

Practica 4

Computación Distribuida 2021-1

Profesora: Karla Vargas

Diego Estrada

Jorge Cortés

Ricardo Desales

José Morán

5 de noviembre de 2020

1. Introducción

En esta práctica tendrás que implementar el algoritmo de consenso (el que no tiene terminación temprana).

2. Desarrollo

Esta práctica estará conformada por tres archivos (con sus respectivas interfaces) y un test:

- `NodoConsenso.py`
- `CanalRecorridos.py` (utilizaremos el mismo canal de la practica anterior.)
- `Test.py`

Tendrás que implementar el método del `NodoConsenso` para la ejecución de este algoritmo.

3. Pruebas

Las pruebas son evaluadas sobre una gráfica completa de seis nodos.

Para la ejecución de las pruebas¹ basta con ejecutar el siguiente comando en la terminal:

```
myUser $> pytest -q Test.py
```

4. Observaciones

- Respeta los constructores proporcionados.
- Tendremos como parámetro el número de fallos f en la función *consenso*.
- Por convención los primeros f nodos serán los que fallen. Esto es, si el conjunto de vértices es $V = \{v_0, v_1, v_2, v_3, v_4, v_5, v_6, v_7\}$ y se tiene que $f = 4$, entonces los vértices que fallaran serán $\{v_0, v_1, v_2, v_3\}$
- Simularemos los fallos haciendo que los nodos ya no envíen mensajes ni procesen nada del algoritmo. Pueden hacerlos fallar en cualquier ronda que ustedes quieran, pero recuerden que si un proceso falla en la ronda r en todas las demás rondas siguientes fallará.

¹Insistimos, pasar las pruebas no implica que su práctica esté correcta. Sin embargo ayudan a mantener cierta estructura entre todas las diferentes implementaciones y son guía para que visualicen el comportamiento de sus implementaciones con respecto a resultados que se conocen como correctos.

- Noten que este algoritmo es sensible a la ronda actual. Recuerden que el número de ronda en `simpy` podemos obtenerlo con `env.now`. Dado el poco tiempo que tenemos, si no logran hacerlo con `simpy`, son libres de simular este comportamiento usando solo Python. (El comportamiento que tienen que simular es que los nodos solo ejecuten el algoritmo un número fijo de veces)
- Atributos importantes de un nodo:
 - *V*: corresponde al arreglo de *id*'s que tiene el nodo. En el pseudocódigo original se nos dice que se trata de la referencia al nodo en cuestión, sin embargo para la práctica sólo usaremos los *id*'s por practicidad.
 - *New*: el conjunto de mensajes de cada nodo. De nuevo, en el pseudocódigo original el consenso se hace sobre las referencias a los nodos en sí, por lo que basta con que en este conjunto se tengan de igual manera sólo los *id*'s de cada uno de los nodos y no las referencias de estos.
 - *lider*: valor del *id*
 - *fallare*: booleano que será verdadero solo si el nodo fallará en algún punto del algoritmo.
- Cualquier duda escríbanos a la brevedad.