

ARQUITECTURA DE REDES, SISTEMAS Y SERVICIOS

Conjunto de problemas 6

1. Emplee el algoritmo de Bellman-Ford para generar la ruta de menor coste desde el nodo A a todos los nodos de la Figura . Emplee un formato similar a la Tabla . Repita la actividad para cada nodo de origen, dibujando el árbol de expansión resultante. Con los resultados describa la tabla de rutas de cada nodo donde para cada destino, ese nodo almacena el sucesor a emplear en el reenvío. Emplee esas tablas para comprobar el camino que sigue un paquete desde un origen cualquiera a un destino cualquiera de la topología. Repita el ejercicio empleando el algoritmo de Dijkstra y un formato similar a la Tabla . Verifique los resultados mediante un programa.

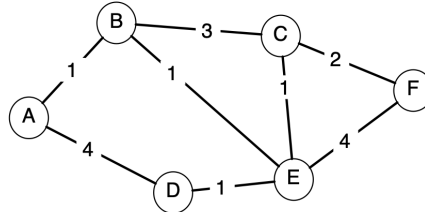


Figura 1

Tabla 1 - Tabla para cálculo de Bellman-Ford

Desde origen X	L _h (destino) Sucesor									
h	A	B	C	D	E	F	G	H	I	J
0										
1										
...										

Tabla 2 - Tabla para cálculo de Dijkstra

L(n)										
A	B	C	D	E	F	G	H	I	J	

2. Repita el problema 1 para las topologías de las figuras 2-5.

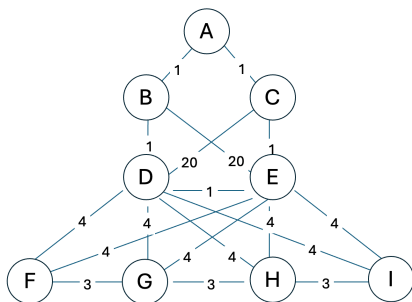


Figura 2 (a)

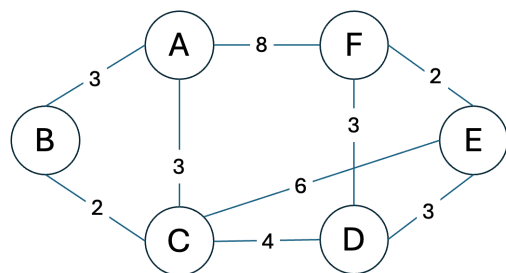


Figura 3 (b)

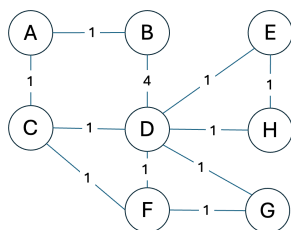


Figura 4 (c)

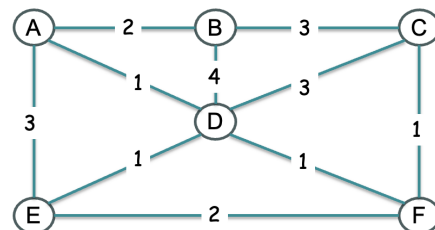


Figura 5 (d)

3. Suponga la red de la Figura 1. Se emplea un algoritmo *Distance Vector* distribuido. Cada nodo envía periódicamente a todos sus vecinos su vector de distancias, es decir, el coste que tiene calculado en ese momento a cada destino. Cuando recibe el vector de distancias de un vecino lo almacena (sustituyendo el que ya tuviera de ese vecino) y recalcula su tabla de rutas. Para este cálculo compara las distancias a cada destino por cada uno de los vecinos empleando los vectores de distancias que tiene almacenados, uno de cada vecino, y los costes de sus enlaces a ellos, y escoge la mejor, empleando la ecuación de Bellman-Ford. Como resultado apunta en la tabla de rutas como sucesor para ese destino el nodo adyacente que le anunció el coste escogido.

Cada nodo envía su vector de distancias (el cual no incluye a los sucesores) a todos sus vecinos cada 30 segundos de forma periódica, independientemente de que haya hecho cambios o no en su vector al aprender nueva información. Los nodos conocen desde el comienzo la existencia de todos los destinos posibles (A, B, ... F), de forma que el vector de distancias tiene un tamaño fijo con una posición prefijada para el coste a cada destino. Todos los nodos al arrancar el programa de encaminamiento tienen coste infinito a todos los destinos salvo a sí mismo (coste 0) y conocen el coste de cada enlace que tienen. Los costes son independientes del sentido.

El nodo A arranca el programa de encaminamiento en el instante 0, momento en el que hará el primer envío del vector de distancias a todos sus vecinos. A partir de ahí lo volverá a enviar (con los cambios que le haya aplicado) en los instantes 30s, 60s, 90s, etc, sin detener nunca este proceso. El nodo B comienza en el instante 1s, el nodo C en 2s, D en 3s, E en 4s y F en 5s, con lo que la segunda ronda de envíos será en los instantes 30s, 31s, 32s, 33s, 34s y 35s, y así sucesivamente. Un nodo cambia el sucesor para un destino solo en el caso de que mejore el coste al destino, es decir, si se empata con el aprendido con anterioridad no se cambia de sucesor. Si el mejor coste calculado es por el mismo sucesor que se tenía aprendido, el coste se actualiza, aunque se haya incrementado respecto al anterior aprendido.

- Calcule las tablas de rutas de cada nodo, llevando a cabo las iteraciones del algoritmo hasta que las tablas se estabilicen.
 - Suponga un instante $t = 30n + 10$, donde n es un número natural lo suficientemente grande para que t sea posterior a la estabilización de todas las tablas de rutas. En el instante t se desconecta el enlace entre los nodos B y E. Esto invalida en B y E todos los costes cuyo sucesor fuera por ese enlace, pasando de nuevo a tener coste infinito. Como ambos nodos tienen guardados los vectores anunciados por el resto de vecinos calculan en ese mismo instante un sucesor alternativo para los destinos afectados por el cambio. Continúe las iteraciones del algoritmo hasta alcanzar una nueva estabilización de las tablas de rutas.
 - De forma análoga al apartado b suponga el fallo de otro enlace cualquiera y repita el cálculo.
 - De forma análoga al apartado b suponga el fallo de un nodo cualquiera y repita el cálculo.
 - Repita el ejercicio con cualquiera de los grafos de las figuras 2-5.
4. Se ha establecido una red óptica entre ciudades europeas como se muestra en la Figura 6. En cada enlace de la red se utiliza WDM (*Wavelength Division Multiplexing*), lo que permite usar 8 longitudes de onda para transmitir en 8 canales simultáneamente, cada uno de ellos a 20Gb/s. El efecto es que cada enlace de la figura lo podemos modelar como 8 enlaces en paralelo, cada uno con esa tasa de transmisión.

Una de las longitudes de onda de cada enlace (la llamaremos longitud de onda de control) se utiliza para comunicar entre sí a las unidades de control de cada nodo, de forma que puedan enviarse mensajes entre ellas, que serán procesados por cada una. Creamos así una red de conmutación de paquetes con la misma topología, empleando los canales de control y solo para este tipo de tráfico.

La red va a explotarse utilizando conmutación de circuitos. Las 7 longitudes de onda restantes son recursos que se pueden utilizar para este fin, estableciendo circuitos entre los nodos que lo soliciten, reservando para un circuito una longitud de onda en cada enlace del camino. El tráfico de ese circuito no es desencapsulado en los nodos del camino, sino que entra y sale del nodo con un procesamiento exclusivamente en capa física (solo se cambia su longitud de onda y se amplifica y refleja la luz hacia el puerto de salida). Estos circuitos permiten a la empresa propietaria de la red ofrecer enlaces de 20Gb/s entre sus clientes, que están situados en las ciudades de la figura.

Al establecer un circuito, por ejemplo, entre dos empresas situadas en Madrid y Berlín, se decide un camino para el circuito y se solicita a los nodos del camino que reserven una longitud de onda en cada enlace necesario para establecer el camino, conectando un camino óptico desde el origen hasta el destino. Para decidir el camino de un circuito se utiliza un algoritmo centralizado. El nodo origen, que tiene la petición del circuito, envía un mensaje a través de la longitud de onda de control hasta llegar a la unidad de encaminamiento que se encuentra en el nodo de París. La unidad de encaminamiento mantiene el estado de todos los circuitos establecidos. Cuando recibe una petición, calcula el mejor camino para el circuito solicitado y envía mensajes a los nodos de ese camino por el canal de control para indicarles que reserven recursos. Finalmente, informa al nodo de origen, con lo cual el circuito queda establecido. Cuando un nodo decide liberar un circuito envía un mensaje a la unidad de control, que informa a los nodos implicados de que pueden liberar los recursos correspondientes a ese circuito. Toda la comunicación por la red de control se hace mediante un mecanismo de inundación con TTL.

En la selección del camino se utiliza el algoritmo de Dijkstra, teniendo en cuenta los recursos ocupados por los circuitos ya establecidos. Para ello, se modela cada enlace entre dos nodos como un solo enlace en el grafo, pero se modifican los pesos de los enlaces cada vez que se establece un circuito, teniendo con ello en cuenta la cantidad de circuitos que están pasando por cada enlace (o la cantidad de longitudes de onda libres). El peso de cada enlace es $w = \infty$ si todos los recursos de un enlace están ocupados. Si hay recursos disponibles es $w = 1 + n$, donde n es el número de circuitos que pasan por ese enlace. De esta forma los enlaces con más ocupación tienen menos tendencia a ser utilizados. Todos los enlaces y circuitos son bidireccionales.

En un instante dado, la red tiene establecidos los circuitos que se muestran en la Tabla 3. En ese momento, llega una petición para un establecimiento de circuito entre Berlín y Madrid. Muestre la ejecución del algoritmo en la unidad de encaminamiento calculando el camino. Dibuje el árbol de expansión que se calculará y muestre el camino del circuito que se establecerá como resultado.

Una vez establecido el circuito de la pregunta anterior llega una petición para un nuevo circuito entre Berlín y Madrid. ¿Qué camino seguirá el segundo circuito?

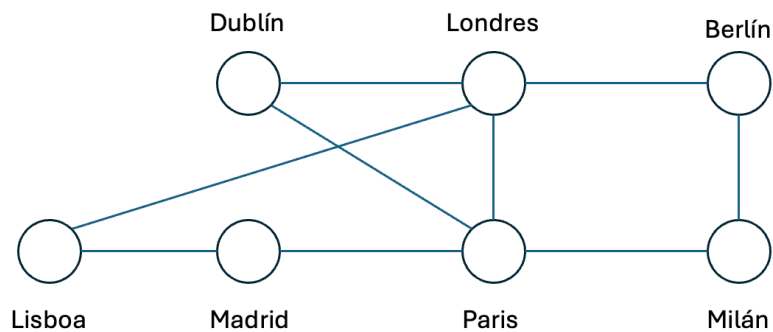


Figura 6

Tabla 3 - Listado de circuitos

Circuito 1	Dublín – Londres – Lisboa
Circuito 2	Dublin – Londres – Berlín
Circuito 3	Dublín – Londres – Berlín – Milán
Circuito 4	Dublín – París – Madrid
Circuito 5	Lisboa – Madrid
Circuito 6	Madrid – París – Milán
Circuito 7	Madrid – París – Milán – Berlín
Ciurcuito 8	Madrid – París – Londres – Berlín

5. Una empresa tiene la red que se muestra en la Figura 7, donde los equipos que se comunican están asociados a los nodos A, F, G y L. Se desea que el tráfico siga los caminos que se ven en la Figura 8. Los caminos principales que se quieren obtener son

- Entre A y F: $A \leftrightarrow B \leftrightarrow C \leftrightarrow D \leftrightarrow E \leftrightarrow F$
- Entre A y L: $A \leftrightarrow B \leftrightarrow C \leftrightarrow J \leftrightarrow K \leftrightarrow L$
- Entre G y F: $G \leftrightarrow H \leftrightarrow I \leftrightarrow D \leftrightarrow E \leftrightarrow F$
- Entre G y L: $G \leftrightarrow H \leftrightarrow I \leftrightarrow J \leftrightarrow K \leftrightarrow L$

Los pesos en los enlaces pueden tener un valor entre 1 y 10. Configure los pesos en cada uno (simétricos) para lograr los caminos deseados mediante un algoritmo de selección del camino de menor coste.

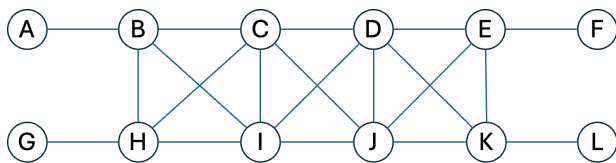


Figura 7

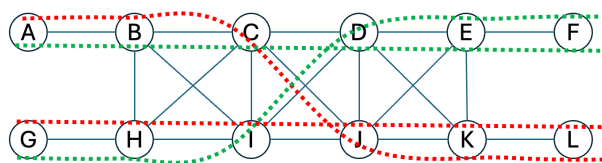


Figura 8

6. En emplea el algoritmo de Dijkstra para calcular un árbol de coste mínimo con raíz en el nodo D de la topología de la Figura 9. En un punto del cálculo en el árbol se han introducido ya los nodos B, C, D, E y F.

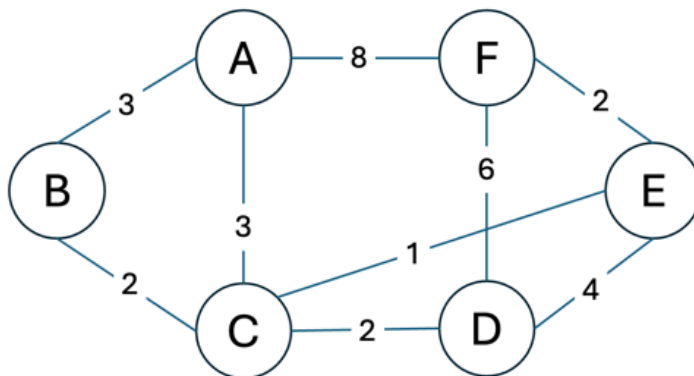


Figura 9 - Topología del problema

Indique qué enlaces se habrán añadido al árbol al añadir esos nodos y cuáles ya nunca se añadirán. En el siguiente paso se añade el nodo A. Explique qué enlace se añade con él y por qué.

7. Calcule el árbol de costes mínimos a partir del grafo de la Figura 10 empleando al nodo de nombre *Nami* como raíz. Los enlaces en línea gruesa tienen un coste de 1 y los de línea fina un coste de 10.

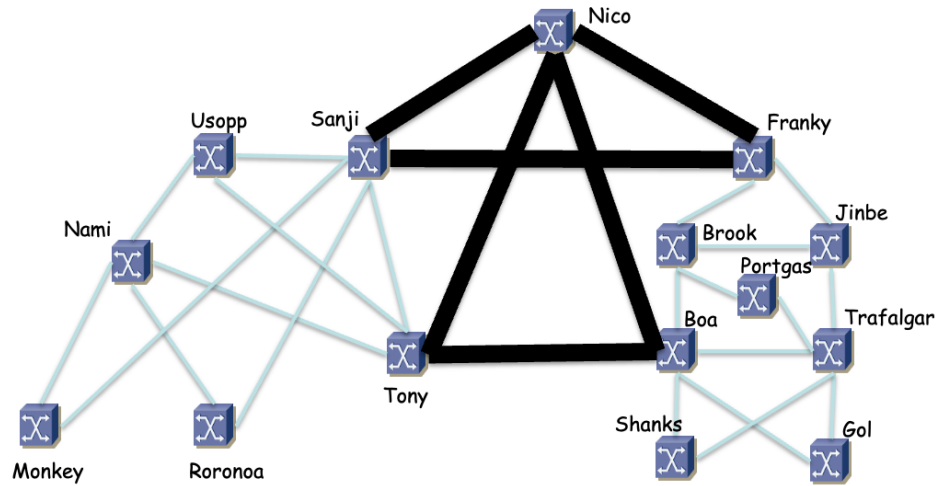


Figura 10 - Topología del problema