47.3.7 NOTES

Most of these routines quietly but politely return `undef` when they fail instead of causing your program to die right then and there due to an uncaught exception. (Actually, some of the new *Socket* conversion functions croak() on bad arguments.) It is therefore essential to check return values from these functions. Always begin your socket programs this way for optimal success, and don't forget to add **-T** taint checking flag to the #! line for servers:

```
#!/usr/bin/perl -Tw
use strict;
use sigtrap;
use Socket;
```

47.3.8 BUGS

All these routines create system-specific portability problems. As noted elsewhere, Perl is at the mercy of your C libraries for much of its system behaviour. It's probably safest to assume broken SysV semantics for signals and to stick with simple TCP and UDP socket operations; e.g., don't try to pass open file descriptors over a local UDP datagram socket if you want your code to stand a chance of being portable.

47.3.9 AUTHOR

Tom Christiansen, with occasional vestiges of Larry Wall's original version and suggestions from the Perl Porters.

47.3.10 SEE ALSO

There's a lot more to networking than this, but this should get you started.

For intrepid programmers, the indispensable textbook is *Unix Network Programming, 2nd Edition, Volume 1* by W. Richard Stevens (published by Prentice-Hall). Note that most books on networking address the subject from the perspective of a C programmer; translation to Perl is left as an exercise for the reader.

The IO::Socket(3) manpage describes the object library, and the Socket(3) manpage describes the low-level interface to sockets. Besides the obvious functions in *perlfunc*, you should also check out the *modules* file at your nearest CPAN site. (See *perlmodlib* or best yet, the *Perl FAQ* for a description of what CPAN is and where to get it.)

Section 5 of the *modules* file is devoted to "Networking, Device Control (modems), and Interprocess Communication", and contains numerous unbundled modules numerous networking modules, Chat and Expect operations, CGI programming, DCE, FTP, IPC, NNTP, Proxy, Ptty, RPC, SNMP, SMTP, Telnet, Threads, and ToolTalk–just to name a few.