Redes de Computadores 2022-1
Proyecto Parte 2: Dijkstra and Smart OSPF

Profesor: David F. Celeita R.

Objetivo:

Diseñar un programa que calcule la ruta más corta de forma dinámica basado en algoritmos Link-State escalable para grafos aleatorios de máximo 50 nodos.

Entrega:

- Póster (5%) – Jueves 12 de Mayo
  *La entrega del poster debe estar antes para poder imprimirlos. Tendremos el cierre del curso de forma presencial con Pizza o donuts ☺
- Paper / Scripts (10%) - Viernes 27 de Mayo (Entrega final del proyecto por E-Aulas)
- Modalidad: Grupos de trabajo.
- Metodología: Observar y aprender haciendo (Hands-on)

Contexto:

Tiempo de desarrollo: 1 mes (Hasta el 27 de Mayo)

2 sesiones de seguimiento de trabajo en grupo (Jueves 28 de Abril y Jueves 5 de Mayo).

Tema: Herramienta de cálculo y visualización de ruta más corta dinámica basado en LS.

Herramienta de trabajo: Python (preferible) / Matlab

Requisitos de entrada:

- El usuario puede seleccionar la creación de un grafo aleatorio de:
  - Número de Nodos $15 \leq \mathrm{n} \leq 50$ (etiquetados con números de 15 a 50)
  - Número de arcos dirigidos $\mathrm{m} = \mathrm{n} + (\mathrm{n}/2)$.
- El peso de cada arco estará relacionado con la congestión del enlace el cuál debe asignarse de forma aleatoria entre $1 \leq C_{ij} \leq 15$. Con esto se debe construir la matriz dispersa estipulando los pares de nodos conectados con cada arco y una columna que defina el costo del enlace. (Node A || Node B || Cost A-B) (Se asume el primer nodo, como nodo de Origen).
- El usuario también puede como segunda opción cargar un archivo CSV o TXT donde se especifique la matriz dispersa del grafo incluyendo costos. (Se asume el primer nodo, como nodo de Origen).
- Una vez cargados los datos, el usuario debe ejecutar el algoritmo (Botón o comando RUN).

Pesos dinámicos:

Con el grafo original, el usuario puede cambiar el peso de los arcos de forma manual (un arco en particular) o automática (aleatoria conservando el requisito entre $1 \leq C_{ij} \leq 15$).

Una vez modificado un peso, el reto se encuentra en vincular la filosofía de un OSPF inteligente que por medio de propagación de Link State Advertisement permita actualizar las tablas de enrutamiento individuales.

La solución puede incluir Machine Learning para tratar de predecir los pesos y mejorar la eficiencia de respuesta ante un cambio en los arcos.

Requisitos de salida:

El programa debe entregar un reporte con:

- Figura del grafo en cada instante de cambio de alguno de los pesos (máximo 10 cambios en el grafo)
- El número de iteraciones ejecutadas totales.
- La tabla de enrutamiento final.
- Figura del grafo (árbol) final con las rutas más cortas desde el nodo de origen

Entregables (Valoración de proyecto parte 2 - 15%):

Una carpeta con los siguientes elementos

Entregable 1 (5%):

- **Poster** estilo presentación de publicación en conferencia en inglés (Archivo de plantilla manejado en el Proyecto parte 1). **Jueves 12 de Mayo**

Entregable 2 (10%):

- **Paper estilo IEEE Transactions** en español (Archivo de Plantilla Latex adjunto) – Abstract (en inglés) + Palabras clave (en inglés) + Introducción + Estado del arte + descripción detallada de scripts, incluyendo diagrama de flujo del programa + Resultados + Conclusiones + Referencias. (30%)
- **Scripts** funcionando
- **Viernes 27 de Mayo**