Universidad del Valle de Guatemala
Redes
Elean Rivas - 19062
Javier Alvarez - 18051
Javier Ramirez - 18099

Algoritmos de Enrutamiento

Algoritmos utilizados

Los algoritmos implementados para este laboratorio fueron:

- Dijkstra: El algoritmo de Dijkstra es un algoritmo de búsqueda de rutas utilizado en teoría de grafos para encontrar el camino más corto desde un nodo de origen a todos los demás nodos en un grafo ponderado no dirigido o dirigido. El algoritmo de Dijkstra garantiza encontrar las rutas más cortas en grafos con pesos positivos ya que no funciona correctamente en grafos con pesos negativos debido a la posibilidad de ciclos negativos.
- Distance vector routing: Este algoritmo tiene como finalidad obtener las rutas más cortas entre un nodo y sus vecinos directos. Esto se hace mediante tablas que genera cada nodo en la topología, que además están sujetas a cambios en base a las tablas calculadas por sus vecinos más cercanos.

Resultados

- Distance Vector Routing

  Para la siguiente topología:

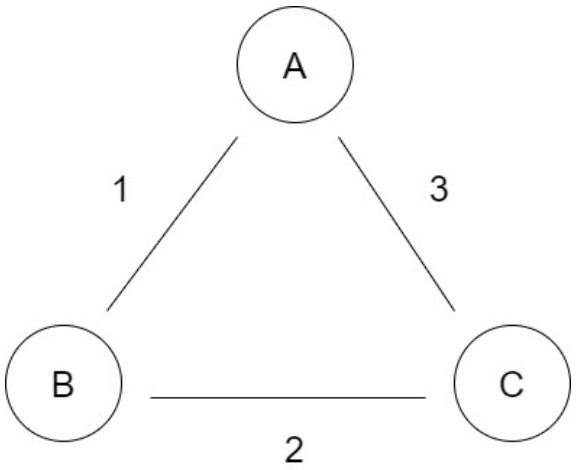

Figura 1: Topología simple de prueba para el algoritmo de Distance Vector.

  La tabla inicial para el nodo A:

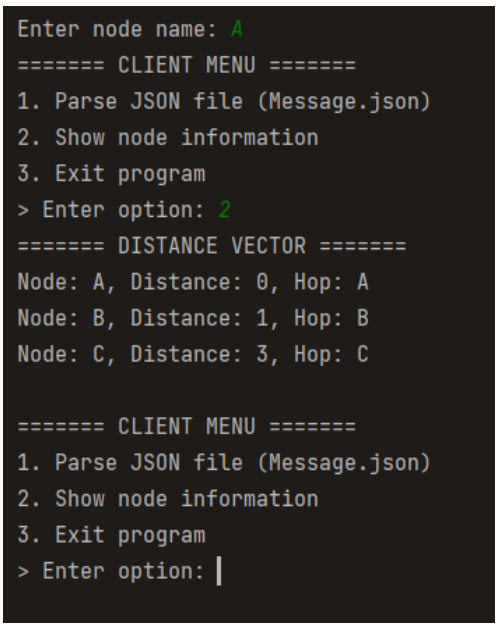

Figura 2: Tabla de Distance Vector inicial del nodo A.

Tabla de A luego de pasarle la tabla inicial del nodo B al nodo A:

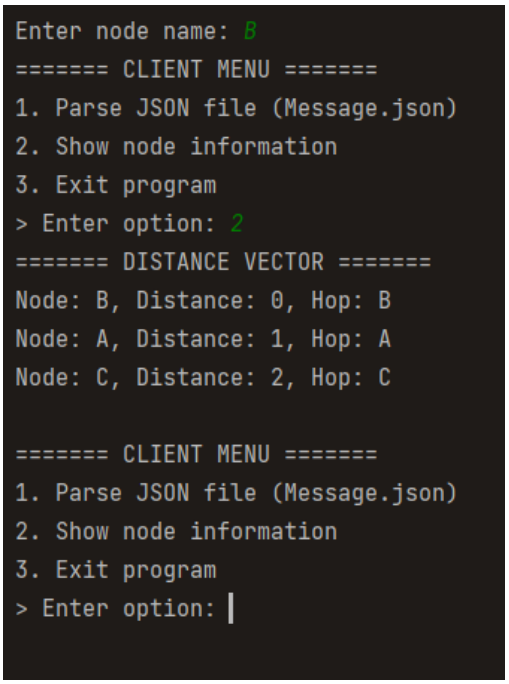

Figura 3: Tabla de Distance Vector inicial del nodo B

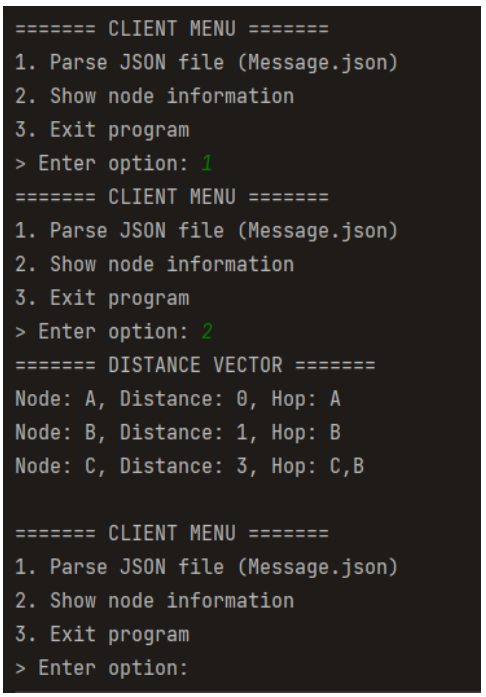

Figura 4: Tabla de Distance Vector actualizada del nodo A al mandarle la tabla de Distance Vector inicial del nodo B (Figura 2).

Discusión

Para el algoritmo de Distance Vector, la tabla inicial de cada nodo se calcula al inicio del programa, luego de verificar que el nombre del nodo que ingrese el usuario se encuentre dentro de la topología definida previamente. Una vez que la tabla inicial ha sido calculada, el programa procede a recibir mensajes de tipo “message” (mensaje de un nodo A a un nodo B), “info” (Mensaje de actualización por parte de otro nodo) o “echo” (Mensaje de reconocimiento). En caso que se pase un mensaje de tipo “info”, el programa procede a parsear un archivo de tipo.json con la información de las tablas calculadas por el nodo emisor, para después comparar las nuevas distancias calculadas con el algoritmo de Bellman-Ford. En caso que la nueva distancia calculada sea igual a la que se encuentra en la tabla, el programa procede a añadir al nodo emisor como canal alternativo de enrutamiento de mensajes. En caso que la distancia calculada sea menor a la que se encuentra en la tabla, se procede a actualizar la tabla con el nodo emisor como único canal de transmisión de mensajes.

Para los resultados del algoritmo de Distance Vector, se puede observar que la tabla de Distance Vector final del nodo A (Figura 4) ha sido actualizada luego de recibir y calcular las nuevas distancias en base a la tabla de Distance Vector inicial del nodo B. Para los nodos A y B, sus distancias se quedaron de la misma manera, mientras que la ruta más corta para el nodo C fue actualizado exitosamente, como se muestra en la topología de la Figura 1.

Comentario grupal

Para este laboratorio, la implementación de los algoritmos, se vio complicada en sobremanera por falta de información acerca de la implementación específica de los archivos JSON y el orden en que la topología está definida. Para futuras implementaciones del laboratorio se sugiere establecer una estructura bien definida para evitar inconvenientes en las distintas implementaciones de los algoritmos.

Conclusiones

En base a lo realizado en este laboratorio, se llegaron a las siguientes conclusiones:

- El uso de protocolos para intercambio de mensajes a través de redes con diferentes topologías es de gran importancia, debido a que se establece una única forma de comunicación y esto a su vez permite rastrear los mensajes a través de la red y detectar posibles errores producidos durante su transmisión.
- El intercambio y recepción de mensajes a través de nodos en una red es compleja y requiere de una comunicación constante e iterativa, de modo que todos los nodos tengan la mayor información posible. Para este laboratorio se asume que todos los nodos están disponibles en todo momento, pero en casos reales es necesario evaluar constantemente si los nodos se encuentran disponibles dentro de una red.

Referencias

- https://www.geeksforgeeks.org/dijkstras-shortest-path-algorithm-greedy-algo-7/