

Universidad del Valle de Guatemala

Redes

Marcos Gutierrez 17909

David Valenzuela 171001

Fernando Hengstenberg 17699

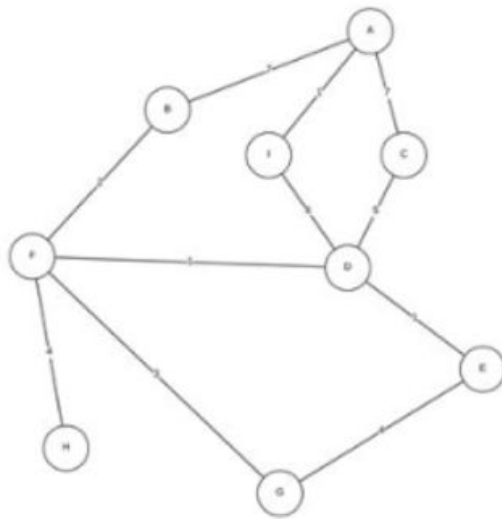
Algoritmos de Enrutamiento

Descripción:

Los algoritmos de enrutamiento funcionan sobre nodos interconectados entre sí, donde cada uno conoce únicamente cuáles son los vecinos que tiene. A partir de ello se estará simulando este tipo de infraestructura utilizando el lenguaje de programación de python para la implementación de la conexión y comunicación de mensajes. Los algoritmos que se desarrollaron se muestran a continuación con su respectiva descripción.

- Flooding.
 - Flooding es un algoritmo de enrutamiento de redes informáticas en el que cada paquete entrante se envía a través de todos los enlaces salientes, excepto del que llegó, Flooding se utiliza en puentes y en sistemas como Uset y el intercambio de archivos de igual a igual, debido a cómo funciona el algoritmo, se garantiza que un paquete va a ser entregado. Es muy fácil de implementar, aunque con un enfoque ineficiente.
 - Ventajas:
 - los routers no están obligados a mantener la base de datos de la información de estado del enlace, tiene un considerable ahorro de espacio de almacenamiento.
 - asegura que el paquete que es enviado sea entregado por la forma en la que está construido el algoritmo.
 - no se realizan cálculos complejos de ruta, por lo tanto se reducen los gastos generales del funcionamiento y complejidad de la aplicación.
- Distance Vector Routing:
 - Es un algoritmo de enrutamiento el cual se basa en el cálculo de la dirección, y el peso de la distancia. Este algoritmo ayuda a encontrar la distancia más corta hacia cualquier nodo que se encuentre en la red. El manejo de la distancia que está dada por tablas de conexiones, el cual almacena la información (peso) de los nodos, cuya información se actualiza cuando hay un cambio en la red
- Link state routing:
 - Este algoritmo de enrutamiento emplea la técnica de flooding para asegurarse que todos los nodos de la red conocen la presencia de los demás nodos. De esta manera nos podemos asegurar que todos los nodos conocen el grafo completo, el cálculo de la distancia más corta se realiza desde el grafo, de esta manera se asegura que el camino recorrido siempre sea el más corto.

diagrama de nodos:



Resultados:

Flooding:
nodos

TERMINAL	PROBLEMAS	SALIDA	CONSOLA DE DEPURACIÓN
<pre> (18072) accepted ('127.0.0.1', 61683) connect 56878e72fdff40a298943dc12bd97256 127.0.0.1 - - [01/Oct/2020 19:46:55] "GET /socket.io/?transport=polling&EIO=3&t=16 01603213.8496234 HTTP/1.1" 200 349 0.0049 91 (18072) accepted ('127.0.0.1', 61687) ----- ['a', 'b'] ['a05b601f036c41acaf5991adcc67bf11', '568 78e72fdff40a298943dc12bd97256'] (18072) accepted ('127.0.0.1', 61694) connect b555033f9ab7401bb51dc783ffed2181 127.0.0.1 - - [01/Oct/2020 19:47:05] "GET /socket.io/?transport=polling&EIO=3&t=16 01603223.6156893 HTTP/1.1" 200 349 0.0009 90 (18072) accepted ('127.0.0.1', 61697) ----- ['a', 'b', 'f'] ['a05b601f036c41acaf5991adcc67bf11', '568 78e72fdff40a298943dc12bd97256', 'b555033f 9ab7401bb51dc783ffed2181'] </pre>		<pre> Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados. Prueba la nueva tecnología PowerShell multi plataforma https://aka.ms/pscore6 PS C:\Users\F. Hengstenberg\Desktop\Cursos\ Semestre 8\Redes\lab3y4\algoritmos-enrutami ento\Flooding> python a.py connection established Ingrese el Nodo de Destino </pre>	<pre> Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados. Prueba la nueva tecnología PowerShell multi plataforma https://aka.ms/pscore6 PS C:\Users\F. Hengstenberg\Desktop\Cursos\ Semestre 8\Redes\lab3y4\algoritmos-enrutami ento\Flooding> python b.py connection established Ingrese el Nodo de Destino </pre>
		<pre> Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados. Prueba la nueva tecnología PowerShell multi plataforma https://aka.ms/pscore6 PS C:\Users\F. Hengstenberg\Desktop\Cursos\ Semestre 8\Redes\lab3y4\algoritmos-enrutami ento\Flooding> python f.py connection established Ingrese el Nodo de Destino </pre>	<pre> Windows PowerShell Copyright (C) Microsoft Corporation. Todos los derechos reservados. Prueba la nueva tecnología PowerShell multi plataforma https://aka.ms/pscore6 PS C:\Users\F. Hengstenberg\Desktop\Cursos\ Semestre 8\Redes\lab3y4\algoritmos-enrutami ento\Flooding> python f.py connection established Ingrese el Nodo de Destino </pre>

envío del mensaje

```

Windows PowerShell
Copyright (C) Microsoft Corporation. Todos
los derechos reservados.

Prueba la nueva tecnología PowerShell multi
plataforma https://aka.ms/pscore6

PS C:\Users\F. Hengstenberg\Desktop\Cursos\
Semestre 8\Redes\lab3y4\algoritmos-enrutami
ento\Flooding> python a.py
connection established
Ingrese el Nodo de Destino
f
ingrese el mensaje
hola
Ingrese el Nodo de Destino

```

recepción del mensaje

```
-----info-----
nodoFuente a
nodoDestino f
saltos 2
nodosUsados ['a', 'b', 'f']
-----mensaje-----
mensaje hola
-----
```

Discusión:

Los algoritmos de enrutamiento son necesarios para ordenar y facilitar la llegada de mensajes hacia los distintos nodos. los tres algoritmos vistos anteriormente son útiles para casos en especial, esto quiere decir que ninguno es mejor que otro ya que están diseñados para funcionar para distintos tipos de enrutamientos. en el caso del algoritmo Flooding es generalmente utilizado para tramas de descubrimiento, puentes de red o bridges. ya que el algoritmo te garantiza la llegada del mensaje desde el nodo fuente hacia el nodo destino, sin embargo al estar conectado un nodo a varios más, un nodo puede llegar a recibir varios mensajes de distintos nodos que también estén conectados a este, por lo que nos vimos obligados a utilizar un código de mensaje para que el nodo final no reciba varias veces el mismo mensaje y lo muestre al usuario, cuando este solo se mando una vez.

El algoritmo Distance Vector Routing determinó la distancia más corta hacia cualquier ruta, el inconveniente que se tuvo con este algoritmo es su lenta ejecución, ya que examina cada uno de las posibles rutas que se hay, recorriendo todo el nodo.

Link state routing tiene un tiempo de ejecución menor, una de las desventajas es que cada nodo que entra tiene que enviar su presencia a todos los nodos, lo que causa un poco de overhead en la red.

Comentarios Personales:

- Fernando Hengstenberg: Cada algoritmo de enrutamiento vistos, son funcionales en casos en especial, pero para el diagrama de nodos presentado para este ejercicio no es muy eficiente el uso de Flooding ya que el nodo final tiende a recibir muchos mensajes siendo uno solo el que se había enviado, por lo que el algoritmo utilizaría mucho ancho de banda y como utiliza todos los nodos conectados no encuentra la ruta más eficiente para el envío del mensaje.
- Marcos Gutiérrez: Saber el modo de enrutamiento es bueno, pero hubiera sido genial poder poner práctica en la vida cotidiana estos algoritmos y ver cómo es su funcionamiento dentro de las redes.
- Josue Valenzuela: El conocer que hay varias técnicas para la distribución de información en una red es muy útil, usualmente esto se da por sentado en muchas ramas de la ciencias de la computación.

Conclusiones:

- La implementación de los algoritmos será en base a las necesidades de las redes.
- El algoritmo DVR ofrecen el beneficio de utilizar el camino menos costos para el envío de mensajes
- Link State Routing, por su desempeño es el algoritmo más óptimo para ser implementado en la mayoría de aplicaciones