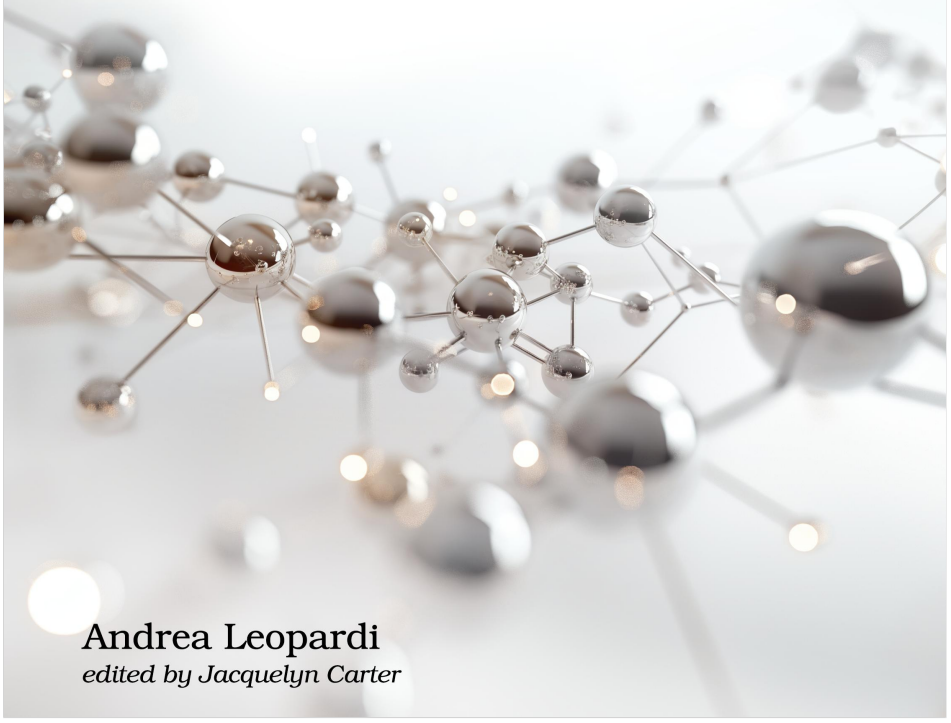
The Pragmatic Programmers

Your Elixir Source

Network Programming in Elixir and Erlang

Write High-Performance, Scalable, and Reliable Apps with TCP and UDP



Andrea Leopardi
edited by Jacquelyn Carter

This extract shows the online version of this title, and may contain features (such as hyperlinks and colors) that are not available in the print version.

For more information, or to purchase a paperback or ebook copy, please visit https://www.pragprog.com.

Introduction

Elixir and Erlang, and their shared ecosystem, are a fantastic fit for writing network code. After all, Erlang was invented to solve a specific *network problem*: routing and handling telephone calls. Even though networks have evolved since then, the underlying problems have remained the same. Real-time web applications have a lot in common with telephone calls—you have clients in different places that should ideally stay connected with each other and exchange data efficiently and reliably.

Erlang's standard library includes all the fundamental tools you need to work with network protocols and to build network applications. You'll learn about ecosystem libraries, as well as the design patterns and best practices to use when writing this kind of software on the BEAM. Erlang and Elixir's ecosystems also have plenty of third-party libraries that abstract away the most common network patterns and use cases (such as HTTP servers and clients).

Is This Book for You?

This book is mainly aimed at developers with Elixir and Erlang experience but without much experience in networks. It assumes a basic understanding of the syntax of these languages as well as of core concepts such as functions, processes, message passing, and OTP. However, the book doesn't assume any network knowledge. Instead, it'll walk you through the basic principle of networks and of network protocols from the ground up.

This book might also be a good fit for you if you are knowledgeable in networks, but you have no experience with Erlang or Elixir: you'll get a fantastic tool under your belt for working with networks. You might have to read up about Elixir- and Erlang-specific concepts here and there, but the book will show you how good of a fit these languages are for this domain.

About this Book

This book is split up into three parts, each focusing on one important network protocol: TCP, UDP, and HTTP.

Let's take a more detailed look at the plan.

We start with an introduction to networks, in Chapter 1, What Is Network Programming, Anyway?, on page ?. Learning how to use the tools that Erlang and Elixir provide requires some understanding of how networks work and the details of the protocols involved.

Part I: TCP

The first part of the book is dedicated to TCP, the most widely used network protocol. Chapter 2, TCP: Exploring the Basics, on page ?, introduces the protocol itself. The following two chapters, Chapter 3, Designing a Chat Protocol and Its TCP Server, on page ? and Chapter 4, Scaling TCP on the Server Side, on page ?, focus on the server-side aspect of working with TCP. Then, Chapter 5, Building TCP Clients, on page ? and Chapter 6, Scaling and Optimizing TCP Clients, on page ? switch to the client side. The last chapter in this part, *the (as yet) unwritten Chapter 7, Securing Protocols: TLS,* is about securing network traffic over TCP by using TLS.

Part II: UDP

In this second part, we explore UDP, a protocol which is quite different from TCP but a widespread one nonetheless. You'll start by learning the protocol basics in Chapter 8, Same Layer, Different Protocol: Introducing UDP, on page ?. Then, you'll learn about techniques to increase the reliability of UDP with fine-grained control (Chapter 9, Adding Guarantees to UDP, on page ?). The last chapter in this part, *the (as yet) unwritten Chapter 10, UDP in the Wild: DNS,* is about DNS, the "internet phone book" and a protocol mostly used on top of UDP.

Part III: HTTP

In the last part of the book, we focus on HTTP, which is a protocol you'll most likely use directly at some point in your career (if not for most of it). In the first chapter, *the (as yet) unwritten Chapter 11, Talking the Internet Protocols: HTTP/1.1 and HTTP/2,* we explore HTTP/1.1 and HTTP/2. Then, we look at a protocol built on top of HTTP and which powers many real-time interactions on the Web: WebSockets (*the (as yet) unwritten Chapter 12, Communicating in Real-Time with WebSockets,*).

About the Code

The code in this book is mostly structured so that you can follow along and type it out yourself if that's your thing. However, we'll sometimes just refer you to code that you can find in the online resources, to avoid interrupting the flow too much.

All the example code is in Elixir, even though the book mentions Erlang in its title. Rather than providing an Erlang version of every snippet you'll encounter, we opted for a concise tutorial on understanding Elixir syntax in case you're only familiar with Erlang syntax. You can find that in Appendix 2, Comparing Elixir and Erlang Syntaxes, on page ?. The concepts discussed in this book are not specific to either language, and the two languages share so much of their DNA that we believe a "translation guide" is enough for everyone to enjoy the content.

Online Resources

The apps and examples shown in this book can be found at the Pragmatic Programmers website for this book.[1] You'll also find the errata-submission form, where you can report problems with the text or make suggestions for future versions.

When you're ready, turn the page and we'll get started.

1. http://pragprog.com/titles/alnpee/