◄ PREVIOUS   NEXT ►

2.1 Introduction

This chapter provides an overview of the protocols in the TCP/IP suite that are used in the examples throughout the book. Our goal is to provide enough detail from a network programming perspective to understand how to use the protocols and provide references to more detailed descriptions of their actual design, implementation, and history.

This chapter focuses on the transport layer: TCP, UDP, and Stream Control Transmission Protocol (SCTP). Most client/server applications use either TCP or UDP. SCTP is a newer protocol, originally designed for transport of telephony signaling across the Internet. These transport protocols use the network-layer protocol IP, either IPv4 or IPv6. While it is possible to use IPv4 or IPv6 directly, bypassing the transport layer, this technique, often called *raw sockets*, is used much less frequently. Therefore, we have a more detailed description of IPv4 and IPv6, along with ICMPv4 and ICMPv6, in Appendix A.

UDP is a simple, unreliable datagram protocol, while TCP is a sophisticated, reliable byte-stream protocol. SCTP is similar to TCP as a reliable transport protocol, but it also provides message boundaries, transport-level support for multihoming, and a way to minimize head-of-line blocking. We need to understand the services provided by these transport protocols to the application, so that we know what is handled by the protocol and what we must handle in the application.

There are features of TCP that, when understood, make it easier for us to write robust clients and servers. Also, when we understand these features, it becomes easier to debug our clients and servers using commonly provided tools such as `netstat`. We cover various topics in this chapter that fall into this category: TCP's three-way handshake, TCP's connection termination sequence, and TCP's TIME_WAIT state; SCTP's four-way handshake and SCTP's connection termination; plus SCTP, TCP, and UDP buffering by the socket layer, and so on.

◄ PREVIOUS   NEXT ►