



Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS

COMPUTACIÓN DISTRIBUIDA

Práctica 02

Jorge Angel Sánchez Sánchez

Fecha de entrega: 18 de Abril del 2024

Desarrollo de la práctica:

- Primero cree una estructura `Nodo` que representa cada participante en el algoritmo. Esta estructura contiene información sobre si el nodo es noble o traidor, su plan inicial (ataque o retirada), y el resultado de su voto. noble: Es un booleano que indica si el nodo es noble
traidor: Es un booleano que indica si el nodo es traidor o no. Si traidor es true
plan y res: Es un booleano que representa el plan inicial/resultado final del nodo. Si plan/res es true, el nodo planea/resultado es atacar; si es false, el nodo planea/resultado es retirarse.
- después creamos el método `comunicacion` que simula la comunicación entre los nodos. En cada ronda, los nodos comparten su plan con otros nodos seleccionados aleatoriamente. Toma como entrada un arreglo de nodos `nodos[]`, el número de nodos `numNodos` y el número de la ronda actual `ronda`. Itera sobre cada nodo y simula que comparte su plan con algunos nodos vecinos seleccionados aleatoriamente. Los nodos vecinos seleccionados reciben el plan del nodo actual. Este proceso se imprime en la consola para mostrar la interacción entre los nodos.
- después el método `dConsenso` que verifica si hay consenso entre los nodos. es decir cuando la mayoría de los nodos están de acuerdo en el resultado del voto. Toma como entrada un arreglo de nodos `nodos[]` y el número de nodos `numNodos`. Cuenta el número de nodos que votaron por el ataque y el número de nodos que votaron por la retirada. Retorna verdadero si la mayoría de los nodos votaron por el ataque, falso en caso contrario.
- después el método `eRey` que selecciona al nodo con mayor apoyo como rey. Toma como entrada un arreglo de nodos `nodos[]` y el número de nodos `numNodos`. Itera sobre cada nodo y cuenta cuántos nodos están de acuerdo con su resultado. El nodo con más apoyo se elige como rey y se devuelve su índice.
- En el Main inicializamos los nodos, simulamos la presencia de nodos traidores, realizamos la comunicación entre los nodos en cada ronda, determinamos si se alcanza un consenso, y elegimos al rey.

Compilación:

- `gcc practica2JorgeSanchez.c -o practica2JorgeSanchez`

Ejecución:

- `./practica2JorgeSanchez`