Guatemala 29 de Agosto de 2024
Jimena Hernandez - 21199
Javier Ramírez - 21600
Mark Albrand - 21004

Redes

Laboratorio 3 - Algoritmos de Enrutamiento

Descripción de la Práctica

En este laboratorio, se analizarán los algoritmos de enrutamiento y sus aplicaciones prácticas en un entorno simulado utilizando el protocolo XMPP. El enfoque principal es entender cómo las tablas de enrutamiento gestionan y optimizan el flujo de datos en una red dinámica, adaptándose a cambios estructurales. Implementaremos y probaremos dos algoritmos: Flooding y Link State Routing. Cada uno de estos algoritmos será ejecutado en un nodo específico dentro de nuestra red simulada en alumchat.lol, permitiendo a cada nodo actuar como un cliente que puede enviar y recibir mensajes.

Descripción de los algoritmos

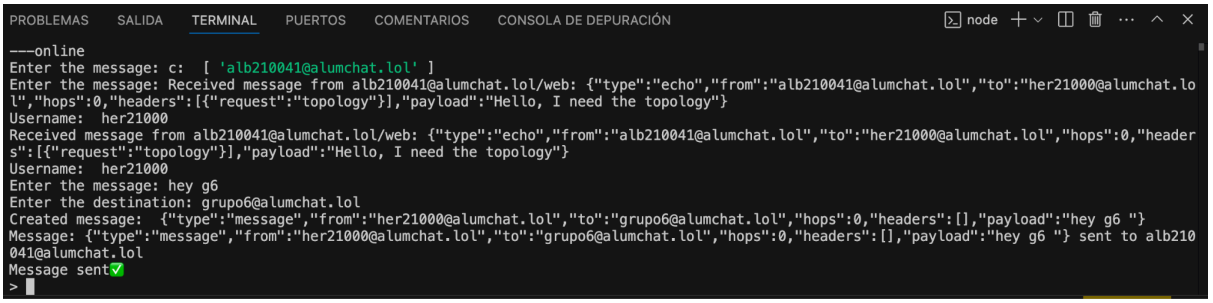
Como se mencionó anteriormente, los algoritmos a implementar serán Flooding y Link State Routing. El primero consiste en un algoritmo de enrutamiento simple donde cada nodo retransmite los paquetes/mensajes entrantes a todos sus enlaces salientes, excepto por el que recibió el paquete, aprovecha todas las rutas posibles y encuentra la más corta naturalmente. Cabe mencionar que, este algoritmo puede resultar en un uso ineficiente del ancho de banda y en la duplicación de mensajes, lo que aumenta la carga en la red. En el segundo algoritmo, cada router construye un mapa de la conectividad de la red, que utiliza para calcular rutas más cortas utilizando el algoritmo de Dijkstra, esto permite una rápida convergencia y adaptación ante cambios en la red, pero requiere más uso de memoria y capacidad de procesamiento. Tanenbaum, Andrew S.; Wetherall, David J. (March 23, 2010).

Implementación

Ambos algoritmos fueron implementados usando javascript, para la implementación del algoritmo de flooding, al estar en línea, se cargan los contactos y permite al usuario enviar mensajes, los cuales son reenviados a todos los contactos si no encuentran directamente al destinatario. Además, se analiza y envía mensajes basados en la interacción del usuario y actualiza dinámicamente las rutas de mensajes según la topología de la red. Para la implementación de Link State Routing, al conectarse, se solicita y procesa la topología de la red para luego permitir el envío de mensajes. Se implementa el algoritmo link state para

determinar rutas óptimas mediante Dijkstra y se actualizan dinámicamente las rutas basadas en la topología recibida.

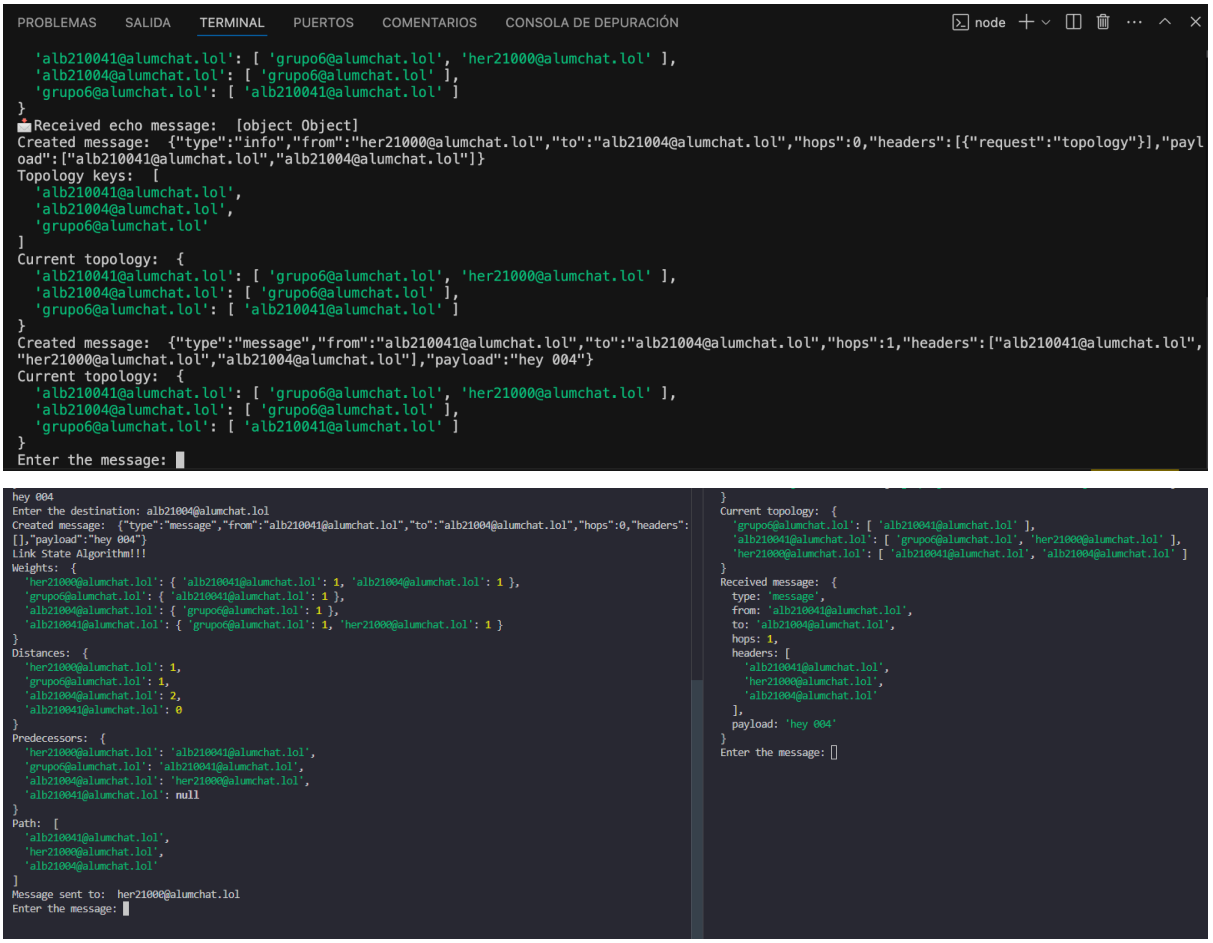
Resultados



Prueba con flooding

```
{
    "type": "names",
    "config": {
        "A": "alb21004@alumchat.lol",
        "B": "grupo6@alumchat.lol",
        "E": "alb210041@alumchat.lol",
        "F": "her21000@alumchat.lol"
    }
}
{
    "type": "topo",
    "config": {
        "A": ["B"],
        "B": ["E"],
        "E": ["B", "F"],
        "F": ["E", "A"]
    }
}
```

Configuración de flooding



Prueba con Link State

Discusión

Para el algoritmo de flooding, el user her21000 envía un mensaje al user grupo6, se puede observar que ya que este contacto no tiene al grupo6 se envía el mensaje a los contactos disponibles de her21000 para que posteriormente envíen el mensaje a sus contactos hasta finalmente llegar al destinatario. Para reducir la redundancia y duplicación de mensajes se realizan comprobaciones antes de enviar y al recibir un mensaje. Antes de enviar se verifica si el destinatario indicado pertenece a la lista de contactos, si es así solo a ese contacto se le envía. Adicionalmente al recibir un mensaje se verifica que antes de reenviarlo no haya sido enviado por la misma cuenta actual.

En el caso del algoritmo de link state, se utilizó la configuración presentada anteriormente. En este caso, se pretende buscar la ruta más corta entre el nodo E al nodo A para enviar un mensaje, para esto existen dos posibles rutas, la primera sería de E a B y regresar a E nuevamente, pero esto no sería lo más óptimo y la segunda, de E a F y luego a A, siendo esta la más corta, se crea el path y se envía el mensaje a F, que posteriormente lo reenvía a A.

Ruta más corta: $E \Rightarrow F \Rightarrow A$

Ruta menos óptima: $E \Rightarrow B \Rightarrow E \Rightarrow \ldots \Rightarrow A$

Con esto se puede comprobar que gracias al uso de Dijkstra se logra encontrar la ruta más corta.

Comentarios

Con la implementación de estos algoritmos se logró identificar que a pesar de que flooding es más sencillo, este puede ocasionar distintos problemas, esto debido al uso ineficiente del ancho de banda y en la duplicación de mensajes. Consideramos que el algoritmo de Link state es mejor ya que no tiene las debilidades mencionadas anteriormente. Este laboratorio nos permitió comprender ambos algoritmos a profundidad y las ventajas y desventajas que ambos pueden tener.

Conclusiones

- Link State es un algoritmo que necesita una robusta sincronización entre clientes (nodos) para poder operar correctamente.
- El protocolo XMPP no es óptimo para enviar mensajes de “solicitudes”, ya que no es posible esperar directamente a la respuesta de un mensaje específico solicitando alguna información.
- Se determinó que la dificultad del algoritmo de link state es mayor pero es más eficiente al no presentar una duplicación de mensajes.

Referencias

Tanenbaum, Andrew S.; Wetherall, David J. (March 23, 2010). Computer Networks (5th ed.). Pearson Education.

Repositorio

https://github.com/Kojimena/Redes-L3.git