

Reporte: Desarrollo de la Parte 3 – Implementación de Dijkstra con GUI

Objetivo

Implementar un sistema que:

- Calcule la ruta óptima para transferir archivos entre nodos de una red virtual (VPN), utilizando el algoritmo de Dijkstra.
- Permita al usuario seleccionar archivos y transferirlos mediante una GUI.
- Compare la ruta óptima con la ruta directa, midiendo los tiempos de transferencia.
- Verifique la recepción del archivo y reporte el tiempo redondo total.

Fase 1: Estructura del Grafo

Se utilizó un grafo ponderado donde los nodos representan dispositivos en la VPN y las aristas representan el **ancho de banda disponible entre nodos**.

Para que Dijkstra priorice el **mayor ancho de banda**, se invirtió la lógica usando pesos = $1 / \text{ancho_de_banda}$. Así, el algoritmo elige caminos más rápidos (con mayor ancho de banda).

Fase 2: Implementación de Dijkstra

Se utilizó una cola de prioridad (heapq) para implementar Dijkstra y encontrar el camino de menor costo desde un nodo origen hasta uno destino.

Salidas:

- Costo total
- Ruta óptima como lista de nodos (ej. ['A', 'B', 'D'])

Fase 3: Transferencia Segmentada por Ruta

En lugar de enviar el archivo directamente, se creó una función para que **cada nodo en la ruta actúe como "puente"**:

1. El archivo se transfiere del nodo A al nodo B.
2. Luego, B a C.
3. Y así sucesivamente, hasta el destino.

Esto simula el comportamiento de una red de túneles entre nodos.

Técnica utilizada: paramiko + sftp.put() para transferencias seguras por SSH.

Fase 4: Comparación con Ruta Directa

Se añadió una segunda función que transfiere el archivo directamente del nodo A al nodo destino (sin pasar por los intermedios).

Ambos tiempos de transferencia (óptimo y directo) se registran y muestran.

Fase 5: Verificación de Recepción

Se implementó una función que accede al nodo destino y revisa si el archivo existe correctamente en /tmp/, lo que confirma una recepción exitosa.

Fase 6: Cálculo de Tiempo Redondo

Se suman los tiempos de transferencia óptima y directa para ofrecer una métrica de tiempo redondo.

Fase 7: Interfaz Gráfica (GUI)

Con tkinter, se creó una GUI que incluye:

- Selección del nodo destino mediante un Combobox.
- Selección de archivo local.
- Botón para iniciar la transferencia.
- Visualización de:
 - Tiempo óptimo
 - Tiempo directo
 - Tiempo redondo
 - Alertas de éxito o error

Conclusión

La versión final del código cumple con todos los requerimientos de la Parte 3 del proyecto:

- Utiliza Dijkstra para seleccionar la mejor ruta.
- Implementa transferencia segmentada entre nodos.
- Mide y compara tiempos reales.
- Ofrece una interfaz gráfica intuitiva y funcional.
- Confirma la recepción del archivo y reporta el tiempo redondo.