



Universidad del Valle de Guatemala
Facultad de Ingeniería
Departamento de Ciencias de la Computación
CC3067 Redes

Laboratorio 3 - Segunda Parte

Interconexión de nodos

1 Antecedentes

Conociendo a dónde enviar los mensajes para cualquier router se vuelve trivial el envío de mensajes. Únicamente es necesario conocer el destino final y se reenvía al vecino que puede proveer la mejor ruta al destino. Toda esa información es almacenada en las tablas de enrutamiento.

No obstante, con el dinamismo con el que se espera que pueda funcionar el Internet es necesario que dichas tablas puedan actualizarse y acomodarse a cambios en la infraestructura. Los algoritmos con los que se actualizan estas tablas son conocidos como algoritmos de enrutamiento.

2 Objetivos

- Implementar los algoritmos de enrutamiento en una red simulada sobre el protocolo XMPP.
- Analizar el funcionamiento de los algoritmos de enrutamiento.

3 Desarrollo

En la primera parte del laboratorio se desarrollaron los algoritmos de enrutamiento y se probó su funcionamiento de forma hard-coded, es decir, se proporcionó de forma manual la información que cada algoritmo necesitaba tanto para la creación de las tablas de enrutamiento como para el envío de un paquete de un nodo origen a un nodo destino.

En la segunda parte se estará simulando este tipo de infraestructura utilizando como base el chat **alumchat.xyz**, para el envío de los mensajes necesarios para construir las tablas de enrutamiento y la funcionalidad de enviar un paquete de un nodo origen a un nodo destino utilizando dicha tabla.

Los grupos serán los mismos que se conformaron en la primera parte.

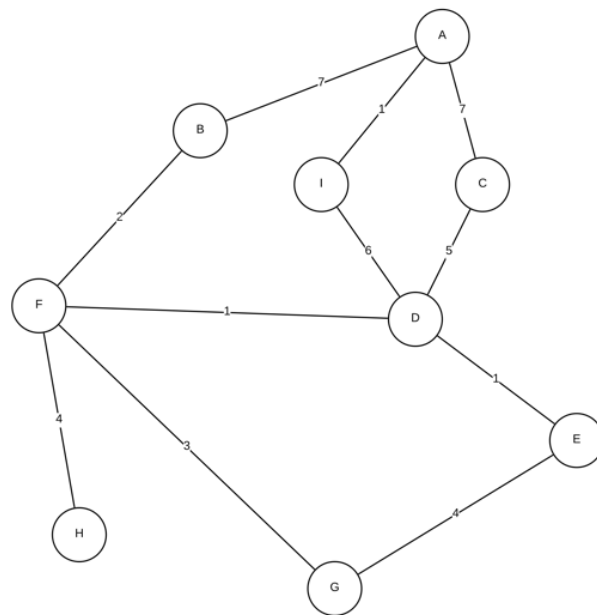


Imagen 1: Mapa de conexiones entre nodos.

En esta propuesta cada uno de los nodos correspondería a un cliente con una dirección @alumchat.xyz a través de la cual puede enviar o recibir mensajes. A manera de ejemplo los nombres podrían ser de la forma:

Nodo	Dirección
A	a@alumchat.xyz
B	b@alumchat.xyz
...	...

3.1 Prueba de los algoritmos (y consideraciones para el Lab 4)

Para probar los algoritmos, se estará asignando una dirección/nombre a cada uno de los alumnos (nodos), quienes usarán su usuario “oficial” de alumchat como JID y credenciales (el usuario es igual a su correo uvg pero con @alumchat.xyz, ejemplo: mnovella@alumchat.xyz). Se estará brindando un archivo con la distribución de nombres (ver Anexo para tal formato).

Los algoritmos a probar son Flooding, Distance Vector y Link State Routing (Dijkstra y Flooding son utilizados en LSR, Dijkstra no se probará directamente).

Adicional, se establecerán mapas de conexiones entre nodos similar al de la Imagen 1. Se estará brindando un archivo con tal topología mencionada al momento de las pruebas (ver Anexo para el formato), la cual deben utilizar para inicializar sus nodos. Cualquiera de los nodos debe de tener la capacidad de enviar y/o recibir un mensaje.

Al iniciar un nodo, este obtendrá la configuración y procederá a descubrir a sus vecinos. El nodo tendrá dos procesos/hilos en simultáneo: el forwarding (escuchar mensajes entrantes, revisar tabla, reenviar) y el routing (escuchar mensajes entrantes con info, actualizar tablas, enviar info, etc.).

Siguiendo el formato establecido en el laboratorio anterior, deberán enviarse y definir distintos tipos de mensajes para el funcionamiento de los algoritmos. Se sugiere un paquete tipo ECHO, para medir delays entre nodos. Se sugiere un paquete DATA, el cual contenga data de usuario (mensajes) en su payload. Se sugiere un paquete TABLE/INFO, el cual contenga información de tablas, ruteo, vecinos, etc.. Pueden agregar otros tipos si así desean y les sirve.

El objetivo es lograr que los algoritmos se estabilicen y los mensajes pasen por los nodos que corresponden, así como el poder responder o adaptarse a nuevos nodos, nodos caídos, etc..

4 Reporte

Al finalizar la actividad debe de realizarse un reporte grupal donde se incluyan las siguientes secciones:

- Nombres y carnés
- Título de la práctica
- Descripción de la práctica
 - o Incluir explicación de los algoritmos utilizados
- Resultados
- Discusión
- Comentario grupal
- Conclusiones

4.1 Rúbrica de evaluación

Elemento		Ponderación	
Código		75%	
	Documentación, orden, comentarios, limpieza, legibilidad/funcionalidad balanceada, etc..		5%
	Implementación de los Algoritmos, de forma eficiente y optimizada. (Parejas: 15 c/u, Tríos: 10 c/u)		70%
Reporte Escrito		25%	
	Encabezado, Ortografía, Formato Adecuado, Descripción de la Práctica		2.5%
	Descripción de los Algoritmos Utilizados y su Implementación		10%
	Resultados		5%
	Discusión		5%
	Conclusiones + Comentarios + Referencias		2.5%

****** Una inasistencia injustificada anula la nota del laboratorio.

\$EXTRA (5%): Aceptar la configuración de forma manual (archivos) y mediante un mensaje de Broadcast del server alumchat.xyz (es decir, desde el server se manda el JSON de configuración como mensaje y el nodo lo recibe y se configura).

Entregar en Canvas:

1. Archivo .pdf con su reporte en grupo
2. Código utilizado para el Laboratorio, y demás, en un zip/rar + link a repositorio

5 Anexo

El formato de los archivos de configuración será acorde al siguiente:

- Topología de la Red:

- archivo: topo-*.txt
- contenido: JSON
- formato ejemplo:

```
{ "type": "topo",  
  "config": { "A": [ 'B' , 'C' ],  
              "B": [ 'A' ],  
              ...,  "D": [ ], ... },  
}
```

- Asignamiento de ID de Nodo:

- archivo: names-*.txt
- contenido: JSON
- formato ejemplo:

```
{ "type": "names",  
  "config": { "A": "foo@alumchat.fun", "B": "bar@alumchat.fun", ... },  
}
```

NOTA IMPORTANTE: No está permitido usar los archivos de configuración para nada más excepto el configurar su propio nodo. Es decir, no pueden usar eso para obtener toda la topología y resolver trivialmente como si fuera Dijkstra puro, de forma estática... en la realidad los routers tienen cables a sus vecinos pero desconocen el resto de la topología, y como pueden ver, los algoritmos a implementar todos son dinámicos.

De igual manera, hacer eso resultará en errores, ya que le quita robustez al algoritmo (cambiamos algo del archivo de nombres, y corremos de nuevo y habrá problemas...). Es el equivalente a que se caiga un nodo, o que se conecte uno nuevo, o que al rato regrese el que se había caído.