Universidad del Valle de Guatemala
Departamento de Ciencias de la Computación
CC3067 Redes
Jorge Andrés Yass Coy

Laboratorio No. 3 - parte 2

Interconexión de nodos

Ana Paola de León 203061
Andrea de Lourdes Lam 20102
Gabriela Paola Contreras 20213

Guatemala, 12 de septiembre de 2023

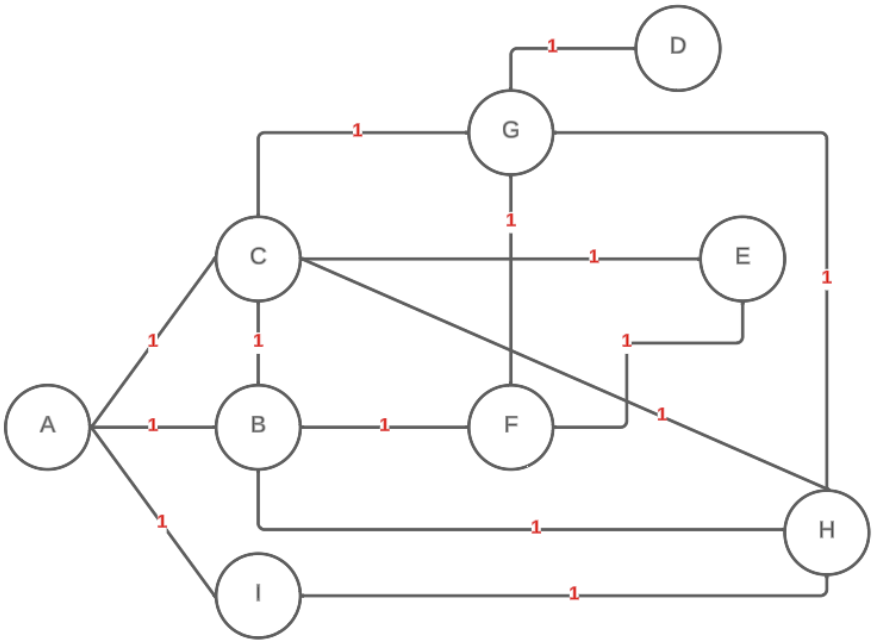
Introducción

El fin de este laboratorio fue implementar los algoritmos de enrutamiento previamente realizados en una red simulada sobre el protocolo XMPP, para ello las tablas realizadas debían de actualizarse y acomodarse a los cambios permitiendo de esta manera que pudiéramos analizar el funcionamiento de estos en su totalidad y simular cómo actúa el Internet

Esta actividad fue realizada en tríos y cada una modificó el algoritmo de enrutamiento previamente seleccionado. Para la realización se crearon grupos con la finalidad de poder conectar nuestros clientes con otros clientes, para ello se concretaron ciertos criterios a seguir en cada uno de los algoritmos, seguidamente se implementó el protocolo XMPP utilizando el servidor de "alumchat.xyz" para el envío de los mensajes necesarios para construir la tabla de enrutamiento en base a sus vecinos y para poder enviar un paquete de un nodo origen a un nodo destino utilizando dicha tabla. Cabe destacar que se hizo uso de dos archivos txt los cuales contenían información de los nodos vecinos y sus direcciones dentro del servidor permitiendo así realizar una simulación sin problemas y logrando enviar mensajes de un nodo a otro. Al finalizar se discutieron diferentes aspectos por medio de los cuales se puede concluir que la interconexión de nodos desempeña un papel fundamental en el establecimiento de comunicación y en la compartición de recursos lo cual facilita de manera significativa el flujo de datos y promueve una colaboración efectiva entre clientes.

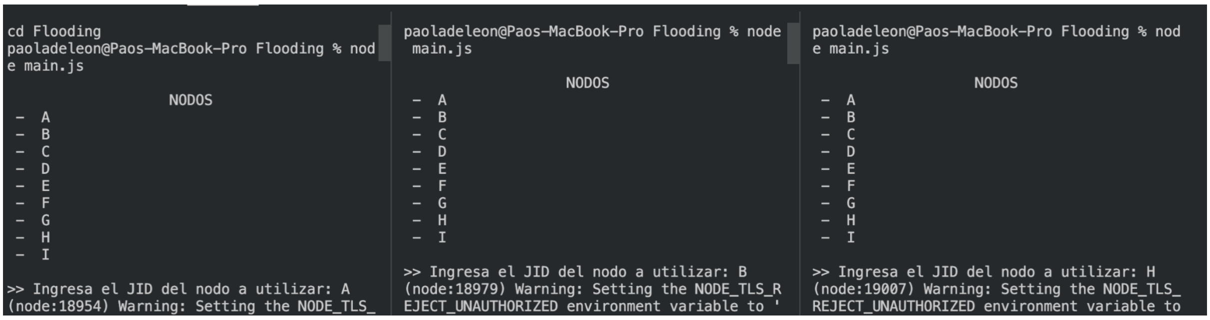
Resultados

- **Topología utilizada**

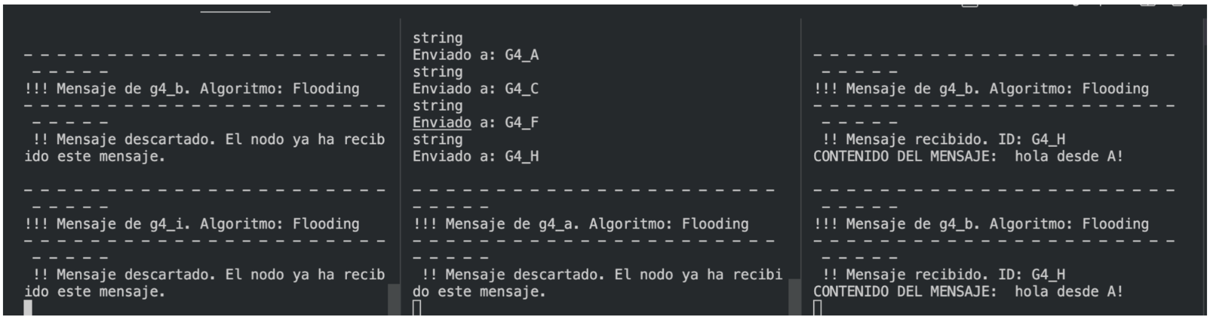


- **Flooding**

  - Conexión con XMPP

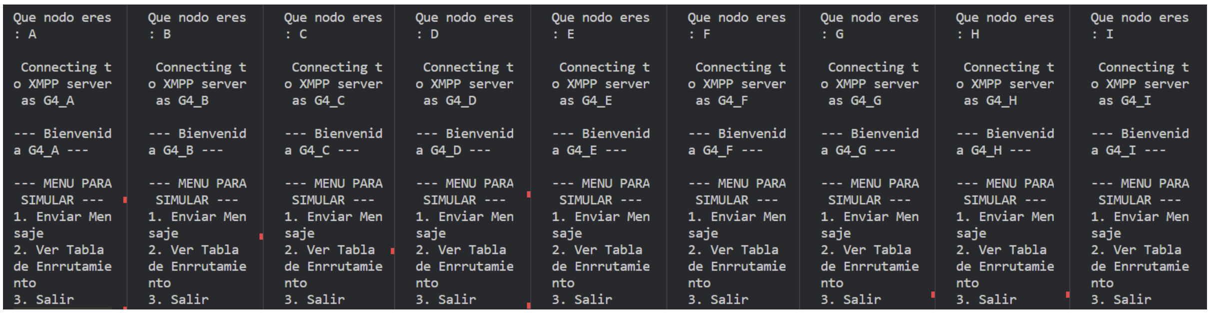


- Envío de mensajes

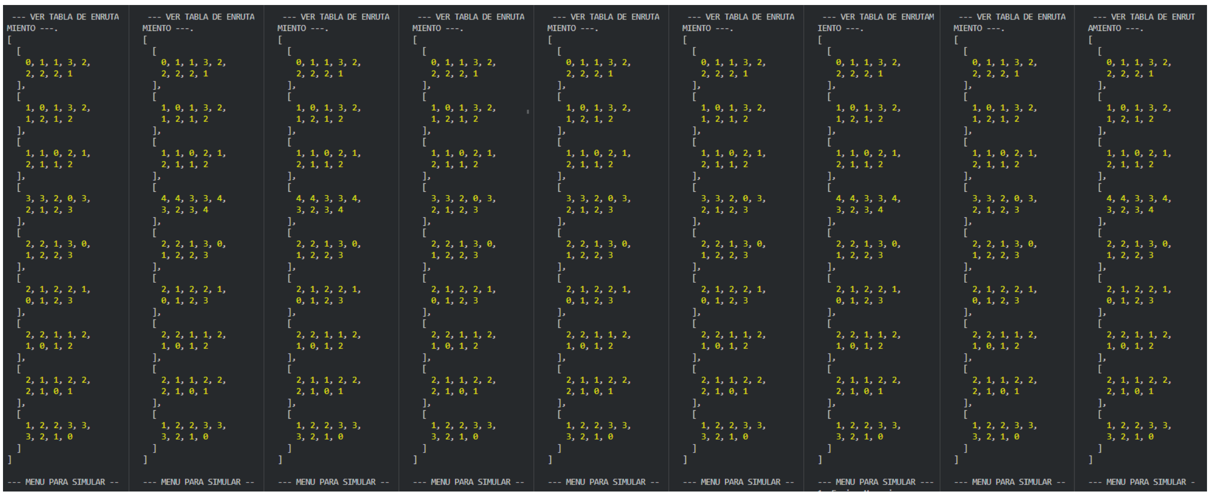


- **Distance Vector Routing**

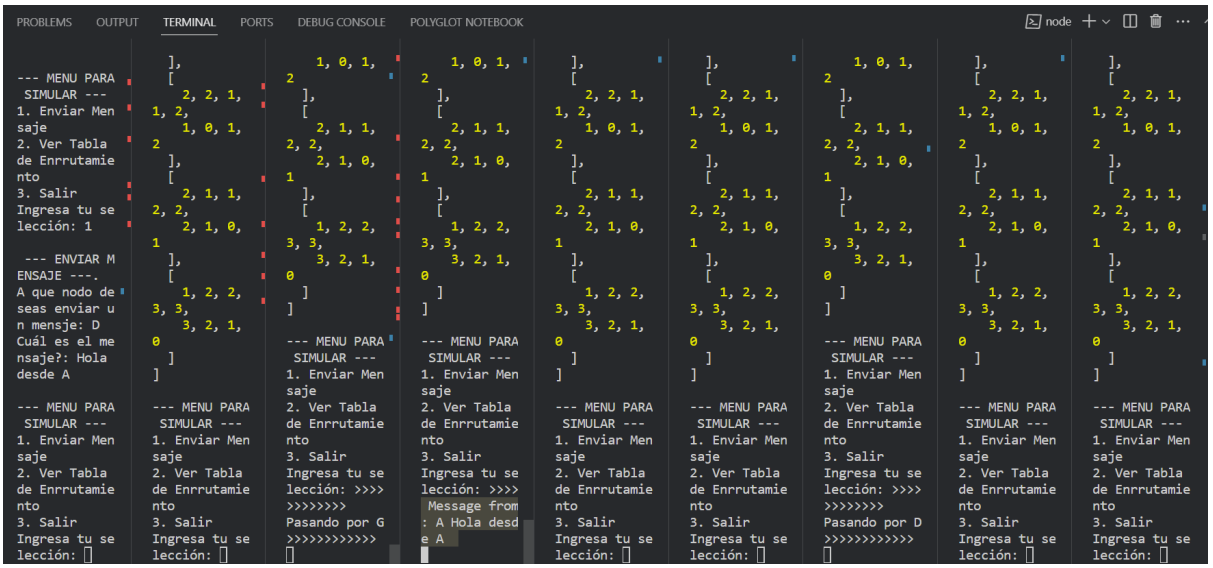
- Conexión con XMPP



- Tablas de los vecinos

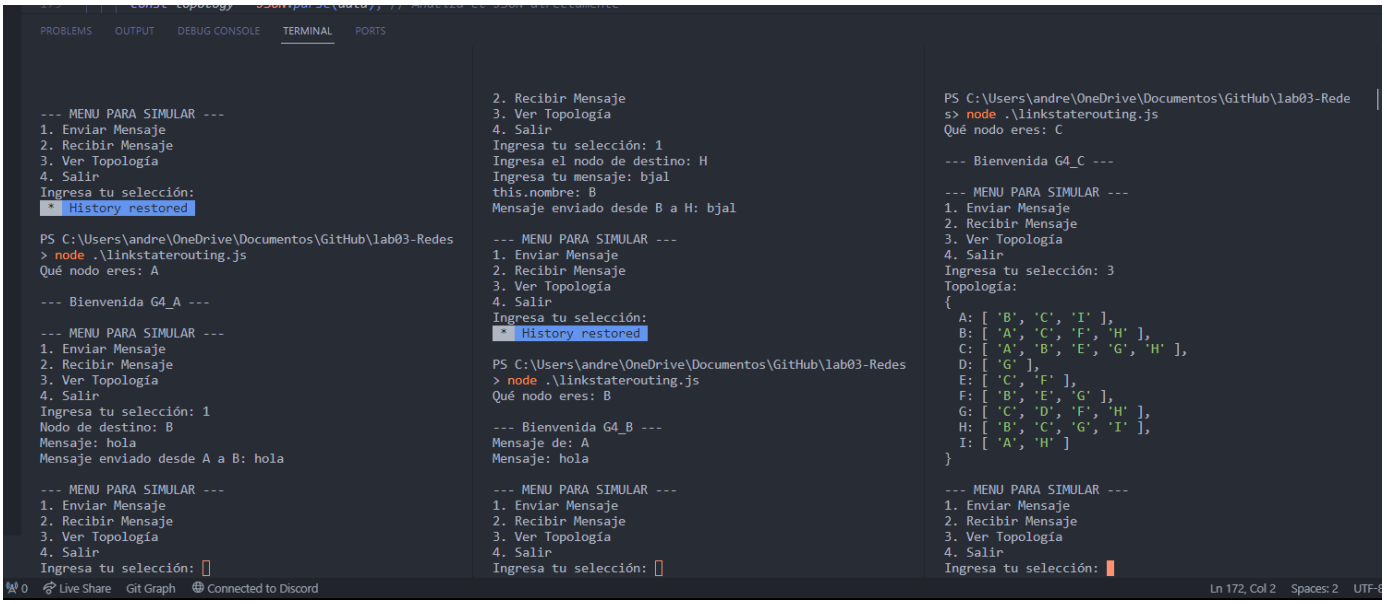


- Envío de mensajes



- **Link State Routing**

- Tablas de los Nodos



Cuando sale “Bienvenido ” y el nombre del nodo es que se logró la conexión con XMPP.
Seguidamente la primera terminal es el nodo A que le envía un mensaje al nodo B, la segunda terminal es el nodo B que recibe el mensaje y la tercera terminal es el nodo C que está mostrando la topología.

Discusión

El objetivo de este laboratorio fue analizar a profundidad el funcionamiento de los algoritmos de enrutamiento utilizando el protocolo XMPP como base de comunicación, para cumplir dicho objetivo se modificaron los algoritmos de Flooding, Distance vector routing y Linked state permitiendo que estos se comunicaran mediante un servidor logrando así que estos se acoplen a cambios en la infraestructura.

Para la realización del algoritmo de Flooding se tomó como base la primera parte del laboratorio, añadiendo múltiples funcionalidades como la conexión al servidor de la clase y el envío de información utilizando una topología dada. En este se logró ver de forma más directa lo complejo que puede llegar a ser este algoritmo pues se realizaron múltiples envíos de paquetes para llegar al destinatario final. Sin embargo, se logró entender a gran escala cómo mediante la aplicación de un algoritmo se puede establecer una ruta para aprovechar la conexión entre otros nodos y así enviar la información de una mejor forma.

El algoritmo de distance vector fue optimizado para poder establecer una conexión con el servidor de alumchat, es importante destacar que esta conexión se realiza de manera automática al momento de colocarle a la terminal que nodo es el que va a simular. Es importante destacar que para contar con los nodos vecinos ahora se trabajó con un documento llamado topo-g4 el cual detalla los vecinos de cada nodo y el costo para todos se estableció como 1 y este se va actualizando mediante el uso de la ecuación Bellman-Ford. Ahora bien, para el envío del paquete, el cual contiene la tabla de enrutamiento, este es enviado de manera automática utilizando la stanza de message lo cual permite que la tabla de enrutamiento se genere rápidamente y sin depender de interacciones manuales; esto se logra utilizando un listener y una condicional if para verificar constantemente que la tabla no cuente con cambios luego de cierto tiempo. Como se puede observar en los resultados obtenidos la tabla logra converger en todos los nodos luego de un tiempo. Ahora bien al contar con la tabla ya completa se le solicita a uno de los nodos si desea mandar un mensaje independientemente si es directamente un nodo vecino o no, al contar con un nodo destino se procede a utilizar la tabla de enrutamiento para ver cual es el mejor camino a tomar y se hace el envío del mensaje, esto se puede observar en la tercera imagen del apartado de resultados de dicho algoritmo. De esta manera podemos corroborar el correcto funcionamiento de nuestro algoritmo haciendo uso de un protocolo XMPP

Para el algoritmo de link state routing se tomó como base la primera parte del laboratorio, pero seguidamente se tuvieron que realizar cambios para poder crear la conexión con el XMPP y para tanto la creación de mensajes, como recibirlos. Para esto se crearon funciones asíncronas. En la clase LinkState, se encuentra el método dijkstra(). Este método implementa el algoritmo de Dijkstra para calcular las distancias más cortas desde el nodo actual a todos los demás nodos en la topología. Se inicializan las distancias y los nodos anteriores en this.distancias y this.anterior, respectivamente. Luego, se utiliza una cola de prioridad (this.fila) para seleccionar los nodos con las distancias más cortas en cada iteración. Se recorren los vecinos de cada nodo y se actualizan las distancias si se encuentra una ruta más corta. Al finalizar, se obtienen las distancias más cortas y los nodos anteriores para cada nodo en la topología. Para la implementación En la clase LinkState, se encuentran los métodos enviarMensaje(destino, mensaje) y recibirMensaje(emisor, receptor, mensaje). Estos métodos verifican si el nodo de destino es un vecino directo antes de enviar o recibir un

mensaje. Se utiliza la información de la topología de red cargada desde un archivo para determinar las relaciones entre los nodos.

Comentario grupal

Como grupo consideramos que este laboratorio fue un poco más sencillo que el anterior pues ya contábamos con una buena parte avanzada, no obstante el tener que ponernos de acuerdo con 6 personas más lo volvió un poco complicado ya que debíamos asegurarnos que nuestros algoritmos iban a funcionar correctamente con los de nuestros compañeros y de realizar todos los cambios necesarios para que no se fueran a confundir con otros algoritmos. Así mismo, el uso del servidor fue algo que impidió mucho avanzar amenamente puesto a que todos nos encontrábamos conectados y en algunas ocasiones hacían que nuestro algoritmo dejará de funcionar pues se enviaban cosas ajenas al paquete para verificar una comunicación. A pesar de todo esos altercados se logró realizar el laboratorio sin problema y al finalizar esta pudimos entender de una manera más sencilla el funcionamiento de las tablas y el papel que estas juegan para lograr contar con una comunicación rápida y eficiente.

Conclusiones

- La implementación de un servidor dentro del algoritmo de vector de distancia gestiona las tablas de enrutamiento de manera eficiente y facilita la comunicación efectiva entre los nodos de la red, logrando así una transferencia de mensajes rápida y sencilla.

- Se demuestra que el algoritmo de Flooding es más un acercamiento teórico al enrutamiento de paquetes y no un algoritmo eficiente que pueda ser replicable en la vida real.

- El algoritmo de Dijkstra es eficiente en redes pequeñas o medianas. Sin embargo, en redes de gran escala, puede ser necesario considerar algoritmos de enrutamiento más avanzados debido a su complejidad computacional.