

UNIVERSIDAD NACIONAL DE CÓRDOBA

INFORME TRABAJO FINAL

ALGORITMOS Y ESTRUCTURAS DE DATOS

---

# Aplicación del algoritmo de Dijkstra para la simulación del tráfico de paquetes en una red de routers y computadoras

---

*Autores:*

Esteban MORALES (35.104.714)

Adelina MAYOL (34.140.737)

Guillermo VALETTI (35.177.596)

*Supervisores:*

Gustavo WOLFMANN

Rubén AYME

12 de febrero de 2014

# Índice

<b>1. Introducción</b>	<b>2</b>
<b>2. Desarrollo</b>	<b>2</b>
2.1. Primera Fase: Diseño e Implementación del Sistema . . . . .	3
2.1.1. Consigna . . . . .	3
2.1.2. Diagrama de Clases Inicial . . . . .	5
2.1.3. Diagrama de Clases Final . . . . .	6
2.2. Segunda Fase: Implementación del Algoritmo . . . . .	7
2.3. Tercera Fase: Pruebas . . . . .	9
<b>3. Conclusion</b>	<b>9</b>

# 1. Introducción

El siguiente informa, expone el desarrollo del Trabajo final de la materia Algoritmos y Estructuras de datos.

La consigna demanda la implementación del algoritmo de Dijkstra o “del menor camino” a la simulación del trafico de paquetes por una Red de Routers y computadoras, conectados entre si, que responden a una configuración inicial.

El objetivo principal, es obtener el camino mas corto para el recorrido de los paquetes a través de la red mediante el algoritmo de Dijkstra, y asi poder valuar las tablas de cada uno de los routers, para que puedan re dirigir los paquetes con destino a routers vecinos, o bien, armar la pagina si el destino es el propio router.

# 2. Desarrollo

El desarrollo tiene una primera fase donde se interpreta la consigna, para conocer el sistema (red de routers) a simular. Se realizo una descripción e implementación con orientación a objeto, permitiendo modularidad y hacer uso de herencia entre las clases Arco – Conexión. Para ello, fueron necesarios los conocimientos brindados en clase de las estructuras implementadas como Nodo, Lista, Cola, Colas Anidadas, Grafo. Una vez proyectado el sistema, en la segunda fase con la codificamos y adaptamos del Algoritmo de Dijkstra al sistema, dado que el resultado devuelto por el algoritmo, no es un valor de uso directo. Por consigna, sabemos que los routers no son dispositivos autodidactas, es por ello, que existe un 'Administrador' que cumple la función de gestionar, entre otras cosas, la ejecución del algoritmo al inicio y cada 30 ciclos, para restablece los valores de las tablas de re direccionamiento de paquetes de cada Router perteneciente a la red. Es por ello, que a partir del cálculo hecho por el algoritmo, paralelamente se trazan las trayectorias más eficientes, es decir, la secuencia de routers más eficiente. La tercera fase, para verificar y testing el modelo, realizamos 3 casos de uso, corroborando los valores arrojados por el algoritmo, y las tablas calculadas para cada router.

## 2.1. Primera Fase: Diseño e Implementación del Sistema

### 2.1.1. Consigna

Desarrollar un programa que simule el tráfico de datos, al "estilo" del funcionamiento de Internet.

Existen (n) máquinas que cumplen la función de routers: enrutan los datos desde la máquina de origen hacia la máquina de destino.

Existen otras (k) máquinas que son las emisoras - receptoras de páginas. Cada una de estas máquinas está conectada a un único router que es el encargado de enviar / recibir las páginas hacia / desde el destino final.

Cada router está conectado a 1 o más routers. Cada router sabe cuáles son las máquinas finales a las cuales está conectado y cuáles son los routers vecinos que tiene, es decir a que otros routers está conectado directamente.

Además cada router tiene una tabla que le indica a que router enviar los datos con un determinado destino.

Cada router tiene una conexión directa con sus vecinos de un determinado ancho de banda.

Cuando un router recibe de una de sus máquinas terminales un paquete para enviar, este lo divide en n paquetes de igual tamaño y va enviando por la ruta elegida de a un paquete por vez. Es decir que un servicio pedido por una máquina cliente se divide y se envía de a tramos.

A su vez, cuando un router va recibiendo de otro router paquetes con un determinado destino, debe reenviarlo al router correspondiente en la ruta, o bien, si el destino final es una máquina a la cual está conectado directamente, debe ir almacenando los paquetes recibidos hasta que estén todos los que correspondan a la página enviada, rearma la página y recién allí se la envía a la máquina destino.

Las direcciones de las máquinas, son tipo IP, pero simplificadas. Tienen dos partes de 1 byte cada una: la primera indica el router y la segunda la máquina conectada al router. Es decir que pueden haber 256 routers con 256 máquinas cada uno.

Como hace cada router para computar la tabla de destinos que posee?

Si la dirección del paquete corresponde a la de un router vecino, hay una conexión directa, por lo que no hay más trámite. Para routers que no son vecinos pueden haber varias rutas alternativas, debiendo el router elegir aquella que tiene

la menor carga de trafico. Una vez determinada la mejor ruta, todos los paquetes enviados a un determinado destino, se envían al router vecino que conforma el camino elegido.

En resumen, cada router tiene las siguientes funciones:

a) recibir una pagina de una maquina cliente, dividirla en los paquetes que corresponda, y enviarla a la cola de trafico de la ruta que corresponda

b) recibir paquetes de los routers vecinos y redireccionarlos hacia el router que corresponda si la dirección del paquete no es la propia del router, o bien si la dirección del paquete es la del router en cuestión, debe esperar a recibir todos los paquetes que corresponden a la pagina enviada y una vez sucedido esto, enviar la pagina a la maquina de destino.

Cada router tiene una cola de envíos para cada router vecino, en donde van encolando los paquetes que tienen que enviarse por ese canal y envía por cada ciclo, todos los que su ancho de banda le permita. En la cola no se deben colocar todos los paquetes de una pagina consecutivos: deben ser intercalados con los paquetes que provengan de otra maquina, para que se vayan enviado parcialmente de todas las maquinas al mismo tiempo. Esto evita que un envío muy pesado atore al server y los otros paquetes demoren mucho en ser enviados.

Existe un administrador del sistema que de vez en cuando recomputa las rutas optimas de todos los routers. Para ello cada router le envía el tamaño de la cola de espera de envíos de paquetes hacia cada router vecino, y con ello el administrador determina la ruta optima pasando por los routers que tengan menor trafico pendiente. Hay que tener en cuenta que cada router envía (k) paquetes por vez en un canal, según el ancho de banda que tenga el canal.

Para determinar el optimo, lo que importa es la cantidad de ciclos que un nuevo paquete debe esperar hasta ser enviado. Ademas se pierde un ciclo al entrar a un router y volver a salir.

O sea que si un router tiene la cola vacía, no tiene un costo de cero, si no de uno, porque el paquete debe esperar hasta el próximo ciclo para ser reenviado.

Una vez que el administrador determina los caminos óptimos, se los informa a cada router.

Esos caminos son utilizados a partir de ese momento hasta que vuelven a recomputarse.

Puede pasar que los paquetes pendientes de enviar de una pagina, utilicen un

camino distinto de los enviados previamente, porque se cambio el camino a utilizar por uno con menos trafico.

El caso es el así: la pagina se dividió en 50 paquetes. Se enviaron 20. Se re-computa el camino optimo y se cambia de ruta. Los 30 paquetes restantes van por otra ruta, que al ser tomada como optima, se pueden llegar a destino antes que los primeros 20. Tener esto en cuenta cuando el router debe rearmar la pagina.

Usted deberá simular todo esto. Para ello construirá un ciclo en se le de el turno a todos los routers para que hagan las tareas que tienen que hacer por vez. Cada 30 ciclos, tomara el control el administrador para recomputar los caminos óptimos y volverán a computarse los ciclos.

Deberá utilizar números aleatorios para simular la generación de paginas a ser enviadas, el destino y el tamaño de cada pagina.

La cantidad de routers, la cantidad de terminales por router, las conexiones directas de los routers y el ancho de banda entre los routers y entre cada terminal y el router asociado deberá ser configurable y definido en un archivo que se utilizara para parametrizar el sistema.

Dada la magnitud del trabajo, aconsejo, diciendo esto con un tono de casi imposición, trabajar en equipos de 2 o 3 personas para dividirse el trabajo. También aconsejo tomarse al menos una semana para pensar como armar el sistema, haciendo un análisis detallado en profundidad de los requerimientos y de la implementación de la solución, antes de programar una sola linea de código.

### **2.1.2. Diagrama de Clases Inicial**

De la interpretación y análisis de la consigna, obtuvimos un perspectiva general del sistema, el cual es representado a partir del siguiente diagrama de clases.







- Conjunto Adyacentes: conjunto de nodos adyacentes al nodo inicio
- Vector Predecesores: guarda los valores de los router predecesores a cada router
- Vector Distancias: almacena las distancias de los caminos mas cortos a cada router
- Lista de Etiquetas: contiene las etiquetas correspondiente a cada uno de los routers, con los valores en el siguiente formato  
`{ r_destino, r_siguiente, peso del trayecto }`

Primeramente, cargamos en conjunto Q , Adyacentes, las etiquetas, haciendo distinción para la etiqueta del nodo inicio `{ orden_nodo_inicio, 0,0 }` y las demás `{ orden_nodo_x, -1, INF }`. Luego recorremos en conjunto Q buscando en las etiquetas de cada uno de los routers, la que contenga el menor trayecto. Con esta condición, seleccionamos el nodo actual o elegido, y lo pasamos al conjunto S. Luego recorremos los arcos, buscando aquellos que tengan origen en el nodo actual, y que su destino no corresponda a un nodo contenido en S. Es entonces, cuando realizamos la comparación de distancias (trayectos almacenados en las etiquetas), con la distancia resultante de la suma de la distancia del trayecto desde el nodo inicio, al nodo actual, las la del trayecto del nodo actual al siguiente para determinar si actualizar o no los valores tanto en el peso de trayectoria, perteneciente a la etiqueta del router siguiente, como en el Vector Distancias y Predecesores.

Seguidamente, normalizo y actualizo los valores en el Vector Predecesor, con los correspondientes valores de nodos adyacentes al nodo inicio, lo que nos permite posteriormente, armar las tablas para dicho router. Finalmente, Se actualiza el campo `r_siguiente` de la etiqueta con el valor correspondiente, calculado en dicho Vector de Predecesores. Terminando así la ejecución del algoritmo, devolviendo la tabla con los nuevos valores de redireccionamiento.

Que diferencia tiene con Floyd-Warshall & Dijkstra? Ambos algoritmos tienen la misma finalidad, obtener el calculo del camino mas corto. Mas la diferencia es que Dijkstra calcula los caminos partiendo desde un único nodo base, hasta los demás del grafo, mientras Floyd aplica lo mismo, pero para todos los nodos de

la red. Es decir, si el costo de calculo de Dijkstra es  $n \cdot n$  operaciones, el costo de FFloyd-Warshall será  $n \cdot n \cdot n$

### 2.3. Tercera Fase: Pruebas

Para cada prueba, tomar: 1) gráfico de la red 2) valores arcos iniciales 3) Vector Distancias 4) Vector Predecesores 5) Tablas de Routers

## 3. Conclusion

Concluimos que el algoritmo de Dijkstra es una herramienta eficiente para gestionar el trafico a través una red de nodos o afines. También hicimos uso de estructuras anidadas, Cola - Cola - Cola (Paquetes), lo cual permite organización en diferentes niveles e igualdad de prioridades, algo muy útil para el sistema.

## Referencias

- [1] CORMEN, T. C ; LEISERSON, C.E ; RIVEST, R.L and STEIN, C *Introduction to Algorithms*, Third Edition The MIT Press. Cambridge, Massachusetts London, England, 2009 *Single-Source Shortest Paths*, 24:658–662.
- [2] <http://www.cplusplus.com/forum/articles/10627/> *Headers and Includes: Why and How* 2009
- [1] <http://choorucode.com/2010/07/22/c-fixing-cyclic-dependencies/> *C++: Fixing Cyclic Dependencies* 2010