`net2`
Technical Specification

Eric Griffis
dedbox@gmail.com

November 30, 2017

Abstract

This is a technical specification for `net2`, a networking abstraction for URL-addressable agents communicating via byte streams. Its purpose is to drive the engineering design process.

Contents

List of Figures

Chapter 1

Introduction

This report defines an abstract model for `net2`.

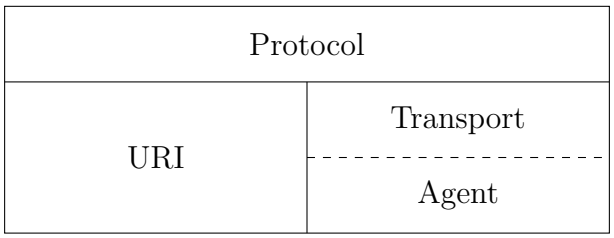

Figure 1.1: The `net2` API stack

Chapter 2

URI

- Explain how URIs work in `net2`.

  "A URI has several parts. The authority defines *who* you are talking to. The scheme defines *how* you talk to them. The path and query define *what resource* you're talking to them about."

- We roll our own URI sub-system.

$\mathcal{S}$ = schemes $\quad$ $\mathcal{A}$ = authorities $\quad$ $\mathcal{P}$ = paths $\quad$ $\mathcal{Q}$ = queries

$\mathsf{scheme} : \mathcal{U} \to \mathcal{S}$ $\quad$ $\mathsf{authority} : \mathcal{U} \to \mathcal{A}$ $\quad$ $\mathsf{path} : \mathcal{U} \to \mathcal{P}$ $\quad$ $\mathsf{query} : \mathcal{U} \to \mathcal{Q}$

Chapter 3

Transport

A *byte stream* is a one-way communications channel. A *connection* is a pair of opposing byte streams.

3.1 Primitive types

References

$\mathcal{L}$ = listeners $\qquad$ $\mathcal{T}$ = transports $\qquad$ $\mathcal{R} = \mathcal{L} \cup \mathcal{T}$

A *reference* is an opaque token that represents a portion of run-time state. A listener represents a connection request queue. A transport represents a connection.

Ports

$\mathcal{I}$ = input ports $\qquad$ $\mathcal{O}$ = output ports

A *port* is a host platform object that represents one end of a byte stream. Ports come in pairs—an input port and an output port. An output port sends bytes to the byte stream. An input port receives bytes from the byte stream.

Literals

$\mathcal{B}$ = byte arrays $\qquad$ $\varnothing$ = void

A *literal* is a fixed unit of data. A byte array is a unit of data exchange. The void literal is returned by operations with a side effect and no useful result.

$[k \mapsto v]\mathrm{D}$	associate k with v in D.
$\mathrm{D} \setminus \{k \mapsto \cdot\}$	remove from D the association keyed by k.
$\mathrm{D}(k)$	lookup k in D.

Figure 3.1: Dictionary notation

3.2 Dictionaries

$$\mathcal{D} = \{\mathcal{R} \to *\}$$

A *dictionary* is an associative array. Dictionaries associate references to some underlying data. Associations can be added, removed, or looked up.

3.3 Run-time state

$$\mathrm{L} : \mathcal{L} \to \mathcal{A} \qquad\qquad \mathrm{T} : \mathcal{T} \to \mathcal{A} \times \mathcal{A} \times \mathcal{I} \times \mathcal{O}$$

The run-time state is a set of dictionaries. All side effects occur during operations on dictionaries in the run-time state.

L records the addresses of listeners. Given listener ℓ and local authority authority a_L, the host platform queues requests to connect to a_L under ℓ for as long as $\mathrm{L}(\ell) = a_L$.

T records the addresses and ports of established connections. Given transport τ, addresses a_L, a_R, and ports p_I, p_O, the host platform establishes a connection between local authority a_L and remote authority a_R for as long as $\mathrm{T}(\tau) = (a_L, a_R, p_I, p_O)$. Updating p_I or p_O will exchange bytes over the connection.

3.4 The Agent API

A registered name driver implements the operations defined in this section.

$$\text{listener} : \mathcal{A} \to \mathcal{L} \qquad\qquad \text{listener}(a_L) = \ell$$

Creates a reference ℓ to a connection request queue associated with local authority a_L.

$$\text{accepter} : \mathcal{L} \to \mathcal{T} \times \mathcal{A} \times \mathcal{I} \times \mathcal{O} \qquad\qquad \text{accepter}(\ell) = \langle \tau, a_R, p_I, p_O \rangle$$

Creates a reference τ to a connection request from remote authority a_R queued under listener ℓ. Opens ports p_I, p_O for exchanging bytes over the connection.

$t ::=$	$\mathsf{listen}(t)$	bind authority	$v ::=$	a	authority
$\mid$	$\mathsf{accept}(t)$	accept connection	$\mid$	b	byte array
$\mid$	$\mathsf{connect}(t)$	connect to authority	$\mid$	ℓ	listener
$\mid$	$\mathsf{release}(t)$	unbind / disconnect	$\mid$	τ	transport
$\mid$	$\mathsf{send}(t, t)$	send bytes	$\mid$	u	URI
$\mid$	$\mathsf{receive}(t)$	receive bytes	$\mid$	$\varnothing$	void

Figure 3.2: Transport syntax

$$\text{connector} : \mathcal{A} \to \mathcal{T} \times \mathcal{A} \times \mathcal{I} \times \mathcal{O} \qquad \text{connector}(a_R) = \langle \tau, a_L, p_I, p_O \rangle$$

Creates a reference τ to a connection from local authority a_L to remote authority a_R. Opens ports p_I, p_O for exchanging bytes over the connection.

$$\text{sender} : \mathcal{B} \times \mathcal{O} \to \mathcal{O} \qquad \text{sender}(b, p_O) = p'_O$$

Creates a port update p'_O that writes byte array b to port p_O.

$$\text{receiver} : \mathcal{I} \to \mathcal{B} \times \mathcal{I} \qquad \text{receiver}(p_I) = \langle b, p'_I \rangle$$

Creates a port update p'_I that reads byte array b from port p_I.

3.5 Creating and destroying connections

$$\mathsf{listen} : \mathcal{A} \to \mathcal{L} \qquad \mathsf{listen}(a_L) = \ell \qquad \frac{\text{listener}(a_L) = \ell}{\mathrm{L} \vdash \mathsf{listen}(a_L) \rightsquigarrow [\ell \mapsto a_L]\mathrm{L} \vdash \ell}\ \textsc{Lsn}$$

Creates a listener ℓ on local authority a_L.

$$\mathsf{accept} : \mathcal{L} \to \mathcal{T} \qquad \mathsf{accept}(\ell) = \tau \qquad \frac{\mathrm{L}(\ell) = u_L \qquad \text{accepter}(\ell) = \langle \tau, a_R, p_I, p_O \rangle}{\mathrm{T} \vdash \mathsf{accept}(\ell) \rightsquigarrow [\tau \mapsto \langle a_L, a_R, p_I, p_O \rangle]\mathrm{T} \vdash \tau}\ \textsc{Acc}$$

Accepts a transport τ from listener ℓ.

$$\mathsf{connect} : \mathcal{A} \to \mathcal{T} \qquad \mathsf{connect}(a) = \tau \qquad \frac{\text{connector}(a_R) = \langle \tau, a_L, p_I, p_O \rangle}{\mathrm{T} \vdash \mathsf{connect}(a_R) \rightsquigarrow [\tau \mapsto \langle a_L, a_R, p_I, p_O \rangle]\mathrm{T} \vdash \tau}\ \textsc{Con}$$

Connects a transport τ from local authority a_L to remote authority a_R.

$$\mathsf{release} : \mathcal{R} \to \varnothing \qquad \mathsf{release}(r) = \varnothing \qquad \frac{}{\mathrm{L} \vdash \mathsf{release}(r) \rightsquigarrow \mathrm{L} \setminus \{r \mapsto \cdot\} \vdash \varnothing}\ \textsc{Rls}$$

Stops listening when r is a listener. Closes the connection when r is a transport.

3.6 Exchanging bytes

$$\mathsf{send} : \mathcal{B} \times \mathcal{T} \to \varnothing \qquad \mathsf{send}(b, \tau) = \varnothing \qquad \frac{\mathrm{T}(\tau) = \langle a_L, a_R, p_I, p_O \rangle \qquad \mathrm{sender}(b, p_O) = p'_O}{\mathrm{T} \vdash \mathsf{send}(b, \tau) \rightsquigarrow [\tau \mapsto \langle a_L, a_R, p_I, p'_O \rangle]\mathrm{T} \vdash \varnothing}\ \textsc{Snd}$$

Sends byte array b over transport τ.

$$\mathsf{receive} : \mathcal{T} \to \mathcal{B} \qquad \mathsf{receive}(\tau) = b \qquad \frac{\mathrm{T}(\tau) = \langle a_L, a_R, p_I, p_O \rangle \qquad \mathrm{receiver}(p_I) = \langle b, p'_I \rangle}{\mathrm{T} \vdash \mathsf{receive}(\tau) \rightsquigarrow [\tau \mapsto \langle a_L, a_R, p'_I, p_O \rangle]\mathrm{T} \vdash b}\ \textsc{Rcv}$$

Receives byte array b over transport τ.

Chapter 4

Protocol

4.1 Codecs

A *codec* is a composable and invertible function for structured data serialization and other restructuring operations. A *frame* is a byte array of computable length, and *framing* is the act of assembling bytes into frames. Invertible framing functions are codecs.

$$\mathcal{C} = \bigcup_{X,Y} \{X \leftrightarrow Y\} \qquad \mathcal{F} = \bigcup_{X} \{X \leftrightarrow \mathcal{B}\}$$

$$\mathsf{decode} : \mathcal{F} \times \mathcal{T} \to X \qquad \mathsf{decode}(f, \tau) = x \qquad \frac{\mathrm{T} \vdash \mathsf{receive}(\tau) \rightsquigarrow \mathrm{T}' \vdash b \qquad f^{-1}(b) = x}{\mathrm{T} \vdash \mathsf{decode}(f, \tau) \rightsquigarrow \mathrm{T}' \vdash x}\ \textsc{Dec}$$

$$\mathsf{encode} : X \times \mathcal{F} \times \mathcal{T} \to \varnothing \qquad \mathsf{encode}(x, f, \tau) = \varnothing \qquad \frac{f(x) = b}{\mathsf{encode}(x, f, \tau) \rightsquigarrow \mathsf{send}(b, \tau)}\ \textsc{Enc}$$

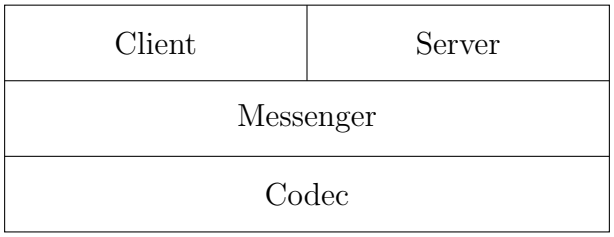

Figure 4.1: The Protocol API stack

Primitive codecs

Codec composition

Composite codecs

4.2 Messengers

Primitive types

$$\mathcal{M} = \text{messengers}$$

Run-time state

$$\mathrm{M} : \mathcal{M} \to \mathcal{C} \times \mathcal{T}$$

Creating and destroying messengers

$$\mathsf{messenger} : \mathcal{F} \times \mathcal{T} \to \mathcal{M} \qquad\qquad \mathsf{messenger}(f, t) = m$$

$$\frac{}{\mathrm{M} \vdash \mathsf{messenger}(f, t) \rightsquigarrow [m \mapsto \langle f, t \rangle]\mathrm{M} \vdash m}\ \textsc{Msn}$$

$$\frac{\mathrm{M}(m) = \langle f, t \rangle \qquad \mathrm{T} \vdash \mathsf{release}(t) \rightsquigarrow \mathrm{T}' \vdash \varnothing}{\mathrm{M}, \mathrm{T} \vdash \mathsf{release}(m) \rightsquigarrow \mathrm{M} \setminus \{m \mapsto \cdot\}, \mathrm{T}' \vdash \varnothing}\ \textsc{RlsM}$$

Reading and writing messages

$$\mathsf{read} : \mathcal{M} \to \mathcal{B} \qquad \mathsf{read}(m) = b \qquad \frac{\mathrm{M}(m) = \langle f, t \rangle}{\mathsf{read}(m) \rightsquigarrow \mathsf{decode}(f, t)}\ \textsc{Rd}$$

$$\mathsf{write} : \mathcal{B} \times \mathcal{M} \to \varnothing \qquad \mathsf{write}(b, m) = \varnothing \qquad \frac{\mathrm{M}(m) = \langle f, t \rangle}{\mathsf{write}(x, m) \rightsquigarrow \mathsf{encode}(x, f, t)}\ \textsc{Wr}$$

4.3 Clients and servers

Glossary

agent A URL-addressable process capable of exchanging bytes.

authority The authority component of a URI. This could be an IP address and port number, or other kinds of extensible registered names [BLFM14].

byte array A finite sequence of bytes.

byte stream A one-way communications channel.

connection A two-way communications channel.

connector A means of requesting a connection to another agent.

dictionary A binary relation between references and run-time state.

host platform The programming platform implementing `net2`.

input port A port that receives bytes.

listener A means of accepting connection requests from other agents.

output port A port that sends bytes.

port One end of a byte stream.

reference An opaque token that identifies a set of related objects.

scheme The scheme component of a URI.

transport A reliable, buffered, and ordered means of exchanging bytes with other agents.

URL A URI, as defined in RFC 3986 [BLFM14], that locates an agent.

References

[BLFM14] Tim Berners-Lee, Roy Fielding, and Larry Masinter. Rfc 3986, uniform resource identifier (uri): Generic syntax, 2005. *URL: http://www. faqs. org/rfcs/rfc3986. html*, 2014.

License

"Work" shall mean the work of authorship, whether in Source or Object form, made available under the License, as indicated by a copyright notice that is included in or attached to the work (an example is provided in the Appendix below).

"Derivative Works" shall mean any work, whether in Source or Object form, that is based on (or derived from) the Work and for which the editorial revisions, annotations, elaborations, or other modifications represent, as a whole, an original work of authorship. For the purposes of this License, Derivative Works shall not include works that remain separable from, or merely link (or bind by name) to the interfaces of, the Work and Derivative Works thereof.

"Contribution" shall mean any work of authorship, including the original version of the Work and any modifications or additions to that Work or Derivative Works thereof, that is intentionally submitted to Licensor for inclusion in the Work by the copyright owner or by an individual or Legal Entity authorized to submit on behalf of the copyright owner. For the purposes of this definition, "submitted" means any form of electronic, verbal, or written communication sent to the Licensor or its representatives, including but not limited to communication on electronic mailing lists, source code control systems, and issue tracking systems that are managed by, or on behalf of, the Licensor for the purpose of discussing and improving the Work, but excluding communication that is conspicuously marked or otherwise designated in writing by the copyright owner as "Not a Contribution."

"Contributor" shall mean Licensor and any individual or Legal Entity on behalf of whom a Contribution has been received by Licensor and subsequently incorporated within the Work.

2. Grant of Copyright License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable copyright license to reproduce, prepare Derivative Works of, publicly display, publicly perform, sublicense, and distribute the Work and such Derivative Works in Source or Object form.

3. Grant of Patent License. Subject to the terms and conditions of this License, each Contributor hereby grants to You a perpetual, worldwide, non-exclusive, no-charge, royalty-free, irrevocable (except as stated in this section) patent license to make, have made, use, offer to sell, sell, import, and otherwise transfer the Work, where such license applies only to those patent claims licensable

by such Contributor that are necessarily infringed by their Contribution(s) alone or by combination of their Contribution(s) with the Work to which such Contribution(s) was submitted. If You institute patent litigation against any entity (including a cross-claim or counterclaim in a lawsuit) alleging that the Work or a Contribution incorporated within the Work constitutes direct or contributory patent infringement, then any patent licenses granted to You under this License for that Work shall terminate as of the date such litigation is filed.

4. Redistribution. You may reproduce and distribute copies of the Work or Derivative Works thereof in any medium, with or without modifications, and in Source or Object form, provided that You meet the following conditions:

   (a) You must give any other recipients of the Work or Derivative Works a copy of this License; and

   (b) You must cause any modified files to carry prominent notices stating that You changed the files; and

   (c) You must retain, in the Source form of any Derivative Works that You distribute, all copyright, patent, trademark, and attribution notices from the Source form of the Work, excluding those notices that do not pertain to any part of the Derivative Works; and

   (d) If the Work includes a "NOTICE" text file as part of its distribution, then any Derivative Works that You distribute must include a readable copy of the attribution notices contained within such NOTICE file, excluding those notices that do not pertain to any part of the Derivative Works, in at least one of the following places: within a NOTICE text file distributed as part of the Derivative Works; within the Source form or documentation, if provided along with the Derivative Works; or, within a display generated by the Derivative Works, if and wherever such third-party notices normally appear. The contents of the NOTICE file are for informational purposes only and do not modify the License. You may add Your own attribution notices within Derivative Works that You distribute, alongside or as an addendum to the NOTICE text from the Work, provided that such additional attribution notices cannot be construed as modifying the License.

   You may add Your own copyright statement to Your modifications and

may provide additional or different license terms and conditions for use, reproduction, or distribution of Your modifications, or for any such Derivative Works as a whole, provided Your use, reproduction, and distribution of the Work otherwise complies with the conditions stated in this License.

5. Submission of Contributions. Unless You explicitly state otherwise, any Contribution intentionally submitted for inclusion in the Work by You to the Licensor shall be under the terms and conditions of this License, without any additional terms or conditions. Notwithstanding the above, nothing herein shall supersede or modify the terms of any separate license agreement you may have executed with Licensor regarding such Contributions.

6. Trademarks. This License does not grant permission to use the trade names, trademarks, service marks, or product names of the Licensor, except as required for reasonable and customary use in describing the origin of the Work and reproducing the content of the NOTICE file.

7. Disclaimer of Warranty. Unless required by applicable law or agreed to in writing, Licensor provides the Work (and each Contributor provides its Contributions) on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied, including, without limitation, any warranties or conditions of TITLE, NON-INFRINGEMENT, MERCHANTABILITY, or FITNESS FOR A PARTICULAR PURPOSE. You are solely responsible for determining the appropriateness of using or redistributing the Work and assume any risks associated with Your exercise of permissions under this License.

8. Limitation of Liability. In no event and under no legal theory, whether in tort (including negligence), contract, or otherwise, unless required by applicable law (such as deliberate and grossly negligent acts) or agreed to in writing, shall any Contributor be liable to You for damages, including any direct, indirect, special, incidental, or consequential damages of any character arising as a result of this License or out of the use or inability to use the Work (including but not limited to damages for loss of goodwill, work stoppage, computer failure or malfunction, or any and all other commercial damages or losses), even if such Contributor has been advised of the possibility of such damages.

9. Accepting Warranty or Additional Liability. While redistributing the Work or Derivative Works thereof, You may choose to offer, and charge a fee for, acceptance of support, warranty, indemnity, or other liability obligations and/or rights consistent with this

License. However, in accepting such obligations, You may act only on Your own behalf and on Your sole responsibility, not on behalf of any other Contributor, and only if You agree to indemnify, defend, and hold each Contributor harmless for any liability incurred by, or claims asserted against, such Contributor by reason of your accepting any such warranty or additional liability.

END OF TERMS AND CONDITIONS

APPENDIX: How to apply the Apache License to your work.

To apply the Apache License to your work, attach the following boilerplate notice, with the fields enclosed by brackets "[]" replaced with your own identifying information. (Don't include the brackets!) The text should be enclosed in the appropriate comment syntax for the file format. We also recommend that a file or class name and description of purpose be included on the same "printed page" as the copyright notice for easier identification within third-party archives.

Copyright [yyyy] [name of copyright owner]

Licensed under the Apache License, Version 2.0 (the "License");
you may not use this file except in compliance with the License.
You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software
distributed under the License is distributed on an "AS IS" BASIS,
WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied.
See the License for the specific language governing permissions and
limitations under the License.