

# Laboratorio 3 – Algoritmos de Enrutamiento

## Parte 3.1: Implementación de Algoritmos

Andy Fuentes

Davis Roldán

Diego Linares

Universidad del Valle de Guatemala

Facultad de Ingeniería

Curso: Redes de Computadoras

## Índice

<b>1. Antecedentes</b>	<b>2</b>
<b>2. Objetivos</b>	<b>2</b>
<b>3. Algoritmos Implementados</b>	<b>2</b>
3.1. Flooding . . . . .	2
3.2. Distance Vector (DV) . . . . .	2
3.3. Link State Routing (LSR) . . . . .	2
<b>4. Pruebas y Resultados</b>	<b>2</b>
<b>5. Conclusiones</b>	<b>3</b>

# 1. Antecedentes

El laboratorio 3 del curso de Redes de Computadoras consiste en implementar distintos algoritmos de enrutamiento sobre una red simulada. Cada nodo de la red se levanta como un proceso independiente y se comunica mediante *Sockets TCP*. Los algoritmos permiten que los nodos descubran rutas, actualicen tablas y transmitan mensajes de forma eficiente.

# 2. Objetivos

- Conocer el funcionamiento de algoritmos de enrutamiento clásicos en redes.
- Implementar los algoritmos **Flooding**, **Distance Vector** y **Link State Routing (LSR)**.
- Probar los algoritmos en una red simulada localmente mediante sockets TCP.
- Analizar el comportamiento de las tablas de enrutamiento en cada protocolo.

# 3. Algoritmos Implementados

## 3.1. Flooding

El algoritmo de **Flooding** consiste en reenviar un paquete recibido a todos los vecinos excepto al emisor. Se implementó un control de duplicados mediante un identificador único por mensaje. Esto garantiza que un mensaje llegue a todos los nodos de la red.

## 3.2. Distance Vector (DV)

El algoritmo de **Vector de Distancia** se basa en que cada nodo comparte periódicamente con sus vecinos su vector de distancias conocido. Se implementó con el método de Bellman-Ford distribuido. Cada nodo mantiene la mejor ruta conocida hacia cada destino y actualiza su tabla al recibir información de sus vecinos.

## 3.3. Link State Routing (LSR)

El protocolo de **Estado de Enlace** utiliza el envío de *Link State Packets (LSP)* hacia toda la red mediante flooding. Cada nodo construye un grafo de la topología y aplica el algoritmo de Dijkstra para calcular las rutas más cortas. En la implementación, los nodos reanuncian periódicamente su LSP y actualizan sus tablas cuando detectan cambios.

# 4. Pruebas y Resultados

Para probar los algoritmos se levantaron tres nodos A, B y C con las siguientes configuraciones:

```
A 127.0.0.1 5001
B 127.0.0.1 5002
C 127.0.0.1 5003
```

La topología utilizada fue:

```
A B 1
A C 4
B C 2
```

Se realizaron pruebas de envío de mensajes con el script `send_data.py`:

```
# Mensaje en Flooding
python send_data.py 127.0.0.1 5001 A C flooding "Hola_desde_A_a_C"

# Mensaje en Distance Vector
python send_data.py 127.0.0.1 5002 B C dv "Mensaje_DV_test"

# Mensaje en Link State Routing
python send_data.py 127.0.0.1 5001 A C lsr "Mensaje_LSR_test"
```

Los resultados mostraron la entrega exitosa de mensajes en cada protocolo, incluyendo el rastro (*trace*) de los saltos.

## 5. Conclusiones

- Se completó la implementación de los tres algoritmos de enrutamiento requeridos en la fase 3.1: Flooding, Distance Vector y Link State Routing.
- Los nodos pudieron intercambiar mensajes y actualizar sus tablas de enrutamiento dinámicamente.
- Esta fase proporciona la base para la siguiente etapa del laboratorio, que consistirá en conectar los algoritmos sobre un servidor XMPP.