UNIVERSIDAD DEL VALLE DE GUATEMALA

Facultad de Ingeniería

Redes

Laboratorio 3 - Segunda parte

Algoritmos de Enrutamiento

MANUEL ALEJANDRO ARCHILA MORAN 161250

DIEGO JOSE FRANCO PACAY 20240

JUAN DIEGO AVILA SAGASTUME 20090

Descripción de la práctica

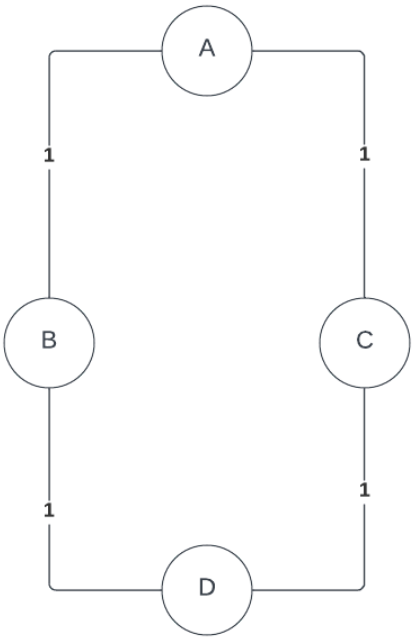
El objetivo de la práctica era la elaboración e implementación de tres algoritmos de enrutamiento, los cuales fueron: Flooding, Link State Routing y Distance Vector, para el envió de mensajes e implementarlos en una red simulada sobre el protocolo XMPP. Para cada uno de estos algoritmos se realizó un cliente de slixmpp que simulará el comportamiento y organizará el traspaso de mensajes según la metodología del algoritmo seleccionado.

El primer algoritmo que se realizó fue Flooding. Este algoritmo consiste en enviar el mensaje desde el nodo inicial a todos sus vecinos. Los nodos vecinos realizan el mismo proceso de enviar el mensaje a todos sus nodos vecinos, así sucesivamente hasta llegar al nodo receptor. Cuando el mensaje alcanza el nodo deseado el mensaje se presenta en pantalla.

El segundo algoritmo implementado fue Link State Routing. Este algoritmo para funcionar correctamente necesita conocer la topología de la red. Es por esto que al empezar el algoritmo se solicita el nombre de todos los vecinos directos que tenga el nodo, y por un lapso de tiempo entre los nodos existentes por medio del algoritmo flooding hacen el paso de información que en este caso sería sus nodos vecinos. Con esto se logra que cada uno de los nodos tenga la información de la topología completa y así saber como enviar mensajes a cualquier nodo sin importar que no sea su vecino directo. Luego para enviar el mensaje, el nodo emisor calcula el mejor camino del nodo actual al receptor usando el algoritmo de Dijkstra. Este nodo actual se encarga de enviar el mensaje al nodo más cercano que se encuentra en el camino óptimo. Como siguiente paso, el nodo que acaba de recibir el mensaje hace este cálculo nuevamente para poder realizar el envío al siguiente nodo en el camino, así sucesivamente hasta llegar al destinatario y poder mostrar el payload.

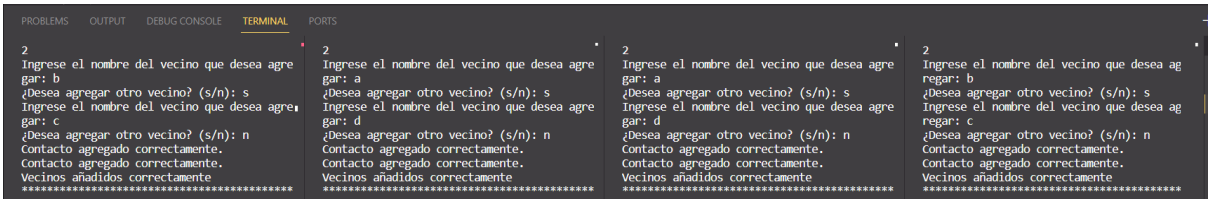
Por último, se realizó el algoritmo de Distance Vector. Este algoritmo se basa en el intercambio continuo de información entre nodos vecinos para determinar las rutas óptimas en una red. Al iniciar el algoritmo, cada nodo posee una tabla que contiene la distancia estimada hacia otros nodos en la red, junto con el próximo salto necesario para alcanzar esos destinos. Cuando un nodo detecta cambios en su tabla de distancia o en la información recibida de nodos vecinos, actualiza su tabla y envía copias actualizadas a sus nodos vecinos. Estos nodos, a su vez, repiten el proceso, lo que lleva a una propagación de la información de enrutamiento en toda la red. Si el nodo que está recibiendo la información detecta que su tabla de enrutamiento ya no se modifica a pesar de recibir información de sus vecinos debe de indicar que ya converge y cada nodo espera un lapso de tiempo de 45 segundos para indicar que ya converge. Este algoritmo tiene la peculiaridad de que es necesario que todos los nodos indiquen su convergencia para poder empezar a enviar y recibir mensajes. Para este proceso el nodo emisor consulta su tabla de distancia para determinar el próximo salto hacia el destino deseado. Este enlace se elige en función de las distancias y los saltos almacenados en la tabla. El mensaje se envía al nodo seleccionado, y este proceso se repite a medida que el mensaje se dirige hacia su destinatario final.

Resultados

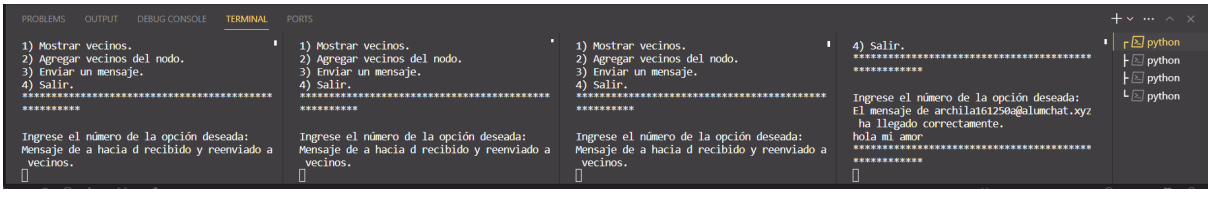


Flooding:

Creación de cada nodo:

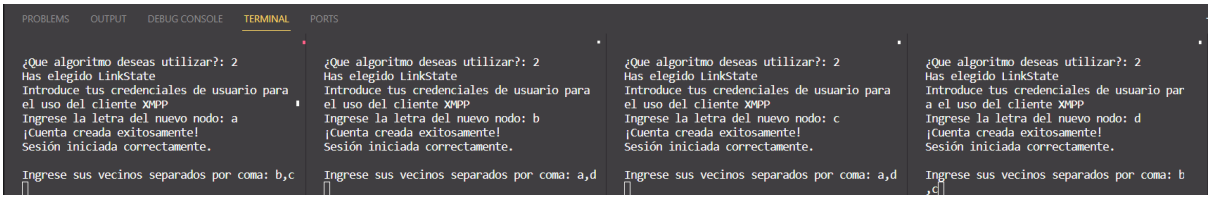


Envío de mensaje:

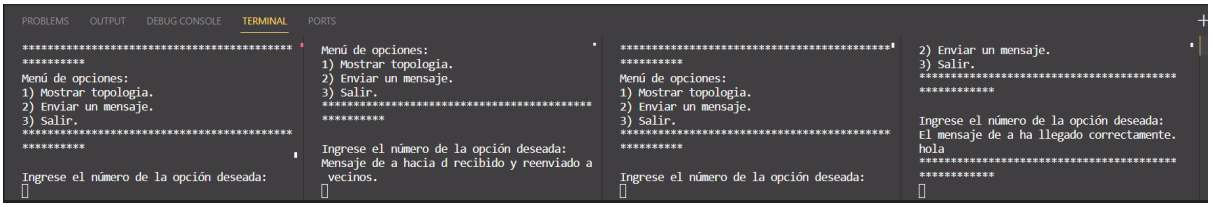


Link State:

Creación de cada nodo:

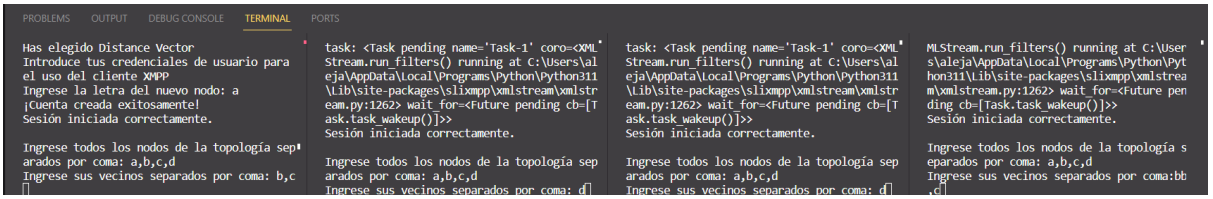


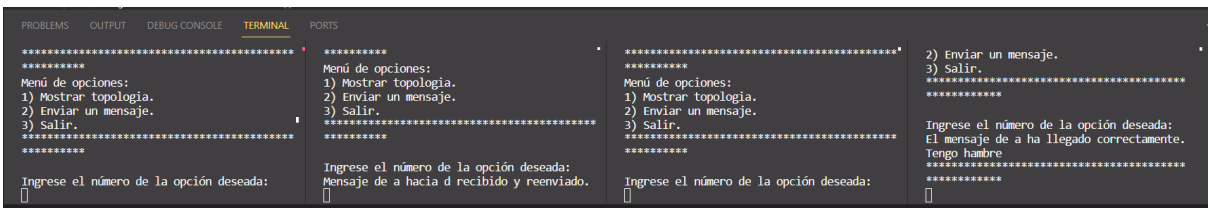
Envío de mensaje:



<u>Distance Vector:</u>

Creación de cada nodo:





Discusión

En primer lugar, debido a que se utilizó el servidor la prueba de los algoritmos fue mucho más rápida y eficiente, ya que en vez de simular la interacción por medio de terminales, cada nodo se comunicaba a través del servidor con sus vecinos y demás nodos según el algoritmo utilizado. Asimismo, una práctica que consideramos importante y que ayudó mucho es que a la hora de terminar de usar un nodo se elimina del servidor para que se pudiera seguir haciendo pruebas y fuera más fácil la interacción, es por esto que cada vez que se ingresa un nuevo nodo, se crea un nuevo cliente con el usuario “archila161250” + nombre del nodo y al terminar este se elimina para que pueda ser utilizado en otra instancia.

Con respecto a los algoritmos, estos se debieron modificar no sólo por la nueva interacción con el servidor sino también por el protocolo establecido por el supergrupo, ya que se tenían que introducir elementos como timestamp, intermediarios, y otros que no se habían considerado con anterioridad.

El algoritmo de Flooding se logró implementar sin mayor modificación ya que la lógica y la transmisión de mensajes se mantuvo igual, obviamente ajustándose para que la comunicación sea entre clientes dentro del servidor. El único cambio significativo fue el uso de un timestamp para evitar la duplicación de mensajes recibidos, que ayuda a limitar que el mensaje llegue una única vez sin importar el nodo pero tomando en cuenta el tiempo en el que fue enviado el mensaje original.

Por otro lado, Link State Routing si sufrió muchas modificaciones ya que en vez de saber toda la topología desde un inicio, se estableció un protocolo de mensajes de tipo "info" en el que cada nodo en forma de flooding envía a todos los nodos su topología, para que sus vecinos se lo envíen a los demás nodos hasta que todos conozcan la topología de todos los nodos, para tener una topología completa y poder proseguir a encontrar la ruta más corta y enviar mensajes.

Finalmente, Distance Vector necesitó varios cambios para que tuviera un funcionamiento ideal. El primer cambio fue el manejo de la tabla, ya que en la implementación pasada la búsqueda de información era ineficiente. Por otro lado, se maneja un sistema de rutas el cual verifica si un nodo tiene una conexión directa con el siguiente nodo en la cadena o si tiene que pasar por un intermediario. Esto se hace al recibir un mensaje y al enviar un mensaje. Después

de implementar estas nuevas medidas el funcionamiento de Distance Vector se había completado.

Comentario grupal

Para nosotros esta parte del laboratorio fue muy interesante debido a la incorporación del protocolo XMPP. Sin duda alguna la parte más difícil fue la ideación del protocolo ya que nosotros escogimos esa ruta con nuestro super grupo. Al tener protocolo listo solo se necesitaba modificar lo que realizamos con anterioridad para cumplir con lo que se nos pedía. Como nosotros conocíamos muy bien nuestro código fue bastante fácil realizar los cambios resultando en una tarea bastante fácil. También cuando realizamos pruebas con los otros grupos de nuestra topología nos dimos cuenta de todos los problemas de los que nos salvamos al implementar un protocolo.

Conclusiones

- Se logró implementar de manera correcta cada uno de los algoritmos de envío de mensajes a través del servidor utilizando un protocolo de comunicación utilizando cada nodo como un cliente.
- Aunque cada algoritmo tuvo que modificarse para la implementación de este laboratorio, los cambios fueron para mejora de dichos algoritmos y para facilidad de uso ya que no sólo permite comunicación sino también envío de información importante para que cada nodo se logre configurar con las necesidades de cada algoritmo.
- Se logró observar de mejor manera las ventajas y desventajas de utilizar cada algoritmo al utilizar un servidor para comunicar cada nodo, y con una topología mucho más grande se puede llegar a ver cómo utilizar cada uno de los algoritmos puede ser de utilidad o no dependiendo de las necesidades de la red y topología.

Referencias

Herramientas Web. (2023). Inundación. Retrieved August 30, 2023, from Lcc.uma.es website: https://neo.lcc.uma.es/evirtual/cdd/tutorial/red/inunda.html#:~:text=El%20principal%20inconveniente%20que%20plantea,establece%20alguna%20forma%20de%20limitaci%C3%B3n.