Liars Lie

Liars Lie is a game where a client queries a set of agents about an integer value. A configurable subset of agents tell the truth and will reveal the true value when asked. On the other hand, each liar will respond with an arbitrary (random) value, but always the same arbitrary value, when asked. The fun is to determine the “true” value v by asking the agents their individual values. We call this v the *network value* below. By running,

`./liarlie -mode [expert|standard]`

You can access the client and query independent agents. The agents and the client communicate according to a grpc protocol. There are two modes of the game, expert and standard. Each mode has 3 commands and you can only use such 3 commands within one mode. Without explicitly setting the **`-mode`** command line flag, the default is standard.

The agents are instantiated in goroutines, lightweight threads and registered with a local port number, also known as the **id** of an agent.

Standard mode

`start --value v --max-value max --num-agents number --liar-ratio ratio`

Should launch a number of independent agents, with `number * (1-ratio)` honest agents always responding with the specified integer value `v`, and `(number * ratio)` liar agents responding `x` with `x != v` and `1 <= x <= max`. This command will start the game and, when ready, produce the agents.config file, which will contain sufficient information to identify and communicate with all agents reliably, and print “ready” on the terminal. Because there are very limited number available ports in one local machine (depending on how many have bee in use), the maximum num-agents you can enter is 2^16-1 (65535). Even then, you might run into situations where the number of agents you can establish is way lower than this number because of the uniqueness of each machine.

Although the ordering of the flags input does not matter, there are several conditions that must satisfy to run this command successfully. Such as,

- max-value and value cannot be 1 simultaneously
- liar-ratio must be a float64 within range of [0, 1]
- num-agents must be an integer greater than 0

`play`

Can be entered after **start**. May be invoked multiple times. Upon each invocation, the client reads the agents.config file (which is to be treated as coming from an external service), connects to the agents using TCP + grpc protocol, plays a round of the game, and print the network value v. It is possible a network value v cannot be found, for example, when the liars in the network successfully fooled the client, then **play** will also printout such situation. Such situation will happen when there are more than one values of the same frequency among all the agents.

`stop`

This command will stop all agents listed in the file agents.config, removes agents.config from the directory, and exit from the executable.

Expert mode

The game continues to work in a situation where the client can only communicate directly with a subset of agents. We also want to be able to reconfigure the network on the fly. Here are the new commands to implement for expert mode:

`extend --value v --max-value max --num-agents number --liar-ratio ratio`

This command checks for the existence of agents.config, and, if present, *extends* the network by launching the **num-agents** agents and appending information about them into agents.config. If agents.config does not exist, then it has the same effect as **start** in standard mode.

If the network has been extended multiple times, the input of **–value**, **–max-value** and **–liar-ratio** are all read from the most recent **extend**. Meaning that, each time when the network is extended, there will be an overhaul of reconfiguration of things such as new number of liars and what value each agent holds, etc. Existing agents are likely to update the value they hold.

`playexpert --num-agents number --liar-ratio ratio`

In expert mode we can determine the network value by directly querying *one* running agent. We call it the proxy. The proxy agent will let itself communicate with all other agents in the network and sends the collected results from others plus its own value back to the client. The liar ratio gives the user assumption on the ratio of liars among agents. For example, if the reality of the network is that there are half of the agents who are liars, but the input of **liar-ratio** from **playexpert** is 1/4, then the client will try to find a **unique** collected value (from the array returned from the proxy) whose frequency is 1/4. The **–num-agents flag** needs to be anywhere between [1, *q*] with *q* being the number of running nodes. What –num-agents is does not really matter in the command because the client will always query exactly one agent.

`kill --id id`

We model network failures by allowing some agents to be killed. This command kills the agent with **id**, but leaves `agents.config` untouched. The **ids** can be found from agents.config. The liar ratio of the network **will not be** reconfigured because of a kill command. Meaning that, if you kill an agent, be it a liar or an honest agent, the network will not maintain its **liar-ratio** from the input of the most recent **extend**.