d. In addition to reducing delay, what are reasons to use message segmentation?

e. Discuss the drawbacks of message segmentation.

P32. Experiment with the Message Segmentation applet at the book’s Web site. Do the delays in the applet correspond to the delays in the previous problem? How do link propagation delays affect the overall end-to-end delay for packet switching (with message segmentation) and for message switching?

P33. Consider sending a large file of F bits from Host A to Host B. There are three links (and two switches) between A and B, and the links are uncongested (that is, no queuing delays). Host A segments the file into segments of S bits each and adds 80 bits of header to each segment, forming packets of $L = 80 + S$ bits. Each link has a transmission rate of R bps. Find the value of S that minimizes the delay of moving the file from Host A to Host B. Disregard propagation delay.

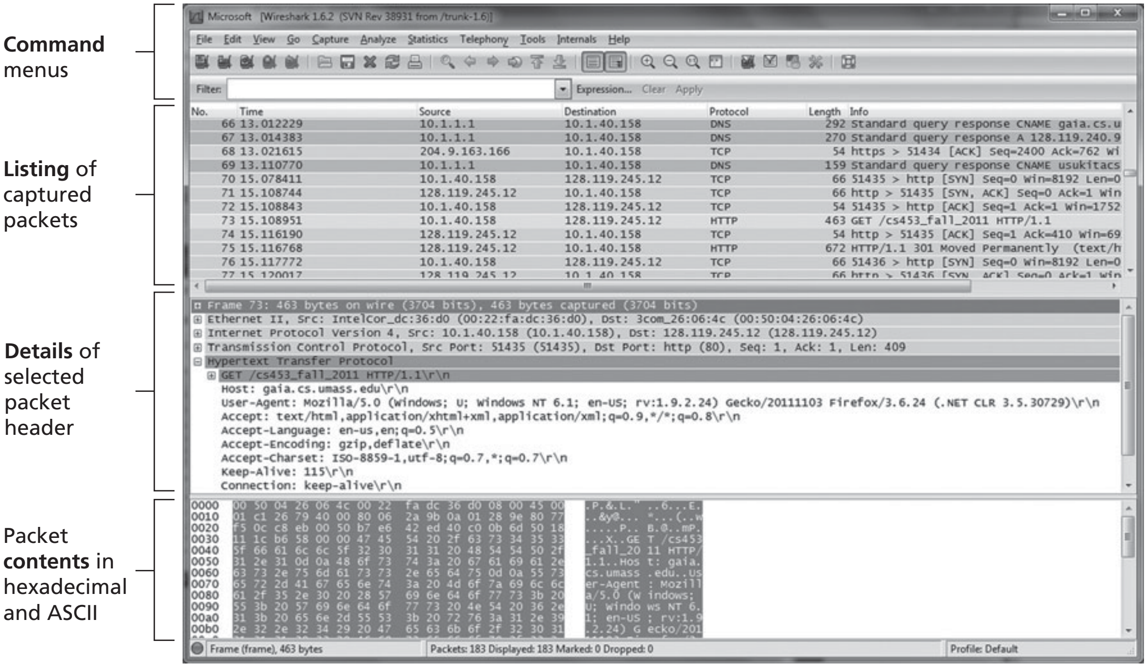
P34. Skype offers a service that allows you to make a phone call from a PC to an ordinary phone. This means that the voice call must pass through both the Internet and through a telephone network. Discuss how this might be done.

Wireshark Lab

“Tell me and I forget. Show me and I remember. Involve me and I understand.”
Chinese proverb

One’s understanding of network protocols can often be greatly deepened by seeing them in action and by playing around with them—observing the sequence of messages exchanged between two protocol entities, delving into the details of protocol operation, causing protocols to perform certain actions, and observing these actions and their consequences. This can be done in simulated scenarios or in a real network environment such as the Internet. The Java applets at the textbook Web site take the first approach. In the Wireshark labs, we’ll take the latter approach. You’ll run network applications in various scenarios using a computer on your desk, at home, or in a lab. You’ll observe the network protocols in your computer, interacting and exchanging messages with protocol entities executing elsewhere in the Internet. Thus, you and your computer will be an integral part of these live labs. You’ll observe—and you’ll learn—by doing.

The basic tool for observing the messages exchanged between executing protocol entities is called a **packet sniffer**. As the name suggests, a packet sniffer passively copies (sniffs) messages being sent from and received by your computer; it also displays the contents of the various protocol fields of these captured messages. A screenshot of the Wireshark packet sniffer is shown in Figure 1.28. Wireshark is a free packet sniffer that runs on Windows, Linux/Unix, and Mac


Command menus
Listing of captured packets
Details of selected packet header
Packet contents in hexadecimal and ASCII
Microsoft [Wireshark 1.6.2 (SVN Rev 38931 from /trunk-1.6)]
File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help
Filter:
Expression... Clear Apply
Frame (frame), 463 bytes
Packets: 183 Displayed: 183 Marked: 0 Dropped: 0
Profile: Default

```
No.  Time       Source          Destination     Protocol  Length Info
  66 13.012229  10.1.1.1        10.1.40.158     DNS         292 Standard query response CNAME gaia.cs.u
  67 13.014383  10.1.1.1        10.1.40.158     DNS         270 Standard query response A 128.119.240.9
  68 13.021615  204.9.163.166   10.1.40.158     TCP          54 https > 51434 [ACK] Seq=2400 Ack=762 Wi
  69 13.110770  10.1.1.1        10.1.40.158     DNS         159 Standard query response CNAME usukitacs
  70 15.078411  10.1.40.158     128.119.245.12  TCP          66 51435 > http [SYN] Seq=0 Win=8192 Len=0
  71 15.108744  128.119.245.12  10.1.40.158     TCP          66 http > 51435 [SYN, ACK] Seq=0 Ack=1 Win
  72 15.108843  10.1.40.158     128.119.245.12  TCP          54 51435 > http [ACK] Seq=1 Ack=1 Win=1752
  73 15.108951  10.1.40.158     128.119.245.12  HTTP        463 GET /cs453_fall_2011 HTTP/1.1
  74 15.116190  128.119.245.12  10.1.40.158     TCP          54 http > 51435 [ACK] Seq=1 Ack=410 Win=69
  75 15.116768  128.119.245.12  10.1.40.158     HTTP        672 HTTP/1.1 301 Moved Permanently  (text/h
  76 15.117772  10.1.40.158     128.119.245.12  TCP          66 51436 > http [SYN] Seq=0 Win=8192 Len=0
  77 15.120017  128.119.245.12  10.1.40.158     TCP          66 http > 51436 [SYN, ACK] Seq=0 Ack=1 Win
```

```
⊞ Frame 73: 463 bytes on wire (3704 bits), 463 bytes captured (3704 bits)
⊞ Ethernet II, Src: IntelCor_dc:36:d0 (00:22:fa:dc:36:d0), Dst: 3com_26:06:4c (00:50:04:26:06:4c)
⊞ Internet Protocol Version 4, Src: 10.1.40.158 (10.1.40.158), Dst: 128.119.245.12 (128.119.245.12)
⊞ Transmission Control Protocol, Src Port: 51435 (51435), Dst Port: http (80), Seq: 1, Ack: 1, Len: 409
⊟ Hypertext Transfer Protocol
  ⊞ GET /cs453_fall_2011 HTTP/1.1\r\n
    Host: gaia.cs.umass.edu\r\n
    User-Agent: Mozilla/5.0 (Windows; U; Windows NT 6.1; en-US; rv:1.9.2.24) Gecko/20111103 Firefox/3.6.24 (.NET CLR 3.5.30729)\r\n
    Accept: text/html,application/xhtml+xml,application/xml;q=0.9,*/*;q=0.8\r\n
    Accept-Language: en-us,en;q=0.5\r\n
    Accept-Encoding: gzip,deflate\r\n
    Accept-Charset: ISO-8859-1,utf-8;q=0.7,*;q=0.7\r\n
    Keep-Alive: 115\r\n
    Connection: keep-alive\r\n
```

```
0000  00 50 04 26 06 4c 00 22  fa dc 36 d0 08 00 45 00   .P.&.L." ..6...E.
0010  01 c1 26 79 40 00 80 06  2a 9b 0a 01 28 9e 80 77   ..&y@... *...(..w
0020  f5 0c c8 eb 00 50 b7 e6  42 ed 40 c0 0b 6d 50 18   .....P.. B.@..mP.
0030  11 1c b6 58 00 00 47 45  54 20 2f 63 73 34 35 33   ...X..GE T /cs453
0040  5f 66 61 6c 6c 5f 32 30  31 31 20 48 54 54 50 2f   _fall_20 11 HTTP/
0050  31 2e 31 0d 0a 48 6f 73  74 3a 20 67 61 69 61 2e   1.1..Hos t: gaia.
0060  63 73 2e 75 6d 61 73 73  2e 65 64 75 0d 0a 55 73   cs.umass .edu..Us
0070  65 72 2d 41 67 65 6e 74  3a 20 4d 6f 7a 69 6c 6c   er-Agent : Mozill
0080  61 2f 35 2e 30 20 28 57  69 6e 64 6f 77 73 3b 20   a/5.0 (W indows; 
0090  55 3b 20 57 69 6e 64 6f  77 73 20 4e 54 20 36 2e   U; Windo ws NT 6.
00a0  31 3b 20 65 6e 2d 55 53  3b 20 72 76 3a 31 2e 39   1; en-US ; rv:1.9
00b0  2e 32 2e 32 34 29 20 47  65 63 6b 6f 2f 32 30 31   .2.24) G ecko/201
```

Figure 1.28 ♦ A Wireshark screen shot (Wireshark screenshot reprinted by permission of the Wireshark Foundation.)

computers. Throughout the textbook, you will find Wireshark labs that allow you to explore a number of the protocols studied in the chapter. In this first Wireshark lab, you'll obtain and install a copy of Wireshark, access a Web site, and capture and examine the protocol messages being exchanged between your Web browser and the Web server.

You can find full details about this first Wireshark lab (including instructions about how to obtain and install Wireshark) at the Web site http://www.awl.com/kurose-ross.