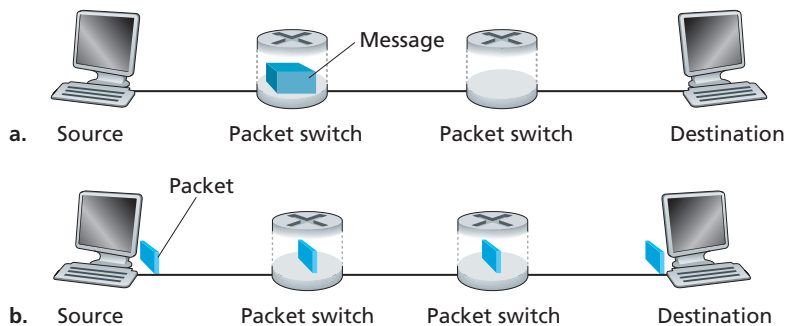


- (b) Repeat (a) but this time choose one city in France and another city in Germany.
- (c) Pick a city in the United States, and perform traceroutes to two hosts, each in a different city in China. How many links are common in the two traceroutes? Do the two traceroutes diverge before reaching China?
- P20.** Consider the throughput example corresponding to Figure 1.20(b). Now suppose that there are  $M$  client-server pairs rather than 10. Denote  $R_s$ ,  $R_c$ , and  $R$  for the rates of the server links, client links, and network link. Assume all other links have abundant capacity and that there is no other traffic in the network besides the traffic generated by the  $M$  client-server pairs. Derive a general expression for throughput in terms of  $R_s$ ,  $R_c$ ,  $R$ , and  $M$ .
- P21.** Consider Figure 1.19(b). Now suppose that there are  $M$  paths between the server and the client. No two paths share any link. Path  $k$  ( $k = 1, \dots, M$ ) consists of  $N$  links with transmission rates  $R_1^k, R_2^k, \dots, R_N^k$ . If the server can only use one path to send data to the client, what is the maximum throughput that the server can achieve? If the server can use all  $M$  paths to send data, what is the maximum throughput that the server can achieve?
- P22.** Consider Figure 1.19(b). Suppose that each link between the server and the client has a packet loss probability  $p$ , and the packet loss probabilities for these links are independent. What is the probability that a packet (sent by the server) is successfully received by the receiver? If a packet is lost in the path from the server to the client, then the server will re-transmit the packet. On average, how many times will the server re-transmit the packet in order for the client to successfully receive the packet?
- P23.** Consider Figure 1.19(a). Assume that we know the bottleneck link along the path from the server to the client is the first link with rate  $R_s$  bits/sec. Suppose we send a pair of packets back to back from the server to the client, and there is no other traffic on this path. Assume each packet of size  $L$  bits, and both links have the same propagation delay  $d_{prop}$ .
- What is the packet inter-arrival time at the destination? That is, how much time elapses from when the last bit of the first packet arrives until the last bit of the second packet arrives?
  - Now assume that the second link is the bottleneck link (i.e.,  $R_c < R_s$ ). Is it possible that the second packet queues at the input queue of the second link? Explain. Now suppose that the server sends the second packet  $T$  seconds after sending the first packet. How large must  $T$  be to ensure no queuing before the second link? Explain.
- P24.** Suppose you would like to urgently deliver 40 terabytes data from Boston to Los Angeles. You have available a 100 Mbps dedicated link for data transfer. Would you prefer to transmit the data via this link or instead use FedEx overnight delivery? Explain.

the protocol stack. Is there an equivalent notion of header information that is added to passengers and baggage as they move down the airline protocol stack?

**P31.** In modern packet-switched networks, including the Internet, the source host segments long, application-layer messages (for example, an image or a music file) into smaller packets and sends the packets into the network. The receiver then reassembles the packets back into the original message. We refer to this process as *message segmentation*. Figure 1.27 illustrates the end-to-end transport of a message with and without message segmentation. Consider a message that is  $8 \cdot 10^6$  bits long that is to be sent from source to destination in Figure 1.27. Suppose each link in the figure is 2 Mbps. Ignore propagation, queuing, and processing delays.

- Consider sending the message from source to destination *without* message segmentation. How long does it take to move the message from the source host to the first packet switch? Keeping in mind that each switch uses store-and-forward packet switching, what is the total time to move the message from source host to destination host?
- Now suppose that the message is segmented into 800 packets, with each packet being 10,000 bits long. How long does it take to move the first packet from source host to the first switch? When the first packet is being sent from the first switch to the second switch, the second packet is being sent from the source host to the first switch. At what time will the second packet be fully received at the first switch?
- How long does it take to move the file from source host to destination host when message segmentation is used? Compare this result with your answer in part (a) and comment.



**Figure 1.27** ♦ End-to-end message transport: (a) without message segmentation; (b) with message segmentation

- 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

- P6. Obtain the HTTP/1.1 specification (RFC 2616). Answer the following questions:
- Explain the mechanism used for signaling between the client and server to indicate that a persistent connection is being closed. Can the client, the server, or both signal the close of a connection?
  - What encryption services are provided by HTTP?
  - Can a client open three or more simultaneous connections with a given server?
  - Either a server or a client may close a transport connection between them if either one detects the connection has been idle for some time. Is it possible that one side starts closing a connection while the other side is transmitting data via this connection? Explain.
- P7. Suppose within your Web browser you click on a link to obtain a Web page. The IP address for the associated URL is not cached in your local host, so a DNS lookup is necessary to obtain the IP address. Suppose that  $n$  DNS servers are visited before your host receives the IP address from DNS; the successive visits incur an RTT of  $RTT_1, \dots, RTT_n$ . Further suppose that the Web page associated with the link contains exactly one object, consisting of a small amount of HTML text. Let  $RTT_0$  denote the RTT between the local host and the server containing the object. Assuming zero transmission time of the object, how much time elapses from when the client clicks on the link until the client receives the object?
- P8. Referring to Problem P7, suppose the HTML file references eight very small objects on the same server. Neglecting transmission times, how much time elapses with
- Non-persistent HTTP with no parallel TCP connections?
  - Non-persistent HTTP with the browser configured for 5 parallel connections?
  - Persistent HTTP?
- P9. Consider Figure 2.12, for which there is an institutional network connected to the Internet. Suppose that the average object size is 850,000 bits and that the average request rate from the institution's browsers to the origin servers is 16 requests per second. Also suppose that the amount of time it takes from when the router on the Internet side of the access link forwards an HTTP request until it receives the response is three seconds on average (see Section 2.2.5). Model the total average response time as the sum of the average access delay (that is, the delay from Internet router to institution router) and the average Internet delay. For the average access delay, use  $\Delta/(1 - \Delta\beta)$ , where  $\Delta$  is the average time required to send an object over the access link and  $\beta$  is the arrival rate of objects to the access link.
- Find the total average response time.
  - Now suppose a cache is installed in the institutional LAN. Suppose the miss rate is 0.4. Find the total response time.

- P10.** Consider a short, 10-meter link, over which a sender can transmit at a rate of 150 bits/sec in both directions. Suppose that packets containing data are 100,000 bits long, and packets containing only control (e.g., ACK or handshaking) are 200 bits long. Assume that  $N$  parallel connections each get  $1/N$  of the link bandwidth. Now consider the HTTP protocol, and suppose that each downloaded object is 100 Kbits long, and that the initial downloaded object contains 10 referenced objects from the same sender. Would parallel downloads via parallel instances of non-persistent HTTP make sense in this case? Now consider persistent HTTP. Do you expect significant gains over the non-persistent case? Justify and explain your answer.
- P11. Consider the scenario introduced in the previous problem. Now suppose that the link is shared by Bob with four other users. Bob uses parallel instances of non-persistent HTTP, and the other four users use non-persistent HTTP without parallel downloads.
- Do Bob's parallel connections help him get Web pages more quickly? Why or why not?
  - If all five users open five parallel instances of non-persistent HTTP, then would Bob's parallel connections still be beneficial? Why or why not?
- P12. Write a simple TCP program for a server that accepts lines of input from a client and prints the lines onto the server's standard output. (You can do this by modifying the `TCPServer.py` program in the text.) Compile and execute your program. On any other machine that contains a Web browser, set the proxy server in the browser to the host that is running your server program; also configure the port number appropriately. Your browser should now send its GET request messages to your server, and your server should display the messages on its standard output. Use this platform to determine whether your browser generates conditional GET messages for objects that are locally cached.
- P13. What is the difference between `MAIL FROM:` in SMTP and `From:` in the mail message itself?
- P14. How does SMTP mark the end of a message body? How about HTTP? Can HTTP use the same method as SMTP to mark the end of a message body? Explain.
- P15. Read RFC 5321 for SMTP. What does MTA stand for? Consider the following received spam email (modified from a real spam email). Assuming only the originator of this spam email is malicious and all other hosts are honest, identify the malicious host that has generated this spam email.

```
From - Fri Nov 07 13:41:30 2008
Return-Path: <tennis5@pp33head.com>
Received: from barmail.cs.umass.edu
(barmail.cs.umass.edu [128.119.240.3]) by cs.umass.edu
(8.13.1/8.12.6) for <hg@cs.umass.edu>; Fri, 7 Nov 2008
13:27:10 -0500
```

- P22. Suppose now that the leftmost router in Figure 5.33 is replaced by a switch. Hosts A, B, C, and D and the right router are all star-connected into this switch. Give the source and destination MAC addresses in the frame encapsulating this IP datagram as the frame is transmitted (i) from A to the switch, (ii) from the switch to the right router, (iii) from the right router to F. Also give the source and destination IP addresses in the IP datagram encapsulated within the frame at each of these points in time.
- P23. Consider Figure 5.15. Suppose that all links are 100 Mbps. What is the maximum total aggregate throughput that can be achieved among the 9 hosts and 2 servers in this network? You can assume that any host or server can send to any other host or server. Why?
- P24. Suppose the three departmental switches in Figure 5.15 are replaced by hubs. All links are 100 Mbps. Now answer the questions posed in problem P23.
- P25. Suppose that *all* the switches in Figure 5.15 are replaced by hubs. All links are 100 Mbps. Now answer the questions posed in problem P23.
- P26. Let's consider the operation of a learning switch in the context of a network in which 6 nodes labeled A through F are star connected into an Ethernet switch. Suppose that (i) B sends a frame to E, (ii) E replies with a frame to B, (iii) A sends a frame to B, (iv) B replies with a frame to A. The switch table is initially empty. Show the state of the switch table before and after each of these events. For each of these events, identify the link(s) on which the transmitted frame will be forwarded, and briefly justify your answers.
- P27.** In this problem, we explore the use of small packets for Voice-over-IP applications. One of the drawbacks of a small packet size is that a large fraction of link bandwidth is consumed by overhead bytes. To this end, suppose that the packet consists of  $P$  bytes and 5 bytes of header.
- Consider sending a digitally encoded voice source directly. Suppose the source is encoded at a constant rate of 128 kbps. Assume each packet is entirely filled before the source sends the packet into the network. The time required to fill a packet is the **packetization delay**. In terms of  $L$ , determine the packetization delay in milliseconds.
  - Packetization delays greater than 20 msec can cause a noticeable and unpleasant echo. Determine the packetization delay for  $L = 1,500$  bytes (roughly corresponding to a maximum-sized Ethernet packet) and for  $L = 50$  (corresponding to an ATM packet).
  - Calculate the store-and-forward delay at a single switch for a link rate of  $R = 622$  Mbps for  $L = 1,500$  bytes, and for  $L = 50$  bytes.
  - Comment on the advantages of using a small packet size.