttons



Ethereal: Display Filter

Display Filters

BGP Update

Filter name: BGP Update

Filter string: bgp.type == 2

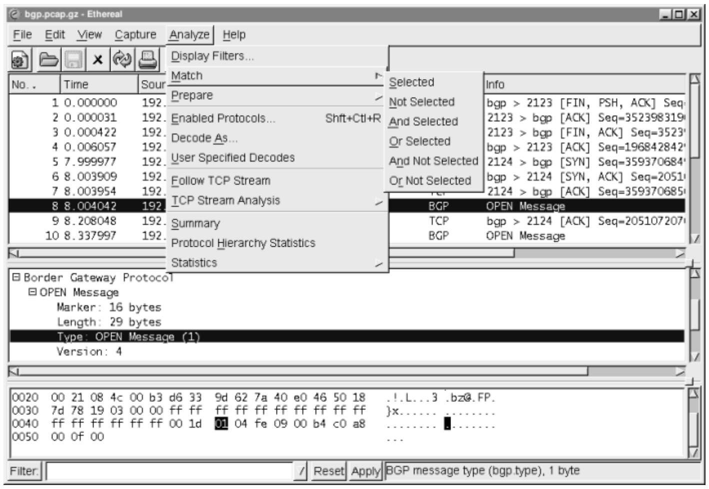
This has only been a very rudimentary introduction to display filtering; a more in-depth discussion can be found in the Chapter 5.

Match and Prepare Submenus

The **Match** and **Prepare** submenus have the same options and behave in the same way with one exception; the **Prepare** submenu items prepare a display filter string and place it in the **Filter** textbox. The **Match** submenu items prepare a display filter string, place it in the **Filter** textbox and apply it to the capture. Because of their close similarity we will only discuss the **Match** submenu.

The **Match** submenu becomes available when you have selected a field in the Protocol Tree Window with an associated filter name that can be used in a display filter string. An example is shown in Figure 4.44.

Figure 4.44 Match Submenu



```
No. .  Time      Sour                                  Info
    1 0.000000   192.                                  bgp > 2123 [FIN, PSH, ACK] Seq
    2 0.000031   192.                                  2123 > bgp [ACK] Seq=35239831
    3 0.000422   192.                                  2123 > bgp [FIN, ACK] Seq=3523
    4 0.006057   192.                                  bgp > 2123 [ACK] Seq=19684284
    5 7.999977   192.                                  2124 > bgp [SYN] Seq=35937068
    6 8.003909   192.                                  bgp > 2124 [SYN, ACK] Seq=2051
    7 8.003954   192.                                  2124 > bgp [ACK] Seq=35937068
    8 8.004042   192.                          BGP     OPEN Message
    9 8.208048   192.                          TCP     bgp > 2124 [ACK] Seq=20510720
   10 8.337997   192.                          BGP     OPEN Message
```

```
⊟ Border Gateway Protocol
  ⊟ OPEN Message
      Marker: 16 bytes
      Length: 29 bytes
      Type: OPEN Message (1)
      Version: 4
```

```
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ........ ........
0050  00 0f 00                                           ...
```

BGP message type (bgp.type), 1 byte

In Table 4.11, we can see the filter string that would be put in the **Filter:** textbox for each of the **Match** submenu options for the example in Figure 4.44.

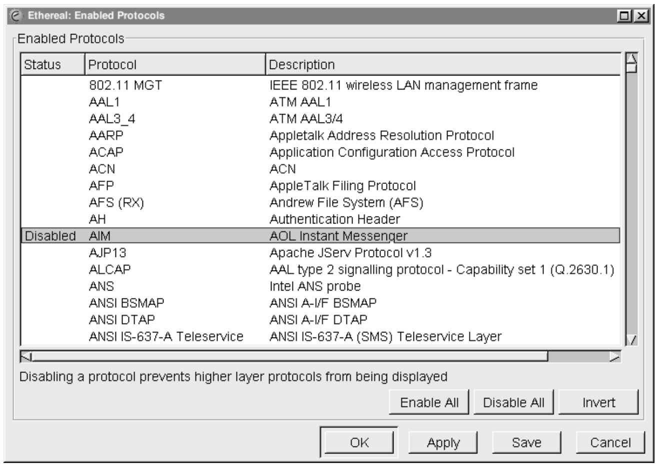
Table 4.11 Match Submenu Option Examples

Menu Option	Display Filter String
Selected	bgp.type == 1
Not Selected	!(bgp.type == 1)
And Selected	(ip.addr == 192.168.0.15) && (bgp.type == 1)
Or Selected	(ip.addr == 192.168.0.15) || (bgp.type == 1)
And Not Selected	(ip.addr == 192.168.0.15) && !(bgp.type == 1)
Or Not Selected	(ip.addr == 192.168.0.15) || !(bgp.type == 1)

Enabled Protocols

The Enabled Protocols dialog box, shown in Figure 4.45, is displayed by selecting **Analyze | Enabled Protocols…**.

Figure 4.45 Enabled Protocols Dialog Box



Status	Protocol	Description
	802.11 MGT	IEEE 802.11 wireless LAN management frame
	AAL1	ATM AAL1
	AAL3_4	ATM AAL3/4
	AARP	Appletalk Address Resolution Protocol
	ACAP	Application Configuration Access Protocol
	ACN	ACN
	AFP	AppleTalk Filing Protocol
	AFS (RX)	Andrew File System (AFS)
	AH	Authentication Header
Disabled	AIM	AOL Instant Messenger
	AJP13	Apache JServ Protocol v1.3
	ALCAP	AAL type 2 signalling protocol - Capability set 1 (Q.2630.1)
	ANS	Intel ANS probe
	ANSI BSMAP	ANSI A-I/F BSMAP
	ANSI DTAP	ANSI A-I/F DTAP
	ANSI IS-637-A Teleservice	ANSI IS-637-A (SMS) Teleservice Layer

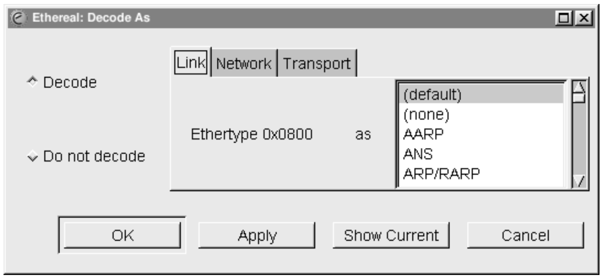
Disabling a protocol prevents higher layer protocols from being displayed

This dialog box allows you to enable or disable the decoding of one or more protocols. You can do this by clicking its Status column to toggle its status between *Enabled* and *Disabled*. Additionally, you can enable all protocols by clicking the **Enable All** button, disable all protocols by clicking the **Disable All** button, or enable all disabled protocols and disable all enabled protocols by clicking the **Invert** button. You can apply these settings to all Ethereal sessions by clicking the **Save** button.

Decode As

To force the decode of a packet, select it in the Summary Window and then select **Analyze | Decode As…**. The Decode As dialog box will be displayed, as shown in Figure 4.46.

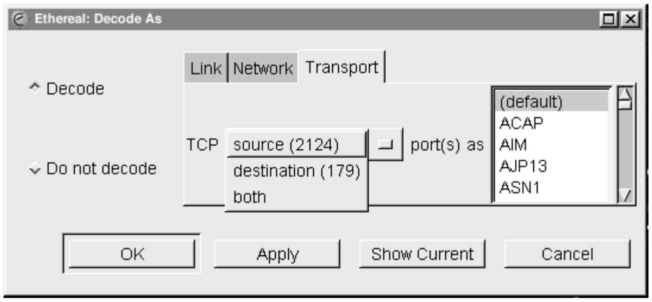
Figure 4.46 Decode As Dialog Box: Link Tab



When Ethereal is decoding a packet it uses magic numbers in each protocol to decide which dissector to use to decode subsequent parts of the packet. Magic numbers are values that specify some higher-level protocol, like Ethertype 0x0800 specifying that an Ethernet packet contains an IP packet, or IP protocol 6 specifying that an IP packet contains a TCP payload, or TCP port 179 specifying that a TCP packet is carrying a BGP payload. There are occasions when you want to override Ethereal's choices in how to decode subsequent parts of the packet based on these magic numbers. The most common examples involve TCP ports. Ethereal frequently decides which dissector to call next for a TCP packet based upon the source or destination port. We may be running a protocol over a non-standard port, like running HTTP over port 7000 for example. The **Decode As** feature allows us to tell Ethereal about such non-standard cases.

Ethereal allows the user to force decodes based upon the magic numbers in the link layer, network layer, or transport layer. For the transport layer we have the option of decoding based on source, destination or both, as shown in Figure 4.47.

Figure 4.47 Decode As Dialog Box: Transport Tab



To force a particular decode you need to answer the questions:

1. After which layer do I want to start forcing my custom decode?
2. Which magic number do I want to key off of to determine whether to decode a packet with my custom decode?
3. Which protocol do I want the remaining traffic in the packet decoded as?

To choose the layer at which you want to start forcing your custom decode, select the appropriate tab (**Link**, **Network**, or **Transport**). You have a choice of which magic numbers to pick for the transport layer, where you can pick **source** port, **destination** port, or **both**. Then, you may select from the list of protocols as to how you want the remaining traffic in the packet decoded.

Click the **Show Current** button to open the Decode As: Show dialog box in order to see which decodes are currently being forced.

Decode As: Show

The Decode As: Show dialog box (Figure 4.48) can also be displayed by selecting **Analyze | User Specified Decodes** from the menu bar.

Figure 4.48 Decode As: Show

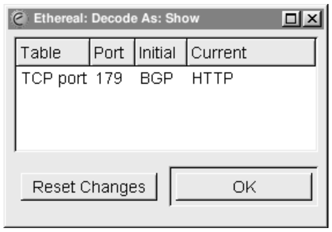


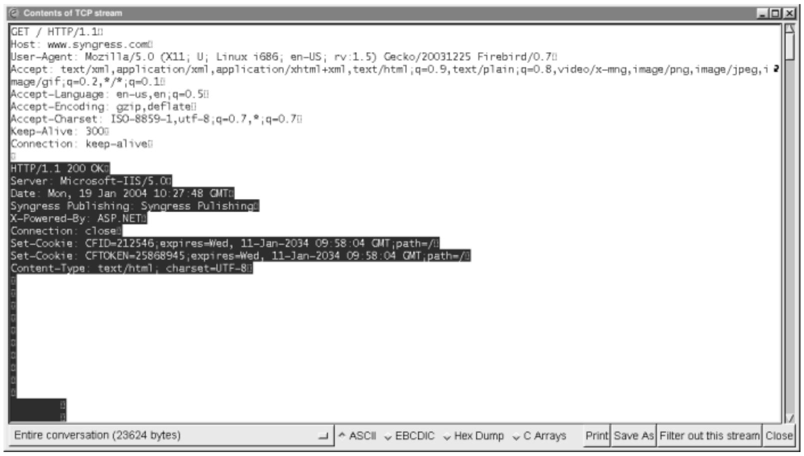
Table	Port	Initial	Current
TCP port	179	BGP	HTTP

This dialog box displays the decodes you have specified through the Decode As Dialog box, one per line. The **Table** column shows the type of magic number for which we are showing the alternate decode, in this case the TCP port. The **Port** column shows the magic number for which we are providing an alternate decode, in this case 179. The **Initial** column shows the dissector that would normally be used to decode the payload of a packet with this magic number and magic number type, in this case BGP. And, finally, the **Current** column shows the dissector currently being used to decode the payload of packets having this magic number and magic number type, in this case HTTP.

Contents of TCP Stream

The Contents of TCP Stream window (Figure 4.49) can be displayed by selecting a TCP packet in the Summary Window and then selecting **Analyze | Follow TCP Stream** from the menu bar.

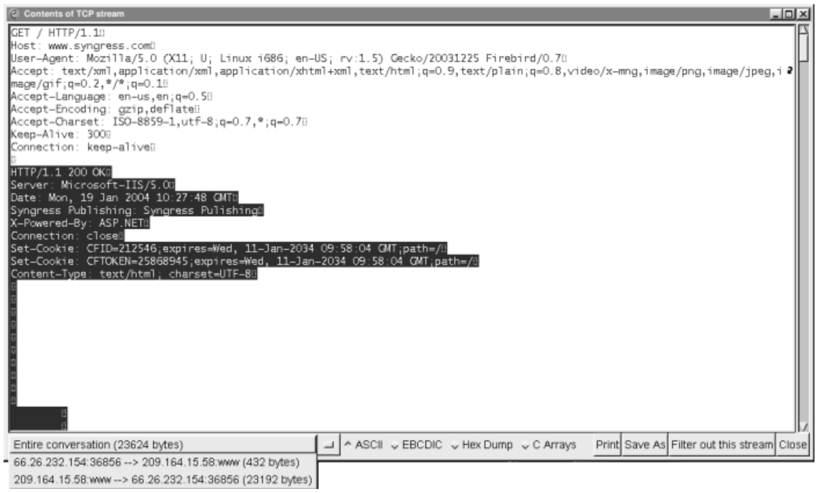
Figure 4.49 Contents of TCP Stream Window



```
GET / HTTP/1.1
Host: www.syngress.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.5) Gecko/20031225 Firebird/0.7
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,i
mage/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 19 Jan 2004 10:27:48 GMT
Syngress Publishing: Syngress Pulishing
X-Powered-By: ASP.NET
Connection: close
Set-Cookie: CFID=212546;expires=Wed, 11-Jan-2034 09:58:04 GMT;path=/
Set-Cookie: CFTOKEN=25868945;expires=Wed, 11-Jan-2034 09:58:04 GMT;path=/
Content-Type: text/html; charset=UTF-8
```

In this example, a TCP packet that was part of an HTTP conversation with the web server for www.syngress.com is shown. By default, one side of the conversation is shown in red (the upper portion), the other in blue (the lower portion). For readability purposes the side of the conversation that is normally blue has been changed to white text on a dark blue background, using the **TCP Streams** color selector. By scrolling down in this window you are able to see all of the data exchanged during this TCP conversation. If you click the **Entire conversation** selector you can choose between displaying the entire conversation, or one of the directions (Figure 4.50).

Figure 4.50 Follow TCP Stream: Direction Selector


Contents of TCP stream
Entire conversation (23624 bytes)
66.26.232.154:36856 --> 209.164.15.58:www (432 bytes)
209.164.15.58:www --> 66.26.232.154:36856 (23192 bytes)
ASCII
EBCDIC
Hex Dump
C Arrays
Print
Save As
Filter out this stream
Close

```
GET / HTTP/1.1
Host: www.syngress.com
User-Agent: Mozilla/5.0 (X11; U; Linux i686; en-US; rv:1.5) Gecko/20031225 Firebird/0.7
Accept: text/xml,application/xml,application/xhtml+xml,text/html;q=0.9,text/plain;q=0.8,video/x-mng,image/png,image/jpeg,i
mage/gif;q=0.2,*/*;q=0.1
Accept-Language: en-us,en;q=0.5
Accept-Encoding: gzip,deflate
Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7
Keep-Alive: 300
Connection: keep-alive

HTTP/1.1 200 OK
Server: Microsoft-IIS/5.0
Date: Mon, 19 Jan 2004 10:27:48 GMT
Syngress Publishing: Syngress Pulishing
X-Powered-By: ASP.NET
Connection: close
Set-Cookie: CFID=212546;expires=Wed, 11-Jan-2034 09:58:04 GMT;path=/
Set-Cookie: CFTOKEN=25868945;expires=Wed, 11-Jan-2034 09:58:04 GMT;path=/
Content-Type: text/html; charset=UTF-8
```

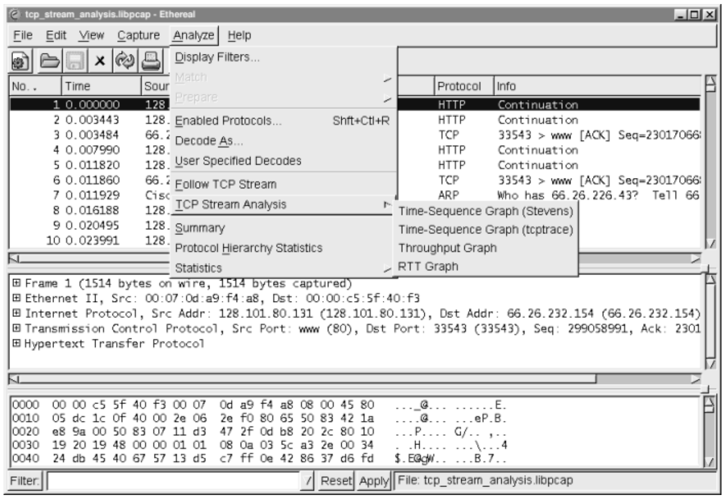
Clicking the **Save As** button will bring up a Save As dialog box for you to save the stream contents as a text file. Clicking the **Print** button will print the capture as text. Note there is no dialog box associated with the **Print** button. The **Filter out this stream** button will append the necessary filter string to the one in the filter bar and close the Contents of TCP Stream window. This can be very handy when going through a large capture. As you look at the possible TCP streams of interest one by one and exclude them from the Summary Window, you have finished considering them so only the unconsidered data remains.

You also have the option of choosing how the TCP stream is presented. In Figure 4.50 the **ASCII** option is selected. By choosing the **EBCDIC** option you could cause the stream to be presented with EBCDIC (Extended Binary Coded Decimal Interchange Code, a proprietary IBM character set). If you choose the **Hex Dump** option you will see a hexadecimal dump of the TCP stream. And, if you choose the **C Arrays** option, the TCP stream will be shown as a series of C arrays (arrays in the C programming language).

TCP Stream Analysis Submenu

The **TCP Stream Analysis** submenu, shown in Figure 4.51, can be displayed by selecting a TCP packet in the Summary Window and selecting **Analyze | TCP Stream Analysis** from the menu bar. **TCP Stream Analysis** submenu options are shown in Table 4.12.

Figure 4.51 TCP Analysis Submenu


tcp_stream_analysis.libpcap - Ethereal
File
Edit
View
Capture
Analyze
Help
Display Filters...
Match
Prepare
Enabled Protocols... Shft+Ctl+R
Decode As...
User Specified Decodes
Follow TCP Stream
TCP Stream Analysis
Summary
Protocol Hierarchy Statistics
Statistics
Time-Sequence Graph (Stevens)
Time-Sequence Graph (tcptrace)
Throughput Graph
RTT Graph
Filter:
Reset
Apply
File: tcp_stream_analysis.libpcap

```
No. .  Time       Sour          Protocol  Info
    1 0.000000   128.          HTTP      Continuation
    2 0.003443   128.          HTTP      Continuation
    3 0.003484   66.2          TCP       33543 > www [ACK] Seq=23017066
    4 0.007990   128.          HTTP      Continuation
    5 0.011820   128.          HTTP      Continuation
    6 0.011860   66.2          TCP       33543 > www [ACK] Seq=23017066
    7 0.011929   Cisc          ARP       Who has 66.26.226.43?  Tell 66
    8 0.016188   128.
    9 0.020495   128.
   10 0.023991   128.
```

```
⊞ Frame 1 (1514 bytes on wire, 1514 bytes captured)
⊞ Ethernet II, Src: 00:07:0d:a9:f4:a8, Dst: 00:00:c5:5f:40:f3
⊞ Internet Protocol, Src Addr: 128.101.80.131 (128.101.80.131), Dst Addr: 66.26.232.154 (66.26.232.154)
⊞ Transmission Control Protocol, Src Port: www (80), Dst Port: 33543 (33543), Seq: 299058991, Ack: 2301
⊞ Hypertext Transfer Protocol
```

```
0000  00 00 c5 5f 40 f3 00 07  0d a9 f4 a8 08 00 45 80   ..._@... ......E.
0010  05 dc 1c 0f 40 00 2e 06  2e f0 80 65 50 83 42 1a   ....@... ...eP.B.
0020  e8 9a 00 50 83 07 11 d3  47 2f 0d b8 20 2c 80 10   ...P.... G/.. ,..
0030  19 20 19 48 00 00 01 01  08 0a 03 5c a3 2e 00 34   . .H.... ...\...4
0040  24 db 45 40 67 57 13 d5  c7 ff 0e 42 86 37 d6 fd   $.E@gW.. ...B.7..
```

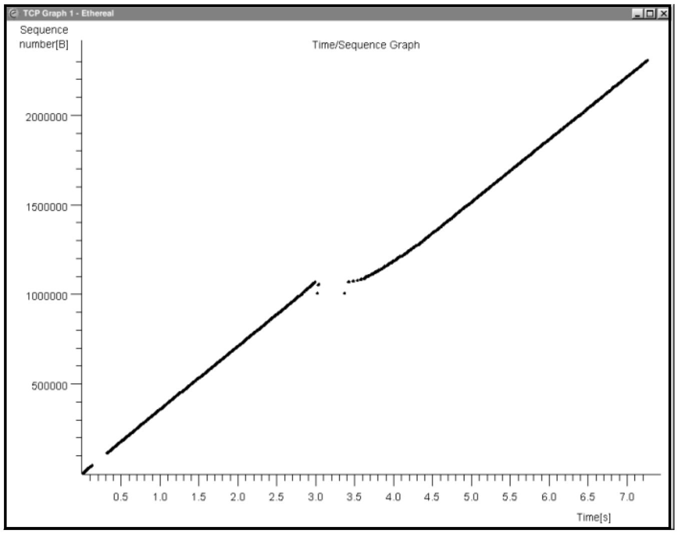
Table 4.12 TCP Stream Analysis Submenu Options

Menu Option	Description
Time-Sequence Graph (Stevens)	Displays a time-sequence graph in the style used by W. Richard Stevens' TCP/IP Illustrated book.
Time-Sequence Graph (tcptrace)	Displays a time-sequence graph in the style used by the tcptrace program, which can be found at www.tcptrace.org/.
Throughput Graph	Displays a graph of throughput versus time.
RTT Graph	Displays a graph of round trip time (RTT) versus sequence number.

Time-Sequence Graph (Stevens)

The time-sequence graph (Stevens) produces a simple graph of TCP sequence number versus time for the TCP stream containing the packet that was selected in the Summary window. The first derivative of this graph is the TCP traffic throughput. In an ideal situation where we have a constant throughput, the graph would be a straight rising line with its slope equaling the throughput. Unfortunately, things are seldom ideal, and you can learn a lot about where the source of throughput issues are coming from by looking at the time-sequence graph. In Figure 4.52, there is a graph showing a throughput problem. You can reproduce this graph by selecting the first packet of the tcp_stream_analysis.libpcap capture file, and selecting **Analysis | TCP Stream Analysis | Time-Sequence Graph (Stevens)**. The captured file used in this graph is a classic example of TCP retransmit and the kind of issues you use the TCP Stream Analysis tool to debug. The full network capture can be found on the accompanying CD, and has been added to the collection of network captures on the Ethereal website.

Figure 4.52 Time-Sequence Graph (Stevens)

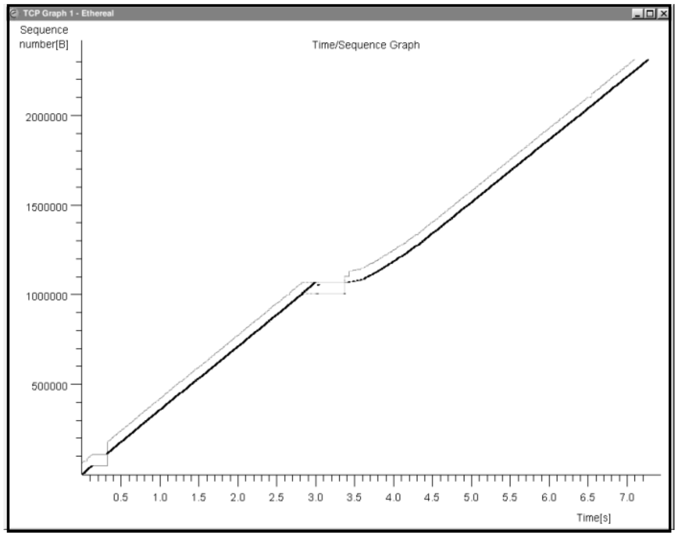


In Figure 4.52 (after about 0.3 seconds), the traffic has a nice even slope (constant throughput) until around 3 seconds, when there is a major disruption, as shown by the discontinuity in the graph. This gap suggests TCP retransmissions. The Steven's style time-sequence graph is simple, but you can see very clearly where your problems are.

Time-Sequence Graph (tcptrace)

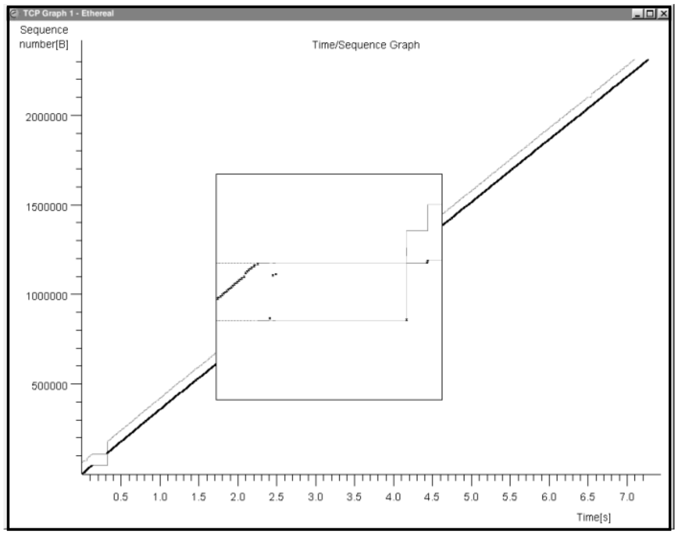
The time-sequence graph (tcptrace) is also primarily a graph of TCP sequence numbers vs. time. Unlike the Stevens' style time-sequence graph, it conveys a lot more information about the TCP stream. In Figure 4.53 you can see that the tcptrace style time-sequence graph of this stream looks very similar to the Stevens' style time-sequence graph.

Figure 4.53 Time-Sequence Graph (tcptrace)



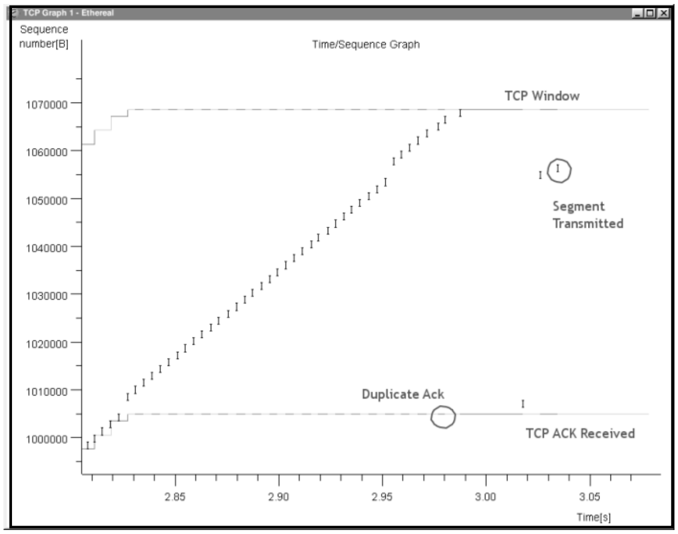
Explaining the elements shown in the tcptrace style time-sequence graph is made easier by using some of the graph manipulation tools that are available in all of the TCP stream analysis graphs. By performing a **Ctrl + right-click**, on the graph, you can magnify a portion of the graph, as shown in Figure 4.54.

Figure 4.54 Time-Sequence Graph (tcptrace): Magnify



The box in the middle of the graph in Figure 4.54 is magnifying the region of discontinuity where packet loss has occurred. To get an even better view of it we can use the zoom feature. By clicking on the graph with the middle mouse button (if you have one) you can zoom in on the part of the graph you are clicking on. **Shift + middle-click** zooms out. When you have zoomed in/out, clicking and dragging with the right mouse button on the graph will allow you to move around in the zoomed graph. A zoom in on the region of discontinuity is shown in Figure 4.55.

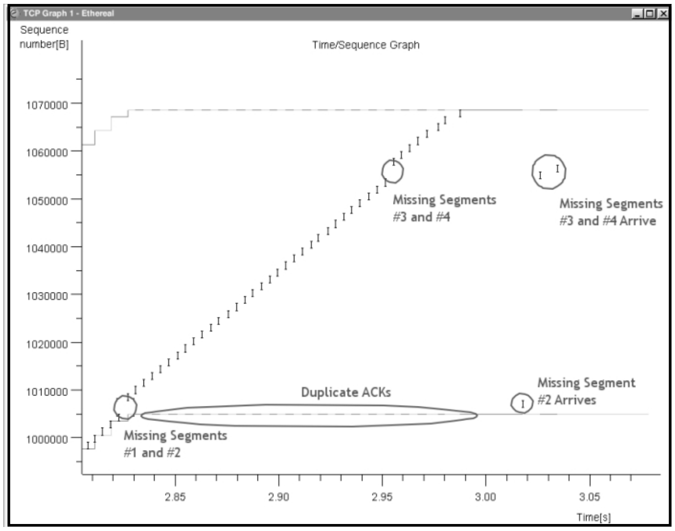
Figure 4.55 Time-Sequence Graph (tcptrace): Zoom



This is a zoom-in on the section of the graph just before the discontinuity. You can see the beginning of the discontinuity on the far right of the graph. Marked in bolded type are the different elements of the tcpgraph style time-sequence graph. The lower line represents the sequence number of the last ACK (TCP Acknowledgement) seen. The top line represents the TCP window. It consists of the sequence number of the last observed TCP ACK plus the last seen TCP window size advertised. The little hash marks on the lower line represent duplicate ACKs, and the little "I" bars represent transmitted segments.

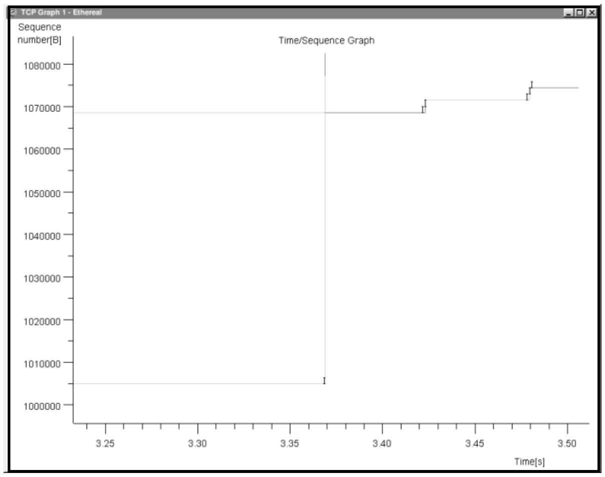
Figure 4.56 is the same graph as Figure 4.55, but with different annotations to magnify what went wrong for this TCP stream. The capture behind this graph was taken from the receiver of a large transmission over TCP. Therefore, we generally only see the segments that we are receiving from the far end. What is seen in this graph is that early on the receiver missed two segments. The receiver continued to ACK the last segment received, and to receive subsequent segments until the segments received filled up the TCP window. A couple of other segments were lost along the way. Finally, we receive the second missed segment, the third missed segment, and then the fourth missed segment. But, because the first missed segment has not yet turned up, the receiver keeps sending the same duplicate ACK.

Figure 4.56 Time-Sequence Graph(tcptrace): Diagnosis



In Figure 4.57 we see how this logjam finally resolves:

Figure 4.57 Time-Sequence Graph (tcptrace): Zoom in on Retransmit

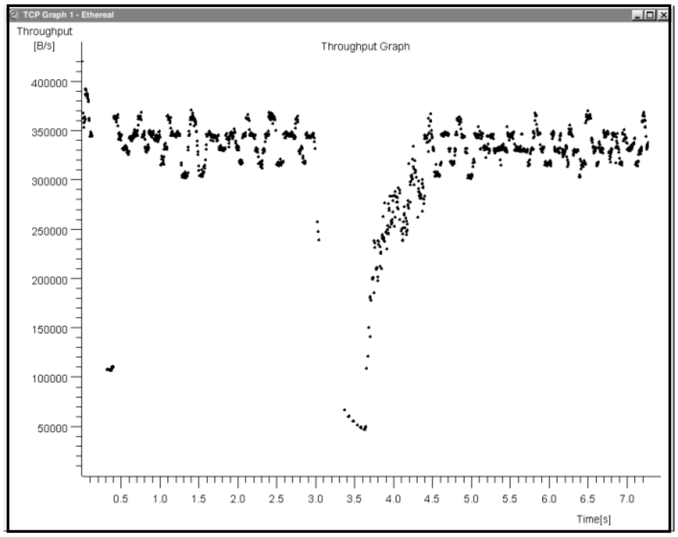


In Figure 4.57 you can see the missing segment, presumed to be a retransmit, arrive. At this point, an ACK is transmitted acknowledging the last received segment, the TCP window increases, and the receiver begins to receive segments again.

Throughput Graph

The throughput graph (Figure 4.58) shows the throughput of the TCP stream versus time.

Figure 4.58 Throughput Graph

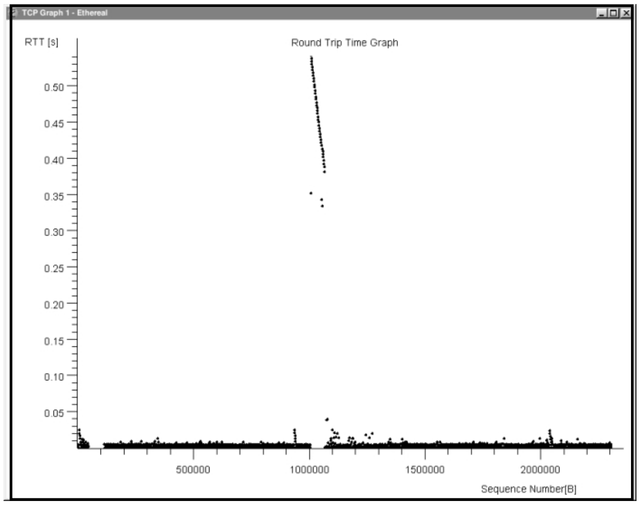


You can see in Figure 4.58 that the throughput fell off dramatically during the retransmit sequence seen in the time-sequence graphs.

RTT Graph

The RTT graph (Figure 4.59) shows the round trip time versus. sequence number.

Figure 4.59 RTT Graph

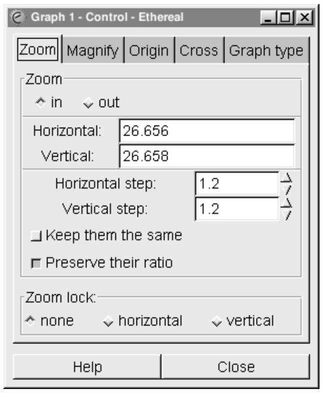


You can see the round trip time spike during around sequence number 1000000, roughly the same sequence number where we saw the discontinuity in the time-sequence graphs.

Graph Control

Throughout this section we will be referring to any of the windows containing a TCP stream analysis graph as a *graph window*. The term graph window may refer to a Stevens' or tcptrace style time-sequence graph**,** a throughput graph, or an RTT graph. Whenever a graph window is created, a Graph Control dialog box is also created, as shown in Figure 4.60.

Figure 4.60 Graph Control Dialog Box: Zoom Tab



Zoom
● in ○ out

Horizontal: 26.656
Vertical: 26.658

Horizontal step: 1.2
Vertical step: 1.2

□ Keep them the same
☑ Preserve their ratio

Zoom lock:
● none ○ horizontal ○ vertical

Notice that the number on the dialog box (1) matches the number on the graph window in Figure 4.56 (1). In the event that multiple graph windows are opened, you can use the index number to associate a Graph Control dialog box with its graph window.

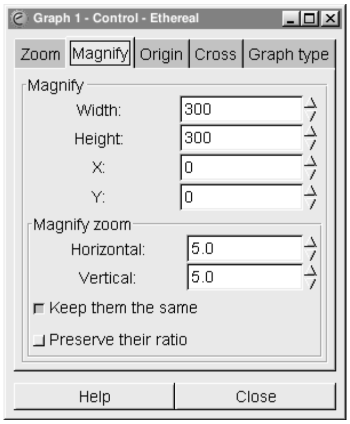
The **Zoom** tab, shown in Figure 4.60, allows us to set the parameters related to the zoom functionality of the graph function. The **Horizontal** and **Vertical** textboxes are not for user entry of data, rather they show the amount of zoom currently employed in the graph window.

The **Horizontal step** and **Vertical step** textboxes allow you to set the horizontal and vertical zoom factors applied to the graph when you **Shift + middle-click** in the graph window. If you enable the **Keep them the same** checkbox, then whenever you change either the horizontal step or vertical step, the other will be changed to the same value. The **Preserve their ratio** checkbox causes the ratio between the horizontal step and the vertical step to be preserved. If the horizontal step was 1.2 and the vertical step was 2.4, when you changed the horizontal step to 1.3, then the vertical step would automatically change to 2.6.

The **Zoom lock** section allows you to lock either the horizontal or vertical so that zoom is not applied to them. If you enable the **horizontal** option, no matter what the value is for horizontal step, zooming will not change the horizontal scale at all. This is also true if you select the **vertical** option; no matter what the value is for vertical step, zooming will not change the vertical scale at all.

The **Magnify** tab, shown in Figure 4.61, allows you to control the parameters associated with the magnify functionality.

Figure 4.61 Graph Control Dialog Box: Magnify Tab



Magnify

Width: 300

Height: 300

X: 0

Y: 0

Magnify zoom

Horizontal: 5.0

Vertical: 5.0

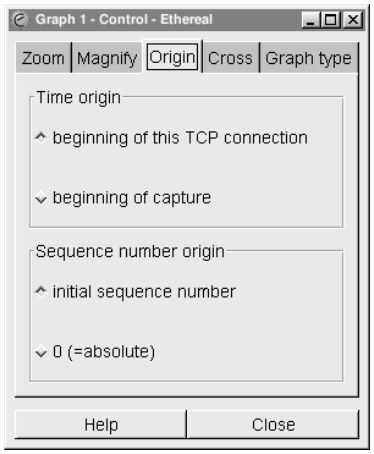
☑ Keep them the same

□ Preserve their ratio

The **Width** and **Height** textboxes allow you to set the width and height of the magnification box that is displayed when you **Ctrl + right-click** in the graph window. The **X:** and **Y:** textboxes allow you to set the x and y offset of the magnification box from the location of the mouse pointer. This can be handy to offset the magnification box to somewhere where it won't occlude the graph. The **Horizontal:** and **Vertical:** textboxes allow you to set the zoom factor used to blow up the graph in the agnification box. The **Keep them the same** checkbox will cause the horizontal and vertical zoom factors to change in accordance with one another, and the **Preserve their ratio** checkbox will cause the ratio between the horizontal and vertical zoom factor to remain constant.

The **Origin** tab, shown in Figure 4.62, allows you to change the various origins of the graph.

Figure 4.62 Graph Control Dialog Box: Origin Tab



Time origin

● beginning of this TCP connection

○ beginning of capture

Sequence number origin

● initial sequence number

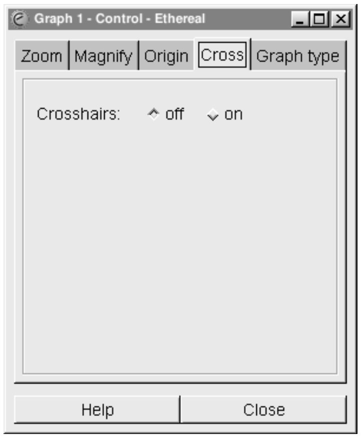
○ 0 (=absolute)

The **Time origin** section will allow you to choose the zero of time for your graph. If you select the **beginning of this TCP connection** option, you establish the beginning of the TCP connection as being graphed as your zero of time. If you select the **beginning of capture** option, you establish the beginning of the capture as your zero of time.

The **Sequence number origin** section will allow you to choose whether your actual TCP sequence numbers or the relative TCP sequence numbers (the TCP sequence numbers minus your initial TCP sequence number) are shown on the graph. It is frequently convenient to use the relative sequence number because it gives you some notion of how much data has been transmitted. If you select the **initial sequence number** option, the relative TCP sequence numbers will be used. If you select the **0 (=absolute)** option, the actual TCP sequence numbers will be used in the graph.

The **Cross** tab, shown in Figure 5.63, allows you to control whether crosshairs follow the mouse pointer in the graph window.

Figure 5.63 Graph Control Dialog Box: Cross Tab

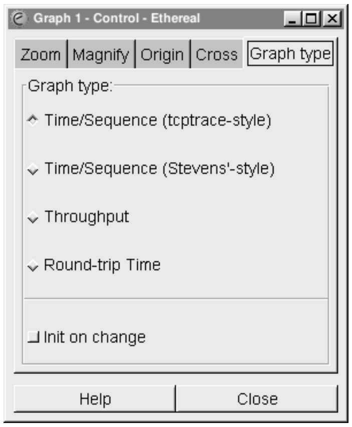


Crosshairs: ● off ○ on

If you select the **off** radio button, there will be no crosshairs following the mouse pointer in the graph window. If you select the **on** option, there will be crosshairs following the mouse pointer in the graph window.

Once you have the graph window displayed, you can use the **Graph type** tab, shown in Figure 5.64, to change which type of graph is being displayed.

Figure 5.64 Graph Control Dialog Box: Graph Type Tab



Graph type:

● Time/Sequence (tcptrace-style)

○ Time/Sequence (Stevens'-style)

○ Throughput

○ Round-trip Time

□ Init on change

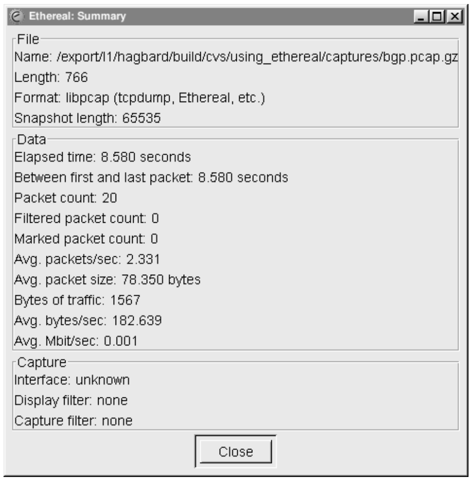
If you select the **Time/Sequence (tcptrace-style)** option, the time-sequence (tcpgraph) window will be displayed. If you select the **Time/Sequence (Stevens'-style)** option the time-sequence (Stevens'-style) window will be displayed. If you select the **Throughput** option, the throughput graph window will be displayed. If you select the **Round-trip Time** option, the RTT graph window will be displayed.

By default, if you have applied a zoom to the graph window for one graph type it will persist if you change graph types. If you enable the **Init on change** checkbox, each time you change graph types the zoom will be reset.

Summary

The Summary dialog box, shown in Figure 4.65, can be displayed by selecting **Analyze | Summary** from the menu bar.

Figure 4.65 Summary Dialog Box



File

Name: /export/l1/hagbard/build/cvs/using_ethereal/captures/bgp.pcap.gz

Length: 766

Format: libpcap (tcpdump, Ethereal, etc.)

Snapshot length: 65535

Data

Elapsed time: 8.580 seconds

Between first and last packet: 8.580 seconds

Packet count: 20

Filtered packet count: 0

Marked packet count: 0

Avg. packets/sec: 2.331

Avg. packet size: 78.350 bytes

Bytes of traffic: 1567

Avg. bytes/sec: 182.639

Avg. Mbit/sec: 0.001

Capture

Interface: unknown

Display filter: none

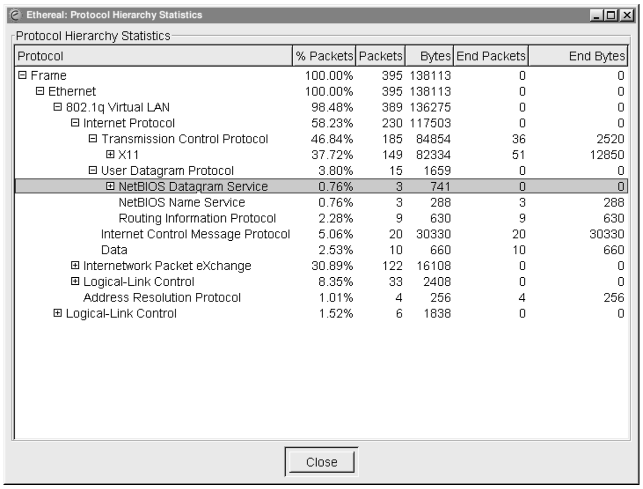
Capture filter: none

This dialog box provides information about the capture file, basic statistics about the capture data, and basic information about the capture.

Protocol Hierarchy Statistics

The Protocol Hierarchy Statistics dialog box, shown in Figure 4.66, can be displayed by selecting **Analyze | Protocol Hierarchy Statistics** from the menu bar.

Figure 4.66 Protocol Hierarchy Statistics Dialog Box


Ethereal: Protocol Hierarchy Statistics
Protocol Hierarchy Statistics
Close

```
Protocol                                 % Packets Packets  Bytes End Packets  End Bytes
⊟ Frame                                    100.00%     395 138113           0          0
   ⊟ Ethernet                              100.00%     395 138113           0          0
      ⊟ 802.1q Virtual LAN                  98.48%     389 136275           0          0
         ⊟ Internet Protocol                58.23%     230 117503           0          0
            ⊟ Transmission Control Protocol 46.84%     185  84854          36       2520
               ⊞ X11                        37.72%     149  82334          51      12850
            ⊟ User Datagram Protocol         3.80%      15   1659           0          0
               ⊞ NetBIOS Datagram Service    0.76%       3    741           0          0
                 NetBIOS Name Service        0.76%       3    288           3        288
                 Routing Information Protocol 2.28%      9    630           9        630
              Internet Control Message Protocol 5.06%   20  30330          20      30330
              Data                           2.53%      10    660          10        660
         ⊞ Internetwork Packet eXchange     30.89%     122  16108           0          0
         ⊞ Logical-Link Control              8.35%      33   2408           0          0
           Address Resolution Protocol       1.01%       4    256           4        256
      ⊞ Logical-Link Control                 1.52%       6   1838           0          0
```

This dialog box provides a tree representation of protocols and statistics associated with them. Table 4.13 provides a description of what the columns mean:

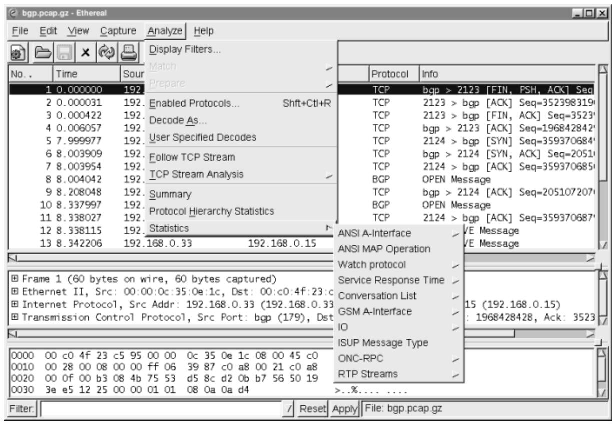
Table 4.13 Protocol Hierarchy Statistics Columns

Column	Description
Protocol	The protocol on which statistics are being reported. The protocol may have sub-items on the tree representing the protocols it contains. For example, IP contains TCP and UDP.
% Packets	Percentage of all packets in the capture that are of this protocol.
Packets	The number of packets in the capture that are of the protocol.
Bytes	The number of bytes in this capture containing this protocol.
End Packets	The number of packets for which this protocol is the last protocol in the decode. For example, a TCP SYN packet containing no data would be an end packet for TCP and counted in TCP's end packets count.
End Bytes	The number of bytes for which this protocol is the last protocol in the decode.

Statistics Submenu

The **Statistics** submenu, shown in Figure 4.67 provides a variety of specialized tools to analyze network traffic. These statistics are reported for certain protocol features.

Figure 4.67 Statistics Submenu



```
No. .  Time      Sour                                Protocol  Info
    1 0.000000   192.                                TCP       bgp > 2123 [FIN, PSH, ACK] Seq
    2 0.000031   192.                                TCP       2123 > bgp [ACK] Seq=352398319
    3 0.000422   192.                                TCP       2123 > bgp [FIN, ACK] Seq=3523
    4 0.006057   192.                                TCP       bgp > 2123 [ACK] Seq=196842842
    5 7.999977   192.                                TCP       2124 > bgp [SYN] Seq=359370684
    6 8.003909   192.                                TCP       bgp > 2124 [SYN, ACK] Seq=2051
    7 8.003954   192.                                TCP       2124 > bgp [ACK] Seq=359370685
    8 8.004042   192.                                BGP       OPEN Message
    9 8.208048   192.                                TCP       bgp > 2124 [ACK] Seq=205107207
   10 8.337997   192.                                BGP       OPEN Message
   11 8.338027   192.                                TCP       2124 > bgp [ACK] Seq=359370687
   12 8.338115   192.                                          VE Message
   13 8.342206   192.168.0.33      192.168.0.15                VE Message
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33                   15 (192.168.0.15)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst                   : 1968428428, Ack: 3523
```

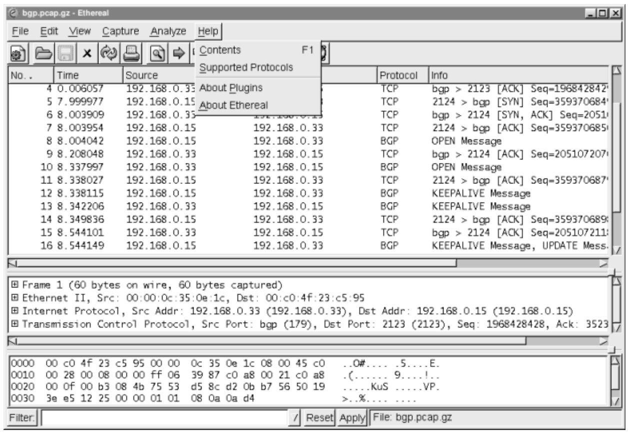
```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

However, the tools in the **Statistics** submenu are quite specialized and beyond the scope of this book.

Help

The **Help** menu is shown in Figure 4.68, and the **Help** options are explained in Table 4.14.

Figure 4.68 Help Menu



```
No. .  Time      Source            Protocol  Info
    4 0.006057   192.168.0.33                         TCP       bgp > 2123 [ACK] Seq=196842842
    5 7.999977   192.168.0.15                         TCP       2124 > bgp [SYN] Seq=359370684
    6 8.003909   192.168.0.33                         TCP       bgp > 2124 [SYN, ACK] Seq=2051
    7 8.003954   192.168.0.15      192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370685
    8 8.004042   192.168.0.15      192.168.0.33       BGP       OPEN Message
    9 8.208048   192.168.0.33      192.168.0.15       TCP       bgp > 2124 [ACK] Seq=205107207
   10 8.337997   192.168.0.33      192.168.0.15       BGP       OPEN Message
   11 8.338027   192.168.0.15      192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370687
   12 8.338115   192.168.0.15      192.168.0.33       BGP       KEEPALIVE Message
   13 8.342206   192.168.0.33      192.168.0.15       BGP       KEEPALIVE Message
   14 8.349836   192.168.0.15      192.168.0.33       TCP       2124 > bgp [ACK] Seq=359370689
   15 8.544101   192.168.0.33      192.168.0.15       TCP       bgp > 2124 [ACK] Seq=205107211
   16 8.544149   192.168.0.15      192.168.0.33       BGP       KEEPALIVE Message, UPDATE Mess
```

```
⊞ Frame 1 (60 bytes on wire, 60 bytes captured)
⊞ Ethernet II, Src: 00:00:0c:35:0e:1c, Dst: 00:c0:4f:23:c5:95
⊞ Internet Protocol, Src Addr: 192.168.0.33 (192.168.0.33), Dst Addr: 192.168.0.15 (192.168.0.15)
⊞ Transmission Control Protocol, Src Port: bgp (179), Dst Port: 2123 (2123), Seq: 1968428428, Ack: 3523
```

```
0000  00 c0 4f 23 c5 95 00 00  0c 35 0e 1c 08 00 45 c0   ..O#.... .5....E.
0010  00 28 00 08 00 00 ff 06  39 87 c0 a8 00 21 c0 a8   .(...... 9....!..
0020  00 0f 00 b3 08 4b 75 53  d5 8c d2 0b b7 56 50 19   .....KuS .....VP.
0030  3e e5 12 25 00 00 01 01  08 0a 0a d4               >..%.... ....
```

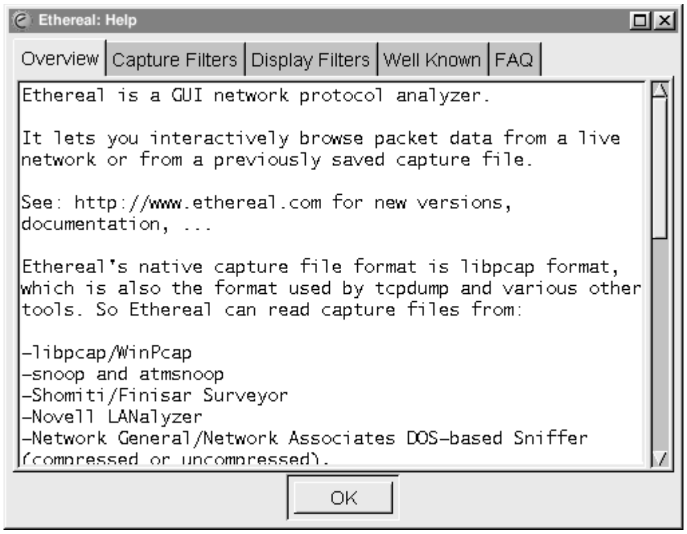
Table 4.14 Help Menu Options

Menu Option	Description
Contents	Displays the Contents for the Ethereal online help.
Supported Protocols	Displays a list of the supported protocols and the display filter fields they provide.
About Plugins	Displays a list of plugins being used and their versions.
About Ethereal	Displays information about Ethereal version and compile information.

Contents

The Contents dialog box, shown in Figure 4.69, can be displayed by selecting **Help | Contents** from the menu bar.

Figure 4.69 Help Contents Dialog Box



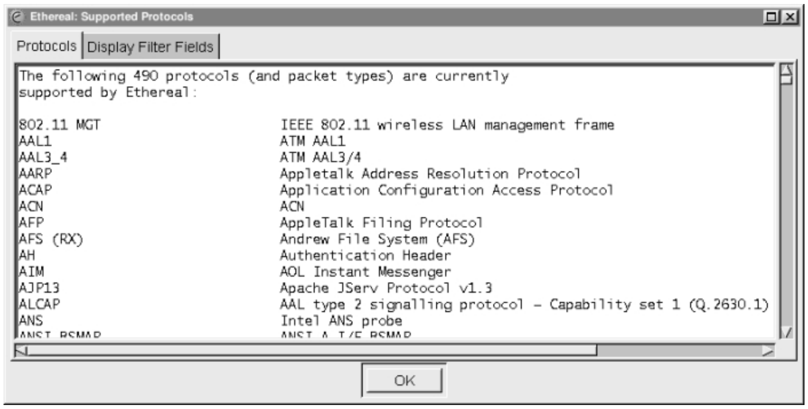
```
Ethereal is a GUI network protocol analyzer.

It lets you interactively browse packet data from a live
network or from a previously saved capture file.

See: http://www.ethereal.com for new versions,
documentation, ...

Ethereal's native capture file format is libpcap format,
which is also the format used by tcpdump and various other
tools. So Ethereal can read capture files from:

-libpcap/WinPcap
-snoop and atmsnoop
-Shomiti/Finisar Surveyor
-Novell LANalyzer
-Network General/Network Associates DOS-based Sniffer
(compressed or uncompressed).
```

This dialog box provides tabs giving an overview of Ethereal, information about Capture Filters, information about Display Filters, some well-known information about networking, and answers to Frequently Asked Questions (FAQs).

Supported Protocols

The Supported Protocols dialog box, shown in Figure 4.70, can be displayed by selecting **Help | Supported Protocols** from the menu bar.

Figure 4.70 Supported Protocols Dialog Box



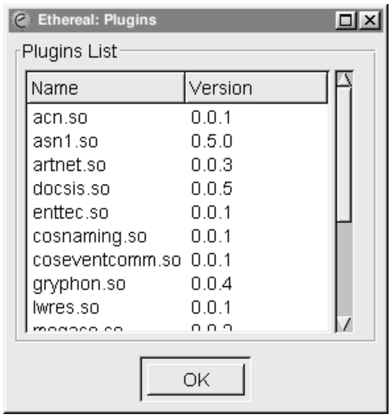
```
The following 490 protocols (and packet types) are currently
supported by Ethereal:

802.11 MGT                      IEEE 802.11 wireless LAN management frame
AAL1                            ATM AAL1
AAL3_4                          ATM AAL3/4
AARP                            Appletalk Address Resolution Protocol
ACAP                            Application Configuration Access Protocol
ACN                             ACN
AFP                             AppleTalk Filing Protocol
AFS (RX)                        Andrew File System (AFS)
AH                              Authentication Header
AIM                             AOL Instant Messenger
AJP13                           Apache JServ Protocol v1.3
ALCAP                           AAL type 2 signalling protocol - Capability set 1 (Q.2630.1)
ANS                             Intel ANS probe
ANSI BSMAP                      ANSI A I/F BSMAP
```

This dialog box provides a list of the protocols supported by the current version of Ethereal and a list of the display filter fields provided in the current version of Ethereal.

About Plugins

The About Plugins dialog box, shown in Figure 4.71, can be displayed by selecting **Help | About Plugins** from the menu bar.

Figure 4.71 About Plugins Dialog Box



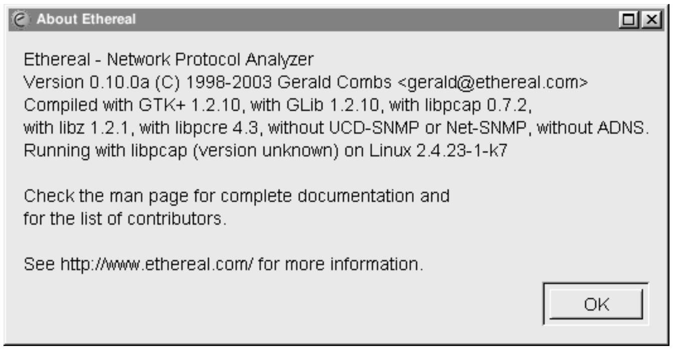
Name	Version
acn.so	0.0.1
asn1.so	0.5.0
artnet.so	0.0.3
docsis.so	0.0.5
enttec.so	0.0.1
cosnaming.so	0.0.1
coseventcomm.so	0.0.1
gryphon.so	0.0.4
lwres.so	0.0.1
megaco.so	0.0.2

This dialog box provides a list of the plugins currently loaded into Ethereal and their versions. Ethereal will load all plugins available to it at start time, but there is no GUI that allows you to load or unload a plugin.

About Ethereal

The About Ethereal dialog box, as shown in Figure 4.72, can be displayed by selecting **Help | About Plugins** from the menu bar.

Figure 4.72 About Ethereal Dialog Box



Ethereal - Network Protocol Analyzer
Version 0.10.0a (C) 1998-2003 Gerald Combs <gerald@ethereal.com>
Compiled with GTK+ 1.2.10, with GLib 1.2.10, with libpcap 0.7.2,
with libz 1.2.1, with libpcre 4.3, without UCD-SNMP or Net-SNMP, without ADNS.
Running with libpcap (version unknown) on Linux 2.4.23-1-k7

Check the man page for complete documentation and
for the list of contributors.

See http://www.ethereal.com/ for more information.

This dialog box contains information about the version of Ethereal you are running and which options it was compiled with. This information is important to know if you are ever reporting a bug to the Ethereal developers.

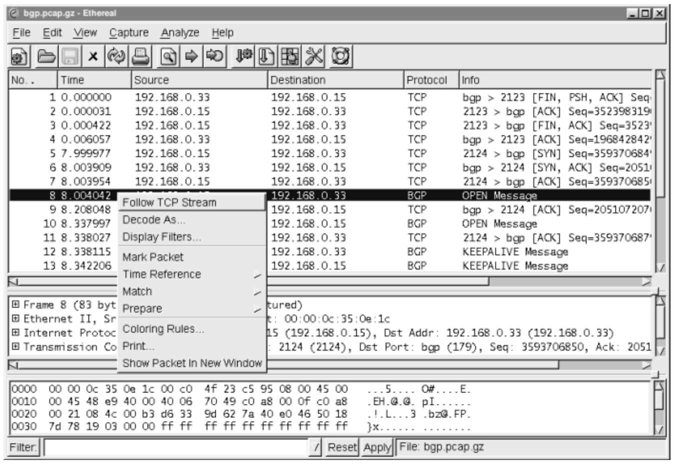
Pop-up Menus

Ethereal has context-sensitive pop-up menus to assist you in performing tasks. None of these menus actually provide any additional functionality beyond what is available through the menu bar, but they are easier and quicker to use in some circumstances.

Summary Window Pop-up Menu

The Summary Window pop-up menu, shown in Figure 4.73, can be displayed by right-clicking on the Summary Window.

Figure 4.73 Summary Window Pop-up Menu



```
No. .  Time        Source          Destination     Protocol  Info
    1 0.000000     192.168.0.33    192.168.0.15    TCP       bgp > 2123 [FIN, PSH, ACK] Seq
    2 0.000031     192.168.0.15    192.168.0.33    TCP       2123 > bgp [ACK] Seq=352398319
    3 0.000422     192.168.0.15    192.168.0.33    TCP       2123 > bgp [FIN, ACK] Seq=3523
    4 0.006057     192.168.0.33    192.168.0.15    TCP       bgp > 2123 [ACK] Seq=196842842
    5 7.999977     192.168.0.15    192.168.0.33    TCP       2124 > bgp [SYN] Seq=359370684
    6 8.003909     192.168.0.33    192.168.0.15    TCP       bgp > 2124 [SYN, ACK] Seq=2051
    7 8.003954     192.168.0.15    192.168.0.33    TCP       2124 > bgp [ACK] Seq=359370685
    8 8.004042                     192.168.0.33    BGP       OPEN Message
    9 8.208048                     192.168.0.15    TCP       bgp > 2124 [ACK] Seq=205107207
   10 8.337997                     192.168.0.15    BGP       OPEN Message
   11 8.338027                     192.168.0.33    TCP       2124 > bgp [ACK] Seq=359370687
   12 8.338115                     192.168.0.33    BGP       KEEPALIVE Message
   13 8.342206                     192.168.0.15    BGP       KEEPALIVE Message

⊞ Frame 8 (83 byt                  tured)
⊞ Ethernet II, Sr                  t: 00:00:0c:35:0e:1c
⊞ Internet Protoc                  15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Co                  : 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051

0000  00 00 0c 35 0e 1c 00 c0  4f 23 c5 95 08 00 45 00   ...5.... O#....E.
0010  00 45 48 e9 40 00 40 06  70 49 c0 a8 00 0f c0 a8   .EH.@.@. pI......
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L...3 .bz@.FP.
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x...... ........
```

The Summary Window pop-up menu provides functionality that has been covered earlier in the chapter. Table 4.15 indicates where to find more information in this chapter on the Summary Window pop-up menu options.

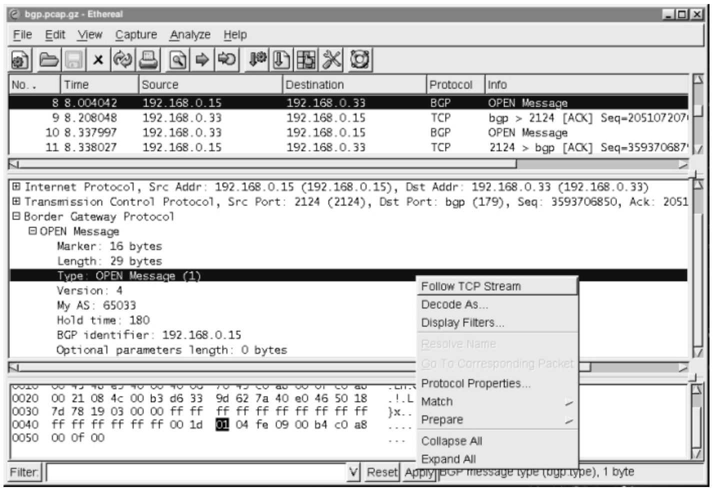
Table 4.15 Summary Window Pop-up Menu References

Menu Option	Reference
Follow TCP Stream	See ”Analyze: Contents of TCP Stream
Decode As…	See ”Analyze: Decode As”
Displayer Filters…	See “Analyze: Display Filters
Mark Packet	See “Edit: Mark Packet”
Time Reference	See “Edit: Time Reference”
Match	See “Analyze: Match”
Prepare	See “Analyze: Prepare”
Coloring Rules…	See “Analyze: Coloring Rules”
Print…	See “File: Print”
Show Packet in New Window	See “View: Show Packet in New Window”

Protocol Tree Window Pop-up Menu

The Protocol Tree pop-up menu, shown in Figure 4.74, can be displayed by right-clicking on the Protocol Tree Window.

Figure 4.74 Protocol Tree Window Pop-up Menu



```
No. .  Time       Source         Destination    Protocol  Info
     8 8.004042   192.168.0.15   192.168.0.33   BGP       OPEN Message
     9 8.208048   192.168.0.33   192.168.0.15   TCP       bgp > 2124 [ACK] Seq=205107207
    10 8.337997   192.168.0.33   192.168.0.15   BGP       OPEN Message
    11 8.338027   192.168.0.15   192.168.0.33   TCP       2124 > bgp [ACK] Seq=359370687
```

```
⊞ Internet Protocol, Src Addr: 192.168.0.15 (192.168.0.15), Dst Addr: 192.168.0.33 (192.168.0.33)
⊞ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051
⊟ Border Gateway Protocol
  ⊟ OPEN Message
      Marker: 16 bytes
      Length: 29 bytes
      Type: OPEN Message (1)
      Version: 4
      My AS: 65033
      Hold time: 180
      BGP identifier: 192.168.0.15
      Optional parameters length: 0 bytes
```

```
0020  00 21 08 4c 00 b3 d6 33  9d 62 7a 40 e0 46 50 18   .!.L
0030  7d 78 19 03 00 00 ff ff  ff ff ff ff ff ff ff ff   }x..
0040  ff ff ff ff ff ff 00 1d  01 04 fe 09 00 b4 c0 a8   ....
0050  00 0f 00                                           ...
```

BGP message type (bgp.type), 1 byte

The Protocol Tree Window pop-up menu provides functionality that has been covered earlier in the chapter. Table 4.16 includes descriptions for some items and indicates where to find more information in this chapter for other items.

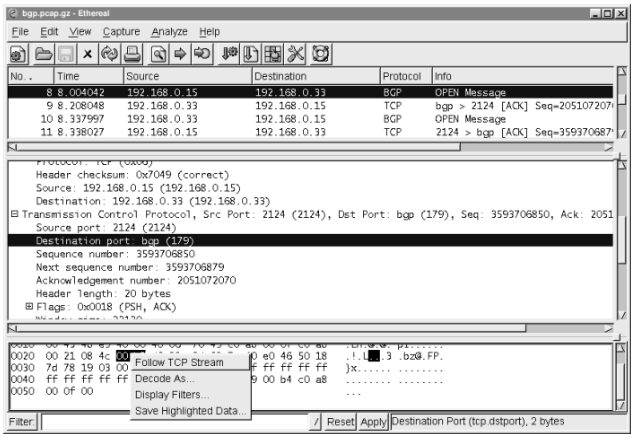
Table 4.16 Protocol Tree Window Pop-up Menu References/Descriptions

Menu Option	Reference/Description
Follow TCP Stream	See “Analyze: Follow TCP Stream”
Decode As…	See “Analyze: Decode As”
Displayer Filters…	See “Analyze: Display Filters”
Resolve Name	Forces resolution of all names for this packet. See the Ethereal Name Resolution sidebar for more information about Ethereal name resolution. Note that this option is only available if all name resolution is disabled.
Go To Corresponding Packet	See “Edit: Go To Corresponding Packet”
Protocol Properties…	See “Edit: Preferences”
Match	See “Analyze: Match”
Prepare	See “Analyze: Prepare”
Collapse All	See “View: Collapse All”
Expand All	See “View: Expand All”

Data View Window Pop-up Menu

The Data View Window pop-up menu, shown in Figure 4.75, can be displayed by right-clicking in the Data View Window.

Figure 4.75 Data View Window Pop-up Menu



No. .	Time	Source	Destination	Protocol	Info
8	8.004042	192.168.0.15	192.168.0.33	BGP	OPEN Message
9	8.208048	192.168.0.33	192.168.0.15	TCP	bgp > 2124 [ACK] Seq=205107207
10	8.337997	192.168.0.33	192.168.0.15	BGP	OPEN Message
11	8.338027	192.168.0.15	192.168.0.33	TCP	2124 > bgp [ACK] Seq=359370687

```
    Header checksum: 0x7049 (correct)
    Source: 192.168.0.15 (192.168.0.15)
    Destination: 192.168.0.33 (192.168.0.33)
⊟ Transmission Control Protocol, Src Port: 2124 (2124), Dst Port: bgp (179), Seq: 3593706850, Ack: 2051
    Source port: 2124 (2124)
    Destination port: bgp (179)
    Sequence number: 3593706850
    Next sequence number: 3593706879
    Acknowledgement number: 2051072070
    Header length: 20 bytes
  ⊞ Flags: 0x0018 (PSH, ACK)
```

```
0020  00 21 08 4c 00                       0 e0 46 50 18   .!.L  .3 .bz@.FP.
0030  7d 78 19 03 00                       f ff ff ff ff   }x...... ........
0040  ff ff ff ff ff                       9 00 b4 c0 a8   ........ ........
0050  00 0f 00                                             ...
```

Destination Port (tcp.dstport), 2 bytes

The Data View Window pop-up menu provides functionality that has been covered earlier in this chapter. Table 4.17 indicates where to find more information in this chapter on the Data View Window pop-up menu options.

Table 4.17 Data View Window Pop-up Menu References

Menu Option	Reference
Follow TCP Stream	See "Analyze: Follow TCP Stream"
Decode As…	See "Analyze: Decode As"
Displayer Filters…	See "Analyze: Display Filters"
Save Highlighted Data…	See "File: Export"

Using Command Line Options

Ethereal supports a large number of command line options to control its behavior when first run. This section will document some of the most commonly used options. Several of the other options are used to manipulate certain aspects of the GUI (font, height of Summary Window, Protocol Tree Window, Data Window, etc.) or set elements like link type or automatic scrolling. However, these options are not as common and will not be covered in this section.

Capture and File Options

The most commonly used Ethereal options are those related to captures and files. Table 4.18 lists some of the most common command line options related to these tasks.

Table 4.18 Capture and File Command Line Options

Command Line Option	Description
-i <interface>	Set the name of the interface used for live captures to <inteface>.
-k –i option.	Start capture immediately. This requires the
-a <test>:<value>	Sets an autostop condition for the capture. <test> may be one of duration or filesize. If the <test> is duration then <value> must be the number of seconds the capture should run before it stops. If <test> is filesize then <value> is the number of kilobytes that should be captured before the capture stops.
-c <count>	Sets the number of packets to read before stopping the capture. After <count> packets have been read the capture will stop.
-r <filename>	Read the capture saved in <filename>.
-w <filename>	Write the capture to <filename>.
-b <count>	Enable the use of <count> files in a ring buffer for captures. A maximum capture size must be specified with the –a filesize:<value> option.

To start capturing immediately on interface **eth0** and write the results to a ring buffer with **3** files of maximum size **100** kilobytes with base filename **foo.bar.libpcap** you would execute the following at the command line:

```
ethereal -i eth0 –k –w foo.bar.libpcap –b 3 –a filesize:100
```

Filter Options

Ethereal will allow you to specify filter information from the command line as well. Table 4.19 lists some of the most commonly used filter related command line options.

Table 4.19 Filter Command Line Options

Command Line Option	Description
-f <capture filter >	Set the tcpdump style capture filter string to <filter string>.
-R <display filter>	Only applicable when reading a capture from a file with the –r option. Applies the display filter <display filter> to all packets in the capture file and discard those that do not match.

To extract all packets from a capture file bgp.pcap.gz with **bgp.type == 2**, you would execute the following at the command line:

```
ethereal –r bgp.pcap.gz –R "bgp.type == 2"
```

Other Options

Other commonly used options are shown in Table 4.20.

Table 4.20 Other Command Line Options

Command Line Option	Description
-N <flags>	Turns on name resolution. Depending on which letters follow –N, various names will be resolved by Ethereal. **n** will cause network name resolution to be turned on, **t** will enable transport name resolution, **m** will enable MAC address resolution, and **C** will enable asynchronous DNS lookups for network name resolution.
-v	Print the Ethereal version information.
-h	Print Ethereal's help information.

Summary

In this chapter you have learned the major components of the Ethereal GUI.You have also learned the major functionality of the Ethereal application and how to access it.

You should now be able to perform network captures, open saved network captures, and print captures.You are also equipped to use display filters to filter the packets displayed in the Summary Window, color the packets in the Summary Window for easier readability, or find a packet in the capture with particular characteristics.

We've shown how to navigate the protocol tree in the Protocol Tree Window to drill down into the contents of a packet, and use the protocol tree fields to prepare new display filter strings.

Finally, you now know how to force a packet or group of packets to be decoded by a particular dissector.You have learned how to enable and disable decoding of particular protocols, and should have some understanding of how to use some of the more commonly used tools in Ethereal to gain better visibility into TCP Streams.

Solutions Fast Track

Getting started with Ethereal

☑ Binary Ethereal packages for Windows, Linux, and various UNIX flavors can be downloaded from www.ethereal.com.

☑ Source code can be downloaded and compiled from www.ethereal.com if the binary packages available don't meet your needs.

☑ Ethereal can be launched by typing **ethereal** at the command line.

Exploring the Main Windows

☑ The Summary Window provides a one-line summary for each packet.

☑ The Protocol Tree Window provides a detailed decode of the packet selected in the Summary Window.

☑ The Data View Window provides the hexadecimal (or hex) dump of the packets' actual bytes.

Other Window Components

☑ The filter bar provides a quick mechanism for filtering the packets displayed in the Summary Window.

☑ Clicking the filter bar's **Filter:** button will display the Display Filter dialog box to help you construct a display filter string.

☑ The **Information** field will show the display filter field name of the field selected in the Protocol Tree Window.

Exploring the Menus

☑ Most preferences can be set in the Preferences dialog box.

☑ There are context-sensitive pop-up menus available by right-clicking on the Summary Window, Protocol Tree Window, or Data View Window.

☑ Packets in the Summary Window can be color-coded for easy reading by using the Apply Color Filters dialog box.

Using Command Line Options

☑ Ethereal can apply display filters to packets read from a file with the **–R** flag, discarding packets that don't match the filter.

☑ Ethereal uses **–r** to indicate a file to read from and **–w** to indicate a file to write to.

☑ Ethereal can be made to start capturing from an interface immediately on startup by using the **–i** and **-k** options.

Frequently Asked Questions

The following Frequently Asked Questions, answered by the authors of this book, are designed to both measure your understanding of the concepts presented in this chapter and to assist you with real-life implementation of these concepts. To have your questions about this chapter answered by the author, browse to **www.syngress.com/solutions** and click on the **"Ask the Author"** form. You will also gain access to thousands of other FAQs at ITFAQnet.com.

Q: Why is Ethereal so slow displaying data during capture? It seems to lock up.

A: Your version of Ethereal may have been compiled without the ADNS (Asynchronous DNS) library. If so, Ethereal is stopping to do a DNS lookup for the source and destination IP address in each packet it decodes. It can take a long time for DNS queries to time out if they fail, and during this time Ethereal may lock up while waiting for those failures. To solve this problem, get a version of Ethereal with ADNS compiled in. To work around this problem unselect **Enable Network Name Resolution** in the Capture Options dialog box when starting a capture or in the File dialog box when opening a capture file.

Q: Why is it when I select some fields in the Protocol Tree Window I don't see the field name in the Information field? How can I filter on the field if I can't find out its name?

A: Ethereal has been developed over many years by a team of volunteer programmers. Many different people have written the dissectors, which decode the protocols in Ethereal, at many different times. Not all dissector authors associated a filterable field with each field they display in the Protocol Tree. You will not be able to filter on such fields. If such filtering is important to you for a particular protocol, you are encouraged to alter the source code for that dissector to include the capacity and submit it to the Ethereal team for inclusion.

Q: Why do I sometimes see an IP address or a TCP/UDP port number or a MAC address twice, once in parenthesis and once not?

A: When name resolution is turned off for an address type, or when no name is found for a given address, Ethereal will insert the actual address into the place where the name would have gone. As a result, a place where you would have seen the name with the address in parentheses (or vice versa) will just show two copies of the address. Don't worry about it ☺

Q: I need more complicated capture filtering than tcpdump-style capture filters provide; can I use Ethereal's display filters to restrict what I capture?

A: The short answer is no. Ethereal will not allow you to use display filters to filter on capture. However, there is a sort of workaround to achieve this. While Ethereal will not allow you to use display filters on capture, Tethereal will. To capture from an interface <interface> to a file <savefile> filtering with a display filter string <filter string> you would execute at the command line:

```
tethereal -i <interface> -w <savefile> -R <filter string>
```

Tethereal will capture from <interface> and only save to <savefile> those packets that match <filter string>. In many cases display filter strings will not be nearly as fast as the tcpdump-style capture filters, but if only display filters will do, this hack will let you use them.

Q: Does Ethereal really capture all the traffic arriving at an interface when capturing in promiscuous mode?

A: That depends. Ethereal gets whatever is captured by libpcap. Sometimes due to high load on the system you are capturing from, or just due to trying to capture from too high bandwidth an interface, packets may be lost for a number of reasons, including being dropped by the kernel. Keep this in mind as you work.

Q: Why am I seeing packets that aren't addressed to or being sent by my local interface even though I've turned off capturing in promiscuous mode?

A: There may be other applications running, like Snort, on the system you are capturing from that have put the interface into promiscuous mode. Whether Ethereal puts the interface in promiscuous mode, or some other application does, if the interface is in promiscuous mode you will see all traffic that arrives at it, not just the traffic addressed to or sent from the interface.