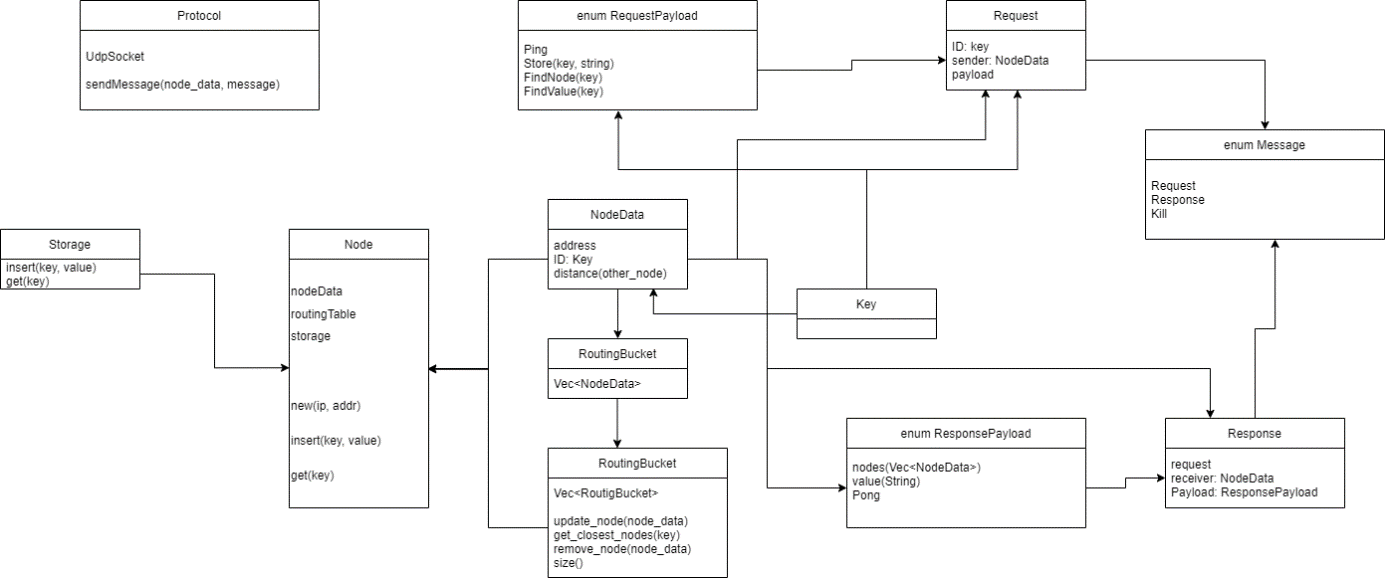
**Kademlia code review**

<https://gitlab.com/jeffrey-xiao/kademlia-dht-rs>

**Architecture :**



Messages are serialized with **serde crate** before being sent to the other nodes via UDP.

**Comments :**

The implementation uses the UDP protocol.

**Bucket table :**

src/routing.rs

Une image contenant texte

Description générée automatiquement

Une image contenant texte

Description générée automatiquement

The bucket table is a list of list of NodeData. NodeData contains the node ID and address, and allows to sort nodes and get distance between them.

**Update bucket table :**

All constants are defined in src/lib.rs.

REPLICATION\_PARAM is the maximum number of entries in a k-bucket, here 20.

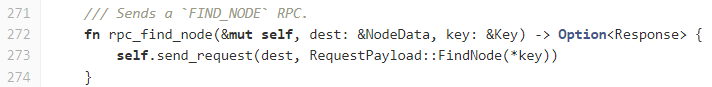
ROUTING\_TABLE\_SIZE is the maximum number of k-buckets in the routing table, here 256.

Une image contenant texte

Description générée automatiquement

**Find\_node :**

Src/node/mod.rs



It will send a request of type FindNode to the destination, with the requested node ID as parameter.

As a response to a SendNode request, the other node will return the closest nodes to the targeted one:

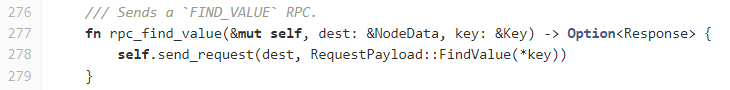
Une image contenant texte

Description générée automatiquement

(It can be done because we are able to calculate the inter-nodes distance).

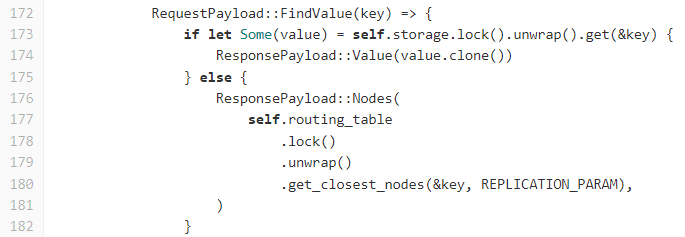
**Find\_value :**

Same as find\_node, a request of type FindValue is sent:



Then, when a FindValue request is received, 2 options:

* We can return the value
* We return the closest nodes able to contain the value



**Ping :**

We send a request to the target of type ping:

Une image contenant texte

Description générée automatiquement

The target answers PONG:



**Join :**

The join procedure is done directly when creating a new node. The entry point is the bootstrap parameter:

Une image contenant texte

Description générée automatiquement

Line 48, the entry point is added to the buket table.

Then let’s have a look to the *bootstrap\_routing\_table()* function:

Une image contenant texte

Description générée automatiquement

First of all, the node will try to find itself to fill its bucket table. Then it will search for random nodes in order to fulfill the table.

This part could be improved: it could be done concurrently in order to avoid loosing time.

**Leave:**

A **kill** message is sent to the node itself:

Une image contenant texte

Description générée automatiquement

When received, the corresponding node will be marked as inactive by itself, the other nodes are not informed.

Une image contenant texte

Description générée automatiquement

This part could be improved, in order to inform the other nodes that we are leaving.

**Benchmarks :**

It looks like the project isn’t maintained anymore, the last commit has been pushed 3 years ago. The merge request hasn’t been considered.

There are 8 files, sliced into structures and enum (Rust is a functional paradigm language).

There is a total of 1245 lines of an excellent Rust code, including the unit tests; which is quite small.

The nodes communicate through UDP, and the messages are serialized using **serde crate**, in bincode format (<https://docs.rs/bincode/latest/bincode/>), as you can see in the file **src/protocol.rs**.

**Criteria for Software Self-Assessment :**

kademlia-dht-rs: Family=vehicle; Audience=partners; Evolution=nofuture; Duration=2; Contribution=none; Url=https://gitlab.com/jeffrey-xiao/kademlia-dht-rs

It’s a library aiming to create a small implementation of the kademlia protocol in Rust, for educational purposes.