

UNIVERSIDAD DEL VALLE DE GUATEMALA

Redes

Sección 21

Jorge Yass



Laboratorio 3

Laboratorio 3 XMPP

Oscar Esteban Donis Martínez - 21610
Abner Iván García Alegría - 21285
Samuel Alejandro Chamale Rac - 21881

Repositorio

<https://github.com/UVG-Works-Hub/routing>

Descripción de la práctica

Esta práctica se enfocó en la implementación y análisis de algoritmos de enrutamiento dentro de una red simulada utilizando el protocolo XMPP. El objetivo principal fue comprender cómo funcionan las tablas de enrutamiento y cómo estas pueden actualizarse dinámicamente en respuesta a cambios en la infraestructura de la red. Los algoritmos implementados incluyeron Flooding y Link State Routing. Cada nodo en la red simulada correspondió a un usuario de XMPP en el servidor alumchat.lol.

Descripción de los Algoritmos Utilizados y su Implementación

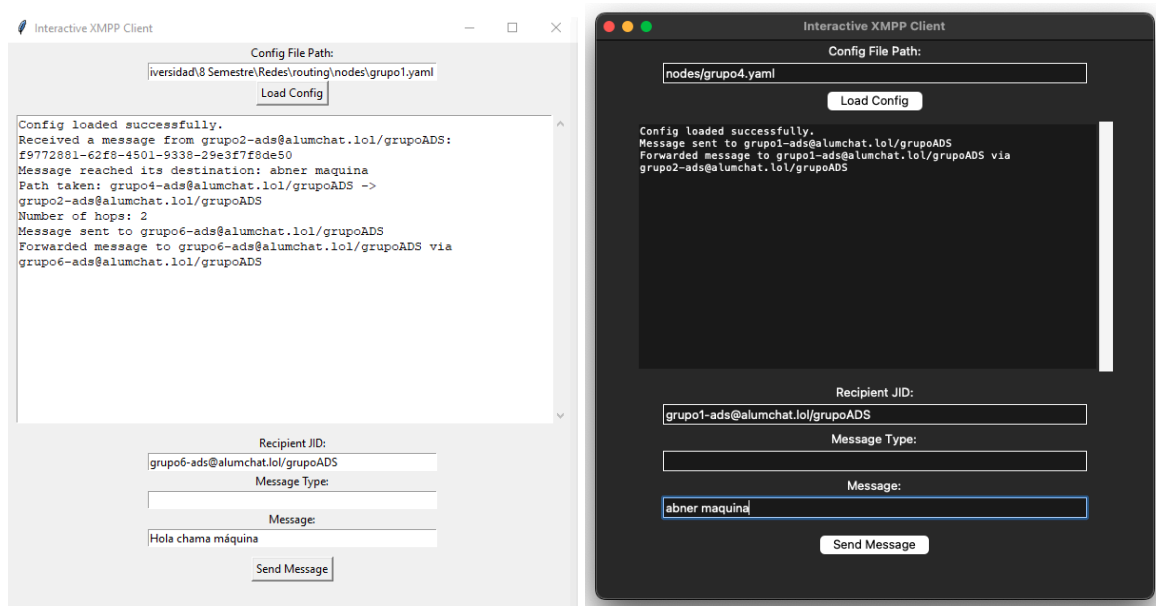
Flooding

- **Descripción:** Flooding es un algoritmo de enrutamiento simple en el que cada paquete recibido por un nodo es reenviado a todos sus vecinos, excepto al vecino del cual se recibió el paquete. Este método asegura que el mensaje llegue a su destino si hay al menos una ruta entre el origen y el destino, pero también puede causar congestión debido a la duplicación de paquetes.
- **Implementación:** El algoritmo de inundación se implementa en el código mediante el método `flood_message()`, que gestiona la propagación de mensajes en la red. Cuando un nodo recibe un mensaje, lo reenvía a todos sus vecinos excepto al nodo desde el cual lo recibió, garantizando así que el mensaje se propague a todos los nodos de la red. Para evitar la retransmisión infinita, el nodo utiliza un identificador único para cada mensaje recibido y mantiene un registro de estos identificadores para asegurarse de que no se reenvíe un mensaje que ya haya procesado.

Link State Routing

- **Descripción:** Link State Routing (LSR) es un algoritmo más avanzado en el cual cada nodo conoce la topología completa de la red y calcula la mejor ruta hacia cada destino utilizando el algoritmo de Dijkstra. A diferencia de Flooding, en LSR cada nodo almacena información detallada sobre sus vecinos y comparte esta información con todos los demás nodos para construir un mapa completo de la red.
- **Implementación:** La implementación del algoritmo LSR en el código se basa en la creación y distribución de una base de datos de estado de enlace entre los nodos de la red. Cada nodo inicializa esta base de datos con los costos de los enlaces hacia sus vecinos y luego comparte esta información a través de mensajes tipo "info" utilizando el método `share_link_state()`. Una vez que un nodo recibe la información de estado de enlace de otros nodos, actualiza su base de datos y utiliza el método `compute_routing_table()` para calcular la ruta más corta hacia cada nodo en la red empleando el algoritmo de Dijkstra.

Resultados



Los resultados mostraron claras diferencias en el rendimiento y la eficiencia de los algoritmos implementados. En redes pequeñas o moderadamente densas, el algoritmo de Flooding garantiza una rápida difusión de los mensajes, pero a costa de un uso excesivo del ancho de banda. En cambio, LSR demostró ser más eficiente en términos de tráfico de red y tiempo de entrega en redes más grandes, gracias a su capacidad para calcular rutas óptimas.

La implementación GUI nos permitió una mejor interacción y visualización de los logs, facilitando la monitorización en tiempo real de la transmisión de mensajes y el comportamiento de los algoritmos. Además, la carga y configuración a través de un archivo YAML hizo que la inicialización del cliente fuera más flexible y adaptable a diferentes escenarios de red.

Discusión

El desarrollo de este laboratorio nos permitió profundizar en la comparación entre dos enfoques clásicos de routing: Flooding y Link-State Routing (LSR). A través de la implementación y prueba de ambos algoritmos, pudimos observar que, aunque Flooding es sencillo de implementar y asegura la entrega del mensaje en redes pequeñas o menos complejas, su naturaleza indiscriminada de enviar mensajes a todos los nodos puede llevar a una saturación de la red en entornos más grandes. Este comportamiento es particularmente problemático en redes densas, donde el tráfico redundante puede sobrecargar los recursos de la red, resultando en una disminución del rendimiento general y una mayor latencia.

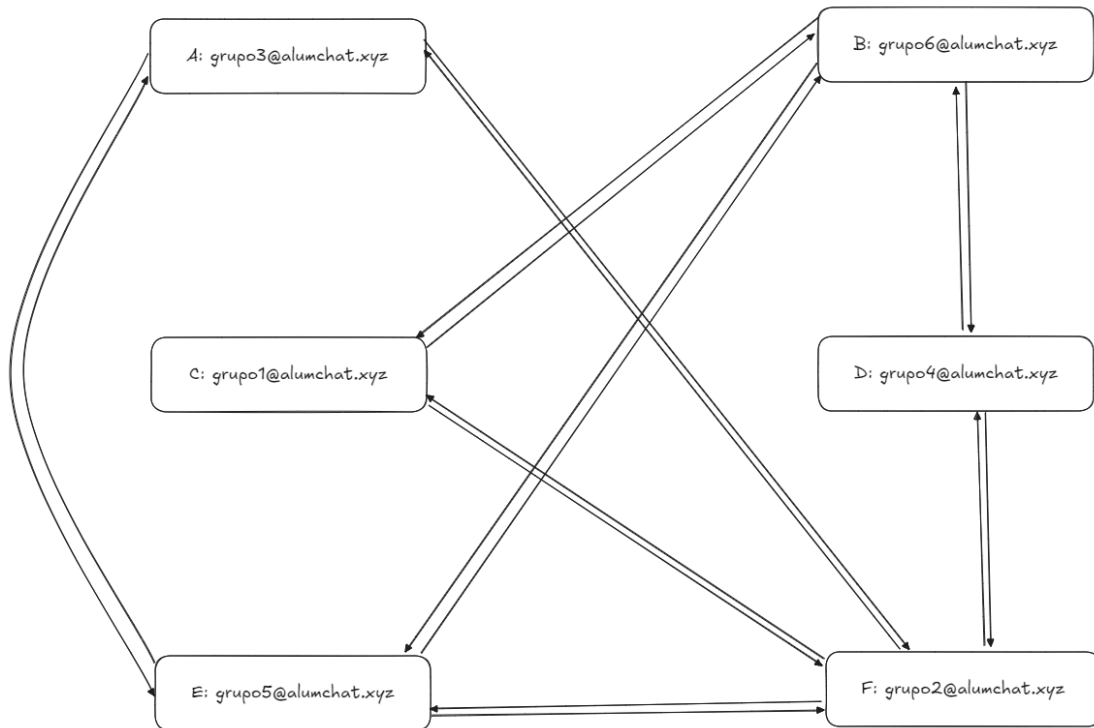
Al ver los resultados observamos que el algoritmo LSR se mostró mucho más eficaz en términos de optimización del uso de los recursos de red. LSR construye una visión completa de la topología de la red, permitiendo que cada nodo calcule las rutas más cortas hacia todos los demás nodos, lo que minimiza el tráfico innecesario y maximiza la eficiencia. Sin embargo, su complejidad de implementación es mayor y requiere un mayor esfuerzo de sincronización entre los nodos, además de ser más costoso en términos de procesamiento y almacenamiento de información sobre la red. A pesar de estos desafíos, LSR es claramente superior en escenarios donde la eficiencia y la gestión del tráfico son críticas, lo que lo convierte en una opción preferida para redes de gran escala y alto rendimiento.

Conclusiones

- El algoritmo de Link State Routing ofrece una solución mucho más eficiente y escalable para el enrutamiento en redes grandes y estables, gracias a su capacidad para calcular rutas óptimas basadas en la información global de la red. Al compartir el estado de los enlaces entre todos los nodos, cada uno puede construir una tabla de enrutamiento que minimiza la latencia y el tráfico innecesario, mejorando así la eficiencia en la transmisión de datos.
- Flooding es una técnica buena cuando hablamos de redes pequeñas, sin embargo, cuando la cantidad de nodos crece, también lo hace la carga que implica enviar un simple mensaje. Esto sin importar que tan lejos o cerca se encuentre el nodo destino de nuestro mensaje, además que se debe de implementar alguna clase de método para evitar que el mensaje se quede en bucle a través de la red.
- El trabajo en equipo fue fundamental para el éxito de nuestro laboratorio 3. La colaboración permitió abordar los problemas desde diferentes ángulos, combinando habilidades y conocimientos de todos los miembros del grupo. Además, la integración de diferentes componentes, como la GUI y la lógica de los algoritmos, fue posible gracias a la coordinación efectiva entre los miembros del equipo. Este laboratorio ayudó a reforzar nuestra capacidad para trabajar de manera colaborativa.
- Algo que se debe de tener en cuenta al momento de implementar un algoritmo como LSR, utilizando Dijkstra, es que muchas veces los pesos no están explícitamente determinados, y muchas veces no solo podemos asumir que todos los nodos poseen una distribución uniforme. Una opción a tomar en cuenta sería ir calculando los pesos basándonos en los tiempos de respuesta.

Comentarios sobre la práctica grupal

Esta práctica grupal nos permitió colaborar entre nosotros de manera efectiva donde cada uno contribuyó con sus conocimientos para poder completar esta práctica. Trabajar juntos en la implementación y pruebas de los algoritmos Flooding y LSR no sólo profundizó nuestra comprensión técnica, sino que también nos enseñó la importancia de la comunicación entre el servidor xmpp y cómo se distribuyen los paquetes hacia los demás nodos. Al realizar este laboratorio nos interesó como estos algoritmos hacen para comunicarse y enviar los paquetes hacia los demás nodos que son nuestros vecinos. Aparte estamos experimentando el trabajo en equipo lo cual esto nos servirá para los demás proyectos que se nos presenten y ayudarnos en el ámbito laboral.



Referencias

- Prezi, O. A. C. O. (s. f.). Algoritmo de inundación(Flooding). prezi.com.
<https://prezi.com/4zf2viwha1a /algoritmo-de-inundacionflooding/>.
- Colaboradores de Wikipedia. (2024, 27 enero). Optimized link state routing. Wikipedia, la Enciclopedia Libre.
https://es.wikipedia.org/wiki/Optimized_Link_State_Routing..
- Equipo editorial de IONOS. (2022, 31 octubre). ¿Qué es XMPP? Resumen de sus funciones y ventajas. IONOS Digital Guide.
<https://www.ionos.com/es-us/digitalguide/servidores/know-how/xmpp/?srsltid=AfmBOopaHUrVWUfN8GirrH9li-apvMTIE7WvYH1-azT5ZzOkRE20OP2v>.
- ¿Qué es el enrutamiento? - Explicación del enrutamiento de redes - AWS. (s. f.). Amazon Web Services, Inc.
<https://aws.amazon.com/es/what-is/routing/#:~:text=Los%20algoritmos%20de%20enrutamiento%20son,con%20varias%20m%C3%A9tricas%20de%20red..>
- ¿Qué es un Algoritmo de Enrutamiento? - Términos y Definiciones de Ciberseguridad. (s. f.).
<https://www.vpnunlimited.com/es/help/cybersecurity/routing-algorithm>.