Guatemala 31 de agosto de 2021

Laboratorio #3

Redes

Abril Palencia 18198
Maria Isabel Ortiz 18176
Cristina Bautista 161260

Descripción de la Práctica

Conocer a un mayor nivel los diferentes algoritmos de enrutamiento, al igual que implementarlos y con esto obtener un mayor entendimiento de cómo funcionan las tablas de enrutamiento.

Implementación de los algoritmos

Resultados

Algoritmo Flooding

Para este algoritmo se necesita un archivo de la topología implementada donde se establecen todos los nodos y sus vecinos. Luego al ejecutar el cliente, el usuario logueado tiene que especificar qué nodo es el que quiere ser. Luego de esto un nodo envía un mensaje especificando el nodo destino y el mensaje. El algoritmo Flooding lee el archivo de la topología mostrando cuáles son sus vecinos y se lo envía a estos vecinos. Por otro lado, los otros clientes estarían escuchando constantemente los mensajes e imprimiendo la información del mensaje y esa función evalúa si este es el nodo destino y si no lo es vuelve a buscar sus vecinos y le envía el mensaje hasta llegar al nodo destino.

En este algoritmo no se logró que el cliente estuviera escuchando constantemente los mensajes y debido a eso no se logró hacer la demostración.

Algoritmo Distance Vector Routing

(DVR) requiere que un enrutador informe a sus vecinos de los cambios de topología periódicamente. Históricamente conocido como el antiguo algoritmo de enrutamiento de ARPANET (o conocido como algoritmo de Bellman-Ford).

Conceptos básicos de Bellman Ford: cada enrutador mantiene una tabla de Vector de distancia que contiene la distancia entre él y TODOS los posibles nodos de destino. Las distancias, basadas en una métrica elegida, se calculan utilizando información de los vectores de distancia de los vecinos.

Algoritmo Link State Routing

Para mandar mensajes usan el router que hacen link-state lo que permite que cada router empiece a reconocer la topología completa. Con la información obtenida se crea la tabla de routing con el shortest path (Dijkstra algorithm). Todos los nodos deben ser confiables.
Algunos highlights:
Link State Packet: Un paquete que contiene información del routing.
Link State Database: Es el conjunto de información que se obtuvo de link state packet
Algoritmo Dijkstra: Calcula el shortest path de un nodo a otro

Dentro del Algoritmo de Dijkstra se encuentra otros pasos:

1. Se toma un nodo como nodo raíz, se le asigna un costo a cada nodo, la información se obtiene de Link State Database
2. Se selecciona otro nodo, que es el más cercano a la raíz y se agrega al árbol
3. Después que se agrega el nodo al árbol, el costo de los otros nodos que todavía no se han seleccionado se van a actualizar.
4. Se repiten los pasos 2 y 3 hasta que todos los nodos se encuentren en el árbol.

En este algoritmo se logró que se escuchará constantemente, pero no se logró comunicar con los nodos vecinos, aunque sí reconoce cuales son los nodos vecinos de cada nodo.

Discusión

Para el algoritmo de distance vector routing se hizo una prueba en donde se crearon nodos y enlaces entre dichos nodos para simular el grafo. En la tercera imagen se puede observar que se instancian 3 routers A, B y C (también representados en el grafo). Luego se itera sobre el algoritmo dvr (Bellman Ford) en desorden para demostrar cómo cambian los valores de la tabla.

Para el algoritmo de Flooding los problemas que se tuvieron fueron más de parte del cliente que la implementación del algoritmo en sí. Una observación dada es que no se contempló es que al reenviar los mensajes a los vecinos no debería enviarlo a los nodos por los cuales ya pasó el mensaje previamente. Por otro lado en el cliente del chat se intentó implementar hilos para que en un hilo se mantuviera escuchando y el otro para cambiar de nodos o enviar mensajes, esta solución no pudo ser implementada. Por último, en este algoritmo se necesita que todos los usuarios que participan como nodos tienen que estar en la lista de contactos porque al enviar mensajes va a leer el mensaje de presencia de todos los usuarios para saber el nodo que es cada usuario y enviarlo según los vecinos del nodo actual. Si no se define un nodo en el mensaje de presencia el cliente no puede enviar o recibir mensajes.

Para el algoritmo de Link State Routing los problemas que se encontraron fueron en primer lugar el cliente, en lo personal, es la parte más complicada de implementar, lo siguiente es la comprensión del algoritmo, ya que este tiene más requerimientos, como que se tiene que tener presente el 'weight' de cada nodo desde el inicio, debe ser confiable. El segundo problema que encontré, es que si sabía que se queda escuchando todo el tiempo y cuando se escoge la opción de escribir un mensaje, si recibe los datos, pero al buscar el usuario de la persona para buscar ese nodo y trazar la ruta manda mensaje que el usuario no se encuentra en la lista.

Conclusiones

- Todos estos algoritmos tienen sus ventajas y desventajas. El uso de cada uno depende mucho de la situación en las que se desean utilizar. Por lo mismo, la popularidad y frecuencia de uso de estos algoritmos han variado en el internet a través de los años.
- Es importante el uso de eventos listener o de hilos para que el cliente esté siempre escuchando, en este caso que esté escuchando los mensajes recibidos.
- El algoritmo de Flooding es el menos eficiente, porque envía a todos el mensaje hasta que llegue al nodo destino, no calcula la ruta más eficiente como los otros dos algoritmos.
- El algoritmo de flooding es el más sencillo de implementar por lo que se describió en el punto anterior, solo busca nodos vecinos y lo envía a todos, no hacer cálculo alguno de una ruta en especifico.

- El algoritmo link state routing es el algoritmo, considero, más complicado de implementar.
- Si no se tienen las ‘weights’ de los nodos, no es útil el algoritmo, ya que depende del algoritmo Dijkstra, para obtener la ruta más corta, y este algoritmo utiliza las ‘weights’ para comparar rutas, antes de enviar el mensaje.