

Universidad del Valle de Guatemala

Redes
Sección 20



Excelencia que trasciende

DEL VALLE
GRUPO EDUCATIVO

Libro de códigos

Diego Lemus - 21469
Fabián Juárez - 21440
Daniel Gomez - 21429

29 de agosto 2024

Link al repositorio

<https://github.com/FabianJuarez182/Lab3-Redes.git>

Descripción del laboratorio

Este laboratorio tiene como objetivo principal la implementación y prueba de algoritmos de enrutamiento en una red simulada utilizando el protocolo XMPP. Los estudiantes deberán comprender el funcionamiento de las tablas de enrutamiento y cómo se actualizan dinámicamente mediante diferentes algoritmos. El laboratorio se divide en dos partes principales: la implementación de los algoritmos de enrutamiento (Flooding y Link State Routing) y la conexión y prueba de estos algoritmos en un entorno simulado utilizando nodos representados por usuarios en un servidor XMPP. Al finalizar, se espera que los algoritmos puedan enrutar mensajes a través de la red de manera eficiente y que sean capaces de adaptarse a cambios en la topología de la red.

Descripción de los algoritmos

1. Flooding: Este algoritmo envía un paquete a todos los nodos vecinos, sin importar su destino final. Cada nodo que recibe el paquete lo reenvía a todos sus vecinos, exceptuando el nodo del cual recibió el mensaje. Este enfoque, aunque sencillo, puede causar redundancia y sobrecarga en la red debido a la duplicación de paquetes.

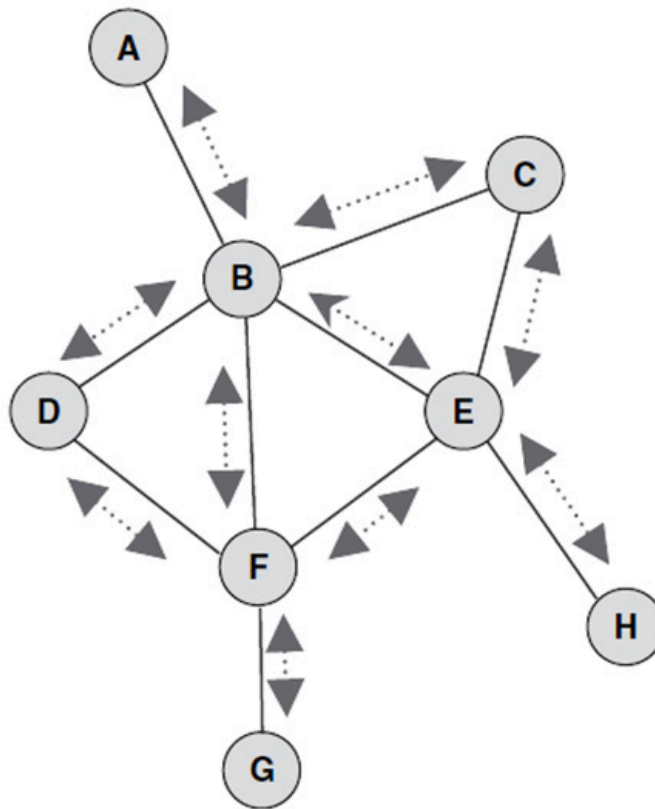


Figura 1: Ejemplo de Flooding.

2. Link State Routing: Este algoritmo utiliza información completa sobre la red para calcular la mejor ruta posible hacia cada destino. Los nodos envían información sobre sus vecinos a todos los demás nodos en la red. Con esta información, cada nodo construye un mapa completo de la red y utiliza el algoritmo de Dijkstra para determinar la ruta más corta hacia cada destino.

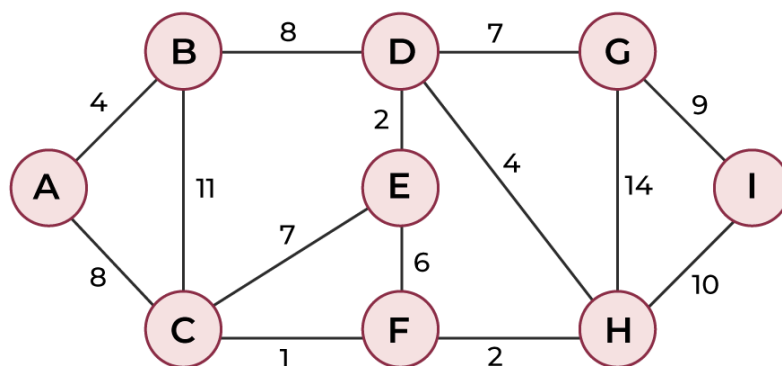


Figura 2: Ejemplo de Link State Routing.

Resultados

En la implementación y prueba de los algoritmos, se observó lo siguiente:

- Flooding resultó ser simple pero ineficiente en términos de uso de ancho de banda, ya que genera múltiples copias de los mismos mensajes, lo que podría causar congestión en la red.

```
maps > {} testjson > ...
1
2 {
3   "type": "map",
4   "routes":
5   {
6     "A": [{"path": "B", "cost": 1}],
7     "B": [{"path": "A", "cost": 1}, {"path": "C", "cost": 1}],
8     "C": [{"path": "A", "cost": 1}]
9   }
10 }
```

Figura 3: Topología de red de prueba.

```
maps > {} nodes.json > ...
1
2 {
3   "type": "tags",
4   "nodes": {
5     "A": { "tag": "A", "user": "lem21469@alumchat.lol", "pass": "lem21469" },
6     "B": { "tag": "B", "user": "lem21469-test@alumchat.lol", "pass": "lem21469-test" },
7     "C": { "tag": "C", "user": "lem21469-f1@alumchat.lol", "pass": "lem21469" },
8     "D": { "tag": "D", "user": "jdgomezv3@alumchat.lol", "pass": "admin" },
9     "E": { "tag": "E", "user": "jdgomezv4@alumchat.lol", "pass": "admin" }
10   }
11 }
```

Figura 4: Asignamiento de nodos.

<pre>PS C:\Users\deloz\OneDrive\Escritorio\Lab3-Redes> node client.js Please enter the node you would like to use (A, B, C, D, E): A f(Connected as lem21469@alumchat.lol/alb2c3) Please enter the command ("flood" or "lsr"): flood Do you want to function as a sender (1) or receiver (2)? ("1" or "2"): 1 You are functioning as a sender. Sending to neighbor: lem21469-test@alumchat.lol/alb2c3 Message originated from this node, ignoring... > {}</pre>	<pre>PS C:\Users\deloz\OneDrive\Escritorio\Lab3-Redes> node client.js Please enter the node you would like to use (A, B, C, D, E): B Connected as lem21469-test@alumchat.lol/alb2c3 Please enter the command ("flood" or "lsr"): flood Do you want to function as a sender (1) or receiver (2)? ("1" or "2"): 2 You are functioning as a receiver. Listening for messages... > Flooding message received from lem21469@alumchat.lol/alb2c3 with payload: "Triggered Flooding Message!" and hops remaining: 3 Propagating message to neighbors... Sending to neighbor: lem21469@alumchat.lol/alb2c3 Sending to neighbor: lem21469-f1@alumchat.lol/alb2c3 > {}</pre>	<pre>PS C:\Users\deloz\OneDrive\Escritorio\Lab3-Redes> node client.js Please enter the node you would like to use (A, B, C, D, E): C Connected as lem21469-f1@alumchat.lol/alb2c3 Please enter the command ("flood" or "lsr"): flood Do you want to function as a sender (1) or receiver (2)? ("1" or "2"): 2 You are functioning as a receiver. Listening for messages... > Flooding message received from lem21469@alumchat.lol/alb2c3 with payload: "Triggered Flooding Message!" and hops remaining: 2 Propagating message to neighbors... Sending to neighbor: lem21469@alumchat.lol/alb2c3 > {}</pre>
---	---	--

Figura 5: Visualización de flooding.

- Link State Routing demostró ser más eficiente, permitiendo que los mensajes se envíen por las rutas más cortas calculadas mediante el algoritmo de Dijkstra. Sin embargo, el requisito de que cada nodo tenga conocimiento completo de la red aumenta la complejidad y el tiempo de convergencia.

```
maps > {} |sr:john > ...
JDgomez2002, 1 minute ago | 1 author (JDgomez2002)
1 {
2   "gom21429@alumchat.lol/a1b2c3": {
3     "jdgomezv1@alumchat.lol/a1b2c3": 1,
4     "jdgomezv2@alumchat.lol/a1b2c3": 2
5   },
6   "jdgomezv1@alumchat.lol/a1b2c3": {
7     "jdgomezv2@alumchat.lol/a1b2c3": 1,
8     "jdgomezv3@alumchat.lol/a1b2c3": 3
9   },
10  "jdgomezv2@alumchat.lol/a1b2c3": {
11    "jdgomezv1@alumchat.lol/a1b2c3": 2,
12    "jdgomezv3@alumchat.lol/a1b2c3": 3
13  },
14  "jdgomezv3@alumchat.lol/a1b2c3": {
15    "jdgomezv1@alumchat.lol/a1b2c3": 2,
16    "jdgomezv4@alumchat.lol/a1b2c3": 2,
17    "jdgomezv2@alumchat.lol/a1b2c3": 3
18  },
19  "jdgomezv4@alumchat.lol/a1b2c3": {
20    "jdgomezv2@alumchat.lol/a1b2c3": 2,
21    "jdgomezv1@alumchat.lol/a1b2c3": 2,
22    "jdgomezv3@alumchat.lol/a1b2c3": 3
23  }
24 }
```

Figura 6: Asignamiento de nodos.

```
jdomez2@MacbookPro Lab3-Redes % node client.js
Please enter the node you would like to use (A, B, C, D, E): B
Connected as jdgomezv1@alumchat.lol/a1b2c3
Please enter the command ("flood" or "lsr#destination"): LSR message received from gom21429@alumchat.lol/a1b2c3 with payload: "LSR Message"
Forwarding LSR message to jdgomezv4@alumchat.lol/a1b2c3 via jdgomezv3@alumchat.lol/a1b2c3 with remaining cost 5
Please enter the command ("flood" or "lsr#destination"): █
```

Figura 7: Visualización de Link State Routing con Nodo 1.

```
PROBLEMS 1 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
jdomez2@MacbookPro Lab3-Redes % node client.js
Please enter the node you would like to use (A, B, C, D, E): C
Connected as jdgomezv2@alumchat.lol/a1b2c3
Please enter the command ("flood" or "lsr#destination"): █
```

Figura 8: Visualización de Link State Routing con Nodo 2.

```
PROBLEMS 1 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
jdomez2@MacbookPro Lab3-Redes % node client.js
Please enter the node you would like to use (A, B, C, D, E): D
Connected as jdgomezv3@alumchat.lol/a1b2c3
Please enter the command ("flood" or "lsr#destination"): LSR message received from jdgomezv1@alumchat.lol/a1b2c3 with payload: "LSR Message"
Forwarding LSR message to jdgomezv4@alumchat.lol/a1b2c3 via jdgomezv2@alumchat.lol/a1b2c3 with remaining cost 2
Please enter the command ("flood" or "lsr#destination"): █
```

Figura 9: Visualización de Link State Routing con Nodo 3.

```
PROBLEMS 1 OUTPUT TERMINAL PORTS COMMENTS DEBUG CONSOLE
jdomez2@MacbookPro Lab3-Redes % node client.js
Please enter the node you would like to use (A, B, C, D, E): E
Connected as jdgomezv4@alumchat.lol/a1b2c3
Please enter the command ("flood" or "lsr#destination"): LSR message received from jdgomezv3@alumchat.lol/a1b2c3 with payload: "LSR Message"
Message received at final destination: LSR Message
Please enter the command ("flood" or "lsr#destination"): █
```

Figura 10: Visualización de Link State Routing con Nodo 4.

Discusión

Cada algoritmo tiene sus ventajas y desventajas dependiendo del contexto en el que se utilice. Flooding es útil en redes pequeñas donde la simplicidad es clave, pero no escala bien. Link State Routing es excelente en redes donde se puede tolerar la sobrecarga de mantener un mapa completo de la red, y es ideal para redes donde la precisión y la rapidez en la convergencia son cruciales. Link State Routing se beneficia demasiado al encontrar las rutas más cortas y permitir que no todos los nodos sean utilizados gracias a la implicación de Dijkstra permitiendo una mayor eficiencia.

El uso del protocolo XMPP permitió simular efectivamente una red de nodos, aunque presentó desafíos adicionales relacionados con la sincronización y la configuración de los nodos. La implementación de los algoritmos requirió una comprensión profunda de cómo se maneja el ruteo en redes, así como la adaptación a un entorno basado en mensajería instantánea como XMPP.

Conclusiones

El laboratorio proporcionó una experiencia práctica valiosa en la implementación y prueba de algoritmos de enrutamiento, demostrando las fortalezas y limitaciones de cada enfoque en un entorno de red simulada. Flooding es fácil de implementar pero no eficiente para redes grandes, Link State Routing es preciso pero complejo.

Comentarios

- El uso de un protocolo como XMPP para la simulación de redes es interesante, pero puede ser complejo en términos de sincronización y manejo de la latencia en la entrega de mensajes.
- Para futuros laboratorios, se podría considerar el uso de herramientas de simulación más especializadas en redes para comparar y validar los resultados obtenidos con XMPP.

Referencias

Wikipedia contributors. (2023). Flooding (computer networking). Wikipedia.
[https://en.wikipedia.org/wiki/Flooding_\(computer_networking\)](https://en.wikipedia.org/wiki/Flooding_(computer_networking))

Flooding in Computer Network. (s. f.).
<https://www.tutorialspoint.com/flooding-in-computer-network>

GeeksforGeeks. (2023). Unicast routing Link State routing. GeeksforGeeks.
<https://www.geeksforgeeks.org/unicast-routing-link-state-routing/>