

## Kerma - Task 1

### Description

Start coding the implementation of your Kerma<sup>1</sup> node. In this exercise, you will set up your repository and exchange a hello message with any peer connecting to you. Then, you will extend your Kerma node so that it can exchange messages and perform peer discovery.

- Decide what programming language you will use (we recommend a language you are proficient in).
- Find a cool name for your node.
- Implement the networking core of your node. Your node must listen to TCP port 18018 for connections and must also be able to initiate TCP connections with other peers. Your node must be able to support connections to multiple nodes at the same time.
- Implement canonical JSON encoding for messages as per the format specified in the protocol.
- Implement message parsing, defragmentation, and canonical JSON encoding and decoding. On receiving data from a connected node, decode and parse it as a JSON string. If the received message is not a valid JSON or doesn't parse into one of the valid message types, send an "error" message to the node. Note that a single message may get split across different packets, e.g. you may receive "type":"ge and tpeers" in separate messages. So you should defragment such JSON strings. Alternatively, a single packet could contain multiple messages (separated by "\n") and your node should be able to separate them. Note that JSON strings you receive may not be in canonical form, but they are valid messages nevertheless.
- Implement the protocol handshake: (a) When you connect to a node or another node connects to you, send a "hello" message with the specified format. (b) If a connected node sends any other message prior to the hello message, you must send an "error" message to the node and then close the connection with that node. Note: Every message you send on the network should have a newline, i.e. "\n" at the end. Your node should use this to help parse and defragment valid messages.
- Implement peer discovery bootstrapping by hard-coding some peers (found on TUWEL).
- Store a list of discovered peers locally. This list should survive reboots.

---

<sup>1</sup>Out of curiosity, Kerma means "coin" in Greek.

- Upon connection with any peer (initiated either by your node or the peer), send a "getpeers" message immediately after the "hello" message.
- On receiving a "peers" message from a connected peer, update your list of discovered peers.
- Devise a policy to decide which peers to connect to and how many to connect to. We suggest to connect to just a few nodes, and not all of them.
- Submit your implementation on TUWEL.

*Important: make sure your node is running all the time. Therefore, make sure that there are no bugs that crash your node. If our automatic grading script can not connect to your node, you will not receive any credit. Taking enough time to test your node will help you ensure this. Below is a (non-exhaustive) list of test cases that your node will be required to pass. We will also use these test cases to grade your submission.*

- *The grader node "Grader" should be able to connect to your node. If you don't pass this test, Grader would not be able to grade the rest of the test cases. So make sure that you test this before you submit. Test this on the machine whose IP address you will submit.*
- *Grader should receive a valid hello message on connecting.*
- *The hello message should be followed by a getpeers message.*
- *Grader should be able to disconnect, then connect to your node again.*
- *If Grader sends a getpeers message, it must receive a valid peers message.*
- *If Grader sends {"type":ge, waits for 0.1 second, then sends tpeers"}, your node should reply with a valid peers message.*
- *If Grader sends any message before sending hello, your node should send an error message and then disconnect.*
- *If Grader sends an invalid message, your node should send an error message. Some examples of invalid messages are:*

1. *Wbgvgvf7rgtyv7tfbg{{{{*
2. *"type":"diufygeuybhv"*
3. *"type":"hello"*
4. *"type":"hello","version":"jd3.x"*
5. *"type":"hello","version":"5.8.2"*

- *If grader sends a set of peers in a valid peers message, disconnects, reconnects and sends a getpeers message, it must receive a peers message containing at least the peers sent in the first message.*
- *Grader should be able to create two connections to your node simultaneously.*

**Due date: 31st October, 11.59pm**

**Remember to provide us with your static public IP by October 25th.**