Reticulum: What’s Next?

Building the Future After the Founder Steps Back

Liam

2026-02-02

FOSDEM 26 – BOF/Unconference

1. What is Reticulum?

1.1 Cryptographic Mesh Networking

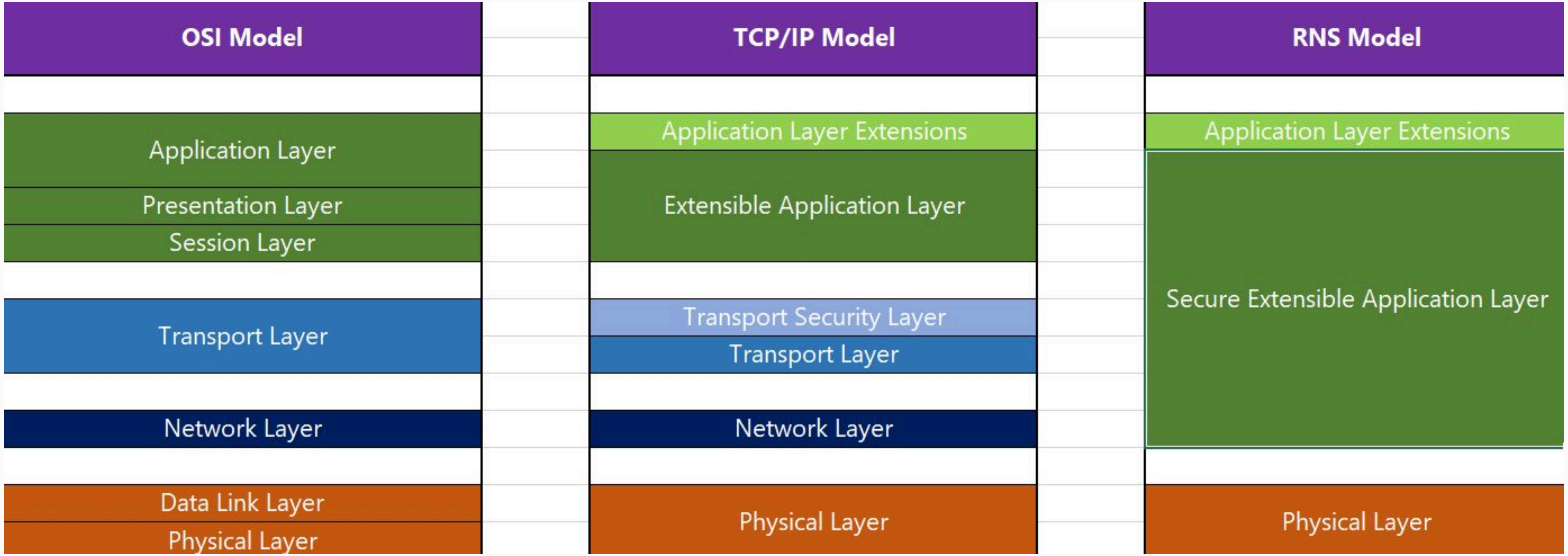
A protocol stack for building resilient networks without infrastructure.

- Transport-agnostic (LoRa, TCP, UDP, I2P, Packet Radio)
- Encryption by default (X25519, AES-128)
- No central authority (no IP, no DNS)
- Destinations are cryptographic hashes, not addresses
- Works from 5 bits/second to gigabit

Not an application framework. An alternative to `socket.h`.

reticulum.network

1.2 Protocol Stack



OSI Model	TCP/IP Model	RNS Model
Application Layer	Application Layer Extensions	Application Layer Extensions
	Extensible Application Layer	Secure Extensible Application Layer
Presentation Layer		
Session Layer		
Transport Layer	Transport Security Layer	
	Transport Layer	
Network Layer	Network Layer	
Data Link Layer	Physical Layer	Physical Layer
Physical Layer		

Figure 1: TCP/IP vs Reticulum: Collapsing complexity into cryptographic simplicity[1]

[1] Source https://linuxinabit.codeberg.page/blog/reticulum/

2. 2025: Year of Changes

2.1 Timeline

April 2025	License changed – anti-AI & “no Harm” clause added
Summer 2025	Mark Qvist takes a break
Fall 2025	RNS 1.0.0 released
December 2025	“Carrier Switch” – withdrawal from public engagement
January 2026	Development continues, but no public interaction

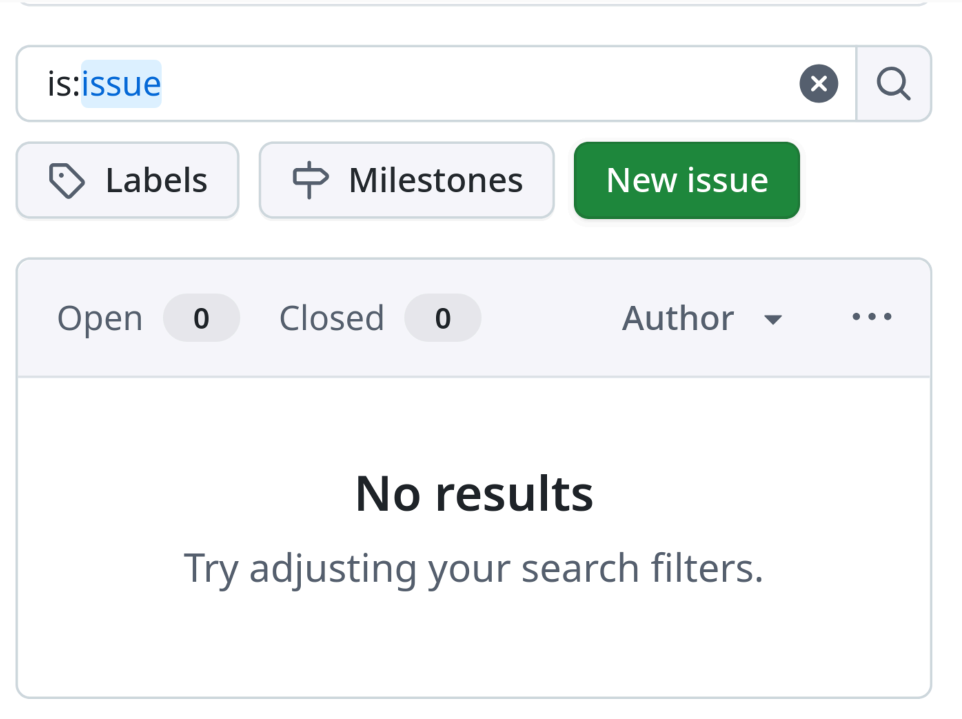
The license change prevents distribution in Debian, F-Droid, Alpine main, and most package managers. The project cannot receive funding through standard open source grants. Without any practical or concrete benefits (License would not stand in court).

2.2 What We Lost

Closed Channels

- Issue tracker hidden (years of technical discussion gone)
- No way to report bugs
- No way to submit patches
- No community support channels
- Documentation frozen

Institutional knowledge is disappearing. Contributors with fixes have nowhere to send them.



is:issue

No results

Try adjusting your search filters.

Issues hidden after archival

2.3 What We Still Have

The Foundation Remains

- Protocol is public domain
- Code is open source (with license restrictions)
- Multiple implementations exist
- Community is organizing independently

> *"If you want Reticulum to continue evolving, you have the power to make that happen. The protocol is public domain. The code is open source. Everything you need is right here."*
>
> — Mark Qvist, December 2025

3. The Ecosystem Today

3.1 Implementation Matrix

Implementation	Language	Status	License	Maintainer
RNS (original)	Python	1.0+	MIT + anti-AI	markqvist (inactive)
RetiNet	Python	1.0 compat	AGPL-3.0	Community
Reticulum-rs	Rust	Active dev	MIT	Beechat
microReticulum	C++	Transport only	MIT	attermann
go-reticulum	Go	WASM working	MIT	quad4
Zig impl	Zig	Early	—	Research

Table 1: Current implementation landscape

3.2 RetiNet: The AGPL Fork

Fork created before license change. Fully RNS 1.0 compatible.

- AGPL license (F-Droid compatible)
- Open contribution model
- Zulip chat with threads (finally!)
- Drop-in replacement for RNS

Today’s goal: Can we switch our setups to RetiNet?

RetiNet on Codeberg

3.3 Reticulum-rs: Rust Implementation

Beechat's implementation, most mature non-Python version.

- TCP, UDP interfaces working
- My contribution: `rnsd-rs` daemon
- Running as my transport node for 3+ weeks
- Goal: upstream collaboration, package for distributions

Talk: Sunday 15:35, UD2.218A

rnsd-rs daemon branch

3.4 The Config Problem

Python Reticulum uses `configobj`, a Python-only config parser. The format looks like TOML but isn’t valid TOML.

Before (configobj):

```
[reticulum]
enable_transport = No
share_instance = Yes
shared_instance_port = 37428

[interfaces]
[[Default TCP Server]]
type = "TCPServerInterface"
interface_enabled = Yes
bind_host = "[::]"
bind_port = 4242
```

After (valid TOML):

```
[reticulum]
enable_transport = false
share_instance = true
shared_instance_port = 37428

[[interfaces]]
name = "Default TCP Server"
type = "TCPServerInterface"
interface_enabled = true
bind_host = "[::]"
bind_port = 4242
```

3.5 Config Converter

Convert your existing RNS config to valid TOML:

- Creates backup automatically
- Preserves comments
- Fixes Yes/No ⟶ true/false
- Fixes `[[Section Name]]` ⟶ `[[interfaces]]` + `name =`

Goal: Configs portable between all implementations.

```
cargo run --example convert_config \
  -- ~/.reticulum/config
```

3.6 Performance

CPU Usage Comparison

Python rnsd: 20-100% spikes

Rust rnsd-rs: $< 4\%$ sustained

Same transport node workload

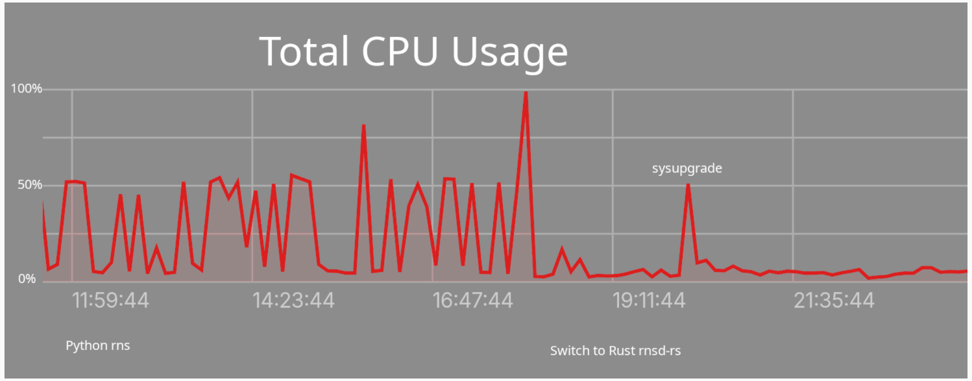


Figure 2: CPU Usage Python vs Rust

3.7 microReticulum: Embedded C++

The only implementation that runs standalone on microcontrollers.

- ESP32, nRF52 targets
- Currently transport/repeater only (no links)
- Development is slow but steady
- Completing this may be easier than porting Rust to no_std

Vision: RNodes that create and process packets without a tethered host.

microReticulum

3.8 Go Implementation + WASM

Reticulum running in the browser via WebAssembly.

Possibilities:

- Local-first apps with mesh fallback
- BLE via Web Bluetooth API
- RNodes via WebUSB
- PWA caches everything offline

Not a fan of browser-as-OS, but: zero install barrier.

Reticulum-Go

4. Columba: Let’s Build a Mesh

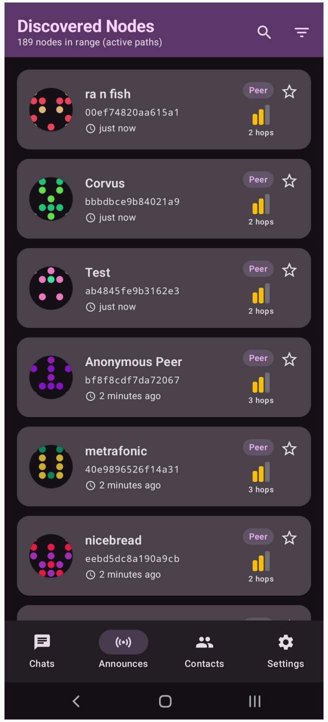
4.1 Native Android Client

Material Design 3 replacement for Sideband.

Features:

- Messaging, files, photos over LXMF
- LXST voice calls (Codec2)
- Location sharing with telemetry
- RNode support (USB + Bluetooth)
- Interface discovery (RNS 1.1.x)
- Identity backup/restore

Built with native Android, not Python. Actually responsive on older phones.



Discovered Nodes
189 nodes in range (active paths)

ra n fish
00ef74820aa615a1
just now
Peer
2 hops

Corvus
bbbdbce9b84021a9
just now
Peer
2 hops

Test
ab4845fe9b3162e3
just now
Peer
2 hops

Anonymous Peer
bf8f8cdf7da72067
2 minutes ago
Peer
3 hops

metrafonic
40e9896526f14a31
2 minutes ago
Peer
3 hops

nicebread
eebd5dc8a190a9cb
2 minutes ago
Peer
2 hops

4.2 Install Now

Columba v0.7.4-beta

Signed APK available

Verification instructions in release notes

What’s New in RC:

- LXST Voice Calls
- RNS 1.1.x Interface Discovery
- RNode via USB cable
- Bluetooth pairing over serial
- Telemetry collector mode

4.3 FOSDEM Network Config

TCP Transport (Internet fallback):

```
[[interfaces]]
name = "ChaosNet TCP Interface"
type = "TCPClientInterface"
interface_enabled = true
target_host = "rns.c3.jitter.eu"
target_port = 4242
```

LoRa Config (RNode):

```
[[interfaces]]
name = "FOSDEM LoRa"
type = "RNodeInterface"
interface_enabled = true
port = "/dev/ttyUSB0"
frequency = 864200000
bandwidth = 125000
txpower = 5
spreadingfactor = 10
codingrate = 6
```

Band	Frequency	BW	SF
EU868	864.2 MHz	125 kHz	10

Interactive Session

1. Install Columba

Scan the QR code

Install the APK

2. Find Your Neighbor

Check "Announces" tab

Send a message

3. Report Issues

What broke?

What's confusing?

Call the Stage via LXST

Destination: `[YOUR_HASH_HERE]`

First successful call gets bragging rights

5. What’s Missing

5.1 Adoption Barriers

Distribution Problem

- Cannot package in Debian (license)
- Cannot distribute on F-Droid (license)
- Cannot fund through standard grants (license)
- Alpine has py3-rns in testing only

Users must install manually. The barrier is too high for mainstream adoption.

Embedded Problem

- Python cannot run on MCUs
- RNodes require tethered host
- microReticulum is transport-only
- No standalone mesh nodes exist

The devices where mesh networking matters most cannot run Reticulum.

5.2 Documentation Gaps

Missing Documentation

- No beginner quick-start
- Assumes deep protocol knowledge
- Hidden issues = lost institutional knowledge
- No contribution guide
- No formal specification

The existing docs assume you're already a Reticulum expert. New users bounce off.

What We Need:

- Tiered handbook (beginner / operator / developer)
- Preserved issue archive from Internet Archive
- Clear contribution workflow
- Interoperability test documentation

See also: Jo Kroese's BOF on formal specification

5.3 Community Fragmentation

Matrix (1500+ members):

- Flood of messages
- No threading
- Hard to find information
- No moderation structure

Alternative: Zulip

- Threaded discussions
- Topic organization
- Searchable history
- Open registration

Zulip Chat

Join the FOSDEM room

6. The Path Forward

6.1 Concrete Next Steps

Technical:

- Distribution packages (Alpine, Debian) for Rust daemon
- Complete embedded implementation (microReticulum or no_std)
- Config format standardization (TOML everywhere)
- Interoperability test suite

Community:

- Tiered documentation handbook
- Contribution workflow
- Bug reporting process that works
- Archive historical GitHub issues

Validation:

- Real deployments in cellular dead zones
- Emergency communication testing
- Telemetry collection (Metrum)

Metrum

IEEE standards-based telemetry (CBOR, SenML) over Reticulum.

Environmental sensing for off-grid deployments.

https://codeberg.org/Metrum/metrum

6.2 How You Can Help

Today:

- Install Columba, find bugs
- Try switching to RetiNet
- Test the config converter
- Join the Zulip

This Week:

- Report issues on Codeberg
- Document your setup
- Share your use case

Ongoing:

- Run a transport node
- Contribute to documentation
- Test interoperability

rnsd-rs

Zulip

6.3 Related Sessions

Reticulum-rs: Python to Rust

Sunday 15:35 – UD2.218A

Beechat presenting their Rust implementation.

Covers the technical details of the port.

Formal Specification BOF

Jo Kroese's session on formalizing the Reticulum protocol.

Complementary to this session: protocol vs. tooling.

Beechat talk schedule

7. Let’s Mesh

7.1 All Links

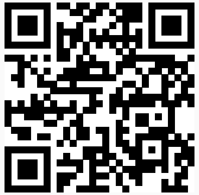
rnsd-rs

Columba

RetiNet

Zulip

microReticulum



Metrum

Beechat Rust

reticulum.network

7.2 Questions / Discussion

A healthy project must separate from its founder.

This is the most exciting time to be part of Reticulum.

The community has the opportunity to take ownership.

Contact Codeberg: lgh

Reticulum Destination

`[YOUR_HASH]`