P2P Protocol

Command Enum

This enum defines commands used in the protocol messages.

Command	Description
`JOIN_PARENT`	Connection to the parent specified as argument
`JOIN_PARENT_RESPONSE`	Response from parent with a list of all nodes in the network
`JOIN_OTHER`	Request to join other nodes
`JOIN_OTHER_RESPONSE`	Response from those nodes with their stats
`KEEP_ALIVE`	Keep-alive ping
`WORK_REQUEST`	Request to perform a job
`WORK_ACK`	Acknowledgement of a work request
`WORK_COMPLETE`	Response of job completion
`SUDOKU_SOLVED`	Notification that a Sudoku puzzle is solved, with stats

Messages

The `Message` abstract class serves as the base class for all protocol messages, providing an abstraction over representation, `__str__`, and `to_dict` methods. This way, subclasses only have to specify new arguments, without having to implement these methods each time.

Note: Each time the `address` argument is used, it refers to the sending node's address, which is a tuple `(ip, port)`. This happens because sockets used to communicate between nodes use random ports, thus it's easier to identify a node by its IP address and binding port, instead of random ones.

JoinParent

When a node is created and a parent is specified, it sends a request to the parent to get the list of all nodes in the network.

Argument	Type	Description
`address`	`Address`	Address of the node requesting to join

JoinParentResponse

This message is a response to the `JoinParent` message. It contains the list of all nodes in the network.

Argument	Type	Description
`nodes`	`list[Address]`	List of all nodes in the network

JoinOther

After receiving the nodes list from the parent, this message is sent to each node to get their stats.

Argument	Type	Description
`address`	`Address`	Address of the node requesting to join

JoinOtherResponse

This message is a response to the `JoinOther` message, containing the node's stats, including solved puzzles and number of validations.

Argument	Type	Description
`solved`	`int`	Number of solved puzzles
`validations`	`int`	Number of validations

KeepAlive

A ping message, used in a scheduled manner to ensure the node is active.

Argument	Type	Description
None	N/A	No arguments required

StoreSudoku

This message is sent to all nodes when a new Sudoku puzzle is created, so that they store it in their states.

Argument	Type	Description
`id`	`str`	Sudoku UUID
`grid`	`list[list[int]]`	Sudoku grid
`address`	`Address`	Address of the node that got the HTTP request

WorkRequest

This message sends a work job to a node.

Argument	Type	Description
`id`	`str`	Sudoku UUID
`sudoku`	`Sudoku`	Sudoku object
`jobs`	`jobs_structure`	Current jobs status for the related sudoku
`job`	`int`	Job (square) number

WorkAck

Acknowledges the receipt of a `WorkRequest`.

Argument	Type	Description
`id`	`str`	Sudoku UUID
`job`	`int`	Job (square) number

WorkComplete

Indicates that the job is complete and may update stats accordingly. It includes the number of validations, for updating the stats.

Argument	Type	Description
`id`	`str`	Sudoku UUID
`sudoku`	`Sudoku`	Sudoku object
`job`	`int`	Job (square) number
`validations`	`int`	Number of validations

SudokuSolved

This message is sent to all nodes when a Sudoku puzzle is solved.

Argument	Type	Description
`id`	`str`	Sudoku UUID
`sudoku`	`Sudoku`	Sudoku object
`address`	`Address`	Address of the node that got the HTTP request

P2PProtocol Class

This helper class creates an abstraction over sending and receiving messages.

It uses `pickle` for serialization, instead of a typical `json` one. This allows native encoding and decoding of Python objects, including custom classes such as the ones above, which simplifies deserialization upon receiving a message, without having to rebuild objects based on the command.

Sending a message (`send_msg`)

Encodes and sends a message through a socket connection passed as argument.

Argument	Type	Description
`connection`	`socket`	Socket connection to send the message through
`message`	`Message`	Message object to be sent

Receiving a message (`recv_msg`)

Receives and decodes a message from a socket connection.

Argument	Type	Description
`connection`	`socket`	Socket connection to receive the message from