What is Reticulum?

Goals

Reticulum is a tool for building networks. These networks can be very small (neighborhood-sized) or very large (globe-spanning).

These networks are *heterogeneous*, meaning that links between nodes can go over a diverse set of physical layers:

- TCP
- LoRa
- KISS (Packet Radio)
- QR Codes
- etc

Goals cont.

Small, heterogeneous networks can be linked together by “bridger nodes”. Imagine small neighborhood networks of LoRa-connected nodes in north and south Austin. If one node in each network has an TCP interface, and is connected to the other, the two networks are now able to talk to each other over this IP bridge.

(Exercise: draw network topology)
(Note: don’t worry about interfaces just yet)

Cryptographic Primitives

- Ed25519 for signatures
- X22519 for ECDH key exchanges
- HKDF for key derivation
- AES-256 in CBC mode
- HMAC-SHA256 for message authentication
- SHA-256
- SHA-512
- Key Ratcheting

Protocol

The Reticulum protocol has a few primitives:

- Destinations
- Nodes
- Interfaces

https://reticulum.network/manual/understanding.html

Destinations

“Reticulum does away with the idea of addresses and ports known from IP, TCP and UDP. Instead Reticulum uses the singular concept of *destinations*. Any application using Reticulum as its networking stack will need to create one or more destinations to receive data, and know the destinations it needs to send data to.

All destinations in Reticulum are *represented* as a 16 byte hash. This hash is derived from truncating a full SHA-256 hash of identifying characteristics of the destination. To users, the destination addresses will be displayed as 16 hexadecimal bytes, like this example:
`<13425ec15b621c1d928589718000d814>`”

Nodes

“Currently, Reticulum distinguishes between two types of network nodes. All nodes on a Reticulum network are *Reticulum Instances*, and some are also *Transport Nodes*. If a system running Reticulum is fixed in one place, and is intended to be kept available most of the time, it is a good contender to be a *Transport Node*.”

“Every Transport node participating in a Reticulum network will only know the most direct way to get a packet one hop closer to it’s destination.”

Routing

When a node is connected to the network, and at regular intervals, it will “announce” itself. Both its public key and identity. Transport nodes will propagate the “announce” message, plus the number of hops. Reticulum has a max hop distance of 128 (compared to 7 in Meshtastic).

Example:

Let’s say Node A is connected to a network. Node B connects to the network and announces itself. The announce travels throughout the network until it eventually reaches Node A. Node A wants to send a message to Node B. It encrypts the message with the public key and sends the message + destination to the nearest transport node. The transport node knows that Node B (the message destination) is X hops away. It checks which of its peers is (X-1) hops away from Node B, and forwards the message. This repeats until the message arrives at Node B.

Interfaces

An interface is like a network interface on a regular machine. For example, your laptop, when plugged into ethernet and also connected via wireless, will have the network interfaces “eth0” and “wlan0”.

Reticulum interfaces must be manually configured. A good exercise to get started with Reticulum is to connect to one of the public IP “Backbones”, which is just a transport node that many other people connect to. We will be connecting to the Amsterdam public backbone in this course.

RNS

RNS is a Python reference implementation of the Reticulum protocol.

In this course we will be installing and using RNS. It can be run as a daemon/service, or run as part of a Python program (it is a library).

https://github.com/markqvist/Reticulum

There are other implementations that are not fully-featured yet. But I think we will start to see the ecosystem blow up in 2026 as global surveillance and totalitarianism ramps up (Reticulum is very popular in Russia)

LXMF

LXMF is a simple message format that Reticulum uses. LXMF is like the "HTTP" of Reticulum.

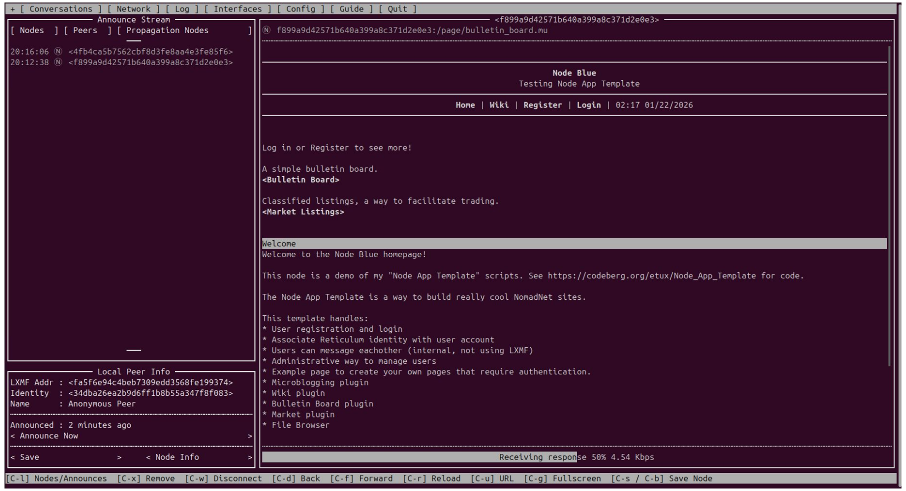
- Destination
- Source
- Ed25519 Signature
- Payload
  - Timestamp
  - Content
  - Title
  - Fields

https://github.com/markqvist/LXMF

Client Applications

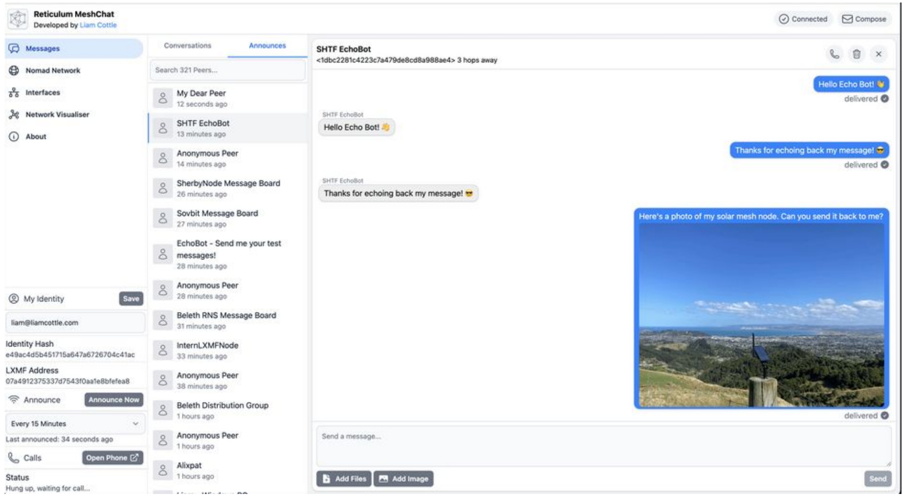
Nomadnet —

Meshchat



```
+ [ Conversations ] [ Network ] [ Log ] [ Interfaces ] [ Config ] [ Guide ] [ Quit ]
              Announce Stream
[ Nodes  ] [ Peers  ] [ Propagation Nodes          ]

20:16:06 Ⓝ <4fb4ca5b7562cbf8d3fe8aa4e3fe85f6>
20:12:38 Ⓝ <f899a9d42571b640a399a8c371d2e0e3>

              Local Peer Info
LXMF Addr : <fa5f6e94c4beb7309edd3568fe199374>
Identity  : <34dba26ea2b9d6ff1b8b55a347f8f083>
Name      : Anonymous Peer
-------------------------------------------------
Announced : 2 minutes ago
< Announce Now                                  >
-------------------------------------------------
< Save                >     < Node Info         >
```

```
                    <f899a9d42571b640a399a8c371d2e0e3>
Ⓝ f899a9d42571b640a399a8c371d2e0e3:/page/bulletin_board.mu
--------------------------------------------------------------------------

                                Node Blue
                        Testing Node App Template

                Home | Wiki | Register | Login | 02:17 01/22/2026


Log in or Register to see more!

A simple bulletin board.
<Bulletin Board>

Classified listings, a way to facilitate trading.
<Market Listings>


Welcome
Welcome to the Node Blue homepage!

This node is a demo of my "Node App Template" scripts. See https://codeberg.org/etux/Node_App_Template for code.

The Node App Template is a way to build really cool NomadNet sites.

This template handles:
* User registration and login
* Associate Reticulum identity with user account
* Users can message eachother (internal, not using LXMF)
* Administrative way to manage users
* Example page to create your own pages that require authentication.
* Microblogging plugin
* Wiki plugin
* Bulletin Board plugin
* Market plugin
* File Browser

--------------------------------------------------------------------------
                        Receiving response 50% 4.54 Kbps
[C-l] Nodes/Announces  [C-x] Remove  [C-w] Disconnect  [C-d] Back  [C-f] Forward  [C-r] Reload  [C-u] URL  [C-g] Fullscreen  [C-s / C-b] Save Node
```



Reticulum MeshChat
Developed by Liam Cottle

Connected | Compose

- Messages
- Nomad Network
- Interfaces
- Network Visualiser
- About

Conversations | Announces

Search 321 Peers...

- My Dear Peer — 12 seconds ago
- SHTF EchoBot — 13 minutes ago
- Anonymous Peer — 14 minutes ago
- SherbyNode Message Board — 26 minutes ago
- Sovbit Message Board — 27 minutes ago
- EchoBot - Send me your test messages! — 28 minutes ago
- Anonymous Peer — 28 minutes ago
- Beleth RNS Message Board — 31 minutes ago
- InternLXMFNode — 33 minutes ago
- Anonymous Peer — 38 minutes ago
- Beleth Distribution Group — 1 hours ago
- Anonymous Peer — 1 hours ago
- Alixpat — 1 hours ago

My Identity — Save
liam@liamcottle.com
Identity Hash
e49ac4d5b451715a647a6726704c41ac
LXMF Address
07a4912375337d7543f0aa1e8bfefea8
Announce — Announce Now
Every 15 Minutes
Last announced: 34 seconds ago
Calls — Open Phone
Status
Hung up, waiting for call...

SHTF EchoBot
<1dbc2281c4223c7a479de8cd8a988ae4> 3 hops away

Hello Echo Bot! 👋
delivered

SHTF EchoBot
Hello Echo Bot! 👋

Thanks for echoing back my message! 😎
delivered

SHTF EchoBot
Thanks for echoing back my message! 😎

Here's a photo of my solar mesh node. Can you send it back to me?
delivered

Send a message...

Add Files | Add Image | Send

LoRa

LoRa is a physical layer for data transmission over 915MHz radio (902-928MHz)

Reticulum does not use LoRaWAN! Because routing and identities are handled by the Reticulum application running on your computer. Instead Reticulum just sends simple LXMF messages from point-to-point.

Our Reticulum program communicates with the LoRa devboard over serial.

Spread-Spectrum Modulation

https://www.youtube.com/watch?v=jHWepP1ZWTk