

Universidad del Valle de Guatemala

Facultad de Ingeniería



Redes

Laboratorio #3 Parte 2

Alejandro José Gómez Hernández

Gabriel Alejandro Vicente Lorenzo

Oscar Oswaldo Estrada Morales

## **Introducción:**

El objetivo de esta práctica se centró en la aplicación práctica de la parte 1 del mismo laboratorio, en vez de estar realizando de manera manual los cambios del json y ejecutando archivo por archivo la implementación, de tal forma que se entendiera el paso de información entre nodos y mensajes en sí, en esta ocasión los nodos son los encargados de este proceso manual en el que se va compartiendo la información entre nodos de tal manera que se conozcan los hijos de todos y sucesivamente enviando mensajes.

## **Antecedentes:**

En esta ocasión, los antecedentes varían desde librerías en distintos lenguajes, implementaciones de algoritmos según necesidades o variaciones. Además, de formas ingeniosas de trabajar con el algoritmo. En esta ocasión se trató de cumplir con una vasta investigación práctica de tal manera que las implementaciones en los distintos lenguajes sean funcionales.

### **Algoritmo de Flooding:**

El algoritmo de Flooding es una técnica básica de enrutamiento en redes de comunicación. Funciona enviando un paquete recibido a todos los nodos vecinos, excepto al nodo emisor original. Aunque simple, presenta desafíos como tráfico redundante y posibles bucles infinitos. Su funcionamiento consta de inicialización, retransmisión y prevención de bucles.

Sus ventajas incluyen robustez, ya que garantiza que un paquete llegará al destino si existe al menos un camino viable. También es útil para descubrir la topología de la red y sus nodos.

Sin embargo, tiene desventajas notables. Genera tráfico redundante que puede congestionar la red y desperdiciar recursos. Además, gestionar problemas como duplicados y bucles infinitos añade complejidad.

Aunque el flooding puro es ineficiente en redes modernas, ha servido de base para algoritmos más avanzados. Por ejemplo, se controla en protocolos de descubrimiento de servicios y enrutamiento basado en vectores de distancia, donde se utiliza de manera controlada para distribuir información.

En resumen, el algoritmo de Flooding es una técnica fundamental en enrutamiento de red, aunque su uso directo se limita debido a su ineficiencia y problemas asociados, pero ha influido en el desarrollo de algoritmos más avanzados en el campo de las comunicaciones.

### **Algoritmo de Distance Vector Routing:**

El algoritmo de Distance Vector Routing es un enfoque fundamental en el enrutamiento de redes, aunque ha sido superado por algoritmos más eficientes en redes modernas. Su funcionamiento se basa en que cada nodo mantiene una tabla de enrutamiento con información sobre distancias y rutas hacia otros nodos, actualizándola mediante el intercambio de datos con vecinos.

El proceso inicia con la inicialización de tablas, asumiendo distancias infinitas salvo a sí mismo. Luego, los nodos vecinos comparten información de enrutamiento, y cada nodo calcula las distancias más cortas a otros nodos usando el algoritmo de Bellman-Ford. Si encuentra una ruta más corta, actualiza su tabla de enrutamiento.

Entre las ventajas del algoritmo se encuentra su simplicidad y rápida convergencia en redes pequeñas y bien administradas. Sin embargo, presenta desventajas notables, como el "conteo a la infinitud," que puede causar bucles si la información de ruta fallida no se propaga a tiempo, y el consumo de ancho de banda debido a las actualizaciones periódicas de tablas de enrutamiento, especialmente en redes grandes.

El algoritmo se ha utilizado en redes locales (LANs) y ha influido en el diseño de protocolos modernos como OSPF y EIGRP. Aunque ha sido reemplazado en la mayoría de las redes de área amplia (WANs), sigue siendo relevante en situaciones específicas y es un concepto esencial en la historia del enrutamiento de redes.

### **Algoritmo de LSR:**

El algoritmo de enrutamiento Link State (LS) es un algoritmo utilizado en redes de computadoras para determinar las rutas más cortas o eficientes dentro de una red. Funciona mediante la recopilación de información sobre el estado de todos los enlaces de red y luego calcula las rutas óptimas utilizando esta información. El proceso consta de varios pasos:

Descubrimiento de la topología: Cada router en la red recopila información sobre sus enlaces y los estados de esos enlaces.

Construcción de una base de datos de estado de enlace: Cada router crea una base de datos que contiene información sobre la topología de la red, incluyendo la identidad de los routers vecinos y la métrica de los enlaces.

Cálculo de las rutas más cortas: Utilizando el algoritmo de Dijkstra o algún otro algoritmo de cálculo de rutas, cada router determina las rutas más cortas hacia todos los destinos posibles en la red.

Actualización de las tablas de enrutamiento: Cada router utiliza la información calculada para actualizar sus tablas de enrutamiento, lo que le permite tomar decisiones informadas sobre cómo enviar paquetes de datos.

Distribución de la información actualizada: Los routers intercambian información de estado de enlace para mantenerse al tanto de los cambios en la red.

El algoritmo LS es escalable y converge rápidamente, pero también requiere un mayor uso de recursos de memoria y procesamiento en comparación con otros enfoques de enrutamiento. Sin embargo, proporciona una visión precisa de la topología de la red y es ampliamente utilizado en redes grandes y complejas.

### Topología:

Para el grupo en cuestión, se fue dirigida la Topología 1 para el laboratorio 3 y 4 que tiene el siguiente formato en json “{“type”:“topo”, “config”:{‘A’: [‘B’, ‘E’, ‘F’, ‘G’], ‘B’: [‘A’, ‘C’, ‘F’, ‘H’], ‘C’: [‘B’], ‘D’: [‘G’, ‘H’], ‘E’: [‘A’, ‘T’], ‘F’: [‘A’, ‘B’, ‘G’, ‘T’], ‘G’: [‘A’, ‘D’, ‘F’, ‘T’], ‘H’: [‘B’, ‘D’], ‘T’: [‘E’, ‘F’, ‘G’]} }”

En esta se pueden apreciar los vecinos que posee cada nodo, haciendo posible la recopilación de información de cada nodo y cómo estas lo intercambian de tal manera que poco a poco cada nodo posee información pertinente de sus vecinos.

Esta topología puede cambiar, y esta asociada a una dirección concreta de xmpp del servidor de @alumchat.xyz. Lo cual es la base para los demás algoritmos lo que permite interactuar de una manera más fluida.

### Envíos:

Cómo se explicó brevemente, la topología en esta segunda parte de la implementación corresponde a la estructura general que se posee o la red en sí. Pero cada uno solo posee acceso a los vecinos correspondientes, obteniendo la información de los demás nodos cuando estos los comparten entre ellos, haciendo posible la actualización de la tabla de enrutamiento y de esta manera aplicar los algoritmos para encontrar la ruta más fácil y sacar la mejor implementación posible.

## Distance Vector

```
PS C:\Users\carev\OneDrive\Escritorio\Distance Vector> mvn exec:java -D"exec.mainClass"="org.example.XMPPMain"
>>
[INFO] Scanning for projects...
[INFO] -----< org.example:ProyectoRedes >-----
[INFO] Building ProyectoRedes 1.0-SNAPSHOT
[INFO] from pom.xml
[INFO] -----[ jar ]-----
[INFO] --- exec:3.1.0:java (default-cli) @ ProyectoRedes ---
Successful connection to the XMPP server.

---DISTANCE VECTOR---
Seleccione un nodo de la lista:
1. alfagalumchat.xyz
2. bettagalumchat.xyz
3. charliegalumchat.xyz
4. deltagalumchat.xyz
5. epsilongalumchat.xyz
6. figalumchat.xyz
7. gegalumchat.xyz
```

## LSR

```
0.) A
1.) B
2.) C
3.) D
4.) E
5.) F
6.) G
7.) H
8.) I

--> 0
Utilizas la cuenta --> vic1@alumchat.xyz

A


|      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|
| 0    | 1    | 9999 | 9999 | 1    | 1    | 1    | 9999 | 9999 |
| 1    | 0    | 1    | 9999 | 9999 | 1    | 9999 | 1    | 9999 |
| 9999 | 1    | 0    | 9999 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 9999 | 9999 | 9999 | 0    | 9999 | 9999 | 1    | 1    | 9999 |
| 1    | 9999 | 9999 | 9999 | 0    | 9999 | 9999 | 9999 | 1    |
| 1    | 1    | 9999 | 9999 | 9999 | 0    | 1    | 9999 | 1    |
| 1    | 9999 | 9999 | 1    | 9999 | 1    | 0    | 9999 | 1    |
| 9999 | 1    | 9999 | 1    | 9999 | 9999 | 9999 | 0    | 9999 |
| 9999 | 9999 | 9999 | 9999 | 1    | 1    | 1    | 9999 | 0    |


Menú:
1. Matriz
2. Enviar mensaje
3. Salir
Seleccione una opción (1/2/3): 3
Saliendo del programa.

8.) I
--> 1
Utilizas la cuenta --> vic2@alumchat.xyz

B


|      |      |      |      |      |      |      |      |      |
|------|------|------|------|------|------|------|------|------|
| 0    | 1    | 9999 | 9999 | 1    | 1    | 1    | 9999 | 9999 |
| 1    | 0    | 1    | 9999 | 9999 | 1    | 9999 | 1    | 9999 |
| 9999 | 1    | 0    | 9999 | 9999 | 9999 | 9999 | 9999 | 9999 |
| 9999 | 9999 | 9999 | 0    | 9999 | 9999 | 1    | 1    | 9999 |
| 1    | 9999 | 9999 | 9999 | 0    | 9999 | 9999 | 9999 | 1    |
| 1    | 1    | 9999 | 9999 | 9999 | 0    | 1    | 9999 | 1    |
| 1    | 9999 | 9999 | 1    | 9999 | 1    | 0    | 9999 | 1    |
| 9999 | 1    | 9999 | 1    | 9999 | 9999 | 9999 | 0    | 9999 |
| 9999 | 9999 | 9999 | 9999 | 1    | 1    | 1    | 9999 | 0    |


Menú:
1. Matriz
2. Enviar mensaje
3. Salir
Seleccione una opción (1/2/3): 2
in
Menú:
1. Matriz
2. Enviar mensaje
3. Salir
Seleccione una opción (1/2/3): 3
Saliendo del programa.
{"type": "message", "headers": {"from": "A", "to": "B", "hop_count": 1}, "payload": "hol
a"}
reenviando mensaje
```

## Conclusiones

- Todos los algoritmos de enrutamiento tienen situaciones específicas que benefician más a su comportamiento. Obteniendo mejores resultados al obtener el camino más corto o eficiente.

- La parte difícil de compartir la topología de manera subdividida es la sincronización de estas actualizaciones, haciendo esencial la manipulación de mensajes entre nodos y sus intercomunicaciones, lo cual puede ser un reto.
- Distance vector tuvo la característica de ser una implementación sumamente compleja y completa debido a su delicada precaución al momento de que los nodos se comuniquen entre sí, debido a que su complejidad va aumentando gradualmente pero entre una considerable cantidad de intercomunicaciones.
- Link State Routing tenía la peculiaridad de tener hilos complejos y algo difíciles de comunicar debido a su compleja forma de pasar información dando toques y luego pasando la información. Lo que hacía la implementación realizada algo engorrosa pero funcional.
- El laboratorio con la parte funcional de la tabla de enrutamiento de una topología es un concepto esencial para entender la manera general en que las redes funcionan, y la comprensión de esta es de alto nivel debido a que alta complejidad al momento de implementarlo, debido a la baja documentación y puntos vulnerables a loops infinitos.

## Referencias

Kurose, J. F., & Ross, K. W. (2020). Computer networking: A top-down approach (8th ed.). Pearson.

Marina, M. K., & Das, S. R. (2001, November). On-demand multipath distance vector routing in ad hoc networks. In Proceedings ninth international conference on network protocols. ICNP 2001 (pp. 14-23). IEEE.

Smith, B. R., Murthy, S., & Garcia, J. J. (1997). Securing distance-vector routing protocols. In Proceedings of SNDSS'97: Internet Society 1997 Symposium on Network and Distributed System Security (pp. 85-92). IEEE.

Tanenbaum, A. S., & Wetherall, D. J. (2011). Computer networks (5th ed.). Prentice Hall.

VGA. Un visualizador genérico de algoritmos. El Algoritmo de Dijkstra o de Caminos mínimos. (2023). Uned.es.

<http://atlas.uned.es/algoritmos/voraces/dijkstra.html#:~:text=El%20algoritmo%20de%20Dijkstra%2C%20tambi%C3%A9n,con%20pesos%20en%20cada%20arista.>