Tarea 2

Diego Méndez Medina

1. Considera un sistema distribuido representado como una gráfica de tipo anillo, cuyos canales son bidireccionales, con $n = mk$ procesos, con $m > 1$ y k impar. Los procesos en las posiciones $0, k, 2k, ..., (m-1)k$ son macados inicialmente como líderes, mientras que procesos en otras posiciones son seguidores. Todos los procesos tienen un sentido de dirección y pueden distinguir su vecino izquierdo de su vecino derecho, pero ellos no tienen información alguna acerca de sus *ids*.

   El algoritmo 1 está destinado a permitir que los lideres recluten seguidores. No es difícil ver que todo seguidor eventualmente se agrega a sí mismo a un árbol enraízado con padre en algún líder. Nos gustaría que todos esos árboles tuvieran aproximadamente el mismo número de nodos.

   - ¿Cuál es el tamaño mínimo y máximo posible de un árbol?
   - Dibuja el resultado de una ejecución para e algoritmo con $k = 5$ y $m = 4$

Algoritmo 1 Algoritmo de reclutamiento para el problema 1

Inicialmente hacer
1. **if** yo soy un íder **then**
2. parent $\leftarrow$ id
3. **send**($<$ *recluta* $>$) a ambos vecinos
4. **else**
5. parent $\leftarrow \perp$
6. **end if**

Al recibir recluta desde p **hacer:**
7. **if** parent $= \perp$ **then**
8. parent $\leftarrow p$
9. **send**($<$ *recluta* $>$) a mi vecino que no es p
10. **end if**

Solución:

2. Realice un análisis preciso de la complejidad de tiempo y la complejidad de mensajes de :

   - El algoritmo de broadcastTree.
   - El algoritmo de convergecast.
   - El algoritmo de broadConvergecastTree.

Solución:

3. ¿Se basan los algoritmos de broadcastTree y convergecast en el conocimiento acerca del número de nodos en el sistema?¿Por qué?

   Solución:

4. Investiga y explica brevemente el concepto de time-to-live(TTL) usado en redes de computadoras y úsalo para modificar el algoritmo flooding visto en clase, de modo que un lider comunique un mensaje M a los procesos a distancia a lo más d del líder (M y d son entradas del algoritmo); todos los procesos a distancia mayor no deberian recibir M. Da un breve argumento que demuestre que tu algoritmo es correcto, y también haz un analísis de tiempo y número de mensajes.

   Solución:

5. Generaliza el algoritmo convergecast para recolectar toda la información del sistema. Esto es, cuando el algoritmo termine, la raíz debería tener todas las entrdas de todos los procesos. Analiza la complejidad de bits, es decir, el total de bits que son enviados sobre los canales de comunicación. (*hint: Cada mensaje de información puede tomar k bits*).

   Solución:

6. *a*) Da un algoritmo distribuido para contar el número de vértices con un árbol enraízado T, iniciando en la raíz.
   Solución:

   b) Extiene tu algoritmo para una gráfica arbitraria G.

   Solución:

7. Da un algoritmo distribuido para contar el número de vértices en cada capa de un árbol enraízado T de forma separada. Analiza la complejidad de tiempo y la complejidad de mensajes de tu algoritmo.

   Solución: