

# CONTENTS IN DETAIL

<b>ACKNOWLEDGMENTS</b>	<b>XVII</b>
------------------------	-------------

<b>INTRODUCTION</b>	<b>XIX</b>
---------------------	------------

Who This Book Is For . . . . .	xx
Installing Go . . . . .	xx
Recommended Development Environments . . . . .	xxi
What's in This Book . . . . .	xxi

## **PART I: NETWORK ARCHITECTURE** **1**

<b>1</b>	
<b>AN OVERVIEW OF NETWORKED SYSTEMS</b>	<b>3</b>

Choosing a Network Topology . . . . .	3
Bandwidth vs. Latency . . . . .	6
The Open Systems Interconnection Reference Model . . . . .	7
The Hierarchical Layers of the OSI Reference Model . . . . .	8
Sending Traffic by Using Data Encapsulation . . . . .	10
The TCP/IP Model . . . . .	12
The Application Layer . . . . .	13
The Transport Layer . . . . .	14
The Internet Layer . . . . .	14
The Link Layer . . . . .	15
What You've Learned . . . . .	15

<b>2</b>	
<b>RESOURCE LOCATION AND TRAFFIC ROUTING</b>	<b>17</b>

The Internet Protocol . . . . .	18
IPv4 Addressing . . . . .	18
Network and Host IDs . . . . .	19
Subdividing IPv4 Addresses into Subnets . . . . .	20
Ports and Socket Addresses . . . . .	23
Network Address Translation . . . . .	24
Unicasting, Multicasting, and Broadcasting . . . . .	25
Resolving the MAC Address to a Physical Network Connection . . . . .	26
IPv6 Addressing . . . . .	26
Writing IPv6 Addresses . . . . .	26
IPv6 Address Categories . . . . .	28
Advantages of IPv6 over IPv4 . . . . .	30
The Internet Control Message Protocol . . . . .	31
Internet Traffic Routing . . . . .	32
Routing Protocols . . . . .	33
The Border Gateway Protocol . . . . .	34

Name and Address Resolution . . . . .	34
Domain Name Resource Records . . . . .	35
Multicast DNS . . . . .	40
Privacy and Security Considerations of DNS Queries . . . . .	40
What You’ve Learned . . . . .	41

## **PART II: SOCKET-LEVEL PROGRAMMING 43**

### **3 RELIABLE TCP DATA STREAMS 45**

What Makes TCP Reliable? . . . . .	46
Working with TCP Sessions . . . . .	46
Establishing a Session with the TCP Handshake . . . . .	47
Acknowledging Receipt of Packets by Using Their Sequence Numbers. . . . .	47
Receive Buffers and Window Sizes . . . . .	48
Gracefully Terminating TCP Sessions . . . . .	50
Handling Less Graceful Terminations . . . . .	51
Establishing a TCP Connection by Using Go’s Standard Library. . . . .	51
Binding, Listening for, and Accepting Connections . . . . .	51
Establishing a Connection with a Server . . . . .	53
Implementing Deadlines. . . . .	62
What You’ve Learned . . . . .	71

### **4 SENDING TCP DATA 73**

Using the net.Conn Interface . . . . .	74
Sending and Receiving Data . . . . .	74
Reading Data into a Fixed Buffer. . . . .	74
Delimited Reading by Using a Scanner . . . . .	76
Dynamically Allocating the Buffer Size. . . . .	79
Handling Errors While Reading and Writing Data . . . . .	86
Creating Robust Network Applications by Using the io Package . . . . .	87
Proxying Data Between Connections . . . . .	87
Monitoring a Network Connection . . . . .	93
Pinging a Host in ICMP-Filtered Environments . . . . .	96
Exploring Go’s TCPConn Object . . . . .	98
Controlling Keepalive Messages . . . . .	99
Handling Pending Data on Close . . . . .	99
Overriding Default Receive and Send Buffers . . . . .	100
Solving Common Go TCP Network Problems. . . . .	101
Zero Window Errors . . . . .	101
Sockets Stuck in the CLOSE_WAIT State. . . . .	102
What You’ve Learned . . . . .	102

### **5 UNRELIABLE UDP COMMUNICATION 105**

Using UDP: Simple and Unreliable . . . . .	106
Sending and Receiving UDP Data . . . . .	107
Using a UDP Echo Server. . . . .	107

Receiving Data from the Echo Server . . . . .	109
Every UDP Connection Is a Listener . . . . .	110
Using net.Conn in UDP . . . . .	113
Avoiding Fragmentation . . . . .	115
What You've Learned . . . . .	117

## **6 ENSURING UDP RELIABILITY 119**

Reliable File Transfers Using TFTP . . . . .	119
TFTP Types . . . . .	120
Read Requests . . . . .	121
Data Packets . . . . .	124
Acknowledgments . . . . .	128
Handling Errors . . . . .	129
The TFTP Server . . . . .	131
Writing the Server Code . . . . .	131
Handling Read Requests . . . . .	132
Starting the Server . . . . .	135
Downloading Files over UDP . . . . .	135
What You've Learned . . . . .	139

## **7 UNIX DOMAIN SOCKETS 141**

What Are Unix Domain Sockets? . . . . .	142
Binding to Unix Domain Socket Files . . . . .	143
Changing a Socket File's Ownership and Permissions . . . . .	143
Understanding Unix Domain Socket Types . . . . .	144
Writing a Service That Authenticates Clients . . . . .	154
Requesting Peer Credentials . . . . .	154
Writing the Service . . . . .	156
Testing the Service with Netcat . . . . .	159
What You've Learned . . . . .	160

# **PART III: APPLICATION-LEVEL PROGRAMMING 163**

## **8 WRITING HTTP CLIENTS 165**

Understanding the Basics of HTTP . . . . .	166
Uniform Resource Locators . . . . .	166
Client Resource Requests . . . . .	167
Server Responses . . . . .	170
From Request to Rendered Page . . . . .	172
Retrieving Web Resources in Go . . . . .	173
Using Go's Default HTTP Client . . . . .	173
Closing the Response Body . . . . .	174
Implementing Time-outs and Cancellations . . . . .	176
Disabling Persistent TCP Connections . . . . .	178

Posting Data over HTTP . . . . .	179
Posting JSON to a Web Server . . . . .	179
Posting a Multipart Form with Attached Files . . . . .	181
What You've Learned . . . . .	184

## 9

### **BUILDING HTTP SERVICES 187**

The Anatomy of a Go HTTP Server . . . . .	188
Clients Don't Respect Your Time . . . . .	191
Adding TLS Support . . . . .	192
Handlers . . . . .	193
Test Your Handlers with httptest . . . . .	195
How You Write the Response Matters . . . . .	196
Any Type Can Be a Handler . . . . .	198
Injecting Dependencies into Handlers . . . . .	200
Middleware . . . . .	202
Timing Out Slow Clients . . . . .	203
Protecting Sensitive Files . . . . .	204
Multiplexers . . . . .	207
HTTP/2 Server Pushes . . . . .	209
Pushing Resources to the Client . . . . .	210
Don't Be Too Pushy . . . . .	214
What You've Learned . . . . .	215

## 10

### **CADDY: A CONTEMPORARY WEB SERVER 217**

What Is Caddy? . . . . .	218
Let's Encrypt Integration . . . . .	218
How Does Caddy Fit into the Equation? . . . . .	219
Retrieving Caddy . . . . .	219
Downloading Caddy . . . . .	219
Building Caddy from Source Code . . . . .	220
Running and Configuring Caddy . . . . .	220
Modifying Caddy's Configuration in Real Time . . . . .	222
Storing the Configuration in a File . . . . .	224
Extending Caddy with Modules and Adapters . . . . .	224
Writing a Configuration Adapter . . . . .	225
Writing a Restrict Prefix Middleware Module . . . . .	226
Injecting Your Module into Caddy . . . . .	231
Reverse-Proxying Requests to a Backend Web Service . . . . .	232
Creating a Simple Backend Web Service . . . . .	232
Setting Up Caddy's Configuration . . . . .	234
Adding a Reverse-Proxy to Your Service . . . . .	235
Serving Static Files . . . . .	236
Checking Your Work . . . . .	236
Adding Automatic HTTPS . . . . .	237
What You've Learned . . . . .	238

<b>11</b>		
<b>SECURING COMMUNICATIONS WITH TLS</b>		<b>241</b>
A Closer Look at Transport Layer Security . . . . .		242
Forward Secrecy . . . . .		243
In Certificate Authorities We Trust . . . . .		243
How to Compromise TLS . . . . .		244
Protecting Data in Transit . . . . .		245
Client-side TLS . . . . .		245
TLS over TCP . . . . .		247
Server-side TLS . . . . .		249
Certificate Pinning . . . . .		252
Mutual TLS Authentication . . . . .		255
Generating Certificates for Authentication . . . . .		256
Implementing Mutual TLS . . . . .		259
What You've Learned . . . . .		265

## **PART IV: SERVICE ARCHITECTURE** **267**

<b>12</b>		
<b>DATA SERIALIZATION</b>		<b>269</b>
Serializing Objects . . . . .		270
JSON . . . . .		276
Gob . . . . .		278
Protocol Buffers . . . . .		280
Transmitting Serialized Objects . . . . .		284
Connecting Services with gRPC . . . . .		284
Creating a TLS-Enabled gRPC Server . . . . .		286
Creating a gRPC Client to Test the Server . . . . .		289
What You've Learned . . . . .		294

<b>13</b>		
<b>LOGGING AND METRICS</b>		<b>295</b>
Event Logging . . . . .		296
The log Package . . . . .		297
Leveled Log Entries . . . . .		300
Structured Logging . . . . .		301
Scaling Up with Wide Event Logging . . . . .		312
Log Rotation with Lumberjack . . . . .		315
Instrumenting Your Code . . . . .		316
Setup . . . . .		317
Counters . . . . .		317
Gauges . . . . .		318
Histograms and Summaries . . . . .		319
Instrumenting a Basic HTTP Server . . . . .		320
What You've Learned . . . . .		326

<b>14</b>		
<b>MOVING TO THE CLOUD</b>		<b>329</b>
Laying Some Groundwork . . . . .		330
AWS Lambda . . . . .		333
Installing the AWS Command Line Interface . . . . .		333
Configuring the CLI . . . . .		333
Creating a Role . . . . .		335
Defining an AWS Lambda Function . . . . .		336
Compiling, Packaging, and Deploying Your Function . . . . .		339
Testing Your AWS Lambda Function . . . . .		340
Google Cloud Functions . . . . .		341
Installing the Google Cloud Software Development Kit . . . . .		341
Initializing the Google Cloud SDK . . . . .		341
Enable Billing and Cloud Functions . . . . .		342
Defining a Cloud Function . . . . .		342
Deploying Your Cloud Function . . . . .		344
Testing Your Google Cloud Function . . . . .		345
Azure Functions . . . . .		346
Installing the Azure Command Line Interface . . . . .		346
Configuring the Azure CLI . . . . .		347
Installing Azure Functions Core Tools . . . . .		347
Creating a Custom Handler . . . . .		348
Defining a Custom Handler . . . . .		349
Locally Testing the Custom Handler . . . . .		350
Deploying the Custom Handler . . . . .		351
Testing the Custom Handler . . . . .		353
What You've Learned . . . . .		353
<b>INDEX</b>		<b>355</b>