

# ALGORITMOS DE ENRUTAMIENTO

## LABORATORIO 3 Y 4

### LINK DEL REPOSITORIO

<https://github.com/JosePabloPonce/laboratorio-3-redes.git>

### DESCRIPCIÓN DE LA PRÁCTICA

Este laboratorio consiste en la implementación y comprensión de los algoritmos de enrutamiento, los cuales funcionan con nodos conectados entre sí y que únicamente conocen a sus nodos vecinos. La primera parte consiste en todo el proceso de implementación, el cual utiliza alumchat.fun como sistema de nodos con cada algoritmo, además de poder ejecutarse en distintas plataformas como lo son Windows, Mac y Linux. Los algoritmos a implementar son: flooding, distance vector routing y link state routing, de los cuales se hablará más adelante. La implementación de estos algoritmos cuenta con alguna de las siguientes funcionalidades: inicio de sesión con la cuenta del alumno, solicitud del nodo asignado al carnet, elección del algoritmo a ejecutar, carga de información que cada algoritmo requiere, cálculo de rutas, visualización de la tabla de ruteo, indica de qué nodo recibe y a qué nodo envía la información.

La segunda parte del laboratorio es poner a prueba la implementación de los algoritmos, para esto se asigna a cada miembro del grupo una dirección que contiene un usuario y contraseña para realizar la conexión, la cual va ser monitoreada. Todos los nodos deben contar con la capacidad de enviar y transmitir un mensaje.

### ALGORITMOS

#### Flooding

El funcionamiento de este algoritmo sucede cuando un paquete de datos llega a un enrutador y este los envía a todos los enlaces salientes excepto al que llegó. Entre las ventajas de este algoritmo se encuentra que es muy fácil de implementar y configurar, ya que el enrutador conoce a sus vecinos, lo que hace que se visiten todos ya sea directa o indirectamente, este es un algoritmo que siempre elige la ruta más corta.

## Distance Vector Routing

A este algoritmo se le conoce también como el algoritmo de Bellman-Ford, emplea un modelo de red mediante un grafo, en donde los nodos representan los routers y las aristas los enlaces de transmisión. Es un algoritmo que calcula sus rutas progresivamente y va mejorándolas en cada iteración ya que los cálculos se realizan parcialmente en cada nodo.

## Link State Routing

Este algoritmo es un protocolo interno utilizado por cada router para compartir información sobre el resto de routers que se encuentran en la red. Cada router crea su tabla de enrutamiento por medio de la topología de red y la intercambia con todos los routers que se encuentran en la red lo que hace que la entrega de información sea más rápida y confiable. Este algoritmo se divide en dos fases: la primera es flooding en donde cada nodo conoce el costo de sus vecinos y también el gráfico completo. La segunda fase es el cálculo de enrutamiento en donde cada nodo utiliza el algoritmo de Dijkstra para calcular las rutas óptimas hacia los otros nodos.

## RESULTADOS

### Topología utilizada en todos los algoritmos

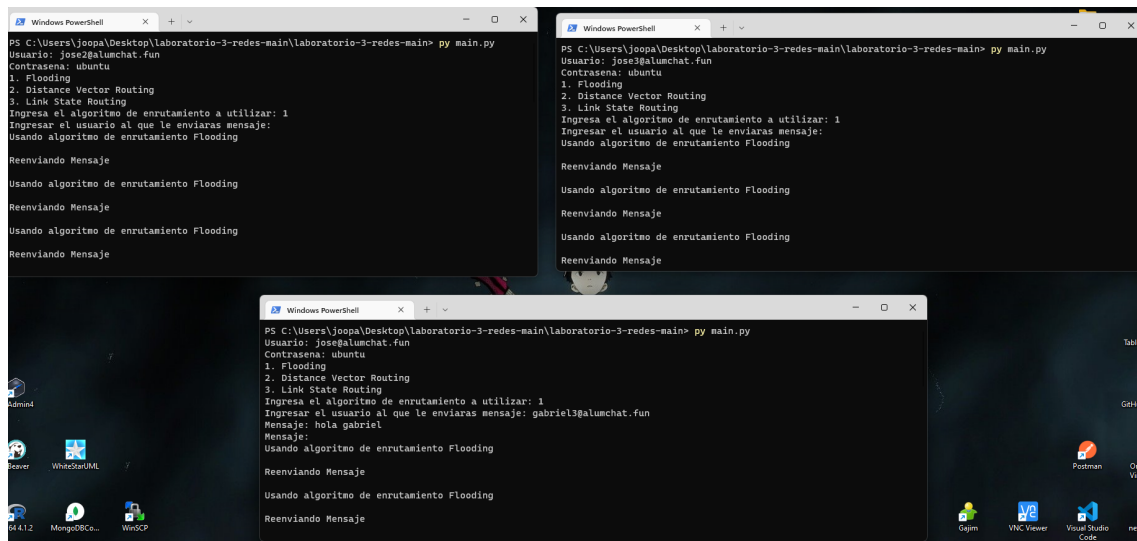
```
{
  "type": "names",
  "config": {
    "A": "jose@alumchat.fun",
    "B": "jose2@alumchat.fun",
    "C": "jose3@alumchat.fun",
    "D": "gabriel2@alumchat.fun",
    "E": "gabriel3@alumchat.fun",
    "F": "qui19255@alumchat.fun"
  },
}
```

```
{
  "type": "topo",
  "config": {
    "A": ["B", "D", "F"],
    "B": ["A", "C", "D", "F"],
    "C": ["B", "E", "F"],
    "D": ["A", "B"],
    "E": ["C", "F"],
    "F": ["A", "B", "C", "E"]}
}
```

El envío de mensajes para probar la funcionalidad de los algoritmos se realizó del nodo A hacia el nodo E.

## Flooding

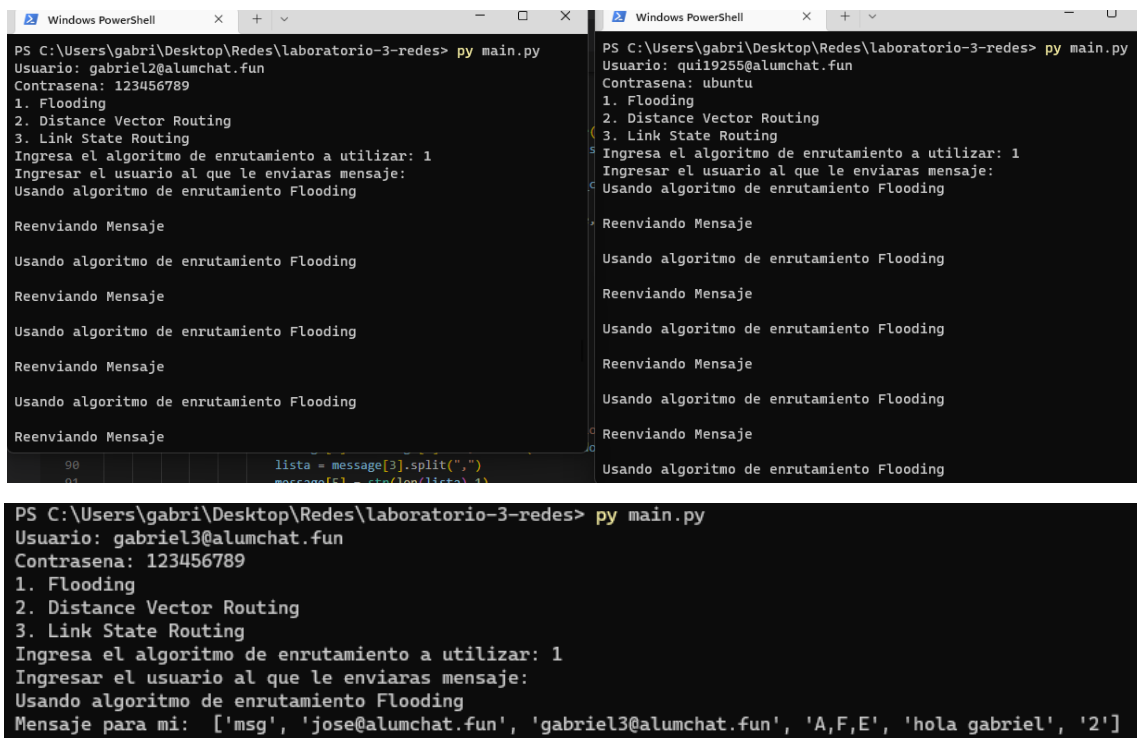
### Envío del mensaje



The image shows three overlapping Windows PowerShell windows. Each window displays the output of a Python script named 'main.py'. The script prompts for a username (jose3@alumchat.fun), a password (ubuntu), and a routing algorithm (1 for Flooding). It then repeatedly sends a message 'hola gabriel' using the Flooding algorithm. The output in each window is identical, showing the sequence of actions: 'Reenviando Mensaje', 'Usando algoritmo de enrutamiento Flooding', and 'Reenviando Mensaje'.

```
PS C:\Users\joopa\Desktop\laboratorio-3-redes-main\laboratorio-3-redes-main> py main.py
Usuario: jose3@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 1
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
```

### Mensaje Recibido



The image shows three overlapping Windows PowerShell windows. Each window displays the output of a Python script named 'main.py'. The script prompts for a username (gabriel2@alumchat.fun), a password (123456789), and a routing algorithm (1 for Flooding). It then repeatedly receives a message 'hola gabriel' using the Flooding algorithm. The output in each window is identical, showing the sequence of actions: 'Reenviando Mensaje', 'Usando algoritmo de enrutamiento Flooding', and 'Reenviando Mensaje'.

```
PS C:\Users\gabri\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: gabriel2@alumchat.fun
Contraseña: 123456789
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 1
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
Usando algoritmo de enrutamiento Flooding
Reenviando Mensaje
```

## Distance Vector Routing

### Envío del mensaje

```
PS C:\Users\joopa\Desktop\laboratorio-3-redes-main\laboratorio-3-redes-main> py main.py
Usuario: jose3@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 2
Ingresa el usuario al que le enviaras mensaje:

PS C:\Users\joopa\Desktop\laboratorio-3-redes-main\laboratorio-3-redes-main> py main.py
Usuario: jose3@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 2
Ingresa el usuario al que le enviaras mensaje: gabriel3@alumchat.fun
Mensaje: hola gabriel3

Camino:
['A', 'P', 'E']
msg[jose3@alumchat.fun[gabriel3@alumchat.fun]A]hola gabriel3[8]F E

Reenviando a: qui19255@alumchat.fun
Mensaje:
```

### Mensaje Recibido

```
self.run_forever()
File "C:\Users\gabri\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 596, in run_forever
self._run_once()
File "C:\Users\gabri\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 1854, in _run_once
event_list = self._selector.select(timeout)
File "C:\Users\gabri\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 324, in select
r, w, _ = self._select(self._readers, self._writers, [], timeo
ut)
File "C:\Users\gabri\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 315, in _select
r, w, x = select.select(r, w, w, timeout)
KeyboardInterrupt
PS C:\Users\gabri\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: gabriel2@alumchat.fun
Contraseña: 123456789
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 2
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Distance Vector
Reenviando a: gabriel3@alumchat.fun
Mensaje Enviado

PS C:\Users\gabri\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: gabriel3@alumchat.fun
Contraseña: 123456789
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 2
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Distance Vector
Mensaje para mi: ['msg', 'jose@alumchat.fun', 'gabriel3@alumchat.fun', 'A,F,E', 'hola gabriel3', '2', 'E']
```

## Link State Routing

### Envío del mensaje

```
PS C:\Users\joopa\Desktop\laboratorio-3-redes-main\laboratorio-3-redes-main> py main.py
Usuario: jose2@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 3
Ingresa el usuario al que le enviaras mensaje:

PS C:\Users\joopa\Desktop\laboratorio-3-redes-main\laboratorio-3-redes-main> py main.py
Usuario: jose3@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 3
Ingresa el usuario al que le enviaras mensaje: gabriel3@alumchat.fun
Mensaje: hola gabriel3
Mensaje:
```

### Mensaje Recibido

```
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 596, in run_forever
    self._run_once()
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 1854, in _run_once
    event_list = self._selector.select(timeout)
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 324, in select
    r, w, _ = self._select(self._readers, self._writers, [], timeo
ut)
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 315, in _select
    r, w, x = select.select(r, w, w, timeout)
KeyboardInterrupt
PS C:\Users\gabriel\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: gabriel2@alumchat.fun
Contraseña: 123456789
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 3
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Link State Routing
Mensaje para mi: ['msg', 'jose@alumchat.fun', 'gabriel3@alumchat.fun', 'A,F,E', 'hola gabriel3', '2']

File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 596, in run_forever
    self._run_once()
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
asyncio\base_events.py", line 1854, in _run_once
    event_list = self._selector.select(timeout)
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 324, in select
    r, w, _ = self._select(self._readers, self._writers, [], timeo
ut)
File "C:\Users\gabriel\AppData\Local\Programs\Python\Python39\lib\
selectors.py", line 315, in _select
    r, w, x = select.select(r, w, w, timeout)
KeyboardInterrupt
PS C:\Users\gabriel\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: quili9255@alumchat.fun
Contraseña: ubuntu
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 3
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Link State Routing
Camino: ['F', 'E']

PS C:\Users\gabriel\Desktop\Redes\laboratorio-3-redes> py main.py
Usuario: gabriel3@alumchat.fun
Contraseña: 123456789
1. Flooding
2. Distance Vector Routing
3. Link State Routing
Ingresa el algoritmo de enrutamiento a utilizar: 3
Ingresa el usuario al que le enviaras mensaje:
Usando algoritmo de enrutamiento Link State Routing
Mensaje para mi: ['msg', 'jose@alumchat.fun', 'gabriel3@alumchat.fun', 'A,F,E', 'hola gabriel3', '2']
```

## DISCUSIÓN

Al realizar el laboratorio se cumplieron los objetivos establecidos porque por medio de la implementación se conocieron los algoritmos de enrutamiento utilizados en las

implementaciones actuales del internet, además del conocimiento sobre el funcionamiento de las tablas de ruteo.

En los resultados se puede observar el funcionamiento de cada uno de los algoritmos en donde se incluye el nodo fuente, nodo destino, los nodos recorridos y el mensaje enviado, qué forman parte de los datos requeridos para realizar las pruebas de implementación.

Se realizó la implementación de los tres algoritmos en un script para qué sea más fácil su uso, en donde se cuenta con un menú general en donde se puede escoger cualquiera de los tres algoritmos qué se quiere utilizar, los datos requeridos son el usuario destino y el mensaje a enviar.

Primero tenemos el algoritmo flooding el cual envía el mensaje a todos sus nodos vecinos, por lo qué se puede observar en las imágenes "Reenviando mensaje" en todos los usuarios(nodos), porque pasa por todos hasta llegar a su usuario destino, qué es quien puede visualizarlo.

Por otro lado tenemos los resultados del algoritmo Distance Vector Routing en donde se puede observar qué no le notifica a los demás usuarios el envío del mensaje, a menos qué sean parte del camino más corto qué el mismo algoritmo determinó. El mensaje llega al usuario destino, quien lo puede visualizar.

Por último se encuentran los resultados del algoritmo Link State Routing, en donde se puede observar qué tampoco le notifica a ningún otro usuario(nodo) del envío de un mensaje, sólo al qué es parte de la ruta más corta, al cual le notifica el camino qué está recorriendo, hasta llegar al nodo vecino el cual puede visualizar el mensaje.

## **COMENTARIO GRUPAL**

La implementación de los algoritmos fue un reto interesante, tuvimos algunas complicaciones pero creemos que todo se resolvió lo mejor posible y gracias a este laboratorio comprendimos mejor el funcionamiento de los algoritmos de ruteo qué se utilizan en las implementaciones actuales de Internet y también sobre las tablas de ruteo.

## **CONCLUSIONES**

- Se realizó con éxito la implementación de los tres algoritmos con los datos requeridos.
- El algoritmo flooding es el más sencillo de implementar porque se envía el mensaje a todos, lo qué puede verse como una desventaja porque el

proceso de envío es más largo comparado con los otros dos algoritmos que buscan la ruta más corta para enviar el mensaje.

- El protocolo XMPP forma parte importante para la implementación de los algoritmos de ruteo ya que se caracteriza por brindar interconexiones de mensajería instantánea.
- Con el paso del tiempo se han ido implementando y actualizando los algoritmos de ruteo utilizados para la implementación del internet, lo que permite que brinden un mejor servicio.

## REFERENCIAS

JavaTpoint (2011-2021). Link State Routing. Extraído de:  
<https://www.javatpoint.com>

Universidad de Almería (2022). Routing Distance Vector. Extraído de:  
<https://sites.google.com/ual.es/hpca>