8.12 dg_cli Function (Revisited)

We now return to the dg_cli function from Figure 8.8 and recode it to call connect. Figure 8.17 shows the new function.

Figure 8.17 dg_cli function that calls connect.

udpcliserv/dgcliconnect.c

```
#include     "unp.h"

void
dg_cli(FILE *fp, int sockfd, const SA *pservaddr, socklen_t servlen)
{
    int     n;
    char    sendline[MAXLINE], recvline[MAXLINE + 1];

    Connect(sockfd, (SA *) pservaddr, servlen);

    while (Fgets(sendline, MAXLINE, fp) != NULL) {

        Write(sockfd, sendline, strlen(sendline));

        n = Read(sockfd, recvline, MAXLINE);

        recvline[n] = 0;        /* null terminate */
        Fputs(recvline, stdout);
    }
}
```

The changes are the new call to connect and replacing the calls to sendto and recvfrom with calls to write and read. This function is still protocol-independent since it doesn't look inside the socket address structure that is passed to connect. Our client main function, Figure 8.7, remains the same.

If we run this program on the host macosx, specifying the IP address of the host freebsd4 (which is not running our server on port 9877), we have the following output:

```
macosx % udpcli04 172.24.37.94
hello, world
read error: Connection refused
```

The first point we notice is that we do *not* receive the error when we start the client process. The error occurs only after we send the first datagram to the server. It is sending this datagram that elicits the ICMP error from the server host. But when a TCP client calls connect, specifying a server host that is not running the server process, connect returns the error because the call to connect causes the TCP three-way handshake to happen, and the first packet of that handshake elicits an RST from the server TCP (Section 4.3).

Figure 8.18 shows the tcpdump output.

Figure 8.18 tcpdump output when running Figure 8.17.

```
macosx % tcpdump
1  0.0                    macosx.51139 > freebsd4.9877: udp 13
2  0.006180 ( 0.0062)     freebsd4 > macosx: icmp: freebsd4 udp port 9877 unreachable
```

We also see in Figure A.15 that this ICMP error is mapped by the kernel into the error ECONNREFUSED, which corresponds to the message string output by our err_sys function: "Connection refused."

> Unfortunately, not all kernels return ICMP messages to a connected UDP socket, as we have shown in this section. Normally, Berkeley-derived kernels return the error, while System V kernels do not. For example, if we run the same client on a Solaris 2.4 host and connect to a host that is not running our server, we can watch with tcpdump and verify that the ICMP "port unreachable" error is returned by the server host, but the client's call to read never returns. This bug was fixed in Solaris 2.5. UnixWare does not return the error, while AIX, Digital Unix, HP-UX, and Linux all return the error.