Ethereal Quick-Start Guide

Instructions on Using the Ethereal Packet Analyzer

June 1, 2006

Table of Contents

Chapter 1: Getting Started

Ethereal is a network packet analyzer. It lets you examine the network traffic flowing into and out of your Windows or Unix machine. Network professionals can use Ethereal to troubleshoot networking problems, but we will use it to see the data that your system sends and receives when you type a web address into a web browser (e.g., Internet Explorer or Mozilla's Firefox).

You can find more information on the Ethereal web site at **www.Ethereal.com** **.**

As a metaphor for Ethereal's operation, pretend you could take a special magnifying glass and look into the network cable coming out of the back of your personal computer. You would see the bits of information, encoded as electrical pulses, flowing into and out of your computer.

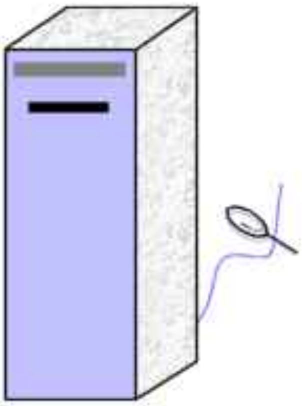


Figure 1: Ethereal lets you see the network traffic entering and leaving your computer.

Ethereal may not work on Windows computers using wireless network adapters. Try switching off Promiscuous mode (Edit / Preferences / Capture). For more discussion of what Ethereal can or can not capture, refer to Appendix 1.

If Ethereal stopped there, it would only be of limited use – it is difficult to make sense out of a raw stream of data.

However, Ethereal also contains a protocol analyzer that understands over 750 protocols. It converts the data stream to a listing of packets flowing in and out of the computer. It allows you to examine an individual packet, and drill down through the layers of encapsulation until the application-level payload is revealed.

Ethereal is developed as open source software. This means that the software is developed as a community effort, and the source code is freely available. Furthermore, it is licensed under the GNU General Public License (http://www.gnu.org/licenses/gpl.html). This license gives you the right to use the software for free. However, you may not sell the software, or a derivative of it, and if you modify the program code, you must be willing to submit the changes back to the open source community.

Refer to Appendix 1 for a discussion of the type of packets that Ethereal captures. This discussion also explains how your particular network configuration may affect the type of packets you see.

I) Current Version

This documentation is based on Ethereal version 0.10.14, installed on a Windows XP platform. Although you may find a newer release available when you download the software, the concepts in this manual should still be relevant.

The Ethereal web site is a rich source of help for both beginners and experts. Although this QuickStart guide recommends specific items on the web site, the reader is asked to use the Ethereal menu system to locate the referenced items. The Ethereal menu system will remain current as changes are made to the web site.

According to the Ethereal web site: "Ethereal is still technically beta software, but it has a comprehensive feature set and is suitable for production use." In fact, Ethereal was first released in 1998, and has been in continuous development since that time. Furthermore, it should be emphasized, that the current release is not a .1 release– it is actually release 0.10.xx.

Ethereal is supported in Unix (including Mac OSX), Linux, and Windows (Me/98/Server 2003 / XP / 2000 / NT 4.0). For a more detailed list of supported environments, see the documentation at the Ethereal web site.

More detailed documentation can be found on the Ethereal web site at: www.Ethereal.com .

II) Installation

The Ethereal installation package will also install WinPcap unless you override the settings. Ethereal will not work unless WinPcap is also installed.

Ethereal can be downloaded directly from the Ethereal web site at www.Ethereal.com . The download is an exe file of approximately 12MB. Save the file to an appropriate location, such as your desktop. When the file is downloaded, double click on it to start the installation process. The default installation settings should work fine. One option that is pre-selected is "WinPcap". This is a required component of Ethereal, and it must be installed for Ethereal to work properly.

III) Setting up Ethereal for First Time Use

When you first start Ethereal you must tell which network adapter to use. Go to the Edit menu and choose "Preferences".

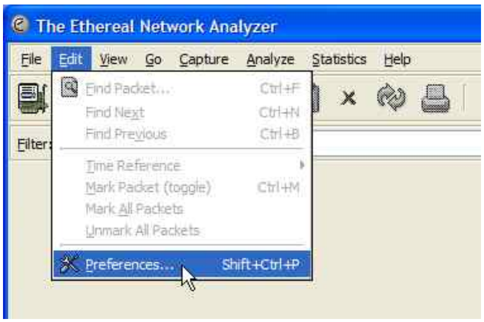
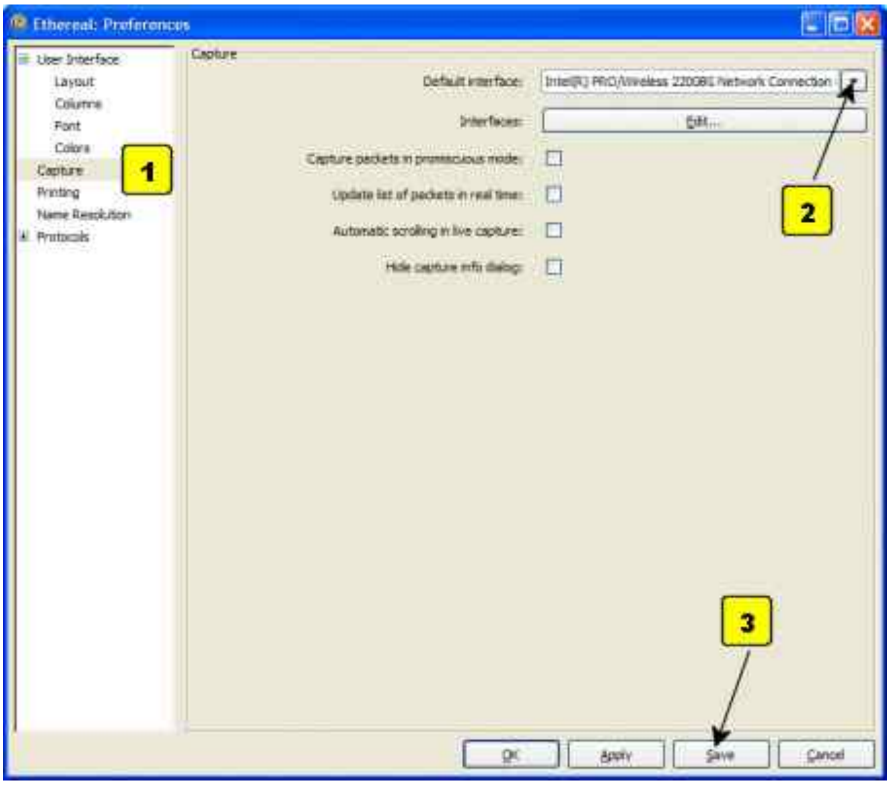


Figure 2: Choose Preferences from the Edit Menu

When the preferences screen appears you must 1) Click on the "Capture" menu; 2) Click on the down arrow and select the correct network card, and 3) Click on the "Save" button.



Capture

Default interface: Intel(R) PRO/Wireless 2200BG Network Connection

Interfaces:

Capture packets in promiscuous mode: □

Update list of packets in real time: □

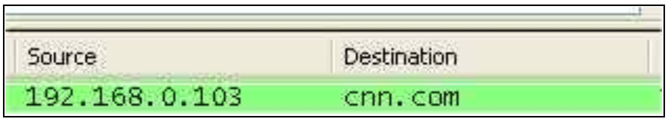
Automatic scrolling in live capture: □

Hide capture info dialog: □

Figure 3: Preferences Dialog

Note: The save button may be hidden. On many displays, the dialog box runs off the bottom of the screen. If you can not see the save button, click on the blue bar at the top of the window and drag the box upward.

There is one more task to complete before you close the Preferences Dialog box. You will probably want to turn the name resolution feature on. First, let me explain what this feature does. Examine Figure 4 below. This figure highlights the source and destination of an individual packet. The source is the IP address of the local client, and the destination is shown as cnn.com.

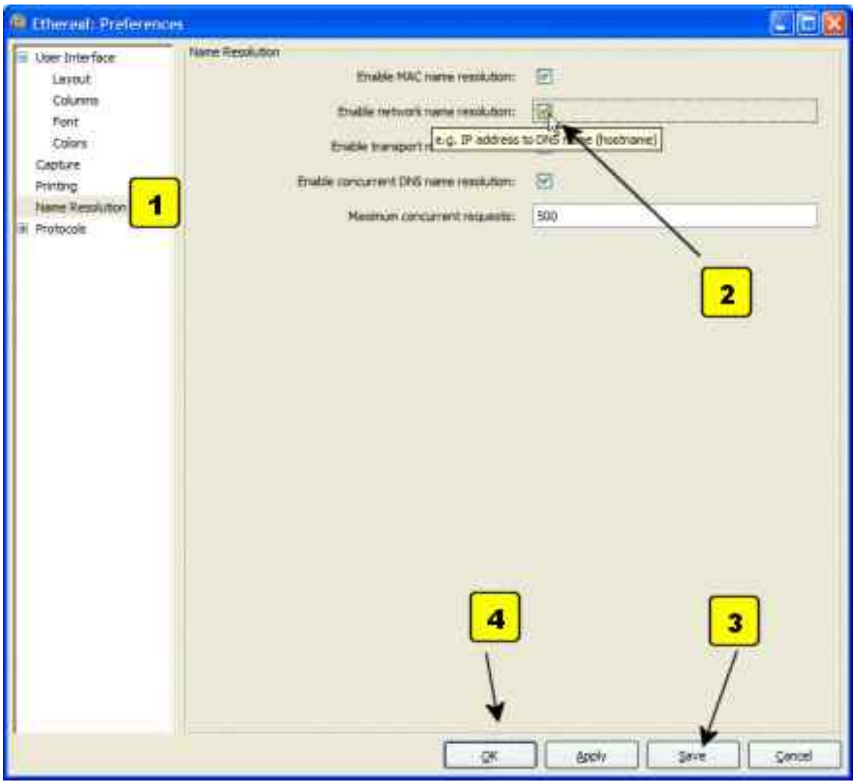


Source	Destination
192.168.0.103	cnn.com

Figure 4: Packet Listing Summary

Network Name Resolution (NNR) tells Ethereal to use names, such as cnn.com, in the summaries. If NNR is turned o ff, you will only see IP addresses in the summary. This setting only affects the summary. Even with names turned on, you can easily see the IP address by clicking on the packet and examining the packet details. However, it is easier to select packets if the names are available to identify network servers.

In order to turn on NNR, you will need to use the Preferences dialog again. If you have closed it, return to the Edit menu and chose Preferences to reopen the dialog. Then, 1) Click on “Name Resolution”; 2) Check the box next to “Enable network name resolution”; 3) Click on “Save”; and 4) Click on “OK” to close the dialog box.



Name Resolution

Enable MAC name resolution: ☑

Enable network name resolution: ☑

e.g. IP address to DNS name (hostname)

Enable transport n

Enable concurrent DNS name resolution: ☑

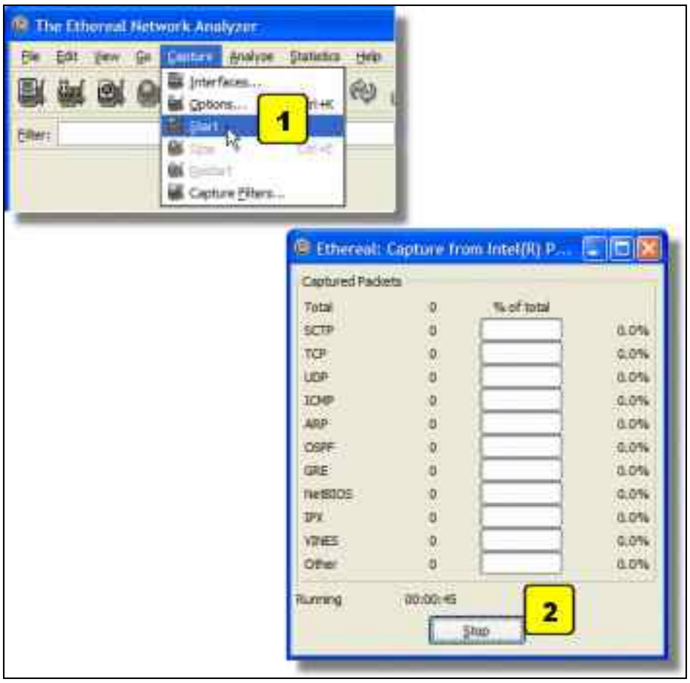
Maximum concurrent requests: 500

Figure 5: Name Resolution

Chapter 2: Using Ethereal

I) Capturing Packets

Now that you have completed setting up Ethereal, you are ready to capture packets coming to and from your machine. 1) Begin the capture process by selecting "Start" from the Ethereal "Capture" menu as shown in Figure 6, below. Do not stop the process yet.



Captured Packets

Total	0	% of total
SCTP	0	0.0%
TCP	0	0.0%
UDP	0	0.0%
ICMP	0	0.0%
ARP	0	0.0%
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 00:00:45

Figure 6: Start Capture Dialog

Depending on the type of network and the network adapter, you may immediately see packets being saved to your machine. On other configurations, you will not see anything until you create some network traffic. For a discussion of what kind of traffic you will see, refer to Appendix 1.

In either case, let's create some packets for Ethereal to capture. Leave Ethereal running and capturing packets– you should still see the box marked with a "2" in Figure 6 above. Go to a web browser (e.g., Internet Explorer, Mozilla's Firefox, Opera, or Safari), and type in a web address, such as www.cnn.com .

When the web page finished loading, go back to Ethereal and push the stop button. Ethereal will process and load summaries of all of the packets sent and received by your machine.

Don't be surprised if Ethereal captures quite a few packets of information. As Appendix 3 explains, displaying a web page requires more separate server requests than most people realize.

II) What if I can't find any packets?

There are several things to check out if you don't see packets after you push the stop button.

1) *When you were setting up Ethereal, did you select the network adapter that is being used to interface with the network?*

   Refer to section III), Figure 2, and Figure 3 in Chapter 1: Getting Started.

2) *Are you using a wireless connection on a Windows machine?*

   Windows is not able to capture packets on some wireless connections. Refer to section IV) in Appendix 1 for a possible workaround and more information.

Filters can 'hide' your traffic. Even if you didn't set a filter, some commands automatically set filters. Refer to Appendix 2 to find out how to clear filters.

3) *Are you using filters?*

   Ethereal can filter results so that you can select the packets that appear. An example of a filter condition would be to only display packets sent to/from a specific IP address. If you set a filter and then have no traffic that matches the filter you will not see any packets. For more information on filters refer to Appendix 2.

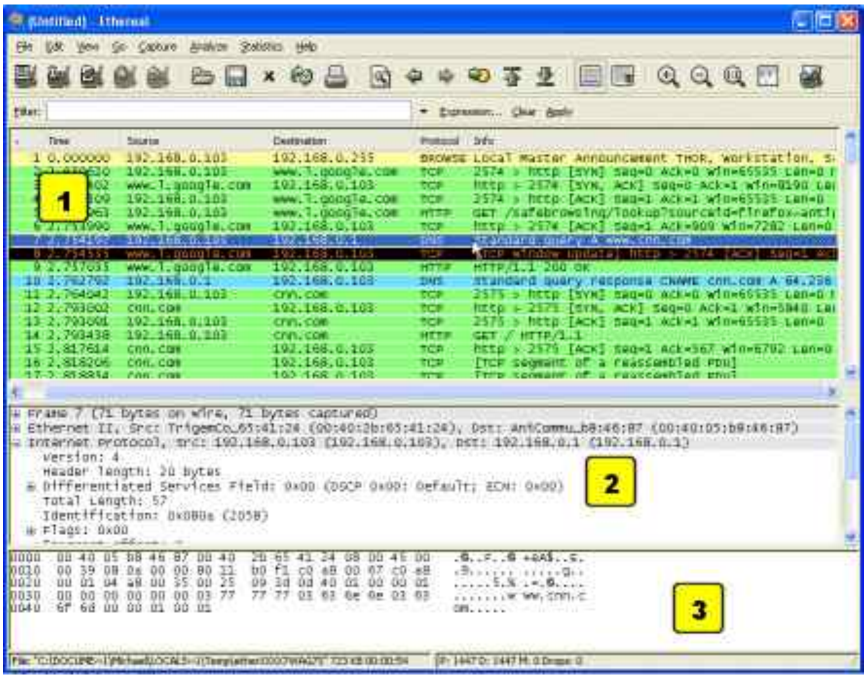
4) *Did you create any traffic for Ethereal tofilter?*

   After you go to the "Capture" menu and choose "Start", you must leave Ethereal running – do not click the "Stop" button on the dialog box. Then go to your web browser and enter a web address, such as www.cnn.com . Finally return to Ethereal and click on the "stop" button.

5) If none of these options worked, go to the Ethereal web site and check the FAQs, the documentation and the wiki at www.Ethereal.com .

III) Looking at PacketsCaptured by Ethereal

Once you have captured a set of packets, Ethereal should present you with a colorful window as shown in Figure 7 below.



(Untitled) - Ethereal
File Edit View Go Capture Analyze Statistics Help
Filter:
Expression... Clear Apply
1
2
3

```
   Time        Source              Destination         Protocol Info
 1 0.000000    192.168.0.103       192.168.0.255       BROWSE   Local Master Announcement THOR, Workstation, S
 2 2.     620  192.168.0.103       www.l.google.com    TCP      2574 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0 
 3       002   www.l.google.com    192.168.0.103       TCP      http > 2574 [SYN, ACK] Seq=0 Ack=1 Win=8190 Le
 4       309   192.168.0.103       www.l.google.com    TCP      2574 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
 5       063   192.168.0.103       www.l.google.com    HTTP     GET /safebrowsing/lookup?sourceid=firefox-anti
 6 2.753990    www.l.google.com    192.168.0.103       TCP      http > 2574 [ACK] Seq=1 Ack=909 Win=7282 Len=0
 7 2.754195    192.168.0.103       192.168.0.1         DNS      Standard query A www.cnn.com
 8 2.754555    www.l.google.com    192.168.0.103       TCP      [TCP Window Update] http > 2574 [ACK] Seq=1 Ac
 9 2.757035    www.l.google.com    192.168.0.103       HTTP     HTTP/1.1 200 OK
10 2.762792    192.168.0.1         192.168.0.103       DNS      Standard query response CNAME cnn.com A 64.236
11 2.764042    192.168.0.103       cnn.com             TCP      2575 > http [SYN] Seq=0 Ack=0 Win=65535 Len=0 
12 2.793002    cnn.com             192.168.0.103       TCP      http > 2575 [SYN, ACK] Seq=0 Ack=1 Win=5840 Le
13 2.793001    192.168.0.103       cnn.com             TCP      2575 > http [ACK] Seq=1 Ack=1 Win=65535 Len=0
14 2.793438    192.168.0.103       cnn.com             HTTP     GET / HTTP/1.1
15 2.817614    cnn.com             192.168.0.103       TCP      http > 2575 [ACK] Seq=1 Ack=567 Win=6792 Len=0
16 2.818206    cnn.com             192.168.0.103       TCP      [TCP segment of a reassembled PDU]
17 2.858834    cnn.com             192.168.0.103       TCP      [TCP segment of a reassembled PDU]
```

```
⊞ Frame 7 (71 bytes on wire, 71 bytes captured)
⊞ Ethernet II, Src: TrigemCo_65:41:24 (00:40:2b:65:41:24), Dst: AntCommu_b8:46:87 (00:40:05:b8:46:87)
⊟ Internet Protocol, Src: 192.168.0.103 (192.168.0.103), Dst: 192.168.0.1 (192.168.0.1)
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN: 0x00)
    Total Length: 57
    Identification: 0x080a (2058)
  ⊞ Flags: 0x00
```

```
0000  00 40 05 b8 46 87 00 40  2b 65 41 24 08 00 45 00   .@..F..@ +eA$..E.
0010  00 39 08 0a 00 00 80 11  b0 f1 c0 a8 00 67 c0 a8   .9...... .....g..
0020  00 01 04 a8 00 35 00 25  09 3d 08 40 01 00 00 01   .....5.% .=.@....
0030  00 00 00 00 00 00 03 77  77 77 03 63 6e 6e 03 63   .......w ww.cnn.c
0040  6f 6d 00 00 01 00 01                               om.....
```

File: "C:\DOCUME~1\Michael\LOCALS~1\Temp\ether0007WAG75" 723 KB 00:00:54   P: 1447 D: 1447 M: 0 Drops: 0

Figure 7: Packet Listing Window

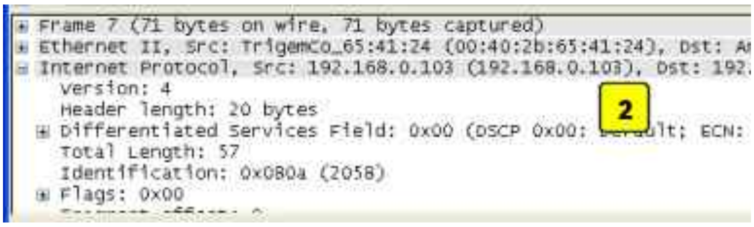
This window is divided into three areas.

i) Window Area 1

At the top is a colorful listing of all of the packets captured. Each line is a single frame or packet that was captured. The colors represent a coding scheme that can be used to quickly detect the type of packet. For example, the predominant color in the graphic above is light green. Light green is the color for HTTP packets.

ii) Window Area 2

When you click on a packet in area 1, the packet structure is shown in area 2. In the screenshot above, the packet shown in dark blue has been selected; therefore area 2 shows more details on that packet. In order to see more details, refer to Figure 8 below. This figure shows an enlarged version of area 2 from the previous figure.



```
⊞ Frame 7 (71 bytes on wire, 71 bytes captured)
⊞ Ethernet II, Src: TrigemCo_65:41:24 (00:40:2b:65:41:24), Dst: A
⊟ Internet Protocol, Src: 192.168.0.103 (192.168.0.103), Dst: 192.
    Version: 4
    Header length: 20 bytes
  ⊞ Differentiated Services Field: 0x00 (DSCP 0x00: Default; ECN:
    Total Length: 57
    Identification: 0x080a (2058)
  ⊞ Flags: 0x00
```

Figure 8: Areas 2 Details (Extract from previous figure)

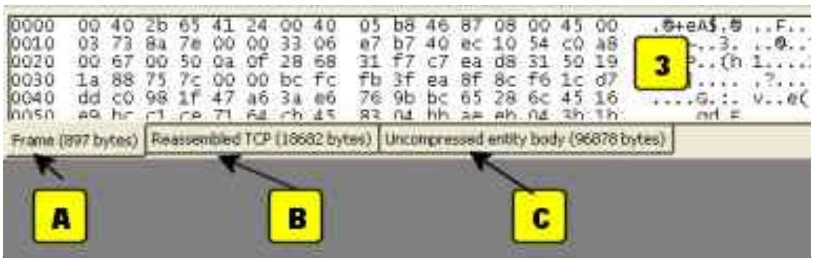
The first line of area two is created by Ethereal– it shows that this is the seventh frame (packet) that Ethereal captured. The next line in area 2 reveals that it was an Ethernet packet. Since the payload of this Ethernet packet was an Internet Protocol (IP) packet, the third line indicates that. You will also notice that there is a plus next to the first two lines and a minus next to the IP line. You can click on a plus to get more details on the packet contents. This has been done for the IP line so that the user can see the header information for the packet.

iii) Window Area 3

Clicking on a portion of the packet in area two changes the display in area 3. Area 3 has two parts. On the left are sixteen columns of two-characters each. This is the raw hexadecimal code that makes up the packet. On the right is the Unicode version of this hexadecimal code. If you click on an http line in window 2, you might notice English looking get commands or html commands in this right area.

iv) Window Area 3 Tabs- Packet Reassembly and gzip

Sometimes the third area of the packet-listing window will also have several tabs. Take a peak at Figure 9 below.



```
0000  00 40 2b 65 41 24 00 40  05 b8 46 87 08 00 45 00   .@+eA$.@ ..F...
0010  03 73 8a 7e 00 00 33 06  e7 b7 40 ec 10 54 c0 a8   ...3. ..@..
0020  00 67 00 50 0a 0f 28 68  31 f7 c7 ea d8 31 50 19   P..(h 1....
0030  1a 88 75 7c 00 00 bc fc  fb 3f ea 8f 8c f6 1c d7   .... .?...
0040  dd c0 98 1f 47 a6 3a e6  76 9b bc 65 28 6c 45 16   ....G.:. v..e(
0050  e9 bc c1 ce 71 64 cb 45  83 04 bb ae eb 04 3b 1b      qd E
```

Frame (897 bytes)
Reassembled TCP (18682 bytes)
Uncompressed entity body (96878 bytes)
3
A
B
C

Figure 9: Area 3 With Tabs

The first tab (A) shows the raw packet data for this frame, just as we discussed earlier; however this listing includes two additional tabs (B) and (C).

Remember that there is a limit on packet size. If you are receiving a large web page, the data for the page will arrive *segmented* into several smaller packets. The machine needs to assemble the entire set of packets to get the entire web page.

Figure 9 above is the data for the last packet in a stream of segmented packets. Ethereal reassembles the payload from sets of segmented packets into one common packet. This reassembled packet is shown as a tab associated with the last packet in the stream (tab B).

However, in this instance, Ethereal also displays a third tab (C). The issue is that the data in tab A and B is compressed using gzip. Most modern browsers support gzip, a compression standard. Use Ethereal to look at the original http "Get" statement sent when your browser requests a web page, and you will probably see that it supports gzip. When a web server detects gzip support it can deliver compressed data instead of plain text. This minimizes the amount of data to be sent and speeds up the transmission time. When the compressed data is received it is the uncompressed back into plain text form. In the instance above, tab (B) is the compressed data received, and tab (C) is the uncompressed data.

IV) Some Options to Analyze Captured Packets

Ethereal has several options to explore and analyze captured data. Feel free to explore the full set of options; however this section will discuss a few key capabilities.

i) Filters

Filters can be used to narrow in the focus on only important packets. See Appendix 2 for a discussion of filters.

Following a TCP stream also hides some of the data by setting a display filter. "Clear" the display filter (Appendix 2) to reveal the entire data set.

ii) Follow TCP Stream

Choose a TCP packet from the packet listing window (Area 1 in Figure 7). Right click on the chosen packet and select "Follow TCP Stream". Ethereal will open a new window and display the set of data as it is seen by the application layer. For example, in the case of a http response, this would be the http data and the web page to be delivered to the browser.

However, the "Follow TCP Stream" command also does something that may confuse you – it automatically filters the packet display so that only packets relating to this stream are displayed. As a result, you may need to "Clear" (Appendix 2) the display filter after using "Follow TCP Stream" if you want to look at other packet data.

iii) Conversations and Endpoints

Under the statistics menu at the top of the main screen you can explore "Conversations" and "Endpoints".

First, remember that the network traffic you capture may have traffic to/from more than one computer. There is a good chance that your LAN protocol is Ethernet, and Ethernet is designed to share a single network among many users. As a result, you may see packets for other users in your packet data. Even if your network is connected through a switch, you may see broadcast packets to other users.

Using endpoints lets you isolate traffic so that you are only looking at traffic to/from a specific machine. An endpoint can be defined by network layer. For example, a single MAC address on your machine is one endpoint. If you are running an email client and a web browser at the same time, all of that traffic will be consolidated through your computer's MAC address. However, if at the TCP layer, an endpoint definition includes the port number of the application. Therefore, at the TCP layer, the traffic for the email client and the web browser will be separated. Ethereal's endpoint report lets you select the network layer of interest, and then to see the summarized endpoint traffic for that layer.

A conversation report is similar to an endpoint report. A conversation is defined as all of the traffic between two specific endpoints. As an example, consider packets at the TCP level. Let's say that you started capturing packets and then went to two web sites: www.cnn.com and www.usatoday.com . The endpoint report on your web browser will combine all traffic from your browser and both of these web sites. A conversation report between your browser and the www.cnn.com site would exclude the data from www.usatoday.com .

V) Saving Captures

Ethereal also allows you to capture a set of packets and save it to a file that can be opened later. In addition to the obvious uses, this allows two unique capabilities.

- Instructors may wish to save one capture file and distribute it to all students. This allows instructors to pose a set of questions on a consistent data set, and to know that each student has appropriate data to answer the questions.

- In some circumstances, for example using a wireless network connection, students may have difficulty capturing packets. In these cases, Ethereal will still be able to analyze packets from saved files. These students can capture a set of packetson any accessible machine; save the captured packets; and transfer the saved file to their personal machine for analysis.

Appendix 1: Packets Captured: Explanation and Troubleshooting

Ethereal is designed to show you all packets that come into and out of your computer. You are probably using Ethernet for your LAN, and Ethernet is a shared-access protocol. As a result, Ethereal would theoretically allow you to see the following types of traffic:

- Packets sent to/from your computer.
- Broadcast packets sent to all computers on your local network.
- Packets sent to/from any other computers on your local network.

However, several factors may keep you from seeing some of the packets on your network.

I) Switches or Routers versus Hubs

Ethernet assumes that your local network looks like some variation of a bus, and that traffic to any computer on the local network will be seen by any other computer on that network.

In practice, Ethernet networks often use a star topology, wherein all of the computers are linked to a central unit. In the early days of Ethernet, this central unit was called a hub. A hub listens to each incoming port and repeats everything that it hears out to every port. Although a hub's physical network topology is a star, logically it acts like a bus topology – every station on the network sees all of the traffic on the network. Therefore, if your network uses a hub, your machine should be able to report both the traffic to your machine and also the traffic to all other machines on your network.

The problem with hubs is that they reduce capacity since each station must pick their packets out of a lot of irrelevant traffic for other stations. Instead, a network can be built using a switch or router as the central unit. You can refer to your textbook for a description of the differences in these devices. However, the simple explanation is that they work to insure that each station only sees the traffic that it needs to see. It is likely that your network's central unit is a switch or a router. If this is the case, your computer (and Ethereal) will be able to see traffic that is addressed to/from your computer and broadcast traffic for all computers on the network, but you will not be able to see packets sent to/from other computers that are not addressed to your computer.

II) Your Network Adapter

Many computers today have more than one network adapter. For example, many laptops have both wireless network adapters (802.11 a/b/g) and wired adapters. You must make sure that Ethereal is listening to the correct adapter or it will not see any traffic. The adapter is setup in the menu "Edit/Preferences/Capture"– make sure you choose to save any changes using the dialog button at the bottom of the window.

One of the options in the capture settings is to set "promiscuous mode". Typically, network adapters will screen out all traffic that is not destined for the computer. With this setting Ethereal will send a message to your network card telling it to pass through all traffic it sees. Even if you are on a broadcast, or hub-type network, Ethereal may not report traffic from/to other computers if promiscuous mode is not turned on.

III) Comment on Cable Modems

Typically, high-speed cable internet connections are shared connections. Theoretically, this means that you should be able to see the network traffic of your neighbors who have cable modems when you use Ethereal. The data entering your premises may include traffic from your neighbors. However, in many (most?) cases this neighbor-traffic is not visible inside your local network. Cable companies typically implement filtering and even authentication services inside their modems that eliminate packets that are not destined for the local system.

When editing preferences, save using the save button. On some monitors the button may be off the bottom of the screen, and you must move the window up to find it. If you don't save you will lose your changes.

IV) Problem with Wireless LANs and Windows

Ethereal may not be able to report packets on a Windows computer using a wireless (802.11 a/b/g) adapter. One suggested workaround is to try turning off promiscuous mode. You can find this setting in the Edit menu under the Preferences menu choice. Once the resulting dialogue box appears, click on the "Capture" menu choice on the left side. Clear the check box so that "Capture packets in promiscuous mode:" is not checked.

Click on the "Save" button at the bottom of the screen, and finally, click on the "OK" button at the bottom of the screen. On some monitors the OK button may be off of the bottom of the screen; your settings will ***not*** be saved if you click another button. Furthermore, your changes will be lost if you close the window by clicking on the x in the top righ t corner of the window.

V) Other Problems and Issues

Other problems and issues may be addressed on the Ethereal web site. Some interesting references include:

- http://wiki.Ethereal.com/CaptureSetup
- http://www.Ethereal.com/docs/
- http://www.Ethereal.com/faq.html
- http://wiki.Ethereal.com/

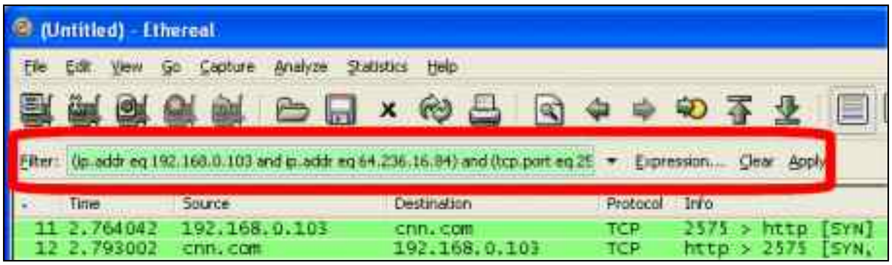
Appendix 2: Filters in Ethereal

Ethereal can filter results so that you only see certain packets. An example of a filter condition would be to only remember packets sent to/from a specific IP address.

Ethereal uses two types of filters, capture filters and display filters. Capture filters are used to decide which packets should be kept. Only packets that meet filter criteria will be kept. Display filters work after the capture is completed. They restrict which packets are shown, but they don't actually discard any information. Capture filters would be more useful on very busy networks when you need to limit the amount of data your machine needs to process. On the other hand, display filters don't actually save any memory; display filters let you temporarily focus an analysis without losing any underlying information.

Capture filters can be set in two different places. Go to the Capture menu and select "Options" and you will find a selection for capture filters. Alternatively, Go to the Capture menu and select "Capture Filters". From the "Capture Filters" dialog box you will see a help menu that will explain how the function works.

Display filters can be entered at the top of the display screen. Figure 10 below shows a display filter entered into the display filter dialog box at the top of the screen.



Filter: (ip.addr eq 192.168.0.103 and ip.addr eq 64.236.16.84) and (tcp.port eq 25

```
   Time         Source           Destination      Protocol  Info
11 2.764042    192.168.0.103    cnn.com          TCP       2575 > http [SYN]
12 2.793002    cnn.com          192.168.0.103    TCP       http > 2575 [SYN,
```

Figure 10: Using Display Filters

The display filter shown in the image above will only display packets if they are from/to IP address 192.168.0.103 and 64.23616.84 and if they arrive on tcp port 2575. This specific filter limited packets to those involved in one web session with CNN.com . If you also captured traffic to USAToday.com, you would not be able to see it until you clicked on "Clear" to the right of the filter area.

Even if you have never entered a filter, some commands automatically enter filters for you– for example the "Follow TCP Stream" command. If you find data is missing, make sure that there is not a display filter entered at the top of the screen. You can click on the word "Clear" to the right of the filter text box.

Some commands, such as “Follow TCP Stream” automatically enter values in the filter field. After you use a command like this, you may need to “Clear” the filter to see the complete set of packets.

Appendix 3: Hits Versus Page Views

It may take more effort than you realize to deliver a web page to your computer. The first step is to get the raw HTML code for the page. Getting this code takes several sets of packets– the details will be left to an exercise to be completed later, but suffice it to say that retrieval includes setup and control packets as well as query and response packets. Furthermore, in most cases the response will be a multi-packet data burst that must be reassembled into a complete http response.

However, once the page is delivered to the application, the system has only completed the first step required to display the web page. Let's consider a simplified web page in HTML, as shown in the box below.

```
<HTML>
<Body>
Look at this pretty Christmas tree.<br>
<img src=tree.jpg>
</Body>
</HTML>
```

Figure 11: Simplified Web Page

This web page will display a short sentence (Look at this pretty Christmas tree.), followed by a line break, and then a picture of a tree. Notice thatthe picture of the tree is not part of the HTML page that is delivered. All that gets delivered with the page is a placeholder that tells the browser to get the picture called tree.jpg and to put it into a specific spot on the page.

So, once the browser deciphers the web page, it knows it must make another request of the web server. Now the browser asks for the picture tree.jpg. As a result, displaying this page takes two hits on the browser. One hit (or request) was for the original web page, and the second hit was for the picture to be embedded into the web page. Each additional picture or external page element is another hit on the web page.

How many pictures are on a single page? 10? 20? A recent analysis of the CNN front page indicated over one hundred and fifty separate files were required to display the page. A lot of these files are graphic files. This includes tiny graphic arrows, almost invisible lines, menu choices, and advertisements. In addition, javascript files, stylesheets, and iFrames can all be external links, and thus can be additional sources of hits.

This topic is appropriate for this guide because it helps explains the plethora of packets that add together to display a single web page. However, it is also interesting to consider the implications for the number of 'hits' a web site gets.

Let's analyze what it takes to get a million hits on a web page. First, assume an average page has 150 images. In comparison, this would be 10% smaller than CNN's front page. Now, assume each visitor sees three pages on the web site. It will take less than 2,300 visitors to get one million hits on this hypothetical web site.

150 Hits/Page
X 3 Pages/Visitor
X 2,300 Visitors
= 1,035,000 hits

Especially in the case of advertisements, these hits may not come from the original web site. Therefore, at the packet level there may be many packets from many different sources that have to be considered as part of the same web page.

Increasingly, developers are making dynamic web pages. This means that some portion of the web page may be continuously updated through interaction between the user and the server. This dynamic process requires ongoing hits on the server, even after the web page is initially ‘complete’.

Since each of these hits results in a new request from the server, the number of packets required to assemble a web page is larger than many people realize.